The International Journal of Engineering and Science (IJES)
|| Volume || 10 || Issue || 2 || Series I || Pages || PP 15-23 || 2021 ||
ISSN (e): 2319-1813 ISSN (p): 20-24-1805

Industrial Robot and Automation

Pratikkumar R. Hingu[1], Dhruv N. Panchal[2]
1 Rheinisch-Westfälische Technische Hochschule (RWTH), Aachen, North Rhine-Westphalia, Germany.
2 Mechanical Engineering, Gandhinagar Institute of Technology, Gandhinagar, India.

--***ABSTRACT***---
This research paper focuses mainly on robotics and industrial level automation respect to the mathematical modelling and programming with simulation. Where the accuracy requirement is nearly 100%, the human resources cannot handle the workload and unable to achieve the required accuracy in less time; hence the industrial robots are used. The research paper mainly focused on the industrial robot and automation design and calculation. The advanced mathematical formula, mechanical & electrical component, and the programming language come together to prepare the required industrial robot. The 6-axis robot is a well-known standard robot, which is commonly used by all kinds of MNCs for heavy and accurate work. When selecting a robot properly, it is necessary to consider the different properties of the robot, including how the robot links are connected and controlled at each joint. Next, a thorough evaluation of robotic kinematics, dynamics, and control strategies, together with all the diagnosis of deep neural networks, will describe recent efforts to accelerate the advancement of intelligent control systems for robotic systems.
KEYWORDS: *Industrial Automation, Industrial robotics, robot, 6-axis, robots, smart manufacturing, system engineering, mechatronic engineering, robot arm.*

Date of Submission: 22-01-2021 Date of Acceptance: 06-02-2021

I. INTRODUCTION

Industrial robotics had significantly enhanced and extended to a variety of activities such as storing, painting, welding, sealing, and continue. It is necessary to monitor precisely the trajectory of the end-effector in such tasks. Industrial robots achieve their maximum financial benefit in implementations at which the volume of the output is high enough to recover spending on hardware, programming, and technical expenses but is too limited to preclude the requirement of specialized purpose-built infrastructure. However, high-volume function includes a regular model or optional modifications, such as those used in the automobile system, which is ideal for robotization. The influence of robotics on the price per unit, in terms of volume. Here, we are mainly discussed about the 6-axis standard industrial robots. First, the simulation model was developed and then, with correct programming, circuit construction, and mathematical calculation physical model was developed. Here, the mathematical formulas, simulation, circuit design, and UI design were briefly discussed.

1.1. 6-Axis anthropomorphic robot

The developed industrial robot is a serial kinematic structure with 6 links connected together by 6 joints, which is starting from the base and adding up all the joints. We have two more points named: MP (Mount Point) and TCP (Tool Centre Point.) TCP is the last edge the robots.

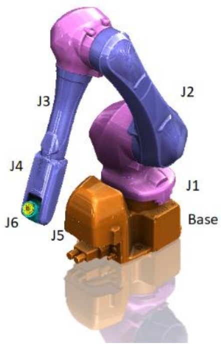

Figure 1: Base and joints

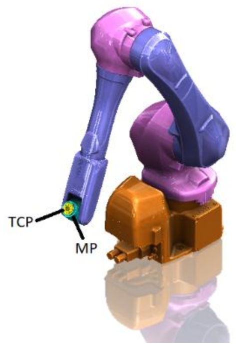

figure 2: MP and TCP

1.2. Movements

The movement shows how the robot reaches the operational spots from one point to another one. The moment is mainly concern with the path planning, which also includes the speed and the workspace of the 6-axis robot.

II. FRAMES

A frame is fundamentally a coordinate system with a particular spatial location. The frames are the geometric base of the robotic arm, and learning how and where to treat them is a crucial step. We will quickly define a few foundational frames when interacting with robots.

2.1. Geometrical framework

We mainly focused on Machine Co-ordinate System, Tool Co-ordinate System, and Workpiece Co-ordinate System as the research paper principally concentrated on single 06-axis robot, we don't need to worry about the Global Coordinate System.

2.2. Frame-operations

The frame operations deal with the two or more-prospect view to the single point, to which the operation performed, and the mathematical and logical program came into the pictures. The frame-operation primarily concerned with translation, and rotation, which are as follows,

2.2.1 Translation: Translation refers to the changing in the location of the object.

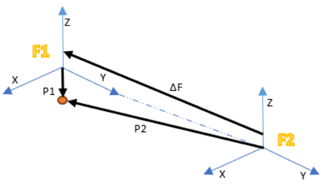

figure 3: Frame translation

$$P_2 = \Delta F + P_1$$

$$\begin{Bmatrix} x_2 = x_1 + \delta_x \\ y_2 = y_1 + \delta_y \\ z_2 = z_1 + \delta_z \end{Bmatrix}$$

$$\text{Ex.: } P_1 = [1,1,1] \quad \Delta F = [0,-8,0] \;\rightarrow\; P_2 = [1,-7,1]$$

2.2.2 Rotation: Rotation represents a change in the orientation.

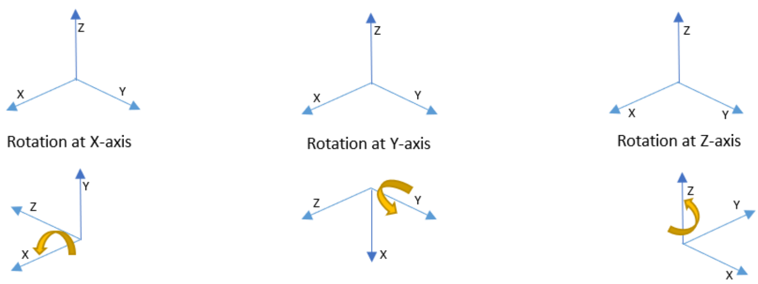

figure 4: Frame rotation

$P_2 = R(\theta)P_1$, where R(θ) is Rotational matrix

$$R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \quad R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In all cases a rotation of zero degrees is equivalent to the identity matrix.

$\rightarrow R(\theta = 0) \equiv I$

Transporting a rotation matrix, we find same matrix with opposite angle of rotation.

$\rightarrow R(-\theta) \equiv R^T$

2.2.3 Euler angle: The rotation of robots usually given in A, B, C angles around the base X, Y, Z coordinate system,

$$R = R_z(C)R_y(B)R_x(A) = \begin{bmatrix} c_z c_y & c_z s_y s_x - s_z c_x & s_z s_x + c_z s_y c_x \\ s_z c_y & c_z c_x + s_z s_y s_x & s_z s_y c_x - c_z s_x \\ -s_y & c_y s_x & c_y c_x \end{bmatrix}$$

Where,

$B = -s_y = a\tan 2(\pm\sqrt{1-(R_{31})^2}, -R_{31})$

$A = c_y s_x$ and $c_y c_x = a\tan 2(\pm R_{33}, \pm R_{32})$

$C = c_z c_y$ and $s_z c_y = a\tan 2(\pm R_{11}, \pm R_{21})$

If $c_y = 0$, *then*

$B = \pm\frac{\pi}{2}\ (s_y = \pm 1)$

$A - C = a\tan 2(R_{22}, R_{12})\ (s_y = 1)$

$A + C = a\tan 2(R_{22}, -R_{12})\ (s_y = -1)$

2.2.4 Translation and rotations combined: Here the T is Homogeneous Matrix, for the understand let the rotation matrix R calculated with translation offset Δ with the respect to the Z-axis.

$$T = \begin{bmatrix} R & \Delta \\ 0 & 1 \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \leftrightarrow \quad \Delta = \begin{Bmatrix} \delta_x \\ \delta_y \\ \delta_z \end{Bmatrix}$$

$$T = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & \delta_x \\ \sin\theta & \cos\theta & 0 & \delta_y \\ 0 & 0 & 1 & \delta_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The first rotation and then the translation would use for programming and estimation. The order here is the most important one. Here under is the example to find the TCP position of fixed base frame using homogeneous translation.

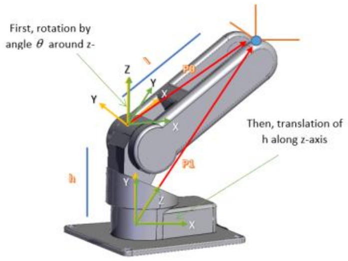

figure 5: TCP position of fixed base

To find the position of P1, we pre-multiply P0 by the matrix T.

$$P_1 = TP_0 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} l\cos\theta \\ l\sin\theta \\ h \\ 1 \end{bmatrix}$$

2.2.5 Inverted Transformations: To find the inverted transformation we have to find initial given the final.

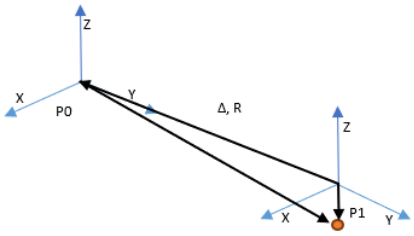

figure 6: Inverted Transformations

$$P_0 = TP_1 \quad \rightarrow \mathrm{R} = \mathrm{R}^{-1} = R^T \quad \rightarrow \Delta = -R^T\Delta$$

$$T^{-1} = \begin{bmatrix} R^T & -R^T\Delta \\ 0 & 1 \end{bmatrix}$$

III. SIMULATION AND DESIGN

With the proper understanding of the mathematical formula and concept, the robot simulation was carried out based on Unity 3D. The Unity 3D engine helps to create virtual Industrial based robot with professional interaction. Before the creation of the physical robotic model, the simulation based on the proper mathematical functional was developed. Unity is capable to control the physical industrial robot through PLC or Raspberry pi with TCP/IP function. Here we have used Arduino, hence the physical model and circuit were done separately. The simulation model and the UI were discussed below.

3.1 System Construction

To prepare the virtual system it is necessary to build the model parts and assembly, first. To initiated the process, the required 3D model and the required environment were created with the FBX file, and then the file was imported into the Unity 3D. With the help of programming scripting, necessary functions were added and safety features were implemented. The real-world scenario was created so it will help to build the actual robot, know its functions and help to improve its functionality. The system framework is as follows:

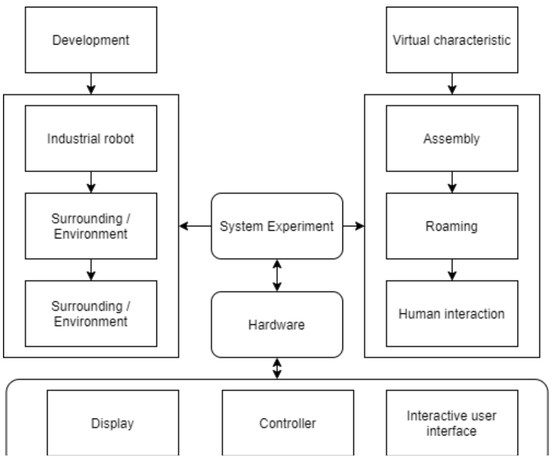

figure 7: Construction flowchart

3.2 Model Implementation

The 6-axis robot was designed in Solidworks, having 6 degrees of freedom, of which 3 degrees of freedom describe the end position of the robot and another 3 degrees of freedom describe the end pose of the robot. The 3D model is then imported to modify each joint, so each point can be controlled separately. The ontology and robot link co-ordinate system is shown here:

figure 8: 6-axis robot design

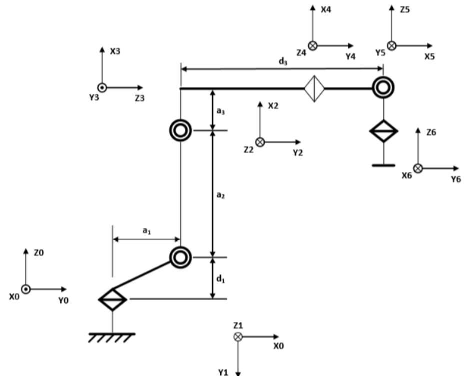

figure 9: robot link co-ordinate system

3.4 Kinematic Analysis:

Here the robot programming, mathematical calculation, and modeling are done with forwarding kinematics as well as inverse kinematics. The forward kinematics is the process in which we are calculating the position and orientation of the TCP given the current values of the joint axes. On the other end, the inverse transformation is the function to find the values of the joint axes given the position and orientation of TCP. These coordinates of TCP are expressed relative to the base frame. Here the main fundamental difference is shown:

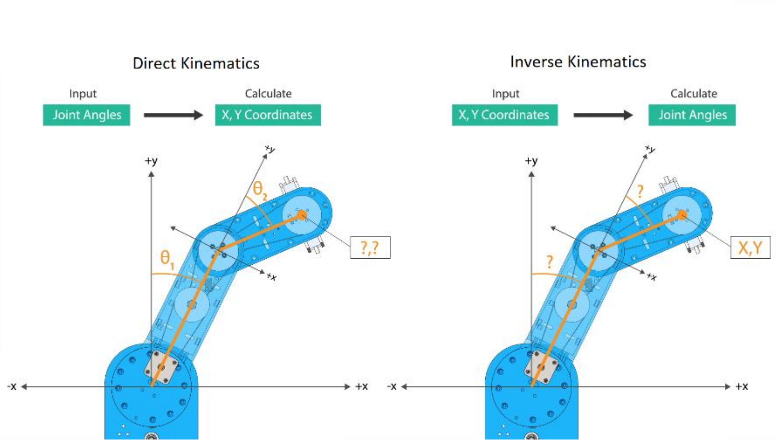

figure 10: kinematic system

As said, forward kinematics work on the principle of joint angles to coordinates. To find the position and rotation of TCP with the respect to the base frame,

$$Position = TCP(X,Y,Z) = T_1T_2T_3T_4T_5T_6O$$
$$= T_{16}O$$
$$Rotation = TCP(A,B,C) = R_1R_2R_3R_4R_5R_6$$
$$where, R_1R_2R_3 = R_{arm} \text{ and } R_4R_5R_6 = R_{wrist}$$

The key point to understand the coding of kinematics formulation of the 6-axis robot is as follows,

```
public Matrix4x4 WKinematics (float[] angles){
        …….
        Matrix4x4 W = T01 * T02 * T03 * T04 * T05 * T06;
        return W;
}.
```

Inverse kinematics solution can be achieve using two methods namely numerical method and analytical method. The numerical method is not stable and slow in process, where the analytical method is preferred respect to the structure of the robot. For better understanding of the inverse kinematics background, the following code is helpful:

```
public float[] InvKinematics(Matrix4x4 G)
{
    // Wrist posture
     Matrix4x4 W = G * this.ETOT.inverse *this.WTOE.inverse;
    // Joint Angle result
     float[] dstAngles = WInvKinematics(W);
    return dstAngles;
}.
```

3.5 System UI of unity:

After the successful formulation and programming of the 6-axis robot, the simulation and real-time workflow can be achieved with Unity 3D. The system UI is designed in a simple way that each joint and link are controlled separately, effectively. The auto mode is also introduced, once the robot is taught their regular or general movement, that the operator can take a break for a while and let the robot can be done the batch work itself. The main system UI consists of manual and auto mode, where each joint, tool and base control can be controlled separately. If the required speed more or lesser than it can be regulated, too. To follow the general path and specific moment, the UI also has an option to teach the robot, and let the robot perform the taught motion, and path in auto mode.

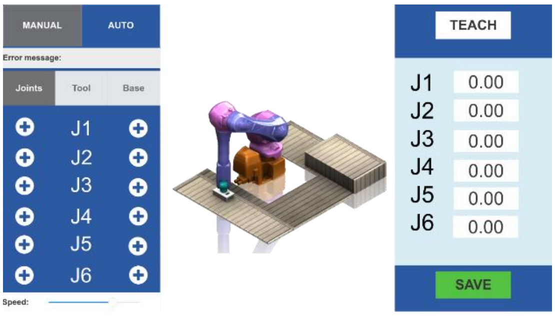

figure 11: Unity system UI

IV. PHYSICAL MODELLING

The physical modeling process starts once the simulated and programmed robot works perfectly as per the requirement, also if any error/bugs, which found during the simulation are solved correctly. Robotic arm parts, gripper, and base (physical parts) manufactured using 3D printing and hard sheet metal, on the electronics side the Arduino is the main brain of the system, and the UI & programming done in MIT App Inventor to control the robot using android smartphone. The HC-05 Bluetooth module helps to make the connection between the robot and android smartphone.

4.1 Circuit Design

The brain of the physical 6-axis robot is Arduino Uno, and the connection stabilized with the help of the HC-05 Bluetooth module. The programming language used for performance is C++. The Arduino Uno provides the memory to the 6-axis robot, to save the steps, repeat the saved steps and reset the function. The programming took place after observing the unity simulation behaviour, which helps to create a more stabilized version of the physical model. Here is the circuit diagram to control each part of the 6-axis robot.

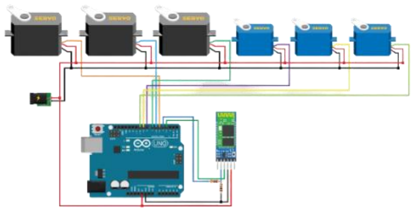

figure 12: Circuit Design

4.2 Android UI Interface

The developed unity model simulated model is compatible with PLC and RaspberryPi, for which we can use a unity 3D UI interface to control the physical model. As we have used Arduino and to control it with a smaller device like an android smartphone, we have to develop a specific app to control each part individually. The UI and android programming are done in MIT App Inventor to give it logical control to collaborate with the Arduino Uno via Bluetooth module to perform a specific operation. The following picture is the user interface to handle the 6-axis robot.

figure 12: Android UI

4.3 Motor calculation and selection

The selection criteria of the motor are the main concern for the physical modelling. The servo motors handle all the required functions. The calculation for motor selection is shown hereunder (to provide calculation method, only waist motor selection process shown here),

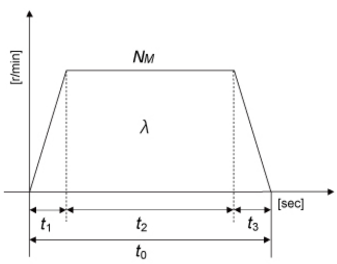

figure 12: Operation pattern

▹ Mechanism Specification:

$A = 220mm$

$B = 50mm$

$C = na$

$e = 0mm$

Arm Mass = $m = 6.5kg$

Other Moment of Inertia: $J_e = 275.708 \times 10^{-4} [kg \cdot m^2]$

Transfer Mechanism Primary Side Diameter :

(*We haven't used any transfer mechanism)

Secondary Side Diameter = $D_{g1} = na$

Primary Side Mass = $D_{g2} = na$

Secondary Side Mass = $m_{g1} = na$

Belt Mass = $m_{gb} = na$

Gear Ratio = $i_g = na$

Efficiency = $\eta_g = na$

Moment of Inertia = $J_g = na$

▹ Operation Pattern:

Travel Amount = λ = 1.8 [mm (deg)]

Rotation Speed = NM = 0.6 [r/min]

Acceleration Time = t_1 = 0.001 [sec]

Constant Speed Operation Time = t_2 = 0.498 [sec]

Deceleration Time = t_3 = 0.001 [sec]

Positioning Time = t_0 = 0.5 [sec]

• Calculation of Inertia

→ Moment of load inertia of the arm / frame

$$J_a = \frac{1}{12} \rho ABC \left(A^2 + B^2 + 12e^2 \right) = \frac{1}{12} m \left(A^2 + B^2 + 12e^2 \right) = 275.708 \times 10^{-4} [kg \cdot m^2]$$

→ Moment of inertia of the transfer mechanism

$$J_g = \frac{1}{8} m_{g1} D_{g1}^{\ 2} + \frac{1}{8} m_{g2} \frac{D_{g2}^{\ 2}}{i_{g2}^{\ 2}} + \frac{1}{4} m_{gb} D_{g1}^{\ 2} = 0 [kg \cdot m^2]$$

→ Total Load Inertia

$$J_L = \frac{J_T + J_m n}{i_g^{\ 2}} + J_e + J_g = 551.416 \times 10^{-4} [kg \cdot m^2]$$

• Calculation of Required Torque

→ Load Torque(External Torque)

$T_e = 0 [N \cdot m]$

→ Number of Operating Pulses

$$A = \frac{\lambda}{\theta s} \times i_g = 90 [\text{ Pulse }]$$

*If it is not an integer, an error will occur in the actual traveling amount

→ Operating Pulse Speed

$$f = \frac{A}{t_0 - t\mathrm{i}} \text{ or } \frac{A}{t_0} = 180 [Hz]$$

→ Acceleration Torque T_a Rotor inertia J_o $n = (3.6^\circ / \theta s) \cdot i$

$$T_a = \left(J_0 \cdot \mathrm{i}^2 + J_L \right) \frac{\pi \cdot \theta s \cdot f}{180^0 \cdot t_1} \text{ or } T_a = \left(J_0 \cdot \mathrm{i}^2 + J_L \right) \frac{\pi \cdot \theta s \cdot f^2}{180^0 \cdot n} = 4.726 [N \cdot m]$$

→ Required Torque

$T = T_a + T_e = 4.726 [N \cdot m]$

•Calculation of the selected motor judgment item

(Safety Factor $S_f = 1$)

$\rightarrow$ Safety factor S Motor torque of the rotation speed $Tr = 12[N \cdot m]$

$$S = \frac{Tr}{T} = 2.54$$

$\rightarrow$ Inertia ratio β

$$\beta = \frac{J_L}{J_0 \cdot i^2} = 4.6$$

$\rightarrow$ Acceleration/Deceleration Rate T_R

$$T_R = \frac{t_i}{f} = 5.54$$

V. CONCLUSION

The MNC required mainly 6-axis robots to perform and handle most of the process, which helps the manufacturing process errorless and fast. It is more convenient to first simulate the design and program the required robot as per industrial standard in unity machine, and after observing the behaviour and resolving the error, approach to building physical system which helps to save power, time, and machine. The unity is growing more popular nowadays, and also it tried to meet the industrial standard of the robot simulation. The new generation, one can say school and university students who cannot afford to build the physical system, but have the potential to learn and build industrial-based robotics system can have the platform to show their creativity, the potential to the world and also can enhance their knowledge through the simulation platform.

REFERENCE

[1]. Lin, M., San, L. and Ding, Y., 2020. Construction of Robotic Virtual Laboratory System Based on Unity3D. IOP Conference Series: Materials Science and Engineering, 768, p.072084.

[2]. Industrial Robot: An International Journal, 2008. Designing Autonomous Mobile Robots: Inside the Mind of an Intelligent Machine. 35(2).

[3]. J. F. Canny and K. Y. Goldberg, ""RISC" industrial robotics: recent results and open problems," Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 1994, pp. 1951-1958 vol.3, doi: 10.1109/ROBOT.1994.351176.

[4]. W. A. Mattingly, D. Chang, R. Paris, N. Smith, J. Blevins and M. Ouyang, "Robot design using Unity for computer games and robotic simulations," 2012 17th International Conference on Computer Games (CGAMES), Louisville, KY, 2012, pp. 56-59, doi: 10.1109/CGames.2012.6314552.

[5]. Green, D., Stanton, N., Walker, G. and Salmon, P., 2005. Using wireless technology to develop a virtual reality command and control centre. Virtual Reality, 8(3), pp.147-155.

[6]. E. Sita, C. M. Horváth, T. Thomessen, P. Korondi and A. G. Pipe, "ROS-Unity3D based system for monitoring of an industrial robotic process," 2017 IEEE/SICE International Symposium on System Integration (SII), Taipei, 2017, pp. 1047-1052, doi: 10.1109/SII.2017.8279361.

[7]. A. Hussein, F. García and C. Olaverri-Monreal, "ROS and Unity Based Framework for Intelligent Vehicles Control and Simulation," 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, 2018, pp. 1-6, doi: 10.1109/ICVES.2018.8519522.

[8]. J. You Wei. Design of Virtual Teaching System for Robot Body Disassembly Based on Unity3D. Mechanical Research & Application, 2018, 31 (04):230-234.

[9]. Urrea, C. and Pascal, J., 2018. Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot. Computers & Electrical Engineering, 67, pp.791-806.

Pratikkumar R. Hingu, et. al. "Industrial Robot and Automation." *The International Journal of Engineering and Science (IJES)*, 10(02), (2021): pp. 15-23.