COBOT : Collaborative Robot

By

Arsalan Latif
Enrollment No. 01-133192-019

Aimen Mughall
Enrollment No. 01-133192-014

Iqra Nadeem
Enrollment No. 01-133192-047

Supervised By

Assistant Professor Mr.Hassan Danish

Session 2019-23

A Report is submitted to the Department of Electrical Engineering,
Bahria University, Islamabad.
In partial fulfillment of requirement for the degree of BS(EE).

Collaborative Robots

Sustainable Development Goals

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	SDG 9	Industry, Innovation, and Infrastructure ☑
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well Being	SDG 11	Sustainable Cities and Communities ☑
SDG 4	Quality Education	SDG 12	Responsible Consumption and Production ☑
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land. ☑
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals

Range of Complex Problem Solving			
	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	
4	Familiarity of issues	Involve infrequently encountered issues	☑
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	
7	Consequences	Have significant consequences in a range of contexts.	☑
8	Interdependence	Are high level problems including many component parts or sub-problems	
Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	☑
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	☑
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	☑
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Certificate

We accept the work contained in this report as a confirmation to the required standard for the partial fulfillment of the degree of BS(EE).

____________________.	____________________
Head of Department	Supervisor
____________________.	____________________
Internal Examiner	External Examiner

Acknowledgments

We want to thanks our supervisor, Mr. Hassan Danish, for exposing us to this area of study and for guiding us through this final project. We thankyou an grateful to our family and friends for their resolute support and never ending motivation over the course of our years of Study, as well as during the excersize of overseeing Research for and producing this thesis. Without them, this attainment would not have been possible. We are grateful.

Abstract

Robots without additional sensors automatically stop when a collision is detected, which causes unneeded downtime and wear and tear on the installation. With the aid of the time-of-flight vision camera Kinect V2 along with ultrasonic distance sensor, a speed and separation monitoring system is examined in this paper. For continuous detection of human workers within a shared workspace, a Microsoft Kinect V2 vision camera is used along with Distance Ultrasonic sensor .It's feasible to Calculate distances between all of Robot joints also along the Human worker with the aid of the joint angle information provided by the robot control. The robot's velocity and acceleration values are then set to safe values after the shortest distance, which also happens to be the critical distance time, is identified. The safety of human-robot interaction can be improved by real-time object or human detection and Distance measurement and calculation, which can avoid the robot from collision with people or objects. With VREP software aid and the setup, the algorithm is tested in both a virtual and actual environment. Robot speed can be adjusted in accordance with the distance people enter a shared workspace by observing the human skeleton and object recognition.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The trend towards automation has led to a significant increase in the demand for industrial robots over the past decade, and this growth is expected to continue. It is estimated that over 1.5 million new industrial robots will be installed across various industries worldwide between 2018 and 2025.

Robots are capable of completing a wide range of tasks, but there are still some tasks that require human involvement. To address this, the next generation of robots will combine the adaptability and problem-solving skills of humans with the strength, endurance, and precision of robots.

There is a growing interest in Collaborative Robots (Cobots) that are designed to work safely with humans and can operate outside of their protective enclosures. In order to meet this demand, new robots are being developed with specialized sensors that allow them to safely work alongside people in the workplace. As of mid-2021, around 40 companies were already promoting cobots, which is a strong indication of their increasing popularity.

Despite their growing popularity, collaborative robots still have limitations and may not be able to perform certain tasks. While they are designed to stop when they encounter obstacles, they may lack proactive collision detection capabilities. However, there is potential for robots to interact with people in real-time and adapt their behavior based on incoming objects. Vision sensors can help set up cobots in a way that is beneficial for their productivity, safety, and future uses.

1.2 Related Work

How to maintain safety in a shared workspace where human robot collaboration is taking place has been the subject of extensive research in recent years. There are many ways within that that have been provided to the scientific community. In recent work, safety was maintained by using a trajectory-dependent dynamic speed and separation monitoring. Another approach made use of numerous Kinect V1 setups to track both human and robot motion and then use that knowledge to decide on safe robot motions. A depth space technique was also discussed in which depth space data was used to create repellent vectors to change the joint velocities of the robots under robot control . security, and potential future uses.

Within a simulated environment, speed and separation are monitored. Modeling both humans and robots in a sample assembly operation allows for the establishment of the minimal protective distance for the entire process. After determining the position and speed of people inside the robot workspace with the help of laser scanners, the safe separation distance is calculated based on the robots' reported position and velocity. Another approach makes use of a ladar sensor to find people. When a moving object is detected, these sophisticated hardware-linked methods cannot tell a person from an object.

1.3 Project background

Due to the trend toward automation, which has gotten significantly more popular over the past ten years and is predicted to continue, there has been an increase in the demand for industrial robots. More than 3-million brand new Industrial-robots will likely be Installed in factories worldwide between 2021 and 2023, according to estimates.

Even though robots are capable of many different tasks, some still need human assistance. As a result, the next generation of robots will meet this need by fusing human Problem solving and adaptability. abilities with robotic Strength, Endurance, and also Precision. Demand for developing brand-new Robots which are Safe for use Outside of their armored environment has increased due to the growing interest in so-called collaborative robots (cobots). Modern robots have specialized sensors for secure interactions with people. The fact that there are more manufacturers and that hundreds of businesses will be promoting cobots in 2022 is proof that these cobots are becoming more and more popular.

However, these cobots appear to have Limitation's and not yet prepared to carry out tasks that require interaction. Robotics are basically programmed to halt in Emergency mode when robots run into a barrier, but it is frequently impossible to discern when a collision is intended. It is obvious that it would be fascinating to design an environment that requires interaction where a Robot interacts with humans in real time and modifies its behavior in response to the state of some nearby objects. Visual sensors can help you design a space that is suitable for work, safe, and future use.

1.4 Project Description

To have a secure and Real time setting, this project makes use of a cutting-edge robotic arm, Ultrasonic distance sensor and Time of Flight Kinect 3D-Camera. The Speed of Robot will be regulated using Human-skeleton tracking and object-detection. Kinect V2 sensor is responsible for capturing the color and depth images and processing the depth data to locate the joints of the human body. The Kinect then sends data to the Arduino controller via serial communication to control the robotic arm based on the detected human body movements. The Arduino controller initializes the servo motors that control the robotic arm and sets up the ultrasonic sensor. It then waits for a signal from the Kinect to start the tracking process. Once a signal is received, the Arduino controller starts reading the distance data from the ultrasonic sensor and moves the robotic arm based on the distance readings.

1.5 Research Objective

This project objective is to recognize and put into practice real-time human-robot interaction by adjusting the robot's speed in accordance with the distance needed to arrive at a secure location. In order to create a secure environment, researchers are also exploring and implementing real-time interactions between humans and robots by adjusting the robot's speed in relation to distance. A robotic arm, ultrasonic distance sensor and a 3D Kinect camera will be used in the proof-of-concept to create an adaptive system.

1.6 Project Scope

This project's scope includes:

- Conducting a literature review to investigate the potential for human-robot interaction.
- Building a robot's kinematic model that enables driving in both joint and Cartesian space.
- Establishing Communication channel among matlab and Controller

to enable safe (speed) control over the cobot and gain total access to it.

- Then process includes the collection of data from the Kinect camera, followed by real-time processing to generate a human skeleton model. The model is then utilized for basic pose recognition and distance calculations. .
- In order to illustrate the setup visually and to demonstrate a proper working algorithms, a 3D virtual test environment (V-REP) that is available must be configured. This will ensure that the various methods are sound and safe to use before being implemented on Robot.
- Using a real-world setup with real-time object tracking to implement speed and separation monitoring.
- Using a straightforward human posture command to give flexibility and dynamic setting to the arrangement.

Chapter 2

Literature Review

2.1 Introduction

Several alternatives to the conventional viewpoint have been put forth by research in robot safety to prevent collisions and maintain a safe working distance between active robot systems and various objects at the working scales of robots. The majority of the literature is concerned with locating and averting potential collisions.

2.2 Purpose of Review

To track any object that comes closer to the robot, a vision sensor for mapping the environment is required. Reviewing serves the purpose of determining how effective an object detection sensor like the Kinect 3D Camera is mainly due to the cheaper prices and easy commercial availability of these sensors.

2.3 Ultrasonic Sensor

2.3.1 Introduction

An ultrasonic sensor functions by emitting high-frequency sound waves and analyzing the reflected sound as an electrical signal to determine the distance of an object. Ultrasonic waves move at a faster pace than sounds that humans can hear. The ultrasonic sensor consists of two crucial components: the transmitter, which generates sound waves using piezoelectric crystals, and the receiver, which detects the sound waves after they have traveled to and from the target.

Figure 2.1: Ultrasonic Sensor

2.3.2 Principle and Working

The sensor calculates the distance to an object by measuring the time it takes for a sound wave to travel from the transmitter to the receiver. This calculation can be expressed using the formula D = 1/2 (T x C), where D represents the distance, T represents the time, and C represents the speed of sound, which is usually around 343 meters per second. For instance, if an ultrasonic sensor was aimed at a box and it took 0.025 seconds for the sound to return, the distance to the box would be calculated as:

$$D = 0.5 \text{ x } 0.025 \text{ x } 343$$

Ultrasonic sensors are mainly employed for proximity sensing, and their applications include self-parking and anti-collision systems in automobiles. They are also utilized in robotic obstacle detection systems and manufacturing technology. When it comes to proximity sensing, ultrasonic sensors are less prone to interference from smoke, gas, and other airborne particles compared to infrared (IR) sensors (although they are still affected by physical factors such as heat).

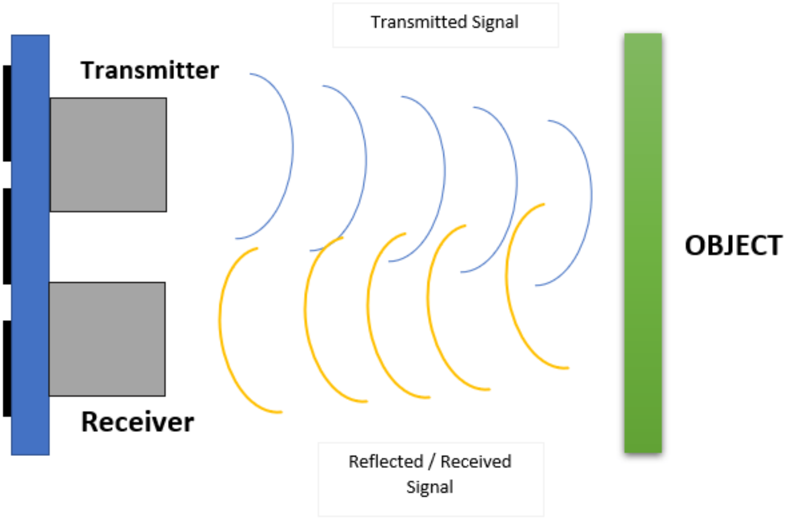


Figure 2.2: Ultrasonic Sensor Principle

In addition to proximity sensing applications, ultrasonic sensors are employed as level sensors to detect, monitor, and control liquid levels in enclosed containers like vats used in chemical plants. The medical field has benefited significantly from ultrasonic technology, which has facilitated the production of images of internal organs, identification of tumors, and monitoring the well-being of fetuses in the womb.

2.4 Kinect Sensor

2.4.1 Introduction

There are many options available today for rendering 3D environments, ranging from the most expensive and precise technologies to affordable devices that are even accessible to consumers. Such inexpensive sensors, like Microsoft's Kinect, have a significant impact on current computer vision and robotics research. Kinect gives Dense-depth prediction with high-frame-rate Color images, despite being initially created for communication

(a) Kinect Camera V1 (b) Kinect Sensor V2

Figure 2.3: Type of Kinect Camera Sensors

in a video game setting. The newest Kinect camera is version 2, and there are two different models: Kinect V1 and Kinect V2. Kinect V2 uses time of flight technology to find distance as opposed to v1, which relied on pattern projection. In this project, the Kinect v2 sensor will be used.

2.4.2 Separation Distance Calculation

Kinect v2 uses an infrared (IR) camera to acquire depth data using optical time-of-flight (TOF) technology. The fundamental idea behind TOF is to track how much time passes between an object emitting light and collecting that light after it has been reflected by it. Pulsed wave and continuous wave (CW) are the two primary technologies used by TOF cameras. In the first scenario, a high-speed counter synchronized with the transmitted signal measures the delay between the transmitted and received pulses. equation-2.1, which uses c as Speed of Light and Δt as the evaluated time-difference, can be used to quickly calculate the distance d given this time delay.

$$d = c\frac{\Delta t}{2} \tag{2.1}$$

The very high speed of light necessitates a very high-speed counter, making this method impossible to implement with silicon components op-

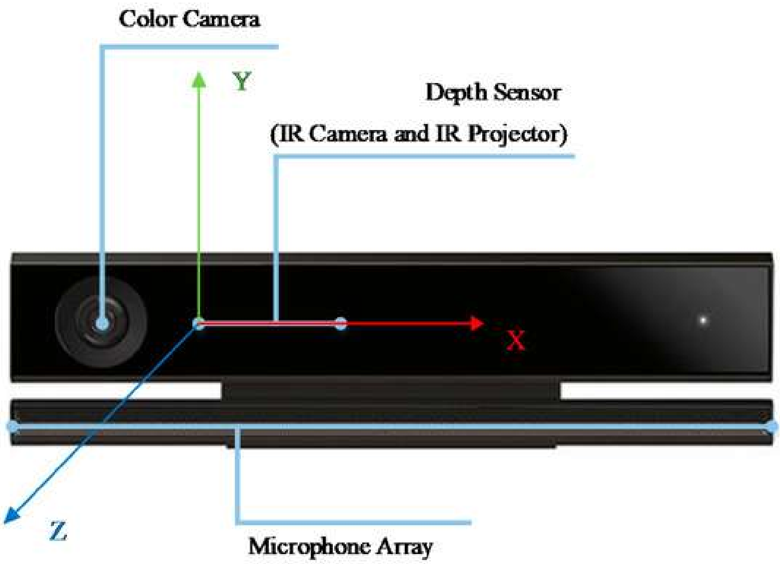


Figure 2.4: Location of the Kinect v2 camera's cameras, sensors, and coordinate system

erating at room temperature. In order to determine pi and distance, the most popular algorithm is Continuous-wave (Cw), which involves emitting Modulated-light and comparing the phase shift to the reflected signal. The phase meter shifts the reflected signal by 90 degrees using four different samples from Q1 to Q4 to determine the phase shift. Using the Equation 2.2, f is signal's Frequency and pi is phase shift picked up by the phase meter, the distance d is then calculated. It is observed that the sensor modulation of the Kinect V2 is not capable of frequency or time integration, unlike other TOF cameras.

$$d = \frac{c}{4\pi f}\Phi \tag{2.2}$$

with

$$\Phi = atan\frac{Q3 - Q4}{Q1 - Q2} \tag{2.3}$$

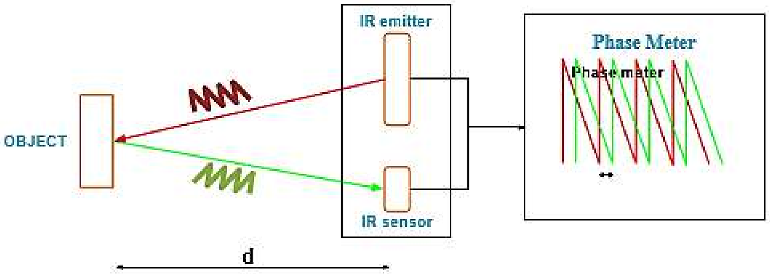


Figure 2.5: The Kinect makes use of the CW ToF principle.

There is a shift distance because the CW measurement is based on phase shift. The blurring distance d is the distance over which the broadcast occurs, and the Kinect equation makes use of several frequencies (roughly 120MHz, 80MHz, and 16MHz) to get rid of this ambiguity effect.

$$d = \frac{c}{f} \tag{2.4}$$

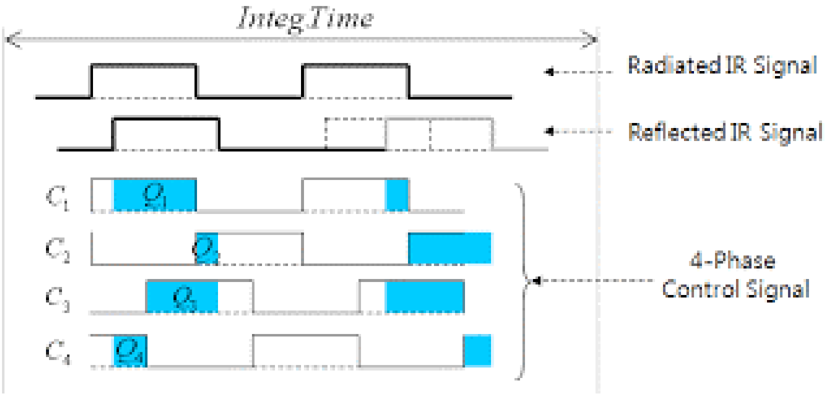


Figure 2.6: The phase change is determined using four samples with a shift of 90 degree. The electric charge per control signal is represented by the numbers Q1 to Q4.

2.4.3 Kinect Communication

MATLAB will be used for calculations and Kinect camera interaction. The process of creating algorithms is made simpler by MATLAB's numerous built-in functions for transforming and manipulating point cloud data.

2.4.4 Human Skeleton Tracking

Depending just on sensors' depths and colors information, the Camera sensor provides a built-in or integrated solution for modeling human skeletons. The algorithm determines where the skeleton's 25 vertices are, as seen in Figure. Because a complete human skeleton is extremely complex,It's crucial to remember that the skeleton created is an image. Although the algorithm is perfect, camera sensor can construct a partial Skeleton and anticipate the locations of additional Joint's when particular part's of the human-body are unknown.

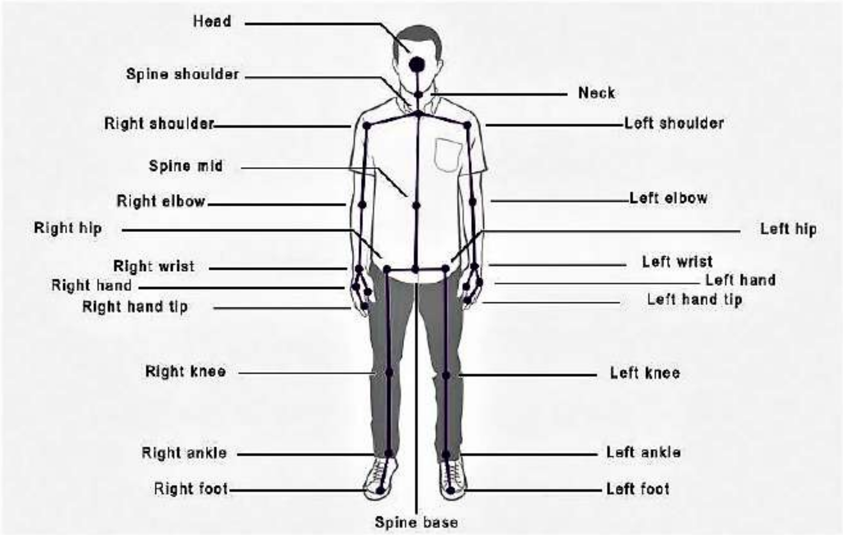


Figure 2.7: The various skeleton joints that the Kinect v2 camera picked up are represented.

2.4.5 Human and object Detection

In this project, the skeleton approach and the point cloud approach to object detection are both examined. Only points are highlighted when point-cloud algorithm is implemented; no identifiable objects are present. As a result, the entire environment must be specified to get rid of pointers from stationary things, such as robotic arm base. Benefit of this theory is that it allows for the detection of any object—be it a person or another machine—that enters the workplace and helps prevent collisions.

The skeleton can only be used to identify people who are entering the workplace. Any unidentified system can be used because the skeleton algorithm is not flawless, but there is no assurance of identity. Although self-occlusion and variations in bone length can cause problems with skeleton tracking, the calculations necessary to locate the joints are handled by the camera hardware, which reduces the computational effort. These shortcomings are tolerated as a concrete evidence facility since other, more reliable technology is employed in industrial applications.

2.5 6-DOF Robotic Arm

2.5.1 Introduction

This research is intriguing because the robotic arm is a crucial part of it.to take a closer look at the characteristics and operation of this robotic arm.

2.5.2 Robotic Arm Specifications

The robotic arm is 6-DOF robotic arm with six rotational joints. Equipped with 6pcs MG996 analog servos controller for the joint and claw.

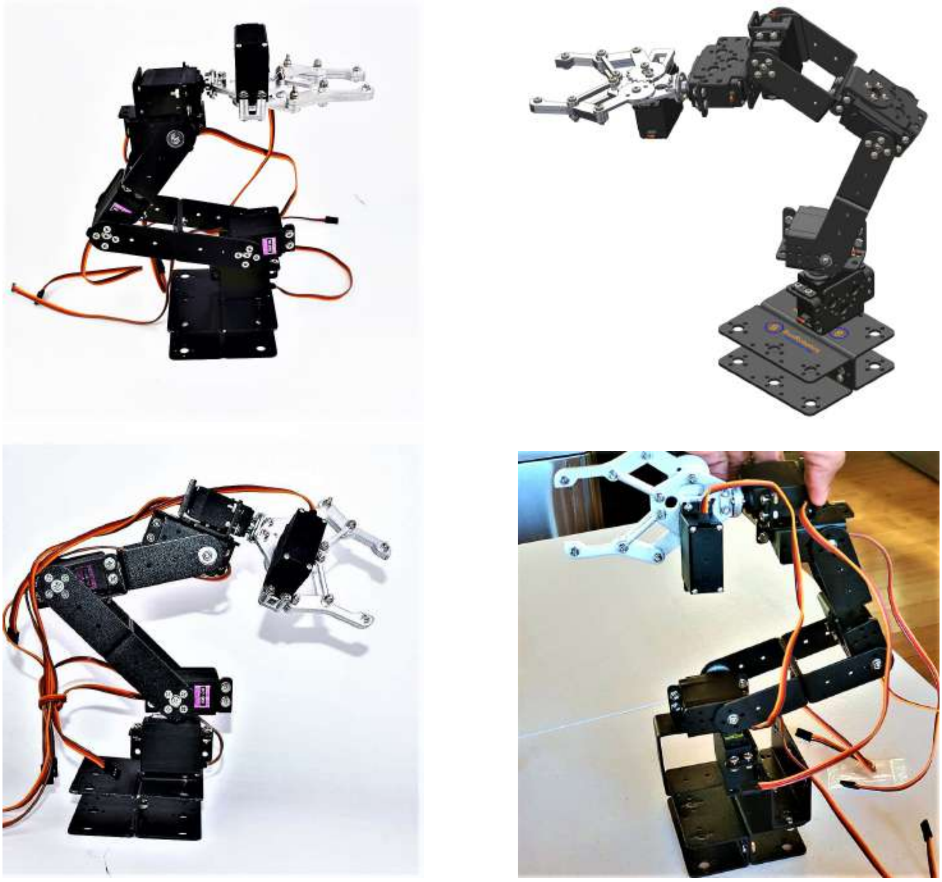


Figure 2.8: 6- DOF Robotic Arm

Table 2.1: Robotic Arm Specifications.

Arm Type	6-DOF Robotic Arm with Servos
Arm material	Aluminum
Arm size	31 * 22 * 7.5cm / 12.20 * 8.66 * 2.95in
Servo's	MG996 analog servos
Servos quantity	6
Servos working voltage	4.8 7.2V
Servos Torque	10kg.cm

2.5.3 Robotic Arm Kinematic Model

Each robot arm can be represented as joints (rigid bodies) connected to each other by joints. This structure can only be represented using lower pairs. A two-dimensionalkinematics architecture of the manipulation is illustrated and is shown in Fig 2.7. Once the angles of the six rotary joints of the robotic arm are known, Position of each joint can be determined uniquely. To calculate exact Position of robotic arm, a coordinate system is assigned to each joint. To locate one coordinate system in relation to another, at least 4 factors must be known. Despite the fact that there are multiple approaches to specify these factors, one of them is the Denavit-Hartenberg notation. Figure 2.8 shows the resultant Denavit Hartenberg interpretation. and therelating Denavit Hartenberg parameters.

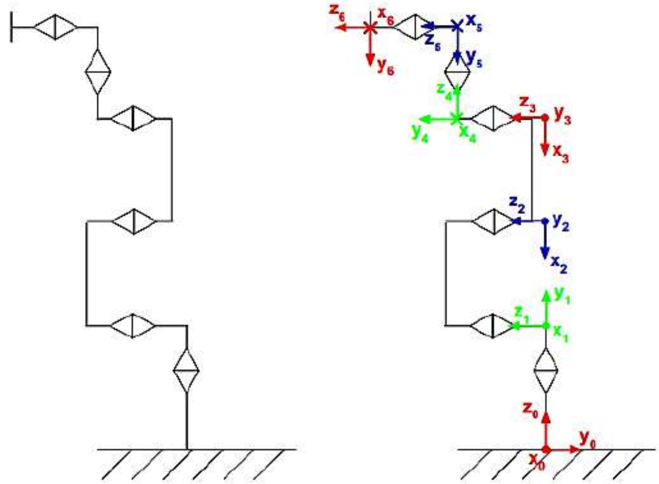
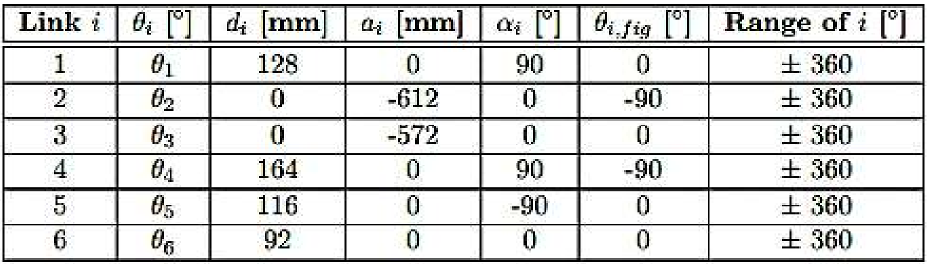


Figure 2.9: 2D Kinematics notation of robotic arm With or Without the Denavit Hartenberg coordinat-systems.



Link i	θ_i [°]	d_i [mm]	a_i [mm]	α_i [°]	$\theta_{i,fig}$ [°]	**Range of i [°]**
1	θ_1	128	0	90	0	± 360
2	θ_2	0	-612	0	-90	± 360
3	θ_3	0	-572	0	0	± 360
4	θ_4	164	0	90	-90	± 360
5	θ_5	116	0	-90	0	± 360
6	θ_6	92	0	0	0	± 360

Figure 2.10: Robotic Arm Denavit-Hartenberg parameters

Chapter 3

Separation Distance-Calculation

In order to analyze data, whether it be from a point cloud, skeleton joint coordinates, or robot arm positions, all points must be referenced to a specific frame of reference. However, since the frame of reference can move, comparing it to other points, such as in Distance-calculations, can be highly impractical. In this chapter, a technique is presented for defining the position and orientation of a coordinate system, as well as for converting its points into a desired world coordinate system. By processing data from both the camera and the robot, it becomes possible to calculate the minimum distance between the robot and a human being.

3.1 Homogeneous transformation Matrices

Most typical method to describe the point's in the Space is to use Cartesian-Coordinates (X, Y, and Z). Each point can be expressed as a vector field so that uniform transformations can be carried out `[WX WY WZ W1]^T` with W is a Weighing coefficient set in the robotics for 1. The result is the following notation to describe the points in accordance to the chosen frame of reference.

$$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.1}$$

The homogeneous transformation matrix H is a 4x4 matrix that is used to convert a position vector from one reference frame to another. The rotational vector R, a position vector p, a projection vector f, and a weighting factor w make up the matrix. Projection in Robotics, Vector is denoted to 01 × 3 and the weight factor is set to 1.

$$H = \begin{bmatrix} R11 & R12 & R13 & a \\ R21 & R22 & R23 & b \\ R31 & R32 & R33 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

3.2 Translation

Position-vector `[a b c 1]^T` is used in Figure 3.1 to represent how Coordinate-system could be translated in Space. But because the orientation doesn't change, the rotation matrix is an identity matrix.

$$H = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

To determine the dimensions of point p in relation to the axes of coordinate system 1, one can easily obtain this information by pre-multiplying the position vector of point p with the uniform transformation matrix described in equation 3.4. (x1, y1, z1) starting from a point p with coordinates (x2, y2, z2) relation to coordinate-system 2, and by using coordinate-system 1 to determine its position.

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x2 \\ y2 \\ z2 \\ 1 \end{bmatrix} = \begin{bmatrix} x2 + a \\ y2 + b \\ z2 + c \\ 1 \end{bmatrix} \tag{3.4}$$

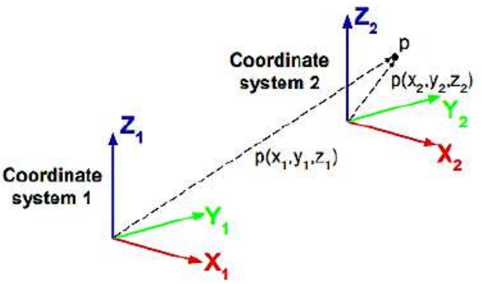


Figure 3.1: A system of translated coordinates.

3.3 Rotation

A reference frame can rotate in addition to being translated, as shown in Figure 3.2. The rotated axes' unit vector in relation to the original coordinate-system is represented in each column of a rotation matrix, which serves as the representation for these rotations. The resulting transformation matrix is presented in Equation 3.5.

$$H = \begin{bmatrix} R11 & R12 & R13 & a \\ R21 & R22 & R23 & b \\ R31 & R32 & R33 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

We will use Figure 3.3's rotation matrix for coordinate system 2 as an illustration. The vectors for the y2 and z2 can be formulated similarly to the previous method, yielding Rotatio- matrix R shown below.

$$R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

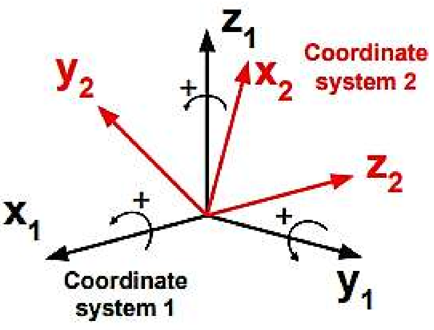


Figure 3.2: Conversion from coordinate system 1 to 2, with +ive angles with conventional orientation.

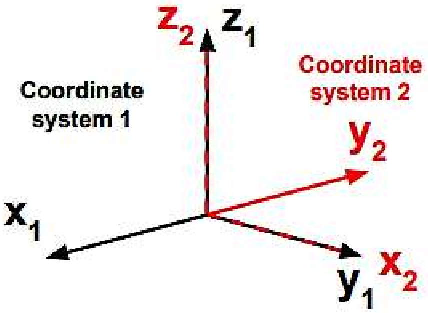


Figure 3.3: Rotated coordinate systems, as an example.

3.4 Transformation

The complete transformation produces the complete homogeneous transformation matrix by combining translation-rotation as discussed in preceding Sections. In order to derive the position vector relative to coordinate system 1, the resulting homogeneous matrix is multiplied beforehand with the position vector from coordinate system 2, using a similar approach to that outlined in Equation 3.4.

$$\begin{bmatrix} x1 \\ y1 \\ z1 \\ 1 \end{bmatrix} = \begin{bmatrix} R11 & R12 & R13 & a \\ R21 & R22 & R23 & b \\ R31 & R32 & R33 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x2 \\ y2 \\ z2 \\ 1 \end{bmatrix} \tag{3.7}$$

As an illustration, we will transform point p with coordinates (x2, y2, z2) in relation to coordinate-system 2 to determine its position in relation to coordinate-system 1. Rotation matrix is the same as that in equation 4.6. The outcome of Equation 3.8 is obtained by filling in the remaining equation with the parameters from Figure 3.4.

$$\begin{bmatrix} x1 \\ y1 \\ z1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & -7 \\ 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ 5 \\ 8 \\ 1 \end{bmatrix} \tag{3.8}$$

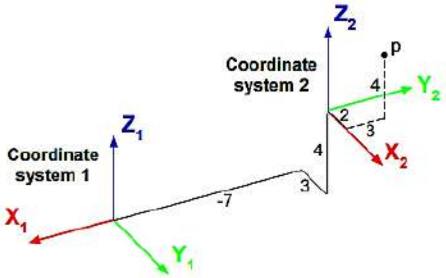


Figure 3.4: Translated-rotated coordinate-systems.

3.5 Separation Distance Calculation

Once the data obtained from the camera and the robot are referenced to a specific world coordinate system, it becomes possible to calculate the minimum distance between the robot and a human operator. This is so that all data may be represented using the same reference frame and Cartesian-coordinates. Use the following equation to calculate the Euclidean distance between two places.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

An alternate method is to figure out how far people and robots are from one another in depth space. In this scenario, instead of translating the depth data from the 3D camera into Cartesian space, the system computes the distances to the rays originating from the camera center.This approach employs the skeleton and point cloud data, however since it just uses raw depth data, it is not further developed.

The minimal distance of separation is obtained by computing Euclidean distance to each joint after getting the skeleton-joints. Since just 25 joints have been acquired, this computation requires very little computer resources. The technique is also used on the filtered point cloud because there are very few points there. The two methods used to determine the distance have no appreciable time difference.

Figure 3.5 displays a comparison of the distance computation. Because the skeleton-tracking technique detects the interior positions of the skeleton-joints and the point cloud approach uses the body's outside shell, it should be noted that the computed distance varies between the two methods. The closest distance to the TCP is determined in this example, however more sites might be readily added.

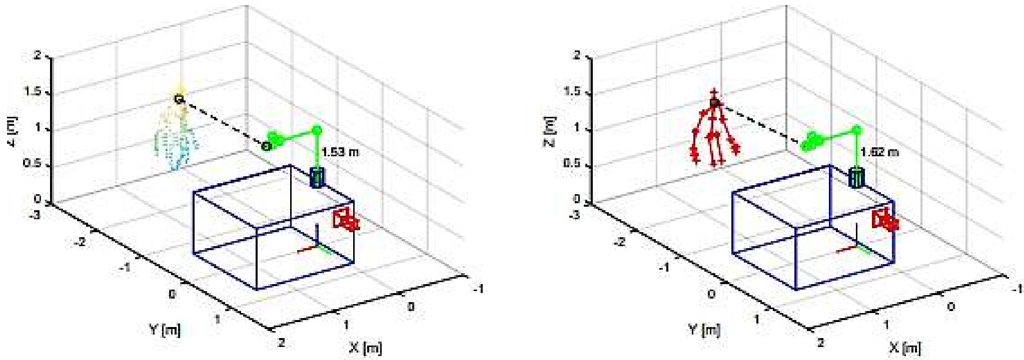


Figure 3.5: Distance calculation Illustration with point cloud (on left side) and skeleton (on right side) approach.

Chapter 4

Human-Robot Interaction

We will examine human-robot interaction in greater detail in this chapter, including potential outcomes put into use robotic -installations. The different interaction methods and Safety requirements are first described. Standardized stopping functions are described in more detail.

Figure 4.1: Human - Robot Interaction

4.1 Introduction

A relatively recent development in robotics called human-robot interaction is changing how we think about traditional production environments. Robotic interaction with humans was initially strictly forbidden and robots were completely fenced.The safety features were implemented to minimize any interaction between robots and workers and ensure that powerful machines do not cause harm to humans. It frequently required a complete stop or to turn off the power source when operators needed to communicate with the robot. It is obvious that this strategy did not increase productivity and efficiency, necessitating the development of new system designs.

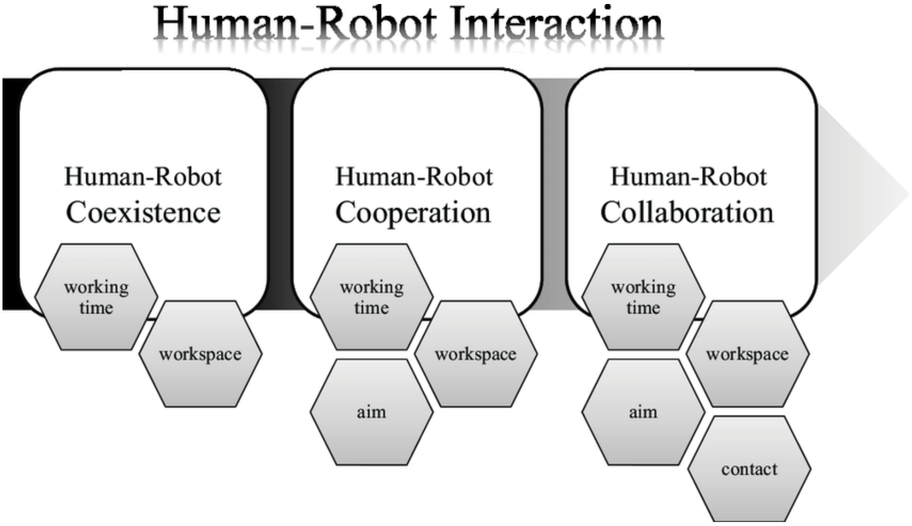


Figure 4.2: Categorisation of Human-Robot-Interaction

Hybrid production systems, which feature close cooperation between the robot and its coworker, are the solution to these issues. A new type of production is made possible by fusing human flexibility and problem-solving abilities with robot strength, endurance, and precision. This new type of production enables much greater human flexibility and allows for much more higher-value and safer work to be done by humans. Human-robot collaboration can be very profitable, especially in small and medium-sized productions, even though this new method involves many additional mandatory safety measures aspects.

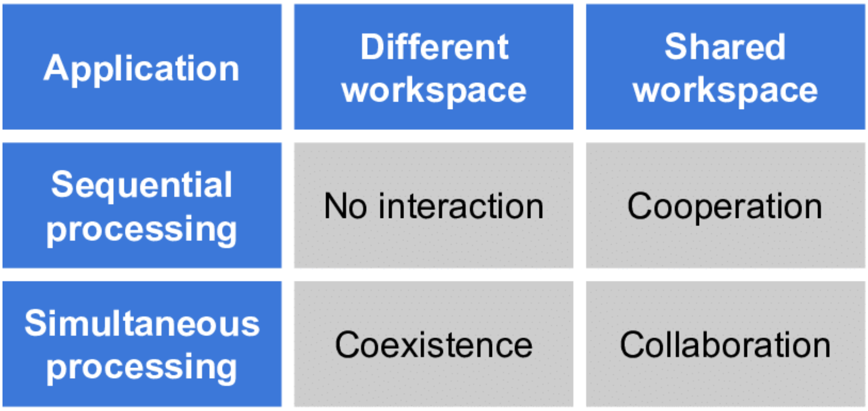
4.2 Types of Interaction

Two interaction parameters can be used to categorize human-robot interactions: Space and time. We can say that there is no interaction if the human and the robot do not move in a shared workspace or at the same time (Figure 4.1).

Coexistence refers to the traditional applications in which there is no need for human intervention during the operation process. To maintain a distinct workspace, the robot is enclosed and all access is controlled. The operator will still need to go into the workspace of the robot for sole Purpose of performing Maintenance, when the cobot is off.

When collaboration exists, Robotic arm and human employee share a workspace, though they are not present at the same time. Installations where people load and unload the robot cell are typical applications. Safety measures must be taken in this situation as well to ensure that robotic arm will not function when object is nearby.

Collaboration, which allows humans and robots to work together in the same space, is the most advanced form of interaction. For the robot to be able to detect people, dependable sensors are needed, and its range of motion must be constrained. Additionally, the setup in the robotics lab will use this kind of interaction.



Application	Different workspace	Shared workspace
Sequential processing	No interaction	Cooperation
Simultaneous processing	Coexistence	Collaboration

Figure 4.3: Different forms of human-robot interactions.

4.3 ISO Standards

To guarantee that robots are implemented safely in a wide range of applications, various institutions have created safety standards. The International Organization for Standardization (ISO), a global network of standardization organizations, is one of these organizations with Belgium being one of 162 nations. Standards describe agreements, specs, or Criterian related product, Service, or Method. Despite what the general public believes, Standards did not developed or approved by Government's or other Authorities.

The ISO 10218-1:2011 and ISO 10218-2:2011 are the standard's used in this project. They are both concerned with the safety requirements for industrial robots. These standards only briefly (about eight pages out of 152) describe the requirements for collaborative operation. As a result, new standards are being created, such as ISO/TS15066, which is specifically intended for collaborative guidance.

4.4 Economic Importance

Collaboration between humans and robots may be the answer to production lines with variable demand and low output quantities, as well as enhancing the effectiveness of present facilities.Integrated manufacturing systems may be a more cost-effective solution, especially for production batches that are too large for manual assembly but too small for fully automated assembly using robots.

Robotics adoption in business could result in more high-skilled job opportunities. It will raise productivity at work and support employers in making the best use of employees' working hours. Additionally, it will

Figure 4.4: Collaborative Robot Cobot Market Size.

support and enable the growth of the entire economy and, in challenging inflationary times, even boost its GDP.Even though it isn't specifically using its skills (such as packaging, machine maintenance, etc.), a collaborative robot is made to function alongside people. However, it might be beneficial. Your employees might benefit from it in order to accomplish some goals. The installation of some robot cells will only serve to provide laborers with the components they need to assemble objects.

While there are many advantages to human-robot collaboration, it is important to remember that there is no one size fits all solution for all production lines, so existing robotic facilities must still be in place. Although it has the potential to develop new uses for robots in areas where they haven't been before, human-robot interaction is an extension of current tools.

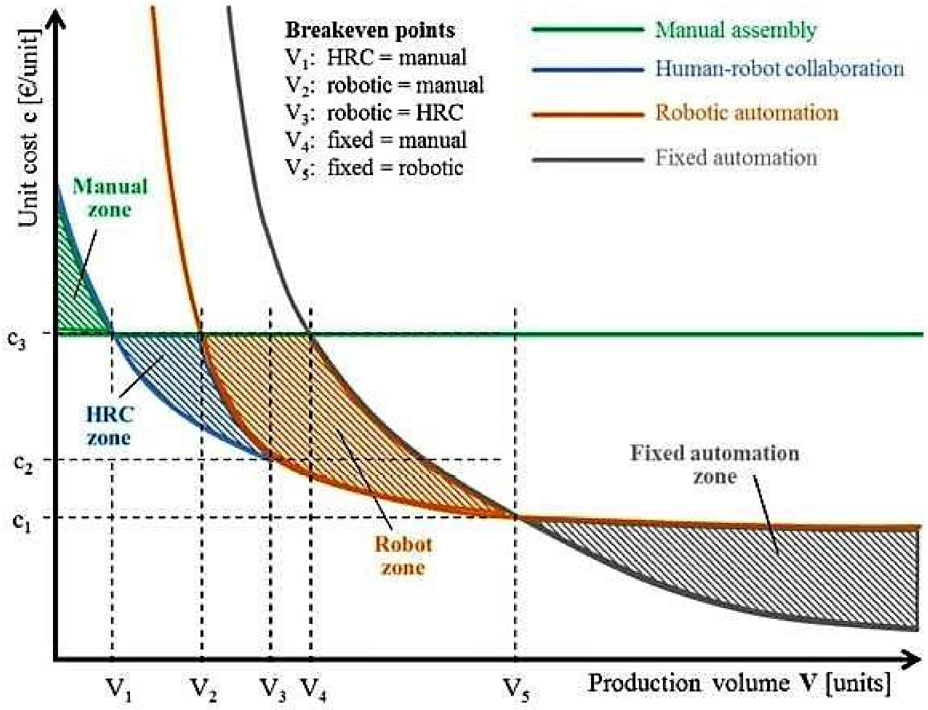
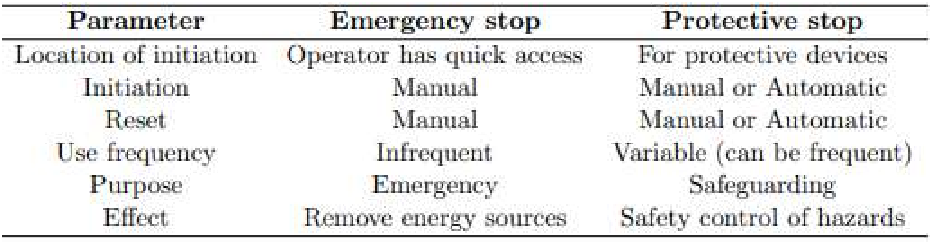


Figure 4.5: Collaborative Robot Cobot Cost based on Production Volume.

4.5 Cobot Stopping functions

The robot will need to respond when a safety limit is reached in order to prevent further damage from occurring. Based on the installation's reason and kind, this will frequently result in a stop to the robotic arm movements, though it can be done in a variety of ways. Two distinct stopping functions, an emergency stop and a protective stop, are mandated by the ISO 10218-1:2011 standard. In figure 4.2, a comparison of the two stop categories is shown.



Parameter	Emergency stop	Protective stop
Location of initiation	Operator has quick access	For protective devices
Initiation	Manual	Manual or Automatic
Reset	Manual	Manual or Automatic
Use frequency	Infrequent	Variable (can be frequent)
Purpose	Emergency	Safeguarding
Effect	Remove energy sources	Safety control of hazards

Figure 4.6: An emergency stop and a protective stop are contrasted.

4.6 Types of Collaborative Robot Operation

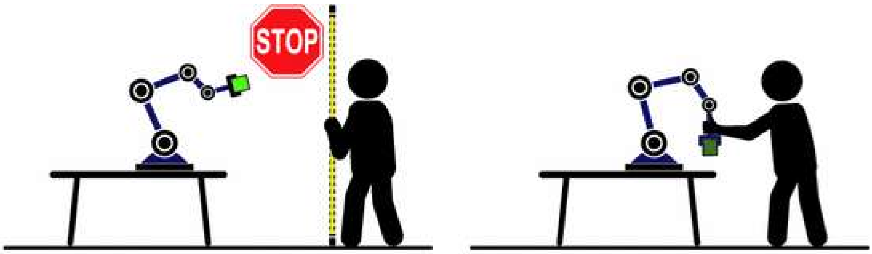
Cobots are inexpensive industrial robots that can operate close to humans without putting them in danger. Cobots can be used for a variety of tasks, including machine maintenance, welding, packaging, and palletizing because they are more adaptable than standard or traditional industrial robots.The four main categories of collaborative robots are safety monitored stop, speed and separation, force and power limiting, and hand guiding.

4.6.1 Force and power Limiting cobot

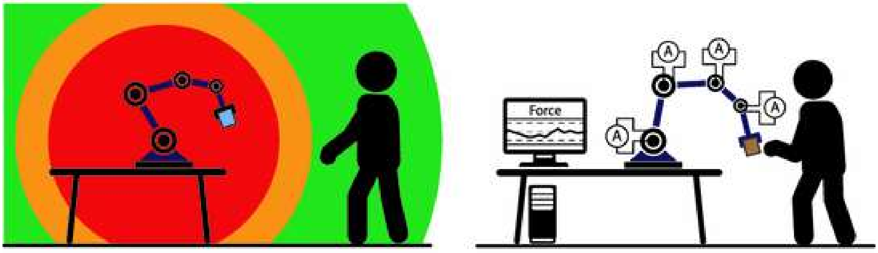
The majority of people probably picture collaborative robots as being force and power constrained cobots. There is no longer a need for safety barriers because they are specifically built to detect human contact and stop operating. Smaller applications are where they work best.

4.6.2 Safety Rated + Monitored Stop cobot

The need to manually restart a robot is eliminated by safety-rated monitored stop robots, which detects when human enters a specified area and cease operation until human leaves that Workspace. The robot automatically resumes its task after the person has left. When a person and a robot



(a) Safety-rated monitored stop (b) Hand guiding



(c) Speed and separation monitoring (d) Power and force limiting

Figure 4.7: Cobot opeartion types.

are not interacting as much, these are typically used.

4.6.3 Fenceless or Speed and separation cobot

In scenarios where limited interaction with humans is necessary, but they still require access to the robot, ”fenceless” collaborative robots with high speed and separation capability can be useful. To ensure safety, laser area scanners are often employed to define the proximity zones around the robot’s work area. As an individual approaches the robot, it decelerates, and if the person enters the workstation’s reachable area, the robot stops altogether. Once the operator exits the designated workspace, the robot resumes its regular speed of operation.

4.6.4 Hand Guided cobot

Cobots with hand-guiding capability feature a safety-rated device at the end of their arm, which enables them to be physically guided or moved by a human operator. This approach allows for rapid and simple programming of new robot placements and trajectories by hand, which is well-suited for mobile applications where cobots move between different stations and need to be retrained to perform various tasks. This method is also effective in scenarios where cobots need to be frequently reprogrammed for new duties.

Chapter 5

System Implementation

The establishment of the implementation in the virtual and physical environments is briefly described in this chapter. The project is implemented virtually in Coppelia Software.

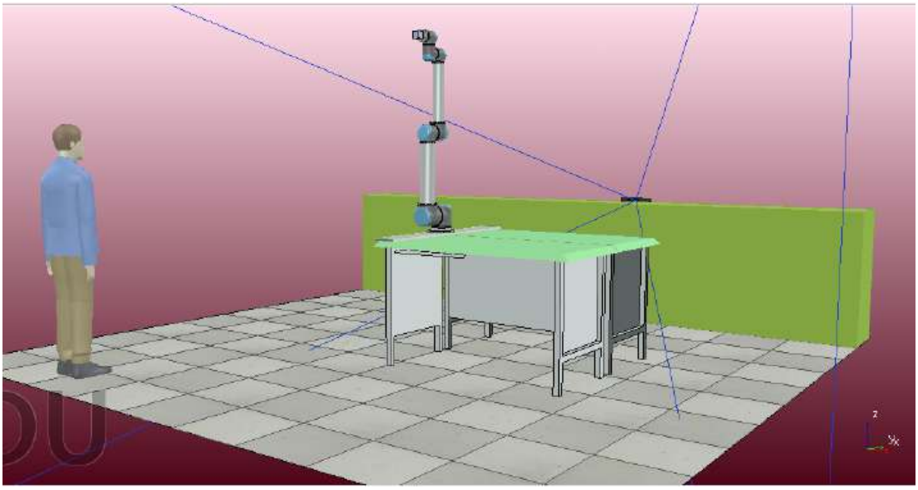


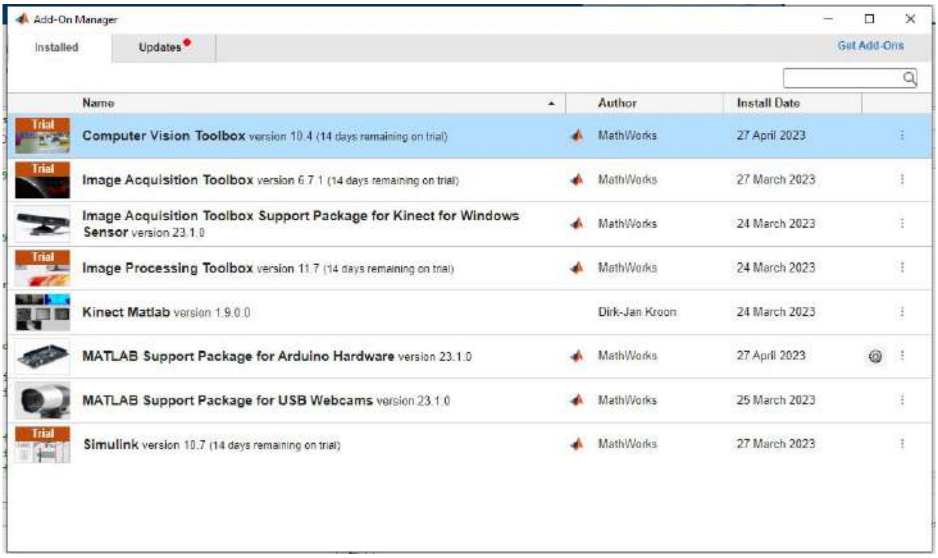
Figure 5.1: Set up Illustration In Coppelia Software.

5.1 Hardware Setup Implementation

5.1.1 Matlab Interface for Serial Communication with Controller

To acquire data from kinect v2 sensor for object detection and object distance measurement, we have to install some packages and tool boxes to perform operation. These packages includes Image Acquisition Toolbox, Computer Vision Toolbox, Kinect Matlab and so on.

The MATLAB Image Acquisition Toolbox also supports the Kinect sensor, which is a popular depth camera system that can be used for a wide range of applications, including 3D scanning, gesture recognition, and augmented reality. To use the Kinect with the Image Acquisition Toolbox, we will need to install the Kinect for Windows SDK, which provides the



Name	Author	Install Date
Trial **Computer Vision Toolbox** version 10.4 (14 days remaining on trial)	MathWorks	27 April 2023
Trial **Image Acquisition Toolbox** version 6.7.1 (14 days remaining on trial)	MathWorks	27 March 2023
Image Acquisition Toolbox Support Package for Kinect for Windows Sensor version 23.1.0	MathWorks	24 March 2023
Trial **Image Processing Toolbox** version 11.7 (14 days remaining on trial)	MathWorks	24 March 2023
Kinect Matlab version 1.9.0.0	Dirk-Jan Kroon	24 March 2023
MATLAB Support Package for Arduino Hardware version 23.1.0	MathWorks	27 April 2023
MATLAB Support Package for USB Webcams version 23.1.0	MathWorks	25 March 2023
Trial **Simulink** version 10.7 (14 days remaining on trial)	MathWorks	27 March 2023

Figure 5.2: Matlab Support Packages for kinect V2 Sensor.

necessary drivers and software for communicating with the Kinect sensor. Once installed SDK, we can use it to acquire depth and RGB images from the Kinect sensor in MATLAB. With this toolbox, we can develop custom applications that leverage the power of the Kinect sensor for a wide range of applications in computer vision and robotics.

5.1.2 Speed and Separation Monitoring (SSM)

For SSM MATLAB code is responsible for capturing the color and depth images from Kinect and processing the depth data to locate the joints of the human body. The code then sends data to the Arduino via serial communication to command or direct the movements of the robotic arm based on the detected human body movements. The Arduino code initializes the servo motors that control the robotic arm and sets up the ultrasonic sensor. It then waits for a signal from the MATLAB code to start the tracking process. Once a signal is received, the Arduino starts reading the distance data from the ultrasonic sensor and moves the robotic arm based on the distance readings. When a human body is detected by the MATLAB code, it sends a signal to the Arduino, which then starts tracking the human body's movements using the ultrasonic sensor. The robotic arm is controlled based on the distance readings from the sensor to follow the movements of the human body.

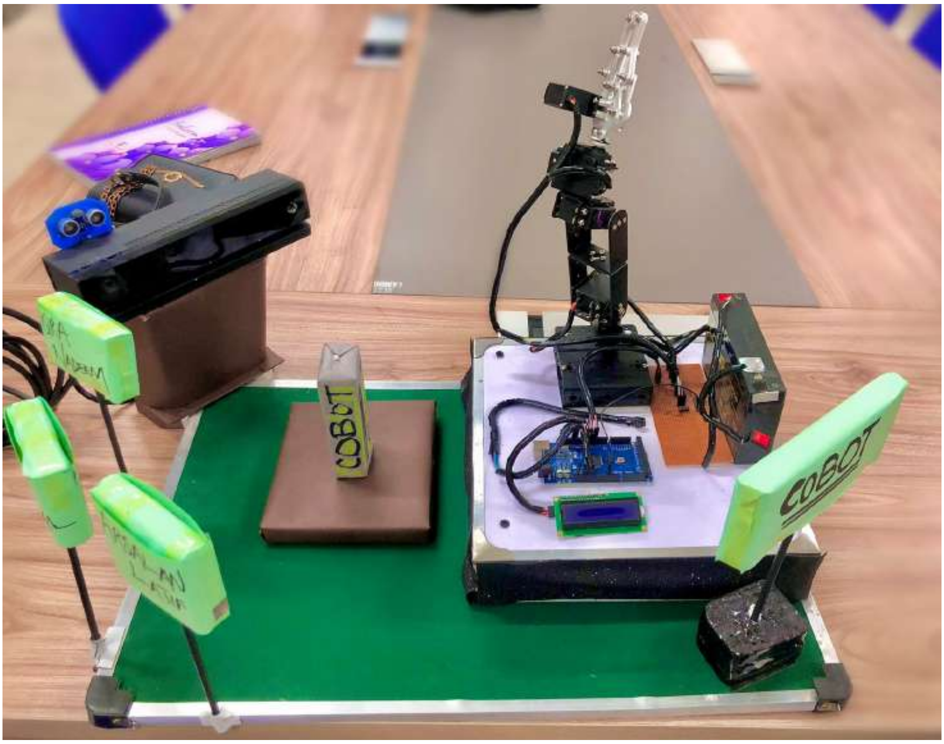


Figure 5.3: Hardware Setup for Speed and Separation Monitoring

The first line of the code "clear all" clears all variables and objects from the workspace. The second line "clc" clears the command window. The third line initializes the serial port communication between Matlab and the Arduino board using the serialport() function. The fourth line resets the image acquisition hardware. The next four lines create videoinput objects for color and depth kinect video. The next four lines configure the depthVid videoinput object to manually trigger each frame acquisition and set the number of frames per trigger and the number of trigger repetitions to infinite. The next line enables the body tracking feature on the depthVid videoinput object. The next line starts the acquisition of video data from the depthVid videoinput object. The next line creates a figure window to display the depth image data. The next line defines a connection map for the joints of the human body skeleton. The while loop continues as long as the figure window is open. The ”trigger” function of the depthVid videoinput object is called to capture a frame of depth image data. The ”getdata” function is used to retrieve the depth map, timestamp, and metadata from the depthVid videoinput object. The ”any” function checks if there are any bodies tracked in the depth image. The ”find” function is used to find the index of the tracked bodies. The number of tracked bodies is calculated using the ”length” function. The colors array is defined to assign a color to each body. The ”imshow” function is used to display the depth image data. If there are tracked bodies, the ”skeletonJoints” variable is assigned the depth indices of the body joints. A nested for loop is used to iterate over each joint connection and each tracked body. The X and Y coordinates of the joint connection points are retrieved and used to draw a line between them using the ”line” function. The ”num2cell” and ”cell2mat” functions are used to convert the Y coordinate to a cell array and then to a matrix. The ”fopen” function opens the serial port connection with the Arduino board. The ”fprintf” function is used to send a character ’a’ to the Arduino board. The ”pause” function is used to delay for 0.6 seconds. The ”fclose” function is used to close the serial port connection with the Arduino board. The Y coordinate value is displayed on the command window using the ”disp” function.

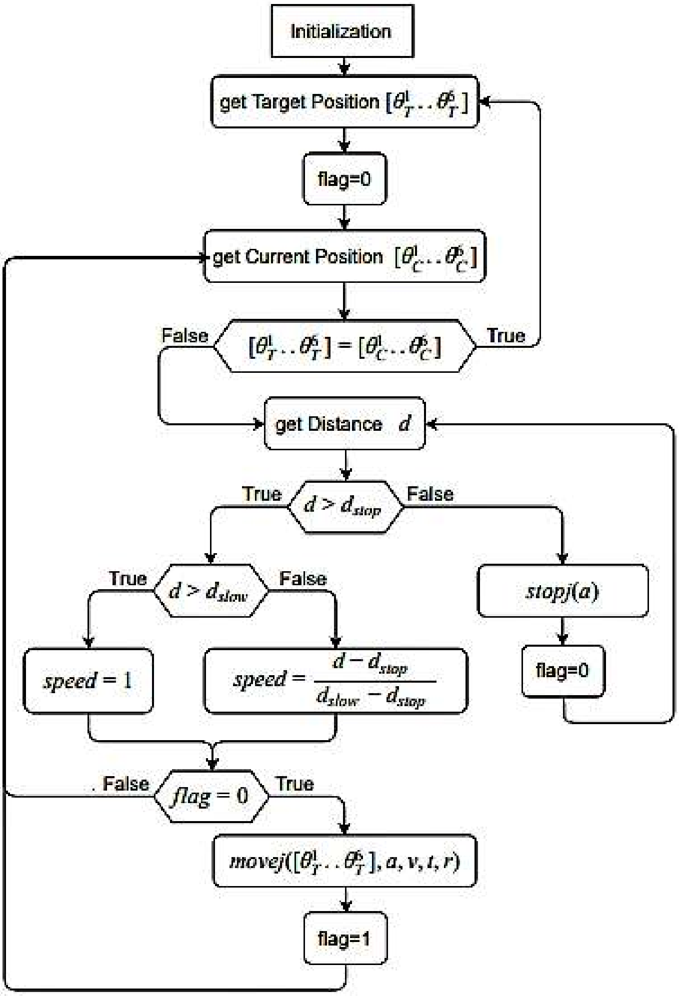


Figure 5.4: Speed and Separation Monitoring flowchart

Chapter 6

Speed and Separation Monitoring Results

For SSM MATLAB code is responsible for capturing the color and depth images from Kinect and processing the depth data to locate The articulations of the human body. The code then sends data to the Arduino via serial communication to command or direct the movements of the robotic arm based on the detected human body movements. The Arduino code initializes the servo motors that control the robotic arm and sets up the ultrasonic sensor. It then waits for a signal from the MATLAB code to start the tracking process. Once a signal is received, the Arduino starts reading the distance data from the ultrasonic sensor and moves the robotic arm based on the distance readings.

6.1 Skeleton Tracking and Distance measurement

Following figure shows the result of skeleton tracking using kinect V2 Sensor for human detection and distance calculation for speed and separation monitoring.Depending just on sensors' depths and colors information, the Camera sensor provides a built-in or integrated solution for modeling human skeletons. The algorithm determines where the skeleton's 25 vertices are, as seen in Figure. Because a complete human skeleton is extremely complex,It's crucial to remember that the skeleton created is an image. Although the algorithm is perfect, camera sensor can construct a partial Skeleton and anticipate the locations of additional Joint's when particular part's of the human-body are unknown. The Kinect v2 sensor can track up to six people simultaneously, with each person's skeleton represented by 25 joints. The joints include the head, neck, shoulders, elbows, wrists, hips, knees, and ankles.

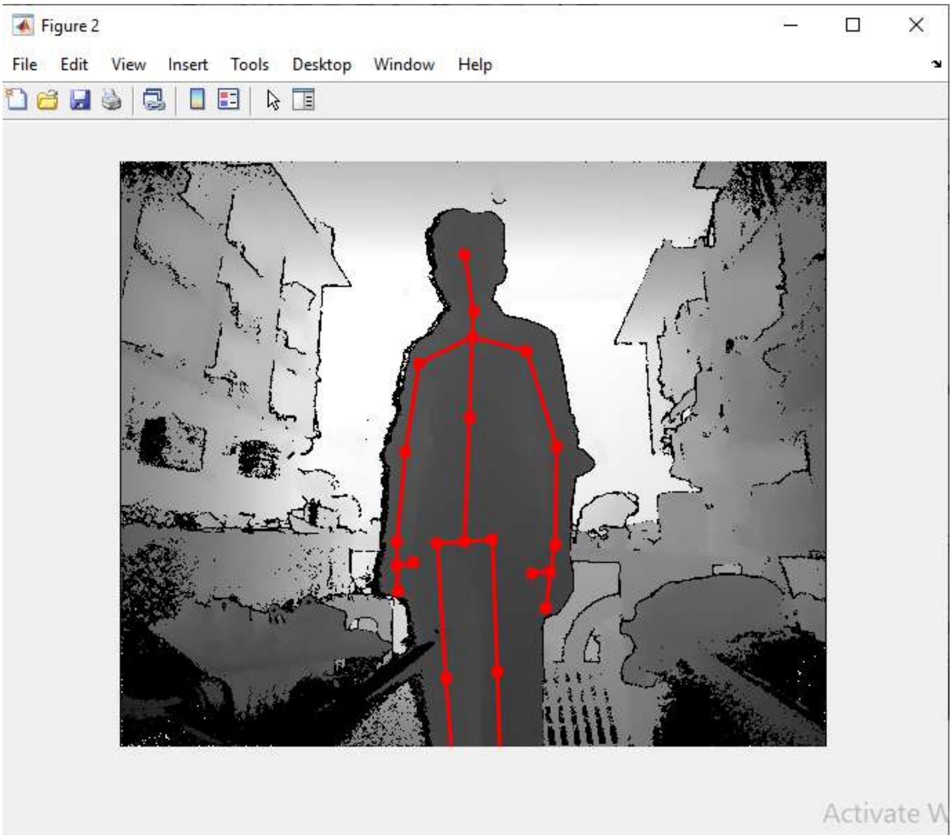


Figure 6.1: Matlab Result of Skeleton Tracking for object detection

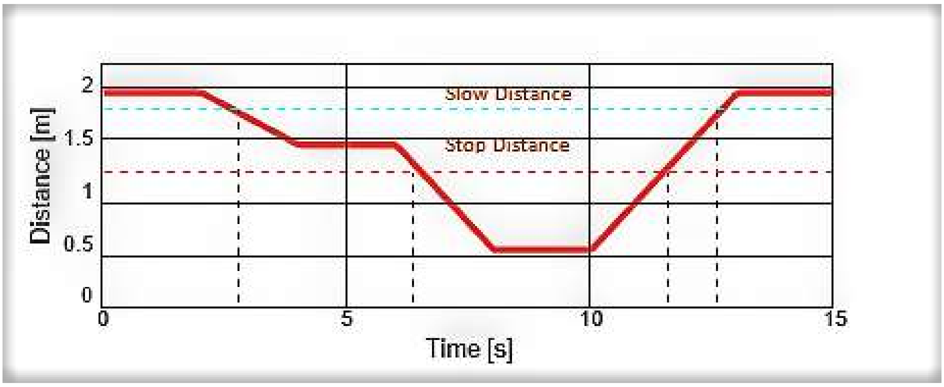


Figure 6.2: Matlab Result of object distance measurement using kinect V2 Sensor

6.2 Hardware Setup Results

Following figures shows the hardware results.

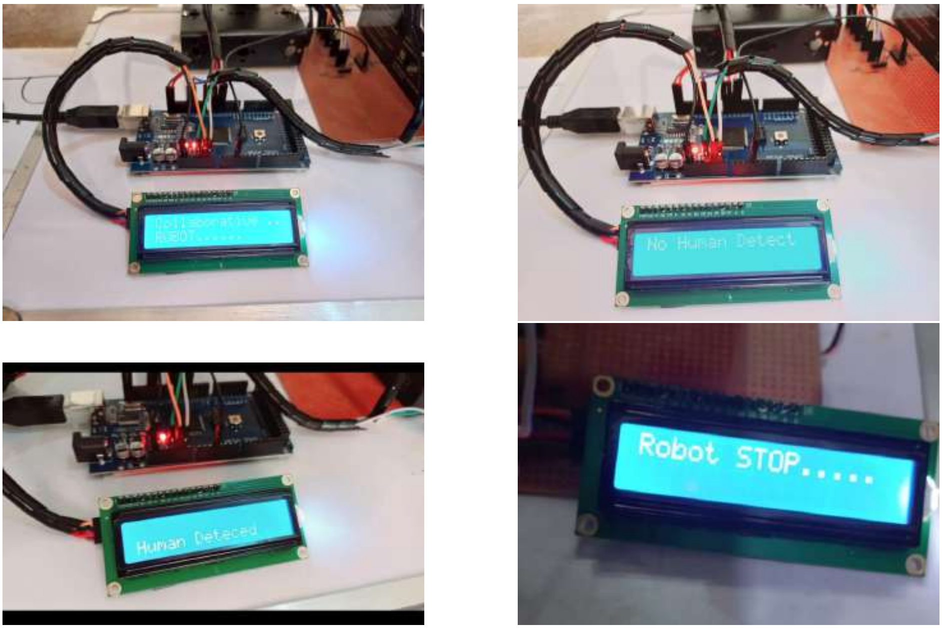


Figure 6.3: Hardware Setup Results

Chapter 7

Conclusion

7.1 Conclusion

To have a secure and Real time setting, this project makes use of a cutting-edge robotic arm, Ultrasonic distance sensor and Time of Flight Kinect 3D-Camera. The Speed of Robot will be regulated using Human-skeleton tracking and object-detection. Kinect V2 sensor is responsible for capturing the color and depth images and processing the depth data to locate the articulations of the human body. The Kinect then sends data to the Arduino controller via serial communication to command or direct the movements of the robotic arm based on the detected human body movements. The Arduino controller initializes the servo motors that control the robotic arm and sets up the ultrasonic sensor. It then waits for a signal from the Kinect to start the tracking process. Once a signal is received, the Arduino controller starts reading the distance data from the ultrasonic sensor and moves the robotic arm based on the distance readings.

This project demonstrates thatthe use of cameras or other visual input devices along with technology that can detect and follow objects in real-time can contribute to a safer collaboration between humans and robots, which aligns with previous research and standards recommendations. Through the use of precautionary stopping and speed adjustments, collisions and impacts with human workers can be prevented, leading to increased efficiency and capacity for the robots. Human tracking can also enable gesture recognition, creating a more dynamic environment where basic stop signs such as raised hands can halt the robot. However, whilethere are indications of positive outcomes., the current arrangement is not yet adequate. It is not possible to replicate the skeleton tracking capabilities of the Kinect camera in a virtual environment, and since only one camera was employed, the system has not been implemented for industrial use.

Chapter 8

References

Bibliography

[1] W. Robotics, “Executive Summary World Robotics 2017 Industrial Robots,” World Robotic Report, pp. 15–24, 2017. [Online]. Available: https://ifr.org/downloads/press/ Executive Summary WR 2017 Industrial Robots.pdf

[2] J. Kr¨uger, T. K. Lien, and A. Verl, “Cooperation of human and machines in assembly lines,” CIRP Annals - Manufacturing Technology, vol. 58, no. 2, pp. 628–646, 2009.

[3] E. H. Østergaard, “White Paper the Role of Cobots,” 2017. [Online]. Available: https://info.universal-robots.com/the-role-of-cobots-in-industry4.0

[4] Universal Robots, “Home Page,” 2018. [Online]. Available: https://www.universal-robots. Com

[5] University of Antwerp, “Op3Mech,” 2018. [Online]. Available: https://www.uantwerpen. be/en/research-groups/op3mech/

[6] Microsoft, “Kinect for Xbox One,” 2017. [Online]. Available: https://www.xbox.com/ en-US/xbox-one/accessories/kinect

[7] K. Khoshelham and S. O. Elberink, “Accuracy and resolution of kinect depth data for indoor mapping applications,” Sensors, vol. 12, no. 2, pp. 1437–1454, 2012.

[8] O. Wasenm¨uller and D. Stricker, “Comparison of kinect v1 and v2 depth images in terms of accuracy and precision,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10117 LNCS, pp. 34–45, 2017.

[9] K. Litomisky, “Consumer RGB-D Cameras and their Applications,” University of California, Riverside, Tech. Rep., 2012. [Online]. Available: http://alumni.cs.ucr.edu/ klitomis/files/RGBD-intro.pdf

[10] Microsoft, “Coordinate mapping,” 2014. [Online]. Available: https://docs.microsoft.com/ en-us/previous-versions/windows/kinect/dn785530(v=ieb.10)

[11] N. V. Oosterwyck, “Repository Master Project,” 2018. [Online]. Available: https: //github.com/NickVanOosterwyck/Master-Project

[12] F. Flacco, T. Kroeger, A. De Luca, and O. Khatib, “A Depth Space Approach for Evaluating Distance to Objects: with Application to Human-Robot Collision Avoidance,” Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 80, pp. 7–22, 2015.

[13] L. Li, "Time-of-Flight Camera An Introduction," Texas Instruments - Technical White Paper, no. January, p. 10, 2014

[14] M. Hansard, S. Lee, O. Choi, R. Horaud, M. Hansard, S. Lee, O. Choi, R. Horaud, and F. Cameras, Time of Flight Cameras : Principles , Methods , and Applications. Seoul: HAL, 2012.

[15] D. Pagliari and L. Pinto, "Calibration of Kinect for Xbox One and comparison between the two generations of microsoft sensors," Sensors (Switzerland), vol. 15, no. 11, pp. 27 569– 27 589, 2015.

[16] MathWorks, "Skeleton Viewer for Kinect V2 Skeletal Data," 2018. [Online]. Available: https://nl.mathworks.com/help/supportpkg/kinectforwindowsruntime/examples/plot-skeletons-with-the-kinect-v2.html

[17] K.-Y. Yeung, T.-H. Kwok, and C. C. L. Wang, "Improved Skeleton Tracking by Duplex Kinects: A Practical Approach for Real-Time Applications," Journal of Computing and Information Science in Engineering, vol. 13, no. 4, p. 041007, 2013. [Online]. Available: http://computingengineering.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4025404

[18] C. Lenz, M. Grimm, T. R¨oder, and A. Knoll, "Fusing multiple Kinects to survey shared Human-Robot-Workspaces," Technische Universit¨at M¨unchen, Munich, Germany, Tech. Rep. TUM-I1214, 2012.

[19] M. W. Spong, S. Hutchinson, and M. Vidyasagar, "FORWARD KINEMATICS: THE DENAVIT-HARTENBERG CONVENTION BT - Robot Dynamics and Control," Robot Dynam-

ics and Control, pp. 1–32, 2003. [Online]. Available: papers2://publication/uuid/E7AC27CB-8AAC-4755-88F2-002FB8929672

[20] S. Kucuk and Z. Bingul, Industrial Robotics: Theory, Modelling and Control, Sam Cubero (Ed.), 2006, no. December.

[21] J. Kay, "Introduction to Homogeneous Transformations and Robot Kinematics," Rowan University Computer Science Department, no. January, pp. 1–25, 2005.[Online]. Available: http://scholar.google.com/scholar?hl=enbtnG=Searchq= intitle:Introduction to Homogeneous Transformations Robot Kinematics0

[22] F. Flacco, T. Kroeger, A. De Luca, and O. Khatib, "A Depth Space Approach for Evaluating Distance to Objects: with Application to Human-Robot Collision Avoidance," Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 80, pp. 7–22, 2015.

[23] T. KEREZOVIC, G. SZIEBIG, B. SOLVANG, and T. LATINOVIC, "Human Safety in ´ Robot ApplicationsReview of Safety Trends," Acta Technica Corviniensis, vol. 7, no. August 2014, pp. 113–118, 2013. [Online]. Available: http://acta.fih.upt.ro/pdf/2013-4/ ACTA-2013-4-20.pd

[24] N. Pedrocchi, F. Vicentini, M. Malosio, and L. M. Tosatti, "Safe human-robot cooperation in an industrial environment," International Journal of Advanced Robotic Systems, vol. 10, 2013.

[25] J. J. Craig, "Introduction to Robotics: Mechanics and Control 3rd," Prentice Hall, vol. 1, no. 3, p. 408, 2004.

[26] I. Bonev, “Universal Robots Inverse kinematics 8 solutions,” 2014. [Online]. Available: https://twitter.com/universalrobot/status/516682281155829760

[27] K. P. Hawkins, “Analytic Inverse Kinematics for the Universal Robots,” pp. 1–5, 2013. [Online]. Available: http://hdl.handle.net/1853/50782

Appendix A

User Manual

Appendix A

Cobot Supplementary Information

An appendix for speed and separation monitoring in a collaborative robot (Cobot) system typically includes guidelines, procedures, and technical specifications to ensure safe and efficient operation. All tests are performed on the set-up in the robotics lab and simulator as shown in Figure A.1

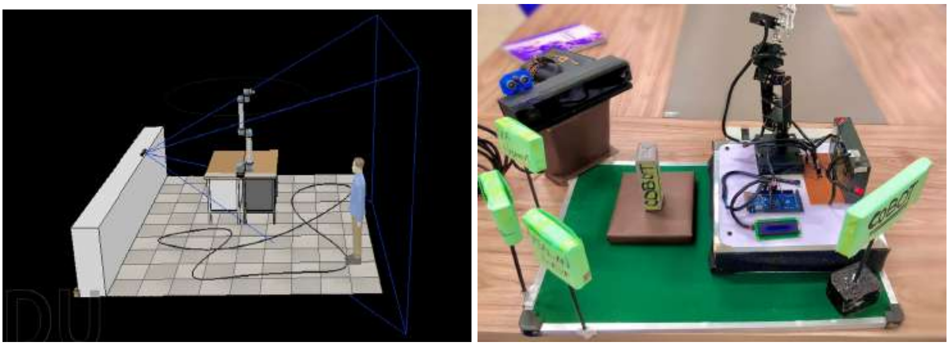


Figure A.1: Illustration of VREP and Hardware Setup

A.1 Purpose

The purpose of this appendix is to outline the requirements and recommendations for speed and separation monitoring in a collaborative robot (Cobot) system. The objective is to maintain safe interaction between human operators and the Cobot while ensuring efficient operation.

A.2 Speed Monitoring

a. Maximum Speed Limit: Define the maximum permissible speed for the Cobot during operation, considering the task requirements, payload, and workspace layout. This limit should prevent any potential harm to human operators in case of accidental collisions or unintended contact.

b. Speed Control Mechanisms: Describe the mechanisms employed to monitor and control the Cobot's speed, such as speed sensors, encoders, or software algorithms. These mechanisms should ensure that the Cobot operates within the defined speed limits.

c. Emergency Stop Functionality: Specify the emergency stop functionality that immediately halts the Cobot's motion in case of any unexpected situations or hazards. Outline the procedures for testing and maintaining the emergency stop system.

A.3 Separation Monitoring

a. Minimum Separation Distance: Define the minimum required separation distance between the Cobot and human operators during operation. Consider factors such as the Cobot's reach, workspace layout, and potential hazards to determine a safe distance.

b. Proximity Sensors: Describe the proximity sensors utilized to monitor the distance between the Cobot and human operators. These sensors should detect any breach of the defined separation distance and trigger appropriate warning signals or safety measures.

c. Warning Systems: Explain the warning systems employed to alert human operators when they approach the defined separation distance. This may include visual indicators, audible alarms, or haptic feedback to ensure timely and effective communication.

A.4 Training and Procedures

a. Operator Training: Emphasize the importance of providing comprehensive training to all operators working with the Cobot system. Training should cover topics such as safe operating procedures, understanding speed and separation limits, emergency response protocols, and proper use of personal protective equipment (PPE).

b. Standard Operating Procedures (SOPs): Develop SOPs that outline the step-by-step procedures for operating the Cobot system, including speed and separation monitoring. SOPs should cover routine operations, maintenance tasks, and emergency situations, ensuring a standardized and safe approach.

A.5 Maintenance and Testing

a. Regular Inspections: Establish a schedule for regular inspections to verify the proper functioning of speed and separation monitoring systems. Inspections may include checks of sensors, emergency stop functionality, and warning systems.

b. Calibration and Adjustment: Specify the requirements for calibration and adjustment of speed and separation monitoring systems to maintain accuracy and reliability.

c. Testing and Validation: Define procedures for conducting periodic tests to ensure the effectiveness of speed and separation monitoring mechanisms. These tests may involve simulated scenarios or real-world trials to validate the system's performance.