High-Precision Grasping and Placing for Mobile Robots

Dissertation

for obtaining the academic title

Doctor Engineer (Dr.-Ing.)

at the Faculty of Computer Science and Electrical Engineering

at the University of Rostock

Prepared at the Center for Life Science Automation (celisca)

Submitted by:

Mohammed Myasar Ali, born on 30th October 1982 in Mosul, Iraq

Rostock, Germany, 2017

urn:nbn:de:gbv:28-diss2017-0100-6

Printed with the support of German Academic Exchange Service
Deutscher Akademischer Austauschdienst (DAAD)

Reviewers:

1. Reviewer:

   Prof. Dr. -Ing. habil. Kerstin Thurow

   Institute of Automation, University of Rostock, Germany

2. Reviewer:

   PD Dr.-Ing. habil. Heidi Fleischer

   Institute of Automation, University of Rostock, Germany

3. Reviewer:

   Prof. Dr.-Ing. Egon Müller

   Faculty of Mechanical Engineering, TU Chemnitz, Germany

Date of Submission: 24.01.2017

Date of Defense: 20.06.2017

Acknowledgement:

The work in this dissertation would not have been completed without the precious assistance and guidance of several special people in a myriad of ways.

First and foremost, I am sincerely thankful to my supervisors Prof. Kerstin Thurow and Prof. Norbert Stoll for their abundant support, prolific advices, and for all that I could learn from their experience. Moreover, for the facilities and possibilities they accorded in the working environment. I truly appreciate their infinite patience and trust in my capabilities that gave me the courage to go forward.

Also, I would like to thank the group of mobile robotics and especially the group leader PD Dr.-Ing. habil. Hui Liu for the help and support during the preparation of this work.

Furthermore, I would like to extend my warm thanks to my colleagues in celisca and the Institute of Automation who I owe for their cooperation and support. Especially, I would like to thank Dr.-Ing. Steffen Junginger, Dipl.-Ing. Lars Woinar, Dr.-Ing. Thomas Roddelkopf, Heiko Engelhardt, and PD Dr.-Ing. habil. Heidi Fleischer.

Particularly, I would like to take this opportunity to express my deepest gratitude and appreciation for those who have played the larger role in the motivation, moral support, and encouragement during the hard times, Prof. Mohit Kumar and Mr. Mohammed F. Ruzaij.

Special thank goes to the German academic exchange service (DAAD) for the financial support during my study.

Last but not least, I would like to dedicate this dissertation to my family: my deceased father and younger brother who their memories will remain with me eternally, my mother, and old brothers for their sacrifices and great role in my life.

Table of Contents

1 Introduction 1

1.1 Mobile Robot Requirements 1

1.2 Autonomous Mobile Robots 2

1.3 Application Areas of Mobile Robots 2

1.4 The Required Sensing for Arm Manipulation 3

1.5 The Control of the Mobile Robotic Arm 5

1.6 Motivation and Goal 6

1.7 Dissertation Outlines 9

2 Literature Overview 11

2.1 State of The Art in Object Manipulation 11

2.2 Kinematic Analysis 27

2.2.1 Forward Kinematics 27

2.2.1.1 The Transformation Matrix 27

2.2.1.2 The Denavit-Hartenberg Representation 29

2.2.2 Inverse Kinematics 32

2.3 Visual Processing 35

2.3.1 Camera Calibration 36

2.3.2 Object Detection and Feature Extraction 36

2.3.3 Pose Estimation for the Detected Object 40

2.3.3.1 Stereo Vision 40

2.3.3.2 Structured Light and Time of Flight with Kinect Sensor 43

2.3.3.3 Pose Estimation Based Single Camera 45

2.4 Feedback Control 46

2.4.1 Joint Sensors 46

2.4.2 Visual Servoing 47

2.4.3 Force Sensors 49

2.5 H20 Mobile Robot 50

2.5.1 H20 Operation Strategy 51

2.5.2 H20 Localization System 52

Table of Contents

3 The Work Concept 55

3.1 The Implementation Strategy 56

3.2 The Concept of Kinematic Analysis 58

3.2.1 Solvability 58

3.2.2 Workspace Analysis and Reachable Space 59

3.2.3 Kinematic Singularities 59

3.2.4 Trajectory Selection and Estimation 59

3.2.5 Repeatability and Accuracy 60

3.3 The Concept of Visual Perception 61

3.3.1 Object Detection and Recognition 61

3.3.1.1 Color Detection and Segmentation 61

3.3.1.2 Shape Detection 63

3.3.1.3 Edges and Corners Detection 63

3.3.1.4 Object Recognition Based on Local Features 64

3.3.2 Object Pose Estimation 64

3.4 The Manipulation with Unstable Arms 66

3.5 Client-Server Communication 68

4 The Solution of Kinematic Problem 71

4.1 H20 Arms' Kinematics and Specifications 71

4.2 Forward Kinematic Solution with Validation 73

4.3 Inverse Kinematic Solution 74

4.4 The Validation of Inverse Kinematic Solution 81

4.5 Workspace 83

4.6 The Selecting of the desired IK Solution 84

5 Blind Manipulation and Gripper Design 85

5.1 The Related Problem 85

5.2 The Repeatability of H20 Arms 86

5.3 Blind Manipulation using lists of joints' values 88

5.4 Blind Manipulation using Kinematic Analysis 89

5.4.1 Angle to Servo Position Conversion 90

5.4.2 Accuracy and Repeatability of the H20 Arms 90
5.4.3 Client-Server Model for Manipulation 92
5.4.4 Blind Labware manipulation 93
5.5 The Design of Grippers and Labware Containers 96
5.6 Labware Holder on the Robot Body 99
6 Visual Labware Manipulation 101
6.1 Handle Detection using Kinect Sensor V1 101
6.2 Extrinsic Calibration 104
6.3 Manipulation of New Handle Using Kinect V2 106
6.4 Labware Identification and Manipulation 111
6.4.1 Labware Identification and Position Estimation 112
6.4.2 Multiple Labware Manipulation 124
6.5 Labware Orientation 129
6.6 Related Work 130
6.6.1 Pushing the Elevator Buttons 130
6.6.2 Object Tracking using Head Control 131
7 Summary and Outlook 135
7.1 Summary 135
7.2 Outlook 137
8 Bibliography 139

List of Figures

Figure 1.1: Mobile robot in grasping task in the hospital. 4
Figure 1.2: 6-joints robotic arm. 5
Figure 1.3: Overall structure of management seystem in life science labs. 7
Figure 1.4: The developed aspects in the H20 robot system. 7
Figure 2.1: Different design of robotic arms. 11
Figure 2.2: UR5 universal robot arm. 13
Figure 2.3: An intelligent service robot with Kinect sensor. 14
Figure 2.4: Robotic arm with fingertip force sensor and Kinect. 15
Figure 2.5: Robotic arm with visual feedback system. 16
Figure 2.6: Mitsubishi RV-3AL industrial robot. 17
Figure 2.7: PR2 robot with Kinect sensor mounted on the head. 18
Figure 2.8: Robotic arm with webcam and mobile platform. 18
Figure 2.9: Arm manipulation for objects with different colors. 19
Figure 2.10: Object grasping using a dexterous robotic arm, ADAM, and a stereo camera. 20
Figure 2.11: Cosero mobile robot grasps a spoon using Kinect sensor. 21
Figure 2.12: Service robot in complex environment for manipulation task. 22
Figure 2.13: A mobile robot for assistance in the kitchen environments. 22
Figure 2.14: Barrett WAM robotic arm in the process of grasping an object. 23
Figure 2.15: LWR robot with hand camera in grasping task. 24
Figure 2.16: 3-DOF robotic arm with encoders and stereo camera 25
Figure 2.17: The humanoid robot ARMAR in a kitchen environment for grasping tasks. 25
Figure 2.18: An example of simple robot arm with one joint. 28
Figure 2.19: Transformation matrix. 28
Figure 2.20: Roll, pitch, yaw orientation system. 28
Figure 2.21: D-H conventions for frame assigning. 30
Figure 2.22: An example of 3-joints arm with its D-H parameters. 31
Figure 2.23: Kinematic analysis. 32
Figure 2.24: 6-DOF Puma robot arm. 33
Figure 2.25: RGB color space. 37
Figure 2.26: HSV color space. 38
Figure 2.27: Different techniques for edge detection. 38
Figure 2.28: Harris cornere detection. 39
Figure 2.29: Matching between the captured image and the sample using SURF. 39
Figure 2.30: Classification of depth measurements techniques. 40
Figure 2.31: The removing of lens distortion. 41
Figure 2.32: Images rectification. 41
Figure 2.33: Finding the corresponding point between two images. 42
Figure 2.34: The 3D point in the image. 42
Figure 2.35: The 3D point in the stereo images. 43
Figure 2.36: Old version of Kinect sensor (V1). 44
Figure 2.37: IR projection of kinect sensor. 44
Figure 2.38: Intel RealSense 3D Camera (F200). 45
Figure 2.39: New version of Kinect sensor (V2). 45

List of Figures

Figure 2.40: A six degrees of freedom robot using a stereo vision. 48
Figure 2.41: Pressure sensors on the grippers. 49
Figure 2.42: H20 Mobile robot with its specifications. 51
Figure 2.43: Robot outside Server/Client communication architecture. 52
Figure 2.44: IR projector with passive landmark. 52
Figure 2.45: (A) Stargazer sensor, (B) Passive landmarks. 53
Figure 2.46: Localization sensor and ceiling landmarks. 53
Figure 3.1: The structure of working concept. 55
Figure 3.2: Architecture of control strategy for the robotic arm manipulation. 57
Figure 3.3: The block diagram of manipulation process. 57
Figure 3.4: Simple arm with obstacle. 60
Figure 3.5: Object detection steps using HSV color space. 62
Figure 3.6: The field of view for Kinect camera V1. 65
Figure 3.7: The depth range of kinect sensor V1. 65
Figure 3.8: Robotic arm with Eye-in-Hand. 67
Figure 3.9: Object and end effector detection using color information. 68
Figure 3.10: Sequential diagram for client – server model. 69
Figure 4.1: H20 arms structure and coordinate frames. 71
Figure 4.2: The position and orientation of the end effector and the simulation plot of arm according to the configuration [0º, -90º, -90º, 0o, 90º, 0º]. 74
Figure 4.3: The position and orientation of the end effector and the simulation plot of arm according to the configuration [0º, -90º, -90º, -90º, 90º, 0º]. 74
Figure 4.4: The position and orientation of the end effector and the simulation plot of arm according to the configuration [0º, -180º, -90º, -90º, 90º, 0º]. 74
Figure 4.5: Singularity case where the 3rd and 5th joints are aligned. 79
Figure 4.6: Singularity case where the 1st and 3rd joints are aligned. 80
Figure 4.7: Singularity case where the 1st, 3rd, and 5th joints are aligned. 81
Figure 4.8: The validation process for the IK solution. 82
Figure 4.9: The validation of IK model according to the configuration [50º, -90º, -90º, -30º, 180º, 10º]. 82
Figure 4.10: The validation of the IK model according to the configuration [30º, -120º, -45º, -23º, 180º, 30º]. 83
Figure 4.11: The validation of the IK model according to the configuration [9º, -90º, -23º, -90º, 126º, -30º]. 83
Figure 4.12: The workspace envelope of the H20 arms. 84
Figure 5.1: The H20 robot grasps the labware container. 85
Figure 5.2: Stargazer sensor with cieling landmarks. 86
Figure 5.3: Odometry system design. 86
Figure 5.4: The laser sensor and its incident beam on the end effector. 87
Figure 5.5: Gaussian distribution of X-values with mean = -0.5 mm and standard deviation = 1.2 mm. 87
Figure 5.6: Gaussian distribution of Y-values with mean = 0.7 mm and standard deviation = 2.5 mm. 87
Figure 5.7: Gaussian distribution of Z-values with mean = 0.1 mm and standard deviation = 0.6 mm. 88
Figure 5.8: The servo joints values during the movement. 89
Figure 5.9: Calibration process using tilt meter. 90
Figure 5.10: The grid paper and the marker attached to the end effector. 91
Figure 5.11: Gaussian distribution for end effector positions after calibration, A: for X-values, B: for Y-values, C: for Z-values. 92

Figure 5.12: The values for calibration process to keep the end effector at the height of 180 mm for different distances. 92
Figure 5.13: The labware on the workstation, A: The posture and yaw orientation of labware, B: The design of handle and grippers....94
Figure 5.14: The framework of labware manipulation. 94
Figure 5.15: The flowchart of blind labware manipulation. 95
Figure 5.16: The relation between performance and flexibilty of grippers design. 96
Figure 5.17: The first design of grippers and labware container handle. 96
Figure 5.18: The handle design with flat panel for fixing a graphical code. 97
Figure 5.19: The new design of grippers and handle with flat panel. 97
Figure 5.20: The design of vertical handle with flat panel. 97
Figure 5.21: The arm configuration for the manipulation of horizontal handle. 98
Figure 5.22: The arm configuration for the manipulation of vertical handle. 98
Figure 5.23: The final design of fingers and labware container. 98
Figure 5.24: The labware manipulation using the final design. 99
Figure 5.25: Holder frame of the labware container that was shown in Fig. 5.17....99
Figure 5.26: Holder frame of the labware container that was shown in Fig. 5.19....100
Figure 5.27: Holder frame of the labware container with vertical handle. 100
Figure 5.28: Holder frame of the labware container that was shown in Fig. 5.23....100
Figure 6.1: The grasping point of a labware handle. 102
Figure 6.2: The flowchart of manipulation algorithm. 102
Figure 6.3: The detection and localization of labware handle using Kinect V1....103
Figure 6.4: The detection of labware and some points on the workbench. 103
Figure 6.5: The detection of marks for placing task. 103
Figure 6.6: The camera extrinsic calibration for arm manipulation. 104
Figure 6.7: Kinect-to-shoulder transformation. 105
Figure 6.8: The final matrix calculation process. 105
Figure 6.9: The kinect holder and how to fix the 3D orientation of Kinect. 106
Figure 6.10: New handle design. 106
Figure 6.11: The recognition of handle using HSV and shape detection. 108
Figure 6.12: The recognition of handle using SURF. 108
Figure 6.13: The recognition of holder using HSV and shape detection. 108
Figure 6.14: The recognition of holder using SURF. 108
Figure 6.15: The grasping point and how the arm reaches it. 109
Figure 6.16: The architecture of client-server model for handle manipulation. 110
Figure 6.17: The flowchart of arm manipulation system. 111
Figure 6.18: The battery and voltage stabilizer for Kinect sensor. 111
Figure 6.19: Different kinds of labwares. 112
Figure 6.20: The 8 positions of labware containers on the workstation. 112
Figure 6.21: The complete FOV of Kinect sensor. 113
Figure 6.22: The ROI image of the workstation after the cropping step. 113
Figure 6.23: Top view of rack with 2 tubes at different positions on the workstation. 114
Figure 6.24: Top view of rack with 24 tubes at different positions on the workstation. 114
Figure 6.25: The recognition of 2-tubes rack on the workbench. 115

List of Figures

Figure 6.26: The success rate of 2-tubes rack identification. 115
Figure 6.27: The image after grayscale conversion and histogram equalization. 115
Figure 6.28: Fixing a mark on the labware lid. 117
Figure 6.29: Labware recognition according to the related mark. 117
Figure 6.30: The identification success rate of the labware mark (smoothed paper). 118
Figure 6.31: The identification success rate of the labware mark (coarse paper). 118
Figure 6.32: The Kinect V2 with polarization filter and intensity filter. 120
Figure 6.33: The impact of using the pol. filter and intensity filter with Kinect V2. 120
Figure 6.34: The success rate of 2-tubes rack identification (with intensity filter). 120
Figure 6.35: Labware recognition according to the related mark (with filters). 121
Figure 6.36: The identification success rate of the labware mark (with filters). 121
Figure 6.37: Labware mark with red features. 122
Figure 6.38: Lid mark identification using HSV filtering and SURF. 122
Figure 6.39: The identification success rate of the labware mark (HSV+SURF). 123
Figure 6.40: The architecture of client-server model for multipla labware manipulation. 124
Figure 6.41: The grasping point of labware container. 125
Figure 6.42: The H20 holder of labware container. 125
Figure 6.43: Pulling springs for the weak wrist joint. 126
Figure 6.44: The workbench holders for placing tasks. 127
Figure 6.45: Placing marks for holders identification. 127
Figure 6.46: Checking the availibility of the required holder. 127
Figure 6.47: The micro switch and servo motor for placing procedure. 128
Figure 6.48: Blind manipulation for multiple labware. 129
Figure 6.49: Red squre detection and pose estimation using POSIT 129
Figure 6.50: The corners of the lid mark. 130
Figure 6.51: Pushing the elevator entry button using F200 camera and old gripper. 131
Figure 6.52: New finger design for pushing the elevator buttons. 131
Figure 6.53: Sequence of pushing the elevator button. 131
Figure 6.54: The joints structure of H20 head. 132
Figure 6.55: AXIS network camera. 132
Figure 6.56: Color tracking with H20 head. 133
Figure 7.1: Feedback tracking using the mark on the end effector. 137
Figure 7.2: Fixing a mark on the labware side for tracking. 138
Figure 7.3: The architecture of feedback control using hand camera. 138

List of Tables

Table 2.1: The summary of transformation matrix. 29
Table 3.1: The difference between Kinect V1 and V2. 66
Table 4.1: The D-H parameters for the H20 arms. 72
Table 5.1: The expected and registered positions. 91
Table 6.1: The handle manipulation tests. 110
Table 6.2: The results summary of 2-tubes rack identification. 116
Table 6.3: The results summary of mark identification (smoothed paper). 119
Table 6.4: The results summary of mark identification (coarse paper). 119
Table 6.5: The results summary of 2-tubes rack identification (with filters). 121
Table 6.6: The results summary of mark identification (with filters). 122
Table 6.7: The results summary of mark identification (HSV+SURF). 123
Table 6.8: The overall summary of the best results. 124
Table 6.9: Manipulation tests of multiple labware. 128

List of Abbreviations

2D	2 dimensional
3D	3 dimensional
ADC	Analogue to Digital Converter
ANN	Artificial Neural Network
Celisca	Center for Life Science Automation
API	Application Programming Interface
PMS	Process Management System
C#	C-sharp programming language
IP	Internet Protocol
TCP	Transmission Control Protocol
PC	Personal Computer
DOF	Degree of Freedom
RRR	Revolute - Revolute - Revolute
RRP	Revolute - Revolute - Prismatic
FOV	Field of View
FPS	Frame per Second
GUI	Graphical User Interface
H20	Hawk 20
RGB	Red, Green, and Blue color space
RIA	Robotic Industry Association
ROI	Region of Interest
FK	Forward Kinematic
IK	Inverse Kinematic
DH	Denavit-Hartenberg
ROS	Robot Operating System
IR	Infrared
HSV	Hue, Saturation, and Value color space
OpenCV	Open Computer Vision
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
IBVS	Image-Based Visual Servoing
PBVS	Position-Based Visual Servoing
FP	False Positive
FN	False Negative
PCL	Point Cloud Library
WARTG	Wrist Angle Relative to Ground
SDK	Software Development Kit
RRC	Robot Remote Center
RBC	Robot on-Board Center
RAC	Robot Arm Component
ROM	Range of Motion

1 Introduction

Automation is the use of control systems in correspondence with the information technology applications to control some processes for reducing the need of human sensory and mental requirements. Currently, automation plays an important role in daily life and in the global economy, where a lot of applications in industrial processes, which were previously carried out by human, depend on automation fields. The main advantage of automation is to replace humans who perform some tedious tasks or the tasks which are performed in dangerous environment such as fire, space, under the water, volcanoes, etc. Also, the automation is necessary for the tasks where human doesn't have the ability to perform like the manipulation of very heavy loads, very large objects, too hot or cold substances, chemical materials and so on. The automation process can be realized using robots which are electromechanical machines designed to move, explore, or transport different objects.

There are two types of robots, stationary and mobile. In general, the robots can be classified according to their applications which are used for such as services robots, industrial robots, medical robots, etc. According to the application, the robots perform the required tasks by its own (autonomous) or with a guidance. In comparison with human, the robot is stronger, tireless, repeatable (preferable in routine work), well-immune, labor saver, and better producer.

Mobile robotics cover the robots that roll, walk, fly or swim. There are four fundamental issues that the mobile robot has to consider: its current location, the target location, how to reach there, and what to do there. To achieve these issues, the mobile robot has to make some measurements and modelling for the environment, localize its position, and plan a path to the required location.

1.1 Mobile Robot Requirements

A mobile robot is a combination of various physical (hardware) and Computational (software) components. Software component is a set of subsystems which are responsible for processing and planning its various aspects. Related to hardware components, it can be considered as a collection of subsystems for:

- Locomotion: how the robot moves and maneuvers in the environment.
- Sensing: how the robot measures the properties for itself and its working environment.
- Reasoning: how the robot maps the measurements into actions.
- Communication: how the robot communicates with the other components in the environment.

1.2 Autonomous Mobile Robots

Autonomous mobile robots are the robots which have a big importance not just for reasons of mobility but because of their ability of autonomy. Their perception ability to maintain a sense of position and to navigate in an unstructured environment, for performing different tasks without a continuous intervention from a human operator, is paramount. The interaction with the environment and taking the decisions without exterior commands can be considered the common characteristic of autonomous robots. This can be performed using different types of sensors or by using artificial intelligence to teach the robots how to take a decision in different situations. The navigation and the arm manipulating for the desired objects are the most important conditions to realize an autonomous performance for the mobile robots. Navigation is the process of creating a path for moving from one place to another, using wheels, legs, or other means, safely and without collision. The arm manipulation is the procedure of controlling the robotic arm to perform some tasks related to the desired objects in the environment.

1.3 Application Areas of Mobile Robots

The range of applications for mobile robots is enormous where these applications can be performed in different indoor and outdoor working environments. The applications area can be classified according to the mobile robots' tasks which they perform. Some examples related to mobile robots' tasks are as follows:

- Services support such as objects transportation (food, medication, etc.).
- Automatic cleaning for large areas such as supermarkets, airports, industrial sites.
- Client guiding in museums and exhibitions.
- Fruit and vegetable picking, fertilization, and planting in agricultural fields.
- Tree cutting, fire preventing, and cleaning in forests.
- Inspecting in hazard environments (catastrophic areas, volcanos, nuclear power plants, oil tanks).
- Constructing and demolishing.
- Space exploration like remote inspection of space stations.
- Military surveillance and monitoring for vehicles and material.
- Surveillance of large areas like big buildings or airports for safety and security purposes.
- Assistance for handicapped and elderly people.
- Entertainment.

1.4 The Required Sensing for Arm Manipulation

One of the essential tasks of mobile robots is their ability to transport different objects within the working environment. This activity requires sensing and understanding the surroundings which is complicated especially when the mobile robots work in unstructured environment. The realization of an effective mobile manipulation requires a high degree of flexibility in perception, motion, and control. Large variations in objects, lighting, and clutter make the manipulation a very difficult task. Mobile manipulation in such environments requires a tightly integrated effort, combining techniques in sensing, motion planning, grasping, and control.

The stationary robot has a fixed position base which makes the activity of object transportation from one place to another easier in comparison with the mobile robot. But the stationary robotic arm has a limited range of motions related to the installation place. In contrast, the mobile robot would be able to travel throughout the working plant, flexibly performing its skills wherever it is most effective [1]. The inaccuracy of reaching the required position with the required orientation, which is related to navigation problems, makes the tasks of object manipulation with the mobile robots even more complicated. The fusion of different sensing sources enables a mobile robotic system to better perceive the environment and increase the system robustness for object manipulation [2].

The term of robot manipulation includes all of those techniques which allow the robotic manipulators to interact with the objects in the working environment for the purpose of transporting them to a desired place. In other words, robot manipulation aims to grasp objects by exerting the force at the physical contacts using the grippers. It is the ability to secure an object inside the robot's grippers while resisting external disturbances. Grippers are the connecting part between the robotic arm and the object to be manipulated. Generally, a gripper in industry is a specific device that is used only to manipulate one or a few objects of similar shape [3]. The robotic grasping strategy has to meet specific and desirable contact properties of the target object like its robustness, reliability, and stability. It is important to maintain the object attached to the gripper in such a way that the set (gripper–object) does not have relative movement in all its trajectory. Thus, the object is not damaged due to the fact that the actuating forces on it do not overpass its mechanic strength [3].

For improving the autonomous manipulation condition, different sensors have to be used. The aim of sensing capabilities is to provide feedback in order to adapt the robot hand with the objects and environment conditions [3]. There are three main sensors can be used: vision, touch, and hearing sensors. Hearing is a complementary sensor that can ease the detection of contacts by rising the contact likelihood after the hearing of a contact. But the most common

sensor for the manipulation work is visual based. The ability to recognize an object, accurately locate and track its pose (position and orientation) is of fundamental importance in robotic grasping [4]. Visual sensing is the most suitable way to get 3D view information to find the pose of the objects in the workspace and to detect contacts too. It can also be used to track the relation between the hand and the manipulated object especially with the unstable robotic arms. But on the other hand, visual sensing is not 100% reliable because of environmental uncertainties, sensing errors, calibration errors, and so on [3]. The classical approach to obtain 3D information from the view is the stereo vision system. This method is based in the disparity between two camera images to extract depth information. Most of the humanoid robots use a stereo vision system on their head for object manipulation. Another approach is to actively project a pattern to the scene and observe its deformation to obtain the 3D structure. This method is used by the Kinect sensor. The impact of Kinect sensor in the robotics community is quite high due to its low cost, high reliability, precision and speed [3]. Fig. 1.1 shows an example of a mobile robot with Kinect sensor which is working in hospitals for objects transportation.

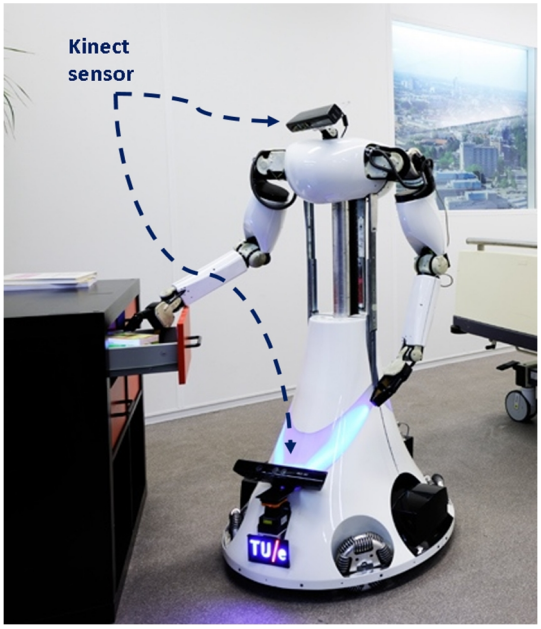


Figure 1.1: Mobile robot in grasping task in the hospital [5].

Another sensor, that can be used to deal with that lack of accuracy, is the touch sensor. In robotic systems, sense of touch is implemented using three main types of sensors: tactile, force torque, and kinesthetic sensors. The tactile and force sensing is very crucial in case of manipulating different unknown objects which have different shapes, sizes, and different surface material properties. Touch information is devoted to local features while vision information is used for global scene understanding [3][6].

1.5 The Control of the Mobile Robotic Arm

Robotic grasping and placing are common in industrial environments and have to be executed with a high degree of reliability. It is a key prerequisite for enabling a large number of mobile manipulation tasks, such as objects transport and retrieval [7]. For the activity of objects transportation with mobile robots, the most important issue is how to control the robotic arm autonomously to achieve the grasping and placing tasks in a safe way with the required accuracy. A robotic manipulator is an electromechanical system with multiple degree of freedom (DOF) that is programmable to accomplish a variety of tasks. Degrees of freedom (DOF) are the number of independent motions which an arm can make to completely specify its configuration [8]. The robotic arm consists of rigid bodies called links that are connected by joints. Fig. 1.2 illustrates an example of general structure for 6-joints robotic arm where arm-type manipulators are considered important elements in robot systems because of their wider movement range and higher general versatility [9].

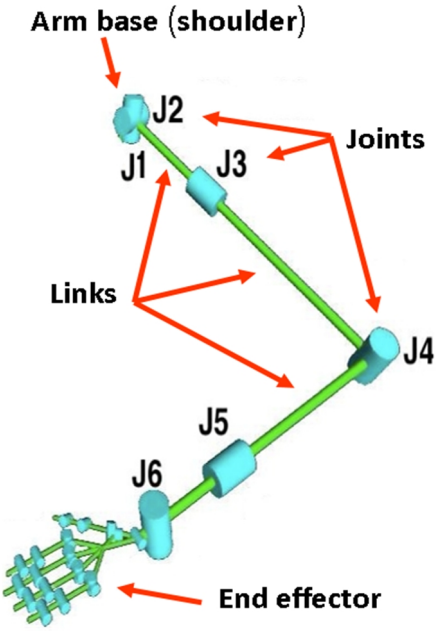


Figure 1.2: 6-joints robotic arm [10].

In order to have a robust service robotic arm, it must have the ability to grasp and manipulate different objects and this requires an accurate kinematic model. Studying the theory of kinematics is important in order to analyze the robot arms. Kinematics is the most basic study of how mechanical systems behave. The mechanical behavior of the mobile robotic arm has to be studied to create a control software for an instance of robotic arm hardware. The kinematic analysis is the relationships between the positions, velocities, and accelerations of the arm

links, which are calculated without considering the forces that cause arm motion. The kinematic is separated in two types, forward kinematics (FK) and inverse kinematics (IK). In forward kinematics, the length of each link and the angle of each joint are given but the pose of arm end effector has to be found. In inverse kinematics, the length of each link and the pose of arm end effector are given but the angle of each joint has to be calculated. In other words, the inverse kinematics is required to convert the 3D Cartesian coordinates of the target to joint coordinates of the robotic arm. The solution of IK problem is more complicated to be found compared to the solution of FK problem especially with the serial arms. With the kinematic model and a motion planning module, it is possible to move the manipulator from one pose to another in a free-collision path. In order to have a robust module for manipulation tasks, it is necessary to compute the pose of the desired object where many of researches use a visual tracking system which is based on image segmented features for this purpose [11]. This task requires certain visual sensors and techniques for locating the objects in 3D space. Then the robot arm runs the pick and place behavior for the target by controlling the arm joints using the kinematics analysis [12][13]. Another issue, which has to be taken in to the consideration for robust arm manipulation, is the design of gripper. The gripper has to be designed very carefully to guarantee a secure arm manipulation.

1.6 Motivation and Goal

The outlook of life sciences laboratories is based on the development of innovative automation solutions for the entire scope. The laboratories are equipped with automation equipment in order to perform practical scientific work like biological testing, sample preparation and analysis. The innovative automation in the life science laboratories ensures high throughput, workflow optimization and reliable measurement results. The robots and especially the mobile robots are very important and necessary in the field of life science automation. All the automation islands and the workstations in multiple life science labs can be connected with each other using the mobile robots. This connection leads to increase the productivity by ensuring 24/7 operation and to save the human resources by reducing the tedious and routine work for the employees. In the center for life science automation, different automation islands are connected with each other by the cooperation of stationary and mobile robots. This requires an appropriate hierarchical management systems to perform the automation of individual areas in such a way that a comprehensive life science processes are realized. Fig. 1.3 shows the overall structure of the hierarchical workflow management system and how the stationary and mobile robots are connected with each other in the life science laboratories.

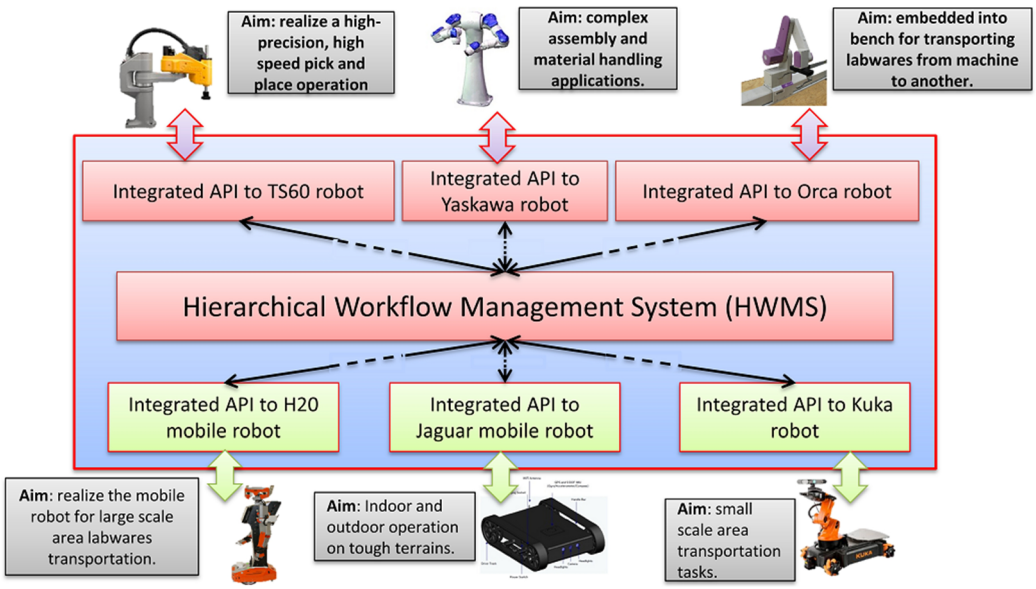


Figure 1.3: Overall structure of management seystem in life science labs.

Five H20 wheeled mobile robots are used for maneuvering between the adjacent labs and automation islands for transporting multiple labware and tube racks which contain chemical and biological components. The H20 robot is a wireless networked autonomous humanoid mobile robot. It has a PC tablet, dual arms, and an indoor GPS navigation system. Different aspects are developed in the H20 system as shown in Fig. 1.4. The arms manipulation, multi-floor navigation with elevator handling, and the collision avoidance with human robot interaction are the main aspects that have been developed. This work focuses on the labware identification and arm manipulation to investigate a high precision grasping and placing during the labware transportation tasks with H20 mobile robots.

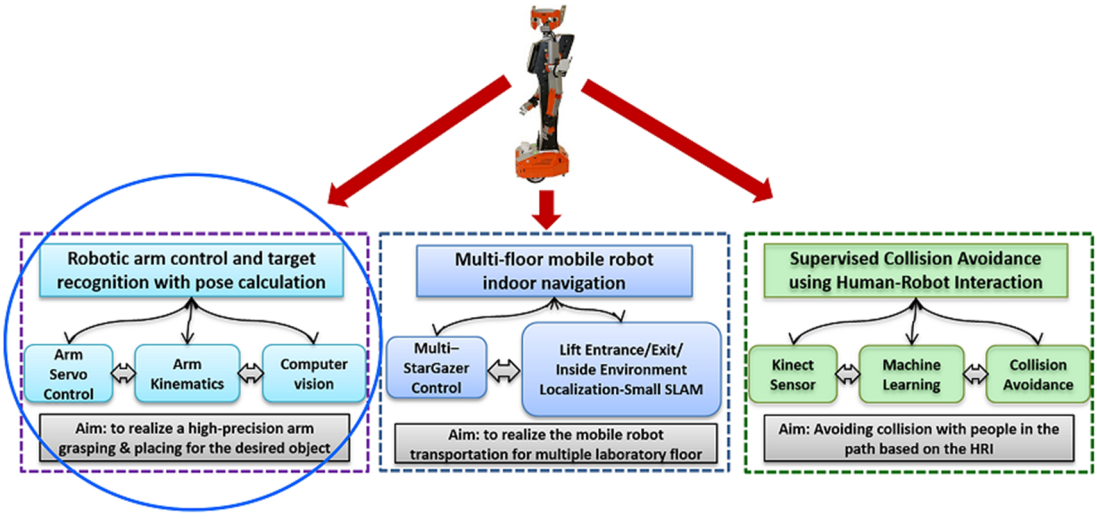


Figure 1.4: The developed aspects in the H20 robot system.

1: Introduction

The H20 mobile robot has a navigation plan to follow a predefined path to a specified station using a guidance control system which depends on stargazer sensor with ceiling landmarks (Hagisonic Company, Korea). The use of landmarks can be considered as a classical navigation principle on which modern robotic navigation is founded. A landmark is a visible feature in the environment whose location is known with respect to some coordinate frame [14]. This navigation system inevitably causes orientation and position error for the robot in front of the workstation. The error is related to two reasons. The first is the strong lighting and sunlight, which makes the stargazer unable to recognize the ceiling landmarks. The second reason is related to the odometry system, which includes encoders that are mounted on the robot wheels to provide the motion information that updates the robot pose. The odometry system accumulates errors for different reasons such as different wheels' diameter, wheel-slippage, wheels misalignment and finite encoder resolution. According to the experiments results and previous studies, the rotation of the robot is the greatest factor in odometry errors [15][16]. Uncertainties in the robot pose compel to search and find the desired labware whenever the robot arrives the workbench. Since these labwares contain chemical and biological components, a high precision grasping and placing performance is required during the transportation between the automation islands. The required accuracy for labware manipulation has to be less than 1 cm to guarantee safe tasks. Therefore, the more direct way of dealing with this problem is to use sensors to provide position and orientation feedback of the target [17]. Then, the joints of the arm have to be configured using the kinematic model to place the arm end effector precisely and safely in the desired pose if it is inside the reachable space. The solution of the kinematic problems of the H20 robot arms has to be found and validated to be integrated to the arm control system. The accuracy and repeatability of the arms have to be checked and calculated to know the related requirement for system development. Multiple tube racks and labware have to be identified and localized visually to be manipulated wherever they are located on the workbench. This requires a suitable visual sensor like a Kinect sensor which can provide the view image with the related depth information to compute the 3D pose of the labware. Visual feedback is a major issue in robotics control where the images data provided by visual sensors are processed to extract valuable information. The quality of the visual feedback control depends on the computation power, sensor resolution, and the algorithm quality which is used. Object manipulation using visual data involves determining the pose of the object with respect to the manipulator base. Since there are different kinds of labware and tube racks which have different shape and design, it is not possible to manipulate them directly using the H20 grippers. Also, the structure of H20 arms and gripper restricts such tasks. Therefore, labware containers have to be designed to be manipulated directly by the H20 robot to realize the transportation task. According to the design of labware container, the arm gripper can be resolved.

1.7 Dissertation Outlines

This dissertation is organized as follows. In chapter 2, literature overview and state of art regarding arm manipulation with grasping and placing technologies are included. The definition of kinematic analysis with its basic terms is illustrated also in chapter 2. The strategies of computer vision to perform the detection and pose estimation for the desired objects are also shown in chapter 2 which will end with a general overview about the H20 mobile robot and its system operation. Chapter 3 includes the general concept about the requirements that are necessary to perform the manipulation strategy and the suitable techniques that can be used. The solving methods of the kinematic problems and how to choose the suitable one for the H20 arms are clarified in the same chapter. Chapter 3 shows also the techniques of object detection depending on color, shape and local features and the specification of Kinect sensor with its technology to obtain the depth information. The chapter ends with an explanation about the client server communication model which is important for system integration by connecting multiple platforms with each other. Chapter 4 shows the structure and the specifications of H20 arms which will be followed with the analytic solution of their kinematic problems. Chapter 4 includes also the validation and simulation process for the kinematic solution using MATLAB and how to select the desired solutions within multiple solutions. In chapter 5, the integration of kinematic solution to the H20 arms controller is shown. It includes also the tests of arm repeatability and accuracy and how to perform a blind manipulation strategy for the labware based on sonar sensor. The second part of chapter 5 illustrates several designs of gripper and labware containers which are crucial to guarantee a secure transportation. Also, several designs of labware holder, which are installed on the body of H20 robot, are shown in this chapter. Chapter 6 shows the visual manipulation of labware using Kinect sensor. Two versions of Kinect sensor V1 and V2 are used to detect and localize the required target. The handle, which is attached to the labware container, is detected using color with shape detection and feature matching algorithm. The chapter shows also the test results of multiple labware identification and manipulation and the integration and connection of the manipulation system with the navigation system through asynchronous TCP/IP socket (client-server model). The chapter ends with the related work concerning the elevator handling by pushing the elevator buttons and the head kinematic with stereovision control. The summary and outlook are included in the final chapter 7.

2 Literature Overview

Grasping technology has been developed to help humans for loading and manipulating objects of different sizes, materials, and conditions. This has been done with two categories of arms, namely tools and prostheses. Tools are those arm systems or devices that help humans with grasping capability to achieve routine, difficult or dangerous grasps. Prostheses are systems that are developed to restore grasping ability for the human arms when hands are lost [3]. This work deals with the arm grasping and placing technologies which are performed using autonomous robots to achieve objects transportation tasks within indoor environment. Fig. 2.1 shows some different designs of arms related to stationary and mobile robots.

Figure 2.1: Different design of robotic arms [18-22].

2.1 State of The Art in Object Manipulation

It is essential to give an overview about some projects which were developed to perform the robotic manipulation tasks in different fields, applications, and environments. The object grasping and placing have been performed with stationary and mobile robots. The stationary robot has a fixed position in the working environment, therefore this kind of robot is used in a specific application and in structured environment. This also means that the stationary robot normally manipulates specific objects which are previously known. Most of stationary robots have a stable manipulator because of the robust structure and accurate joints provided with feedback system. On the other hand, the mobile robot normally moves from one place to another in unstructured environment and has to deal with different kinds of objects. This difference makes the application of object manipulation with mobile robots is more complicated in comparison with the stationary robot. In typical manipulation scenarios, the

workspace of a stationary mounted robotic arm is too small. One possible solution to achieve a larger workspace is to construct manipulator with a restricted workspace but with a mobile base. However, the object manipulation technologies, which have been performed with stationary robots, can be considered as the basis for the grasping and placing development with mobile robotics. Most of the studies state that in order to provide a stable grasp, the grasping system requires complete information regarding the robot kinematics with its stability, gripper capabilities, sensor capabilities and also about the workspace where the objects are placed [23]. In this section, a general overview will be given related to some object manipulation applications with stationary and mobile robots. More details about the technologies, which were used in these applications, will be explained in the next sections of this chapter.

Wu et al. described the theory and implementation of neural networks for hand-eye calibration and inverse kinematics of a 6-DOF arm of UR5 universal robot equipped with a stereo vision system as shown in Fig. 2.2 [24]. The feedforward neural network and the network training with error propagation algorithm were applied. In this work, the aim of hand-eye calibration is to determine the 3D location of an object in the robot base frame, based on the 2D coordinates of two cameras. The hand-eye calibration problem of stereo vision system can be dealt with through triangulation methods. Regarding the inverse kinematics problem, there are several techniques to find the solution. In general, they can be classified into analytical solution and numerical solution. For robotic arms with higher DOF or with non-spherical wrist, the analytical solution of inverse kinematics problem becomes complex due to the high nonlinearities in the kinematics model. In this work, table-tennis ball is manipulated using the UR5 universal robot. Object detection algorithm is applied to extract the ball depending on the ball color. The frames from each camera of the stereo vision system are converted from RGB (red, green, blue) to HSV (hue, saturation, value) color space. Regarding the training of the neural network for hand-eye calibration, first, the ball was attached to the end effector of the robot arm. Then, the ball was randomly moved to 2,000 different positions by the robot arm within its workspace. After that, the coordinates of the ball in image space were detected and both, the 3D and 2D information were saved in a table. The mean absolute error for this step was x =1.81mm, y=1.10mm, z=1.07mm. Regarding the training of neural network for inverse kinematics, the end effector of the robot arm was moved to random positions and orientations within its workspace. Then, the joint angles, end effector positions and end effector orientations were saved at each time. These two steps were repeated several times in order to better approximate the uniform distribution of the sample data points. The maximum mean error of the joints was $8.57X10^{-3}$ rad with a standard deviation of $10.95X10^{-3}$ rad.

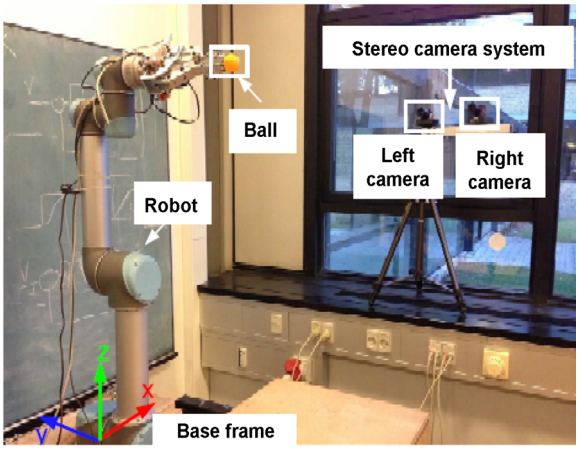


Figure 2.2: UR5 universal robot arm [24].

Although the using of a neural network approach for solving calibration and inverse kinematics problem is easier and time saving, the results values are not exact especially for the solution of inverse kinematic problem. This strategy is not accurate enough to be used for performing a high precision manipulation. The analytical solution enables the arm end effector to reach the target very accurately in case that the robotic arm is stable enough.

Chung et al. presented an intelligent service robot to help people in transporting different objects which are specified by a user within indoor environment [25]. As shown in Fig. 2.3, the robot consists of a Kinect camera, notebook, DSP chip, multi-joint (4-axis) arm, infrared (IR) sensors and iNEMO sensor module[1]. The iNEMO sensors include accelerometers to measure the linear acceleration and earth gravity, gyroscopes to measure angular velocity, and magnetic sensors that output reliable heading information. In order to achieve the task, the Kinect has to capture the target object where the Kinect sensor can be considered as one of the best visual sensors which is used for object detection and localization. Then, sequences of image processing algorithms depending on color information are applied to recognize and track the target. YCbCr color space was used where Y represents the luminance, Cb represents the chrominance of blueness, and Cr represents the chrominance of redness. Whenever the target is detected and localized by the Kinect, its center point has to be found. This information is used to control the robot moving to the object. The robot turns right or left till the target center point is aligned with the center point of the Kinect view. Then the robot moves forward until that the front IR sensor detects the object. When a sufficient distance is sensed by the IR sensor, the gripper on the robot arm will grasp the object and place it at the desired location. This work shows an example for object manipulation using a very simple arm that has a very limited workspace. Regarding to arm structure, it can be used to manipulate a specific design of objects.

[1] The iNEMO sensor module is made by STMicroelectronics, which has several highly accurate MEMS sensors.

The strategy of object detection using color information can be easily affected by the indoor light or sunlight. This leads to inaccurate performance for object manipulation. Also, multiple sources of information have to be used at the same time to recognize the desired object like color, size, shape, etc. This improves the quality of detection algorithm where the using of one information source complicates the issue of finding the required target.

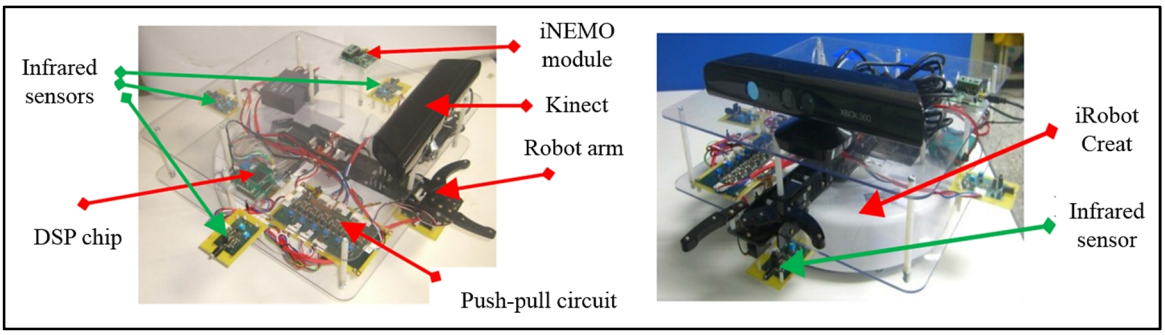


Figure 2.3: An intelligent service robot with Kinect sensor [25].

Bimbo et al. presented a method to detect and estimate a 6D pose of an object by fusing the data from Kinect sensor, tactile sensors and joint encoders for grasping tasks [4]. This fusion of sensors information increases the system robustness. A Microsoft Kinect camera is mounted on the 4-DOF robotic arm and 6-axis force and torque sensors are fixed on the fingertips of a multi-fingered robotic hand as shown in Fig. 2.4. The 3D model of the required objects has been previously acquired and stored in a database. Then, the process to find the required object and its initial pose was performed using SIFT (Scale Invariant Feature Transform) matching algorithm [26]. The second source of information came from the touch sensing to precisely estimate the location of the contact.

This method aims to correct the errors arising from miscalibration, inaccuracy of the vision system or limitations such as occlusions. Given an initial pose acquired by the vision system and the contact locations on the fingertips, an iterative process optimizes the estimation of the object pose by finding a transformation that fits the required object to the finger tips. Regarding the performance of a real system, the mean error in the final estimation was about 5 mm. The dealing with symmetries on an object considers as one of the limitations of this algorithm. The function may have a large number of solutions in case that the object model matches with the contact locations in several positions. This limitation is closely related with the number of fingers contacting the object.

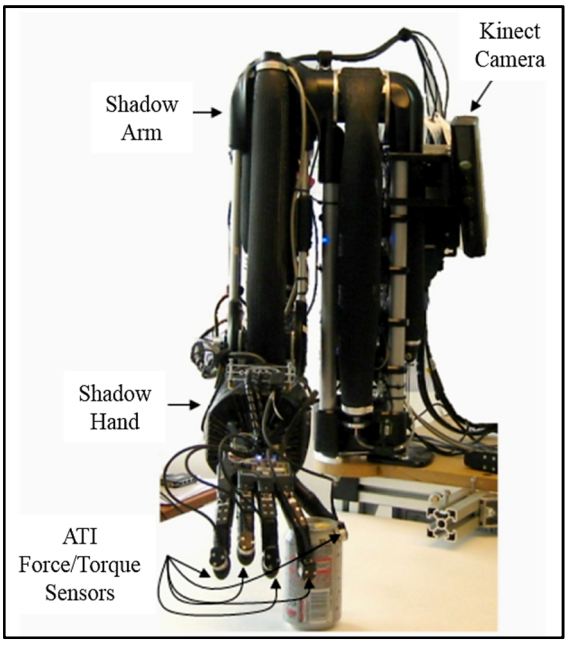


Figure 2.4: Robotic arm with fingertip force sensor and Kinect [4].

Dhawan et al. used a visual feedback system for the robot arm to reach the required destination [27]. The robotic arm consists of three rotational joints and there is an electromagnet part attached to the end effector to pick up the objects. The kinematic analysis is used to control the joints movement of the arm to reach the destination point. The system recognizes the targets using color, shape, and features information. The features recognition depends on SIFT (Scale Invariant Feature Transform) algorithm to obtain a set of features correspondences between the object model and the captured image. The object recognition process was controlled using hand gestures given by the user. There is an internal mapping between the hand gesture and the required objects to be picked. For target localization process, the distance between the arm end effector and the target position is calculated by a number of pixels in X and Y directions. This is done by using close loop feedback control system. Feedback signals were sending continuously from the visual system to the arm till the end effector reaches the target area as shown in Fig. 2.5. Whenever the arm end effector reaches the required area, the electromagnet part gets switched on to pick up the desired object. The arm then moves to the placing spot and the electromagnet is switched off to detach the object. This application can be used in different fields such as manufacturing industry, nuclear industry, and packaging industry. This robotic arm has a very limited workspace because it consists of 3 joints. Also, the target localization process is not accurate enough because it just moves the arm to be close to the target area. Then, the magnetic end effector will pick up the object. Therefore, very limited kinds of objects, which have metal surface, can be handled with this robotic arm.

Figure 2.5: Robotic arm with visual feedback system [27].

Tsay et al. adopted an eye-in-hand vision system to provide visual information for the robotic manipulator to compensate the uncertainties in the pose of the manipulator with the object [28]. The robot manipulator used here is the Mitsubishi RV-3AL industrial robot which has 6 revolute joints with 6-DOF as shown in Fig. 2.6. RV-3AL has been fixed on a mobile base with four passive caster wheels at its four corners. The goal of using this mobile robot is to replace humans in handling and transporting wafer carriers in semiconductor production lines. It has the ability to manipulate different industrial object with maximum weight of 3 kg. This study focuses mainly on developing a novel position-based look-and-move control strategy to guide the manipulator to approach the object and precisely position its end effector in the desired pose. Several schemes are required to implement such a control strategy, including image enhancement, edge detection, corners detection, and camera calibrations. A vision system has been installed on the robot to guide the robot manipulator to perform pick-and-place operations. The vision system includes one CCD (Charge Coupled Device) camera. A target point-feature set is extracted from the image by the proposed image processing algorithms, and used to determine the relative pose. Some tasks must be executed in advance. These are camera calibration, hand/eye calibration, and determining the desired relative pose between the camera and the object to grasp the object. A positioned-based approach is presented to keep all of the target point-feature set in the camera's field of view. The idea is to move the end effector according to the position of the centroid in the image plane. If the centroid is on the right, left, top, or bottom of the image plane, then the end effector keeps moving, following a relative motion command that it moves slightly in the positive or negative XY direction, until all of the target point-feature set is on the image plane. Executing all the steps of the control strategy approximately takes between 6 seconds and 13 seconds. The manipulator is very stable; therefore, the position error range is 0.3mm~1.2mm, and the orientation error range is 0.07° ~ 0.96°. It is very difficult to reach this accuracy with unstable robotic arms. The look-and-move control strategy here doesn't depend on kinematic model to reach the target. This leads to failure in grasping task in case that the target is not within field of camera view during the arm movement. However, the look-and-move control strategy with kinematic model can be a good solution to improve the accuracy of object manipulating in case of dealing with unstable arms.

Figure 2.6: Mitsubishi RV-3AL industrial robot [29].

Chitta et al. described an approach for mobile manipulation for household objects using PR2 (Personal Robot 2) [7][30]. It has an omnidirectional base and two 7-DOF arms with encoders attached on each joint to provide joint angle information as feedback control. It is also equipped with a tilting laser scanner mounted on the body and two stereo cameras. In addition to the stereo cameras, a Microsoft Kinect sensor is mounted on the head of the robot as shown in Fig. 2.7. The end effector is a parallel jaw gripper whose fingertips are equipped with capacitive sensor arrays, each consisting of 22 individual cells. For a large number of different household objects in unstructured environment, visual information is often not sufficient for robust operation. Robustness can be increased with the addition of tactile and proprioceptive feedback. In the unstructured environments, robust grasping requires to solve the problem of uncertainty to cope with errors ranging from sensing and scene interpretation to mechanism calibration.

An analytical solution has been used to solve the problem of IK for the 7-DOF arms of PR2 robot. For visual processing, a combination of two-dimensional (2D) and three-dimensional (3D) has been used. 2D visual information has been used for object recognition and tracking and as feedback for visual-serving controllers. The implementation of 3D visual information depends on the filtered point clouds to segment the surfaces such as tables or shelves and objects on these surfaces. A database of prebuilt 3 D models for common household objects is then used to recognize and register some of the objects in the scene. Mesh models of these objects are used to represent them in the environment model. The grasp planning algorithm requires that a point cloud from a depth camera, such as the Microsoft Kinect or the PR2 stereo pairs, be separated in individual clusters, each belonging to one target object. Once an object's point cloud has been segmented, grasps are computed using heuristics based on both the overall shape of the object and its local features. A software framework has been developed using ROS (Robot Operating System) [31], PCL (Point Cloud Library), and OpenCV (Open Computer Vision) with other sets of software tools.

The success rate for grasping process in this application was 88%. This grasping rate is not sufficient to guarantee a safe manipulation tasks. This problem is related to the large number of household objects which the robot has to grasp. By decreasing the number of the required targets, the grasping success rate will be increased.

Figure 2.7: PR2 robot with Kinect sensor mounted on the head [30].

Dragusu et al. reported a mobile robotic system which consist of video camera with processing tool, robotic arm and mobile platform as shown in Fig. 2.8 [32]. The robotic arm is Cyton V2 which is used for practical laboratory applications. The Cyton V2 manipulator has 7-DOF plus gripper and can perform advanced control by exploiting the kinematic analysis. Also, it has intelligent actuators that give feedback on position, speed, load, voltage, and temperature. The system is designed for processing images captured by the webcam mounted on the arm to extract the required information and send it to the robotic arm. The webcam captures the images to be processed for object detection or for contour coordinates extraction. These methods are implemented using a series of image processing techniques like edge detection, border extraction, contour detection, contour extraction, along with match template and with the aid of OpenCV library. Based on these captures, there were developed algorithms that enable the arm to either sort the recognized objects to be having different shapes, or to draw shapes based on the coordinates received.

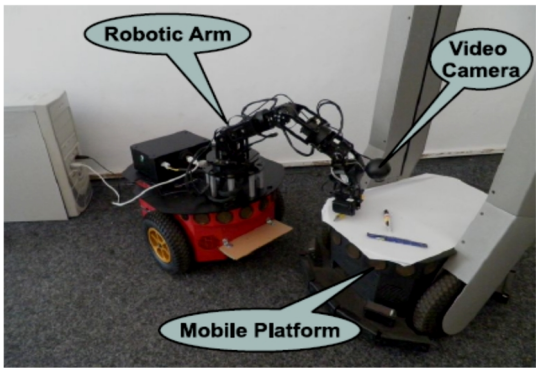


Figure 2.8: Robotic arm with webcam and mobile platform [32].

Sanchez-Lopez et al. presented an application for visual servoing and object grasping with a 7-DOF static robotic arm under a binocular stand-alone configuration [11]. The system consists of stereo cameras, objects with different colors, and a robotic arm with colored gripper which was controlled using the kinematic analysis as shown in Fig. 2.9. The proposed method depends on tracking and pose estimation for the end effector with the required objects. It is composed of sequence of three processes. The first one is a fast segmentation in HSV color space where the end effector and the grasped objects have different colors. This step considers the segmentation using pixels' classification and Euclidean distance. The HSV color segmentation is useful for control the brightness lighting because one of the main problems in dynamic environments is lighting conditions. For this case, the conversion from RGB to HSV was performed. The second step is the selection of a region of interest, feature point extraction and tracking over that region. It is important that the color of the grippers has enough linear reparability from the colors of the targets. Finally, the centroid for every color was found where the feature points can be used to estimate the homography between world reference frame and image frame to compute the error between the desired position and the gripper position. This vision system is susceptible to errors where the percent error of pixels' misclassification is 5.65 %. It means, in case, that the gripper is too far from the visual sensor or there is bright light in the scene, the segmentation will fail. Therefore, it is desirable to perform the detection process using features matching, or object recognition in order to make the system more robust to noise.

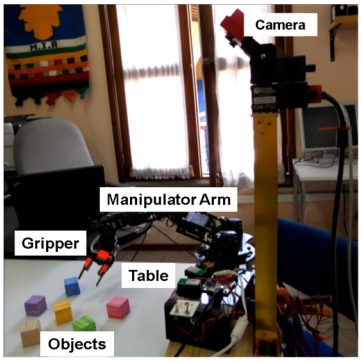


Figure 2.9: Arm manipulation for objects with different colors [11].

Chen et al. described a robotic grasping system integrating 3D object localization with stereo vision in unstructured environments [33]. A typical object manipulation task can be subdivided into several components, such as object detection and localization, arm motion planning, and visual servoing. The system consists of a 6-DOF ADAM dexterous robotic arm and a stereo camera module for eye-to-hand configuration based on position-based visual servoing (PBVS) to locate objects in 3D space as shown in Fig. 2.10. The Kinematic analysis has been used to control the arm motion planning. The object detection and localization strategy has been

performed for two different classes of objects: textured objects and single-colored objects. The methods which depend on SIFT local features and color segmentation in a HSV color space have been used to detect objects and locate their 2D positions in the image. The use of SIFT local features depends on extracting textural information. The SIFT feature is fully invariant to image rotation and scale. It is also robust against a substantial range of affine distortions and change of illumination. Many features can be extracted from typical images, which lead to robustness in extracting small objects among clutter. For object localization and 3D position calculation with stereo vision, 2D detection and segmentation were performed for the left and right images. This approach mainly consists of five steps: camera calibration, image rectification, the difference of color on mapping pixels, aggregation of difference and disparity optimization. In 3D computer vision, the calibration of the camera is the process by which the intrinsic and extrinsic parameters are calculated. The intrinsic calibration describes the transformation relationship between the camera coordinates frame and the image coordinates frame. Common intrinsic parameters of the camera include focal distance, the location of image formation on the objective center, the ratio of length and width for the pixel, and the parameters of twist and shape change of the camera lens [33]. The extrinsic parameters include the rotation matrix to describe the location and orientation of the camera with respect to other coordinate systems.

Figure 2.10: Object grasping using a dexterous robotic arm, ADAM, and a stereo camera [33].

Stueckler et al. described efficient methods for solving everyday mobile manipulation tasks in domestic environment which need a vast set of perception and action capabilities [34]. For this process, the mobile robot not only requires localization, mapping, path planning, and obstacle avoidance abilities to safely navigate through the environment. It also needs to integrate object detection, recognition, and manipulation. A typical requirement for a service robot is not just to achieve the task, but to perform it in reasonable time. This work presented fast methods to flexibly grasp objects from planar surfaces with cosero mobile robot. Cosero has two 7-DOF anthropomorphic arms which resemble average human body proportions and reaching capabilities. It also has an omnidirectional drive to maneuver in the narrow passages. The robot senses the environment in 3D with a Microsoft Kinect sensor mounted on its head with a pan-

tilt unit in the neck as shown in Fig. 2.11. 3D scene perception uses the depth images of Kinect at high frame rates for the segmentation of the required objects on planar surfaces. A differential inverse kinematics with redundancy resolution has been implemented to control the anthropomorphic arms motion. This way of solving the kinematic problem doesn't provide the exact values for the joints' angles which leads to inaccurate reaching for the arm end effector. Also, this kind of robot, which deals with different domestic objects, needs to be equipped with tactile feedback from touch sensors which are very crucial for these tasks.

Figure 2.11: Cosero mobile robot grasps a spoon using Kinect sensor [34].

Grundmann et al. presented a methodology for object recognition and 6d pose calculation for robotic grasping using stereo vision system [35]. This object recognition method is capable for multi-object estimation in complex scenes as shown in Fig. 2.12. It handles partial occlusions and deals with large sets of different and alike objects. Object occlusions and the appearance of similar objects in the same image often constitute problems to robust detection. The object recognition process uses local interest points from the SIFT algorithm as features for object classification. All objects data is extracted in an offline model generation process from large training data sets of a total of 100 different household items. The stereo vision system of the robot consists of two Avt Pike F-145C fire wire cameras with a resolution of 1,388 X 1,038 pixels each. The precise intrinsic and stereo calibration of the cameras are essential for pose calculation and they were performed using MATLAB with camera calibration toolbox. The build process starts with computing the SIFT interest points (IPs) for each image and calculates 3D points by triangulation of IPs in each stereo image. The application of the SIFT algorithm on stereo images enables the calculation of precise 6d object pose. The precise detection of the object pose allows accurate positional representation which is important for grasping tasks. The low precision of the pose of surrounding objects can jeopardize manipulation attempts due to reduced clearance. The experiments demonstrated a true positive detection rate of 72%. The resulting pose errors had a standard deviation of 3.4mm in the Z direction of the camera and 1.4mm in X and Y directions. The resulting detection rate of this method is not sufficient to guarantee high precision manipulation tasks.

Figure 2.12: Service robot in complex environment for manipulation task [35].

Yamazaki et al. described a mobile robot for a cooking assistance in the kitchen environments [36]. Recognition and manipulation functions were developed and integrated for handling kitchen tools and foods. The robot has two 7-DOF arms with 2 high resolution cameras mounted on the head as shown in Fig. 2.13. Each arm has a force sensor embedded on the wrist and its end effector is gripper type hand. Dual arm motion was generated by the approach of inverse kinematics using a single Jacobean matrix. Related to the objects detection, it is known that the kitchen tools and foods have less distinctive texture on its surface and they are susceptible to the effect of illumination. Several simple image processing methods and combinations of them have been used for overcoming these conditions. The used functions for image processing were edge extraction, area segmentation based on color or intensity, background subtraction, 2D geometrical shape detection, and edge segments based matching. For example, to recognize the cutting board, a simple 3D geometrical model was used. This process was achieved by matching the model edges with the captured image edges. Also, a particle filter based pose recognition was applied to improve the correspondence process which can be affected by image noises and light reflection. Related to vegetable detection, the HSV color system was used for achieving a robust object segmentation because it is insusceptible to illumination influence.

Figure 2.13: A mobile robot for assistance in the kitchen environments [36].

Collet et al. presented an approach for object recognition and pose registration for domestic objects manipulation [37]. The method has been performed for grasping different objects with a Barrett WAM robotic arm which has 7-DOF with human-like kinematics as shown in Fig. 2.14. The Inverse-Kinematics BiDirectional Rapidly-exploring Random Tree algorithm (IKBiRRT) has been used for arm motion planning [38]. The arm trajectory plan starts from its current configuration and ends at a configuration of placing the end effector at an acceptable position for grasping. The process of recognition and pose calculation is separated into an offline object modelling stage and an online recognition and registration stage. For 3D modelling of objects, SIFT algorithm has been used to extract local descriptors from several images. Each model is optimized to fit a set of calibrated training images, thus obtaining the best possible alignment between the 3D model and the real object. Given a new test image, the online local descriptors are matched with the stored models using a combination of the RANSAC (Random sample consensus) [39] and Mean Shift algorithms [40] which allows for efficient search of multiple instances of the same object. For pose estimation accuracy, the average translation error was 0.67 cm and the average rotation error was 3.81°. The results of this method show that the overall success rate for grasping was 91%.

The author also presented a framework for Multiple Object Pose Estimation and Detection (MOPED) that seamlessly integrates single-image and multi-image object recognition and pose estimation in one scalable framework [41]. A robust performance has been achieved with iterative clustering estimation (ICE) algorithm that iteratively combines feature clustering with robust pose estimation. The task of recognizing a single object and determining its pose from a single image requires solving two sub-problems: finding enough correct correspondences between image features and model features, and estimating the model pose that best agrees with that set of correspondences.

Figure 2.14: Barrett WAM robotic arm in the process of grasping an object [37].

Anh et al. proposed a high-speed object tracking and visual servoing method based on a window approach and a local feature descriptor [42]. A LWR robot with an eye-in-hand

"Firefly" camera was used to perform the tasks of grasping and putting the desired object in the desired place as shown in Fig. 2.15. The window approach for one of the local feature descriptors, SURF algorithm (Speeded-Up Robust Features) has been used to select robust and intuitive features of the object for visual servoing [43]. The task is to position the robot's end effector in the desired pose relative to the object accurately. These features decouple the translations and rotations from the image Jacobian and keep the object inside the field of view of the camera. The visual servo controller uses geometrical features that are computed directly from the set of SURF interest points, which makes a method robust to the loss of features caused by occlusion or changes in the view point. This method depends on the approach of image-based visual servoing (IBVS) which is used to control the robot to a specific pose in the environment. This can be done by matching the image features between the reference and the captured images to regulate the estimated error to zero. Therefore, visual features based on local feature descriptors such as scale invariant feature transformation (SIFT) and speeded-up robust features (SURF) offer particular advantages for the purpose of the IBVS. Most importantly, these methods are independent of changes in scale, orientation, illumination, and affine transformations of objects. These methods also allow the controller to deal with the object occluded partially, which always occurs in dynamic environments. The proposed method requires three steps to perform the grasping task. In step 1, the robot searches around the workspace to find and detect the object using the window approach of the SURF method. In step 2, the robot uses the features obtained from step 1 in the visual servo controller to track the object. In step 3, the robot chooses the grasping position and grasps the object. Various experiments demonstrated that objects can be grasped safely and stably using the proposed method.

Figure 2.15: LWR robot with hand camera in grasping task [42].

Nishida et al. proposed a feedback control method based on simultaneous use of positional/visual sensors for a robot arm [44]. The positional sensors, which have been used in this method, are incremental encoder attached to each joint to measure its angle. A stereo camera set has been used as visual sensor to measure the position of the robot end effector. The

position and the orientation of each camera were set as shown in Fig. 2.16 to guarantee a sufficient positioning resolution. The proposed method has been implemented with a 3-DOF robot manipulator where feature points have been marked on the end effector and on the target object for visual feedback. The process starts firstly when the end effector is set to be outside of the camera view and then the robot moves toward the desired position by inverse kinematic analysis with the incremental encoders. Whenever the end effector is presented in the field of the camera view, the control process based on camera information starts working. The proposed method can guarantee the stability of the robot motion and achieve fine positioning. It also needs neither kinematic calibration nor camera calibration.

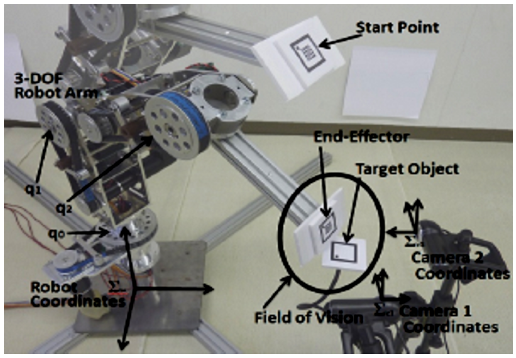


Figure 2.16: 3-DOF robotic arm with encoders and stereo camera [44].

Azad et a. presented a powerful stereo-based object recognition and 6D localization strategy which are built on the base of local features for the purpose of grasping with humanoid robot [45]. The work was implemented by the humanoid robot ARMAR to serve using its perception module for various grasping and manipulation tasks in a kitchen scenario as shown in Fig. 2.17. The strategy was performed with two integrated methods for textured objects as well as objects that can be segmented globally and are defined by their shape. Thus, it covers the cases of objects with complex texture and complex shape. In both subsystems, the 6D pose is calculated by making explicit use of the stereo system to attain maximum depth accuracy.

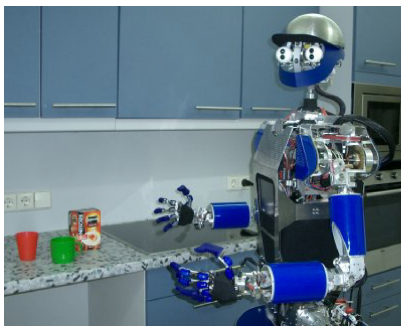


Figure 2.17: The humanoid robot ARMAR in a kitchen environment for grasping tasks [45].

2: Literature Overview

From the previous state of art, it is clear that the kinematic analysis for the robotic arm is very essential and required to control its motion. Tow general methods can be used to create the kinematic model, analytic and numeric. The numerical method doesn't guarantee to find the exact solution in comparison with the analytical solution. For the tasks of objects manipulation, the visual sensors play a main role in detection and localization the targets. High-resolution images with sufficient numbers of frames per second are required for these tasks. The 3D visual sensor like Microsoft Kinect gives very good solutions for the 3D mapping of the view. The depth information of the view can be obtained easily with the Kinect in comparison with the stereo vision model which needs a complicated sequences of image processing. The single camera can also be used for localization but just for specific objects, which are previously known, using scaling approach. Related to object detection and recognition, color segmentation is one of the methods which is optimal for real time applications. Different color systems can be used for segmentation like RGB, HSV, and YCbCr. Color segmentation with HSV color system can be considered as the most reliable way because it is insusceptible to illumination and lighting influence in comparison with RGB system. On the other hand, the object recognition using its local features is invariant to image rotation, scale and change in illumination. The SIFT and SURF algorithms are examples of using the local features descriptors to obtain a set of features' correspondences between the object model which is already saved in a database and the captured image. The depth information can also be used for object detection but just in the structured environment where the required objects are already known and their models are saved in a database. The feedback control is very crucial in case of dealing with unstable and weak robotic arm or using an inaccurate kinematic model. Different approaches are used to provide a feedback signal which guides the manipulator to the required pose. Angular sensors can be attached to the arm joints to provide feedback information related to the joints' angles. Touch sensors can also be used as a feedback control to guarantee a safe and secure grasping especially in case of dealing with different objects. Visual servoing can be considered as one of the common ways which points the arm end effector to the desired object in the required manner. Two major approaches can be used for visual servoing, eye-in-hand and eye-to-hand. Eye-in-hand approach can be performed using hand camera which can be single, stereo, or 3D visual sensor. The hand camera guides the arm end effector to reach the desired pose and compensate the errors. Eye-to-hand approach uses the visual sensor to track the end effector and the desired object simultaneously to decrease the positional error between them and to guarantee a perfect reaching to the target with the required pose. The design of the arm grippers is also one of the prerequisites which have to be taken in consideration to ensure a robust and reliable performance for the grasping tasks. In the next sections, an overview about the kinematic analysis, object detection and localization, and feedback control will be illustrated.

2.2 Kinematic Analysis

Modeling a robotic arm involves the study of its kinematic behavior. A kinematic model is concerned with the arm motion without considering the forces producing that motions. The kinematics of a robotic arm deals with the study of the geometric and time based properties of the motion and in particular how various links of a robot move with respect to one another and with time. It provides an analytical description of the spatial movements of a robot i.e. a relationship between position and orientation of robot end effector and its joint variables. The problem of kinematic modeling is usually categorized into two sub-problems. First is the forward or direct kinematics (FK), which is the problem of solving the Cartesian position and orientation of a mechanism where the kinematic structure and the joint configuration are given. The second sub-problem is inverse kinematics (IK), which computes the joint variables using the given information of a robot's end effector position and orientation. In case of serial robotic arms, the IK problem is more complex than direct kinematic problem [46].

2.2.1 Forward Kinematics

In robotics, it is often necessary to be able to "map" joint coordinates to end effector coordinates. This map or the procedure used to obtain end effector coordinates from joint coordinates is called direct or forward kinematics. To solve the forward kinematics, two commonly methods are used which based on Denavit-Hartenberg (DH) parameters and successive screw displacements. Both methods are systematic in nature and more suitable for modeling serial manipulators. Also geometric methods are frequently used by some researchers for the serial manipulators of relatively simple geometry [47]. In [48], a new method was derived (quaternion algebra) to solve the forward kinematic problem. The next subsections will clarify the transformation matrix and the D-H representations which are the basics to understand the problems of FK and IK and how to solve them.

2.2.1.1 The Transformation Matrix

For mobile robot arm manipulation, it is required to know the pose of desired objects in both camera coordinate and robotic arm coordinate. In Fig. 2.18, a simple robot arm example is depicted where the camera (c), arm base (w), joint (j), and gripper (g) coordinate frames are indicated. If the gripper has to pick up a widget, the widget's pose has to be found in camera coordinates. Then, we need to transform that pose into arm base coordinate (external calibration) to evaluate if it is accessible or not. Finally, we have to transform that pose from arm base coordinates to gripper coordinates to determine when we should close the jaws of the gripper. The key to do this is to find the homogeneous transformation matrix which is used to

calculate the new coordinate values for a robot parts relative to each other or other coordinate system [49].

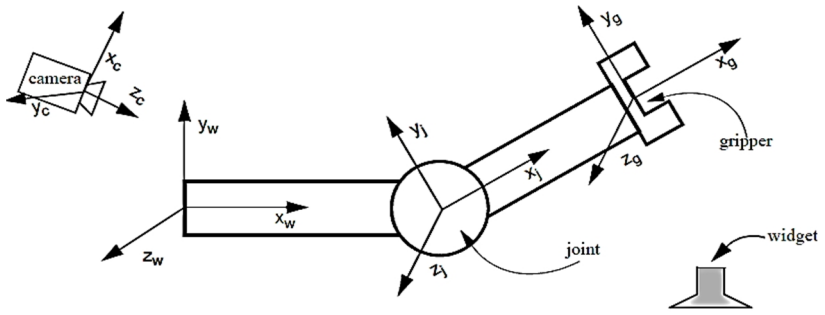


Figure 2.18: An example of simple robot arm with one joint [49].

In the previous example, if we want to find the gripper pose (g) related to the arm base (w), then we have to do the following:

1- Find T_1 transformation matrix from arm base coordinates (w) to joint coordinates (j).
2- Find T_2 transformation matrix from joint coordinates (j) to gripper coordinates (g).
3- Multiply T_1 by T_2 to find the final transformation matrix ($T_{Final} = T_1 . T_2$).

The final matrix from step 3 includes the position and orientation of the grippers related to the arm base as shown in Fig. 2.19. This is the way of solving the FK problem after substituting the joints values in the final matrix. Fig.2.20 shows the orientation system.

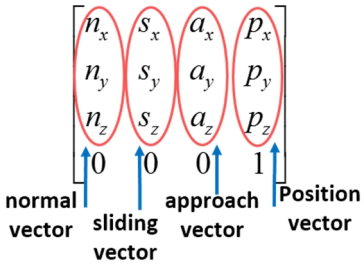


Figure 2.19: Transformation matrix.

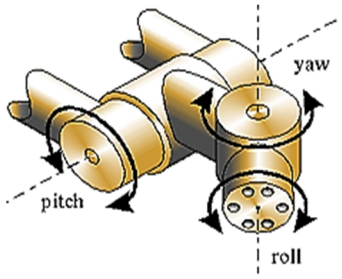


Figure 2.20: Roll, pitch, yaw orientation system [50].

The following Table shows how to find the transformation matrix from one point to another in the space including the translation and rotation:

Table 2.1: The summary of transformation matrix.

To convert the K coordinate frame into the J coordinate frame, then it has to:	To convert a point in J coordinates into a point in K coordinates, multiply that point by:
Translate Along K's X axis by a, Along K's Y axis by b, Along K's Z axis by c,	$\begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Rotate about K's X axis by θ	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Rotate about K's Y axis by θ	$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Rotate about K's Z axis by θ	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

The transformation matrix for fixed camera systems has to be computed related to different reference frames like robot arm base, or end effector using at least three points or features in the image plane [51]. The pose estimation involves this transformation by corresponding the 2D points in the camera with the 3D points of end effector [11].

2.2.1.2 The Denavit-Hartenberg Representation

The Denavit-Hartenberg representation is used to describe the translation and rotation relationship between the arm adjacent links where it provides a guide for locating coordinate systems on each link of a multi-link kinematic chain. Denavit and Hartenberg proposed to define the manipulator with four joint-link parameters for each link [52]. Fig. 2.21 shows a pair of adjacent links, link(i-1) and link i, their associated joints, joint (i-1), i and (i+1), and axis (i-2),(i-1) and i respectively. A frame {i} is assigned to link (i) as follows:

- The Z_{i-1} lies along the axis of motion of the i_{th} joint.
- The X_i axis is normal to the Z_{i-1} axis, and pointing away from it.
- The Y_i axis completes the right – handed coordinate system as required.

There are four parameters used in manipulator analysis. Three are fixed and are purely geometric, these are the link twist (α_i), the joint offset (a_i), and the last is the link offset (d_i). The final parameter is variable and it is the joint angle (θ_i) where (i) refer to the link number [53][52]. The definitions of D-H parameters are as follows:

- Link length (a_i): The distance measured along x_i axis from the point of intersection of x_i axis with z_{i-1} axis to the origin of frame {i}.
- Link twist (α_i): The angle between z_{i-1} and z_i axes measured about x_i-axis in the right-hand sense.
- Joint distance (d_i): The distance measured along z_{i-1} axis from the origin of frame {i-1} to intersection of x_i axis with z_{i-1} axis.
- Joint angle (θi): The angle between xi-1 and xi axes measured about the zi-1 axis in the right-hand sense.

By following the D-H rules, the homogeneous transformations between adjacent links are defined.

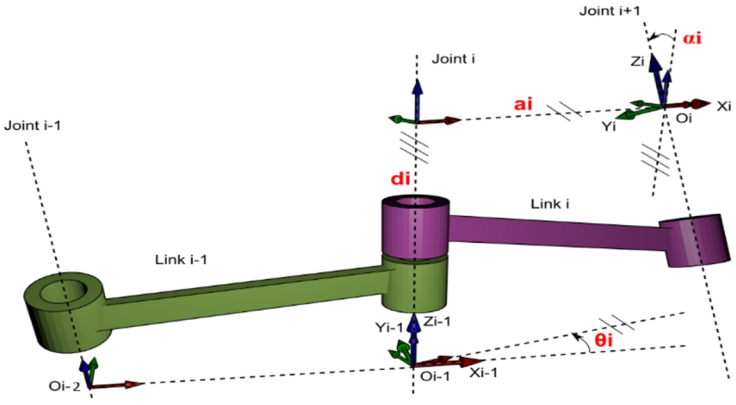
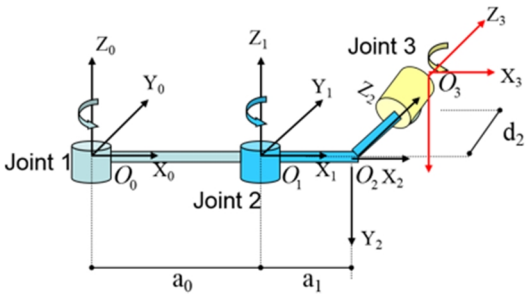


Figure 2.21: D-H conventions for frame assigning [52].

The Fig. 2.22 shows an example of a simple arm with its D-H parameters. This example shows clearly how to find the D-H parameters for the robotic arm. The joint offset (a_i) is the distance from the intersection point of (Z_{i-1}) with (X_i) to the origin point (O_i) along (X_i). The link twist (α_i) defines the relative location of the two-joint axis by finding the rotation angle from (Z_{i-1}) to (Z_i) about (X_i) using a right-hand rule for angle determination. The third parameter is the link offset (d_i) which is the distance from the intersection point of (Z_{i-1}) with (X_i) to the origin point (O_{i-1}) along (Z_{i-1}). The final parameter is the joint angle (θ_i) which is the rotation angle from (X_{i-1}) to (X_i) about (Z_{i-1}).



D-H link parameter table				
Joint i	α_i	a_i	d_i	θ_i
1	0	a_0	0	θ_0
2	-90	a_1	0	θ_1
3	0	0	d_2	θ_2

Figure 2.22: An example of 3-joints arm with its D-H parameters [54].

In general, the transform, that defines a frame (i) relative to the frame (i-1), should be constructed. By defining a frame for each link, the kinematics problem has been broken into (n) sub-problems depending on the number of joints and the degree of freedom (DOF). In order to solve each of these sub-problems, further process has to be achieved by breaking each sub-problem into four sub-sub-problems. Each of these four transformations will be a function of one link parameter only and will be simple enough. The four transformation matrices can be obtained using Table 2.1 as follows:

1- The matrix for the rotation of (θ_i) about the (Z_{i-1}) axis = $\begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

2- The matrix for the translation of (a_i) along the (X_i) axis = $\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$

3- The matrix for the translation of (d_i) along the (Z_{i-1}) axis = $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$

4- The matrix for the rotation of (α_i) about the (X_i) axis = $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

By multiplying the four matrices, the general transformation matrix (D-H matrix) can be obtained as follows:

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

2.2.2 Inverse Kinematics

The analysis or procedure that is used to compute the joint coordinates for a given pose of an end effector coordinates is called inverse kinematics. The solution of inverse kinematics plays an active role in object manipulation because it is an important issue to make the arm end effector reaches the desired object accurately. Basically, this procedure involves solving a set of equations. However, the equations are, in general, nonlinear and complex. Also, even if it is possible to solve the nonlinear equations, uniqueness is not guaranteed. There may not (and in general, will not) be a unique set of joint coordinates for the given end effector coordinates [55]. IK importance cannot be found only in robotics, but also in other fields like 3D games and augmented reality. In contrast to the forward kinematics, IK does not have a unique solution. The IK solutions which ensure collision-free operation and minimum joint motion are considered more optimum [56]. Fig. 2.23 shows the difference between forward and inverse kinematics. In forward kinematics, the arm joints' values are known and inserted to the forward kinematic model to find the end effector pose related to the arm base or shoulder. On the other hand, with inverse kinematic, the end effector or the required pose related to the arm base is known and inserted to inverse kinematic model to calculate the arm joints values. There are some issues, which have to be taken into the consideration for solving the inverse kinematic and controlling the robotic arm, such as singularities, joint limits and reachable workspace.

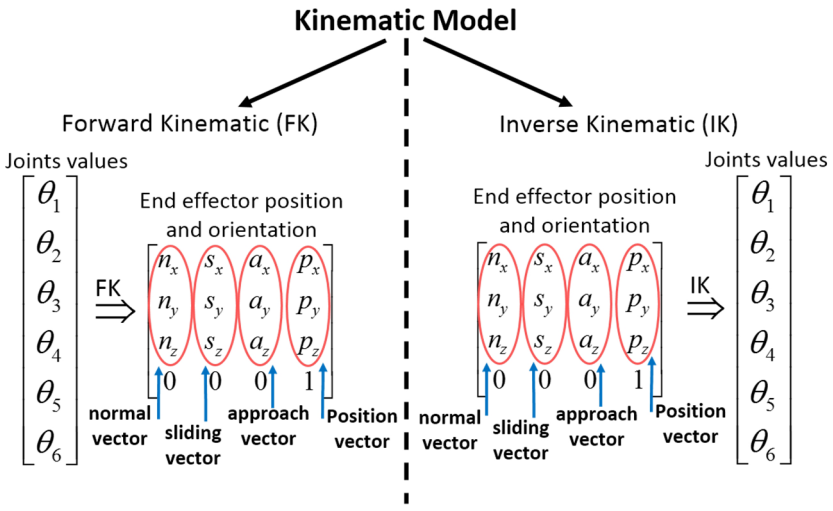


Figure 2.23: Kinematic analysis.

All proposed manipulator solution strategies of IK problem can be splitted into two broad classes: analytical solutions and numerical solutions. Because of their iterative nature, numerical solutions generally are much slower than the corresponding analytic solution and not guaranteed to find all the solutions [57]. Iterative numerical solution to kinematic equations is a whole field of study in itself [58]. Some researchers use the Kinematics Dynamics Library toolchain to provide a numerical IK solver for the robotic arms [59]. Within the class of analytic solutions, two methods of obtaining the solution are defined: algebraic and geometric (closed-form). The two methods are similar and differ perhaps in approach only by depending on the FK solution (using D-H convention) with polynomial equations. The calculating of numerical solutions is generally time consuming and giving an approximate value. Also, with the numerical methods, there is no guarantee to find the solution relative to the calculation with analytic expressions. Hence, it is considered very important to design a manipulator so that an analytical solution exists. Manipulator designers discovered this very soon, and now virtually all industrial manipulators are designed sufficiently simply that an analytical solution can be developed. Generally, there are many solution methods for the robot inverse kinematics problem. for example, Paul's analytic solution [60], Murray's classic elimination method [61], and Jacobian transpose methods [62][63]. Man et al. reported a mathematical approach to analyze the kinematics of a humanoid robot using the screw theory [64]. Analytical solutions of the inverse kinematics have been found for some particular structures of 6-DOF serial manipulator. According to the previous studies, inverse kinematics can be solved analytically if all adjoining joint axes are parallel or orthogonal to each other as in Fig. 2.24 of Puma robotic arm [57]. Lee et al. presented an analytical method with geometric approach for solving the inverse kinematics problem of PUMA robots [65].

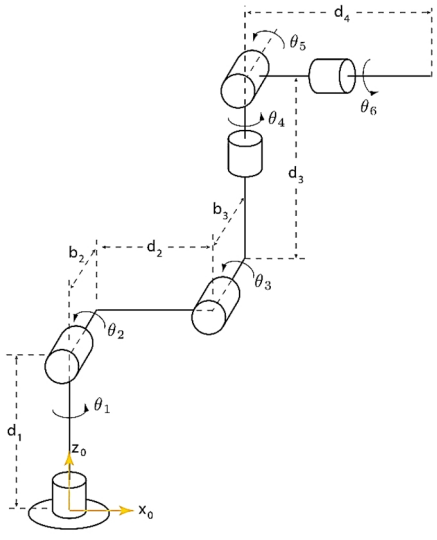


Figure 2.24: 6-DOF Puma robot arm [66].

Billingsley et al. reported several techniques to solve the inverse kinematic problems for different manipulators such as screw algebra, dual matrices, dual quaternian, iterative methods, and Pieper's approach for solving Euler angles [67]. The above methods can be demonstrated to work on a few and particular manipulators, but they do not offer a procedural approach to be used with every type of serial-link manipulator. Many of these methods are also not simple enough to automate or use immediately because they require the derivation of unique IK equations or algorithms for different classes of robot arms. Cubero presented a practical and robust "Blind Search" numerical method which can solve the joint variables for any type of serial-link manipulator, regardless of the number and types of DOF it has [67]. The IK method will search for small joint angle changes that are necessary to minimize the position error between the end effector frame and the target frame. The speed, stability and reliability of solutions depends heavily on the selection of suitable search parameters, such as step size to the next target point, incremental displacement magnitudes for each link, weighting factors for calculating the total error, and an error tolerance to define the acceptable error of IK solution. These variables need to be adjusted for each type of robot manipulator. Performance of these algorithms can be regulated by trial and error to achieve a satisfactory balance between solution accuracy, search stability, and computation time for real-time control. Nie et al. proposed a method of sequential retrieval to solve numerically the inverse kinematics problem for a 6-DOF robot arm which has not been analytically solved yet [68]. Aristidou et al. proposed a novel heuristic iterative method, called Forward And Backward Reaching Inverse Kinematics (FABRIK) for solving the IK problem numerically in different scenarios [69]. FABRIK avoids the use of rotational angles or matrices, and instead uses a forward and backward iterative approach and finds each joint position via locating a point on a line. Wu et al. proposed a method which depends on the training of neural network to solve the problem of inverse kinematics for 6-DOF robotic arm [24]. The method was performed by moving the end effector of the robot arm randomly to different positions and orientations within its workspace with saving the joint angles and the end effector pose for every movement. The procedure was repeated several times in order to better approximate the uniform distribution of the sample data points. T. Zhao et al. proposed a method to divide the IK problem of a 7-DOF humanoid arm into sub-problems to find the closed-form solution taking the constraint of the elbow position into consideration [70]. The 6-DOF robotic manipulators are widely used in the industrial applications because of their ability to reach the target in different orientations. For this reason, the kinematic analysis of the 6-DOF manipulators is a rigorously researched topic where the scientific community have been reported various modeling techniques to solve the kinematic problems. For examples, T. Ho et al. proposed a fast closed-form inverse kinematic solution for a specific 6-DOF arm [71]. Huang et al. proposed a strategy to find an analytical solution for 6-DOF arm which was verified by MATLAB simulation and by the real motion of

the robotic arm [55]. Xie et al. formulated the kinematic equations of 6-DOF space manipulator using a specific analytical method. The proposed method avoided a large amount of inverse matrices multiplications and the expression was convenient for trajectory planning and simulation [9]. Cui et al. studied the kinematics of 6-DOF humanoid robot manipulator and the inverse kinematic was found analytically through D-H convention where the geometry and square transformations were put forward in order to separate joints variables from kinematic equations [72]. Iqbal et al. developed the kinematic model of 6-DOF robotic arm analytically and analyzed its workspace. The model was validated using MATLAB and implemented on a real robotic arm [56]. In General, most researchers prefer the numerical methods for solving the IK problems to avoid the difficulty of finding the analytical solution. Normally, the analytical approach is appropriate for real time applications because all the solutions can be found and it is computationally fast in comparison with the numerical approach. As mentioned before, the analytical solution can be classified into geometric (closed-form) and algebraic. For the geometric method, the complexity of finding the IK solution increases when the manipulator has more than 4 joints. Furthermore, the solution approach cannot be generalized from one manipulator to another because it depends on the number of manipulator joints, their types, structure, and coordinates frames. The closed-form solution can only be found for specific types of robotic arms, which have particular structure with 6-DOF or less. For easy understanding of the analytical solution, an example for solving the forward and inverse kinematics problems of a 3-DOF robotic arm can be found in [73].

2.3 Visual Processing

The important role of visual sensors in robotic manipulators is the ability to provide feedback to the robot controller by analyzing the environment around the manipulator within the workspace. A vision-guided robotic system is more suitable for tasks such as grasping or aligning objects in the workspace compared to conventional feedback controls, like force feedback controllers, inertial and orientation sensors which are installed on the robotic manipulator. The main challenge in vision-based robot control systems is to extract a set of robust and useful feature points or regions and using them to control the motion of the robotic manipulator in real-time [74]. In fact, without visual information, manipulating devices can operate only in structured environments, where every object and its relative position and orientation is known a priori. With the need of real-time applications, visual sensors were started to be utilized in the automatic control to measure the geometric characteristics of the target. Some approaches related to the calibration process of visual sensors and also to the methods and strategies of object detection and pose estimation will be illustrated in the next subsections.

2.3.1 Camera Calibration

Camera calibration is the process of determining some specific parameters of the camera and it is the early step of computer vision applications. This step is a crucial for finding the pose of the target and also for 3D reconstruction of the required object. A camera has two sets of parameters, intrinsic parameters which describe the internal properties of the camera, and extrinsic parameters which describe the location and orientation of the camera with respect to other coordinate systems. A real time camera calibration is a vital requirement of the system since the algorithms of image segmentation have to be able to work under the changes of lighting conditions, scale and with blurred images, due to robotic vibrations [74]. The intrinsic parameters may be adopted to describe the transformation relationship between the camera coordinates and the image coordinates. Common intrinsic parameters of the camera include focal distance, the location of image formation on the objective center, the ratio of length and width for the pixel, and the parameters of twist, lens skew coefficients and distortion coefficients [33]. The extrinsic parameters, which include the rotation matrix and displacement vector, may be used in describing the pose of an object related to the coordinates frame of a specific 3D point using the specified camera [75]. Calibrations techniques rely on sets of world points whose relative coordinates are known and whose corresponding image-plane coordinates are also known. More details, about how to implement the calibration process for the cameras, can be found in [76-80].

2.3.2 Object Detection and Feature Extraction

The influence and impact of digital images on modern society, science, technology and art are huge. Image processing has become such a critical component in contemporary science and technology that many tasks would not be attempted without it. It is a truly interdisciplinary subject and is used in computer vision, robotics, medical imaging, microscopy, astronomy, geology and many other fields [81][82][83]. Image segmentation, defined as the separation of the image into regions, is the first step leading to image analysis and interpretation. The goal is to separate the image into regions that are meaningful for the specific task [84]. Zitova et al. classified the segmentation techniques into five groups which are: threshold based, edge based, region based, clustering based and deformable model based [85]. On the other hand, Lucchese et al. divided the image segmentation methods into three wide groups as follows [86]:

1. **Global Knowledge:** The most common algorithms based on the image global features are represented by histograms and contrast. Histogram thresholding is the simplest segmentation process. Many objects or regions at images are characterized by constant

reflectivity or light absorption of their surfaces. Then, a threshold can be determined to segment objects and background.

2. **Edge-based segmentation methods:** These techniques are based on information about edges in the image. Perhaps, these kinds of methods are the earliest approaches and still remain important in some computer vision fields. The functionality of edge-based segmentation algorithm depends on finding discontinuities in color values.
3. **Region-based methods:** These methods construct regions directly. The region growing methods are usually good when noise is present at regions. Homogeneity is an important property of regions and it is used as the main segmentation criterion in region growing. A criterion for homogeneity can be based on gray-level, color, shape local features and some others.

The vision research area for intelligent robots is called ”computer vision”, which is now regarded as a fundamental and scientific approach to investigate how artificial vision can be best achieved and what principles underlie it [87]. In the image processing, the noise, strong light and other interferences should be reduced before applying the required algorithms. Also, the captured images require sometimes light compensation due to the dramatic impact by the light status.

Object detections based on color segmentation is one of the quickest methods. The speed of this technique makes it very popular for real-time applications because it does not require a prior information about the target. There are different representation systems to define the space of color such as RGB, HSV, and YCbCr. The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors as shown in Fig. 2.25. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. The main purpose of using the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers. It has also been used in conventional photography.

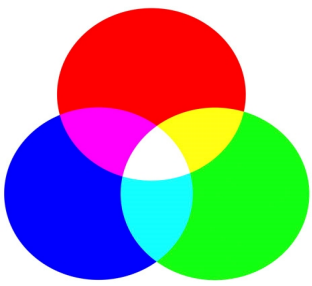


Figure 2.25: RGB color space [88].

HSV is one of the most common cylindrical-coordinate representations of points in an RGB color model, which rearrange the geometry of RGB in an attempt to be more perceptually relevant than the Cartesian representation. They were developed in the 1970s for computer graphics applications, and are used for color pickers, color-modification in image editing, and less commonly for image analysis and computer vision [89][90]. HSV stands for hue, saturation, and value, and is also often called HSB (B for brightness). In each cylinder, the angle around the central vertical axis corresponds to "hue", the distance from the axis corresponds to "saturation", and the distance along the axis corresponds to "lightness", "value" or "brightness". Fig. 2.26 shows an illustration of HSV color space.

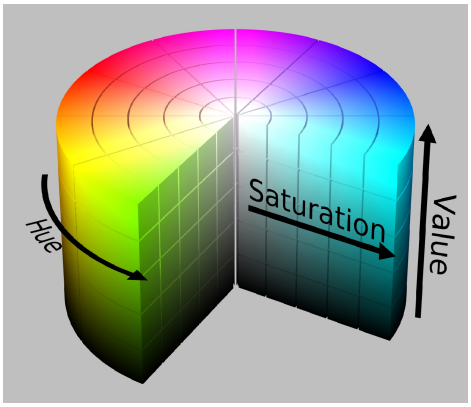


Figure 2.26: HSV color space [91].

YCbCr color space is also used for object detection where Y represents the luminance, Cb represents the chrominance of blueness, and Cr represents the chrominance of redness. It is important to adjust the values of Y, Cb and Cr, respectively according to the change of the light in the working environment. Related to the segmentation based on edge detection, there are a lot of techniques such as Canny, Sobel, Laplace, etc. as shown in Fig. 2.27. The Canny method has a good performances with noisy images [92]. This algorithm can only be applied on a grayscale blurred image, and it will find all the edges, based on which contours will be formed.

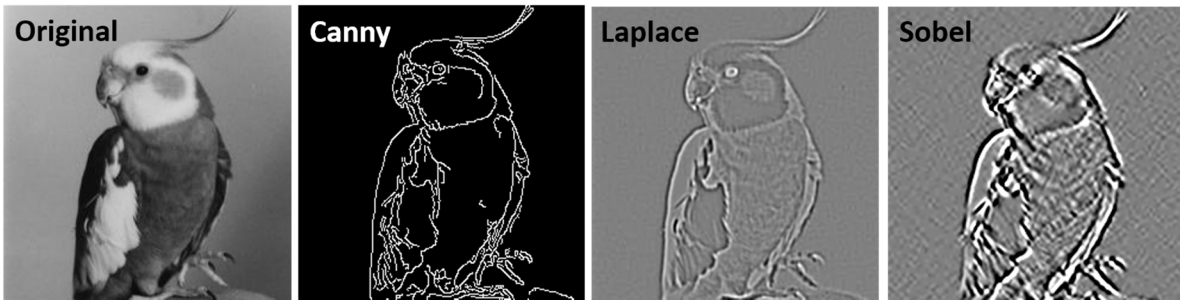


Figure 2.27: Different techniques for edge detection [93].

For corner detection, one of the most popular methods is Harris algorithm [94]. This technique implements invariance to any corner and its response will be positive in region with corners, negative in regions with borders and small flat regions. Fig. 2.28 shows the result of the Harris algorithm.

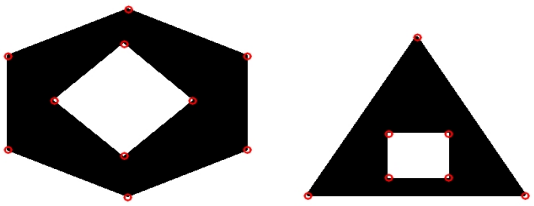


Figure 2.28: Harris cornere detection [95].

Shape is also one of the properties which can rely on it for image segmentation and object detection. Normally, shape detection techniques depend on the edges, corners, and lines detection to identify the required shape. Hough transform method is one the feature extraction techniques which is used in computer vision for line detection and shape recognition [96]. The most effective approach for object recognition depends on the successful extraction of a sufficient number of local features and textural information for each object. An advantage of the feature-based approaches is their ability to recognize objects in the presence of lighting, translation, rotation and scale changes. Several methods have been proposed for features detection. The most popular of them are SIFT (Scale Invariant Feature Transform) [26][97][98], SURF (Speeded-Up Robust Features) [43], FAST (Features from Accelerated Segment Test) [99][100], BRIEF (Binary Robust Independent Elementary Features) [101], ORB (Oriented FAST and Rotated BRIEF) [102], and Maximally Stable Extremal Regions [103]. Fig. 2.29 shows the result of object recognition using SURF method.

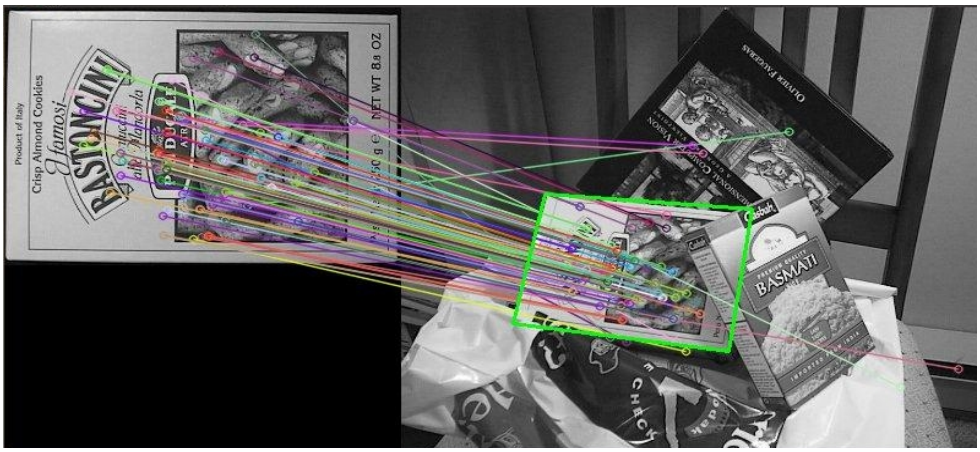


Figure 2.29: Matching between the captured image and the sample using SURF [104].

2.3.3 Pose Estimation for the Detected Object

In general, the pose estimation for the target is performed based on visual sensors images and feature correspondences. The description of the position and orientation of a 3D rigid body is significantly more complicated. The depth information of the desired object is the fundamental source to perform the pose estimation algorithms [105]. Fig. 2.30 illustrates the classification of depth measurements techniques:

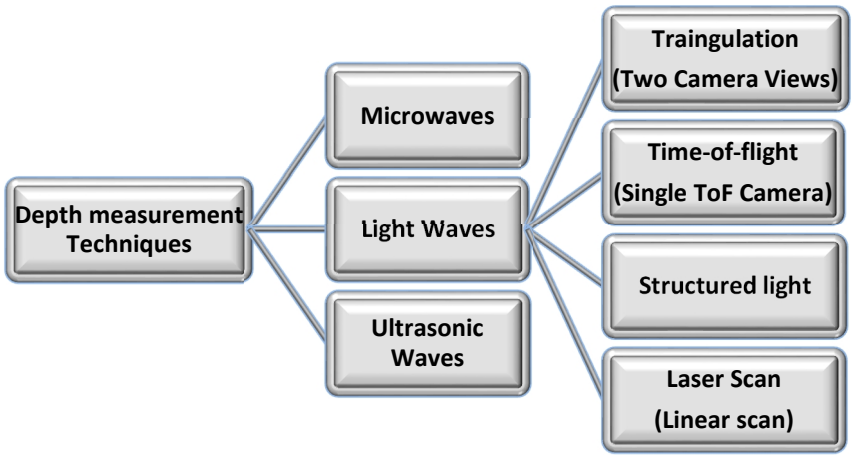


Figure 2.30: Classification of depth measurements techniques [106].

Higher accuracy depth estimation is commonly implemented in stereovision systems. With two camera, that are situated in a well know position, a distance to an object or point in the image can be estimated by applying a trigonometric function using a pure rotation between two images [107][108]. Robots with only one camera can also estimate the distance using scaling strategy for the objects which are previously known. The estimation is based on using a camera with zoom, then changing the focal length to acquire a set of images that help to estimate the position [109][110]. 3D camera like Kinect sensor V1, which uses the structured light technique with triangulation, is also one of the most important visual sensors. It provides a high quality color and depth information with high frequency performance that help to find new solutions for object localization problems [111-115]. More details about some techniques of pose estimation will be illustrated in the next subsections.

2.3.3.1 Stereo Vision

The stereo imaging capability is very familiar in the computer vision society because it imitates the ability that the human eyes give. Computers accomplish this process by finding correspondences between points that are seen by one camera and the same points which are seen by the other camera. With such correspondences and a known baseline separation between

cameras, the 3D location of the points can be computed. The search for the corresponding points in the left and right images can be computationally expensive, but some procedures can be performed to narrow down the search space as much as possible. In practice, stereo imaging involves four steps [116]:

1. Mathematically remove radial and tangential lens distortion. This step is called undistortion and the outputs will be undistorted images as shown in Fig. 2.31.

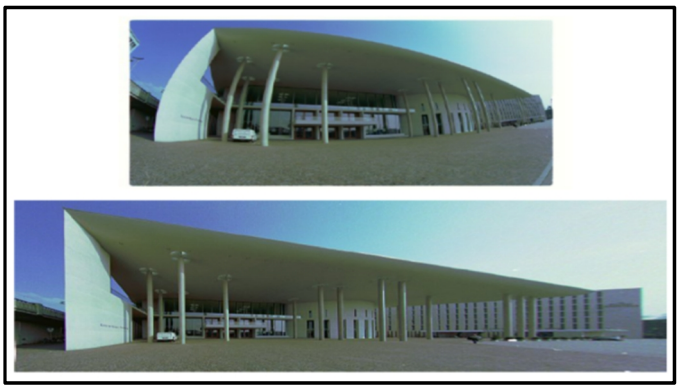


Figure 2.31: The removing of lens distortion [117].

2. Adjust for the angles and distances between the two cameras. This process is called rectification and the outputs will be images that are row-aligned and rectified as shown in Fig. 2.32.

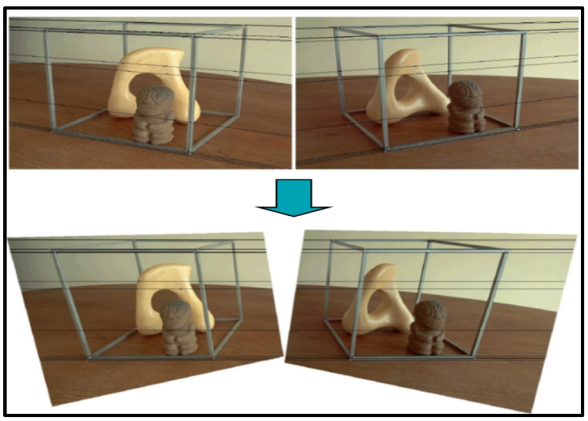


Figure 2.32: Images rectification [117].

3. Find the same features in the left and right camera views. This step is known as correspondence. The output of this step will be a disparity map, where the disparities are the differences in X-coordinates on the image planes for the same point viewed in the left and right cameras ($d = X_l - X_{r)}$ as shown in Fig. 2.33.

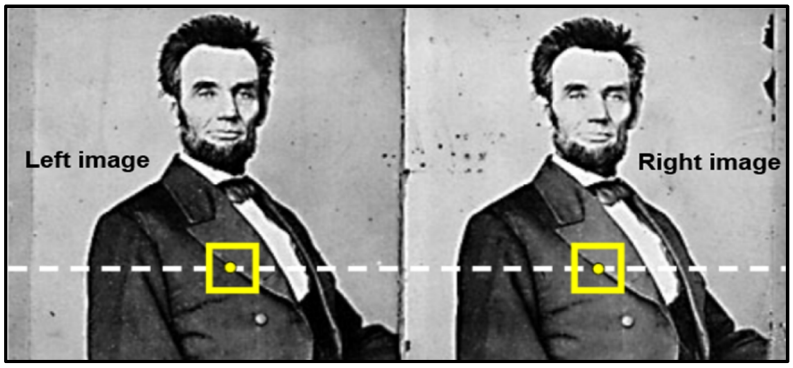


Figure 2.33: Finding the corresponding point between two images [117].

4. If the geometric arrangement of the cameras is known, then the disparity map can be turned into distances by triangulation. This step is called reprojection, and the output is a depth map as illustrated in the Fig. 2.34, and 2.35.

Related to Fig. 2.34, the upper black arrow is the axis of the real image plane, f is the focal length of the used camera, O is the center of projection, the blue arrow in the middle is the axis of the front image plane, X is the distance in X-axis between the required 3D object point B in the real view and the center of projection, D is the projection of B on the front image plane, X_i is the distance between D and the center of projection, Z is the distance in Z-axis between the required 3D object point B in the real view and the center of projection. Using the similar triangle theorem, it can be obtained that (X_i / f = X / Z). The focal length can be obtained using a prior knowledge or by performing camera calibration.

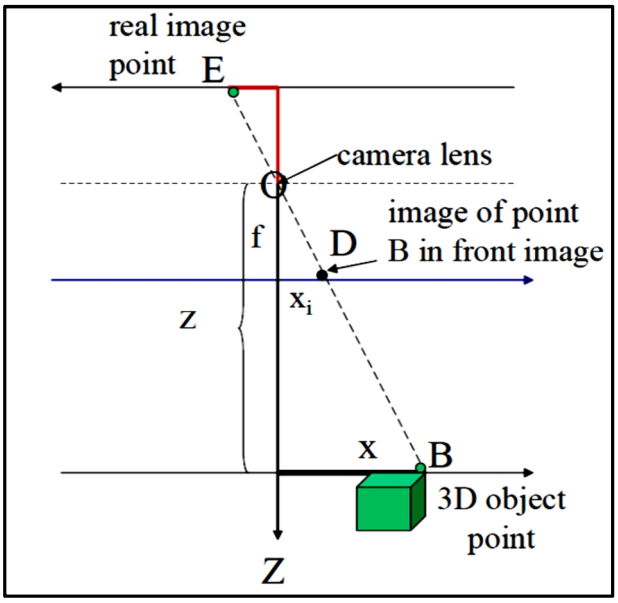


Figure 2.34: The 3D point in the image [117].

In the Fig. 2.35, it can be assumed that the stereo vision system is simple where the optic axes of the two cameras are parallel. With the similar triangle theorem, it can be obtained that: (Z / f = X / Xl), (Z / f = X-b / Xr), and (Z / f = Y / Yl = Y / Yr) where b is the baseline or the distance between the two cameras and Y-axis is perpendicular to the page.

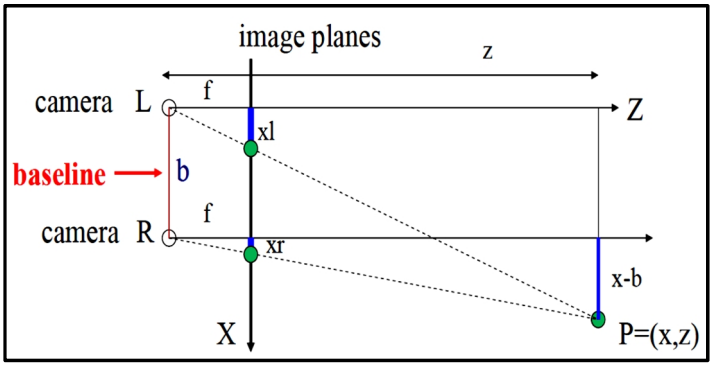


Figure 2.35: The 3D point in the stereo images [117].

According to the previous information, the position of point can be derived as follows [117]:

$$Z = f * b / (Xl - Xr) = f * b / d \quad (2.1)$$

$$X = Xl * Z / f \quad \text{or} \quad X = b + Xr * Z / f \quad (2.2)$$

$$Y = Yl * Z / f \quad \text{or} \quad Y = Yr * Z / f \quad (2.3)$$

The other process which has to be performed, for obtaining 3D map based stereo vision, is the calibration. Stereo calibration is the process of computing the geometrical relationship between the two cameras in space. In contrast, stereo rectification is the process of "correcting" the individual images so that they appear as if they had been taken by two cameras with row-aligned image planes. With such a rectification, the optical axes (or principal rays) of the two cameras are parallel and they are intersected at infinity. Stereo calibration depends on finding the rotation matrix R and translation vector T between the two cameras. The main difference between stereo calibration and single camera calibration is that, in single camera, a list of rotation and translation vectors between the camera and the chessboard views will be obtained. In stereo calibration, a single rotation matrix and translation vector that relate the right camera to the left camera will be obtained [116].

2.3.3.2 Structured Light and Time of Flight with Kinect Sensor

The Kinect sensor from Microsoft company supports to create new ways that make some sense in different fields of the life. The fusion of the RGB camera, the depth sensor and the microphone system into a single device support the software developers and give them the

opportunity to develop the computer interaction field. There are two versions of Kinect sensor where Fig. 2.36 shows the first version V1.

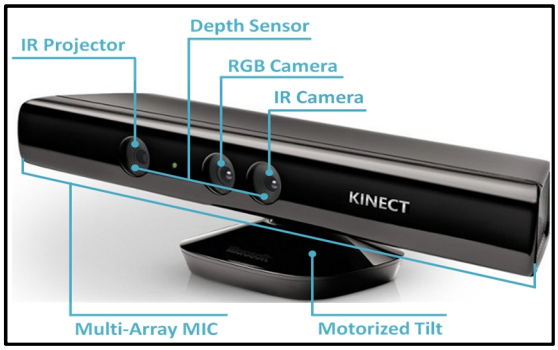


Figure 2.36: Old version of Kinect sensor (V1) [118].

The Kinect V1 uses the method of triangulation technique for position calculation where beams of laser are projected in view and an image of the scene is captured by an IR camera to register the position of the light spots. The distance is calculated by the equation [119]:

$$D = \frac{B}{\frac{x_0}{f}+1/tan\,\alpha} \tag{2.4}$$

Where D is the distance to the object that the laser spot was projected on, B is the baseline distance between the camera optical center and the laser projector, x_0 is the position of the laser spot in the image, f is the focal length of the camera and α is the angle between the optical axis of the camera and the projected laser. The Kinect V1 depends on the structured light imaging technique for depth measurement. It projects a known light pattern into 3D view, as shown in Fig. 2.37, and the depth calculation is performed on the basis of the triangulation between the known pattern and the observed reflected pattern. The distortion of the light pattern allows for calculating the 3D structure [120].

Figure 2.37: IR projection of kinect sensor [121].

To explain how the depth sensor of Kinect V1 works briefly, it can be said that the Kinect projects a known pattern (speckles) of IR light where the projection is generated by a diffuser and a diffractive element for IR light. Afterward, the IR camera observes the scene and the calibration between the projector and camera has to be known to Compute the 3D map of the initial frame and then compute the speckle shift in X-direction to renew the 3D map. The size and shape of the speckles depends on their distance and orientation to the Kinect sensor [120]. Also, Intel RealSense 3D Camera (F200), which is shown in Fig. 2.38, uses the same methodology of Kinect sensor V1 for depth calculation [122].

Figure 2.38: Intel RealSense 3D Camera (F200) [122].

On the other hand, the new version of Microsoft Kinect sensor (V2) uses "time of flight" technology to determine the features and motion of certain objects. It relies upon a novel image sensor that indirectly measures the time it takes for pulses of laser light to travel from a laser projector, to a target surface, and then back to an image sensor. By using the "time of flight" technology, the Kinect V2 can see just as well in a completely dark room as in a well-lit room. Fig. 2.39 shows the Kinect sensor V2.

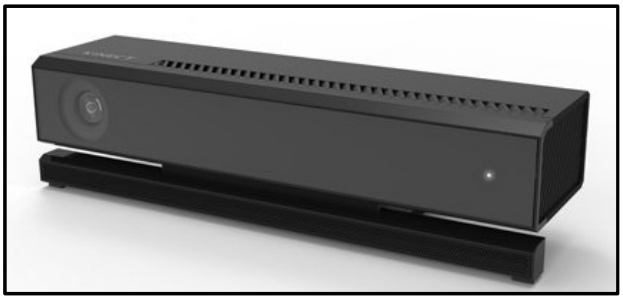


Figure 2.39: New version of Kinect sensor (V2) [123].

2.3.3.3 Pose Estimation Based Single Camera

The strategy of pose estimation can be performed with a single camera using scaling methodology where the object width and height are used for depth estimation. Object size technique is a classical depth estimation approach commonly used to detect the distance to the

object. The object area increases when it is closer to the camera, and the object area becomes smaller when it is far [124][107][125]. Model-Based object pose method has been described for finding the pose of an object which is previously known using a single image [126][127]. The method combines two algorithms; the first algorithm, POS (Pose from Orthography and Scaling) approximates the perspective projection with a scaled orthographic projection and finds the rotation matrix and the translation vector of the object by solving a linear system. The second algorithm, POSIT (POS with ITerations), uses also the approximate pose found by POS algorithm but in iteration loop to compute better scaled orthographic projections of the feature points. Then, POS will be applied to these projections instead of the original image projections.

2.4 Feedback Control

The feedback control for the robotic arm is very crucial to achieve high precision object manipulating tasks. The feedback control is very required especially in case of using inaccurate kinematic model or dealing with unstable arm which has weak joints. This can be achieved using sensors which are divided into two kinds, internal sensors and external sensors. The internal sensors, such as encoders attached to the joints, are used to detect the configuration of the robotic arm. The external sensors, such as cameras or touch sensors, are used to recognize the working environment. Motion stability was proven based on internal sensors or joint angle sensors in many researches [128]. For this case, the target pose must be represented by an internal sensor coordinates or a joint angle coordinates. Therefore, inverse kinematics model, which is the transformation from a target coordinates to a joint angle coordinates, is needed. If the coordinate's transformation includes some errors, the desired arm motion leads to failure in reaching the target even though the desired joints' angles are achieved. Therefore, additional external sensors as visual sensors are strongly required to solve this problem in many cases [129][130]. The visual sensors can guide the arm end effector to the desired pose even if there are errors in the kinematics model. However, visual sensors have some demerits in comparison with internal sensors. At first, the space of the camera view is limited. If a larger space is required, the resolution of the positioning degrades. At second, when the camera image is lost, the robotic arm cannot be controlled. It likely occurs when some unexpected objects blind the camera. At third, the sampling time of visual feedback is not short in comparison with internal sensors such as encoders. More details about using the internal and external sensors in the feedback control will be illustrated in the next subsections.

2.4.1 Joint Sensors

Different sensors can be attached to the joints to provide close-loop feedback control for the robotic arm motion. Angular sensor is one of solutions to provide a feedback value of joint

angle which can be affected during the arm motion or because of heavy load. Torque sensors can also be used for arm motion control by measuring the torques from sensors attached to the joints. The torque measured at joints is used to compensate the parameter variations and uncertainties of the arm caused by a load attached to the end of the arm [131][132][133].

2.4.2 Visual Servoing

A visual servoing system is a feedback control system based on visual information. It is essential for autonomous robots working in unknown or unstructured environments. In general, this system is composed of one or more cameras, a processing or computing unit, and a specific image processing algorithms to control the robot's end effector relative to the desired work piece. Reaching a particular position and grasping an object is a complex task which requires lots of sensing activities. These activities have to be performed in a right sequence and in a right time in order to make a smooth and stable grasp. One main fundamental observation with robotic systems using visual servoing for grasping is that a robot cannot determine the accurate spatial location of the object if the camera is located far away. Therefore, the robot should move closer towards the object for better accuracy. Based on the system error signal, there are two fundamentally different approaches to visual servoing control: Position-Based Visual Servo (PBVS) and Image-Based Visual Servo (IBVS) [134]. Position-based visual servoing (PBVS) uses the observed image features from a calibrated camera and a known geometric model of the target to determine the pose of the target with respect to the camera [135]. Knowing the target's geometry is essential for the pose estimation calculation. The camera intrinsic parameters (focal length, principal point, skew coefficient, and distortion coefficient) and the observed image plane features are needed as well. The robotic manipulator then moves toward the target pose and the control is performed in the task space. Image-based visual servoing does not include the pose estimation step. It uses the image features for arm control directly. The control is performed in an image coordinate space [136]. The desired camera pose with respect to the target is defined implicitly by the image feature values at the desired pose. The image features are highly non-linear functions of camera pose which make IBVS to be a challenging control problem. IBVS differs fundamentally from PBVS in omitting the estimation of the relative pose of the target. With IBVS, the control problem can be formulated in terms of image coordinates. Moving the feature points in the image space implicitly changes the pose in the Cartesian space. Based on the visual sensors configuration of the robot, the visual servoing systems are classified as two major classes called Eye-in-Hand and Eye-to-Hand [137]. In General, and in the robotic manipulation field, the IBVS approach control is used with Eye-in-Hand system where the camera is rigidly mounted on the robot end effector. On the other hand, the PBVS approach control is used with Eye-to-Hand system where the visual sensor is fixed in the workspace or on a fixed robotic part to observe the object or to

observe the end effector with the object to create a feedback control. One of these configurations or combination of them is used in various servoing tasks based on the requirements of the robotic applications and based on the manipulator stability and accuracy [138].

It is quite straightforward to apply the image based visual servoing approach to a multi-camera system [139][140]. If a stereo vision system is used and a 3D point is visible in both left and right images, it is possible to use it as a visual feature. In the case of using a monocular vision system where only a single camera is concerned, a certain number of assumptions, such as camera calibration are required to be made. Stereo visual servoing has many advantages over the classical monocular visual servoing approaches where the depth information can be recovered without any geometrical model of the observed object. Fig. 2.40 shows an example of a manipulator robot with a stereo vision system [141].

Figure 2.40: A six degrees of freedom robot using a stereo vision [121].

Robotic visual servoing tends to be widely used in medical and surgical applications to position instruments or perform the medical operations. For instance, “Laparoscopic Surgery” is minimally invasive, which needs only several small incisions in the abdominal wall to introduce instruments such as scalpels, scissors, etc., and a laparoscopic camera, such that the surgeon can operate by just looking at the camera images. To avoid the need for another assistant and to free the surgeon from the control task, an independent system that automatically aims the laparoscope is highly desirable. Several researchers tried to use visual servoing techniques to guide the instrument during the surgical operations as in [142-147]. In an another application, visual servoing systems are used with space robots to perform autonomous on-orbit servicing which includes approaching and docking to a target satellite and grasping some complex parts for the purpose of refueling and servicing as in [148-152].

2.4.3 Force Sensors

The two most important senses, which provide sufficient environmental information for human to perform interaction tasks, are the force sense and the visual sense. The target pose acquired from the visual sensor may be not accurate due to sensor limitations and bad calibration or occlusions. Therefore, touch or force sensors can be used with the robot to increase the precision of the object manipulation [153][154][155]. Tactile sensing technologies enable safer and enhanced interaction of robots with the environment for a more widespread application. The monitoring using haptic systems could help to reduce the severity and the errors of the grasping task. Tactile systems are based on arrays of sensors which cover the arm grippers. The sensors measure many contact parameters (force or pressure) at different locations for more precise grasping task. Chitta et al. presented an example of using the tactile sensing with grasping [156]. In this example, the gripper is equipped with fingertip capacitive sensor arrays, each consisting of 22 individual cells distributed on the tips, sides, and inner pads. During the final stages of the grasp, set of reactive behaviors based on information from these sensors can be performed to adapt the grasp for the unexpected contacts. The first reactive behavior attempts to maneuver the gripper around the object when unexpected contacts are detected by backing up and moving in the direction of the contact. The second behavior executes a compliant grasp when one fingertip contacts the object before the other. The arm is moved in coordination with the gripper to keep that fingertip in place, preventing the object from being pushed out while the gripper closes. The final behavior adjusts the grasp tasks, which are likely unstable, by shifting the end effector in an attempt to center the contact points between the fingertips. This can be judged by seeing the contacts only at the fingertips or only on one side of the fingertip sensor arrays. Fig. 2.41 shows an example of touch sensors (pressure sensors) that are installed on the gripper.

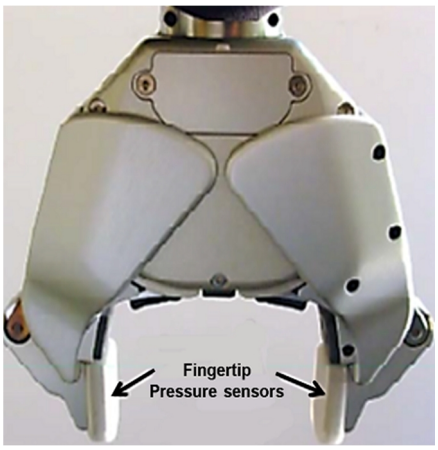


Figure 2.41: Pressure sensors on the grippers [154].

At the end of literature overview, it is clear that the kinematic analysis of the robotic arm is very essential to guide it to the target. Different methods can be used to solve the kinematic problem but the analytical methods are the most complicated and accurate at the same time. They are computationally faster than the numerical methods and the exact solutions can be obtained which make them preferable for real time applications. The robotic arm needs the pose information of the target to be inserted to the inverse kinematic model to perform the manipulation task. The visual information of the target can be considered the best and more suitable reference to guide the arm. The stereovision sensor and the 3D cameras like Kinect are the most common sensors in the robotic applications. But, the Kinect is the most preferable sensor for such applications because it provides the view depth frame directly. The required target, which has to be manipulated, can be found in the view using different strategies. The general features of the target like its color or shape can be used for detection purposes. Also, the target local features like edges or corners can be utilized for matching process to recognize the target. Several coordinate transformation steps have to be applied to transform the target position from the camera space to the manipulator space.

In the labs of the center for life science, H20 mobile robots are used to manipulate and transport different labware. A general introduction about the H20 mobile robot and how it moves in the working environment will be illustrated in the next section. It is crucial to give an overview about the features of H20 mobile robot with its operation and localization systems to show its ability and to consider the requirements that is needed for system development.

2.5 H20 Mobile Robot

The H20 robot is designed and built on i90 robot base and its main features are indoor GPS navigation system, touch screen tablet, two arms, dual-camera animated head (Hawk head), and auto-docking / recharging station. Fig. 2.42 shows the H20 robot with its sensors and external components [157]. The touch screen tablet on chest is used for displaying images and playing video and audio. The dual arms consist of six joints with 6-DOF + 2-DOF gripper. They reach about 60cm (2ft) as a maximum distance with max lifting weight of 800g (optional 1 kg). The animated head has dual 640x480 color cameras which are fixed on pan-tilt module. The dimensions of H20 robot are 43cm (L) X 51cm (W) X 140cm (H). It has an auto-docking and recharging station where 2 hours' nominal operation time are provided for each recharging. It has an indoor GPS and localization system which consists of stargazer sensor with artificial ceiling landmarks to provide position and direction information covering the working area. The navigation and localization system provides a collision-free point-to-point autonomous navigation with max speed of 75 cm / sec. The navigation sensors include 5 sonar and 10 IR range sensors. The range sensors are used for collision avoidance and for environment

detection. The H20 is fully wireless networked (802.11g) with tele-operation and remote monitoring and its application development tools are OS independent.

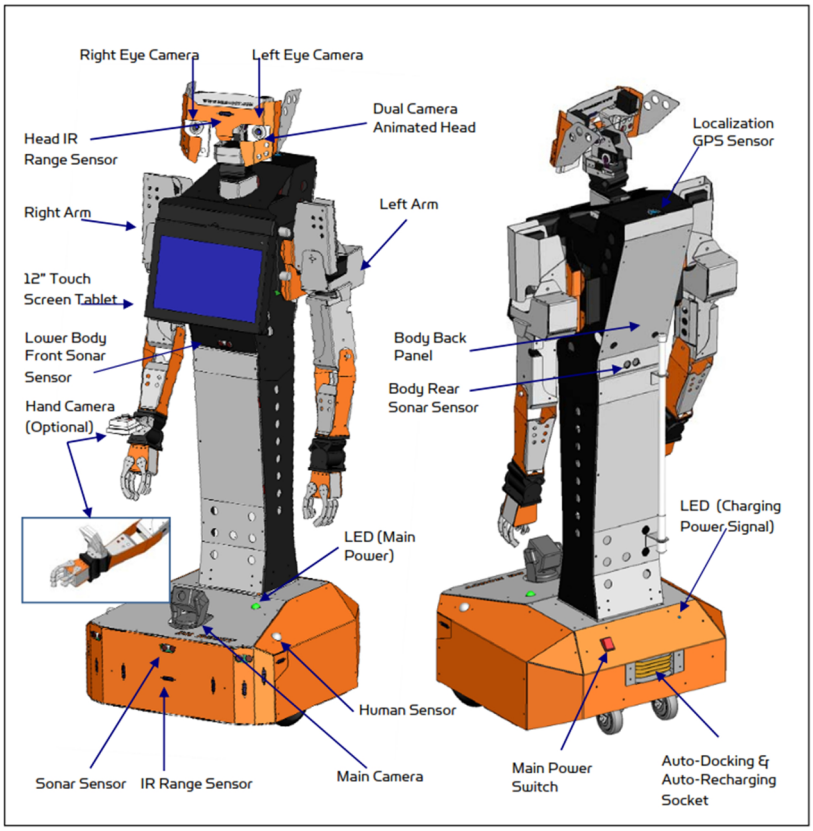


Figure 2.42: H20 Mobile robot with its specifications [137].

2.5.1 H20 Operation Strategy

The H20 is a wireless networked robot and also there is a local network for connecting the host PC to the other devices on robot via Ethernet. The host PC is used to run the H20 Control Center and Services program and also used as display. The robot network is connected to the user network wirelessly using IEEE 802.22b/g/n wireless bridge as shown in Fig. 2.43. It includes a pair of PicoStation, one is on the robot and the other is connected with the user local network and it is called ground PicoStation. The host PC could be off the robot and can be connected with it via either network cable (directly to the ground PicoStation or through a network switch / hub) or wirelessly. The robot can be controlled by the user via a remote Client program (Dr Robot Inc., Ontario, Canada) from anywhere all over the world using internet connection. Two main issues have to be considered to proceed the robot navigation. The first issue is the localization that makes the robot able to determine its position and orientation within its environment, and the second issue is the path planning which lets the robot know where it should go and how.

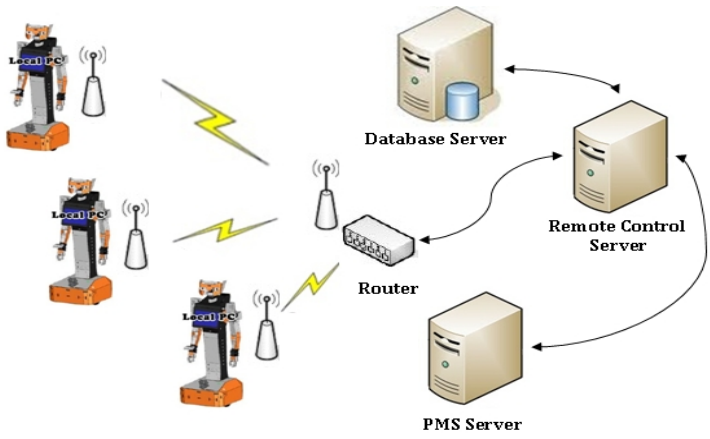


Figure 2.43: Robot outside Server/Client communication architecture [158].

2.5.2 H20 Localization System

The localization system of H20 mobile robots depends on stargazer sensor with ceiling landmark. Stargazer is a unique sensor system for indoor localization of intelligent mobile robots and it consists of IR projector and an image processing unit. The unit analyzes and processes the image of infrared ray that is reflected from a passive landmark with a unique ID which is fixed on the ceiling as shown in Fig. 2.44. It is normal for this sensor to be influenced by some sources in the environment such as sunlight rays or fluorescent light [159].

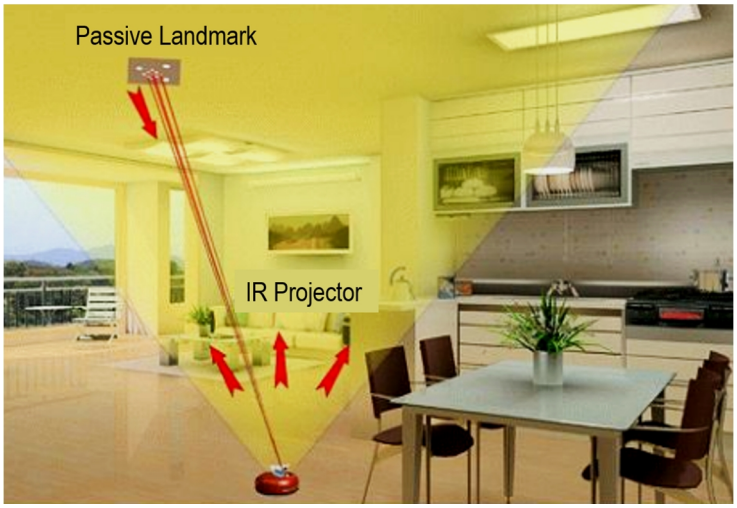
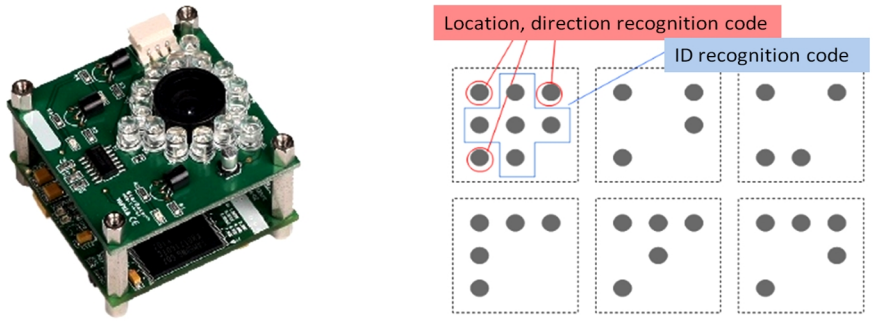


Figure 2.44: IR projector with passive landmark [159].

The H20 robot can determine its position in the room using the stargazer sensor on the robot and the landmarks that were mounted on the ceiling of that room. The stargazer provides the robot position and direction information in collaboration with landmarks which can be distinguished from each other by coding as shown in Fig. 2.45.



(*A*) (*B*)

Figure 2.45: (A) Stargazer sensor, (B) Passive landmarks [159].

The range between the sensor and landmarks is measured and calibrated automatically. Also, the relationship between two landmarks (according to the position and direction) is measured and calibrated automatically through the landmark mapping procedure. Each localization sensor can have up to (4.095) landmarks which don't need any electrical power. The Fig. 2.46 shows the localization sensor with ceiling landmarks in the working environment.

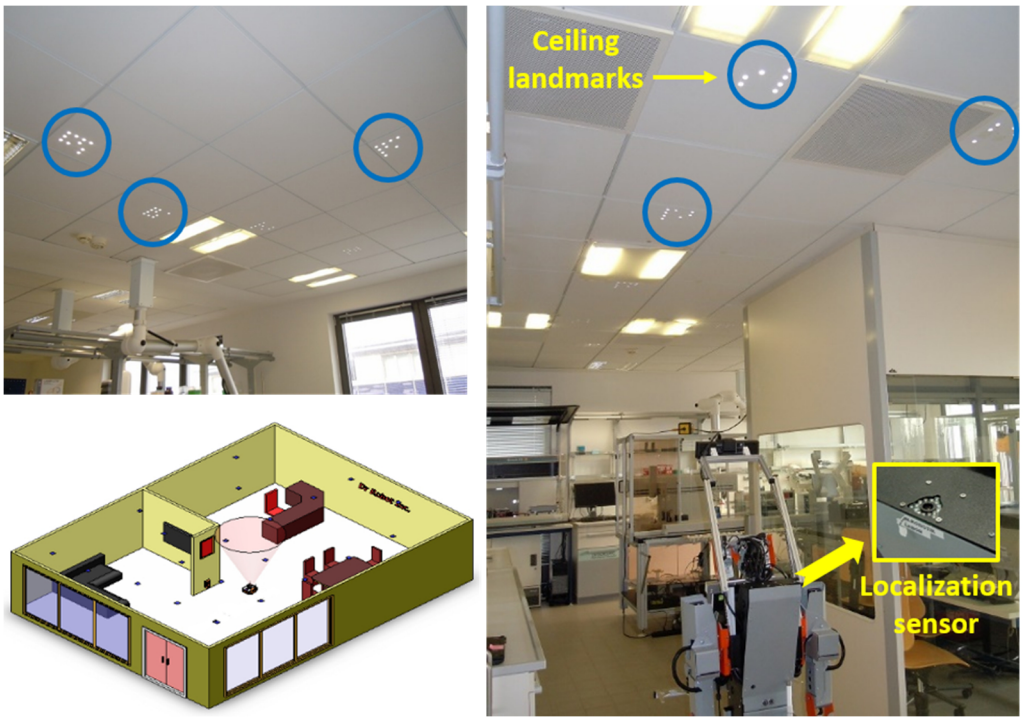


Figure 2.46: Localization sensor and ceiling landmarks [157].

3 The Work Concept

The laboratories future in the life science field is based on the development of automation technologies to ensure high throughput, workflow optimization, and reliable measurement results. Different kinds of stationary robots can be used to realize these purposes by performing the required tasks on the workstations and automation islands. These workstations, which are located in multiple labs, need to be connected with each other to improve the whole process. This can be achieved by using the mobile robots that are significant to increase the productivity by ensuring 24/7 operation and to save the human resources by reducing the tedious and routine work for the employees. The goal of this work is to use the mobile robots to connect the automation islands in the life science laboratories. This goal can be implemented by transporting different kinds of labware and tube racks. These labware contain chemical and/or biological components that require a secure grasping and placing actions on the workbench. When planning the pick and place motion, one have to successively plan:

1. The end effector pose to grasp the labware,
2. The end effector pose to place the labware at the desired location,
3. The stable pose of the labware on the workbench,
4. The trajectory of the robot arms to achieve the grasping and placing tasks.

This plan requires essentially the technology that support the robotic arms with finding the joints configuration to place its end effector at the desired pose by solving the inverse Kinematics problem. The other essential issue is to recognize and localize the desired labware by extracting the related image features and compute its pose relative to the arm coordinate system. The arm gripper design has to be taken into consideration to realize a secure manipulation for the target. Also, this gripper can be attached with tactile sensors which are important in case of dealing with different objects in the environment. Fig. 3.1 shows the structure of working concept.

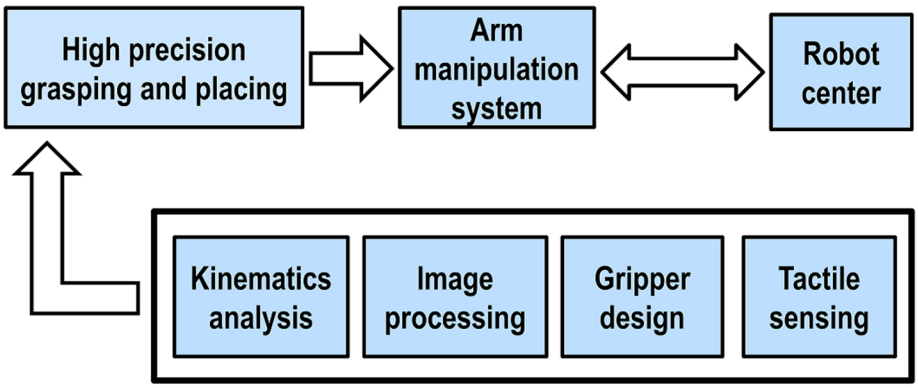


Figure 3.1: The structure of working concept.

3.1 The Implementation Strategy

In the center for life science automation, five H20 wheeled mobile robots, which have dual 6-DOF arms, are used for maneuvering between the adjacent labs for transporting multiple labware and tube racks. The H20 has a guidance control system which depends on stargazer sensor with ceiling landmarks. This system inevitably causes errors in the robot pose whenever it arrives the automation islands. The errors are related to the strong lighting which affects the recognition of ceiling landmarks by the stargazer. The odometry system, which includes encoders mounted on the robot wheels, causes also pose errors by providing inaccurate motion information. The odometry system accumulates errors for different reasons such as different wheels diameter and wheel-slippage. Therefore, it is crucial to localize the required labware every time the robot reach the workbench. The joints of the H20 arms have to be configured using the inverse kinematic model to guide the gripper to the goal. It is necessary to solve the inverse kinematic problem analytically to provide the exact values for the arm joints. Multiple labware and placing holders need to be recognized and localized visually. Therefore, distinct and adequate visual features for the required objects have to be existed. Color segmentation with shape recognition and features matching algorithms can be used for identification purposes. The effects of sun light and strong lighting condition in the working environment have to be avoided to realize successful identification process. The workstation consists of 6-8 positions for different labware. This wide workstation, which reaches about 1.5m length, requires a suitable visual sensor. This sensor has to be fixed on a proper position on the H20 body to provide a clear view where the stereo vision of H20 head is not compatible for this task. The Kinect sensor is considered the most appropriate 3D camera which can provide the normal image with the related depth information to localize the labware or the placing holder. The grasping and placing strategy for the required labware has to be performed wherever the goal is located on the workbench. This requires changing in the robot position to realize successful manipulation. According to arm structure and to the different labware designs, it is not possible to manipulate the labware directly. Therefore, labware containers are necessary to be designed for the transportation task. The labware container design determines the shape of arm gripper. Fig. 3.2 shows the architecture of control strategy for the arm manipulation system.

Multiple workstations are connected with each other by the cooperation of stationary and mobile robots. The overall procedure of the labware manipulation starts with the user or with the process management system (PMS). The management system decides which target has to be manipulated by the robot. It is vital to separate the arm manipulation system to multiple platforms to simplify the coding tasks and to facilitate finding the coding bugs. The arm

manipulation system can be separated to two parts which are the target localization and the arm controller. The target localization software utilizes the information received from the visual sensor to detect the target and to estimate its pose. The target information is sent to the arm controller software to guide the robotic arm and to perform the task. The connection between the two parts is performed using a TCP/IP socket through client-server communication model. This model is used to exchange information between the connected parts. This strategy is very useful also to realize the system integration with the other parts. Fig. 3.3 shows the block diagram of manipulation process.

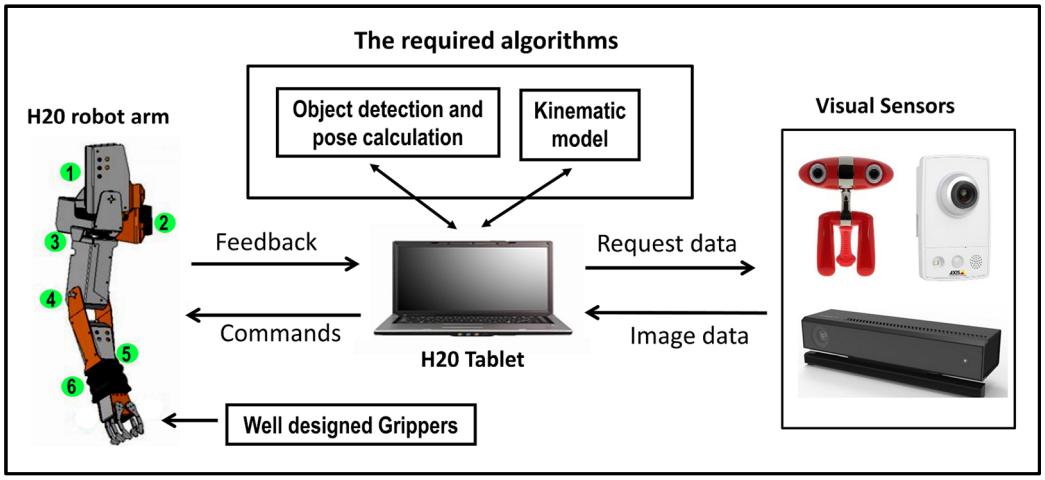


Figure 3.2: Architecture of control strategy for the robotic arm manipulation.

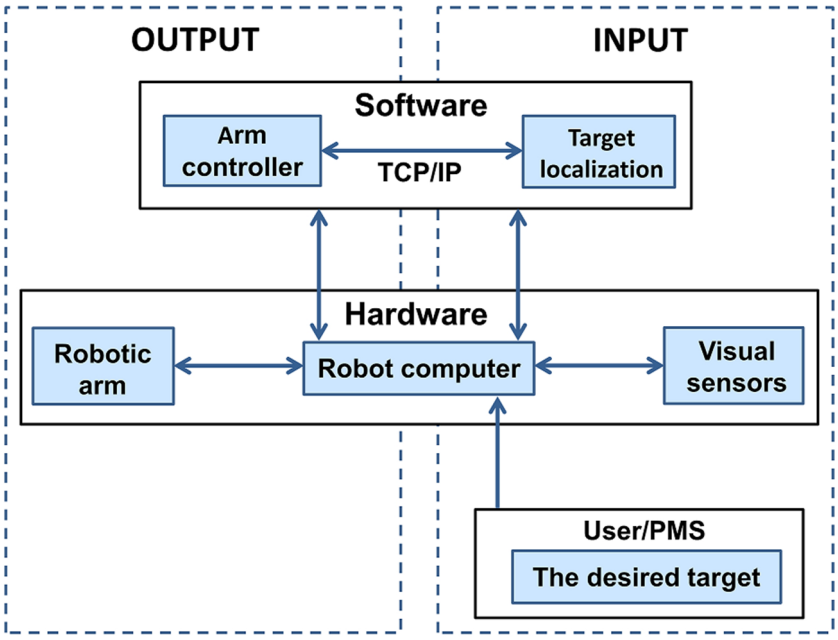


Figure 3.3: The block diagram of manipulation process.

3.2 The Concept of Kinematic Analysis

The kinematic analysis is very essential to control the robotic arm and enable it to reach the required pose of the target. This process requires finding the solution of kinematic problems. The solving of forward kinematic problem of serial robotic arm is easy in comparison with the solving of inverse kinematics. Different methods can be used to solve the IK problem where the suitable method can be used according to the specifications of the robotic arm. In the next subsections, the solvability of IK problem for the H20 arms will be explained. Also, some other related issues will be illustrated, which have to be taken into the consideration when the kinematic model is used to control the robotic arm, like, singularities, accuracy with repeatability, and reachable workspace.

3.2.1 Solvability

As mentioned before in chapter 2, the IK problem can be solved using two approaches: analytic and numeric. Most researchers resort to the numerical methods for solving the IK problems to avoid the difficulty of finding the analytical solution. Normally, the analytical approach is appropriate for real time applications because all the solutions can be found and it is computationally fast and accurate in comparison with the numerical approach. The analytical solution can be classified into geometric (closed-form) and algebraic. For the geometric method, the complexity of finding the IK solution increases when the manipulator has more than 4 joints. Furthermore, the solution approach often cannot be generalized from one manipulator to another. The closed-form solution can only be found for specific types of robotic arms, which have particular structure with 6-DOF or less. These robotic arms are characterized either by having several intersecting joint axes, or by having many of the link twist equal 0 or ±90 degrees and many link offsets and link lengths equal zero. Peiper indicates that in case there are 3 consecutive joints axes which are parallel to each other or intersecting at a single point then the closed-form solution can be existent [160]. The closed-form solution for the H20 arms can be found because the three shoulder joint axes intersect at a single point. It is very necessary to use the closed-from method with the H20 arms because this analytical method gives the exact solution with accurate values which is required to perform the grasping and placing tasks in high precision way. The required accuracy of reaching the target has to be less than 1 cm because the H20 robot deals with labwares which contain chemical and biological components.

3.2.2 Workspace Analysis and Reachable Space

Objects in 3D space have 6-DOF, three for position at space and the other three for orientation. In mobile robotics, it is very common to situate the objects in 3D space referenced by arm base frame. Therefore, it is necessary to know whether the robotic arm can reach the desired pose or not. The workspace of a robot arm is a primary performance parameter in addition to its speed and accuracy. The workspace of an arm, also termed as its work envelope, expresses a robot's ability to reach specific area. Given the information about range of motion (ROM) of arm joints and length of its links, the workspace can be determined [56]. The calculation of workspace is very important to decide whether the desired object is inside the reachable space or not. The reachable space can be calculated by randomly sampling the joints' values while using the forward kinematics to determine the pose of the end effector [161]. Since the reachable space is linked to the arm base, the target pose which is given in the global coordinate system has to be transformed to the shoulder coordinate system [162]. The workspace of the H20 arms has to be found to decide if the grasping or placing tasks can be performed or not. The H20 robot has to be close enough to the workbench so that the location of the labware within the arm workspace can be guaranteed.

3.2.3 Kinematic Singularities

Singularities are manipulator configurations in which one or more degrees of freedom are made redundant. This reduces the ability of the arm to move in 3D space, even though the area could be within its workspace. At a singular configuration, the manipulator loses one or more degrees of freedom. The singular configurations are classified into two categories based on the location of end effector in the workspace: boundary singularities and internal singularities. In the all situations, it is essential that singularities are avoided. Therefore, one of the important criterion for a good design of the manipulator configuration is to minimize the singularities. In many cases, IK solutions cannot be found since the end effector frame is located near to an internal singularity within the workspace, causing IK equations become unsolvable. Such problems are described by McKerrow in [163]. It is necessary to solve the IK problem for the singularity cases of the H20 arms to enable the end effector reaches all the possible pose within the workspace.

3.2.4 Trajectory Selection and Estimation

When controlling a robotic arm to go from one position to another, it is not enough to determine the joints' values and end effector position. It may be necessary to continuously control the trajectory or the path taken by the arm during its moving toward the target position. This is essential to avoid obstacles in the workspace. More importantly, there are tasks where the

trajectory of the end effector is critical. For example, during robotic welding process, it is necessary to maintain the end effector at a desired orientation and at a fixed distance away from the work piece while moving uniformly along a desired path [8]. The conventional path planning problem formulation involves searching the configuration space of a robotic system for a collision-free path that connects a start configuration to a goal configuration. Some problems have to be taken into the consideration for arm path planning. For example, the target may not be collision-free because of the obstacles in the environment. Also, the target may be unreachable from the current arm configuration causing the planner to fail, even when collision free alternative inverse kinematic solutions exist [164]. If the configuration, which was found, brings the end effector to the desired pose, the IK solution has to be checked against self-collisions, collisions with obstacles and joint limits in order to avoid invalid configurations. If the checking reports collisions, the solution is rejected and the search is continued [30]. For the example in the Fig. 3.4, one of the two possible solutions to reach the point (B) causes a collision. The fact that a manipulator has multiple solutions causes problems, because the system has to be able to choose one. For example, if the manipulator is at point A and we wish to move it to point B, a good choice would be the solution that minimizes the amount that each joint is moved. Hence, in the absence of the obstacle, the upper dashed configuration would be chosen [57]. Some approaches are proposed to avoid these difficulties of trajectory estimation by integrating the search for inverse kinematics solutions directly into the planning process as in [165-169].

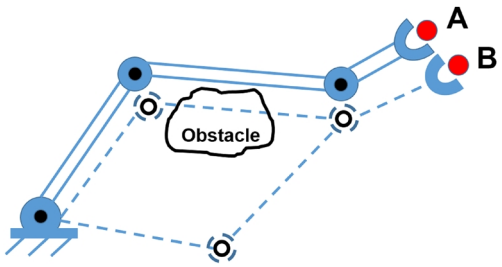


Figure 3.4: Simple arm with obstacle [57].

3.2.5 Repeatability and Accuracy

The repeatability of the manipulator means how precisely a manipulator can return to a taught point. Any time a goal position and orientation are specified in Cartesian terms, the inverse kinematics of the device must be computed in order to solve for the required joint angles. Systems that allow goals to be described in Cartesian terms are capable of moving the manipulator to points that were never taught or points in its workspace to which it has perhaps never gone before. Such points can be called as computed points. Such a capability is necessary for many manipulation tasks. For example, if a computer vision system is used to locate a part

that the robot must grasp, the robot must be able to move to the Cartesian coordinates supplied by the vision sensor. The precision with which a computed point can be attained is called the accuracy of the manipulator. Calibration techniques can be devised that allow the accuracy of a manipulator to be improved through estimation of that particular manipulator's kinematic parameters [170]. The repeatability and the accuracy tests of H20 arms are very necessary to determine the requirements to realize a successful grasp and place for the labware.

3.3 The Concept of Visual Perception

Different approaches with different sensors related to the visual perception have been studied in chapter 2. In this section, the most effective methods of object detection and pose estimation with suitable visual sensors will be illustrated.

3.3.1 Object Detection and Recognition

Different features can be extracted from the images to perform the object detection and recognition such as color, shape, edges, size, and local features. Some methods use multiple image features to create a robust strategy for detecting the desired object. The most important features which can be used in object detection and labware identification will be illustrated in the next subsections.

3.3.1.1 Color Detection and Segmentation

Color segmentation is one of the easiest methods for tracking an object from one image frame to the next. It is very popular for real-time applications where it does not require a prior information about the target. There are different representations of color system which can be used to extract the desired object form the images such as RGB, HSV, and YCbCr.

RGB-based Feature Extraction: RGB is a convenient color model for computer graphics since the human visual system works in a way that is approximately similar to an RGB color space. Since changing the brightness and lighting conditions cause a lot of differences and problems in the captured colors, tolerances for red, green and blue values has to be taken into consideration to detect the objects based on RGB color space. Predefined RGB values for each feature color are saved, then for each frame captured by the vision system, all the pixels are examined and the pixels located in the range of the saved RGB values with considered tolerances are detected and marked. According to specific equations and with having the total number of detected pixels (N) with their positions in image coordinates, it is possible to calculate the centroid of the detected object which is important for manipulation tasks. While it is possible to simply use the RGB values for object detection, it is often not the desirable

approach, since factors such as the viewing orientation and location of the light source affect these values. Weijer et al. proposed some color invariant models in an attempt to rectify this issue [171].

HSV-based Feature Extraction: The most important flaw about RGB-Based detection methods is the changing of light and brightness in the environment which affects the detection so badly and sometimes it is not possible to identify the color objects at all. Knowing the fact that the brightness is an independent value in HSV color space, a proper tolerance can be considered to cover all the values of a predefined color. This will certainly make the implementation of the color detection more stable [172]. The conversion from RGB to HSV is performed by the following equations [173]:

$$V = max(R, G, B) \tag{3.1}$$

$$S = \begin{cases} \frac{V - min(R,G,B)}{V}, if\ V \neq 0 \\ 0, if\ V = 0 \end{cases} \tag{3.2}$$

$$H = \begin{cases} \frac{60(G-B)}{S}, if\ V = R \\ 120 + \frac{60(B-R)}{S}, if\ V = G \\ 240 + \frac{60(R-G)}{S}, if\ V = B \end{cases} \tag{3.3}$$

The HSV technique was used for object detection in [174]. First, the camera image is converted to the HSV image format. The converting to the HSV format also produces better results for creating binary image and makes the computing faster. The HSV image is then converted to binary image for proper recognition of the object shape. The binary image consists of only 2 colors. One color is for the object and the other is for everything else. The next step is the smoothening of the binary image to avoid rough boundary and removing discontinuous changes in pixel values. This is achieved by applying a Gaussian Filter to the binary image. Fig. 3.5 illustrates the steps of object detection using HSV color space.

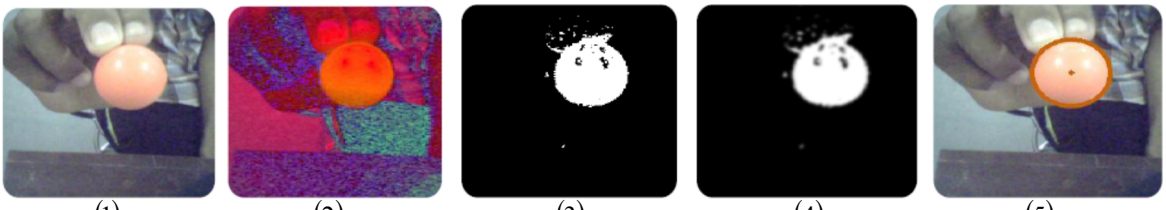


Figure 3.5: Object detection steps using HSV color space [174].

YCbCr-based Feature Extraction: YCbCr color space is also used for object detection. According to the color change of the light in the prevailing circumstances, it is important to instantly adjust the values of Y, Cb and Cr, respectively [25]. The conversion formula is as follows:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.2549 & 0.5059 & 0.0980 \\ -0.1451 & -0.2902 & 0.4392 \\ 0.4392 & -0.3647 & -0.0706 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} \quad (3.4)$$

According to the previous studies and implemented approaches, HSV can be considered the most powerful system for color segmentation because it is insusceptible to illumination. Different colors can be used to distinguish multiple objects in the view. Other source information like shape and size can be used additionally in case of existence undesired objects in the view which have the same color.

3.3.1.2 Shape Detection

By modeling an object as a set of shape primitives, such as circles, triangles, and rectangles, a set of rules can be used to recognize a set of desired objects [175]. The classical Hough transform was concerned for the identification of lines in the image, but later the Hough transform has been extended to identify arbitrary shapes, most commonly circles or ellipses [107]. The detection, which depends on multiple information such as color, edges, size, and shape, at the same time, can be a good strategy to perform a robust segmentation for the desired object. The desired object can be attached with a specific color with a particular shape to be detected and manipulated by the robotic arm.

3.3.1.3 Edges and Corners Detection

In order to obtain the required information from the visual sensors, it is necessary to apply some techniques like edge or corner detection to the captured image. The edge detection can be used for image segmentation where some filters can be used to extract the edges such as Canny, Sobel, and Laplace filters. The coordinates of these contours can be extracted in order to be sent to the robotic arm [176]. For corner detection, one of the most popular methods is the Harris Algorithm [94]. Shi et al. used the Harris method to find good corners for object tracking [177]. Tsay et al., presented an approach for accurately detecting the corners and centroid of the desired object [28]. The detected corners and centroid serve as the target point-feature set in visually guided robotic manipulation.

3.3.1.4 Object Recognition Based on Local Features

In some researches, object recognition approaches are divided into two categories. The first approach utilizes appearance features of objects such as the color and intensity. The second approach utilizes features extracted from the object itself to be matched with the features in the database related to the object of interest [178]. SIFT and SURF consider the most common methods which use local features extraction to perform the object recognition. Since, in the labs of life science, the H20 robot deals with different kinds of labware, an effective method is required to recognize them. SURF considers the most suitable algorithm for this task. The speeded-up robust features (SURF) algorithm has been demonstrated as an efficient object recognition method with a fast scale- and rotation-invariant detector and descriptor. The object tracker increases the speed by providing the complete region in the image that is occupied by the object at every time instant. The object region is jointly estimated by iteratively updating the object location and region information obtained from the previous frame. After finishing the process of object recognition based on the SURF algorithm, the ROI which contains the object is obtained using the object position. In the successive image, only the interest points in this updated ROI are extracted and matched to find the object. Different images or marks can be attached to the desired object or labware to be recognized on the workstation. SURF can be considered a robust method to perform the identification task for multiple labware on the workstation.

3.3.2 Object Pose Estimation

The pose of the target is very essential to guide the robotic arm. This can be found using a very reliable method with robust visual sensor. Some previous researches were depending on a stereo vision system with high resolution cameras to perform a good pose estimation. But the stereo imaging need a lot of prerequisites and processing steps to find the depth information which is the basis of position estimation. To handle this issue, a 3D camera Microsoft Kinect sensor is used. The Kinect can provide the depth information without the deep process and complicated steps that the stereo vision needs. It consists of RGB camera, depth sensor, and microphone. The Kinect becomes a very important sensor in the robotic applications especially for the tasks of object manipulation. Therefore, the localization process of multiple labware and placing holders can be performed using the Kinect. There are two versions of Kinect and it is very important to know about the specifications and the physical capabilities for each of them. Related to the old version (Kinect V1), the structured light with triangulation method is used to obtain the depth information. The field of view of RGB and depth camera are 57 degrees horizontally, 43 degrees vertically with tilt range (up and down) -27 to +27 degrees as shown in figure 3.6. The range of distance in default mode is 0.8m to 4.0m physically and from 1.2m

to 3.5m practically. On the other hand, the range of distance in near mode is 0.4m to 3.0m physically and from 0.8m to 2.5m practically. The modes for depth sensing (default or near) are selected and activated through the programming of Kinect and as appropriate for the scenario. Fig. 3.7 shows the two modes of depth sensing. There are some limitations which have to be taken in consideration during the work with Kinect sensor such as depth accuracy, the shadow in depth image, the strong lighting, the interference of other IR signals, etc. All these limitations influence the response of the Kinect sensor.

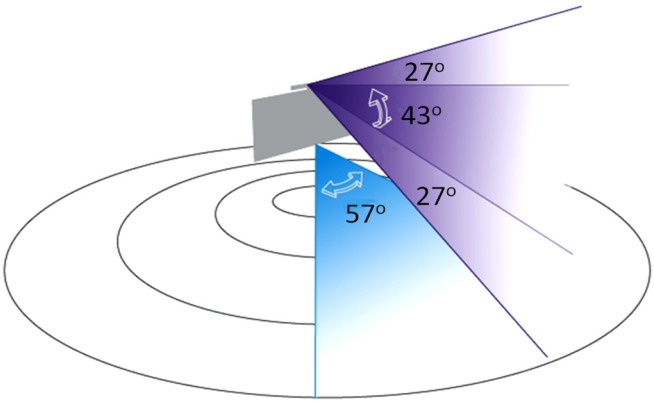


Figure 3.6: The field of view for Kinect camera V1 [179].

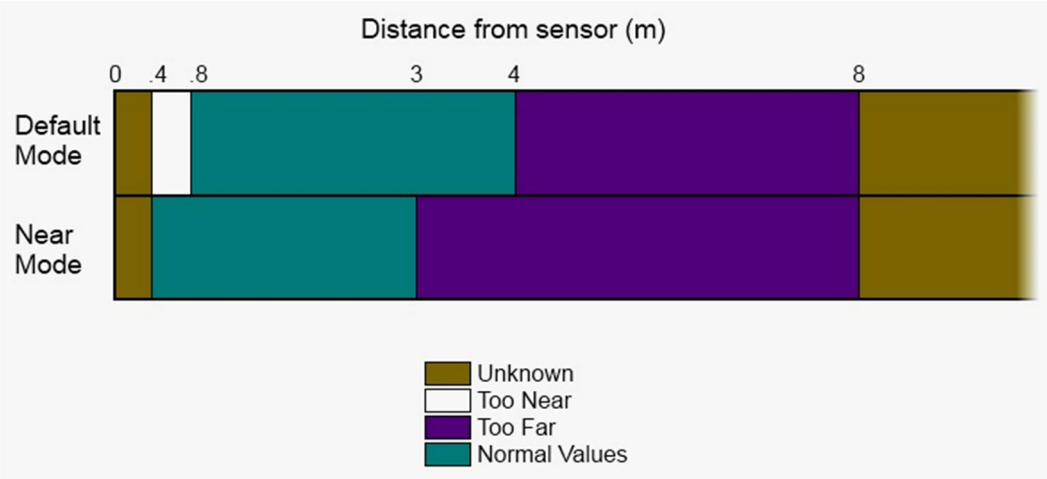


Figure 3.7: The depth range of kinect sensor V1 [180].

Related to the new version (Kinect V2), "time of flight" technology is used to provide the depth information and to determine the features and motion of certain objects. The face recognition, motion tracking, and resolution with Kinect V2 are much more precise than the Kinect V1. The high resolution of the RGB and depth cameras make it very desired to be used for object recognition and localization. The RGB camera of Kinect V2 captures color frames with a resolution of 1920X1080 pixels, whereas the IR camera, which is used for depth frame

acquisition, has 512X424 pixels resolution. In comparison with the Kinect V1, which has 640X480 color frame resolution and 320X240 depth frame resolution, it can be noticed that the depth measurements in V2 are more accurate than V1. Also, the computed point clouds show a better resolution. Related to the camera field of view, the Kinect V2 has a wider horizontal and vertical view in comparison with the Kinect V1. Therefore, the Kinect V2 is the most preferable visual sensor that can be used to perform the identification and localization of multiple objects for robotic manipulation. Table 3.1 shows the difference between Kinect V1 and V2.

Table 3.1: The difference between Kinect V1 and V2.

Features	Kinect for windows V1	Kinect for windows V2
RGB camera	640 X 480 @30 FPS	1920 X 1080 @30 FPS
Depth camera	320 X 240	512 X 424
Max depth distance	~ 4.5 M	~ 4.5 M
Min depth distance	40 cm in near mode	50 cm
Horizontal FOV	57 degrees	70 degrees
Vertical FOV	43 degrees	60 degrees
Tilt motor	Yes	No
Skelton joints defined	20 joints	26 joints
Full skeletons tracked	2	6
USB standard	2.0	3.0
Supported OS	Win 7, Win 8	Win 8, Win 10

3.4 The Manipulation with Unstable Arms

In case of dealing with unstable arms which have weak joints, it is very essential to use a close-loop feedback control to guarantee the reaching of the robotic arm to the target accurately. One of the reliable strategy is the visual servoing which can be used for tracking the object or the end effector with the object at the same time. In order to track an object in the workspace with visual sensors, continuous or discontinuous images have to be captured. Continuous images (video) provide more information about the workspace and this requires significantly more computation power which is undesirable. Discontinuous images provide less information but can be more difficult to track an object especially if both the object and the robotic manipulator are moving. The basic idea of tracking an object depends on identifying the same features of an object from different images taken at different times and viewpoints. There are two major systems for visual servoing, eye-in-hand and eye-to-hand. Fig. 3.8 shows a structure of robotic arm with eye-in-hand.

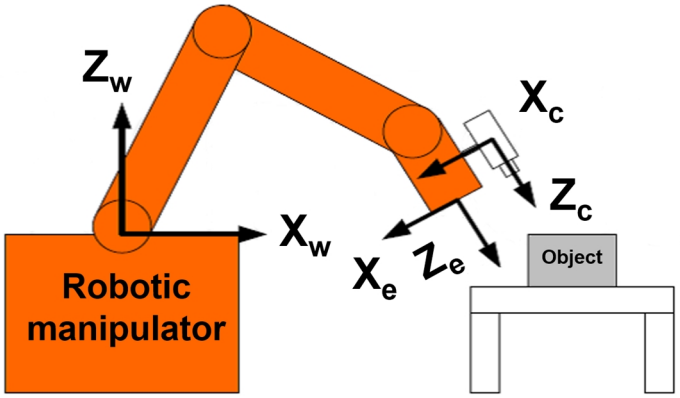


Figure 3.8: Robotic arm with Eye-in-Hand.

Monocular (one camera) systems use a camera either as a global sensor or as an Eye-in-Hand configuration. These systems usually adopt some form of model based visual techniques to facilitate the estimation of the depth between the camera and the object [181]. If the Eye-to-Hand configuration is used, a geometric model of the object is commonly used to retrieve the full pose of the object. On the other hand, in the Eye-in-Hand configuration, feature-based tracking techniques are widely used [182]. POSIT algorithm is one of the familiar approaches which is used to find the pose of the known object related to a single camera [126][127]. A single camera minimizes the processing time required to extract the visual information. However, in the case where the object model is unknown, the loss of depth information limits the servoing operations with one camera and complicates the control design. 3D camera or two cameras in a stereo configuration can be used to provide complete 3D information about the scene and the object [183]. One of the common approaches with stereo vision is to estimate the disparity which is then used for depth estimation [184]. The fundamental problem of disparity estimation is to match the corresponding features between two or more images. Since two different imaging devices are used, the stereo vision systems require twice as much computational effort in image processing as the monocular systems. In the look-and-move strategy, a vision based controller generates the inputs of the joint controller of robot. Therefore, the stability of the system relies on both robot joint controller and the vision-based controller.

Sanchez-Lopez et al. presented an example of PBVS approach with Eye-to-Hand system depending on object and end effector detection using color information [11]. It is important to compute the position of end effector in every moment and compare it with the target position to calculate the transformation matrix between both planes [185]. This kind of system assumes that the position of the end effector has to be in the field of view of the visual sensor. Also, it is assumed that the end effector and the visual sensor are in perpendicular planes to compute

the orientation easily. HSV color system has been used for object segmentation. The end effector has a specific color that is different from other objects in the scene. As shown in Fig. 3.9, the end effector color is orange but this color can be changed and it is important that it has enough linear separability from the colors of objects. After the segmentation process, the center of mass for every color (the end effector and the objects) has to be calculated. This is important to create a region of interest for visual servoing purposes where it will be used for computing the error between the desired position and the gripper position. Since the H20 robot has unstable arms with weak joints, one of the approaches eye-in-hand or eye-to-hand can be applied with the arms to improve the accuracy of reaching the target.

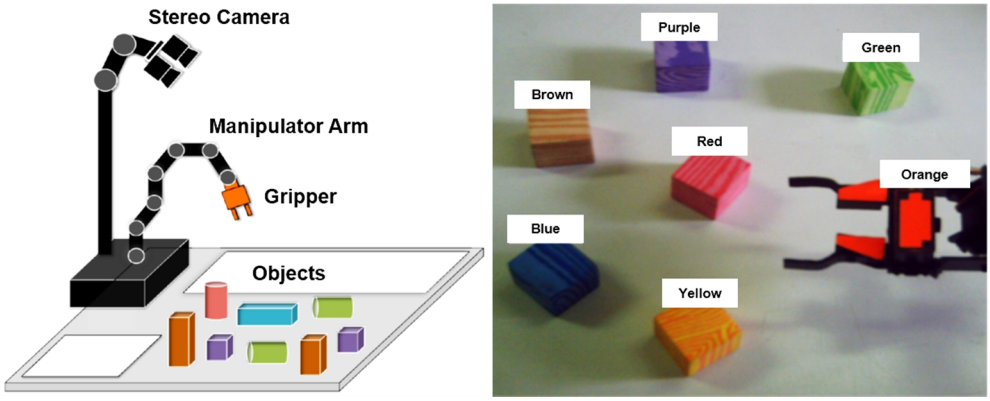


Figure 3.9: Object and end effector detection using color information [11].

3.5 Client-Server Communication

For the purpose of system integration, the control software for each device or platform has to be connected and communicated with the other parts. Sometimes, the control platforms are developed using different programming languages and it is difficult to integrate them in a single program. Also, the integration of multiple coding platforms, which are developed using similar programing language, into a single one can be complicated. This is due to the different functionalities and huge number of coding lines that are generated which make the possibility of solving the coding bugs very complex. Thus, for a flexible process execution, it is required to develop a software interface that can interact with all coding platforms simultaneously. As all the robot devices and computer are connected under a common LAN with different IP addresses, the interface program can interact with them over Ethernet by connecting to the specific IP address. The client initiates the connection with the server by connecting to a specific socket address using TCP/IP. The socket address is a combination of IP address and a port number. The communication process between the client and the server is described using a sequential diagram shown in Fig. 3.10 and is explained as follows:

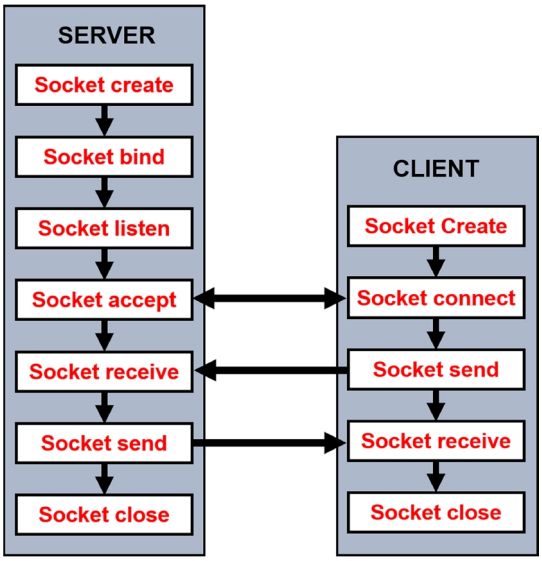


Figure 3.10: Sequential diagram for client – server model.

Initially, sockets are created on both server and client (by default, a server socket is created with the execution of server program). Client requests a socket connection with server on specific port. If the requested port is free to use, server establishes a connection and it is ready to communicate with client. Once, the connection is established, the client program send/receive information to/from the server program. Both sockets are closed when data transfer is successful. The information is transferred in the form of strings.

The arm manipulation system can be separated into two coding platforms. The first one is the object detection and localization process using the visual sensor. The second platform is the arm controller which include the kinematic model with joints control. The two platforms can be connected through a client-server model to exchange the required information during the manipulation task. The arm manipulation system in turn can be connected with the higher platform that can be the navigation platform or the management system to realize a fully integration system.

4 The Solution of Kinematic Problem

The kinematics analysis plays an active role in object manipulation because it is an important issue to make the arm end effector reach the desired object accurately. Also, there are other issues, which have to be taken into the consideration when we need to control the robotic arm, like, singularities, joint limits and reachable workspace. As mentioned in chapter 3, the closed form solution will be used for the H20 arms and this can only be found for specific types of robotic arms, which have particular structure with 6-DOF or less. Peiper indicates that in case there are 3 consecutive joints axes which are parallel to each other or intersecting at a single point then the closed-form solution can be existent [160]. The closed-form solution for the H20 arms can be found because the three shoulder joint axes intersect at a single point as shown in Fig. 4.1.

4.1 H20 Arms' Kinematics and Specifications

The H20 mobile robot has dual arms each consists of 6 revolute joints (2-shoulder joints, upper arm, elbow joint, lower arm, wrist) and a 2-DOF gripper, reaching 60 cm (2ft), with maximum lifting weight of 800g. The arm has 6-DOF, two of them are located at the shoulder, two in the elbow, and the last two DOF are assigned on the wrist. The length of the upper arm d3 and lower arm d5 are 236 mm and 232 mm respectively. Also, the distance between the wrist joint and the end effector is 69 mm. Fig. 4.1 shows the structure and the joints coordinate frames for both left and right arms.

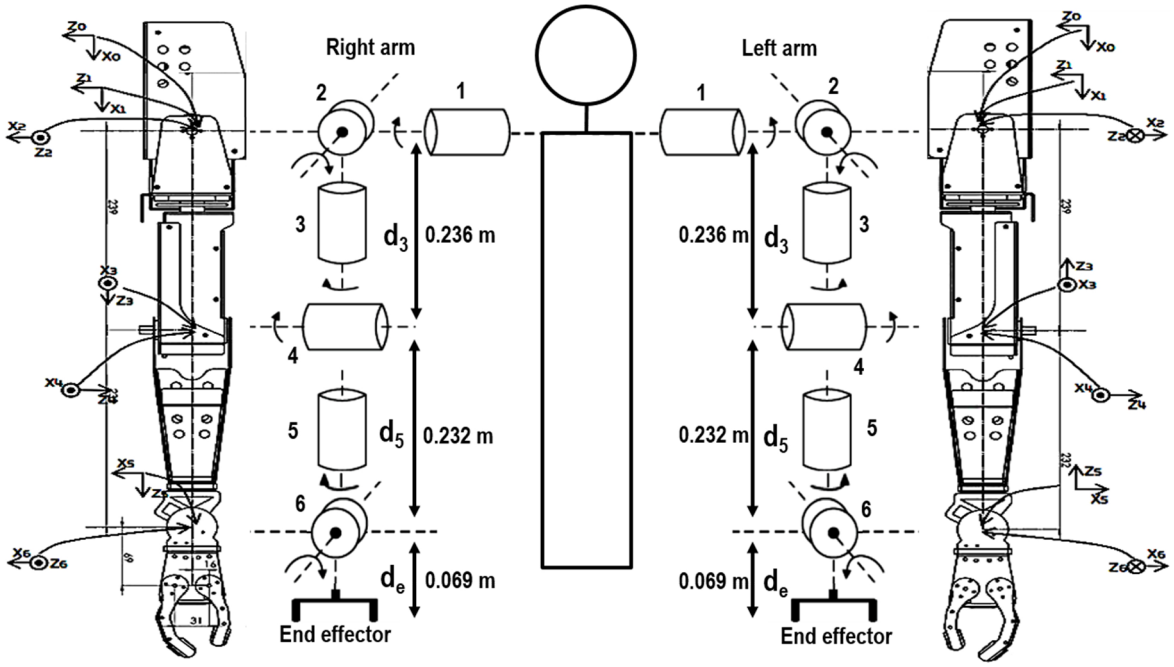


Figure 4.1: H20 arms structure and coordinate frames.

The Denavit-Hartenberg rules provide a guide for locating coordinate systems on each link of a multi-link kinematic chain. By following the D-H rules, the homogeneous transformations between adjacent links are defined. There are four parameters used in the manipulator analysis: the link length (a_{i-1}), the link twist (α_{i-1}), the link offset (d_i), and the joint angle (θ_i) where (i) refers to the link number [52]. The D-H parameters of the H20 arms are shown in Table 4.1 where joint (1) represents the shoulder joint, and joint (6) represents the wrist, and so on [186].

Table 4.1: The D-H parameters for the H20 arms.

Left and Right Arms						
θ_i	$\alpha_{(i-1)}$ (L)	$\alpha_{(i-1)}$ (R)	$a_{(i-1)}$ (LR)	d_i (m) (L)	d_i (m) (R)	Joints limit (LR)
θ_1	0°	0°	0	0	0	-20°~192°
θ_2	90°	-90°	0	0	0	-200°~-85°
θ_3	90°	-90°	0	-0.236	0.236	-195°~15°
θ_4	-90°	90°	0	0	0	-129°~90°
θ_5	90°	-90°	0	-0.232	0.232	0°~180°
θ_6	-90°	90°	0	0	0	-60°~85°

According to the D-H coordinates system, the general homogeneous transformation matrix of the H20 arms T_i (i=1~6) is as the following equation [186]:

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{(i-1)} \\ \sin\theta_i \cos\alpha_{(i-1)} & \cos\theta_i \cos\alpha_{(i-1)} & -\sin\alpha_{(i-1)} & -\sin\alpha_{(i-1)} d_i \\ \sin\theta_i \sin\alpha_{(i-1)} & \cos\theta_i \sin\alpha_{(i-1)} & \cos\alpha_{(i-1)} & \cos\alpha_{(i-1)} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1) $$

Due to Denavit-Hartenberg method, we can get the homogeneous transformation matrices of the H20 arms. The transformation matrices related to the right arm are as the following:

$$ {}^{0}_{1}T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad {}^{1}_{2}T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^{2}_{3}T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0.236 \\ -\sin\theta_3 & -\cos\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{3}_{4}T = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^{4}_{5}T = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0.232 \\ -\sin\theta_5 & -\cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{5}_{6}T = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

For the grasping task, it is crucial to find the matrix, which describes the translation between the wrist joint and the end effector. This translation is in Y-axis according to the coordinate's frame of a wrist joint and the matrix is as follows:

$$ {}^{6}_{E}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.069 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

4.2 Forward Kinematic Solution with Validation

The forward kinematics is how to find the end effector pose relative to the arm base for the given joint angles. The forward kinematic solution can be obtained using Eq. (4.2). The solution in the final matrix includes the orthogonal vector (n_x, n_y, n_z), the orientation vector (o_x, o_y, o_z), the approach vector (a_x, a_y, a_z), and the end effector position vector (p_x, p_y, p_z).

$$ {}^{0}_{E}T = {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T \cdot {}^{4}_{5}T \cdot {}^{5}_{6}T \cdot {}^{6}_{E}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.2} $$

Matlab software with Robotics toolbox has been used to validate the Forward Kinematic solution of the H20 arms. The numerical results of forward kinematic with simulation plot give clear insight for the kinematic behavior of the robotic arm. The validation process has been done after giving different joints configuration as an input to the forward Kinematic model (4.2) and to the Matlab toolbox. The results have been compared and plotted and they were identical.

The first configuration of joint angles [θ_1, θ_2, θ_3, θ_4, θ_5, θ_6] has been given as [0^o, -90^o, -90^o, 0^o, 90^o, 0^o]. The position and orientation of the end effector related to the arm base have been computed using (4.2). In the Matlab environment, the D-H parameters have been given. Then, using the command 'fkine' with Robotics toolbox, the same result has been attained. Fig. 4.2 illustrates the position and orientation of the end effector relative to the arm base and also the simulation plot of the arm.

$$ {}^{0}_{E}T = \begin{bmatrix} 0 & 1 & 0 & 0.537 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

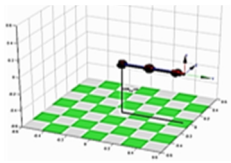


Figure 4.2: The position and orientation of the end effector and the simulation plot of arm according to the configuration [0^o, -90^o, -90^o, 0o, 90^o, 0^o].

Giving another joint configuration [θ_1, θ_2, θ_3, θ_4, θ_5, θ_6] as [0^o, -90^o, -90^o, -90^o, 90^o, 0^o], the position and orientation of the end effector relative to the arm base have been computed using (4.2) as well as Robotics toolbox. Fig. 4.3 illustrates the results according to the given joint configuration. Fig. 4.4 shows the results when the joints configuration is [θ_1, θ_2, θ_3, θ_4, θ_5, θ_6] = [0^o, -180^o, -90^o, 0^o, 90^o, 0^o].

$$ {}^{0}_{E}T = \begin{bmatrix} 0 & 0 & -1 & 0.236 \\ 0 & 1 & 0 & 0.301 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

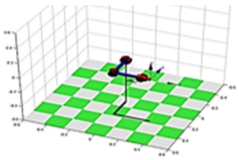


Figure 4.3: The position and orientation of the end effector and the simulation plot of arm according to the configuration [0^o, -90^o, -90^o, -90^o, 90^o, 0^o].

$$ {}^{0}_{E}T = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.301 \\ 0 & 0 & -1 & 0.236 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

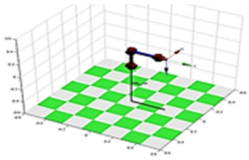


Figure 4.4: The position and orientation of the end effector and the simulation plot of arm according to the configuration [0^o, -180^o, -90^o, -90^o, 90^o, 0^o].

4.3 Inverse Kinematic Solution

M. A. Ali et al. proposed a reverse decoupling mechanism method to solve the IK problem of humanoid robots analytically [187]. The strategy of this method depends on viewing the kinematic chain of the manipulator in reverse order with decoupling the position and

orientation. In other words, the arm can be viewed in reverse order so that the pose of the arm base can be described relative to the end effector. This method includes also decision equations to choose the suitable solution within multiple solutions. R. O'Flaherty et al. utilized the same method to find the closed-form solution of the IK problem for HUBO2+ humanoid robot [188].

A reverse decoupling mechanism method has been used to find the full solution analytically for 6-joints by viewing the kinematic chain of H20 arms in reverse order and then decoupling it into the positioning and orientation mechanisms. Finally, the inverse transform technique is utilized in deriving a consistent joint solution. Using this knowledge, the arm can be viewed in reverse so that the last three joints make up the shoulder, thus the position and orientation of the shoulder frame can be described relative to the hand frame. This method gives 8 possible solutions for each specified set of end effector position and orientation. This approach has been applied to the left and right arms of H20 robot. Computer simulations using Matlab were performed to verify the correctness of the closed-form solutions. To solve the IK problem in this reverse way, the following steps have been performed [186]:

1- Finding the forward kinematic matrix as the following equation:

$$ {}^{0}_{E}T = {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T \cdot {}^{4}_{5}T \cdot {}^{5}_{6}T \cdot {}^{6}_{E}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

2- Find the inverse of the forward kinematic matrix (${}^{0}_{E}T'$) which equals to ${}^{E}_{0}T$.

3- Find the solution of joints 4, 5, and 6 using following equation:

$$ {}^{5}_{6}T \cdot {}^{6}_{E}T \cdot {}^{0}_{E}T' = {}^{5}_{4}T \cdot {}^{4}_{3}T \cdot {}^{3}_{2}T \cdot {}^{2}_{1}T \cdot {}^{1}_{0}T . $$

4- The joints solutions will be found by equating specific elements in the final matrix of the left-hand side with the related elements in final matrix of the right-hand side. Then, the trigonometric identities with some mathematical operations will be used to find the desired solution.

5- Find the solution of joints 1, 2, and 3 using following equation and as the previous step:

$$ {}^{3}_{4}T . {}^{4}_{5}T . {}^{5}_{6}T . {}^{6}_{E}T . {}^{0}_{E}T' = {}^{3}_{2}T . {}^{2}_{1}T . {}^{1}_{0}T $$

To start solving the inverse kinematic problem of the left and right arms, the inverse of forward kinematic matrix has to be found as follows:

$$ {}^{E}_{0}T = {}^{E}_{6}T \cdot {}^{6}_{5}T \cdot {}^{5}_{4}T \cdot {}^{4}_{3}T \cdot {}^{3}_{2}T \cdot {}^{2}_{1}T \cdot {}^{1}_{0}T = \begin{bmatrix} in_x & io_x & ia_x & ip_x \\ in_y & io_y & ia_y & ip_y \\ in_z & io_z & ia_z & ip_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.3) $$

To find the solution for the joints 4, 5, and 6, we multiply the both sides of (4.3) by $({}^{5}_{6}T \cdot {}^{6}_{E}T)$ and the result is as follows:

$$ {}^{5}_{4}T \cdot {}^{4}_{3}T \cdot {}^{3}_{2}T \cdot {}^{2}_{1}T \cdot {}^{1}_{0}T = {}^{5}_{6}T \cdot {}^{6}_{E}T \cdot {}^{E}_{0}T. \quad (4.4) $$

The right side of (4.4) is the following matrix:

$$ \begin{bmatrix} r_1 n_x & r_1 o_x & r_1 a_x & -S_6(ip_y + d_e) + C_6 ip_x \\ r_1 n_y & r_1 o_y & r_1 a_y & ip_z \\ r_1 n_z & r_1 o_z & r_1 a_z & -C_6(ip_y + d_e) - S_6 ip_x \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.5) $$

and the left side of (4.4) is:

$$ \begin{bmatrix} l_1 n_x & l_1 o_x & l_1 a_x & -d_3 S_4 C_5 \\ l_1 n_y & l_1 o_y & l_1 a_y & d_3 S_4 S_5 \\ l_1 n_z & l_1 o_z & l_1 a_z & d_3 C_4 + d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.6) $$

where S and C are the abbreviation of sine and cosine of angle respectively. By equating the position elements of (4.5) and (4.6), we get:

$$ -S_6(ip_y + d_e) + C_6 ip_x = -d_3 S_4 C_5, \quad (4.7) $$

$$ ip_z = d_3 S_4 S_5, \quad (4.8) $$

$$ -C_6(ip_y + d_e) - S_6 ip_x = d_3 C_4 + d_5. \quad (4.9) $$

To solve the joint angle θ4, we suppose the following terms: $(ip_y + d_e = rS_\gamma)$ and $(ip_x = rC_\gamma)$, where $r = (\sqrt{(ip_y + d_e)^2 + (ip_x)^2})$, $\gamma = (\text{atan2}\ (ip_y + d_e, ip_x))$, and (atan2) is the two argument arc tangent function. These terms are obtained according to the arms coordinates frame with reverse order. By substituting these terms into (4.7) and (4.8) and using the angle sum identities we get the following equations:

$$rC_{6+\gamma} = -d_3 S_4 C_5 , \tag{4.10}$$

$$rS_{6+\gamma} = d_3 C_4 + d_5 . \tag{4.11}$$

The equation of C_4 for the left and right arms can be obtained by squaring (4.8), (4.10), and (4.11) and adding them. The solution is as follows:

$$C_{4(LR)} = \frac{(ip_y + d_e)^2 + (ip_x)^2 + ip_z{}^2 - d_3{}^2 - d_5{}^2}{2 d_3 d_5} . \tag{4.12}$$

Thus, the solution of θ_4 for the left and right arms can be found.

$$\theta_{4(LR)} = \text{atan2}\left(\pm \, real \left(\sqrt{1 - C_4^2} \right), C_4 \right) . \tag{4.13}$$

The arc tangent function (atan2(sinθ, cosθ)) is more consistent to be used for finding the angle value (θ) than arc cosine and arc sine. This is because of the inaccurate behavior to determine the required angle in case of using the arc sine and arc cosine functions. The complex numbers are generated in case the target position is not within the reachable workspace of the arm. Therefore, (*real*) function is used to ignore the imaginary parts of complex numbers and take only the real part in the joint solutions. Thus, the solution that is closest to the target position can be obtained [188]. After finding the value of θ_4, the equation of S_5 can be solved using (4.8) to get the solution of θ_5 as follows:

$$S_{5(L)} = \frac{ip_z}{S_4 d_3}, \; S_{5(R)} = \frac{-ip_z}{S_4 d_3} , \tag{4.14}$$

$$\theta_{5(LR)} = \text{atan2}\left(S_5 , \pm \, real \left(\sqrt{1 - S_5^2} \right) \right) . \tag{4.15}$$

The solution of θ_6 can be obtained on dividing (4.11) by (4.10) to get the following equation:

$$\frac{S_{6+\gamma}}{C_{6+\gamma}} = \frac{d_3 C_4 + d_5}{-d_3 S_4 C_5} . \tag{4.16}$$

Thus, the value of θ_6 can be found as follows:

$$\theta_{6(LR)} = \text{wrapToPi}(\text{atan2}(d_3 C_4 + d_5 , -d_3 S_4 C_5) - \gamma), \tag{4.17}$$

where (wrapToPi) is a function to wrap the angle to the interval between $-\pi$ and π [188].

To find the solution for the joints 1, 2, and 3, the both sides of (4.3) are multiplied by (${}^{3}_{4}T\cdot{}^{4}_{5}T\cdot{}^{5}_{6}T\cdot{}^{6}_{E}T$) and the result is as follows:

$${}^{3}_{2}T\cdot{}^{2}_{1}T\cdot{}^{1}_{0}T={}^{3}_{4}T\cdot{}^{4}_{5}T\cdot{}^{5}_{6}T\cdot{}^{6}_{E}T\cdot{}^{E}_{0}T. \tag{4.18}$$

The left side of (4.18) is the following matrix:

$$\begin{bmatrix} S_1S_3+C_1C_2C_3 & C_2C_3S_1-C_1S_3 & -S_2C_3 & 0 \\ C_3S_1-C_1C_2S_3 & -C_1C_3-C_2S_1S_3 & S_2S_3 & 0 \\ -C_1S_2 & -S_1S_2 & -C_2 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.19}$$

By taking the element (3,3) in the left and right sides of (4.18) and equating them, the equation of C_2 results, which is used to find θ_2 as follows:

$$\begin{aligned} C_{2(L)} &= ia_y(C_4C_6-C_5S_4S_6)+ia_x(C_4S_6+C_5C_6S_4)-ia_zS_4S_5, \\ C_{2(R)} &= -ia_y(C_4C_6-C_5S_4S_6)-ia_x(C_4S_6+C_5C_6S_4)-ia_zS_4S_5, \end{aligned}$$

$$\theta_{2(LR)}=\text{atan2}\left(\pm\ real\ \left(\sqrt{1-C_2^2}\right),C_2\right). \tag{4.20}$$

For the solution of θ_3, the elements (1,3) and (2,3) in the left and right sides of (4.18) are compared and divided by each other to get S_3 and C_3 which are used to find the solution of θ_3:

$$\begin{aligned} C_{3(L)} &= ia_y(C_6S_4+C_4C_5S_6)+ia_x(S_4S_6-C_4C_5C_6)+ia_zC_4S_5, \\ C_{3(R)} &= -ia_y(C_6S_4+C_4C_5S_6)-ia_x(S_4S_6-C_4C_5C_6)+ia_zC_4S_5, \\ S_{3(L)} &= -ia_yS_5S_6+ia_xC_6S_5+ia_zC_5,\quad S_{3(R)}=ia_yS_5S_6-ia_xC_6S_5+ia_zC_5, \\ \theta_{3(LR)} &= \text{atan2}(S_3,C_3). \end{aligned} \tag{4.21}$$

The last step is to find the solution of θ_1, which is obtained by comparing the elements (3,1) and (3,2) in the left and right sides of (4.18) and dividing them by each other to get S_1 and C_1 which are used to find the solution of θ_1 as the following:

$$\begin{aligned} C_{1(L)} &= in_y(C_4C_6-C_5S_4S_6)+in_x(C_4S_6+C_5C_6S_4)-in_zS_4S_5, \\ C_{1(R)} &= in_y(C_4C_6-C_5S_4S_6)+in_x(C_4S_6+C_5C_6S_4)+in_zS_4S_5, \\ S_{1(L)} &= io_y(C_4C_6-C_5S_4S_6)+io_x(C_4S_6+C_5C_6S_4)-io_zS_4S_5, \end{aligned}$$

$$S_{1(R)} = io_y(C_4C_6 - C_5S_4S_6) + io_x(C_4S_6 + C_5C_6S_4) + io_zS_4S_5,$$
$$\theta_{1(LR)} = \text{atan2}(S_1, C_1). \quad (4.22)$$

According to the previous joints' solutions, it can be noticed that θ_2, θ_4, and θ_5 have two solutions. This causes 8 total solutions for the inverse kinematics problem of the H20 arms. The strategy about how to choose the optimal solution will be discussed in the next sections.

To find the all possible solutions of IK problem, the solution of the singularity cases has to be found. Singularities are arm configurations in which one or more degrees of freedom are eliminated when some joints' axes align with each other. Thus, the number of solutions for the IK problem will be infinite. Three cases of singularity have been determined within the joints limits. The inverse kinematic solution for every case of singularity is as follows:

1) When $\theta_4 = 0$ and $\theta_2 \neq -\pi$: The axis of the third joint is aligned with the fifth joint axis as shown in Fig. 4.5.

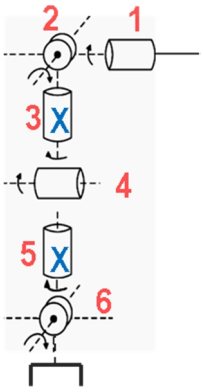


Figure 4.5: Singularity case where the 3rd and 5th joints are aligned.

To find the IK solution for this case, we keep the previous value of θ_3 and define $\theta_T = \theta_3 + \theta_5$. Firstly, we make $\theta_4 = 0$, then we start with the solution of θ_6 by using the elements (1,4) and (4,2) of (4.3) and dividing them by each other to get S_6 and C_6. The solution of θ_6 for the left and right arms are found as follows:

$$\frac{-ip_x}{-ip_y - d_e} = \frac{S_{6(LR)}}{C_{6(LR)}},$$

$$\theta_{6(LR)} = \text{atan2}(S_6, C_6). \quad (4.23)$$

For the solution of θ_1, we compare the elements (3,1) and (3,2) in the left and right sides of (4.4).

$$\theta_{1(LR)} = \text{atan2}\,(S_6 io_x + C_6 io_y, S_6 in_x + C_6 in_y)\,. \tag{4.24}$$

The elements (3,1) and (3,3) in the left and right sides of (4.4) are compared to get the solution of θ_2 which is as follows:

$$\theta_{2(L)} = \text{atan2}\,((-S_6 in_x - C_6 in_y)/C_1, C_6 ia_y + S_6 ia_x)\,, \tag{4.25}$$
$$\theta_{2(R)} = \text{atan2}\,((-S_6 in_x - C_6 in_y)/C_1, -C_6 ia_y - S_6 ia_x)\,.$$

And if $C_1 < 0$, then $\theta_2 = \text{wrapToPi}\,(\theta_2 + \pi)$.

The solution of θ_5 can be found by firstly solving θ_T. This is done by using the elements (1,3) and (2,3) in the left and right sides of (4.4) with the angle sum identities to get S_{3+5} and C_{3+5}. Thus, the solution of θ_T is as follows:

$$S_{(3+5)(L)} = -ia_z\,,\ S_{(3+5)(R)} = ia_z\,,\ C_{(3+5)(LR)} = ia_x C_6 - ia_y S_6\,,$$

$$\theta_{T(LR)} = \text{atan2}\left(S_{(3+5)}, C_{(3+5)}\right). \tag{4.26}$$

If $S_{2(L)} < 0$, then $\theta_{T(L)} = \text{wrapToPi}\,(\theta_{T(L)} + \pi)$. And if $S_{2(R)} > 0$, then $\theta_{T(R)} = \text{wrapToPi}\,(\theta_{T(R)} + \pi)$.

Finally, we can get the solution of θ_5 as follows:

$$\theta_{5(LR)} = \text{wrapToPi}\,(\theta_T - \theta_3). \tag{4.27}$$

2) When $\theta_2 = -\pi$ and $\theta_4 \neq 0$: The axis of the first joint is aligned with the third joint axis as shown in Fig. 4.6.

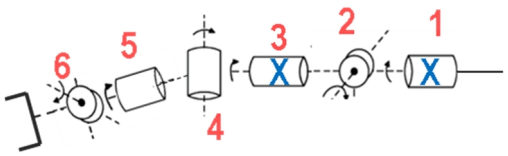


Figure 4.6: Singularity case where the 1st and 3rd joints are aligned.

The solutions of θ_4, θ_5, and θ_6 are found using the same equations of 4.13, 4.15, and 4.17. Also, we keep the previous value of θ_1 and define $\theta_T = \theta_1 + \theta_3$. To find the solution of θ_5, firstly we make $\theta_2 = -\pi$. Then, we start with the solution of θ_T by using the elements (3,1) and (3,2) in the left and right sides of (4.4) with the angle sum identities to obtain S_{1+3} and C_{1+3}. The solution is as follows:

$$S_{(1+3)(L)} = -io_x S_6 - io_y C_6 , \; S_{(1+3)(R)} = io_x S_6 + io_y C_6 ,$$
$$C_{(1+3)(L)} = -in_x S_6 - in_y C_6 , \; C_{(1+3)(R)} = in_x S_6 + in_y C_6 ,$$
$$\theta_{T(LR)} = \text{atan2}\left(S_{(1+3)}, C_{(1+3)}\right). \quad (4.28)$$

If $S_4 > 0$, then $\theta_{T(L)} = \text{wrapToPi}\,(\theta_{T(L)} + \pi)$.

If $S_4 < 0$, then $\theta_{T(R)} = \text{wrapToPi}\,(\theta_{T(R)} + \pi)$.

Finally, we can get the solution of θ_3 as follows:

$$\theta_{3(LR)} = \text{wrapToPi}\,(\theta_T - \theta_1). \quad (4.29)$$

3) When $\theta_4 = 0$ and $\theta_2 = -\pi$: The joints 1, 3, and 5 are collinear as shown in Fig. 4.7.

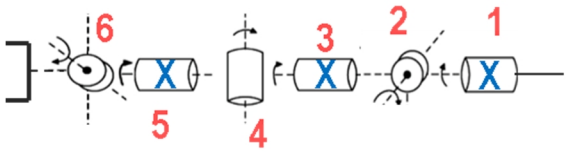


Figure 4.7: Singularity case where the 1st, 3rd, and 5th joints are aligned.

For this case, the solutions of θ_6 is found using the same equations in case 1 (when $\theta_4 = 0$ and $\theta_2 \neq -\pi$). Also, we keep the previous values of θ_1 and θ_3, then we define $\theta_T = \theta_1 + \theta_3 + \theta_5$. To find the solution of θ_5, firstly we make $\theta_2 = -\pi$ and $\theta_4 = 0$. Thus, we start with the solution of θ_T by using the elements (2,1) and (2,2) in the left and right sides of (4.4) with the angle sum identities to obtain S_{1+3+5} and C_{1+3+5} as follows:

$$S_{(1+3+5)(L)} = in_z , \; S_{(1+3+5)(R)} = -in_z , \; C_{(1+3+5)(L)} = -io_z , \; C_{(1+3+5)(R)} = io_z ,$$

$$\theta_{T(LR)} = \text{atan2}\left(S_{(1+3+5)}, C_{(1+3+5)}\right). \quad (4.30)$$

Finally, we can get the solution of θ_5 as follows:

$$\theta_{5(LR)} = \text{wrapToPi}\,(\theta_T - \theta_1 - \theta_3). \quad (4.31)$$

4.4 The Validation of Inverse Kinematic Solution

MATLAB software with Robotics toolbox has been used to validate our inverse kinematics solution [186]. The joints results with the simulation plot give a clear proof for the inverse

kinematic behavior of the robotic arms. The validation process has been done after giving random joints values as an input to the forward Kinematic model. The pose information of the end effector, which we get from the FK model, is inserted to the inverse kinematic model that includes the joints limits with the selecting algorithm. Then, the joints values that are inserted to the FK model are compared with the result of IK model as shown in Fig. 4.8.

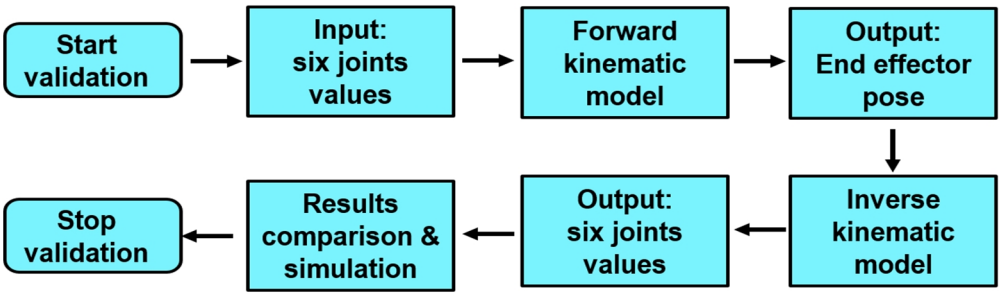
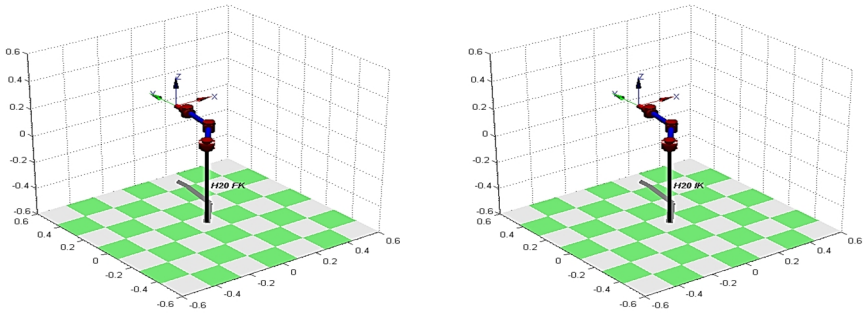


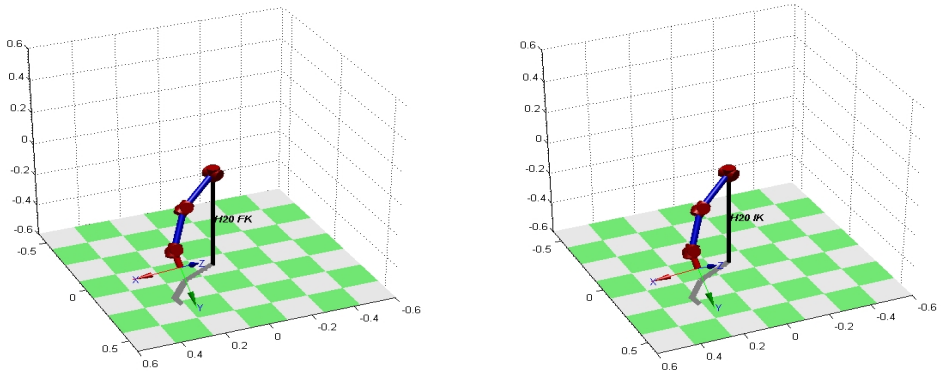
Figure 4.8: The validation process for the IK solution.

Different arm configurations have been used for validation processes. The results have been compared and plotted and the joints values which were inserted to the FK model are identical with the joints values which were obtained from the IK model as shown in the example of Fig. 4.9. In this example, Fig. 4.9.A shows the simulation plot of the arm with the position and orientation of the end effector according to the joint values inserted to the FK model. Fig. 4.9.B represents the result of the IK solution. The simulation plots of Fig. 4.10 and Fig. 4.11 show the validation result for different arm configurations [186].



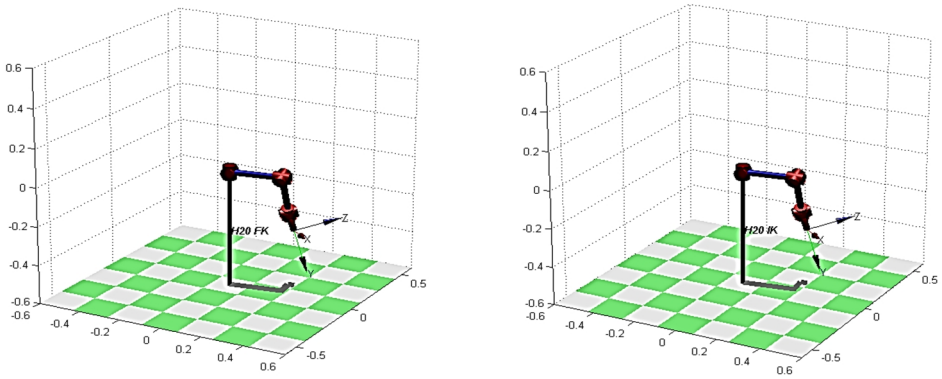
(*A*) (*B*)

Figure 4.9: The validation of IK model according to the configuration [50°, -90°, -90°, -30°, 180°, 10°].



(*A*) (*B*)

Figure 4.10: The validation of the IK model according to the configuration [30°, -120°, -45°, -23°, 180°, 30°].



(*A*) (*B*)

Figure 4.11: The validation of the IK model according to the configuration [9°, -90°, -23°, -90°, 126°, -30°].

4.5 Workspace

The workspace is the space which is swept out by the arm end effector after executing all possible motion. The workspace is one of the essential parameters for robotic arm performance in addition to its speed and accuracy. The calculation of the arm workspace is very important to decide whether the desired object, which has to be manipulated, is inside the reachable space or not. The workspace of the robotic arm can be determined according to the links length, joints type, and joints limit. The length of H20 arm is 0.537 m (d3 = 0.236 m, d5 = 0.232 m, de = 0.069 m) as shown in Fig. 4.1. Related to the joints, the H20 arm consists of 6 rotary joints and their limits' values have been mentioned in Table 4.1. MATLAB software with Robotics toolbox has been used to calculate the workspace of the H20 arm by inserting the links length and all the possible joint values within the joints limit to the FK model to find the position of the end effector for every sample. Finally, all possible positions which the robotic arm can reach will be found [186]. The simulation of workspace envelope for the H20 arm is shown in Fig. 4.12 [186].

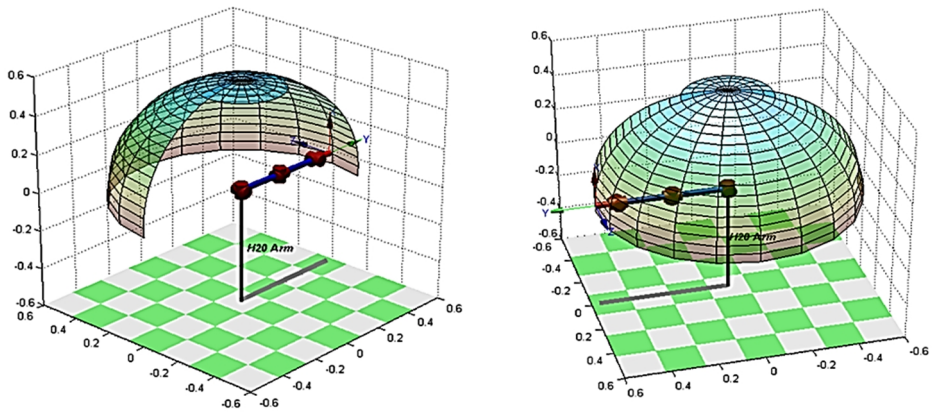


Figure 4.12: The workspace envelope of the H20 arms [186].

4.6 The Selecting of the desired IK Solution

For object manipulating task, the pose information of the object relative to the arm base determines whether this object is inside the workspace or not. In case that the object is outside the workspace, then there is no solution for the inverse kinematic problem. But, there will be 8 solutions for the required pose if it is inside the reachable workspace. To choose the suitable solution within the 8 solutions, the joints limits have to be taken into the consideration. If one joint value related to a specific solution is not within the joint limit, the solution will be ignored. In case that there are multiple solutions, where all the joints' values of every solution are within the joints' limits, a selecting algorithm has to be used. The solution with minimum joint motion is selected by finding the sum of squared joint values for the previous configuration and for every possible solution ($\theta_1^2 + \theta_2^2 + \theta_3^2 + \theta_4^2 + \theta_5^2 + \theta_6^2$) [188]. The next step is to compare the sum of previous configuration with the sum of every solution. The solution with closest sum value to the sum value of previous configuration is the desired solution [186]. For the work in the life science automation labs, the desired solution within multiple solutions can be estimated. This is because the environment is structured where the desired target is known and it has a fixed pose on the workbench. Thereby, the range of the joints' values for the desired solution can be estimated.

5 Blind Manipulation and Gripper Design

In this chapter, the procedures of using the H20 mobile robots for labware transportation blindly in a life science environment are presented. This strategy depends on using the sonar sensor with arm kinematics control to perform he grasping and placing tasks. Since the H20 robot can't manipulate the labware directly due to the arm structure and labware shape, a labware container is required. Different designs of labware containers are developed and presented in this chapter also. Specific arm gripper for each design of labware container is resolved to achieve the manipulation and transportation tasks. Fig. 5.1 shows the H20 mobile robot and how it grasps the labware container which is attached to a handle.

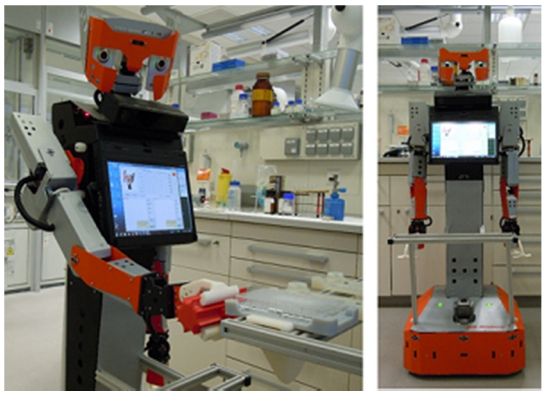


Figure 5.1: The H20 robot grasps the labware container.

5.1 The Related Problem

The H20 mobile robot transportation system has been developed by other research groups at the Center for Life Science Automation (celisca) by solving some key technical issues. These issues included the development of: 1) a remote control system based on wireless communication [158], 2) the path planning strategies [189][190], 3) low-cost robot localization and collision avoidance system [191][192], 4) an autonomous charging management [193]. For objects transportation, the mobile robots usually follow a predefined path to a specified station using a guidance control system as shown in Fig. 5.2. The H20 mobile robots use the stargazer sensor with ceiling landmarks for maneuvering between the adjacent labs (Hagisonic Company, Korea). This system inevitably causes orientation and position error of ±5cm in Z-axis and ±3cm in X-axis in front of the workstation. The error is related to two reasons. The first is the strong lighting and sunlight, which makes the stargazer unable to recognize the ceiling landmarks. The second reason is related to the odometry system, which includes encoders that are mounted on the robot wheels to provide the motion information that updates the robot pose (see Fig. 5.3.). The odometry system accumulates errors for different reasons such as different wheels' diameter, wheel-slippage, wheels misalignment and finite encoder

resolution. According to the experiments results and previous studies, the rotation of the robot is the greatest factor in odometry errors. Accordingly, the robot has to find the target position every time it arrives the workbench. The required accuracy for labware manipulation has to be less than 1 cm to guarantee safe tasks. Therefore, the more direct way of dealing with this problem is to use sensors to provide the position and orientation of the target. Then, the joints of the arm have to be configured using the kinematic model to place the arm end effector precisely and safely in the desired pose if it is inside the reachable space.

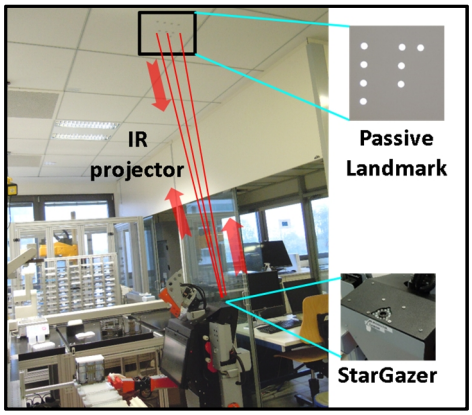


Figure 5.2: Stargazer sensor with cieling landmarks.

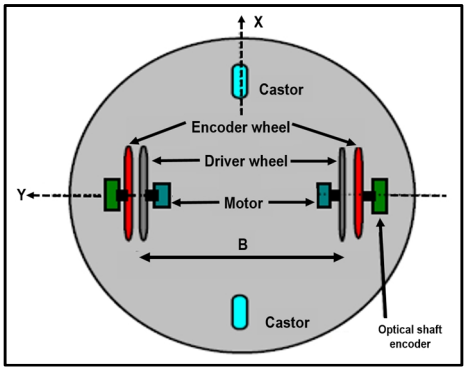


Figure 5.3: Odometry system design [194].

5.2 The Repeatability of H20 Arms

The checking of the accuracy and repeatability of the robotic arms is a basic and essentially step for the tasks of objects manipulation. The repeatability of the robotic arm means how precisely this arm can return to a taught position. The precision depends on some elements such as the resolution of the control system and the imprecision of the mechanical linkages and DC servo motors. To check the repeatability of the H20 arm, a fixed position laser sensor with grid paper on the end effector has been used. The laser sensor has the ability to measure the distance

with an accuracy of ±2 mm. Grasping configuration has been prepared randomly by giving particular values for the arm joints to make sure that the end effector reaches a specific position. The grasping movement has been repeated 40 times. The distance between the end effector and the sensor has been registered and also the position of the laser beam on the grid paper has been marked at the end of every movement. Fig. 5.4 shows the laser sensor and the incident laser beam on the end effector. According to the results that have been obtained from this experiment, the accuracy of the used arm to reach a specific position according to the related configuration is as the following: (X:±2mm, Y:±4mm, Z:±2mm) [195].

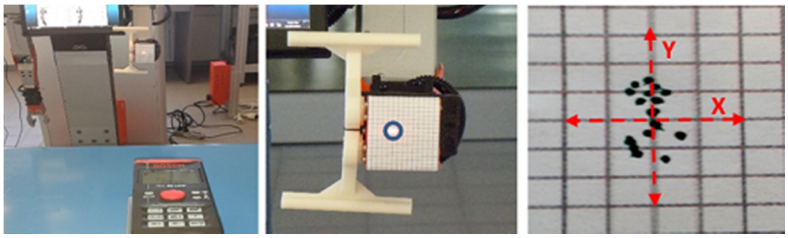


Figure 5.4: The laser sensor and its incident beam on the end effector.

Fig. 5.5 shows the Gaussian distribution for the positions of the end effector according to the X-axis. Fig. 5.6 and Fig. 5.7 show the related results according to the Y-axis and Z-axis respectively. The expected position of end effector is at (x=0 mm, y=0 mm, z=0 mm).

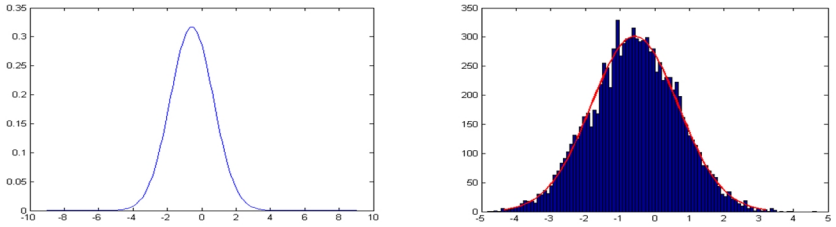


Figure 5.5: Gaussian distribution of X-values with mean = -0.5 mm and standard deviation = 1.2 mm.

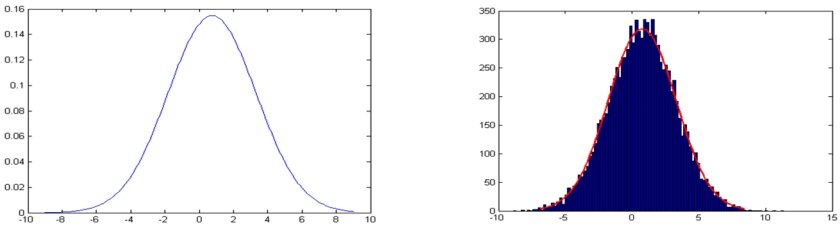


Figure 5.6: Gaussian distribution of Y-values with mean = 0.7 mm and standard deviation = 2.5 mm.

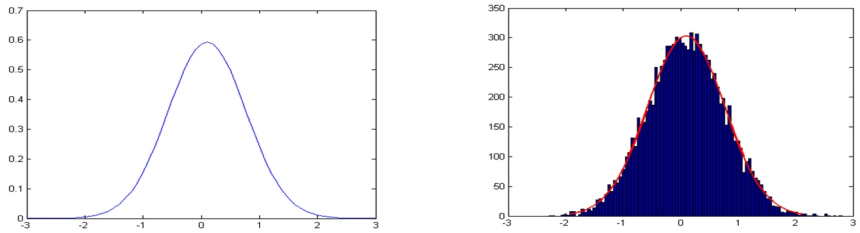


Figure 5.7: Gaussian distribution of Z-values with mean = 0.1 mm and standard deviation = 0.6 mm.

5.3 Blind Manipulation using lists of joints' values

A blind grasping and placing strategy has been performed based on the ultrasonic measurement to find the distance between the robot base and the workbench [196][197]. There are two sonar sensors built on the H20 base for the function of collision avoidance. These two sensors measure two channel of distances between the workstation and the robot base whenever the robot arrives the related predefined location. The values of two channels are used to correct the orientation of the robot in front of the workbench by equalizing them. Then, the measured distances is utilized to decide about the suitable values which have to be given to the arm joints to perform the manipulation task. There is no feedback information to correct the robot position in the left-right axis. XML files, which contain the required servo joints' values of arm control, have been previously prepared at different distances for the robot related to the workbench. Two factors determine the required number of XML files. The first factor is how many positions on the workbench the arm has to reach. The second is the error range of robot distance related to the target position on the workbench. The distance value of the robot and the target position are used to select the suitable XML file which guides the arm to the target. All the controlling parameters of arm joints are extracted from the XML file and loaded to the arm hardware servo modules. Sixteen joints' parameters are defined in the arm file where 8 parameters are related to each arm (6 arm joints in addition to 2 joints for gripper). Only one target position has been predefined for each arm in this work which requires to control just 3 joints [196][197]. Fig. 5.8 shows the values of servo motor values of the arm movement. It can be noticed that just the joints 1, 4, and 6 have been used where the joints 2, 3, and 5 have fixed values during the movement. This makes the movement of the robotic arm in 2D space. This teaching way of arm control is just suitable for stationary robots and only in case of working in a structured environment where all the parameters are fixed and previously known. It is not sufficient to be used with the mobile robots according to many drawbacks. The calculation of the required servo value for each joint with the teaching procedure needs a long time. The joints' values for each step in the arm movement have to be saved in a XML file consecutively. This procedure has to be performed for each arm and at different positions of

the robot related to the target position. Additionally, the grasping task has different consecutive joints' values in comparison with the placing task which leads to prepare particular files for each task. Furthermore, all the 6 joints of the arm have to be controlled to perform a 3D movement and this requires huge numbers of XML files. The files number will be doubled for every additional position of the target on the workbench which is not applicable to realize multiple labware manipulation.

J1	0	230	230	140	140	140	140	205	205	-160	-220	-320	-402	-390	-290	230
J2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J4	0	0	0	-150	-150	-150	-246	-340	-340	-340	-310	-310	-210	-195	-195	-340
J5	0	-870	-870	-870	-870	-870	-870	-870	-870	-870	-870	-870	-870	-870	-870	-870
J6	0	0	595	595	595	595	595	321	-55	-105	-185	-185	145	145	145	145

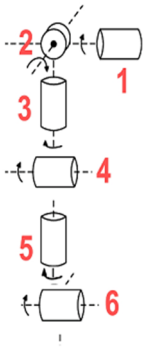


Figure 5.8: The servo joints values during the movement.

This blind strategy can achieve the manipulation task for only one position with an average accuracy of 2 cm. The required accuracy for object manipulation has to be less than 1 cm to guarantee safe tasks. Also, with this strategy, there is no flexibility. It needs to prepare movement plans for each arm and for each robot at different positions in front of the workstation. It is not possible to generalize the prepared files of one robot to be used for other H20 robots. The arm movement plans have to be prepared for every workstation in the working environment. Also, in case of changing the position of the target in the workstation or changing the position of workstation itself, then the related arm files can't be used and new files have to be prepared. Therefore, the professional and scientific way of dealing with this problem is to use the kinematic analysis. The inverse kinematic model calculates the required joints' values of the robotic arm according to the target pose on the workbench. Finally, with the blind method, there is no feedback information related to the target pose. Therefore, different strategies and visual sensors have to be used to provide the target pose for the kinematic model.

5.4 Blind Manipulation using Kinematic Analysis

A labware manipulation strategy has been implemented using the IK solution and ultrasonic sensor to calculate the distance between the H20 robot base and the required work bench. Some essential steps such as calibration and accuracy testing have been implemented as preliminary procedures before performing the labware manipulation strategy.

5.4.1 Angle to Servo Position Conversion

As an initial step for applying the kinematic model with the robotic arm, a conversion process has been performed to convert the required angles' values of the joints to the related servo motors positions. The positioning resolution of the H20 arm servos (joints) is 0.09°/unit. It means that the servo needs to move 1000 units to rotate 90° degrees. But this resolution value can't be used with the H20 arms because they are unstable and have weak joints where the joints compliance causes positional errors. The effects of gravity, weight of arm parts, payload, and inertia cause the elasticity of each joint [198]. Also, the differences between the actual physical joint zero position and the physical joint zero position reported by the robot controller normally causes accuracy errors for the robotic arm. To cope with this issue, a digital tilt meter has been used (see Fig. 5.9). Different angle values have been configured for every joint using the tilt meter and the value of the related servo motor has been registered at each configuration. This process helps to build the equations of angle to servo conversion which is important for decreasing the accuracy errors [186].

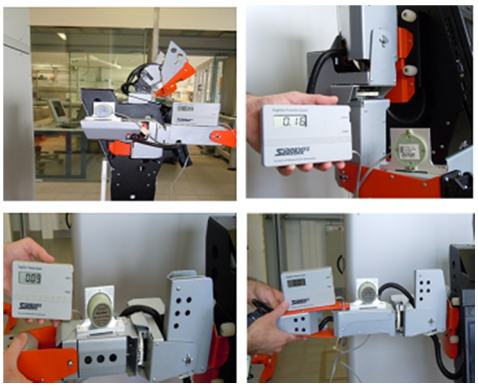


Figure 5.9: Calibration process using tilt meter.

5.4.2 Accuracy and Repeatability of the H20 Arms

The checking of the accuracy and repeatability of the robotic arms is an essential step for the tasks of object manipulation. The repeatability of the robotic arm describes how precisely this arm can return to a taught position. In general, larger robots have larger errors in repeatability. On the other hand, the accuracy of the robotic arm describes how precisely this arm can reach the required position. One of the main technological limitations in the robotics industry is how to improve the accuracy by reducing the error between the tool frame and the goal frame. The precision depends on some elements such as the resolution of the control system, joint compliance, and the imprecision of the mechanical linkages and DC servo motors. Also, the accuracy and the repeatability depend upon many other different factors such as friction, temperature, loading, and manufacturing tolerances. In case that the robotic arm doesn't

provide the required accuracy, the arm has to be calibrated. Robot calibration can be performed using both contact and noncontact probing methods. Non-contact methods include the using of beam breakers, laser sensors, visual servoing, etc. [198]. The accuracy and repeatability of the H20 arm have been checked using a grid paper and a marker attached at the end effector as shown in Fig. 5.10. A grasping configuration has been prepared to make the end effector reaches the position of X=30mm, Y=180mm, Z=380mm, related to the arm shoulder.

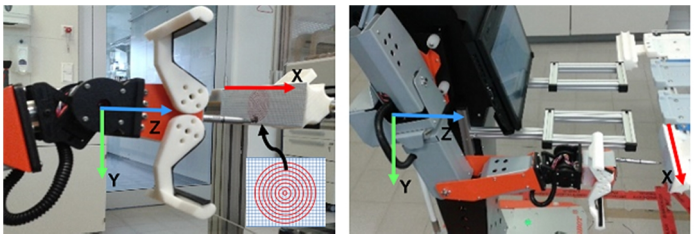
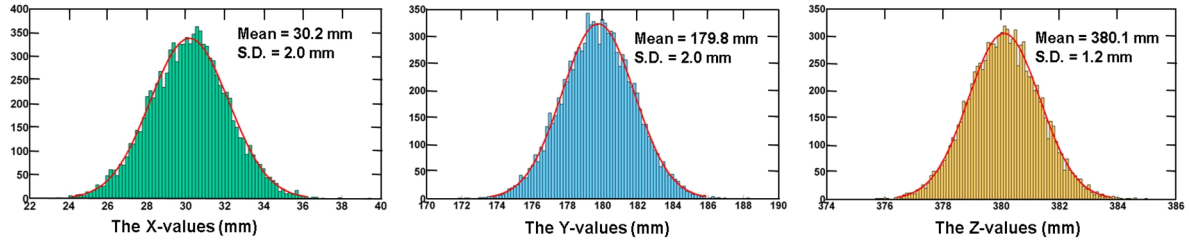


Figure 5.10: The grid paper and the marker attached to the end effector.

The arm grasping movement has been repeated 40 times where the position of the end effector has been registered at the end of every movement with the indication of the marker on the grid paper. The registered positions range of the end effector are described in Table 5.1. It can be noticed from the case "before calibration" in Table 5.1 that there is an accuracy error in the Y-axis where the expected position is not within the range of the registered positions. On the other hand, the expected position in X and Z axes are within the range of the registered positions. The reason of this error is the weakness of the joints with the joint compliance which is effected by the weight of arm parts. To improve the accuracy of reaching the required position, the robotic arm has to be calibrated. The accuracy and repeatability of the H20 arm have been checked again after performing a calibration process and the positions of the end effector have been registered as shown in the case "after calibration" in Table 5.1. Also, the Gaussian distributions of the end effector positions after calibration with the related mean and standard deviation are shown in Fig. 5.11. According to the results that have been obtained from this experiment, the accuracy of the used arm to reach a specific position according to the related configuration is as the following: (X:±4mm, Y:±4mm, Z:±2mm) [186].

Table 5.1: The expected and registered positions.

Case	X (mm)	Y(mm)	Z(mm)
The expected position (end effector related to shoulder)	30	180	380
The registered positions (before calibration)	26 ~ 34	195 ~ 203	378 ~ 382
The registered positions (after calibration)	26 ~ 34	176 ~ 184	378 ~ 382



(*A*) (*B*) (*C*)

Figure 5.11: Gaussian distribution for end effector positions after calibration, A: for X-values, B: for Y-values, C: for Z-values.

A calibration process for the robotic arm has been performed to keep the end effector at fixed height of 180 mm for different distances between the shoulder and end effector where the value 180 mm represents the height between the robot shoulder and the labware handle on the workstation. This has been done by inserting specific Y-value to the IK model at each specific distance as shown in Fig. 5.12. For example, in case that the required distance is 400 mm then, the Y-value which has to be inserted to IK is 155 mm, to keep the end effector at the height of 180 mm.

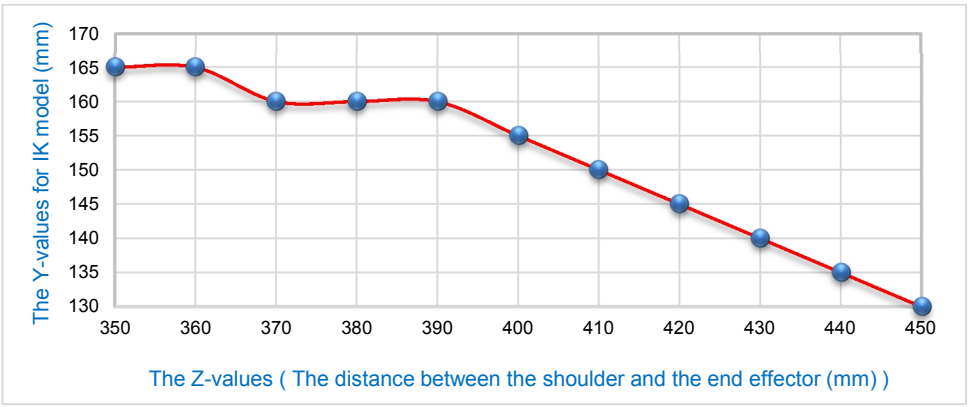


Figure 5.12: The values for calibration process to keep the end effector at the height of 180 mm for different distances.

5.4.3 Client-Server Model for Manipulation

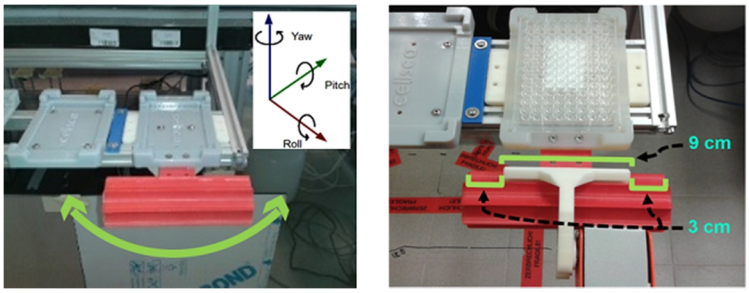
The programming code to control each device or component related to the life science automation system has been developed using a specific language. It is complicated and problematic to integrate all the control platforms in a single one because sometimes each platform includes a huge number of code lines or the platforms have been developed using different programming languages. Therefore, it is required to develop a communication system that enables the simultaneous interaction of all the devices for a flexible process execution. The control systems of the robot components are connected in a common LAN. The client-server communication model can enable the control system of each component to interact with the

others over Ethernet using a specific IP address. The client initiates the process with the server by requesting a connection to a specific socket address using TCP/IP where the socket address is a combination of IP address and a port number. If the requested port is free, then the server will establish the connection to communicate with the client. For the task of labware manipulation, a client-server model has been developed to connect the robot on-board center (RBC) with the robot arm control (RAC). The RBC (client) sends the commands to the RAC (server) that include the orders related to the required task (grasping or placing) with other information related to the position of the target which can be obtained using specific sensors such as ultrasonic sensor. Then, the RAC will insert the received information to the kinematic model to control the robotic arm.

5.4.4 Blind Labware manipulation

For the verification of the developed IK solution with a real application, an object manipulation strategy has been performed to achieve the labware transportation in different labs of the center for life science automation (celisca) using H20 robots. This strategy depends on kinematic model and DUR5200 ultrasonic sensor. The ultrasonic sensor can be used for different applications such as map building for mobile robot environment, collision avoidance, robot range finder and distance detection. The DUR5200 ultrasonic range sensor module can detect the range information from 4 cm to 340 cm. The distance data is precisely calculated by the time interval between the instant when the measurement is enabled and the instant when the echo signal is received.

For the process of grasping and placing, the API and the communication process between the RBC and the RAC have been implemented. As the H20 robot arrives the desired location in front of the workstation, the first step, which will be performed, is the orientation correction of the robot to be straight. As the labware on the workstation has a specific posture (see Fig. 5.13.A), the pitch and roll orientation related to the robot are fixed. But the yaw orientation has to be corrected. The yaw orientation has been corrected using two sonar sensors mounted on the base of H20 robot (see Fig. 5.14). The distance (Z) from each channel to the workstation is checked and the robot is rotated till the values of both sensors are equalized. Also, the height of every workstation in the labs is known. But, there is an error value of ±2cm in the X-axis of desired robot position in front of the workstation. This error has been compensated by the design of grippers and labware handle as shown in Fig. 5.13.B. There is a space range of ±3cm between the lengths of grippers and handle where this space range compensates the X-error of robot position to guarantee a secure grasping.



(*A*) (*B*)

Figure 5.13: The labware on the workstation, A: The posture and yaw orientation of labware, B: The design of handle and grippers

Using the client-server communication model, the orders are sent from the RBC to the RAC. The order includes the required task (grasping or placing), the height of the workstation, and the distance between the robot base and the workstation. Depending on this information, the labware pose related to the arm shoulder is found and checked whether it is inside the arm reachable space or not. If it is inside the reachable space, the IK solution will be calculated and the joints limits will be checked. If there are multiple solutions, then a decision procedure will be used to select the solution with minimum joints motion. Afterwards, the converting equations will be used to convert the angle value of each joint to servo position value to move the arm to the required pose as shown in Fig. 5.14. The average of accuracy with this strategy of labware manipulation is less than 1cm and the task of grasping or placing takes about 40 seconds to finish the process. The flowchart of labware manipulation strategy is shown in Fig. 5.15 [186].

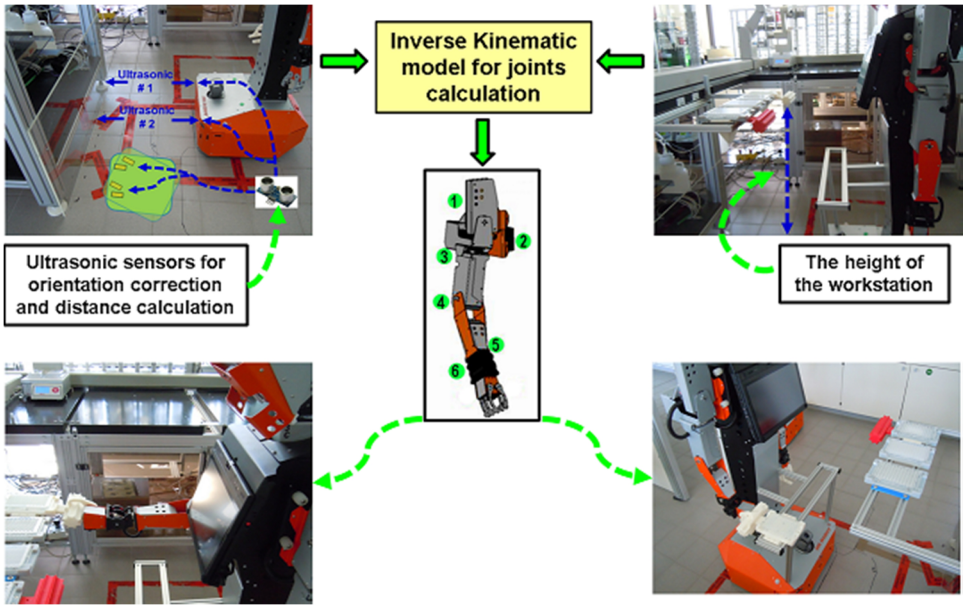


Figure 5.14: The framework of labware manipulation.

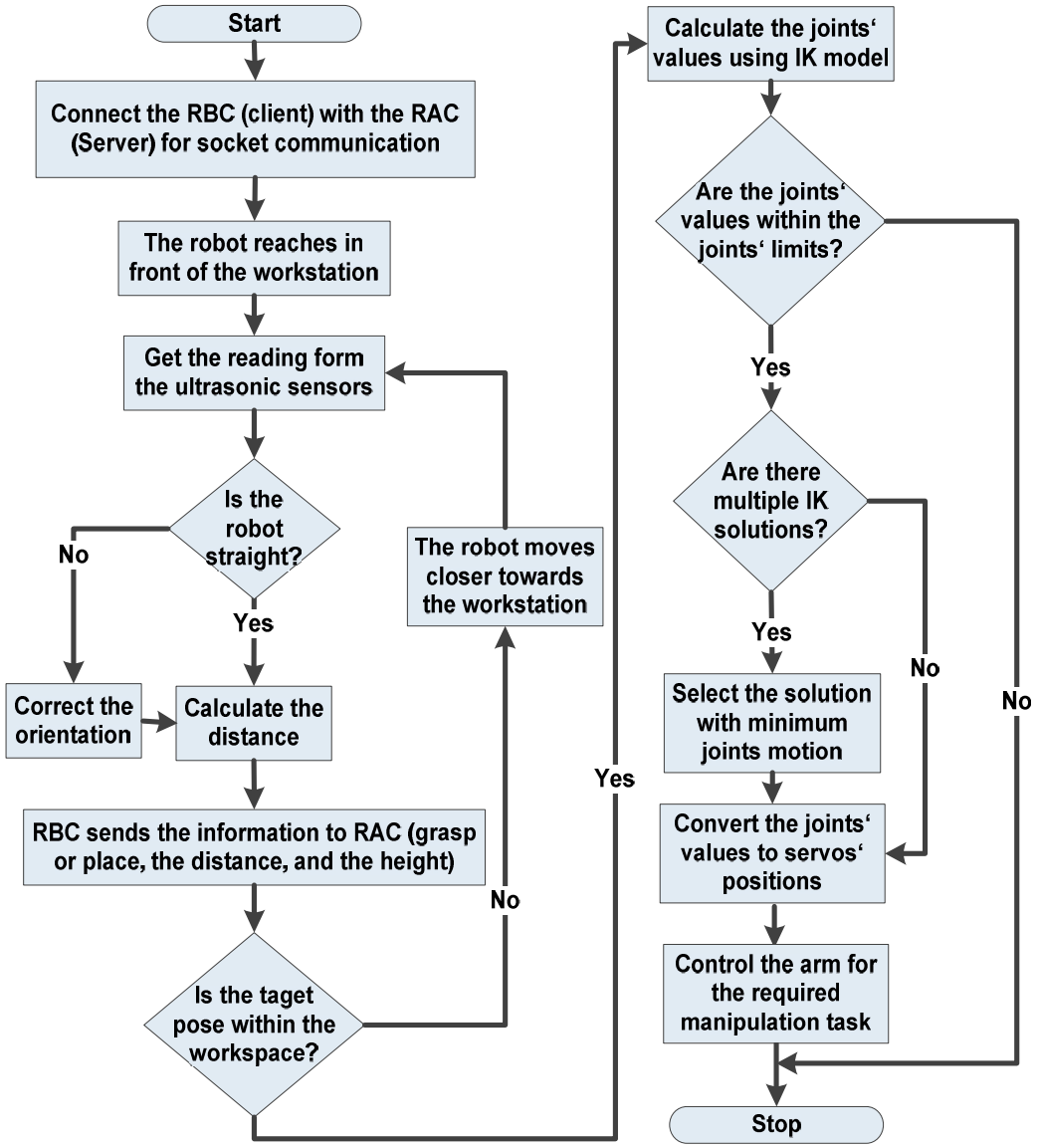


Figure 5.15: The flowchart of blind labware manipulation.

The manipulation using kinematic model is more scientific in comparison with the strategy which depends on previously prepared XML files of joints' values. The arm moves in a 3D space to reach the target position wherever it is located on the workbench. Also, the manipulation with kinematic model is flexible because it can't be affected by the changing of workstation position. The arm kinematic model of one robot can be generalized to the all other H20 robots and this leads to save the time and effort. Related to the accuracy, the manipulation using kinematic analysis has better accuracy (less than 1cm) in comparison with the previous one (2cm). However, the success rate of the manipulation tasks using blind strategy depends mainly on the accuracy of robot position. Therefore, the visual sensors are very necessary to provide feedback information of the target pose to improve the manipulation performance.

5.5 The Design of Grippers and Labware Containers

The design of the grippers is very essential to guarantee a safe manipulation for different targets. The life science laboratories include different labware which have different models and shapes. Therefore, it is impossible for the H20 arms to deal with these labware directly. To cope with this issue, labware container, which fits all labware models is designed. Particular style of the gripper is required for each specific design of labware container to perform the manipulation task. Fig. 5.16 shows the relation between the grasping performance and flexibility related to different grippers design. In this section, the designs of labware container with grippers and labware holder will be discussed.

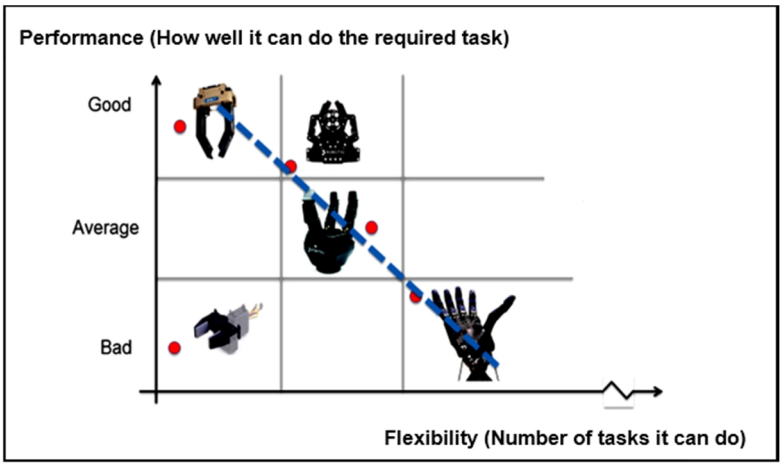


Figure 5.16: The relation between performance and flexibilty of grippers design [175].

The design of labware containers and grippers is one of the important issues which have to be taken into consideration to guarantee a secure manipulation for the labware which contains chemical and / or biological components. Fig. 5.17 shows the initial design of gripper and labware container. A rubber has been attached to the end effector and gripper to increase the friction for safe manipulation. The labware container is attached with a handle to simplify the manipulation task. The handle has a specific single color to be distinguished from the others. Also, the labware container base has been designed to be guided by the placing holder in case of positioning errors [199].

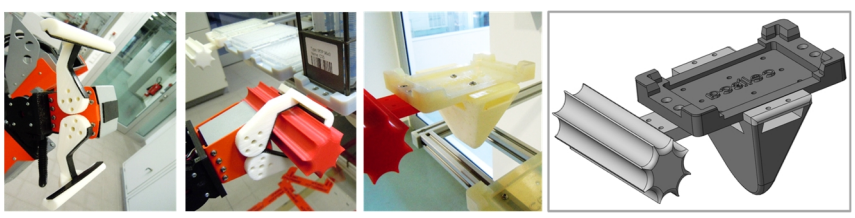


Figure 5.17: The first design of grippers and labware container handle.

To improve the identification of multiple labware containers and to increase the friction with the arm end effector, a new handle design with flat panels on the upper and front sides has been resolved. The upper flat panel is used for fixing different colored or pictorial marks (glyph, 2D barcode, etc.) to distinguish multiple handles as shown in Fig. 5.18.

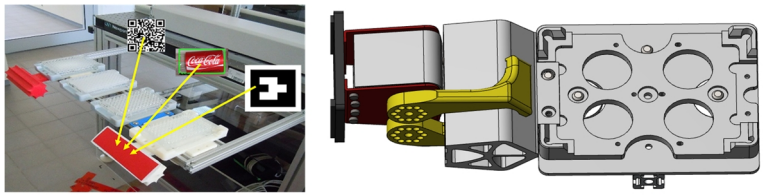


Figure 5.18: The handle design with flat panel for fixing a graphical code.

A rubber has been fixed on the front panel of the handle for secure manipulation. Proper gripper has been designed also to fit the new handle. The labware container base has two cones to be guided by the placing holder which in turn has two circular holes to guarantee a secure placing tasks as shown in Fig. 5.19 [199]. The possible payload which can be handled with this model is about 350g.

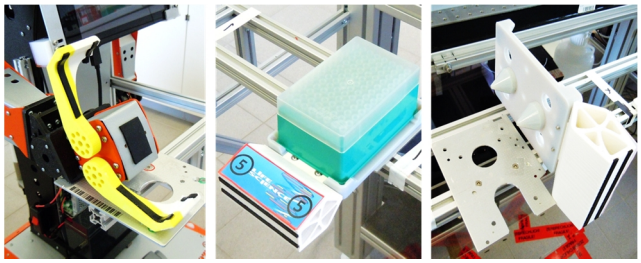


Figure 5.19: The new design of grippers and handle with flat panel.

To manipulate heavier labwares, a vertical handle has been designed. There is a flat panel on the upper side of the handle for fixing pictorial marks as shown in Fig. 5.20. With the vertical handle, the arm configuration will be in the form which locks the weak wrist joint when the gripper grasp the handle. In this case, the lifting process depends on the elbow joint which has more powerful torque [186]. The possible payload that can be lifted safely with this design is 500g.

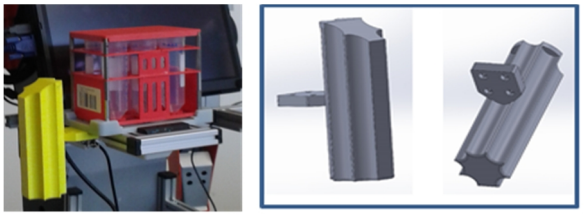


Figure 5.20: The design of vertical handle with flat panel.

Fig. 5.21 shows the wrist and elbow joint of the H20 robotic arm for the case of grasping horizontal handle (the previous one). The wrist joint is very weak and it is unable to lift heavy labwares which leads to unsecure manipulation. With the vertical handle, the lifting movement of the wrist joint will be locked as shown in Fig. 5.22. The elbow joint, that is more powerful than wrist joint, will be responsible for lifting the heavy labwares from the holder. This strategy has been cancelled due to the inconsistent motion of the arm during the manipulation.

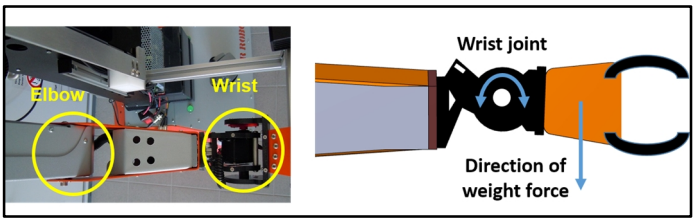


Figure 5.21: The arm configuration for the manipulation of horizontal handle.

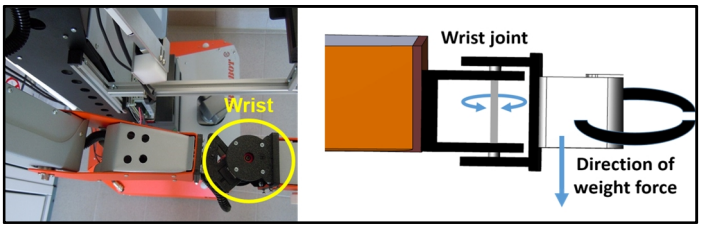


Figure 5.22: The arm configuration for the manipulation of vertical handle.

New fingers with labware container have been designed for labware manipulation as shown in Fig. 5.23. The goal of this final design is to decrease the lever arm of the wrist joint by removing the attached handle. In this case, the labware weight center will be closer to the wrist. This leads to decrease the required torque which the wrist joint has to provide for lifting the labware as shown in Fig. 5.24. To implement this strategy, the arm end effector of the H20 has been rotated to be suitable for this kind of manipulation. Also, the movement limit of the shoulder joint has been increased by removing the shoulder lock. More additional support has been added to the wrist joint by fixing pulling springs with suitable tension. Different springs with different tension values have been tested to guarantee a secure application which doesn't affect the servo motor of the joint. The maximum payload which can be handled with this model is 700g [200].

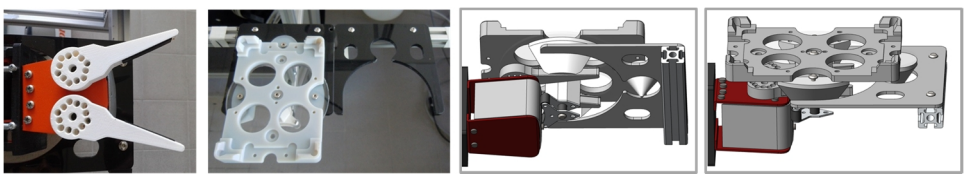


Figure 5.23: The final design of fingers and labware container.

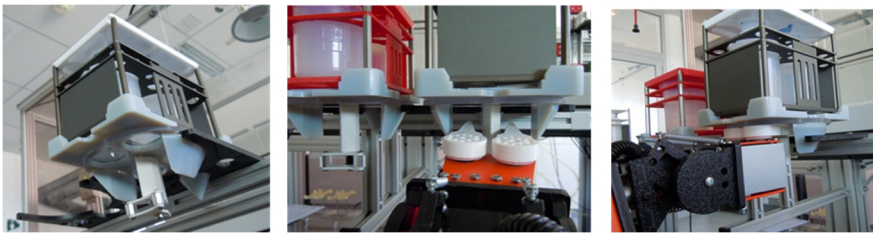


Figure 5.24: The labware manipulation using the final design.

5.6 Labware Holder on the Robot Body

To guarantee a safe transportation for the labware and to avoid the spilling of its contents, different designs of labware holders have been fixed on the body of H20 robot. It is important to keep the labware in horizontal posture because it contains chemical and biological components. This posture has also to be taken into consideration during the arm motion planning for grasping and placing. The other important issue is that the robot movement during transportation tasks and the arm movement during the manipulation tasks have to be performed at a reasonable speed. Fig. 5.25 shows the labware holder frame which has been designed for the labware container that was shown in Fig. 5.17. Related to the labware container that was shown in Fig. 5.19, another holder frame has been designed and fixed on the H20 body that is shown in Fig. 5.26. For the vertical handle, the arm configuration and movement plan are different in comparison with the previous cases because of the arm design and workspace. Therefore, the only one possibility to locate the holder frame is to fix it in front of the H20 tablet as shown in Fig. 5.27. This actually affects the appearance of the robot and limits the action with the H20 tablet. The other disadvantage is that the arm movement for grasping and placing is not secure enough to avoid the spilling of labware contents.

Figure 5.25: Holder frame of the labware container that was shown in Fig. 5.17.

Figure 5.26: Holder frame of the labware container that was shown in Fig. 5.19.

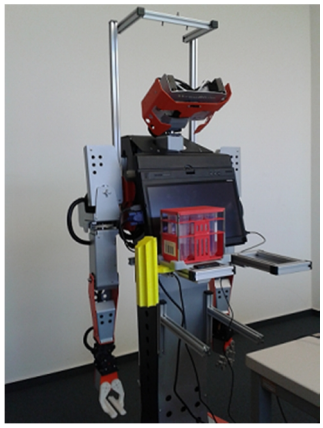


Figure 5.27: Holder frame of the labware container with vertical handle.

Related to the final design of the labware container which was shown in Fig. 5.23 and Fig. 5.24, a labware holder has been resolved to fit the labware container. Two labwares can be placed on this holder to be transported to the required laboratory where the holder has two positions for the left and right arms as shown in Fig. 5.28.

Figure 5.28: Holder frame of the labware container that was shown in Fig. 5.23.

6 Visual Labware Manipulation

The visual sensors represent a very important source of feedback information to be used in the object manipulation tasks. Stereo vision and 3D camera are the most common visual sensors that can be used in the robotic field. 3D camera, like Kinect sensor, can be considered as the best solution to detect and localize the required target in the working environment. This chapter includes the possible methods and algorithms that have been used to perform the visual manipulation for the required labware.

6.1 Handle Detection using Kinect Sensor V1

For the process of object detection, there are a lot of approaches to extract features from raw data such as edge detection, color detection, and shape detection. The Kinect sensor V1 has been used to apply the strategy of object detections, which is based on color information. For robotic manipulation, the detection component is reduced to the simplest possible system that fulfills the desired requirements in a controlled environment. Color segmentation is one of the quickest and easiest methods because it does not need prior information about the object. The speed of this technique makes it very popular for real-time applications. The desired color is detected by applying a special filter on the image frame of Kinect. The color is defined as a range of RGB values. The filter keeps the desired pixels, which their values are within the range, and fills the rest with black color. In case of presenting more than one object, which have the same color in the scene, it will be impossible to detect the desired object depending on color space only. Therefore, it is necessary to use additional data sources, such as size, distance, or shape in order to discard the objects that are not in our interest. After the color filtering has been applied, the desired size range is defined by setting the minimum and maximum number of pixels for width and height. Then, the object with the desired color within the size range is recognized by drawing a bounding box around its edges. The shape and distance information can also be used as additional conditions for object detection with this algorithm. Some simple shapes can be defined (rectangle, ellipse, triangle, and so on) as a specific feature for the object of interest to be detected. For the grasping task, a robust grasping point has to be selected to guide the robot arm. The selection of the grasping point depends on the structure of the desired object with arm gripper, the detection algorithm, and on the camera angle of view. Most of the grasping applications use the centroid of the object as a grasping point. The scenario works through the features extracted from 2D-RGB images looking for points that are related to the desired object being detected. Then, a mathematical equation is used based on an assumption that the required (x, y) position of the target is the mean of all (x, y) position values of the target in the view. This assumption works on the basis that the detected objects are generally symmetrical. This can be done by calculating the total value of all the x

and y positions. Then, the total is divided by the total number of points related to the object as the following equation:

$$X_c = \frac{\sum_i^n xi}{n} \ , \ Y_c = \frac{\sum_i^n yi}{n} \ , \tag{6.1}$$

where n is the number of pixels in the detected object. Also, the center point can be determined by finding the centroid of the bounding box drawn around the object using the box corners' positions. For the manipulation task, a cylindrical handle has to be detected and the grasping point, which is in the center point of the handle bottom edge, has to be found as shown in Fig. 6.1 [195].

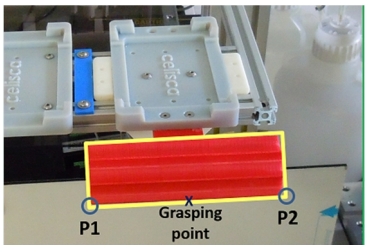


Figure 6.1: The grasping point of a labware handle.

Eq. (6.1) has been applied for the points of the bottom edge to find its center. Another way of finding the grasping point is by calculating the average of the end positions (P1, P2) of the bottom edge. Finally, the Kinect sensor will provide the localization information by finding the 3D coordinate (x,y,z) of the grasping point depending on the information of depth frame. The Kinect sensor is very useful to get better results for the object localization since the distance of the desired object can be immediately found without the need of deep studying for the image data like some other algorithms do. The flowchart of manipulation algorithm is shown in Fig. 6.2.

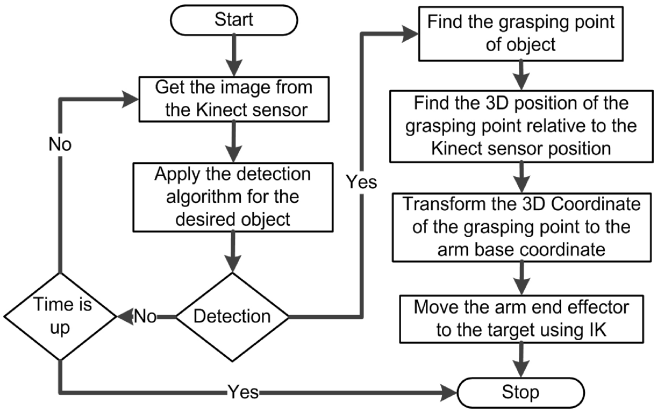


Figure 6.2: The flowchart of manipulation algorithm.

A single colored handle attached to a labware container has been detected and localized with Kinect sensorvV1 using RGB color and size information as shown in Fig. 6.3 [195]. The position of the labware relative to the robot determines which arm (the left or the right) will be used for a suitable grasping. One main fundamental observation for grasping with visual sensor is that the mobile robot should be in front of the workbench with close distance to get accurate position information for the desired object and to guarantee that the handle is inside the reachable space.

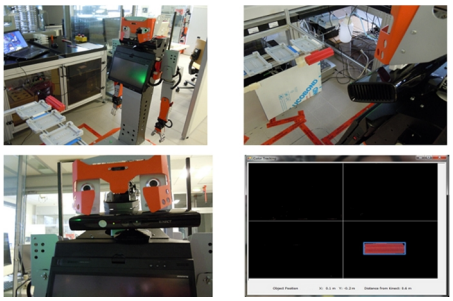


Figure 6.3: The detection and localization of labware handle using Kinect V1.

Some feature points or marks have been added on the workbench to be detected and localized. The positions of these points will be used to improve the measurements of the position and orientation for the labware handle. Fig. 6.4 shows how the labware handle and some points on the holder have been detected according to the color and size information.

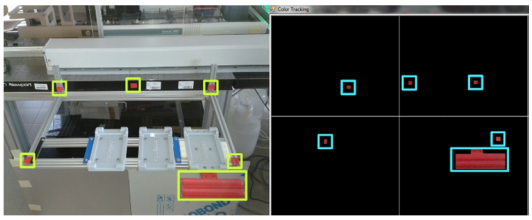


Figure 6.4: The detection of labware and some points on the workbench.

For the placing strategy, it is very important to add some signs or marks on the corners of the labware holder. These marks will be detected and localized to find the position where the robot has to place the labware [195]. Fig. 6.5 shows the detection of marks for placing task.

Figure 6.5: The detection of marks for placing task.

This process of detection and localization has been performed with Microsoft visual studio 2010 using C# programming language.

6.2 Extrinsic Calibration

Since the visual sensor is used to find the 3D position and orientation of the desired object, but this information has to be transformed to be related to the arm base. Then, the IK model is used to find the joints solutions for the desired pose. Fig. 6.6 shows a simple example about how to perform the extrinsic calibration where the transformation matrix from visual sensor to the arm base is as follows:

$$ {}^{camera}_{shoulder}T = \begin{bmatrix} n_x & s_x & a_x & J \\ n_y & s_y & a_y & -K \\ n_z & s_z & a_z & L \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Where J is the distance between the visual sensor position and arm base position along X-axis, K is the distance between the visual sensor position and arm base position along Y-axis, and L is the distance between the visual sensor position and arm base position along Z-axis.

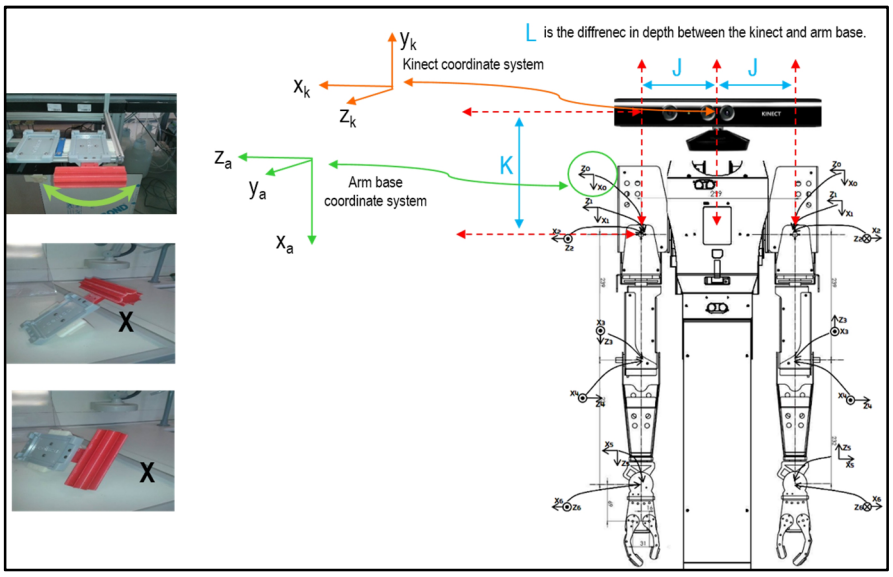


Figure 6.6: The camera extrinsic calibration for arm manipulation.

It can be seen clearly from Fig. 6.6 that the roll and pitch vectors of the handle pose are fixed according to its placing posture on the workstation. The yaw vector values depend on the orientation of robot in front of the workstation which also can be fixed in case that the robot is straight oriented. In order to find the final pose information of the target related to the arm base, the Kinect-to-shoulder transformation matrix has to be multiplied with the pose matrix of target

related to Kinect. Then, the final matrix is inserted to the IK model to control the arm joints and guide the end effector to the target. In case that the Kinect is not straight positioned and there is a tilt difference with robot arm base, then additional transformation matrix has to be found. The Kinect-to-shoulder transformation includes the difference in the position and the tilt angle (t) between them according to the Kinect holder as show in Fig. 6.7. This transformation is vital to use the visual input as a reference for manipulation or interaction.

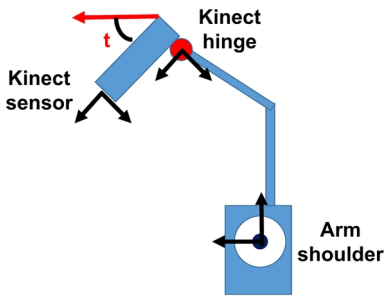


Figure 6.7: Kinect-to-shoulder transformation.

The Kinect-to-shoulder transformation consists of two steps. The first is the transformation from the Kinect sensor to the hinge, then, the transformation from hinge to shoulder. The transformation matrices are as follows:

$$ {}^{k}_{h}T = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{h}_{sh}T = \begin{bmatrix} \cos t & -\sin t & 0 & d \\ \sin t & \cos t & 0 & e \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

where, a,b,c represent the position differences in x,y,z axes between the Kinect and hinge. On the other hand, d,e,f represent the position difference in x,y,z axes between the hinge and arm shoulder. The tilt angle is represented by "t". The transformation from the Kinect sensor to hinge is easier in the process because the changes are just in the translation but not in the rotation. To find the final matrix to be inserted to the IK model, the process shown in Fig. 6.8 is performed.

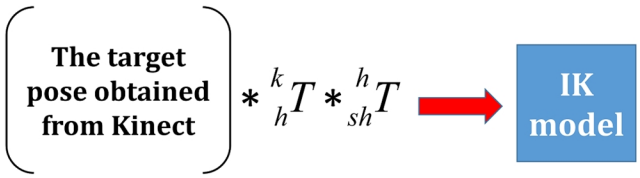


Figure 6.8: The final matrix calculation process.

The extrinsic calibration process has been applied between the Kinect and the arm base of H20 robot. A Kinect holder has been designed very carefully to be fixed on the body of H20 robot as shown in Fig. 6.9. Several factors have to be taken into the consideration at the same time for designing and fixing the Kinect holder. The Kinect pose has to realize a sufficient and clear view for all the labwares on the workstation. The Kinect sensor has to be at sufficient distance to the workstation to get a sufficient position information of the targets where the FOV of RGB camera is wider than the FOV of depth camera. It means that not all the points in the normal view has a related 3D position value. Also, the Kinect holder has to be in a suitable tilt which doesn't affect the stargazer view which is located behind the H20 head. The holder has to be very stable and fixed to guarantee that there is no any kind of changes in the Kinect position and orientation during the robot movements. Some parts have been added to the Kinect to fix its tilt and rotation movement on the holder as shown in Fig. 6.9.

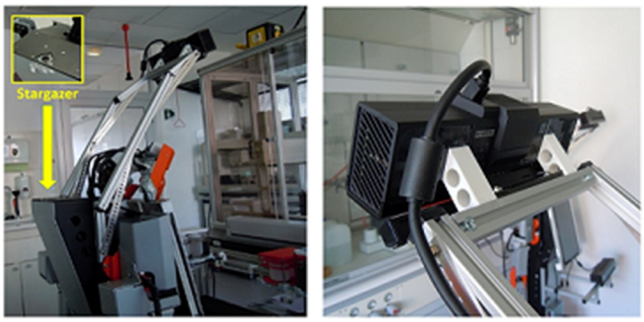


Figure 6.9: The kinect holder and how to fix the 3D orientation of Kinect.

6.3 Manipulation of New Handle Using Kinect V2

To improve the identification of different handles and to increase the friction with the arm end effector, a new handle design with flat panels on the upper and front sides has been resolved. The upper flat panel is used for fixing different colored or pictorial marks (glyph, 2D barcode, etc.) to distinguish multiple handles as shown in Fig. 6.10.

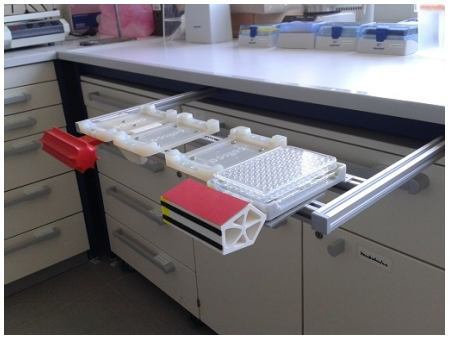


Figure 6.10: New handle design.

The H20 mobile robot with the Kinect sensor V2 are used to perform the grasping and placing tasks for the required handle where the desired labware is positioned. The tasks of handle manipulation are based on the following scheme: in the 1st step, the robot checks around the workspace to detect the desired handle which has to be transported to the required lab. In the 2nd step, the position of the target related to the arm shoulder is calculated. In the 3rd step, the robot uses the position information obtained from the 2nd step to insert it into the arm kinematic model. In the 4th step, the arm moves to the required position and grasps the handle. After finishing the grasping task successfully, the robot moves to the desired destination to place the handle on the required holder position. The HSV color system has been used to detect the required handle where the RGB color system can easily be affected by changes of lighting conditions. HSV system can be considered the most powerful system to be used for color segmentation because it is more robust to control the brightness. To differentiate multiple handles, different color can be used. In case of existing more than one object in the view that has the same color, then it will be crucial to use additional data sources, such as area, or shape to find the objects of our interest. HSV color segmentation method with shape and area information was used with Kinect V2 to detect the required handle as shown in Fig. 6.11. The flat panel of the handle has a red color with a rectangle shape and the area of the detected object is checked whether it is within the required range or not. Thus, it can be guaranteed that the detected object is related to the flat panel of the required handle. Histogram equalization is applied to the image to increase its global contrast as a preprocessing step before the HSV color segmentation. At the end, a polygon with cross is drawn around the target to define it and to identify its center point. For the transportation tasks, the information about the required handle, grasping station, and placing station are sent to the H20 mobile robot. The required labware, which is needed to be transported, has to be placed on the labware container attached with the target handle. Furthermore, a pictorial mark with specific textures and features can also be fixed on the handle to be recognized by Kinect V2 using SURF algorithm as shown in Fig. 6.12 [199]. SURF can be considered as an efficient object recognition algorithm with rotation- and scale-invariant descriptors and detectors. The process of recognition is separated into an offline object modelling stage and an online recognition stage. The objects' images are saved in the database in the offline stage. Given a new test image, the online local descriptors are matched with the stored models in the database to perform the identification process.

Figure 6.11: The recognition of handle using HSV and shape detection.

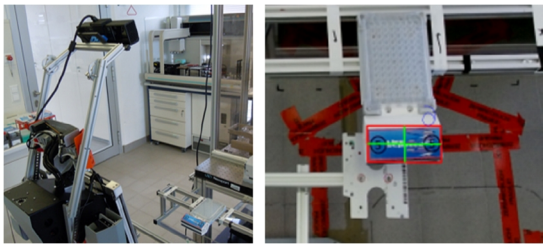


Figure 6.12: The recognition of handle using SURF.

The recognition strategies are also applied to detect the holder for the placing tasks. Color, shape, and area information can be used to detect the required position for labware placing. Also, some pictorial marks can be fixed on the holder to be recognized by the Kinect V2 as shown in Fig. 6.13 and Fig. 6.14 respectively.

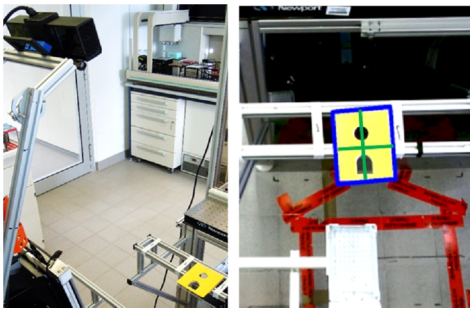


Figure 6.13: The recognition of holder using HSV and shape detection.

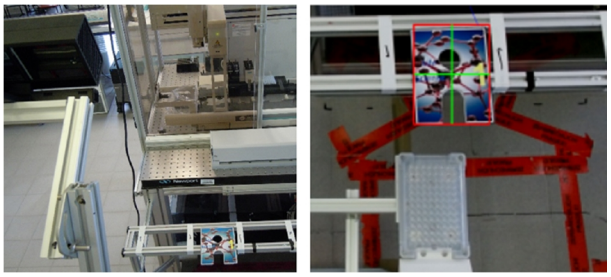


Figure 6.14: The recognition of holder using SURF.

After the target recognition, the coordinates of the center point of this target in the image frame has to be found. Then, the position of this center point related to the Kinect is identified using a mapping process. The required point is mapped from the color frame to the depth frame of Kinect. Then, another step is performed to map the point from depth frame to the Kinect space coordinates. The position of the center point is used as reference for calculating the grasping or placing point positions which the arm end effector has to reach. After identification of the position of the required point related to the Kinect, an extrinsic calibration is applied to transform the position information to be related to the arms' shoulders. Then, the inverse kinematic model is used to move the arm to the target and perform the task [199]. Fig. 6.15 shows the center and the grasping points of the handle and how the arm reaches it.

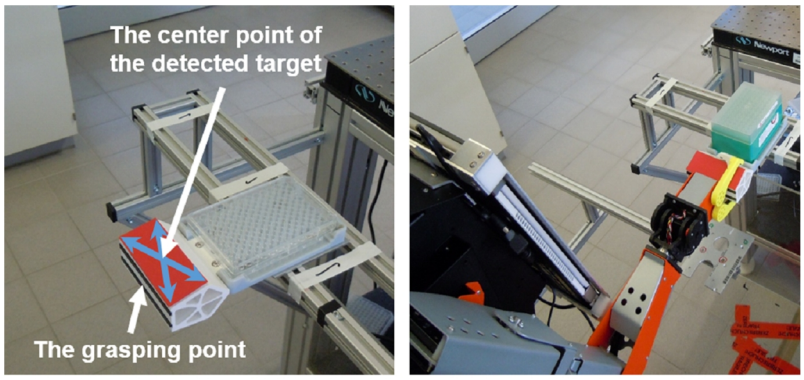


Figure 6.15: The grasping point and how the arm reaches it.

For the labware transportation framework with the H20 mobile robot, the system consists of three main coding platforms which are the robot navigation control (RNC) [201][202], the robot arm control (RAC), and the Kinect control (KC) for object detection and localization. It is a complex issue to integrate the three main platforms in a single coding project where each one includes a huge number of coding lines. Also, this integration is problematic because it increases the possibility of coding bugs which affect the whole system. To cope with this issue and to realize an intelligent performance, it is required to separate the coding platforms and develop a communication system. The client-server communication model (asynchronous socket) can enable the control system of each platform to interact with the others over Ethernet. The client-server connection is realized through a specific socket address using TCP/IP where the socket address is a combination of IP address and a port number. For the task of labware transportation, two client-server models have been developed. The first model is to connect RNC with the RAC. The second model is to connect the RAC with the KC where the RAC has a client and a server platform at the same time as shown in Fig. 6.16. The RNC (client) sends the orders to the RAC (server) that include the required task (grasping or placing) with other information related to the target specifications. The RAC (client) sends the target specifications

to the KC (server) to be found. Then, the KC sends back the target position to the RAC which in turn inserts the received information to the kinematic model to control the robotic arm. In case that the target is not existing in the Kinect view or it is not possible to manipulate it, a feedback information is sent to the navigation control related to the next procedure [199]. This feedback information includes several decisions such as changing the robot position or performing the next task. The changing of robot position helps to get better view for the Kinect sensor and to make the target within the arm workspace.

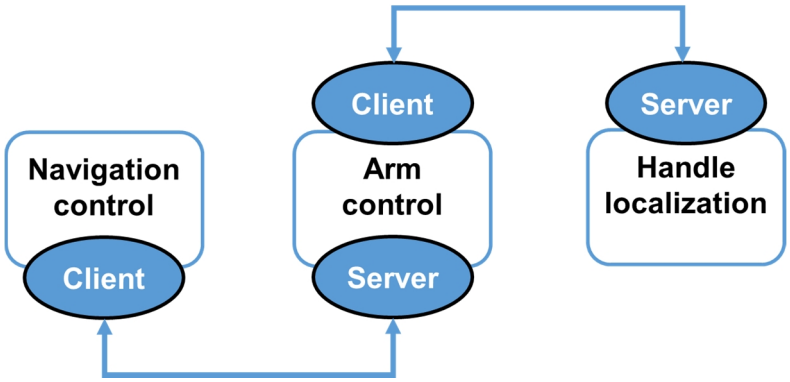


Figure 6.16: The architecture of client-server model for handle manipulation.

The grasping and placing tests were performed by placing the robot at different positions in front of the workstation. Prior to each grasping or placing attempt, the robot is placed in a new arbitrary position. The robot has a positioning error in the range of ±2cm in X-axis (left/right of the workstation) and ±3cm in Y-axis (distance) in front of the workbench. As can be seen in Table 6.1, the results show the overall success rate for grasping and placing which confirms that the statement of the used methods are acceptable enough for labware manipulation and transportation [199]. The entire required time for performing the task is about 65 seconds including the recognition, localization, IK calculation, and arm movement. The flowchart of arm manipulation system is shown in Fig. 6.17. According to the arm's workspace and to the robot position in front of the workstation, each arm can reach 2 labwares which are alongside each other. The work has been developed using Microsoft Visual Studio 2015 with C# programming language. The project is running on a Windows 10 platform in the H20 tablet.

Table 6.1: The handle manipulation tests.

Methods	Attempts	Successful grasp	Successful place
HSV filtering	50	92%	90%
SURF	50	94%	94%

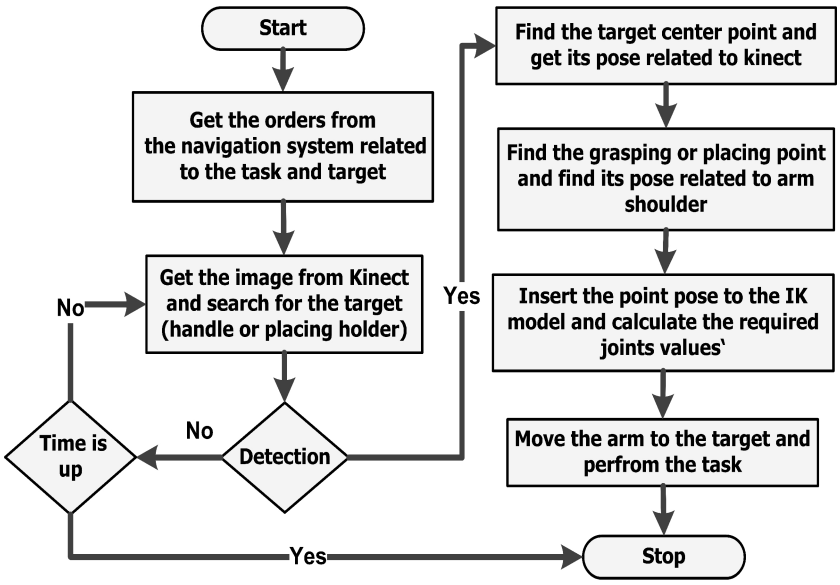


Figure 6.17: The flowchart of arm manipulation system.

A 12V-18A/h battery with voltage stabilizer have been installed on the robot body to provide the Kinect sensor with the required power during the transportation tasks as shown in Fig. 6.18. Some connectors are fixed with the power cable of Kinect to give the possibility of using the AC and DC power for Kinect at the same time [200].

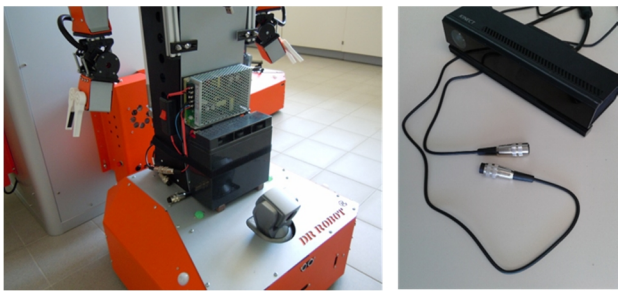


Figure 6.18: The battery and voltage stabilizer for Kinect sensor.

6.4 Labware Identification and Manipulation

As it was mentioned before in section 5.5 that the possible payload, that can be manipulated using the labware container design attached with handle, is bout 350g. The reason of this limited weight is related to the weak wrist joint of H20 arms and to the labware container design. The labware container with the attached handle requires high torque for manipulating because of the long lever arm of the wrist joint. The lever arm here represents the distance between the labware weight center and the wrist joint. For this case, the suitable way to increase the manipulated payload is to decrease the lever arm. This way is much preferable than changing the wrist joint with more powerful one because it requires to change the arm structure.

To achieve that, a new finger with labware container have been designed as it was shown in Fig. 5.23 and Fig. 5.24. The handle is removed from the labware container which make the end effector with the wrist joint closer to the labware weight center. Subsequently, less torque is required to handle heavier objects.

To perform the visual labware manipulation with this design, the labware itself has to be recognized. Different object recognition algorithms can be used to recognize different labwares such as SIFT, SURF, and FAST, which are widely used for object recognition applications. Fig. 6.19 shows some models of different labwares which have to be recognized and manipulated using the Kinect sensor and H20 robot. The labwares are placed on the workstations which each has 8 positions of labware containers as shown in Fig. 6.20. The required labware has to be distinguished and manipulated wherever it is located on the workstation. Therefore, this procedure needs a clear and wide view from the visual sensor with sufficient position estimation for the target.

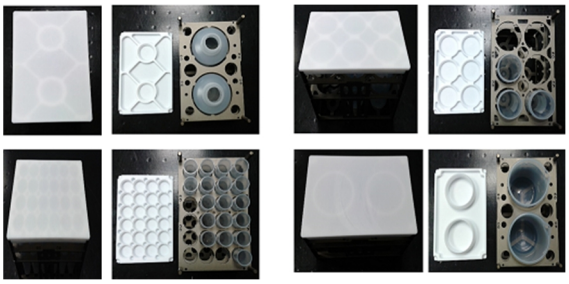


Figure 6.19: Different kinds of labwares.

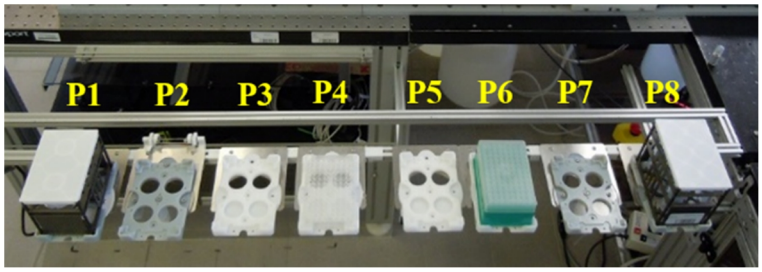


Figure 6.20: The 8 positions of labware containers on the workstation.

6.4.1 Labware Identification and Position Estimation

Different labwares have been recognized using Kinect sensor V1 and V2 with SURF algorithm. Kinect V2 has wider field of view horizontally and vertically. Also, the camera resolution and depth sensor resolution are much better in comparison with V1 which give a feasible reason to be preferable for performing the identification and manipulation tasks. Since the navigation and manipulation systems of H20 robots are developed using C# programming language,

SURF algorithm is the most appropriate method that can be implemented using this programming language. Different tests have been performed to show the performance of identification and position estimation for labwares at different positions on the workstation and under different lighting conditions. The strong and glossy light affects the recognition process because it blinds the visual sensor and changes the appearance of labware. Also, the sunlight may affect the process in case that the workstation is close to window. In addition, the objects around the workstation may also reflect some light which in turn affect the labware identification process. Therefore, different preprocessing steps need to be applied to the Kinect image such as contrast with brightness correction, histogram equalization, HSV conversion, grayscale conversion, etc. The average required time for labware identification is about 3 seconds. This time is long which has to be decreased according to the application perspective. The reason of this long time is related to the Kinect image which has high resolution (1920X1080) with wide view as shown in Fig. 6.21. The features extraction and identification of the target from such image need a long time. To cope with this issue, a cropping step is applied to extract the region of interest (ROI) from the image. The ROI in this case is the workstation with the labwares where the cropping process is performed in the Y-axis of the image only as shown in Fig. 6.22. There is an error range of ±3cm in the distance between the robot and the workbench. Accordingly, the cropping area can be estimated. The recognition process using the cropped image (1920X400) requires about (1.1 ~ 1.5 sec.) to be performed. This means that the required time has been decreased to the half after applying the cropping step.

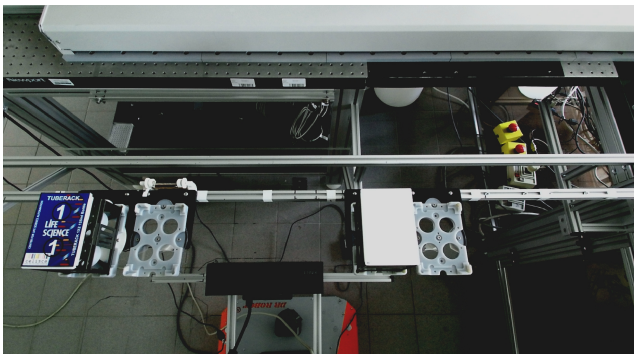


Figure 6.21: The complete FOV of Kinect sensor.

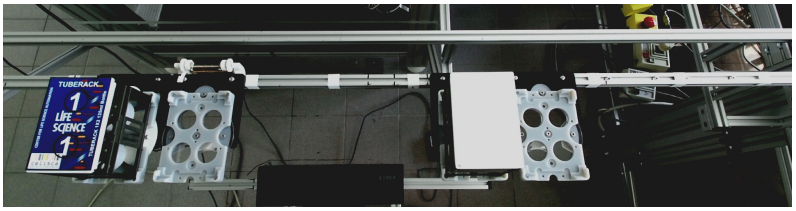


Figure 6.22: The ROI image of the workstation after the cropping step.

The identification process have been performed for some labwares without lid which are shown in Fig. 6.23 and Fig. 6.24. It is clear from these figures that the labware has different top view at different positions on the workstation. The changes in the top view features of labware are related to the 3D property of the rack with tubes and also related to the view angle of Kinect sensor. The effects of different lighting at different positions play a role also in the top appearance of labware.

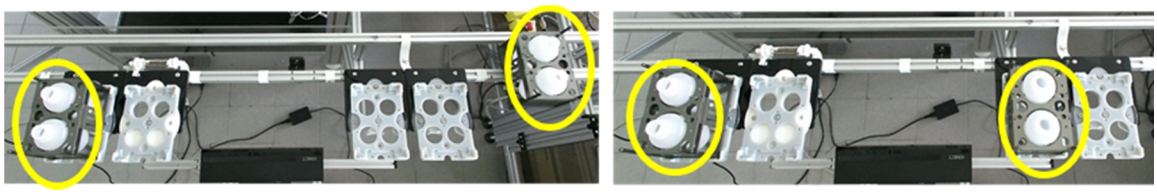


Figure 6.23: Top view of rack with 2 tubes at different positions on the workstation.

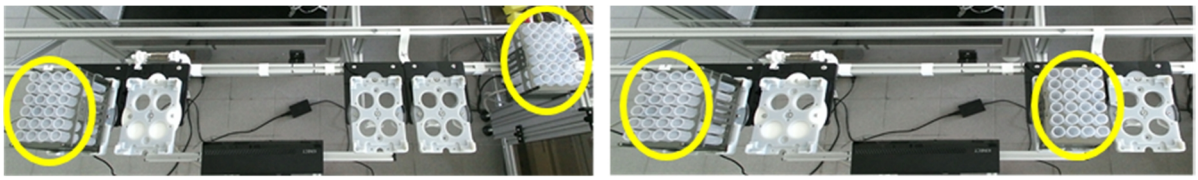


Figure 6.24: Top view of rack with 24 tubes at different positions on the workstation.

The recognition process for 2-tubes rack has been performed at 8 different positions on the workstation [203]. According to Fig. 6.20, the 8 positions are ordered from left to right in the sequence of P1-to-P8. Fig. 6.25 shows the recognition of tube rack which is assigned by drawing a polygon around it with cross to specify the center point. The top view image of the tube rack at P4 has been saved in the database to be used for feature matching with SURF. Fig. 6.26 shows the success rate of identification process which has been tested 20 times at each position with different lighting conditions and different preprocessing steps. The first test has been done without turning on the glossy ceiling light. The raw image from Kinect has been used without applying any preprocessing steps. The identification results of the first test at each position are titled with (N) in the Fig. 6.26. The success rate at P4 is 100% (20/20) because the rack top view is very clear at this position. At P3, the success rate is 60% (12/20) whereas it is zero at the other positions. The overall success rate for the first test is 20% (32/160). This rate is related to the 3D property of the rack top view which gives different features at each position according to the view angle of Kinect sensor. The second test has been done after turning on the ceiling glossy light. The second test is tiled with (N_L) and the overall success rate is ~19% (30/160) as shown in Fig. 6.26.

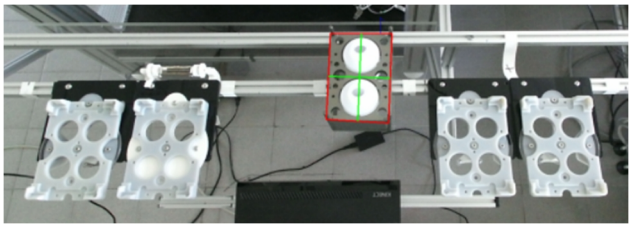


Figure 6.25: The recognition of 2-tubes rack on the workbench.

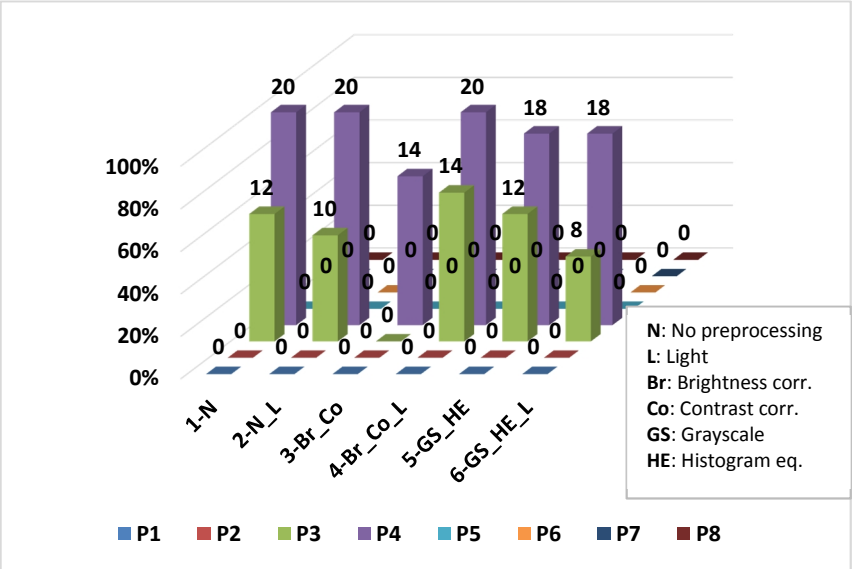


Figure 6.26: The success rate of 2-tubes rack identification.

The third and fourth tests have been performed after applying the brightness and contrast corrections as preprocessing steps for the Kinect image. These tests have been done under different lighting conditions (Br_Co & Br_Co_L) as shown in Fig. 6.26. The overall success rates are ~9% and ~21% respectively. The fifth and sixth tests have been performed after converting the Kinect image to grayscale and applying the histogram equalization as shown in Fig. 6.27. The last tests have been done also under different lighting conditions (GS_HE & GS_HE_L) as shown in Fig. 6.26. The overall success rates are ~19% and ~16% respectively.

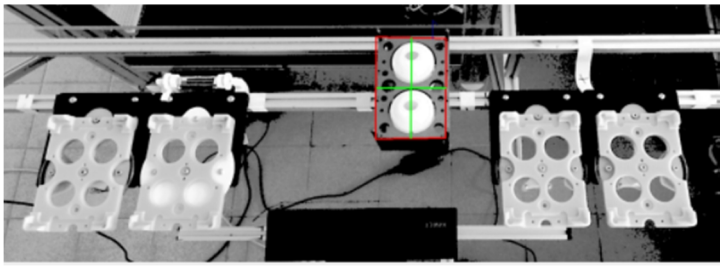


Figure 6.27: The image after grayscale conversion and histogram equalization.

According to the previous tests, the best overall success rate (~21%) has been obtained after applying the brightness with contrast correction and under lighted condition. The false positive rate for the best case is about ~14% (23/160). The false positive problem has to be avoided because it affects the manipulation process through identifying a wrong target in the view. As a conclusion from the overall results, it can be realized that the labware without lid has to be located at a fixed position on the workstation to be identified and manipulated by the robot. The labware top view at specific position is saved in the database to be used for features identification for obtaining an acceptable success rate. This can be noticed clearly according to the overall success rate at P4 which is about ~92% (110/120). Table 6.2 shows the success rates summary of the identification tests for the 2-tubes rack where the term O.S.R.T means the overall success rate for each test and the term O.S.R.P means the overall success rate at each position.

Table 6.2: The results summary of 2-tubes rack identification.

Test	P1	P2	P3	P4	P5	P6	P7	P8	O.S.R.T
N	0%	0%	60%	100%	0%	0%	0%	0%	**20%**
N_L	0%	0%	50%	100%	0%	0%	0%	0%	**~19%**
Br_Co	0%	0%	0%	70%	0%	0%	0%	0%	**~9%**
Br_Co_L	0%	0%	70%	100%	0%	0%	0%	0%	**~21%**
GS_HE	0%	0%	60%	90%	0%	0%	0%	0%	**~19%**
GS_HE_L	0%	0%	40%	90%	0%	0%	0%	0%	**~16%**
O.S.R.P	**0%**	**0%**	**~47%**	**~92%**	**0%**	**0%**	**0%**	**0%**	

To improve the success rate of multiple labwares identification at each position on the workbench, specific mark has been fixed on each labware lid. Fig. 6.28 shows different marks which can be used to distinguish different labwares according to the Kinect view. The mark contains an information about the related labware like its name and type with specified number. The labware information with the background picture give sufficient mark features to differentiate multiple labware. The more features the mark has, the higher success rate of identification will be realized. Also, it is very important to care about the colors of the background mark to decrease the effects of strong lighting conditions. It is known that the dark colors are much better to avoid the light reflection in comparison with the bright colors. The Kinect V2 has been used to find the required labware according to the related mark as shown in Fig. 6.29 [203]. The marks have been printed using normal paper (smoothed paper) that can be affected by the strong and glossy lighting conditions. Ten different marks can be identified.

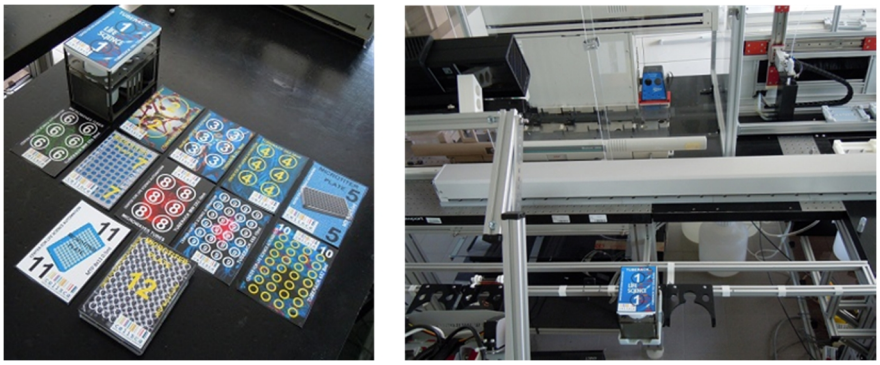


Figure 6.28: Fixing a mark on the labware lid.

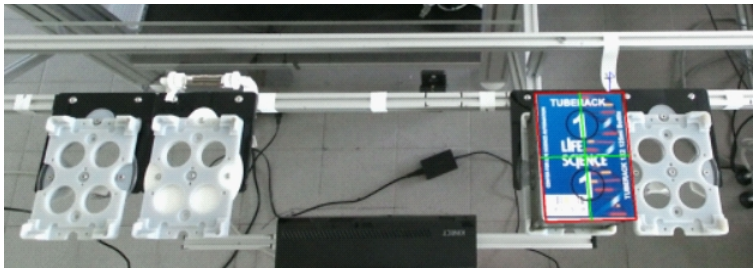


Figure 6.29: Labware recognition according to the related mark.

The labware mark has been recognized at different positions on the workbench and under different lighting conditions. Different preprocessing steps have been applied to the Kinect image before using SURF algorithm. Fig. 6.30 shows the improvements of finding the labware mark at each position. The good success rate results from the 2D property of lid mark which is not easily to be influenced by the Kinect view angle. The other reason is related to the adequate features in the mark that help to find the required mark in the view. It is clear in Fig. 6.30 that the success rate is decreased at the positions that are located at the horizontal ends of image like P1, P7, and P8. This is because of the light reflection from the mark and the Kinect view angle at these positions. It can be noticed also that the grayscale conversion with histogram equalization decreased the success rate of identification. On the other hand, the brightness with contrast correction improves the recognition process that is slightly affected under strong lighting conditions. The overall success rates for this case without and with glossy ceiling light are ~98% and 95% respectively at all the 8 positions. The other important improvement that has been realized is zero false positive rate. But, the main goal which has to be achieved is to obtain ~100% success rate and especially in strong and glossy lighted environment. Therefore, the lid mark identification tests have been repeated using coarse paper for mark printing. This kind of paper is more robust against light reflection because it absorbs the color and the incident light. The tests with applying the brightness and contrast corrections give the best results as shown in Fig. 6.31. The overall success rates for this condition without and with strong ceiling

light are ~99% and ~97% respectively with zero false positive rate. It is clear that the success rate under strong lighting condition has been improved form 95% to be ~97% after using the coarse paper. Tables 6.3 and 6.4 show the success rates summary of the identification tests for the labware mark using smoothed paper and coarse paper respectively.

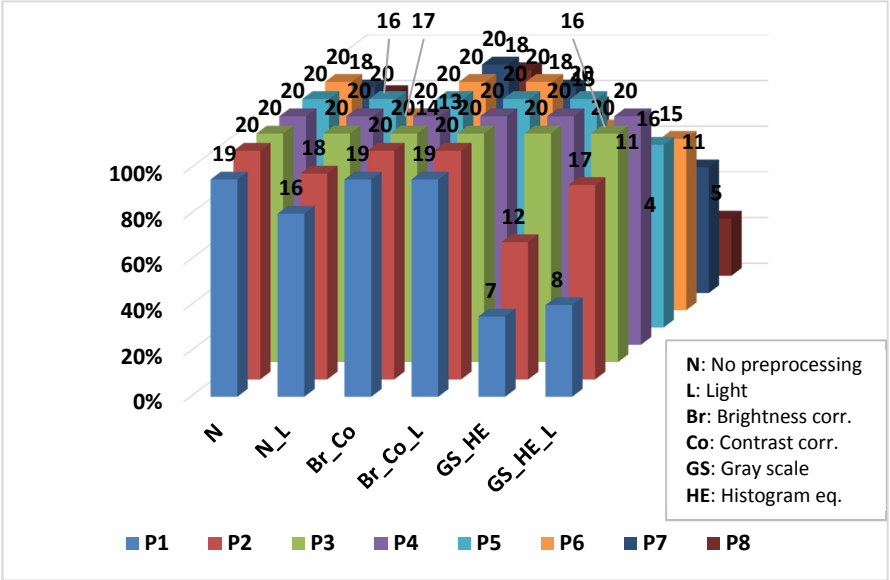


Figure 6.30: The identification success rate of the labware mark (smoothed paper).

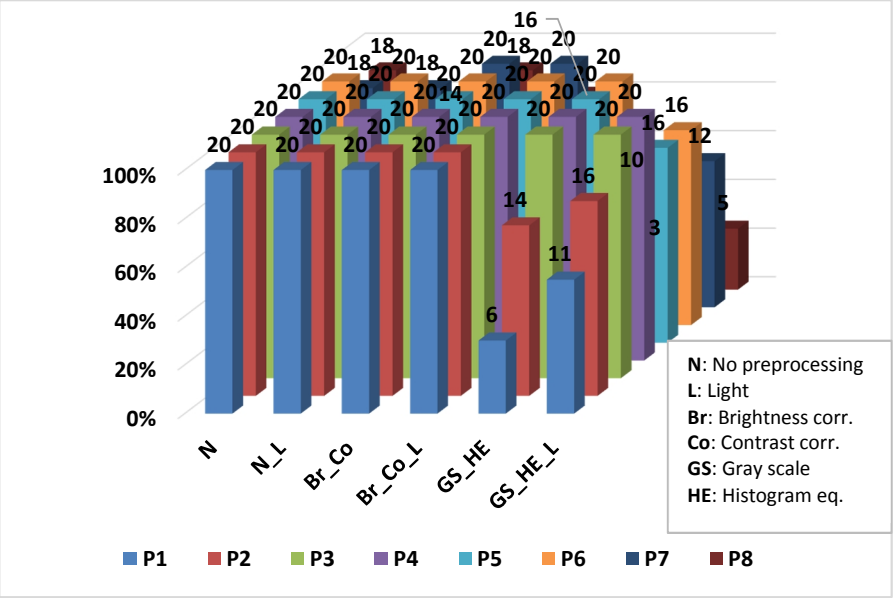


Figure 6.31: The identification success rate of the labware mark (coarse paper).

Table 6.3: The results summary of mark identification (smoothed paper).

Test	P1	P2	P3	P4	P5	P6	P7	P8	O.S.R.T
N	95%	100%	100%	100%	100%	100%	90%	80%	**~96%**
N_L	80%	90%	100%	100%	100%	85%	70%	65%	**~86%**
Br_Co	95%	100%	100%	100%	100%	100%	100%	90%	**~98%**
Br_Co_L	95%	100%	100%	100%	100%	100%	90%	75%	**95%**
GS_HE	35%	60%	100%	100%	100%	80%	55%	20%	**~69%**
GS_HE_L	40%	85%	100%	100%	80%	75%	55%	25%	**70%**
O.S.R.P	**~73%**	**~89%**	**100%**	**100%**	**~97%**	**90%**	**~77%**	**~59%**	

Table 6.4: The results summary of mark identification (coarse paper).

Test	P1	P2	P3	P4	P5	P6	P7	P8	O.S.R.T
N	100%	100%	100%	100%	100%	100%	90%	90%	**~97%**
N_L	100%	100%	100%	100%	100%	100%	90%	70%	**~97%**
Br_Co	100%	100%	100%	100%	100%	100%	100%	90%	**~99%**
Br_Co_L	100%	100%	100%	100%	100%	100%	100%	80%	**~97%**
GS_HE	30%	70%	100%	100%	100%	100%	50%	15%	**~70%**
GS_HE_L	55%	80%	100%	100%	80%	80%	60%	25%	**~72%**
O.S.R.P	**~81%**	**~92%**	**100%**	**100%**	**~97%**	**~97%**	**~82%**	**~62%**	

To cope with the problem of strong lighting and sunlight effects, changing the camera exposure time can be one of the most appropriate solutions. This leads to decrease the light effects on the image acquired from the visual sensor. Since, the Kinect V2 is used as a visual sensor for labware manipulation, it is not possible to change the setting of exposure time. Microsoft Company had already blocked the camera setting of Kinect V2 which prevent the possibility of controlling the exposure time or any other camera settings. On the other hand, it is possible to control the camera settings with Kinect V1. But, the Kinect V2 still can be considered the more desirable choice according to the higher resolution of normal and depth cameras in addition to the wider FOV.

To decrease the effects of glossy light with Kinect V2, polarization and intensity filters have been affixed on the Kinect camera as shown in Fig.6.32. It is known that the polarization filters can increase the color saturation and decrease reflections from glass, metals or other shiny surfaces. On the other hand, the intensity filter can decrease the brightness in the image and sharpen its edges and features. This can be clearly noticed in Fig. 6. 33 where the Fig. 6.33.A represents the cropped mark from the Kinect normal image. Whereas, the Fig. 6.33.B shows the impact on the cropped mark after fixing the polarization and intensity filters on the Kinect camera [203].

Figure 6.32: The Kinect V2 with polarization filter and intensity filter.

(*A*) (*B*)

Figure 6.33: The impact of using the pol. filter and intensity filter with Kinect V2.

The Kinect V2 with the filters has been used to recognize the 2-tubes rack without lid on the workbench. Fig. 6.34 shows the tests result using different methods an under different lighting conditions. According to the tests at P4, it is clear that the working environment has to be lighted enough to get an acceptable success rate. The reason is related to the intensity filter that decreases the lighting intensity. Thus, a darker image will be obtained. Table 6.5 shows the success rates summary of the identification tests for the 2-tubes rack after using the polarization and intensity filters.

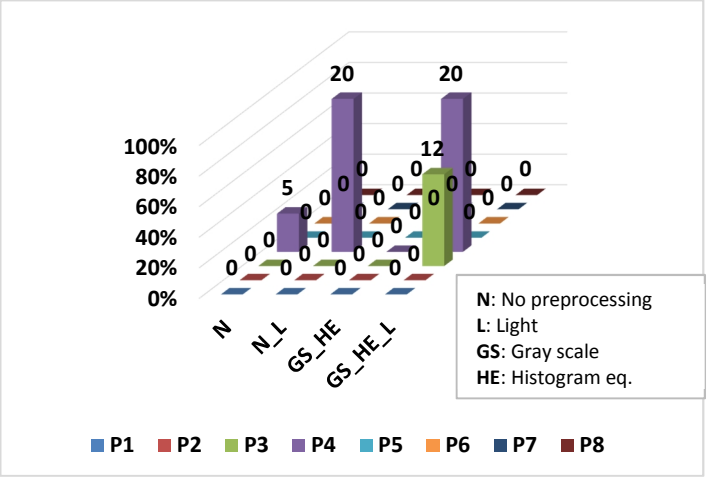


Figure 6.34: The success rate of 2-tubes rack identification (with intensity filter).

Table 6.5: The results summary of 2-tubes rack identification (with filters).

Test	P1	P2	P3	P4	P5	P6	P7	P8	O.S.R.T
N	0%	0%	0%	25%	0%	0%	0%	0%	~3%
N_L	0%	0%	0%	100%	0%	0%	0%	0%	~12%
GS_HE	0%	0%	0%	0%	0%	0%	0%	0%	0%
GS_HE_L	0%	0%	60%	100%	0%	0%	0%	0%	20%
O.S.R.P	0%	0%	15%	~56%	0%	0%	0%	0%	

The lid mark has been identified also using the Kinect with filters as shown in Fig. 6.35. Smoothed and coarse paper have been used for mark printing. The tests have been done under different lighting conditions as shown in Fig. 6.36. It is clear that the best results have been obtained with the using of coarse paper and filters. The identification success rate under strong and glossy lighting condition is 100% at the all 8 positions on the workstation. Tables 6.6 shows the success rates summary of the identification tests for the labware mark using filters.

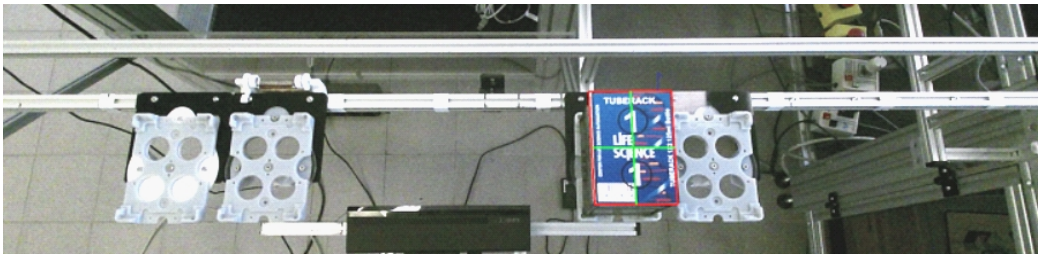


Figure 6.35: Labware recognition according to the related mark (with filters).

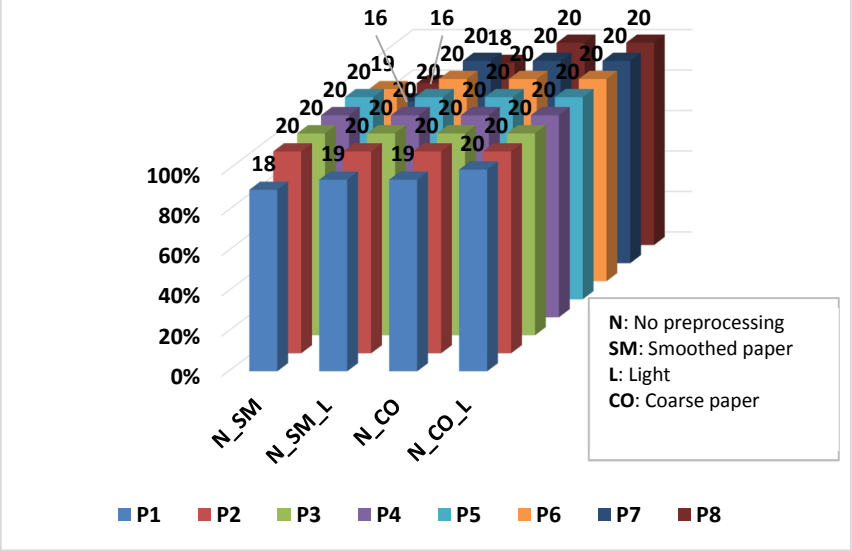


Figure 6.36: The identification success rate of the labware mark (with filters).

Table 6.6: The results summary of mark identification (with filters).

Test	P1	P2	P3	P4	P5	P6	P7	P8	O.S.R.T
N_SM	90%	100%	100%	100%	100%	95%	80%	80%	**~93%**
N_SM_L	95%	100%	100%	100%	100%	100%	100%	90%	**~98%**
N_CO	95%	100%	100%	100%	100%	100%	100%	100%	**~99%**
N_CO_L	100%	100%	100%	100%	100%	100%	100%	100%	**100%**
O.S.R.P	**95%**	**100%**	**100%**	**100%**	**100%**	**~99%**	**95%**	**~92%**	

The required time to perform the labware identification is in the range between (1 ~ 1.5 sec.). The labware recognition time can be decreased by removing the unwanted and unimportant features from the image. This has been performed by applying the HSV color filtering as a preprocessing step before using SURF algorithm. The lid mark has been designed with red color as shown in Fig. 6.37. The process starts with applying brightness correction with histogram equalization to the Kinect image to increases the color saturation. Then, the color segmentation starts with keeping the red colors in the Kinect image and removing all the rest colors. The final image is converted to grayscale by changing the red color to black with white background as shown in Fig. 6.38. Then, the features identification process is performed using SURF to recognize the required labware according to the related lid mark. This process requires about 0.5 sec. because most of the unwanted features have been removed from the image before applying the features identification step [203].

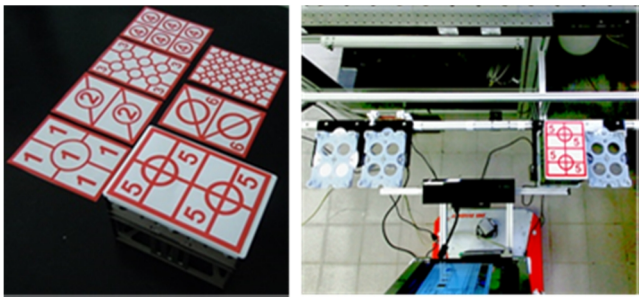


Figure 6.37: Labware mark with red features.

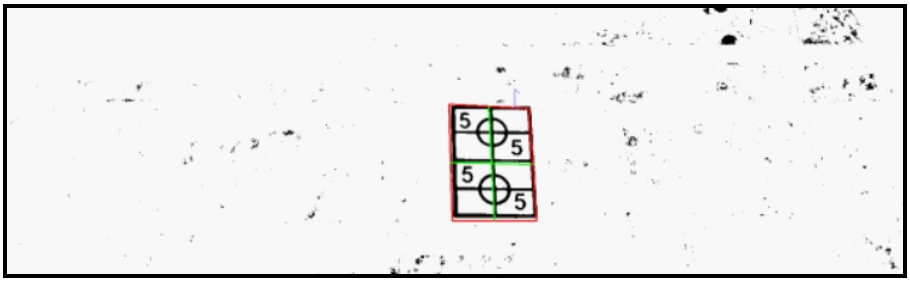


Figure 6.38: Lid mark identification using HSV filtering and SURF.

Fig. 6.39 shows the results of recognition tests using HSV segmentation and SURF algorithm. The chart shows the zero success rate after using the filters in the non-lighted environment. The best success rate (~96%) can be obtained with applying the histogram equalization with using the filters in the lighted environment. Tables 6.7 shows the results summary of this test.

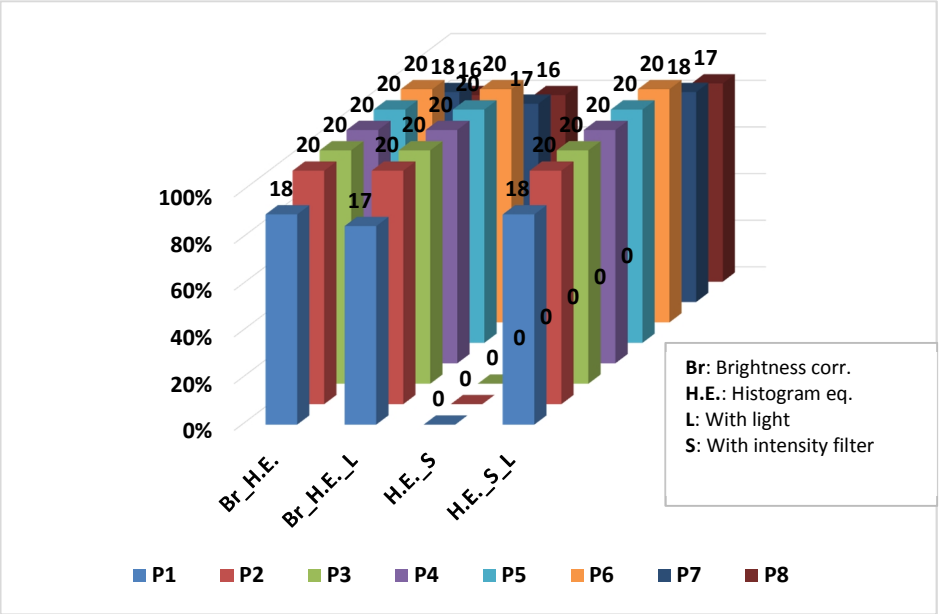


Figure 6.39: The identification success rate of the labware mark (HSV+SURF).

Table 6.7: The results summary of mark identification (HSV+SURF).

Test	P1	P2	P3	P4	P5	P6	P7	P8	O.S.R.T
Br_H.E.	90%	100%	100%	100%	100%	100%	90%	80%	**95%**
Br_H.E._L	85%	100%	100%	100%	100%	100%	85%	80%	**~94%**
H.E._S	0%	0%	0%	0%	0%	0%	0%	0%	**0%**
H.E._S_L	90%	100%	100%	100%	100%	100%	90%	85%	**~96%**
O.S.R.P	**~66%**	**75%**	**75%**	**75%**	**75%**	**75%**	**~66%**	**~61%**	

The success in labware manipulation and transportation depends significantly on the success of labware recognition with position estimation. According to the all previous results, the 100% success rate can be realized by using a lid mark with sufficient features printed on a coarse paper. Also, polarization filter with intensity filter are used with the Kinect V2 under sufficient lighting conditions. Furthermore, it has to be mentioned that in case of existing more than labware with the same lid mark, the most obvious top view will be selected. Table 6.8 summarizes the best results of the overall tests after applying the brightness and contrast correction on the raw image and under strong lighting condition.

Table 6.8: The overall summary of the best results.

Test case	Success rate	False pos. rate	Time
Without lid	21%	14%	1.4sec
With lid mark	~97%	0%	1.3sec
Mark with filters	100%	0%	1.1sec
Red mark with filters	~96%	0%	0.5sec

6.4.2 Multiple Labware Manipulation

The system of multiple labware manipulation and transportation consists of three main coding platforms which are the robot navigation control (RNC) [201][202], the robot arm control (RAC), and the labware identification control (LIC) using Kinect V2. These platforms are connected with each other through 2 client-server communication models (asynchronous socket). The first model is to connect RNC with the RAC. The second model is to connect the RAC with the LIC where the RAC has a client and a server platform at the same time as shown in Fig. 6.40. The RNC (client) sends the orders to the RAC (server) that include the required task (grasping or placing) with other information related to the required labware or placing location. The RAC (client) sends the target specifications to the LIC (server) for identification process. Then, the LIC sends feedback information about the target to the RAC to be processed. The RAC in turn sends the performance status to the RNC [200].

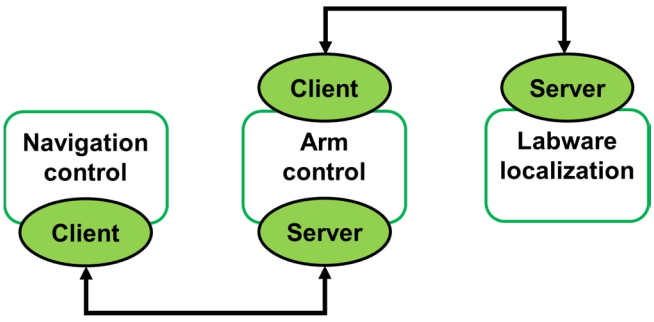


Figure 6.40: The architecture of client-server model for multipla labware manipulation.

The labware manipulation process starts when the H20 robot reaches the grasping location (workstation). The LIC receives the information about the labware which has to be grasped. The area of workstation in the Kinect image is cropped. Then, the features identification process is performed to find the required lid mark in the view. After the recognition step for the required area, the coordinates of the center point of this area in the Kinect image frame has to be found. Then, the position of this center point related to the Kinect is found using a mapping process. The required point is mapped from the image frame to the depth frame of

Kinect. Then, another step is performed to map the point from depth frame to the Kinect space coordinates. The position of the center point is used as reference for calculating the grasping or placing point positions where the arm end effector has to reach.

For the grasping task, the arm end effector has to reach the grasping point in the labware container as shown in Fig. 6.41 [200]. It means that the center point position of the labware lid has to be calibrated to find the grasping point position related to the Kinect sensor. Since, there are different kinds of labwares which have different height, it is not usable to depend on the labware height to find the grasping point position related to Kinect. In this case, it is crucial to depend on the workstation height as reference to find the grasping point height.

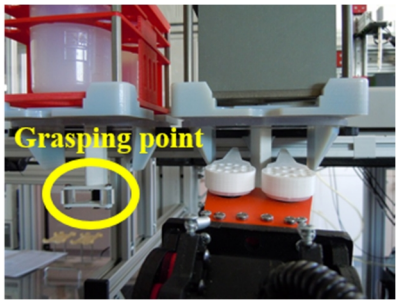


Figure 6.41: The grasping point of labware container.

After the grasping point position is found related to Kinect position, an extrinsic calibration is performed to calculate the grasping point position related to the arm base. Then, the inverse kinematic model is used to move the arm to the target and perform the task. The H20 arm puts the required labware on the H20 holder (see Fig. 6.42) to be transported safely to the location where it should be placed.

Figure 6.42: The H20 holder of labware container.

The labware position related to Kinect determines which arm has to be used for the required task. If the grasping task for the required labware is performed with a specific arm (right or left), then the placing task for the same labware has to be achieved using the same arm. It is not possible to perform the placing task with the other arm because the labware is positioned on the H20 holder according to the used arm. For this reason, it is very essential to perform the placing task at a position where the required arm can reach.

Since the H20 arms are unstable and they have weak wrist joints, it is very necessary to know the weight of the labware which has to be manipulated. The weight information helps to calibrate the wrist joint to keep the labware straight during the manipulation process to avoid the spilling its contents. Furthermore, pulling springs with suitable elasticity have been fixed on the wrist joint to help the arm in manipulating heavy labwares as shown in Fig. 6.43. The maximum payload, which can be handled using this strategy, is about 700g.

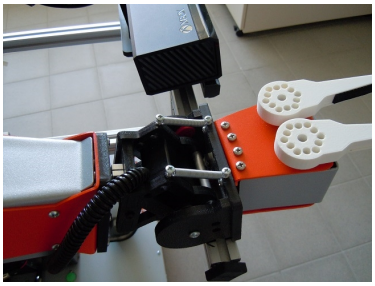


Figure 6.43: Pulling springs for the weak wrist joint.

In case that the required target is not existing in the Kinect view or it is out of the arms workspace, a feedback information is sent to the navigation control. This feedback information includes several decisions such as changing the robot position or skipping to the next task. The changing of robot position helps to get better view for the Kinect sensor and to make the target within the arms workspace. According to the arms workspace and to the robot position in front of the workstation, each arm can reach 2 labwares which are positioned alongside each other.

To achieve the placing tasks visually, the required holder on the workbench has to be found. Fig. 6.44 shows some empty holders where the placing task can be performed. It is clear that P1 and P8 holders are visually blocked in case of existing a labware beside them. This belongs to the view angle of Kinect. Also, the holders can't be identified and differentiated to find the desired one.

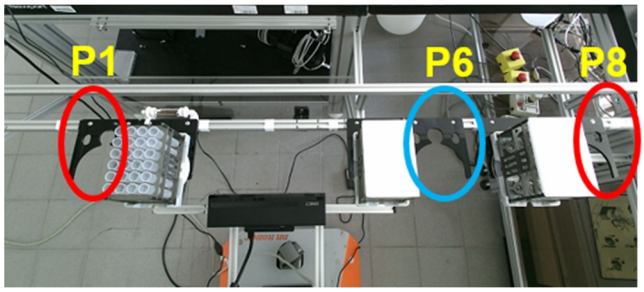


Figure 6.44: The workbench holders for placing tasks.

To cope with this issue, placing marks have been designed to be fixed in front of the holders as shown in Fig. 6.45. The placing marks are used for identification and position estimation of the required holder. Ten different placing mark can be identified. The labwares on the workbench don't block the placing marks to be seen by the Kinect sensor. The same strategy of identification for the labware lid mark can be used to recognize the required placing mark [200].

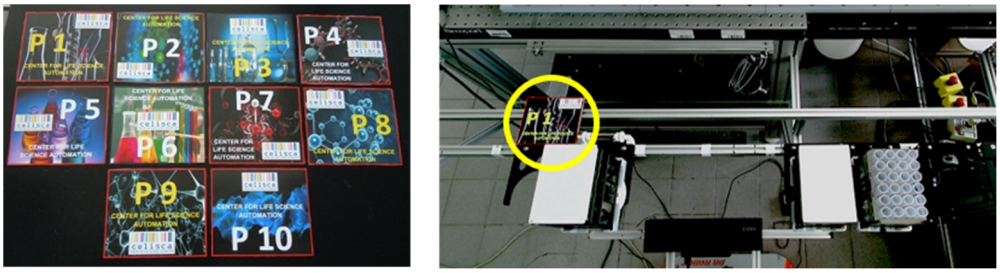


Figure 6.45: Placing marks for holders identification.

Each holder on the workbench are defined using a specific mark. But, it very essential and crucial to know if the holder is occupied by a labware or not. This has been done using a mechanical part where the placing mark can be affixed. The mechanical part has a rotational joint to show and hide the placing mark from the Kinect view to check if the holder is empty or not. In other words, if the Kinect can't see the placing mark of the required holder, it means that this holder is already occupied as shown in Fig. 6.46 [200].

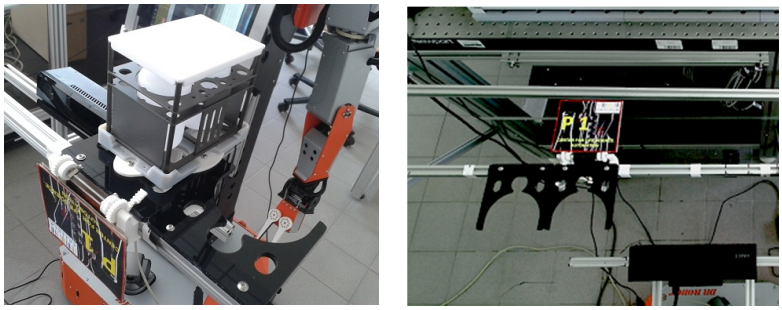


Figure 6.46: Checking the availibility of the required holder.

The mechanical part can be used just in case that the labware has enough weight to move the rotational joint. Since, the labwares have different weights, another idea has been applied. The idea depends on using a micro switch, micro servo motor, and servo controller to rotate the placing mark as shown in Fig. 6.47.

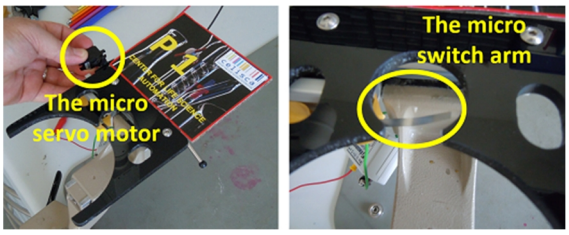


Figure 6.47: The micro switch and servo motor for placing procedure.

The grasping and placing tests have been performed for 50 times and Table 6.9 shows the overall success rate of these attempts. For the visual manipulation, the robot needs about 7 seconds before the starting of arm movement. The robot has to be stable and not trembling when it reaches the workstation and this requires about 3 seconds as delay time. The 4 seconds of time is for sockets communications, recognition, position calculation, and sending the order to the arm controller. The entire required time for performing the visual grasping is about 69 seconds including the recognition, localization, IK calculation, and arm movement. The required time for visual placing is about 59 sec. The work has been developed using Microsoft Visual Studio 2015 with C# programming language. The project is running on a Windows 10 platform in the H20 tablet.

Table 6.9: Manipulation tests of multiple labware.

Attempts	Successful grasp	Successful place
50	92%	90%

A blind strategy has been applied also for multiple labware manipulation using a sonar sensor as it is mentioned in section 5.4. This has been performed by identifying two positons for the robot in front of the workstation. The shift distance between these 2 positions of robot (P1 & P2) is 29 cm. At each position, the robot can manipulate 4 labwares where each arm can manipulate two of them as shown in Fig. 6.48. The blind strategy is defined by depending on the required robot position and required labware position according to right arm or left arm as follows: P1R1, P1R2, P1L1, P1L2, P2R1, P2R2, P2L1, and P2L2 [186]. The success rate of blind strategy depends significantly on the accuracy of robot position in front of the workstation. The required time for the blind grasping is about 62 seconds whereas the blind placing task needs about 52 seconds.

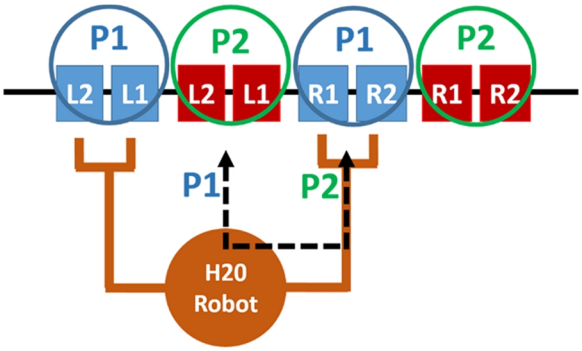


Figure 6.48: Blind manipulation for multiple labware.

6.5 Labware Orientation

The target pose can be found with a normal camera using coplanar POSIT algorithm which stands for POSe ITerations. It requires image coordinates of the desired object points (minimum 4 points). It also requires the real model coordinates of these points. It means that the target has to be previously known. Furthermore, the effective focal length of the used camera has to be known. Color, shape, edges, and corners detections have been used with coplanar POSIT algorithm to find the pose of the desired object as shown in Fig. 6.49.

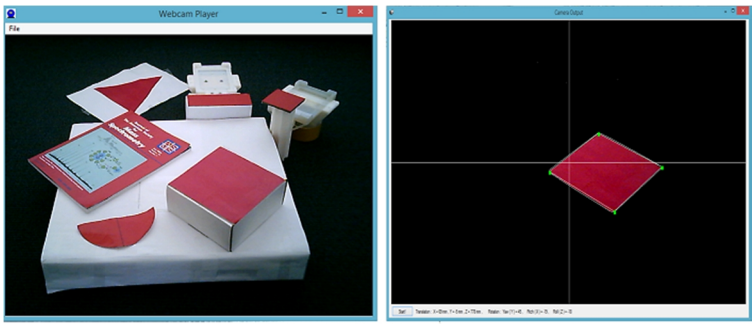


Figure 6.49: Red squre detection and pose estimation using POSIT.

POSIT algorithm has been performed also with the Kinect sensor to find the orientation of the labware related to the robot. The orientation angle is very important to correct the arm motion in case that the robot is not straight in front of the workstation. This has been done by finding the corners' positions of the lid mark in the image coordinates. Also, the physical corners coordinates related to the mark center point have to be calculated. This information is inserted to the POSIT algorithm to calculate the difference in orientation between the labware and the Kinect camera or the robot. The orientation angle represents the yaw angle of the robotic arm. Fig. 6.50 shows the corners of the lid mark which are used to calculate the orientation angle.

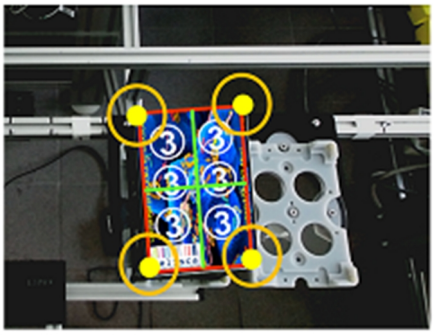


Figure 6.50: The corners of the lid mark.

6.6 Related Work

6.6.1 Pushing the Elevator Buttons

For labware transportation in multi-floor working environment, the robot has to move between the floors using the elevator. This requires pushing the elevator buttons with the robotic arms. Position based visual servoing algorithm with eye-on-hand method has been used. The F200 3D camera is used to find the position of the required buttons. Then, the IK model is used to guide the arm finger to the target according to look-and-move strategy. The first test has been done using the old version of gripper as shown in Fig. 6.51. Then, a new finger with base has been designed in a similar way with the final design of labware container to fit the arm gripper as shown in Fig. 6.52. Every time the robot reaches the elevator, the arm grasps the finger from the robot holder and guide it to be in a close position to the required button. The approximate positions of the elevator buttons related to the arm base have been saved in a database. Whenever the robot arrives the elevator and needs to push a specific button, the arm will grasp the finger and move towards the required button and stop nearby. This step makes the 3D hand camera close to the desired button for detection and localization purpose. The color detection and optical character recognition (OCR) algorithm are used to find the entry button and the buttons with floors numbers. The position of the required button related to the camera is found and calibrated to be related to the fingertip. Then, the arm moves to push the button using the inverse kinematic model according to the target position. Fig. 6.53 shows the sequence of pushing the entry button of the elevator. The connection between the arm controller and the hand camera has been done through client-server model.

Figure 6.51: Pushing the elevator entry button using F200 camera and old gripper.

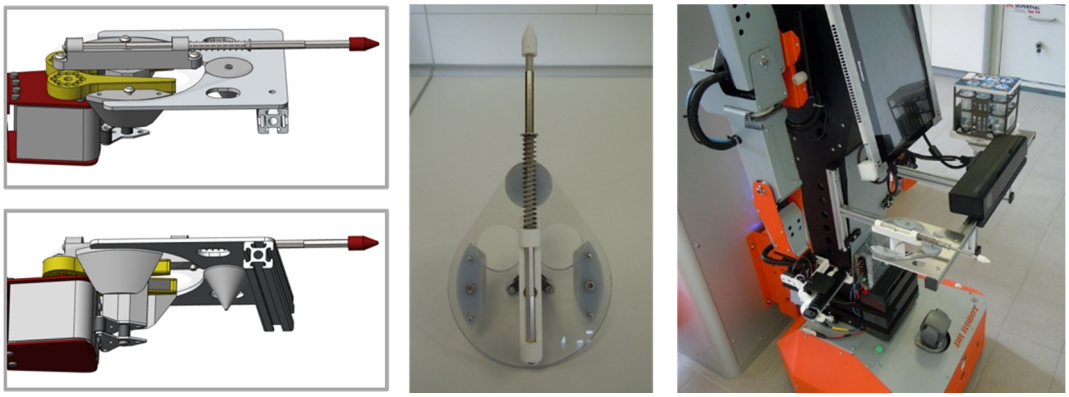


Figure 6.52: New finger design for pushing the elevator buttons.

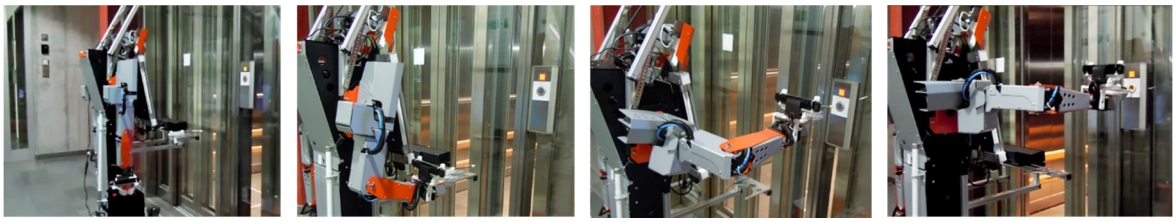


Figure 6.53: Sequence of pushing the elevator button.

6.6.2 Object Tracking using Head Control

A stereo vision is one of the common method that can be applied in many different applications such as tracking the required objects or tracking the human face in human-computer interaction task. This method requires that the target has to be seen by two cameras to estimate its depth and position information. It is important to be sure that the video quality, resolution, and contrast are identical in the both cameras. In stereo vision system, the two cameras have to be mounted very accurately to guarantee that all their axes are parallel. To perform the object tracking, the stereo vision system has to be mounted on a pan-tilt joints module. The H20 head has the two joints of pan-tilt module where the 2 cameras are fixed as shown in Fig. 6.54. The transformation matrix from the neck to the head is as follows [200]:

$$ {}^{N}_{H}T = {}^{N}_{H\,0}T \cdot {}^{H\,0}_{H\,1}T \cdot {}^{H\,1}_{H\,2}T \cdot {}^{H\,2}_{H}T = \begin{bmatrix} C_1C_2 & -S_1 & S_2C_1 & l_{H2}S_2C_1 \\ C_2S_1 & C_1 & S_1S_2 & l_{H2}S_1S_2 \\ -S_2 & 0 & C_1 & l_{H1}+l_{H2}C_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

The solution of inverse kinematic problem for the 2 joints of head is as follows:

$$ \theta_1 = \text{atan2}\left(S_y, -S_x\right). \tag{6.2} $$

$$ \theta_2 = \text{atan2}\left(n_x, a_x\right). \tag{6.3} $$

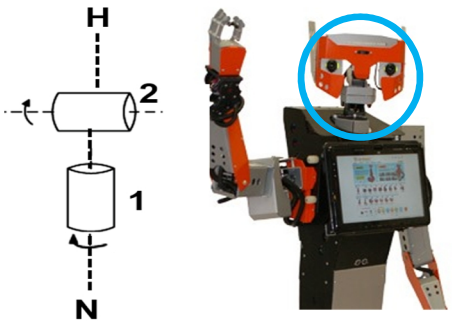


Figure 6.54: The joints structure of H20 head.

The cameras of H20 head are AXIS network cameras which is shown in Fig. 6.55. The AXIS camera offers both wireless (IEEE 802.11g) and wired connection to the network for flexible installation. It has a passive infrared sensor for detecting movement. It includes also a white LED for illuminating the scene. Furthermore, it offers two-way audio communication with integrated microphone and speaker. Related to the image properties, it provides multiple, individually configurable video streams in H.264 as well as Motion JPEG and MPEG-4 with very good quality at 30 frames per second in VGA resolution.

Figure 6.55: AXIS network camera.

The stereo cameras of H20 head have been used to perform the object tracking which is very essential to estimate the depth information of the view. Color, size, shape, edges, and corners detections have been used with the stereo cameras. Since the detected target is observed by the

two cameras, it will have different positions in the 2 images acquired from the 2 cameras. The difference in the target's coordinates is used to estimate its distance from the cameras.

To start tracking the object, its center point position in the 2 images coordinates has to be found. Then, the average of center point coordinates in the 2 images has to be calculated. Thus, the target center point will be right opposite to the middle point between the cameras. Fig. 6.56 shows the tracking of the detected object using the H20 head control. The target has been detected using RGB color, shape, and size information. The object tracking has been achieved also using HSV color segmentation with shape and area. Furthermore, SURF algorithm has been applied with each camera of H20 head to track the recognized object during its movements.

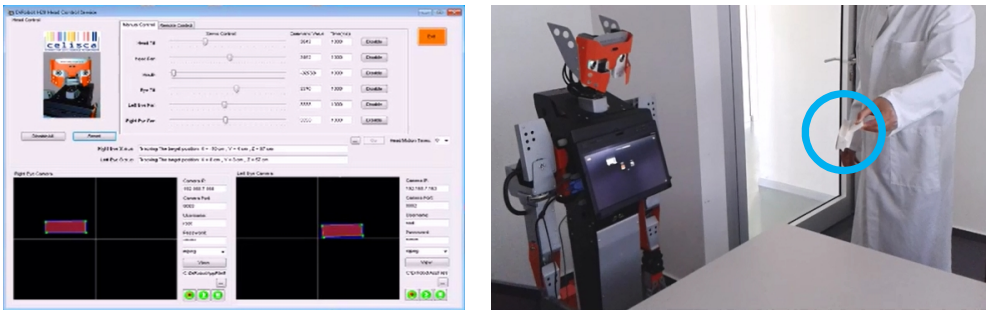


Figure 6.56: Color tracking with H20 head.

7 Summary and Outlook

7.1 Summary

Mobile robots are used to support efficient transportation and perform the required tasks safely, in addition, they increase productivity and save human resources. The activity of grasping an object and placing it at a required place is the goal that has to be achieved in life science laboratories. The grasping and placing tasks have to be performed very carefully since the mobile robots deal with lab ware containing chemical and/or biological components. Any kind of spilling or cross contamination has to be avoided. Normally, two types of sensors are considered as key for autonomous grasping and placing tasks: vision and tactile sensors. The ability of visual sensors to provide feedback for the robotic arm controller plays an important role in object manipulation and it is more suitable compared to other conventional feedback controls. The grasping and placing strategies with the arms of a H20 mobile robot have been performed to achieve a high precision and a safe transportation of the desired labware. This ability requires finding the pose of these objects with respect to the arm followed by using an accurate kinematic model to move it from one pose to another precisely and in a safe path. The solution of inverse kinematics for the robotic arm is one of the fundamental issues, which has to be found to realize high precision grasping and placing. The analytical solution can be found only for some robotic arms with particular structure and it is more accurate and complex at the same time in comparison with the numerical solution. Therefore, lately, all industrial robotic arms are designed to guarantee that the analytical solution can be obtained. The H20 robot has dual arms consisting of 6 revolute joints with 6-DOF. For each arm, the forward kinematics has been derived and the closed-form solution for the inverse kinematic problem with different cases of singularities has been found. A reverse decoupling mechanism method was used to solve the inverse kinematic problem analytically by viewing the arm kinematic chain in reverse order. The kinematics solution was validated using MATLAB with Robotics toolbox. The derived solution of the IK problem can be used for any other robotic arm which has the same joints structure and coordinate frames. A decision method was used to determine the optimal solution within multiple solutions of inverse kinematic depending on the joints' limits and minimum joints motion. The workspace analysis of the arm was found and simulated. The repeatability of the robotic arm has been tested using a laser sensor to find the accuracy of the arm movements. As an initial step for applying the kinematic model with the real H20 arms, a conversion process has been performed to convert the required angle values of the joints to the related servo motors positions. This step is very crucial since the H20 robot has weak and unstable arms where the effects of gravity, weight of arm parts, and payload increase the elasticity and the bending of each joint. A verification process was performed on the real H20 arms by applying a labware manipulation strategy using the kinematic solution and sonar

sensor to achieve the transportation tasks in real life science laboratories. Experimental results demonstrate that the arm end effector can reach the desired target within a precision of ±0.4 cm approximately. In service robot applications, the detection and pose calculation of the desired objects are very essential to manipulate them. Single colored objects have been detected and localized using a Kinect sensor V1 with a RGB color segmentation algorithm. The depth information is a challenge to achieve the object localization but Kinect gives good solutions to most of the related problems. Also, the Kinect V2 has been used to realize an intelligent manipulation for different objects in life science laboratories. The Kinect V2 can be considered one of the efficient sensors for performing object localization in a fast way. The H20 mobile robots has been equipped with Kinects V2 as visual sensors to recognize the required target and identify its pose related to the robot body. Different grippers and handles which are attached to a labware container are designed for this purpose. The design of the hand grippers, handles, and placing holder plays a very important to guarantee secure manipulation tasks. The detection and localization methods have been performed for single colored and textured targets related to the handle to be grasped and to the placing holder. The methods are based on HSV color segmentation and also on local features recognition with SURF (Speeded-Up Robust Features). The HSV color system is more robust against the light effects in comparison with the RGB system. Client-Server communication models have been developed to connect the navigation control with the arm controller which is in turn also connected with the Kinect control for visual processing. The client-server communication model is very useful to avoid the complexity of system integration which include multiple platforms. The initial style of the grippers was designed to grasp a handle attached to a labware container. The maximum payload which can be handled using this manner is about 350 g. This limited weight is related to the long lever arm of the wrist joint which needs more torque to handle heavier objects. Therefore, new grippers and labware containers have been designed to decrease the lift arm of the wrist joint by removing the handle for lifting heavier labware up to 700 g. This strategy requires to recognize the labware itself to be manipulated. Therefore, the Kinect sensor V2 with features matching algorithm SURF has been used. Some marks have been designed to be fixed on the labware lid for this reason. According to the Kinect view and workstation space, 10 different labware types with 10 different placing marks can be distinguished. Some preprocessing steps have been performed to the image before applying SURF method including HSV color segmentation, cropping, grayscale conversion, and histogram equalization. An extrinsic calibration has been applied to transform the labware position from the Kinect coordinates to the related arm shoulders coordinates. Additional part has been added to the workbench to recognize the required position for labware placing and to check whether it is empty. According to the arm workspace and to the robot position in front of the workstation, each arm can reach 2 labwares which are alongside each other. In case that the required labware is not within the

arm workspace, the robot has to change its position to reach it. The entire required time for performing the task is about 60~70 seconds. A blind manipulation method has been applied also for multiple labware manipulation depending on the holder positions and sonar sensor. The projects has been developed using Microsoft Visual Studio 2015 with C# programming language. The project is running on a Windows 10 platform in the H20 tablet.

7.2 Outlook

Different strategies and methods can be applied to improve the arm manipulation system. Several hardware and software changes can lead to realize better performance. The H20 robots have unstable arms because they have weak joints where the joints compliance causes positional errors. The effects of gravity, weight of arm parts, payload, and inertia cause the elasticity of each joint. One of the main technological limitations in the robotics industry is how to improve the accuracy by reducing the error between the robotic hand and the target. The precision depends on some elements such as the resolution of the control system, joint compliance, and the imprecision of the mechanical linkages and DC servo motors.

Feedback control is very important to improve the precision of reaching the end effector to the desired target. Force feedback or joint movement feedback are very important to realize accurate arms. Touch sensors can be fixed on the gripper to guarantee a secure manipulation.

Visual feedback using visual sensors is very helpful to improve the manipulation performance. Two approaches can be used for visual feedback, eye-on-hand and eye-to-hand for tracking the end effector till it reaches the target. Fig. 7.1 shows an example of using eye-to-hand approach to track the target and end effector at the same time.

Figure 7.1: Feedback tracking using the mark on the end effector.

Related to the eye-on-hand approach, the hand camera can be used as a visual feedback control to grasp the target more accurately. Specific marks can be fixed on the labware side (see Fig. 7.2) to be tracked to correct the pose error in end effector for the grasping task. Fig. 7.3 shows the architecture of feedback control using hand camera.

Figure 7.2: Fixing a mark on the labware side for tracking.

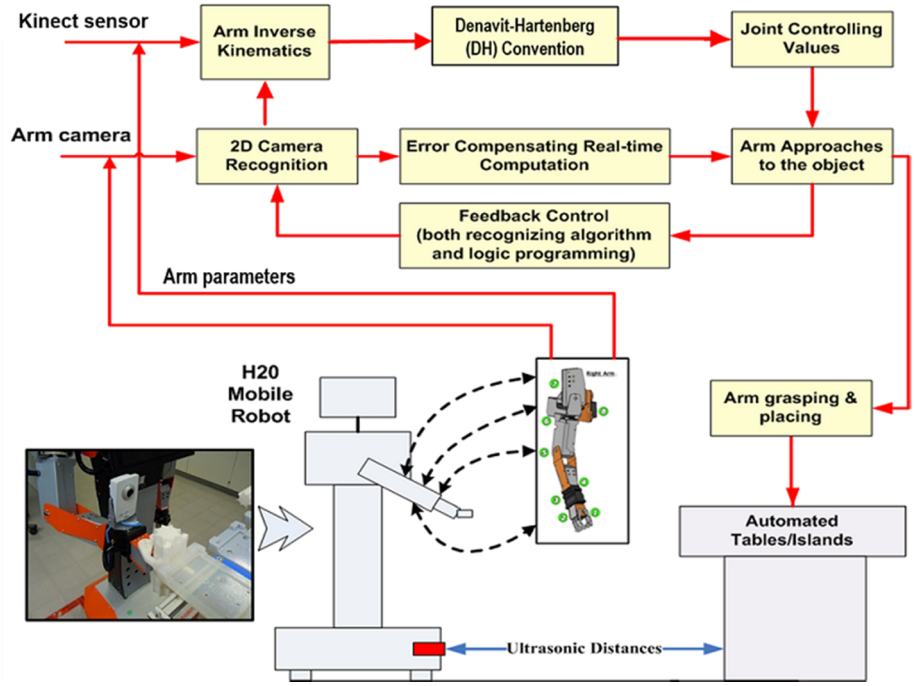


Figure 7.3: The architecture of feedback control using hand camera.

Related to the multiple labware identification system, several strategies can be applied to increase the number of the identified labware. 2D barcode recognition or character recognition can be used to differentiate multiple labware. Different 2D barcodes or characters can be fixed on the labware lid for identification.

8 Bibliography

[1] R. Siegwart and I. R. Nourbakhsh, Introduction to autonomous mobile robots. Cambridge, Mass: MIT Press, 2004.

[2] R. C. Luo, Y. C. Chou, and O. Chen, “Multisensor Fusion and Integration: Algorithms, Applications, and Future Research Directions”, in IEEE International Conference on Mechatronics and Automation, Heilongjiang, China, 2007, pp. 1986 -1991.

[3] G. Carbone, Grasping in Robotics. London, Springer, 2013.

[4] J. Bimbo, S. Rodriguez-Jimenez, H. Liu, X. Song, N. Burrus, L.D. Senerivatne, M. Abderrahim, and K. Althoefer, "Object pose estimation and tracking by fusing visual and tactile information", in IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Hamburg, Germany, 2012, pp. 65-70.

[5] De Ingenieur (2015, October 29). BETER NADENKEN OVER SOCIALE ROBOTS (1st ed.) [Online]. Available: https://www.deingenieur.nl/artikel/beter-nadenken-over-sociale-robots.

[6] J. S. Son, R. Howe, J. Wang and G. D. Hager, "Preliminary results on grasping with vision and touch", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 96), Osaka, Japan, 1996, pp. 1068-1075.

[7] S. Chitta, E. Jones, M. Ciocarlie, and K. Hsiao, “Mobile Manipulation in Unstructured Environments: Perception, Planning, and Execution”, in IEEE Robotics and Automation Magazine, vol. 19, no. 2, pp. 58–71, 2012.

[8] Introduction to Robot Geometry and Kinematics, lecture notes, V. Kumar.

[9] J. Xie, S. Yan, and W. Qiang, “A method for solving the inverse kinematics problem of 6-DOF space manipulator,” in International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA), Harbin. China, 2006, pp. 379-382.

[10] Laboratory of Autonomous Robotics and Artificial Life. (2015, February 15). Evolution of Object Manipulation Skills in Humanoid Robots [Online]. Available: http://laral.istc.cnr.it/res/manipulation/.

[11] J. R. Sanchez-Lopez, A. Marin-Hernandez, and E. R. Palacios-Hernandez, “Visual detection, tracking and pose estimation of a robotic arm end effector”, in the Proceeding of the Robotics Summer Meeting, Veracruz, Mexico, 2011, pp. 41–48.

[12] R. Y. Tsai, "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision", in IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, Florida, pp. 364-374, 1986.

[13] A. J. Sanchez and J. M. Martinez, "Robot-arm pick and place behavior programming system using visual perception", in International Conference on Pattern Recognition (ICPR), Barcelona, Spain, 2000, pp. 507-510.

[14] P. Corke, Robotics, Vision and Control, vol. 73. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[15] J. Borenstein, H. R. Everett, and L. Feng, “Where am I? Sensors and methods for mobile robot positioning”, University of Michigan, USA, vol. 119, no. 120, 1996.

[16] J. Borenstein, “The CLAPPER: A dual-drive mobile robot with internal correction of dead-reckoning errors”, in IEEE International Conference on Robotics and Automation, San Diego, CA, 1994, pp. 3085–3090.

[17] I. Lysenkov and V. Rabaud, “Pose estimation of rigid transparent objects in transparent clutter”, in IEEE Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 2013, pp. 162-169.

[18] H20 Robot. (2014, April 03). Dr Robot products [Online]. Available: http://www.drrobot.com/products_h20.asp.

[19] R700: VECTOR ROBOTIC ARM (2014, April 23). Assembled Robots [Online]. Available: http://www.globalspecialties.com/programmable-robotics/assembled-robots/item/780-vector-robotic-arm.html.

[20] Emerging Technologies: Mobile Robotics in Distribution Center Automation (2014, May 14). [Online]. Available: https://www.bastiansolutions.com/blog/index.php/2011/06/08/emerging-technologies-in-mobile-robotics-for-distribution-center-automation/#.WU1tITKGPcs.

[21] Dual-Arm Military Robot (2014, May 25). Robotics [Online]. Available: http://www.alliedmotion.com/applications/robot_parent_cat/index.html.

[22] Controls a robotic arm with your own arm! (2014, May 28). [Online]. Available: http://reset.etsii.upm.es/en/projects/robotic-arm/.

[23] R. Diankov, T. Kanade, and J. Kuffner, "Integrating grasp planning and visual feedback for reliable manipulation", in 9th IEEE-RAS Conference on Humanoid Robots, Paris, France, 2009, pp. 646-652.

[24] H. Wu, W. Tizzano, T. T. Andersen, N. A. Andersen, and O. Ravn, “Hand-Eye Calibration and Inverse Kinematics of Robot Arm using Neural Network”, in Robot Intelligence Technology and Applications 2, vol. 274, pp. 581–591, 2014.

[25] H.-Y. Chung, C.-C. Hou, Y.-S. Chen, and C.-L. Chao, “An intelligent service robot for transporting object”, in IEEE International Symposium on Industrial Electronics, Taipei, Taiwan, 2013, pp. 1–6.

[26] G. Lowe, "Object recognition from local scale-invariant features", in IEEE International Conference on Computer Vision, Corfu, Greece, 1999, pp. 1150-1157.

[27] A. Dhawan, A. Bhat, S. Sharma, and H. Kaura, “Automated Robot with Object Recognition and Handling Features”, International Journal of Electronics and Computer Science Engineering, vol. 2, no. 3, pp. 861–873, 2013.

[28] T. J. Tsay, M. S. Hsu, and R. X. Lin, “Development of a mobile robot for visually guided handling of material”, in IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan, 2003, pp. 3397–3402.

[29] Mitsubishi Robot. (2016, February 03). Mitsubishi RV-3AL industrial robot [Online]. Available:http://www.mitsubishirobot.com.

[30] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan, “Towards reliable grasping and manipulation in household environments”, in 12th International Symposium on Experimental Robotics (ISER), Springer Berlin Heidelberg, 2014, pp. 241–252.

[31] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: An open-source robot operating system”, in IEEE International Conference on Robotics and Automation (ICRA), Workshop on Open Source Software, Kobe, Japan, 2009, pp. 1–6.

[32] M. Dragusu, A. N. Mihalache, and R. Solea, “Practical applications for robotic arms using image processing”, in International Conference on System Theory, Control and Computing (ICSTCC), 2012, pp. 1–6.

[33] C. H. Chen, H. P. Huang and S. Y. Lo, "Stereo-based 3D localization for grasping known objects with a robotic arm system", in the World Congress on Intelligent Control and Automation (WCICA), Taipei, Taiwan, 2011, pp. 309–314.

[34] J. Stueckler, R. Steffens, D. Holz, and S. Behnke, “Real-Time 3D Perception and Efficient Grasp Planning for Everyday Manipulation Tasks”, in Proceedings of 5th European Conference on Mobile Robots (ECMR), Örebro, Sweden, 2011, pp. 177-182.

[35] T. Grundmann, R. Eidenberger, M. Schneider, M. Fiegert, and G. v Wichert, “Robust high precision 6D pose determination in complex environments for robotic manipulation”, in IEEE International Conference on Robotics and Automation, Alaska, 2010, pp. 1–6.

[36] K. Yamazaki, Y. Watanabe, K. Nagahama, K. Okada, and M. Inaba, “Recognition and manipulation integration for a daily assistive robot working on kitchen environments”, in IEEE International Conference on Robotics and Biomimetics, Tianjin, China, 2010, pp. 196–201.

[37] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, 2nd ed. Cambridge, UK , New York: Cambridge University Press, 2003.

[38] D. Berenson, S. S. Srinivasa, D. Ferguson, A. Collet, and J. J. Kuffner, “Manipulation planning with workspace goal regions”, in IEEE International Conference on Robotics and Automation (ICRA’09), 2009, pp. 618–624.

[39] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”, in Magazine of Association for Computing Machinery, vol. 24, no. 6, pp. 381–395, 1981.

[40] Y. Cheng, “Mean shift, mode seeking, and clustering”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pp. 790–799, 1995.

[41] A. Collet, M. Martinez, and S. Srinivasa, "The MOPED framework: Object recognition and pose estimation for manipulation", International Journal of Robotics Research, vol. 30, no. 10, pp. 1284 - 1306, 2011.

[42] L. T. Anh and J. B. Song, "Object tracking and visual servoing using features computed from local feature descriptor", in International Conference on Control Automation and Systems (ICCAS), Gyeonggi, South Korea, 2010, pp. 1044-1048.

[43] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features", Journal of Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359, 2008.

[44] R. Nishida and S. Kawamura, “A new feedback robot control method based on position/image sensor integration”, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 5012–5017.

[45] P. Azad, T. Asfour and R. Dillmann, “Stereo-based 6D object localization for grasping with humanoid robot systems”, in IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, California, 2007, pp. 919-924.

[46] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot modeling and control. New York, John Wiley & Sons, Inc., 2006.

[47] L-W. Tsai, Robot Analysis: The mechanics of serial and parallel manipulators. New Jersey, John Wiley and Sons Ltd, 1999.

[48] S. Sahu, B.B. Biswal and B. Subudhi, "A novel method for representing robot kinematics using quaternion theory", IEEE Sponsored Conference on Computational Intelligence, Control and Computer Vision in Robotics & Automation, Rourkela, India, 2008, pp. 76-82.

[49] Introduction to Homogeneous Transformation and Robot Kinematics, lecture notes by Jennifer Kay, Rowan University, Computer Science Department, 2005.

[50] Spherical wrist (2014, August 17). [Online]. Available: http://zalzala.org/IKMA/LinkedDocuments/RobotsHumans/sld021.htm.

[51] Y. H. Liu and H. Wang, "Adaptive visual servoing of robot manipulators", Advances in Robot Control, pp. 55-82, Springer Berlin Heidelberg, 2006.

[52] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices", Trans ASME Journal of Applied Mechanisms, vol. 22, pp. 215 -221, 1955.

[53] E. J. McCaffrey, "Kinematic analysis and evaluation of wheelchair mounted robotic arms", Graduate School Theses and Dissertations, 2003, [Online]. Available: http://scholarcommons.usf.edu/etd/1428.

[54] M. Uyguroglu. (2014, March 17). Denavit-Hartenberg Convention [Power point slides] [Online]. Available: http://kisi.deu.edu.tr/userweb/abdullah.secgin/Denavit-Hartenberg_Convention.ppt.

[55] G. S. Huang, C. K. Tung, H. C. Lin, and S. H. Hsiao, "Inverse kinematics analysis trajectory planning for a robot arm", in Asian Control Conference (ASCC), Kaohsiung, Taiwan, 2011, pp. 965-970.

[56] J. Iqbal, R. ul Islam, and H. Khan, "Modeling and Analysis of a 6 DOF Robotic Arm Manipulator", Canadian Journal on Electrical and Electronics Engineering, vol. 3, pp. 300-306, 2012.

[57] J. J. Craig, Introduction to Robotics Mechanics and Control. New Jersey, Pearson Education, Inc., 2005.

[58] D. Baker and C. Wampler, "On the Inverse Kinematics of Redundant Manipulators", International Journal of Robotics Research, Vol. 7, No. 2, pp. 3-21, 1988.

[59] The Kinematics Dynamics Library (KDL), [Online]. Available: http://www.orocos.org/kdl.

[60] R. P. Paul, B. E. Shimano, and G. Mayer, "Kinematic control equations for simple manipulators", IEEE Transactions on Systems, Man, and Cybernetics, vol. 11, pp. 449–455, 1981.

[61] R. M. Murray, Z. Li, S. S. Sastry. A mathematical introduction to robotic manipulation. Boca Raton, Florida, CRC Press, 1994.

[62] J. Zhao and N. I. Badler, "Inverse kinematics positioning using nonlinear programming for highly articulated figures", ACM Transactions on Graphics, vol. 13, pp. 313-336, 1994.

[63] S. R. Buss, "Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods", IEEE journal on robotics and automation, vol. 17, pp. 1–19, 2004.

[64] C. Man, X. Fan, C. Li and Z. Zhao, "Kinematics analysis based on screw theory of a humanoid robot", Journal of China University of Mining & Technology, Vol. 17 No. 1, pp. 49–52, 2007.

[65] C. G. S. Lee and M. Ziegler, “Geometric approach in solving inverse kinematics of PUMA robots”, IEEE Transactions on Aerospace and Electronic Systems, vol. 20, pp. 695-706, 1984.

[66] MEAM.Design : PUMA260 (2014, August 25). [Online]. Available: http://medesign.seas.upenn.edu/index.php/Guides/PUMA260.

[67] J. Billingsley and R. Bradbeer, Eds., Mechatronics and Machine Vision in Practice. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[68] L. Nie and Q. Huang, “Inverse kinematics for 6-DOF manipulator by the method of sequential retrieval”, in Proceedings of the International Conference on Mechanical Engineering and Material Science, Yangzhou, China, 2012, pp. 255–258.

[69] A. Aristidou and J. Lasenby, “FABRIK: A fast, iterative solver for the Inverse Kinematics problem”, Journal of Graphical Models, vol. 73, no. 5, pp. 243–260, 2011.

[70] T. Zhao, J. Yuan, M. Zhao, and D. Tan, “Research on the Kinematics and Dynamics of a 7-DOF Arm of Humanoid Robot”, in IEEE International Conference on Robotics and Biomimetics (ROBIO), Kunming, China, 2006, pp. 1553–1558.

[71] T. Ho, C. G. Kang, and S. Lee, “Efficient closed-form solution of inverse kinematics for a specific six-DOF arm”, International Journal of Control Systems and Automation, vol. 10, pp. 567–573, 2012.

[72] Y. Cui, P. Shi, and J. Hua, "Kinematics analysis and simulation of a 6-DOF humanoid robot manipulator", in International Asia Conference on Informatics in Control, Automation and Robotics (CAR), vol.2, pp.246-249, 2010.

[73] M. M. U. Atique and M. A. R. Ahad, “Inverse Kinematics solution for a 3DOF robotic structure using Denavit-Hartenberg Convention”, 2014, [Online]. Available: https://www.researchgate.net/publication/268239568_Inverse_Kinematics_solution_for_a_3DOF_robotic_structure_using_Denavit-Hartenberg_Convention.

[74] S. Q. Xie, E. Haemmerle, Y. Cheng and P. Gamage (2008). Vision-Guided Robot Control for 3D Object Recognition and Manipulation, Robot Manipulators, Marco Ceccarelli (Ed.), InTech, 2008, pp. 521-546.

[75] R. Yang and M. Pollefeys, “Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, 2003, pp. 211–218.

[76] L. Heng, B. Li, and M. Pollefeys, “Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry”, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013, pp. 1793–1800.

[77] R. Melo, M. Antunes, J. P. Barreto, G. Falcao, and N. Goncalves, “Unsupervised Intrinsic Calibration from a Single Frame Using a "Plumb-Line" Approach”, in IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 2013, pp. 537–544.

[78] R. Sagawa, and Y. Yagi, "Accurate calibration of intrinsic camera parameters by observing parallel light pairs", in IEEE International Conference on Robotics and Automation (ICRA), Pasadena, California, 2008, pp. 1390-1397.

[79] I. Lundberg, M. Bjorkman, and P. Ogren, “Intrinsic camera and hand-eye calibration for a robot vision system using a point marker”, in IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, 2014, pp. 59–66.

[80] E. Malis, “Hybrid vision-based robot control robust to large calibration errors on both intrinsic and extrinsic camera parameters”, in European Control Conference (ECC), Porto, Portugal, 2001, pp. 2898–2903.

[81] O. Marques, Practical Image and Video Processing Using MATLAB, John Wiley & Sons, Inc., 2011.

[82] [Online]. Available: http://www.imageprocessingplace.com.

[83] C. Theis, I. Iossifidis and A. Steinhage, “Image Processing Methods for Interactive Robot Control”, in IEEE International Workshop on Robot and Human Interactive Communication, Paris, France, 2001, pp. 424-429.

[84] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps”, International Journal on Robotics Research, vol. 34, pp. 705–724, 2015.

[85] B. Zitova, and J. Flusser, “Image registration methods: A survey”, Journal on Image and Vision Computing, vol. 21, pp. 977-1000, 2003.

[86] L. Lucchese and S.K. Mitra, “Color Image Segmentation: A State-of-the- Art Survey”, INSA-A Journal: Special Issue on Image Processing, Vision and Pattern Recognition, pp. 207-221, 2001.

[87] M. Ejiri, T. Miyatake, H. Sako, A. Nagasaka and S. Nagaya, “Evolution of Real-time Image Processing in Practical Applications”, in IAPR Workshop on Machine Vision Applications, Tokyo, Japan, 2000, pp. 177-186.

[88] RGB (Red Green Blue) (2014, October 30). [Online]. Available: http://colorizer.org/.

[89] C. Zhang and L. Huang, “Content-Based Image Retrieval Using Multiple Features,” Journal of Computing and Information Technology, vol. 22, no. LISS 2013, pp. 1–10, 2014.

[90] C. Junhua and L. Jing, “Research on Color Image Classification Based on HSV Color Space,” in International Conference on Instrumentation, Measurement, Computer, Communication and Control, Harbin City, China, 2012, pp. 944–947.

[91] HSV color space (2014, October 30). [Online]. Available: https://stackoverflow.com/questions/22588146/tracking-white-color-using-python-opencv.

[92] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd Edition, New Jersey, Prentice Hall, 2002.

[93] JPEGtool library (2015, October 01). [Online]. Available: http://pakuj.brek.sk/jpegtool/jpegtool.html.

[94] C. Harris and M.J. Stephens, “A combined corner and edge detector,” in Fourth Alvey Vision Conference, Manchester, UK, 1988, pp. 147–152.

[95] How to detect corner with specific angle degree (2015, June 07). [Online]. Available: https://stackoverflow.com/questions/15838555/how-to-detect-corner-with-specific-angle-degree.

[96] P. Hart, “How the Hough transform was invented [DSP History]”, IEEE Signal Processing Magazine, vol. 26, no. 6, pp. 18–22, 2009.

[97] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Key points”, International Journal of Computer Vision, Vol. 60, No. 2, pp. 91–110, 2004.

[98] W. Chen, Y. Zhao, W. Xie, and N. Sang, “An improved sift algorithm for image feature-matching,” in International Conference on Multimedia Technology, Hangzhou, China ,2011, pp. 197–200.

[99] E. Rosten, R. Porter, and T. Drummond, "Faster and Better: A Machine Learning Approach to Corner Detection", IEEE Transaction on Pattern Analysis Machine Intelligence, vol. 32, no. 1, pp. 105–119, 2010.

[100] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in European conference on Computer Vision, Graz, Austria, 2006, pp. 430–443.

[101] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in European Conference on Computer Vision, Crete, Greece, 2010, pp. 778–792.

[102] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564–2571.

[103] S. Obdrzalek and J. Matas, "Object Recognition using Local Affine Frames on Distinguished Regions," in British Machine Vision Conference, Cardiff, UK, 2002, pp. 113–122.

[104] S. Brahmbhatt, Real-time object detection in OpenCV using SURF (2015, November 11). [Online]. Available: http://robocv.blogspot.de/2012/02/real-time-object-detection-in-opencv.html.

[105] C. Choi and H. I. Christensen, "3D pose estimation of daily objects using an RGB-D camera," in IEEE International Conference on Intelligent Robots and Systems (IROS), Vilamoura-Algarve, Portugal, 2012, pp. 3342–3349.

[106] Kinect sensor (2014, March 11). [Online]. Available: https://www.slideshare.net/bhoomitmorkar/kinect-sensor.

[107] T. González Sánchez, "Artificial Vision in the Nao Humanoid Robot," Master thesis, Rovira i Virgili University, Tarragona, Spain, 2009.

[108] C. Wöhler, "Three-Dimensional Pose Estimation and Segmentation Methods," 3D Computer Vision, London, Springer-Verlag London, pp. 89–137, 2013.

[109] M. Zhu, K.G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce and K. Daniilidis, "Single Image 3D Object Detection and Pose Estimation for Grasping", in IEEE International Conference on Robotics & Automation, Hong Kong, China, 2014, pp. 3936- 3943.

[110] Z. Ou, W. Liu, and J. Su, "A bilinear model based solution to object pose estimation with monocular vision for grasping," in International Conference on Mechatronics and Automation, Beijing, China, 2011, pp. 501–506.

[111] J. Hernandez, A. Quintanilla, J. Lopez, F. Rangel, M. Ibarra, and D. Almanza, "Detecting objects using color and depth segmentation with Kinect sensor", in the Iberoamerican Conference on Electronics Engineering and Computer Science, Guadalajara, Mexico, 2012, pp. 196-204.

[112] F. Paniagua, "Object Recognition using the Kinect,", Master thesis, Royal Institute of Technology, School of Computer Science and Communication, Stockholm, Sweden, 2011.

[113] A. Khan, F. Moideen, J. Lopez, W. L. Khoo and a. Z. Zhu, "KinDectect: Kinect Detecting Objects," in International Conference on Computers for Handicapped Persons,: Linz, Austria, 2012, pp. 588-595.

[114] S. Izadi, R. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. Davison, and A. Fitzgibbon, "Kinectfusion: real-time dynamic 3D surface

reconstruction and interaction", in International Conference and Exhibition on Computer Graphics and Interactive Techniques, Vancouver, Canada, 2011, Article No. 23.

[115] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 2011, pp. 1228–1234.

[116] G. Bradski and A. Kaehler, "Learning OpenCV", California, O'Reilly Media Inc., 2008.

[117] L. Shapiro, Computer vision lectures (2015, July 27). [Online]. Available: https://courses.cs.washington.edu/courses/cse455/.

[118] R. Smeenk, Live Kinect holography experiment (2014, May 19). [Online]. Available: http://smeenk.com/live-kinect-holography/.

[119] G. Gerig, Structured Lighting slides (2014, May 19). [Online]. Available: http://www.sci.utah.edu/~gerig/CS6320-S2012/Materials/CS6320-CV-S2012-StructuredLight.pdf.

[120] Rob Miles, "Start.Here.Learn.the.Kinect API", California, O'Reilly Media, Inc., 2012.

[121] Kinect With Night shot (2014, May 11). [Online]. Available: http://graphics.stanford.edu/~mdfisher/Kinect.html.

[122] Intel RealSense 3D Camera F200. (2016, February 05). [Online].

Available: https://software.intel.com/en-us/realsense/home.

[123] Kinect for Windows Team, Kinect v2 sensor (2014, July 23). [Online]. Available: https://blogs.msdn.microsoft.com/kinectforwindows/2014/03/27/revealing-kinect-for-windows-v2-hardware/.

[124] S. K. Nayar, Y. Nakagawa, "Shape from Focus", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, pp 824 - 831, 1994.

[125] C. Choi, S.-M. Baek, and S. Lee, "Real-time 3D object pose estimation and tracking for natural landmark based visual servo", in IEEE International Conference on Intelligent Robots and Systems, Nice, France, 2008, pp. 3983–3989.

[126] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code", Computer Vision and Image Understanding, Vol. 15, pp. 123–141, 1995.

[127] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, "Iterative Pose Estimation Using Coplanar Feature Points", Computer Vision and Image Understanding, Vol. 63, pp. 495–511, 1996.

[128] B. Siciliano and O. Khatib, "Springer handbook of robotics", Berlin, Springer, 2008.

[129] K. HASHIMOTO, "A review on vision-based control of robot manipulators", Advanced Robotics, Vol. 17, pp. 969–991, 2003.

[130] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control", IEEE Transactions on Robotics and Automation, Vol. 12, pp. 651–670, 1996.

[131] H.-C. Cho, J.-K. Min, and J.-B. Song, "Hybrid position and force control of a robot arm equipped with joint torque sensors", in International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Korea, 2013, pp. 577–579.

[132] K. Kosuge, H. Takeuchi and K. Furuta, "Motion control of a robot arm using joint torque sensors", IEEE Transactions on Robotics and Automation, Vol. 6, pp. 258 - 263, 1990.

[133] A. Parmiggiani, M. Randazzo, L. Natale, G. Metta, and G. Sandini, "Joint torque sensing for the upper-body of the iCub humanoid robot", in IEEE International Conference on Humanoid Robots, Paris, France, 2009, pp. 15–20.

[134] K. Hashimoto and T. Noritsugu, "Performance and sensitivity in visual servoing", in IEEE International Conference on Robotics and Automation, Lueven, Belgium, 1998, pp. 2321-2326.

[135] W. J. Wilson, C. C. Williams Hulls and G. S. Bell, "Relative end-effector control using Cartesian position based visual servoing", IEEE Transactions on Robotics and Automation, Vol. 12, pp. 684-696, 1996.

[136] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches", IEEE Robotics & Automation Magazine, Vol. 13, pp. 82-90, 2006.

[137] G. Flandin, F. Chaumette and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing", in IEEE International Conference on Robotics and Automation, California, USA, 2000, pp. 2741-2746.

[138] G. Chesi and K. Hashimoto, "Static-eye against hand-eye visual servoing", in IEEE Conference on Decision and Control, Nevada, USA, 2002, pp. 2854-2859.

[139] P. Martinet and E. Cervera, "Stacking Jacobians properly in stereo visual servoing", in IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 717-722.

[140] G. D. Hager, W. C. Chang and A. S. Morse, "Robot hand-eye coordination based on stereo vision", IEEE Control Systems, Vol. 15, pp. 30-39, 1995.

[141] F. Alkhalil, L. Barbe, and C. Doignon, “Visual servoing of an articulated object based on stereovision", in IEEE International Symposium on Robotic and Sensors Environments, Montreal, Canada, 2011, pp. 71–76.

[142] W. Guo-Qing, K. Arbter and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation", IEEE Engineering in Medicine and Biology Magazine Engineering in Medicine and Biology Magazine, Vol. 16, pp. 40-45, 1997.

[143] A. Krupa, J. Gangloff, C. Doignon, M.F. de Mathelin, G. Morel, J. Leroy, L. Soler, J. Marescaux, “Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing”, IEEE Transactions on Robotics and Automation, Vol. 19, pp. 842–853, 2003.

[144] C. Staub, A. Knoll, T. Osa, and R. Bauernschmitt, “Autonomous High Precision Positioning of Surgical Instruments in Robot-Assisted Minimally Invasive Surgery under Visual Guidance”, in International Conference on Autonomic and Autonomous Systems, Cancun, Mexico, 2010, pp. 64–69.

[145] P. Hynes, G. I. Dodds, and A. J. Wilkinson, “Uncalibrated visual-servoing of a dual-arm robot for surgical tasks”, in IEEE International Symposium on Computational Intelligence in Robotics and Automation, California, USA, 2005, pp. 151–156.

[146] R. Wen, L. Yang, C.-K. Chui, K.-B. Lim, and S. Chang, “Intraoperative visual guidance and control interface for augmented reality robotic surgery”, in IEEE International Conference on Control and Automation, Xiamen, China, 2010, pp. 947–952.

[147] S. Lv, D. Zhang, and J. Gu, “Research of automatic needle locating based on stereo visual servoing”, in International Conference on Biomedical Engineering and Informatics , Yantai, China , 2010, pp. 1779–1783.

[148] Y. Shi, B. Liang, X. Wang, W. Xu and H. Liu S. Ye, et al., "Study on intelligent visual servoing of space robot for cooperative target capturing", in International Conference on Information and Automation, Shenyang, China, 2012, pp. 733-738.

[149] A. H. Abdul Hafez, V. V. Anurag, S. V. Shah, K. M. Krishna, and C. V. Jawahar, “Reactionless visual servoing of a dual-arm space robot”, in IEEE International Conference on Robotics and Automation, Hong Kong, China, 2014, pp. 4475–4480.

[150] M. Jin, H. Yang, Z. Xie, K. Sun, and H. Liu, “The ground-based verification system of visual servoing control for a space robot”, in IEEE International Conference on Mechatronics and Automation, Kagawa, Japan, 2013, pp. 1566–1570.

[151] Y. Shi, B. Liang, X. Wang, W. Xu, and H. Liu, “Modeling and simulation of space robot visual servoing for autonomous target capturing”, in International Conference on Mechatronics and Automation, Chendu, China, 2012, pp. 2275–2280.

[152] G. Zhang, B. Wang, L. Wang, and H. Liu, “A hybrid visual servoing control of 4 DOFs space robot”, in International Conference on Mechatronics and Automation, Changchun, China, 2009, pp. 3287–3292.

[153] M. Prats, P. J. Sanz, and A. P. del Pobil, "Vision-tactileforce integration and robot physical interaction", in IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009, pp. 3975–3980.

[154] K. Hsiao, S. Chitta, M. Ciocarlie and E. G. Jones "Contact-reactive grasping of objects with partial shape information", in International Conference on Intelligent Robots and Systems. Taipei, Taiwan, 2010, pp. 1228-1235.

[155] D. Kruse, J. T. Wen, and R. J. Radke, “A Sensor-Based Dual-Arm Tele-Robotic System”, IEEE Transactions on Automation Science and Engineering, Vol. 12, pp. 4–18, 2015.

[156] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao, “Perception, planning, and execution for mobile manipulation in unstructured environments”, IEEE Robotics & Automation Magazine, Vol. 19, pp. 58–71, 2012.

[157] H20 Quick Start Guide (user manual), Dr Robot Inc., Ontario, Canada, 2001-2015.

[158] H. Liu, N. Stoll, S. Junginger, and K. Thurow, “A common wireless remote control system for mobile robots in laboratory”, in IEEE Instrumentation and Measurement Technology Conference (I2MTC), Graz, Austria, 2012, pp. 688–693.

[159] Localization system StarGazer for Intelligent Robots - User’s Guide, Hagisonic Co., LTD., Daejeon, Korea, Ver. 4, 2016.

[160] D. L. Pieper, “The kinematics of manipulators under computer control”, Ph.D. Dissertation, California, Stanford University, 1968.

[161] N. I. Badler, C. B. Phillips, and B. L. Webber, "Simulating Humans: Computer Graphics Animation and Control", New York, Oxford, Oxford University Press, 1993.

[162] D. Kee and W. Karwowski, "Analytically derived three-dimensional reach volumes based on multi joint movements", The Journal of the Human Factors and Ergonomics Society, Vol. 44, pp. 530–544, 2002.

[163] P. J. McKerrow, "Introduction to Robotics", Sydney, Wesley, 1991.

[164] D. Bertram, J. Kuffner, R. Dillmann, and T. Asfour, “An integrated approach to inverse kinematics and path planning for redundant manipulators”, in IEEE International Conference on Robotics and Automation, Florida, USA, 2006, pp. 1874–1879.

[165] S. LaValle and J. Kuffner, "Randomized kinodynamic planning", International Journal of Robotics Research, Vol. 20, pp. 378–400, 2001.

[166] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, “Humanoid motion planning for dual-arm manipulation and re-grasping tasks”, in IEEE International Conference on Intelligent Robots and Systems, Missouri, USA, 2009, pp. 2464–2470.

[167] M. V. Weghe, D. Ferguson, and S. Srinivasa, "Randomized path planning for redundant manipulators without inverse kinematics", in IEEE International Conference on Humanoid Robots, Pennsylvania, USA, 2007, pp. 477 - 482.

[168] G. Welch and G. Bishop, "An Introduction to the Kalman Filter", Chapel Hill, University of North Carolina at Chapel Hill, 1995.

[169] S. J. Kim, J. H. Lee and J. M. Lee "Trajectory estimation of a moving object using Kalman filter and Kohonen networks", Robotica, Vol. 25, pp. 567-574, 2007.

[170] J. M. Hollerbach, "A Survey of Kinematic Calibration", Cambridge, the Robotics Review 1, Pages 207 - 242. 1989.

[171] J. V. D. Weijer and C. Schmid, "Coloring Local Feature Extraction", Computer Vision, Vol. 3952, pp. 334-348, 2006.

[172] P. Ganesan, V. Rajini, B. S. Sathish, and K. B. Shaik, “HSV color space based segmentation of region of interest in satellite images”, in International Conference on Control, Instrumentation, Communication and Computational Technologies, Kanyakumari District, India, 2014, pp. 101–105.

[173] L. Shuhua and G. Gaizhi, “The application of improved HSV color space model in image processing”, in International Conference on Future Computer and Communication, Wuhan, China, 2010, pp. 10-13.

[174] A. Gupta, "Grasping Known Objects with Aldebaran Nao", Department of Computer Science and Engineering, IIT Kanpur, India, 2014.

[175] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, “Automatic grasp planning using shape primitives”, in IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003, pp. 1824–1829.

[176] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, pp. 898– 916, 2011.

[177] J. Shi and C. Tomasi, "Good features to track", in IEEE Conference on Computer Vision and Pattern Recognition, Washington, USA, 1994, pp. 593-600.

[178] M. Ciocarlie, C. Pantofaru, K. Hsiao, G. Bradski, P. Brook, and E. Dreyfuss, "A side of data with my robot: Three datasets for mobile manipulation in human environments", IEEE Robotics & Automation Magazine, Vol. 18, pp. 44–57, 2011.

[179] Kinect for Windows, Human Interface Guidelines. Microsoft Corporation, Washington, USA. 2013.

[180] K. f. W. Team. (2012, January 20). Near Mode: What it is (and isn’t), Kinect for windows blog. [Online]. Available: https://blogs.msdn.microsoft.com/kinectforwindows/2012/01/20/near-mode-what-it-is-and-isnt/.

[181] F. Chaumette, "Potential problems of stability and convergence in image-based and position based visual servoing", the conference of vision and control, Vol. 237, pp. 66-78, 1998.

[182] J. A. Piepmeier, B. A. Gumpert, and H. Lipkin, "Uncalibrated eye-in-hand visual servoing", in IEEE International Conference on Robotics and Automation, Washington, D.C., USA, 2002, pp. 568-573.

[183] E. Cervera, F. Berry and P. Martinet, "Is 3D useful in stereo visual control?", in IEEE International Conference on Robotics and Automation, Washington, D.C., USA, 2002, pp. 1630-1635.

[184] D.-J. Kim, R. Lovelett, and A. Behal, "Eye-in-hand stereo visual servoing of an assistive robot arm in unstructured environments", in IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009, pp. 2326-2331.

[185] M. Prats, P. Martinet, A. P. del Pobil and S. Lee, "Robotic execution of everyday tasks by means of external vision/force control", Intelligent Service Robotics, Vol. 3, pp. 1-13. 2008.

[186] M. M. Ali, H. Liu, N. Stoll, and K. Thurow, "Kinematic Analysis OF 6-DOF Arms for H20 Mobile Robots and Labware Manipulation for Transportation in Life Science Labs", Journal of Automation, Mobile Robotics & Intelligent Systems, vol. 10, no. 4, pp. 40–52, 2016.

[187] M. A. Ali, H. A. Park, and C. G. Lee, "Closed-form inverse kinematic joint solution for humanoid robots", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010, pp. 704–709.

[188] R. O'Flaherty, P. Vieira, M. Grey, P. Oh, A. Bobick, M. Egerstedt, and M. Stilman, "Kinematics and Inverse Kinematics for the Humanoid Robot HUBO2", Georgia Institute of Technology, Atlanta, GA, USA, Technical Report, 2013.

[189] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "A floyd-genetic algorithm based path planning system for mobile robots in laboratory automation", in IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 2012, pp. 1550–1555.

[190] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "A Floyd-Dijkstra hybrid application for mobile robot path planning in life science automation", in IEEE International Conference on Automation Science and Engineering, Seoul, Korea, 2012, pp. 279–284.

[191] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "A fast method for mobile robot transportation in life science automation", in IEEE Instrumentation and Measurement Technology Conference (I2MTC), Minneapolis, MN, USA, 2013, pp. 238– 242.

[192] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "Mobile Robot for Life Science Automation", International Journal of Advanced Robotic Systems, Vol. 10, pp. 1-14, 2013.

[193] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "An application of charging management for mobile robot transportation in laboratory environments", in IEEE Conference on Instrumentation and Measurement Technology, Minneapolis, USA, 2013, pp. 435–439.

[194] K. S.Chong, and L. Kleeman, "Mobile robot map building for an advanced sonar array and accurate odometry", International Journal of Robotics Research, vol. 18, no.1, pp. 20–36, 1999.

[195] M. M. Ali, H. Liu, R. Stoll, and K. Thurow, "Arm grasping for mobile robot transportation using Kinect sensor and kinematic analysis", in IEEE International Conference on Instrumentation and Measurement Technology (I2MTC), Pisa, Italy, 2015, pp. 516–521.

[196] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "A new method for mobile robot arm blind grasping using ultrasonic sensors and Artificial Neural Networks", in IEEE

Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, 2013, pp. 1360-1364.

[197] H. Liu, N. Stoll, S. Junginger, and K. Thurow, "A Fast Approach to Arm Blind Grasping and Placing for Mobile Robot Transportation in Laboratories", International Journal of Advanced Robotic Systems, vol. 11, pp. 1-12, 2014.

[198] K. L. Conrad, P. S. Shiakolas, and T. C. Yih, "Robotic calibration issues: Accuracy, repeatability and calibration", in the 8th Mediterranean Conference on Control and Automation (MED2000), Patras, Greece, 2000.

[199] M. M. Ali, H. Liu, N. Stoll, and K. Thurow, "Intelligent Arm Manipulation System in Life Science Labs Using H20 Mobile Robot and Kinect Sensor", in IEEE International Conference on Intelligent Systems (IS'16), Sofia, Bulgaria, 2016, pp. 382-387.

[200] M. M. Ali, H. Liu, N. Stoll, and K. Thurow, "Multiple Lab Ware Manipulation in Life Science Laboratories using Mobile Robots", in IEEE International Conference on Mechatronics, Prague, Czech Republic, 2016, pp. 415-421.

[201] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A New Robust Method for Mobile Robot Multifloor Navigation in Distributed Life Science Laboratories", J. Control Sci. Eng., vol. 2016, Jul. 2016.

[202] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A Backbone-Floyd Hybrid Path Planning Method for Mobile Robot Transportation in Multi-Floor Life Science Laboratories", in IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Baden-Baden, Germany, 2016, pp. 406-411.

[203] M. M. Ali, H. Liu, N. Stoll, and K. Thurow, "An Identification and localization Approach of Different Labware for Mobile Robot Transportation in Life Science laboratories", in IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, 2016, pp. 353-358.

Declaration of Education Honesty

I declare that up to now I have processed no dissertation procedure and I have not applied for a dissertation procedure before.
This dissertation was never submitted to any other University or College.
Furthermore, I declare that this dissertation “High Precision Grasping and Placing for Mobile Robots” was created single-handed by me. I have used no other sources and utilities than named in the thesis. Content, literally or with regard to, is marked as citations.
This dissertation was never used for another academic qualification.

Mohammed Myasar Ali

Rostock, January 2017

List of publications

- **Mohammed M. Ali**, H. Liu, N. Stoll, and K. Thurow, "Kinematic Analysis OF 6-DOF Arms for H20 Mobile Robots and Labware Manipulation for Transportation in Life Science Labs", Journal of Automation, Mobile Robotics & Intelligent Systems, vol. 10, no. 4, pp. 40–52, 2016.

- **Mohammed M. Ali**, H. Liu, R. Stoll, and K. Thurow, "Arm grasping for mobile robot transportation using Kinect sensor and kinematic analysis", proceedings, IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Pisa, Italy, 2015, pp. 516–521.

- **Mohammed M. Ali**, H. Liu, N. Stoll, and K. Thurow, "Intelligent Arm Manipulation System in Life Science Labs Using H20 Mobile Robot and Kinect Sensor", proceedings, IEEE International Conference on Intelligent Systems (IS'16), Sofia, Bulgaria, 2016, pp. 382-387.

- **Mohammed M. Ali**, H. Liu, N. Stoll, and K. Thurow, "An Identification and localization Approach of Different Labware for Mobile Robot Transportation in Life Science laboratories", proceedings, IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, 2016, pp. 353-358.

- **Mohammed M. Ali**, H. Liu, N. Stoll, and K. Thurow, "Multiple Lab Ware Manipulation in Life Science Laboratories using Mobile Robots", proceedings, IEEE International Conference on Mechatronics, Prague, Czech Republic, 2016, pp. 415-421.

- **Mohammed M. Ali**, H. Liu, N. Stoll, and K. Thurow, "Recognition and Position Estimation for Multiple Labware Transportation Using Kinect V2 and Mobile Robots", Journal of Advances in Science, Technology and Engineering Systems, 2017, Accepted.

- **Mohammed M. Ali**, H. Liu, N. Stoll, and K. Thurow, "Grasping and Placing Operation for Labware Transportation in Life Science Laboratories using Mobile Robots", Journal of Advances in Science, Technology and Engineering Systems, 2017, Accepted.

- Mohammed Myasar Ali: „High-Precision Grasping and Placing for Mobile Robotics Arm". Poster, Statusseminar 2014, Nachwuchsgruppe „Life Science Automation – Systems and Process Technologies". Rostock (D), 02.07.2014.

- Mohammed Myasar Ali: „Grasping and Placing – High Performance Features for Mobile Robots". Poster, Statusseminar 2016, Nachwuchsgruppe „Life Science Automation – Systems and Process Technologies". Rostock (D), 28.11.2016.

Theses

1. Mobile robots based intelligent control systems can be used for multiple labware manipulation and transportation to connect all the workstations in laboratory environments.

2. The labware manipulation system using mobile robots can be integrated to the whole process of the laboratory automation. This integration leads to save the human resources and to increase the productivity by ensuring 24/7 operation.

3. The grasping and placing tasks are required to realize the labware manipulation control. These tasks have to be performed very carefully because the labware contain chemical and/or biological components. Any kind of spilling or cross contamination has to be avoided.

4. The H20 robots are used for labware manipulation and transportation. They are wireless networked humanoid mobile robots. Each one has dual arms with six revolute joints.

5. The design of arm grippers and labware containers are very important to guarantee secure grasping and placing tasks for the required labware.

6. The ability of objects manipulation requires finding the pose of these objects followed by using an accurate kinematic model for the robotic arm to guide it to the target.

7. The inverse kinematic model (IK) is required to convert the 3D Cartesian coordinates of the target to joint coordinates of the robotic arm.

8. The analytical solution of inverse kinematic problem for H20 arms has been found. The kinematics solution has been validated and the arm workspace has been simulated using MATLAB with Robotics toolbox.

9. A blind strategy for multiple labware manipulation using the kinematic model and sonar sensors can be used to achieve the transportation tasks in life science laboratories.

10. An intelligent visual manipulation strategy is developed using the Kinect sensor. The Kinect is a 3D camera that can directly provide the depth information of the view.

11. Single colored objects are detected and localized using the Kinect sensor. The object features (color, shape, and area) can be used for detection process.

12. Different color systems can be used for detection, like RGB (Red, Green, and Blue) and HSV (Hue, Saturation, and Value). The HSV system is more robust against the light effects.

13. The recognition and localization process can be performed also for textured objects based on local features recognition with SURF algorithm (Speeded-Up Robust Features).

14. SURF algorithm is applied for multiple labware identification using the Kinect. Some marks can be designed to be fixed on the labware lid to improve the identification process.

15. The identification and localization processes are performed for the labware (for grasping task) and for the labware holder on the workstation (for placing task). Ten different labware types with ten different placing holders can be distinguished.

16. Some preprocessing steps are necessary to be applied on the image before applying SURF including cropping, grayscale conversion, histogram equalization, and HSV color segmentation.

17. In case that the required target is not within the arm workspace, the robot has to change its position to reach it.

18. The labware manipulation control is connected with the Multifloor navigation system which is in turn connected with other parts like Robot Remote Center (RRC) and Process Management System (PMS) to perform the transportation tasks automatically.

19. Client-Server communication models are used to connect the Multifloor navigation control with the arm control which is in turn also connected with the Kinect control for visual processing. The client-server communication model is very useful to avoid the complexity of system integration which include multiple platforms.

20. A series of grasping and placing experiments have been achieved in this dissertation to verify the performance of the developed system. The experimental results show that the labware manipulation strategies using mobile robots are effective and efficient for the life science automation.

21. The control software platforms with their GUIs have been developed using Microsoft Visual Studio 2015 and C# programming language. The platforms are running on a Windows 10 system in the H20 tablet.

Abstract

Mobile robots in life science automation are very necessary to realize fully automated processes which in turn increase the productivity and save human resources. Whereas stationary robots connect the automation islands in a small workspace, mobile robots have the ability to connect the workstation in different laboratories. This can be achieved by performing the transportation task of multiple labware and tube racks through the whole working environment. The transportation task requires a grasping and placing ability by the robotics arms where the manipulator control is one of the main research areas in robotics. This dissertation, studies the integration between arm kinematic model and position estimation of the target to perform grasping and placing tasks for multiple labware using H20 mobile robots. The H20 robot has a dual arms consisting of 6 joints with 6 DOF and 2 DOF gripper. To perform the grasping and placing tasks, the kinematic solution of the H20 arms has to be found. The kinematic analysis is very necessary to guide the robotic arm to the target pose accurately. The kinematic problem has been solved analytically to find the exact values for the arm joints which enable the arm end effector to reach the target. The solution has been validated with MATLAB using Robotics toolbox. The kinematic model has been integrated to the arm controller of the H20 robot. This model has been verified firstly by performing a grasping and placing strategy based on sonar sensors. Since the mobile robot has positioning errors every time arriving the workstation, the labware has to be found and localized for each time. To enhance the arm manipulation performance, vision data is integrated into the arm control systems to identify the required labware. Multiple labware have to be recognized and localized on a wide workstation. The kinect sensors V1 and V2 have been used for this task. Kinect is a 3D camera providing accurate depth information to compute the spatial pose of the target. The handle, which is attached to a labware container, and the labware itself have been identified using several strategies. The HSV color detection with size and shape information has been used to recognize multiple handles related to multiple labware. In addition, some marks have been fixed on the labware lids to differentiate and identify them using features matching algorithm SURF. The client server communication model has been used to integrate multiple platforms with each other. Several models of grippers and labware containers have been designed to guarantee a secure manipulation and to handle heavy labware. According to the industrial perspective, the manipulation tasks can be accomplished generally with low complexity grippers to realize a high degree of reliability within a structured environment. The grasping and placing tasks for multiple labware have been performed successfully with the H20 robots to investigate the transportation plan in the center for life science automation.

Zusammenfassung

Mobile Roboter finden zunehmend Anwendung in der Vollautomatisierung von Prozessen zur Erhöhung der Produktivität und Einsparung von Personal. Während stationäre Roboter Automationsinseln auf kleinem Raum verbinden, verfügen mobile Roboter über die Möglichkeit der Verbindung von Automationssystemen in unterschiedlichen Laboratorien. Dies kann durch Transportaufgaben von unterschiedlicher Labware und Tube Racks realisiert werden. Für den Transportprozess benötigt der Roboterarm die Fähigkeit des Greifen und Platzieren, was gegenwärtig eine wichtige Forschungsrichtung darstellt.

Diese Dissertation befasst sich mit der Kombination des kinematischen Modells des Roboterarms und der Positionsbestimmung des Targets, um ein optimales Greifen und Platzieren von Labware durch einen H20 Roboter zu ermöglichen. Der H20 Roboter hat 2 Arme mit jeweils 6 Gelenken mit 6 DOF. Die Greifer haben 2 DOF. Um das Greifen und Platzieren anwenden zu können muss eine kinematische Lösung für die H20 Arme gefunden werden. Zur Durchführung der Prozesse des Greifens / Platzierens muss die kinematische Lösung der H20 Arme bestimmt werden, damit der Roboterarm mit hoher Präzision zur Targetposition geführt werden kann. Das kinematische Problem wurde analytisch gelöst, um die exakten Werte der Armgelenke zu bestimmen, mit denen das Target erreicht werden kann. Die Lösung wurde unter Nutzung der MATLAB Robotics Toolbox validiert. Das kinematische Modell wurde in die Steuerung der H20 Arme integriert. Das kinematische Modell wurde zunächst für Greif-/Platzierungsprozesse unter Nutzung von Ultraschallsensoren eingesetzt. Aufgrund der auftretenden Positioniergenauigkeiten des Roboters an der Übergabestation, müssen die Identifizierung und Lokalisierung der Labware jeweils neu erfolgen. Um die Performanz der Armmanipulation zu verbessern, wurden bildbasierte Verfahren für die Identifizierung der Labware in die Armsteuerung integriert. Verschiedene Labwaretypen müssen dabei auf einer Multipositionsübergabestation erkannt und lokalisiert werden. Hierzu wurden die Kinect Sensoren V1 und V2 eingesetzt. Kinect verfügt über eine 3D Kamera, die genaue Tiefeninformationen liefert, die direkt für die Bestimmung der Targetposition eingesetzt werden können. Der Halter der Labware sowie die Labware selbst wurden unter Nutzung unterschiedlicher Strategien identifiziert. Der Farbdetektor ‘‘HSV“ wurde verwendet, um verschiedene Griffe und Labware unter Zuhilfenahme von Informationen der Größe und Form der Objekte, zu identifizieren. Darüber hinaus wurden die Labwaredeckel mit speziellen Markierungen versehen, um eine eindeutige Differenzierung und Identifizierung unter Nutzung von Feature Matching Algorithm SURF zu realisieren.

Eine Client-Server-Kommunikation wurde eingesetzt, um die unterschiedlichen Plattformen miteinander zu verbinden.

Verschieden Modelle wurden für die Greifer und Labwarecontainer konstruiert, um eine sichere Manipulation der schweren Labware zu gewährleisten. Unter industriellen Gesichtspunkten sollen die Manipulationsprozesse generell mit Greifern geringer Komplexität erfolgen, um eine hohe Zuverlässigkeit in der eingesetzten Umgebung zu erreichen.
Die Greif-/Platzierungsprozesse konnten mit den entwickelten Lösungen erfolgreich an den H20 Robotern getestet werden.