

Ain Shams University  
Faculty of Engineering

**Autonomous & Manual Controlled Upper Humanoid Robot**

Submitted by :

|  |  |
| --- | --- |
| Robeir Remon Farid | 1500594 |
| Rami Wafik Attia |  |
| Hassan Sami Fahmy |  |
| Alaa Ayman Elremaily |  |
| Aya Ayman Elremaily |  |

Supervised by :

**Dr. Mohammed Ibrahim Dr. Shady Ahmed Maged**

Table of Contents

[1. INTRODUCTION 4](#_Toc41713181)

[1.1. BACKGROUND 4](#_Toc41713182)

[1.2. HISTORY BRIEF 5](#_Toc41713183)

[1.3. MOTIVATION 7](#_Toc41713184)

[1.4. AIMS 8](#_Toc41713185)

[1.5. OBJECTIVES 8](#_Toc41713186)

[2. SYSTEM OVERVIEW 9](#_Toc41713187)

[2.1. SYSTEM SCHEMATIC 9](#_Toc41713188)

[2.2. HUMAN MOTION 9](#_Toc41713189)

[2.3. VDI ENGINEERING DESIGN THEORY 10](#_Toc41713190)

[3. CONCEPTUAL DESIGN 11](#_Toc41713191)

[3.1. INTRODUCTION 11](#_Toc41713192)

[3.2. DESIGN CONSIDERATION 11](#_Toc41713193)

[3.3. REQUIREMENTS 11](#_Toc41713194)

[4. INMOOV 11](#_Toc41713195)

[4.1. INTRODUCTION 11](#_Toc41713196)

[4.2. MECHANICAL 11](#_Toc41713197)

[4.2.1. MECHANICAL DESIGN 11](#_Toc41713198)

[4.3. URDF MODEL 11](#_Toc41713199)

[4.4. ELECTRICAL 11](#_Toc41713200)

[4.4.1. CIRCUIT DESIGN 11](#_Toc41713201)

[4.4.2. PCB LAYOUT 11](#_Toc41713202)

[4.4.3. WIRING 11](#_Toc41713203)

[4.4.4. CIRCUIT SCHEMATIC 11](#_Toc41713204)

[4.4.5. POWER CALCULATION 11](#_Toc41713205)

[4.5. SENSOR SELECTION 11](#_Toc41713206)

[4.6. GPU SELECTION (NVIDIA JETSON NANO DEVELOPMENT KIT “B01”) 11](#_Toc41713207)

[5. MANUAL ARM CONRTOL 13](#_Toc41713208)

[5.1. INTRODUCTION 13](#_Toc41713209)

[5.2. EXOSKELETON 13](#_Toc41713210)

[5.2.1. ARM CONTROL 13](#_Toc41713211)

[5.2.2. FINGERS CONTROL 13](#_Toc41713212)

[5.3. WIRELESS JOYSTICK 13](#_Toc41713213)

[5.4. KINECT CONTROL 13](#_Toc41713214)

[5.5. HEAD CONTROL USING VR 13](#_Toc41713215)

[6. AUTONOMOUS ARM CONTROL 13](#_Toc41713216)

[6.1. BALL CATCHING 13](#_Toc41713217)

[6.1.1. INTRODUCTION 13](#_Toc41713218)

[6.1.2. BALL DETECTION AND TRACKING 14](#_Toc41713219)

[6.1.3. TRAJECTORY PATH ESTIMATION 16](#_Toc41713220)

[6.1.4. INVERSE KINEMATICS SOLVING 17](#_Toc41713221)

[6.1.5. CONNECTING JETSON NANO WITH SERVOS 17](#_Toc41713222)

[6.2. OBJECT DETECTION AND TRACKING USING NEURAL NETWORK 17](#_Toc41713223)

[6.2.1. INTRODUCTION 17](#_Toc41713224)

[6.2.2. DETECT PERSON AND DOG TESTING 17](#_Toc41713225)

# INTRODUCTION

## BACKGROUND

Humanoid robots, while being one of the smallest groups of service robots in the current market, have the greatest potential to become the industrial tool of the future.

It has been well documented that there will be increase in the number of robots over the next decade. According to the Boston Consulting Group, by 2025, robots will perform 25% of all labor tasks. This is due to improvements in performance and reduction in costs.

Before the coronavirus pandemic and the economic uncertainty, Statistics Market Research Consulting expected that the Global Humanoid Robot Market would reach $13 billion by 2026. While future market behavior is now unclear, robot usage is on the rise: Chinese companies were rushing to deploy robots and automation technology, as doctors were grappling with COVID-19.

Developing a humanlike robot involves copying the appearance of humans as well as emulating the capabilities, expression of emotions and possibly even having thoughts. Making such robots involves advances in many disciplines including mechanical and electrical engineering, materials science, computer science, artificial intelligence, and control. To make such smart machines that look and act like a human there is a need to integrate capabilities that are at the cutting edge of the related technology. The materials to be used need to be resilient, lightweight and multifunctional.

These robots need sensors to visualize the train, hear sound, as well as sense touch, pressure and temperature. The robots need to use light batteries or generator for power that can be operated over a long time without recharge. In addition, the robots need to interpret the information that is measured by the sensors to perceive and be aware of the surrounding terrain and to sense hazards and risks. Humanlike robots need to have effective control and artificial intelligence algorithms in order to be operated like humans and interact with its environment and humans.

Developing a humanlike robot involves copying the appearance of humans as well as emulating the capabilities,

expression of emotions and possibly even having thoughts. Making such robots involves advances in many

disciplines including mechanical and electrical engineering, materials science, computer science, artificial

intelligence, and control. To make such smart machines that look and act like a human there is a need to integrate

capabilities that are at the cutting edge of the related technology [Bar-Cohen and Breazeal, 2003; Bar-Cohen and

Hanson, 2009]. The materials to be used need to be resilient, lightweight and multifunctional. Further, the mobility

system that is used needs to allow walking via two legs and maintain stability while able to traverse complex terrains

(e.g. climb stairs and avoid obstacles). These robots need sensors to visualize the train, hear sound, as well as sense

touch, pressure and temperature. The robots need to use light batteries and/or generator for power that can be

operated over a long time without recharge. In addition, the robots need to interpret the information that is measured

by the sensors to perceive and be aware of the surrounding terrain and to sense hazards and risks. Humanlike robots

need to have effective control and artificial intelligence algorithms in order to be operated like humans and interact

with its environment and humans [Plantec, 2003]. The produced robots need to have body parts and related

functions similar to a human as much as possible

machines were driven by mechanical energy that was stored in a spring. Examples include the "The Flute Player"

that was produced by the French engineer Jacques de Vaucanson in 1737 and the “Writer” that was made by the

Swiss clockmaker Jacquet-Droz and completed in 1772. The era of robotics as we know it today where the machine

is equipped with artificial intelligence has began in 1946 with the first introduction of the digital computer, the

ENIAC computer, which was the first large-scale general-purpose electronic computer [McCartney 1999]. The first

time that the possibility of building thinking and learning machines was raised was in 1950 [Turing, 1950]. Progress

in developing powerful microprocessors with high computation speed, very large memory, wide communication

bandwidth, and more effective software tools made the most impact on the development of intelligent robots. With

the advancements in microelectronics and intelligent software more sophisticated robots have been emerging with

concepts and methodologies that are inspired and guided by nature [Arkin 1998; Bar-Cohen and Breazeal, 2003;

Bar-Cohen and Hanson, 2009; Gould, 1982].

With the advances in technology, humanlike robots are increasingly becoming easier to make as lifelike using

effective autonomous operation algorithms, humanlike materials, and the capability to emulate the movement and

functionality (seeing, hearing, smelling, etc.) of humans. Using state of the art microprocessors, materials, sensors,

software, and many other technologies are leading to increasingly more capable robots. These advances are

allowing them to perceive, interpret, respond, and adapt to their environment. Robotic products are already being

developed for entertainment, education, healthcare, home security, military, and many others. Currently,

entertainment applications are the most beneficiary of this technology where humanlike robotic toys are

commercially available in many stores. Further, industry has begun to collaborate with scientists to make their

characters in movies appear more realistic and to move more like people. Also, robotics researchers are increasingly

collaborating with artists to make their robots appear more expressive and believable.

3. MAKING A HUMANLIKE ROBOT

Developing a humanlike robot involves copying the appearance of humans as well as emulating the capabilities,

expression of emotions and possibly even having thoughts. Making such robots involves advances in many

disciplines including mechanical and electrical engineering, materials science, computer science, artificial

intelligence, and control. To make such smart machines that look and act like a human there is a need to integrate

capabilities that are at the cutting edge of the related technology [Bar-Cohen and Breazeal, 2003; Bar-Cohen and

Hanson, 2009]. The materials to be used need to be resilient, lightweight and multifunctional. Further, the mobility

system that is used needs to allow walking via two legs and maintain stability while able to traverse complex terrains

(e.g. climb stairs and avoid obstacles). These robots need sensors to visualize the train, hear sound, as well as sense

touch, pressure and temperature. The robots need to use light batteries and/or generator for power that can be

operated over a long time without recharge. In addition, the robots need to interpret the information that is measured

by the sensors to perceive and be aware of the surrounding terrain and to sense hazards and risks. Humanlike robots

need to have effective control and artificial intelligence algorithms in order to be operated like humans and interact

with its environment and humans [Plantec, 2003]. The produced robots need to have body parts and related

functions similar to a human as much as possible

Developing a humanlike robot involves copying the appearance of humans as well as emulating the capabilities,

expression of emotions and possibly even having thoughts. Making such robots involves advances in many

disciplines including mechanical and electrical engineering, materials science, computer science, artificial

intelligence, and control. To make such smart machines that look and act like a human there is a need to integrate

capabilities that are at the cutting edge of the related technology [Bar-Cohen and Breazeal, 2003; Bar-Cohen and

Hanson, 2009]. The materials to be used need to be resilient, lightweight and multifunctional. Further, the mobility

system that is used needs to allow walking via two legs and maintain stability while able to traverse complex terrains

(e.g. climb stairs and avoid obstacles). These robots need sensors to visualize the train, hear sound, as well as sense

touch, pressure and temperature. The robots need to use light batteries and/or generator for power that can be

operated over a long time without recharge. In addition, the robots need to interpret the information that is measured

by the sensors to perceive and be aware of the surrounding terrain and to sense hazards and risks. Humanlike robots

need to have effective control and artificial intelligence algorithms in order to be operated like humans and interact

with its environment and humans [Plantec, 2003].

Developing a humanlike robot involves copying the appearance of humans as well as emulating the capabilities,

expression of emotions and possibly even having thoughts. Making such robots involves advances in many

disciplines including mechanical and electrical engineering, materials science, computer science, artificial

intelligence, and control. To make such smart machines that look and act like a human there is a need to integrate

capabilities that are at the cutting edge of the related technology [Bar-Cohen and Breazeal, 2003; Bar-Cohen and

Hanson, 2009]. The materials to be used need to be resilient, lightweight and multifunctional. Further, the mobility

system that is used needs to allow walking via two legs and maintain stability while able to traverse complex terrains

(e.g. climb stairs and avoid obstacles). These robots need sensors to visualize the train, hear sound, as well as sense

touch, pressure and temperature. The robots need to use light batteries and/or generator for power that can be

operated over a long time without recharge. In addition, the robots need to interpret the information that is measured

by the sensors to perceive and be aware of the surrounding terrain and to sense hazards and risks. Humanlike robots

need to have effective control and artificial intelligence algorithms in order to be operated like humans and interact

with its environment and humans [Plantec, 2003]

The main applications for which these robots are being considered include health-care, entertainment, home or office security, and military. Humanoids are also suitable for some procedurally-based vocations such as operating like hospital workers, receptionists, guards, and more, and they can speak in various languages, dance to the sound of music, and play musical instruments.

Companies like [**Softbank Robotics**](https://www.softbankrobotics.com/us/) have created human-looking robots to be used as medical assistants and teaching aids. Currently, humanoid robots are excelling in the medical industry, especially as companion robots.

However, companies are now using humanoid robots to fill engineering tasks. A four-year joint research project was conducted by [**Joint Robotics Laboratory**](https://jrl-umi3218.github.io/index.html)and [**Airbus Group**](http://www.airbus.com/) to use humanoid robotic technology in aircraft manufacturing facilities. By using humanoid robots on aircraft assembly lines, Airbus looks to relieve human operators of some of the more laborious and dangerous tasks. The human employers could then concentrate on higher value tasks. The primary difficulty is the confined spaces these robots have to work in and being able to move without colliding with the surrounding objects

## HISTORY BRIEF

1967



Waseda University started the WABOT project in 1967. The WABOT-1 robot was completed in 1972 and was the world’s first full-scale android humanoid robot. It was the first robot able to walk and communicate with people in Japanese, navigate a room and grip and transport objects. They later went on to create WABOT-2 which was capable of reading a musical score and playing an electric keyboard.

2010



[NASA and General Motors revealed Roboaut 2](https://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-robonaut-58.html), a highly advanced humanoid robot that was part of the Discovery shuttle launch in 2011. Robonaut was designed to assist NASA with space walks and has enough dexterity to use tools and work alongside astronauts in future space expeditions.

NASA later went on to develop [Valkyrie](http://valkyrie.inf.ed.ac.uk/), a robot that has been developed with the ability to ultimately setup habitats on Mars prior to human arrival.

2014

[Softbank Robotics release Pepper](https://www.softbankrobotics.com/emea/en/pepper), which quickly became the leading commercially available social robot. Pepper was rolled out in Softbank’s mobile stores in Japan and has since been use in Carrefour and Renault stores across France.

2016

[Hanson Robotics release Sophia](https://www.hansonrobotics.com/sophia/), a social robot with silicone skin, and the ability to interact with people and display more than 50 facial expressions. Sophia has been covered my media around the world and has participated in many high-profile interviews. The Sophia robot is

also the first ever robot to be granted citizenship of a country.

2020

At a time when healthcare workers at hospitals are prone to coronavirus infection, humanoid robots at AIIMS has mitigated risk by performing contactless monitoring of patients

Milagrow human tech has installed humanoid Milagrow ELF at dedicated COVID-19 wards.



## MOTIVATION

Our motivation for this project is to advance research in this area as it is needed for our country at this moment especially at the coronavirus pandemic because as the virus spreads to the rest of the world, robots are being deployed in many countries.

This project can be used for research and space exploration, personal assistance and caregiving, education and entertainment, search and rescue, manufacturing and maintenance, public relations, and healthcare.

Also, to advance our abilities in the control theory, machine vision and artificial intelligence to be graduated as a modern mechatronics engineers, this project was perfect as it accomplishes this goal.

## AIMS



The virtual presence robot can be controlled by a person who is far away from the robot which is used in education and medical use but the person can only see the environment , talk using the tablet and move the robot base but unfortunately he can’t take any action in the robot environment so our aim is to design, manufacture and control a robot that mimics human appearance which can be controlled from a remote location instantly providing a virtual presence and also can perform some human functions like object detection and catching objects on the fly taking into consideration the safety aspect as it will be working in the vicinity of humans which will be used exactly like the old virtual presence design robot also with more options and functions.

## OBJECTIVES

Our objective is to provide a relatively cheap humanoid robot with a differential base structure. It can be used among various medical and social or educational applications so that it can be used as a nurse for people infected with coronavirus and also can be used for catching flying objects.

# SYSTEM OVERVIEW

## SYSTEM SCHEMATIC

## HUMAN MOTION

Human hand motion during point-to-point reaching was initially observed as straight lines in the task  
space, though this was later disputed by others, who observed curved trajectories. It has been proposed  
that the straight lines were in fact the result of experimental protocols in which the subjects were asked  
to manipulate an instrumented tool to facilitate data capture. Other researchers have suggested that the  
curved motions are due to gravitational effects. It has been observed during this literature review that in  
many of the straight-line motion cases, subjects had their arm movement (including redundant DOF)  
constrained to the horizontal plane, in which changes in gravitational effects are largely negated.

Optimization has been widely accepted as a driver of human motion, though the cost function has been  
disputed. It seems that a number of different cost functions have produced valid results in various scenarios. These seem to evolve over the course of research from purely kinematic considerations (often  
at the joint level) to dynamic quantities, such as muscle forces or change in joint torque.

Though muscle models are often considered as viable aspects of movement cost functions, their high  
contribution to increased model complexity and redundancy means that many researchers only  
implement highly simplified muscle representations in their simulations, in which the muscle quantity and  
placement do not match those of a human. The inclusion of even these factors adds a great deal of  
additional redundancy to the control of a limb.

It is also true that researchers tend to rely on simplified kinematic models of the arm, focusing often on  
2DOF or 3DOF planar systems. Some examples have included three-dimensional movements achieved  
with a 4DOF model of the human arm. Many studies have constrained human motion to planar horizontal  
tasks involving the manipulation of some tool. Generalized movement models have then been based on  
data collected in these specific, gravity-negated scenarios. Modern models that include dynamics and  
kinematics seem much more appealing than pure kinematic models, especially considering the results of  
neurological movement analysis.

There is evidence of both kinematic and dynamic limb and external object representations (models) within  
the human central nervous system. It has been proposed that after periods of learning, where these  
models are created or adapted, the mechanisms of the brain employ feedforward control schemes that  
utilize these models to produce movement abilities that would not be possible with the slow feedback  
mechanisms of the body.

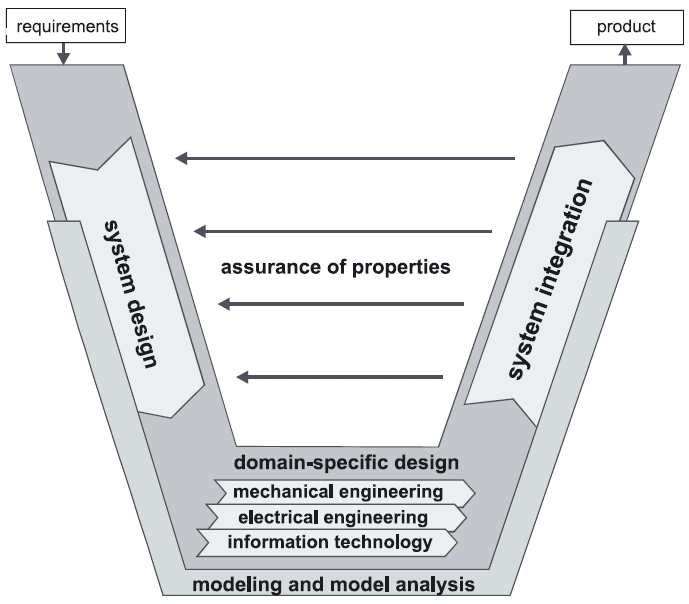
Considering the number of factors that influence human motion, it is proposed that non-exclusively planar  
motion paths are influenced by several aspects, including solving the redundancy problem by optimization of some physiological quantity with additional dependency on external dynamic factors, such as change of gravitational forces and direction of movement [2]

## VDI ENGINEERING DESIGN THEORY

The German Society for Engineers has published a guideline for mechatronic systems,VDI 2206:2004, which includes a ‘V-model’ of design development by implication, the procedural models of VDI 2225:1975, VDI 2222:1977 and VDI 2221:1985 are included in the ‘domain specific design’. Blanchard [2004] shows a similar model with respect to software systems. We claim a similarity to the Procedural Model of Design Engineering [Hubka 1992b, 1992a, 1996]:

The ‘domain-specific design’ is represented by separate functions in the function structure, which may specify functions that can be realized by mechanical, electrical, chemical, software, or any other system.

‘integration’ can and should take place in any of the relevant structures, but is especially necessary in the constructional structure because cooperation among the specialists is especially necessary here, and The cycle of ‘substantiate, verify, improve’ at the end of each design stage in the Procedural Model [Hubka 1992a, 1992b, 1996] leads to a feedback to any previous stage, not just to the horizontally referenced level, although this level may be the most likely target.



# CONCEPTUAL DESIGN

## INTRODUCTION

## DESIGN CONSIDERATION

## REQUIREMENTS

# INMOOV

## INTRODUCTION

## MECHANICAL

### MECHANICAL DESIGN

#### UPPER PART

#### LOWER PART (BASE DESIGN)

## URDF MODEL

## ELECTRICAL

### CIRCUIT DESIGN

### PCB LAYOUT

#### HAND PCB LAYOUT

#### WRIST PCB LAYOUT

#### SHOULDER AND ELBOW PCB LYOUT

#### HEAD PCB LAYOUT

#### VOLTAGE REGULATING PCB LAYOUT

#### FINAL PCB LAYOUT

### WIRING

### CIRCUIT SCHEMATIC

### POWER CALCULATION

## SENSOR SELECTION

## GPU SELECTION (NVIDIA JETSON NANO DEVELOPMENT KIT “B01”)

Most of the Autonomous and Artificial Intelligence (AI) systems are now using graphics processing unit (GPU) to make the system faster in processing and for our project which is autonomous we decided to choose the cheapest GPU made by Nvidia but it is still more expensive than the Raspberri Pi so we decided to make a comparison between Raspberry pi and Nvidia Jetson Nano and we conclude that because of the heating of raspberry pi during work and there is a heat sink in jetson nano and also its much more faster due to the GPU and RAM we chose the Nvidia Jetson Nano Development Board.

Comparison between Nvidia Jetson Nano and Raspberry pi

|  |  |  |
| --- | --- | --- |
| Point of comparison | Nvidia Jetson Nano Dev Board | Raspberry pi 3 |
| CPU | 1.43 GHz 64-bit quad-core ARM Cortex –A57 | 1.4 GHz 64-bit quad-core ARM Cortex –A53 |
| GPU | 128-Core Nvidia Maxwell | Broadcom Video Core IV |
| GPIO Header | 40-pin | 40-pin |
| RAM | 4GB LPDDR4 | 512 MB LPDDR2 SDRAM |

NVIDIA Jetson Nano enables the development of millions of new small, low-power AI systems. It opens new worlds of embedded IoT applications, including entry-level Network Video Recorders (NVRs), home robots, and intelligent gateways with full analytics capabilities.

NVIDIA Jetson uses AI to power the future of robotics, intelligent video analytics, The World's Smallest AI Supercomputer for Embedded and Edge Systems.



A NEW DIMENSION IN AI

At just 70 x 45 mm, the Jetson Nano module is the smallest Jetson device. This production-ready System on Module (SOM) delivers big when it comes to deploying AI to devices at the edge across multiple industries—from smart cities to robotics.

BIG COMPUTE PERFORMANCE

Jetson Nano delivers 472 GFLOPs for running modern AI algorithms fast. It runs multiple neural networks in parallel and processes several high-resolution sensors simultaneously, making it ideal for applications like entry-level Network Video Recorders (NVRs), home robots, and intelligent gateways with full analytics capabilities.

SMALL POWER DEMANDS

Jetson Nano frees you to innovate at the edge. Experience powerful and efficient AI, computer vision, and high-performance computing at just 5 to 10 watts.

# MANUAL ARM CONRTOL

## INTRODUCTION

## EXOSKELETON

### ARM CONTROL

### FINGERS CONTROL

## WIRELESS JOYSTICK

## KINECT CONTROL

## HEAD CONTROL USING VR

# AUTONOMOUS ARM CONTROL

## BALL CATCHING

### INTRODUCTION

For centuries, humans have been fascinated by the idea of creating machines similar to existing life. Although robots have replaced the human worker at repetitive tasks in industrial scenarios for many years, they do not have yet reached the capability and autonomy to accomplish everyday human tasks. This becomes even clearer when considering what variety of dynamic activities humans are able to perform, e. g. when doing sports.

Catching a thrown ball with a hand is not easy – neither for humans nor for robots . It demands for a tight interplay of skills in mechanics, control, planning and visual sensing to reach the necessary precision in space and time. Because of this, ball catching has been used for almost 20 years now as a challenging benchmark system to develop and test robotics key technologies

In all the works the general setup is in principle the same: a stereo vision system tracks the ball and predicts the balls trajectory, then the point and time, where and in which orientation the robot should  
intercept the ball on its trajectory, is determined. Next, the robot configuration to reach the catch point is computed and finally a path is generated, which brings the robot from its start configuration to the desired catch configuration.

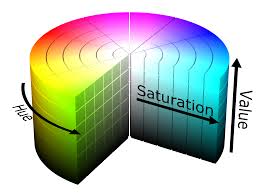
### BALL DETECTION AND TRACKING

#### BALL DETECTION USING OPENCV

The first part of this task is to detect the place of the ball in the image and tracking its place as it’s moving in the image so we used the opencv library which is an open source library using a computer vision techniques.

The first step is to change the RGB image into HSV image (Hue , Saturation ,Value) which is better in detecting colors in an image .

The HSV color wheel sometimes appears as a cone or cylinder, but always with these three components:



HUE

Hue is the color portion of the model, expressed as a number from 0 to 360 degrees:

**Red** falls between 0 and 60 degrees.

**Yellow** falls between 61 and 120 degrees.

**Green** falls between 121 and 180 degrees.

**Cyan** falls between 181 and 240 degrees.

**Blue** falls between 241 and 300 degrees.

**Magenta** falls between 301 and 360 degrees.

SATURATION

Saturation describes the amount of gray in a particular color, from 0 to 100 percent. Reducing this component toward zero introduces more gray and produces a faded effect. Sometimes, saturation appears as a range from 0 to 1, where 0 is gray, and 1 is a primary color.

VALUE (OR BRIGHTNESS)

Value works in conjunction with saturation and describes the brightness or intensity of the color, from 0 to 100 percent, where 0 is completely black, and 100 is the brightest and reveals the most color.

So to detect a ball in an image we have to set a lower and upper boundaries for the color of our ball’s color.

Second part is to draw a contour on the detected parts of the same color we have just defined to track the ball in the next frames to detect it in a real time video and also to calculate the centroid of that contour to use it in calculations and predict the path of the ball.

Contour : Edge (Contour) Detection is an image processing technique for finding the boundaries of objects within images.

Contours are a curve joining all the continuous points along the boundary , having same color or intensity.

Algorithm for finding the contours is by using “cv.findContours()” applied on binary image, then process the contours (find its areas ,enclosing circle and centroid)

Third part is to solve a problem of the noises to prevent finding objects in the frame with the same colors we have just declared so we can find the area of all contours using cv.contourArea() and choose to track the biggest area only.

Fourth part is to draw the ball contour around our object which is the ball using :

cv.drawContour(image , contours , index , color , thickness)

index : the index no. of the contours found if index =-1 then draw all contours, and color is the color of the contour line .

Then the Last step is to get\_contour\_center : to use it to find the pos and velocity of the ball in next steps of predicting the path of the ball.

#### CALCULATING THE DEPTH OF THE BALL

After detecting the ball and tracking it in the video we have to know the real depth of the ball from the camera in cm.

So we found that there is only two methods to find the depth : \_  
First method is to use a stereo vision technique using two or more web cams and also there are two ways to find depth by using two or more cams which are using the depth map and the triangulate technique.

Second is using a stereo camera which has an IR emitters and receivers to detect the depth of an object from the camera e.g. Microsoft Xbox 360 Kinect Sensor

##### USING TWO WEBCAMS BY TRANGULATE METHOD

##### USING KINECT SENSOR

### TRAJECTORY PATH ESTIMATION

#### USING EULER METHOD

Euler's method approximates an exact solution to a differential equation as an iterative algebraic equation. Look at it like this - given a differential equation like

**dx / dt = x2 −t**

We can loosen the definition of d = lim (Δ→0) Δ and instead revert back to using Δ with a nonzero limit and refer to it as the step size - by how much are we jumping in time between iterations.

**(dx / dt) ≈ Δx / Δt = xi+1 − xi / Δt**

And for reference

**d2x / dt2 ≈( xi+1 − 2xi + xi−1 ) / Δt2**

Using the first, we can approximate the solution to the differential equation

**( xi+1 − xi  / Δt ) = x2i − ti ⟹ xi+1 = xi + Δt (x2i − ti)**

Notice that to begin this iterative process, we first need to know the initial condition to calculate the next position. Likewise, using the second derivative approximation requires two initial conditions.

This is extremely useful when plotting solutions/gathering data to differential equations that are difficult to solve, as it approximates the solution one step at a time.

In Our application we have our centroid of our object in each frame so by using the distance and velocity bet each two frames we can use the diff equations to predict the trajectory path.



Figure Test Path Estimation Juggling 1 Figure Test Path Estimation Juggling 2



Figure Test Path Estimation Juggling 3 Figure Test Path Estimation Juggling 4



Figure Test Path Estimation Juggling 5 Figure Test Path Estimation Catching

### INVERSE KINEMATICS SOLVING

#### MOVEIT

#### VREP

#### GRAPHICAL METHOD

### CONNECTING JETSON NANO WITH SERVOS

## OBJECT DETECTION AND TRACKING USING NEURAL NETWORK

### INTRODUCTION

### DETECT PERSON AND DOG TESTING