Fabrication of Voice Controlled Robotic ARM

V Mounika[1], Chikkam Anandini [2], V Hari Kiran[3]
[1]*Assistant Professor, Department of Mechanical Engineering, Avanthi Institute of Engineering and Technology, Makavarapalem, Anakapalli District, Andhra Pradesh, India-531113*
[2]*Student, Department of Mechanical Engineering, Andhra University, Visakhapatnam, Visakhapatnam District, Andhra Pradesh, India -530003*
[3]*Associate Professor, Avanthi Institute of Engineering and Technology, Makavarapalem, Anakapalli District, Andhra Pradesh, India-531113*

Abstract: **This work "voice controlled robotic arm" was designed and developed in such way that the arm movement has four degrees of freedom and is controlled by the voice commends using an opensource android mobile application. The arm control board consists of Arduino [2] uno, PCA9685 16 channel servo driver and trans receiver Bluetooth module. The movement of arm is facilitated by the four servo motors connected with servo driver at the receiver side. The voice commands from the application is converted into digital signals by blue tooth RF transmitter for an appropriate range of 100 meters to the robot arm. At the receiver end data gets decoded by the blue tooth trans receiver and is fed into micro controller which performs the required task by the listening to the voice commands of the user. The software code is written in embedded C and uploaded to micro controller. The kinematic structure of robot arm 3D CAD model has been done in 3D CAD software (Auto cad).All CAD models are 3D printed with PLA material, assembled and tested. The kinetics and analysis of the structural components has to be carried out for further evaluation. This robotic arm is used in various industrial applications such as handling of hazardous materials in the areas where the greatest imminent threat to humans.**

Key Words: voice-controlled robot, Arduino uno, PCA9685, 3D CAD, servo driver, hazardous material

INTRODUCTION

Robotic is defined as the study, design and use of robotic systems for manufacturing. With the rise in manufacturing industrial activities, a robotic arm[7] is invented to help various industries to perform a task or work instead of using manpower [1].Robots are generally used to perform unsafe, hazardous, highly repetitive, and unpleasant tasks. Robot can perform material handling[21], assembly, arc welding, resistance welding, machine tool load and unload function, painting and spraying, etc. It is very useful because it possesses high precision, intelligence and endless energy levels in doing work compared to human being. For an example, a robotic arm is widely used in the assembling or packing line by lifting the small objects with repetitive motion that human couldn't bear to do in a long period of time. The light material lifting task can be done by the robotic arm efficiently and time-saving because it is not restricted by fatigue or health risks which man might experience.

An industrial robot is a re-programmable multifunctional manipulator design to move material, parts, tools, or specialized devices through variable programmed motion for performance of a variety of tasks. This is the definition from the Robot Institute of America to reflect main features of modern robot systems. An industrial robot system can include any devices or sensors together with the industrial robots to perform its tasks as well as sequencing or monitoring communication interfaces.

PARTS OF THE ROBOT ARM IN FABRICATION:

The various parts considered for assembly are:

1. Servo motor mg946r – 3 no's
2. Arduino Uno – micro controller
3. Voltage regulator – 12 volts to 5 volts
4. Terminal Board
5. Arms
6. Servo Drive
7. The Stepper motor with pulley
8. G2T IDLER Pulley
9. G2T Timing belt
10. Linear motion bearing
11. Stepper driver module
12. AVG 28 Silicon wire

13. Smooth Stainless Steel rod
14. Frame base

Servo motor mg946R:
MG946R Towerpro Digital Metal Servo 13KG High torque. It is upgraded version of TowerPro MG945. The new PCB and IC control system which makes it more accurate. Its internal gearing and motor are also upgraded to improve dead bandwidth and centering. MG946R is for 1/8 buggy monster.

Figure 1 : Servo motor mg946R

Stepper Driver module:
Stepper motors are DC brushless and synchronous motors. They rotate in discrete steps of predefined values and are able to rotate both clockwise and anticlockwise. Unlike other DC motors, they provide a precise position control according to the number of steps per revolution for which the motor is designed. That means a complete revolution of a stepper motor is divided into a discrete number of steps. They are commonly used in CNC machines, Robotics, 2D and 3D printers. The DRV8825 driver module which is used to control a stepper motor in a relatively simple manner. Using only two pins of Arduino and DRV8825 driver module, we can control the speed of the rotation as well as the direction of rotation of a stepper motor.

Figure 2: Stepper Driver DRV

G2T Idler Pulley:
In a belt drive system, idlers are often used to alter the path of the belt, where a direct path would be impractical. Idler pulleys are also often used to press against the back of a pulley in order to increase the wrap angle (and thus contact area) of a belt against the working pulleys, increasing the force-transfer capacity. Belt drive systems commonly incorporate one movable pulley which is spring- or gravity-loaded to act as a belt tensioner, to accommodate stretching of the belt due to temperature or wear. An idler wheel is usually used for this purpose, on order to avoid having to move the power-transfer shafts.

Figure 3: Idler Pulley

Stepper Motor Nema 17, 2:
NEMA 17 is a hybrid stepping motor with a 1.8° step angle (200 steps/revolution). Each phase draws 1.2 A at 4 V, allowing for a holding torque of 2.8 kg-cm. They are very useful in various applications, especially which demands low speed with high precision. Many machines such as 3D Printers, CNC Router and Mills, Camera Platforms, XYZ Plotters, etc. It is a brushless DC motor, so the life of this motor is dependent upon the life of the bearings. The position control is achieved by a simple Open Loop control mechanism so it doesn't require complex electronic control circuitry. The motor's shaft has been machined for good grip with a pulley, drive gear, etc. and especially avoiding stall or slip.

Figure 4: Stepper Motor

GT2 Timing Belt:

A timing belt is a non-slipping mechanical drive belt and the term may refer to either, Toothed belt, a flexible belt with teeth molded onto its inner surface. Timing belt (camshaft), a toothed belt used to drive the camshaft(s) within an internal combustion engine (a specific application of a toothed belt).

A timing belt is typically made from rubber, although some belts are instead made from polyurethane or neoprene. The structure of the belt is reinforced with corded of fibres (acting as tension members) and the toothed surface is reinforced with a fabric covering. Rubber degrades with higher temperatures, and with contact with motor oil. Thus, the life expectancy of a timing belt is lowered in hot or leaky engines. Also, the life of the reinforcing cords is also affected by water and antifreeze, so it is important that belt that can be exposed to water is able to drain the water away quickly.

Figure 5: timing belt

Voltage regulator:

A voltage regulator is designed to automatically 'regulate' voltage level. It basically steps down the input voltage to the desired level and keeps that in that same level during the supply. This makes sure that even when a load is applied the voltage doesn't drop.

Thus, a voltage regulator is used for two reasons:-

- To regulate or vary the output voltage of the circuit.
- To keep the output voltage constant at the desired value in-spite of variations in the supply voltage or in the load current.
- Voltage regulators find their applications in computers, alternators, power generator plants where the circuit is used to control the output of the plant. Voltage regulators may be classified as electromechanical or electronic. It canalso be classified AC regulators or DC regulators

Linear Motion Bearing:

Linear bearings generally use a pad, bushing, or roller system to carry a load on a rail that need not be a straight line. The rail can be most any length, although that dimension is limited by the actuator. The durability of the bearing is determined by the load and required speed. Furthermore, rails can generally be any profile – simple flat surfaces, round polished rods, or complex profiles with polished ground surfaces on which balls or cylindrical rollers can ride. Hard (Rockwell 60) and ground bearing surfaces work best.

Figure 6: Linear Motion Bearing

Arduino Uno – micro controller:

Microcontrollers like the Arduino UNO, which employs the ATmega328, are popular for use in electronics projects at the basic level. There are 14 digital I/O pins and 6 analogue I/O pins on the board, as well as an ICSP header, power jack, USB connector, reset button, and other components. A USB cable or the board's onboard DC power source can be used to power the charging process. The design of the board utilized by the novice in their work undergoes routine innovation and problem fixes[2, 18]. The Arduino UNO board[5], shown in Figure 12, has a list of hardware components for motor control, Bluetooth, the internet, and other functions, and it can communicate with those devices. The Arduino UNO is a particular sort of Arduino device that is mostly utilized by novices in electronics projects and circuit design. The board is acceptable for usage and chosen over other Arduino devices because of a number of advantages. The user's requirements determine which Arduino products are best, although the Arduino UNO is a standard board when compared to other Arduino products[19].

Arduino programming

```
/*Multi function robot with Arduino.
 * code written by  A Sudhakar  09866794715 4c
design engg.
 */
String readString = "";
#include "AccelStepper.h"
#include <AFMotor.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver fourseedesign =
Adafruit_PWMServoDriver();
AccelStepper stepperX(1, 5, 2);
#include <Keypad_I2C.h>
```

```
#include <Keypad.h>
#include <Wire.h>
#define home_switchX 9
#define servo1 0
#define servo2 1
#define servo3 2
#define servo4 3
long initial_homing=-1;
const byte ROWS = 4;
const byte COLS = 3;
#define I2CADDR 0x20
char hexaKeys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
byte rowPins[ROWS] = {0, 1, 2, 3};
byte colPins[COLS] = {4, 5, 6,};
Keypad_I2C keypad( makeKeymap(hexaKeys),
rowPins, colPins, ROWS, COLS, I2CADDR);
void setup() {
  Wire.begin( );
  keypad.begin( );
  Serial.begin(9600);
  pinMode(home_switchX, INPUT_PULLUP);
  fourseedesign.begin();
  fourseedesign.setPWMFreq(60);
  fourseedesign.setPWM(servo1, 0, 590);
  fourseedesign.setPWM(servo2, 0, 450);
  fourseedesign.setPWM(servo3, 0, 120);
  fourseedesign.setPWM(servo4, 0, 310);
  delay(5);
  stepperX.setMaxSpeed(500.0);      // Set Max Speed
of Stepper (Slower to get better accuracy)
  stepperX.setAcceleration(1000.0);  // Set
Acceleration of Stepper
   while (digitalRead(home_switchX)) {  // Make the
Stepper move CCW until the switch is activated
    stepperX.moveTo(initial_homing);  // Set the
position to move to
    initial_homing--;  // Decrease by 1 for next move if
needed
    stepperX.run();  // Start moving the stepper
        delay(1);
}
stepperX.setCurrentPosition(0);  // Set the current
position as zero for now
stepperX.setMaxSpeed(500.0);      // Set Max Speed
of Stepper (Slower to get better accuracy)
stepperX.setAcceleration(1000.0);
initial_homing=1;
 while (!digitalRead(home_switchX)) { // Make the
Stepper move CW until the switch is deactivated
    stepperX.moveTo(initial_homing);  // Set the
position to move to
    initial_homing++;  // Decrease by 1 for next move
if needed
    stepperX.run();  // Start moving the stepper
    delay(1);
 }
 stepperX.setCurrentPosition(0);  // Set the current
position as zero for now
 stepperX.setMaxSpeed(500.0);      // Set Max Speed
of Stepper (Slower to get better accuracy)
 stepperX.setAcceleration(1000.0);  // Set
Acceleration of Stepper
stepperX.moveTo(30);
while (stepperX.currentPosition() != 30 ){
stepperX.run();
}
stepperX.setCurrentPosition(0);  // Set the current
position as zero for now
 stepperX.setMaxSpeed(500.0);      // Set Max Speed
of Stepper (Slower to get better accuracy)
 stepperX.setAcceleration(1000.0);  // Set
Acceleration of Stepper
}
void loop(){
while (Serial.available()) {
char c = (char)Serial.read();
readString += c;
}
char key = keypad.getKey();
if(key=='1') {
   pick1();
}
if(key=='2') {
   pick2();
}
if(key=='3') {
  proceed1();
}
if(key=='4') {
  proceed2();
}
if(key=='#') {
```

```
  Serial.print("press one or three for drop position one, press two or four for drop position two");
}
if (readString.length() > 0) {
if ((readString == "hello")||(readString == "hi")){
Serial.print("hello, how are you");
}
else if (readString == "what's your name") {
Serial.print("avanthi 2024");
 for (int S1value = 590; S1value >= 30; S1value--) {
    fourseedesign.setPWM(servo1, 0, S1value);
    delay(2);
    }
    for (int S1value = 30; S1value <= 590; S1value++)
{
    fourseedesign.setPWM(servo1, 0, S1value);
    delay(2);
    }
    Serial.print(" avanthi 2024");
    }
else if (readString == "close") {
 for (int S1value =590; S1value >= 30; S1value--) {
    fourseedesign.setPWM(servo1, 0, S1value);
    delay(10);
     }
    Serial.print("gripper close");
      }
      else if (readString == "open") {
 for (int S1value = 30; S1value <= 590; S1value++) {
    fourseedesign.setPWM(servo1, 0, S1value);
    delay(10);
     }
    Serial.print("gripper open");
      }
      else if (readString == "home") {
  Home();
 stepperX.setMaxSpeed(500.0);      // Set Max Speed of Stepper (Slower to get better accuracy)
 stepperX.setAcceleration(1000.0);  // Set Acceleration of Stepper
 stepperX.moveTo(30);
while (stepperX.currentPosition() != 30 ){
stepperX.run();
}
Serial.print("homing done");
}
 else if (readString == "drop one") {
        pick1();
      }
      else if (readString == "drop two") {
        pick2();
      }
       else if (readString == "start one") {
        proceed1();
      }
      else if (readString == "start to") {
        proceed2();
      }
else {

Serial.print("sorry !!, please control from key pad, press # for menu ");
readString = "";
}
delay(100);
}
void Home() {
    fourseedesign.setPWM(servo1, 0, 590);
  fourseedesign.setPWM(servo2, 0, 450);
  fourseedesign.setPWM(servo3, 0, 120);
  fourseedesign.setPWM(servo4, 0, 310);
}
void pick1() {
stepperX.setMaxSpeed(500.0);      // Set Max Speed of Stepper (Slower to get better accuracy)
 stepperX.setAcceleration(1000.0);  // Set Acceleration of Stepper

stepperX.moveTo(800);
while (stepperX.currentPosition() != 800 ){
stepperX.run();
}
  Home();
    for (int S4value = 310; S4value <= 530; S4value++) {
    fourseedesign.setPWM(servo4, 0, S4value);
    delay(8);
    }
     for (int S3value =120; S3value <= 300; S3value++) {
    fourseedesign.setPWM(servo3, 0, S3value);
    delay(6);
     }
     delay(2000);
     for (int S1value = 590; S1value >=530; S1value--)
{
    fourseedesign.setPWM(servo1, 0, S1value);
    delay(8);
```

```
  }
  for (int S3value =300; S3value >= 120; S3value--)
{
  fourseedesign.setPWM(servo3, 0, S3value);
  delay(6);
  }
  for (int S4value = 530; S4value >= 310; S4value--
) {
  fourseedesign.setPWM(servo4, 0, S4value);
  delay(8);
  }
 stepperX.setMaxSpeed(500.0);      // Set Max Speed
of Stepper (Slower to get better accuracy)
 stepperX.setAcceleration(1000.0);  // Set
Acceleration of Stepper

stepperX.moveTo(30);
while (stepperX.currentPosition() != 30 ){
stepperX.run();
}
for (int S3value =120; S3value <= 300; S3value++) {
  fourseedesign.setPWM(servo3, 0, S3value);
  delay(6);
  }
   delay(1000);
 for (int S1value = 530; S1value <=590; S1value++)
{
  fourseedesign.setPWM(servo1, 0, S1value);
  delay(8);
  }
   delay(1000);
  for (int S3value =300; S3value >= 120; S3value--)
{
  fourseedesign.setPWM(servo3, 0, S3value);
  delay(6);
  }}
void pick2() {
stepperX.setMaxSpeed(500.0);      // Set Max Speed
of Stepper (Slower to get better accuracy)
 stepperX.setAcceleration(1000.0);  // Set
Acceleration of Stepper
  stepperX.moveTo(800);
while (stepperX.currentPosition() != 800 ){
stepperX.run();
}
 Home();
  for (int S4value = 310; S4value <= 530;
S4value++) {
  fourseedesign.setPWM(servo4, 0, S4value);
  delay(8);
  }
  for (int S3value =120; S3value <= 300;
S3value++) {
  fourseedesign.setPWM(servo3, 0, S3value);
  delay(6);
  }
  delay(2000);
  for (int S1value = 590; S1value >=530; S1value--)
{
  fourseedesign.setPWM(servo1, 0, S1value);
  delay(8);
  }
  for (int S3value =300; S3value >= 120; S3value--)
{
  fourseedesign.setPWM(servo3, 0, S3value);
  delay(6);
  }
  for (int S4value = 530; S4value >= 310; S4value--
) {
  fourseedesign.setPWM(servo4, 0, S4value);
  delay(8);
  }
stepperX.setMaxSpeed(500.0);      // Set Max Speed
of Stepper (Slower to get better accuracy)
 stepperX.setAcceleration(1000.0);  // Set
Acceleration of Stepper
  stepperX.moveTo(370);
while (stepperX.currentPosition() != 370 ){
stepperX.run();
}
for (int S3value =120; S3value <= 300; S3value++) {
  fourseedesign.setPWM(servo3, 0, S3value);
  delay(6);
  }
   delay(1000);
 for (int S1value = 530; S1value <=590; S1value++)
{
  fourseedesign.setPWM(servo1, 0, S1value);
  delay(8);
  }
   delay(1000);
  for (int S3value =300; S3value >= 120; S3value--)
{
  fourseedesign.setPWM(servo3, 0, S3value);
  delay(6);
  }}
void proceed1() {
```

```
 int var = 0;
while  ( var < 5) {
 pick1();
 var++;
 }
 }
void proceed2() {
 int var = 0;
while  ( var < 10) {
 pick2();
 var++;
 }
 }
```

Working principle

The parallel jaw gripper has at least two fingers which can be moved towards each other along one axis. Usually, the fingers can be moved independently from each other in order not to shift the object, but they are only able to perform simple operations like open and close. Thereby, a longitudinal or side movement is impossible. A parallel jaw performs, a manual control to steer the gripper must be possible for enabling the highest flexibility. The principles which are followed in this type of parallel jaw gripper are: The force closure grip: (The characteristic of the force closure grip is that the gripper keeps the object in a stable state by compensating all forces and torques created by the object. The sum of all forces and sum of all torques must equal zero ($\Sigma F=0$; $\Sigma M=0$). The force closure grip can be differentiated into a grip with friction and without friction. The force closure grip without friction is much idealized and not very common in daily use; therefore, it is no further mentioned. The force closure grip with friction requires at least contact points for gripping a planar object and at least 4 contact points for a three-dimensional object.

CONCLUSION

The design and development of pick and place robot has been carried out. A prototype was confirmed functional working of robot system. This system would make it easier for human beings to pick and place the risk of handling suspicious objects, which could be hazardous in its present environment and workplace. Complex and complicated duties can be achieved faster and more accurately with this design.

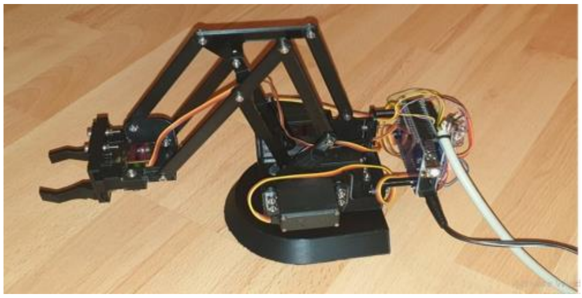
Figure 7: 3D printed assembly

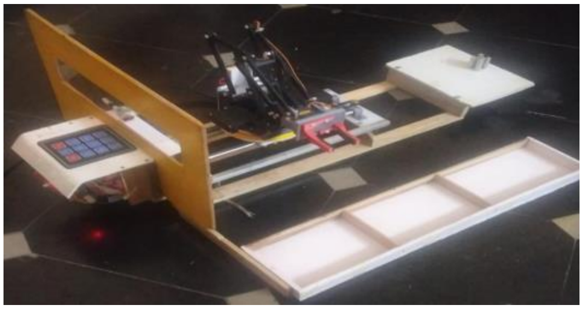
Figure 8 : Complete Assembly

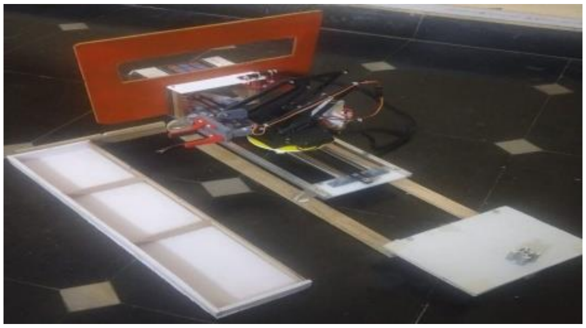
Figure 9: 3D printed assembly

The results are highlighted as below:

- The design is simple, safe and speed operation, efficient use of space.
- Further, the machine can be used as a study equipment for understanding pick and place operation.
- Future work could be taken up by developing fully automatic the system.

Future Scope:

- Improved Accuracy and Precision: Enhancing the arm's ability to interpret voice commands more accurately and execute tasks with greater precision can open doors to various industries, including manufacturing, healthcare, and logistics.
- Integration with AI and Machine Learning: Implementing AI and machine learning

algorithms can enable the robotic arm to learn from past interactions, adapt to different environments, and optimize its performance over time.

- Multi-functionality: Expanding the capabilities of the robotic arm to perform a wider range of tasks, such as assembly, sorting, or even complex surgeries in the medical field, can increase its utility and marketability.
- Remote Operation and Telepresence: Introducing features for remote operation and telepresence can allow users to control the robotic arm from a distance, enabling applications in hazardous environments, space exploration, or remote assistance scenarios.
- Accessibility: Designing the system to be accessible to individuals with disabilities or limited mobility can have significant societal impact, empowering them to perform tasks independently and improve their quality of life.

REFERENCE

1] Abdullahi, Zakariyya Hassan, et al. 2015. Mobile robot voice recognition in control movements. International Journal of Computer Science and Electronics Engineering (IJCSEE) 3.1: 11 16. B

2] ARDUINO.CC, “Arduino – Introduction”, 2015 [Online] Available: http://arduino.cc/en/Guide/ Introduction. [Accessed: 25- Feb - 2015].

3]Pratik Chopra and Harshad Dange, VOICE CONTROLLED ROBOT, Department of Electronics Engineering, K. J. Somaiya College of Engineering, 2006-2007.

4]United Nations Conference on Trade and Development. Impact of Rapid Technological Change on Sustainable Development; OCLC: 1145601349; United Nations: San Francisco, CA, USA, 2020. [Google Scholar]

5] "Pick and place robotic arm using Arduino", Int. J. Sci. Eng. Technol. Res. (IJSETR), vol. 6.12, pp. 1568-1573, 2017.

6] P.H. Pande, P.V. Saklecha, P.D. Pawar and A. N. Shire, "Arduino Based Robot for Pick and Place Application", International Journal of Electronics Communication and Soft Computing Science Engineering (IJECSCSE), pp. 23-27, 2018.

7] Michael E Moran, "Evolution of robotic arm" in International Journal of Advanced Research in Computer and Communication Engineering, March 2013.

8] M. S. Bin Bahrudin, R. A. Kassim, and N. Buniyamin, “Development of Fire alarm system using Arduino Uno and Raspberry pi,” 2013 Int.Conf. Electr. Electron. Syst. Eng. ICEESE 2013, pp. 43–48, 2013.

9] Nikesh. R. Vange, Atul.V. Nar, Dhananjay.B. Surve, Anita. P. Trimukhe, Manisha.M. Patil, Rajesh.A. Patil, Object Sorting Robotic Arm Based on Color Sensing Mechanism, International Journal of Emerging Technology and Advanced Engineering, (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 5, Issue 5, May 2015)

10] Automation 2021, 2(4), 238-251; https://doi.org/10.3390/automation2040015

11] I. Standard, 8373: 1994.

12] “A robotic gripper for picking up two objects simultaneously”Taylan Atakuru, Evren Samur

13] “A statistical review of industrial robotic grippers” Lionel Birglen, Thomas Schlicht

14] “Axiomatic Design of a linear motion robotic claw with interchangeable Grippers”Bjartmar Freyr Erlingsson, Ingólfur Hreimsson, Páll Indriði Pálsson, Sigurður Jóhann Hjálmarsson, Joseph Timothy Foley

15] “Design and implementation of 3-DOF gripper for maintenances tasks in EAST vacuum vessel” Hongtao Pan, Xiang Gao, Jianjun Huang, Huibin Sun, Xiaodong Lin, Jiangang Li,Yinxian Jie, Qing Zang, Yuntao Songc, Eric Villedieu

16] “design and optimisation of robotic gripper: a review” Vaibhav Raghav, Jitender Kumar and Shailesh S.Senger

17] G. Katal, S. Gupta, and S. Kakkar, "DESIGN AND OPERATION OF SYNCHRONIZED ROBOTIC ARM."

18] J. L. Jones, "Robots at the tipping point: the road to iRobot Roomba," Robotics & Automation Magazine, IEEE, vol. 13, pp. 76-78, 2006.

19] A. C. McDonald, "Robot technology: theory, design and applications," Sir Sanford Fleming Coll., Ontario1986.

20] S.Yu,X.Yu, B. Shirinzadeh, and Z. Man, "Continuous finite-time control for robotic manipulators with terminal sliding mode," Automatica, vol. 41, pp. 1957-1964, 2005.

21] Dr V Hari Kiran, Modeling, analysis and fabrication of hotel servicing robot “Futuristic Trends in Mechanical Engineering”e-ISBN: 978-93-5747-

908-0, IIP Series, Volume 3, Book 4 , Part 4 ,Chapter 1, pp 147-160
22] M. Hägele, K. Nilsson, and J. N. Pires, "Industrial robotics," in Springer handbook of robotics, ed: Springer, 2008, pp. 963-986