[International Conference on Contemporary Advances in Mechanical Engineering]

Development of digital twin of an articulated robotic arm

Adarsh Aggarwal[a], S S Dhami[a]1

[a] *Department of Mechanical Engineering, National National Institute of Technical Teachers Training and Research, Chandigarh 160019,INDIA*

Abstract

A digital twin represents a physical product, procedure, or service in the digital world. This technology can be used to replicate and mimic processes to gather data and carry out specific tasks. In this work, a digital twin is built to connect the virtual and actual robot so that the articulated robotic arm can be controlled virtually. The design and model for the robotic arm that was used in the Gazebo simulation environment were built using the Robotic Operating System (ROS). The TCP protocol connects a virtual robotic arm with a real robotic arm for bi-directional communication. Raspberry pi is used with sensors to collect the acceleration and temperature data and publish it on a cloud database for further analysis for the real-time condition monitoring of the robot.

Keywords: Digital Twin, ROS, Gazebo, IoT, Industry 4.0;

1. Introduction

A digital twin comprises a physical and digital product and the link between the physical and digital products. It enables real-time communication between physical and virtual models. Data from the real product can be read and used for analysis, simulations, and other purposes by using the digital twin. Data, actuation orders, control signals, and other information can be given to the physical twin using a digital twin [1]. The purpose of the digital twin is to create a single interface via which these multiple data repositories may be accessed.

Robots can be used for laborious tasks and other tasks that are considered hazardous for people, such as in the mining and construction industries. For example, to help with physically demanding occupations, construction robots can be used alongside people. A combination of humans and machines, however, raises security issues.

Therefore, a real-time human and robot tracking system can ensure safety by providing humans with information about robots. A digital twin can help in quicker production and risk assessment, preventing future problems, Real-time remote observation, and more effective financial judgment [2].

1.1. Digital Twin Implementation Technique

A Digital Twin can be created with different methods. One method is to create a 3D CAD of the robotic arm and use Virtual Reality (VR) equipment to control the arm remotely; however, the software is needed to build a VR scene that resembles the prototype setting and to calibrate the location of the robotic arm in the virtual world. In another method, a digital twin is created to connect the virtual and actual robots so that the articulated robotic arm can be controlled virtually. The design and model for the robotic arm used in the Gazebo simulation environment are built using the Robotic Operating System (ROS) [5]. The Message Queuing Telemetry Transport (MQTT) platform is used to connect the virtual and actual robotic arms for two-way communication [6] [7].

* Corresponding author.
E-mail address: aggarwal.adarsh98@gmail.com(Adarsh Aggarwal) , ssdhami@nitttrchd.ac.in (S S Dhami)

Figure 1 gives a simplified idea of different ways to develop a digital twin.

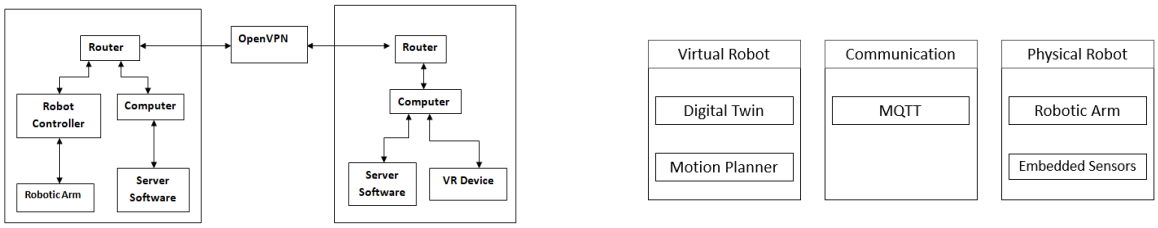


Figure 1: Flowcharts of different techniques

2. Objective

This work aims to establish bi-directional communication between the virtual and physical robot for controlling the physical robot remotely and to gather real-time data about the physical condition of the robot using sensors and raspberry-pi and push it on the cloud server for visualization purposes.

3. Materials and methods

The educational 6-axis articulated robotic arm is used in this work along with sensors and a microprocessor to develop the digital twin of the articulated robotic arm and for data acquisition about the physical condition of the robot.

3.1. Robot operating system (ROS)

A ROS package, which is open-source. It has a huge number of tools and software packages that enable the quick development of the majority of robotic applications. Every part required to build a robot from the ground up has been included. It includes a set of drivers for widely used gadgets, enabling hardware attachment without the requirement for driver coding. Moreover, it contains techniques and tools for robotic system development, visualization, and simulation. Writing ROS packages may be done in several different languages, such as Python, C++, Java, MATLAB, and others[3] [4].

3.2. Wlkata Mirobot

It is six axes robotic arm, a required standard specification for most industrial robotic applications. It is an educational robot that is a lightweight and pre-assembled device whose outer shell is made of ABS engineering plastic and weighs only 1.5kg. Figure 2 shows the work envelope of the Wlkata Mirobot robotic arm. Its significant specifications and maximum and minimum joint angles are listed in Table 1.

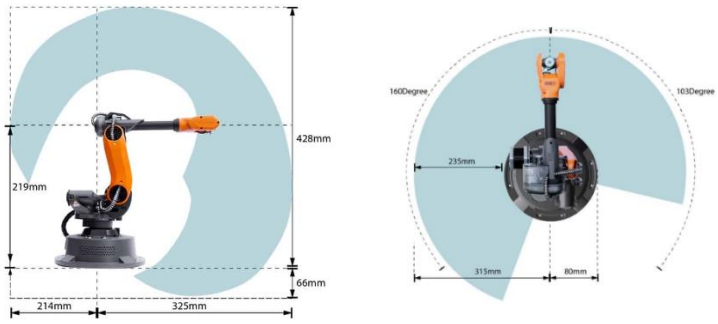


Figure 2:Wlkata Mirobot work envelope

Table 1: Technical specification of wlkata Mirobot

S.No.	Properties	Description
1.	Axle number	6+1
2.	Maximum payload	400g
3.	Type of the six joints	High accuracy stepping motor + reducer
4.	Repeated positioning accuracy	0.2mm
5.	Working environment	-10°C~ 60°C
6.	Joint1	-100°to +100°
7.	Joint2	-60°to +90°
8.	Joint3	-180°to +50°
9.	Joint4	-180°to +180°
10.	Joint5	-180°to +40°
11.	Joint6	-180°to +180°

3.3. Sensors and microprocessor

The temperature sensor DHT-11 and accelerometer ADXL-345 are used to measure the physical robotic's temperature and acceleration, respectively, Figure 3.

(a) (b)

Figure 3: (a) Temperature sensor (b) accelerometer

Primary specifications of the temperature sensor and accelerometer are given in Table 2.

Table 2: Specification of temperature sensor

Sensor	Properties	Description
DHT-11	Power	3.3-5v
	Humidity range	20-80%
	Temperature Range	0-50°C
	Sampling Rate	1 Hz
	Working environment	-10°C~ 60°C
ADXL-345	Interface	I2C
	Measurement Range	± 16g
	Resolution	range up to 13 bits at +/- 16 g
	Operating Voltage (VDC)	5
	Operating Current	23uA

4. Digital Twin development

The digital twin was developed using RViz, Move_It, and Gazebo software that run on a ROS environment. RViz is visualization software that shows the visualization of a robot after taking the joint positions from Move_it software. The simulation of the virtual robot is superimposed in a gazebo program, a physics simulator that builds a world and simulates the robot, and tells the robot's actual behavior that is expected. The flow chart in Figure 4 can be referred to for a better understanding.

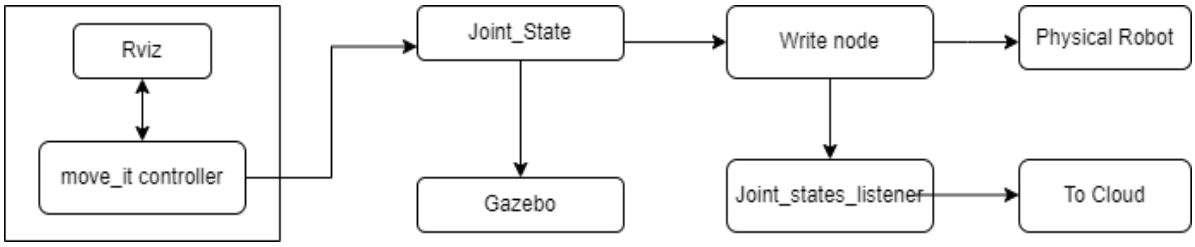


Figure 4: Flow chart of digital twin

The Wlkata ROS package [8] comprises the *RViz node, gazebo node, and move_group node*. Two additional nodes, *write_node* and *Joint_States_listener*, are created to establish bi-directional communication. RViz publishes messages on the subject *joint_state*, which is subscribed by the gazebo node. The joint position given in *RViz* is subscribed by the gazebo node, which runs the simulation and shows the position of the virtual robot. The *write_node* reads the joint position message using a callback function and sends a command to the physical robot to move to a particular position *Joint_states_listener node* is a python script that subscribes to the *write_node* and publishes the data on the cloud using MQTT protocol. Figure 4 shows the flow chart of the process.

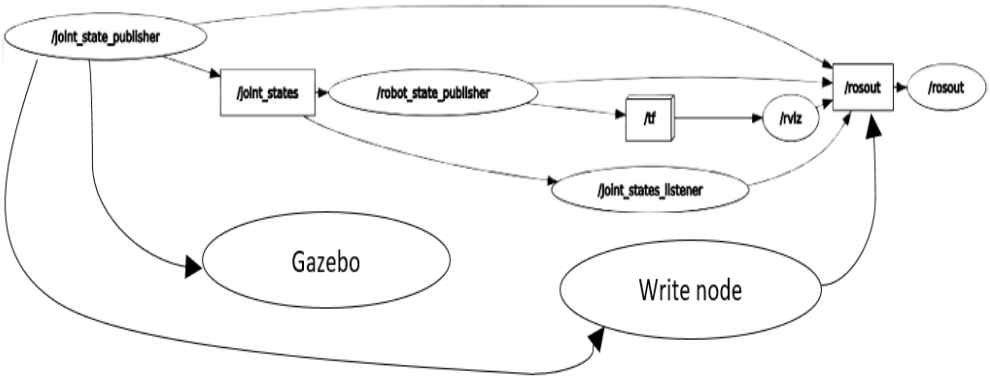


Figure 5: ROS node architecture of the process

5. Data acquisition

The physical condition of the robot is monitored using the installed temperature and accelerometer sensors. Sensors are connected to the raspberry pi [9]. The accelerometer is attached to link 1 of the robotic arm, whereas the temperature sensor is placed at the base. Data collected from the sensors is sent to a free tier version of the Aws cloud database for future analysis. The complete flowchart of the process can be seen in Figure 5.

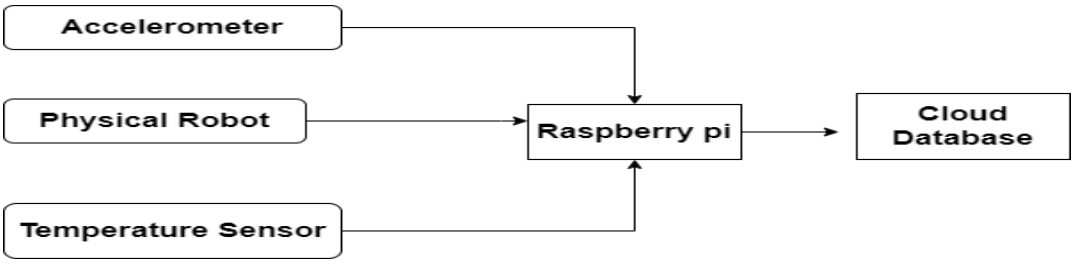


Figure 6: Flowchart of data acquisition

The actual setup can be seen in Figure 7.

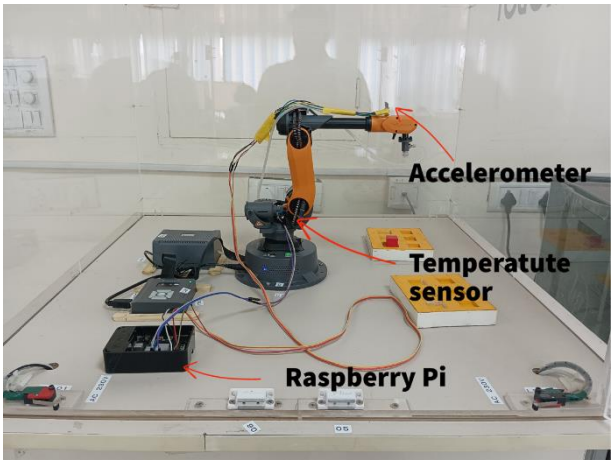


Figure 7: Actual setup of the robot

6. Experimentation

This experiment aims to validate and analyze the connectivity between physical and virtual robots. The digital twin is tested, and data is accumulated using sensors. The experiment was performed in two stages. In the first stage, the values of the joint angles were given as inputs using the Joint State controller node in the ROS to set the position of the virtual robotic manipulator. The same input was transmitted to the physical robot write_node, resulting in the physical movement of the actual robotic arm. The actual position attained by the robot was fetched in terms of its joint angles values from the robot controller using a script developed in WLKATA Python SDK.

In the second stage, the robot was programmed to perform a pick and place task at different speeds to determine how speed impacts the temperature and acceleration of the robot placed at the base motor and upper link, respectively. Proper cooling time was given to the robot between the experiments to acquire temperature readings. The sensors' values are pushed to the cloud database hosted on a free Amazon cloud service, 'aws.'

7. Results

Several experiments were performed in which different joint positions were given to the robot, starting from the robot's home position. Results for four experiments are shown in Figure 8. When the joint position is changed via the virtual controller, the virtual robot moves to the given position along with the physical robot. The end positions with different joint states can be seen in Figure 8.

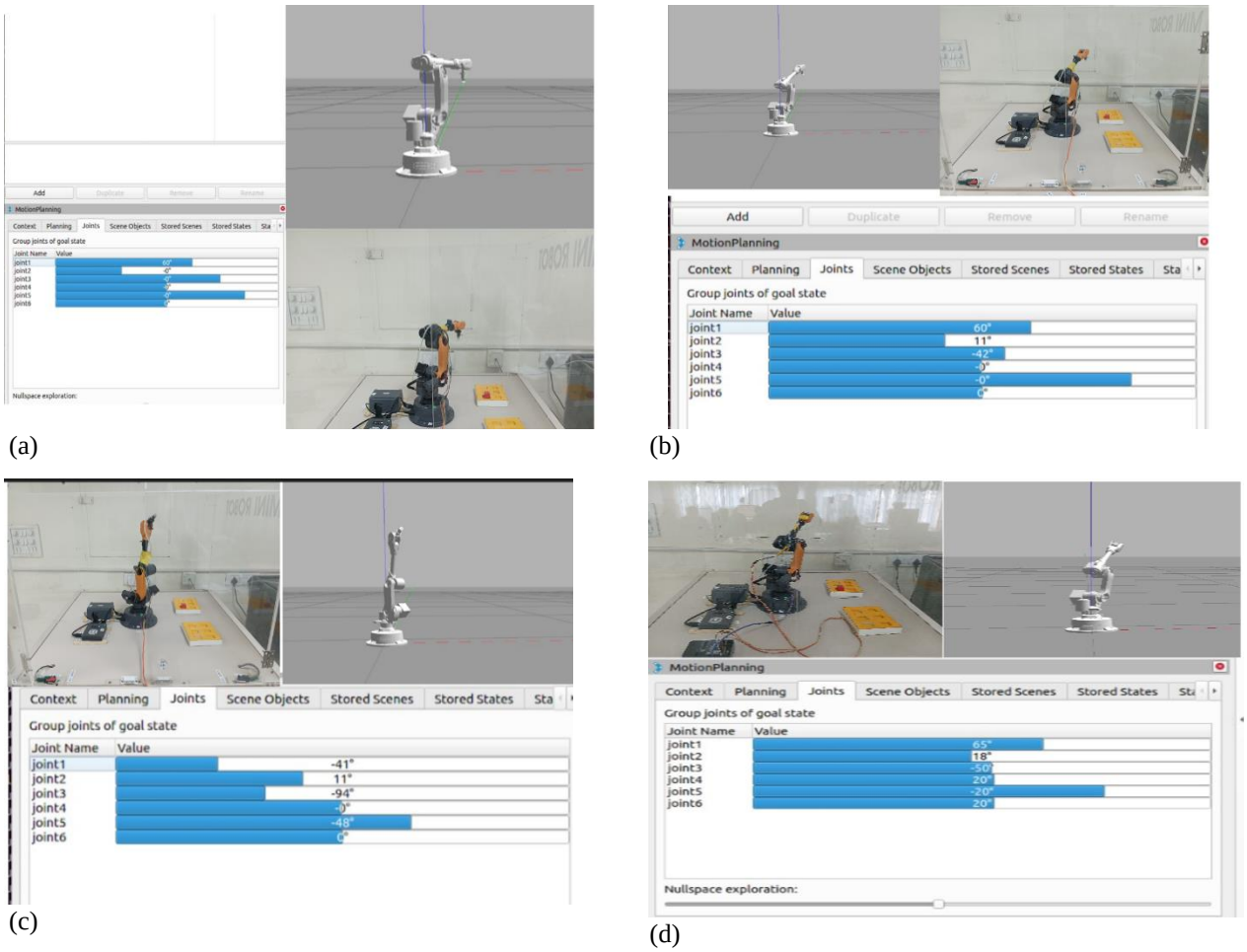


Figure 8: Actual and virtual position of the robot manipulator: (a) scenario 1 (b) scenario 2 (c) scenario 3 (d) scenario 4

The actual joint position is determined using a python script is compared with the given robotic position from the virtual controller, and the absolute error was calculated. Results for the four experiments are shown in Table 3, and its graph can be visualized in Figure 9.

Joint Number	Experiment 1			Experiment 2			Experiment 3			Experiment 4		
	Given Angle (°)	Actual Angle (°)	% error	Given Angle (°)	Actual Angle (°)	% error	Given Angle (°)	Actual Angle (°)	% error	Given Angle (°)	Actual Angle (°)	% error
1	60	59.998	0.003	60	59.998	0.003	-41	-40.988	0.029	65	65.011	0.029
2	0	0	0	11	11.002	0.0181 8	11	10.996	0.0364	18	17.968	0.0364
3	0	0	0	-42	-41.968	0.076	-94	-94.010	0.011	-50	-49.986	0.011
4	0	0	0	0	0	0	0	0	0	20	20	0
5	0	0	0	0	0	0	-48	-47.968	0.067	-20	-20.003	0.067
6	0	0	0	0	0	0	0	0	0	20	20.001	0

Table 3: Actual and given joint positions of the physical robotic arm and its percentage error

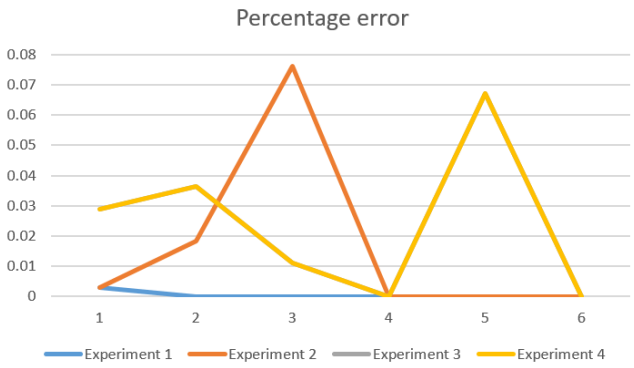


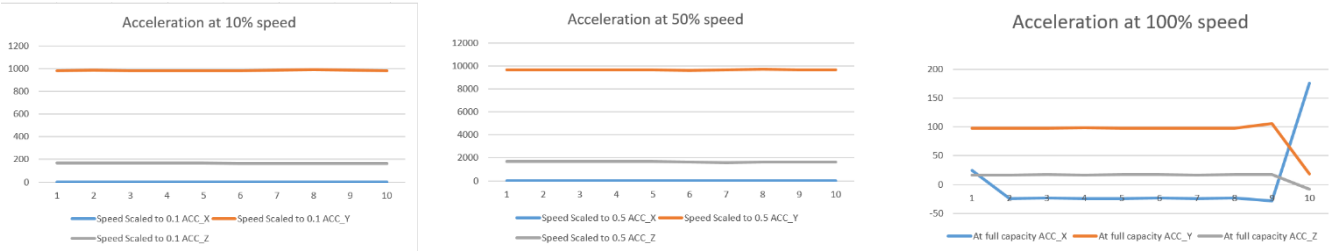
Figure 9: Absolute percentage error

In the second stage of experiments, the robot's speed was kept at 10%, 50%, and 100% of the full speed. The values of acceleration in x,y, and z coordinates of the upper links are shown in Table 5, along with the temperature of the base motor.

Table 4: Acceleration of robot at different speeds in mm/s2

Speed Scaled to 0.1				Speed Scaled to 0.5				Speed Scaled to 1			
ACC_X	ACC_Y	ACC_Z	Temp	ACC_X	ACC_Y	ACC_Z	Temp	ACC_X	ACC_Y	ACC_Z	Temp
0	983.4496	169.9908	31	0	9637.806	1665.91	31	24.598	98	16.17	31
0	987.2912	169.9908	31	0	9675.454	1665.91	31	-24.598	98	16.17	31.5
0	983.4496	169.9908	31.5	0	9637.806	1665.91	31.5	-23.814	97.608	16.954	32
0	983.4496	169.9908	32	0	9637.806	1665.91	32	-24.206	98.392	15.778	32.5
0	983.4496	169.9908	32	0	9637.806	1665.91	32.5	-24.206	98	16.954	32.5
0	979.608	166.1492	32	0	9600.158	1628.262	32.5	-23.422	98	16.954	32.7
0	987.2912	162.3076	32.5	0	9675.454	1590.614	33	-24.206	98	16.562	33.2
0	991.1328	166.1492	32.5	0	9713.101	1628.262	33.5	-23.03	98	16.954	33.7
0	987.2912	166.1492	33	0	9675.454	1628.262	33.5	-28.812	105.35	17.64	34
0	983.4496	166.1492	33	0	9637.806	1628.262	33.5	175.322	18.424	-8.428	34.1

The graphical visualization of acceleration of the upper link and temperature of the base motor at different speeds can be done in Figures 10 and 11, respectively.



(a) (b) (c)

Figure 10:acceleration (a) at 10% speed (b) at 50% speed (c) at full speed

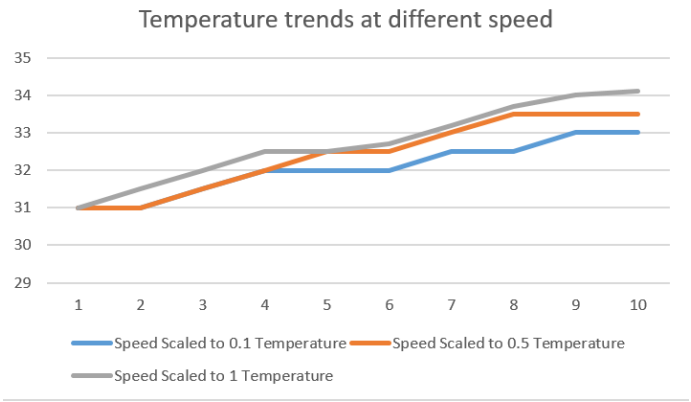


Figure 11: Temperature values at a different speed.

It was observed that both temperature and acceleration increase with an increase in speed. However, at full speed, a jerky motion is observed at the beginning and at the end of the motion.

8. Conclusion

The concept of digital twinning simulates the system's behavior by creating digital representations of actual items. An attempt has been made to build a digital twin of an articulated robotic arm using open-source software.ROS, Gazebo, RViz, and commercially available economic hardware viz. Raspberry Pi controller and sensors for acceleration and temperature measurement. The digital twin of the robot was used to operate the physical robot remotely. The maximum percentage error in the virtual and physical robot's position is 0.08 percent, which is negligible. The critical health conditioning parameters such as acceleration in the upper link and temperature in the base motor were captured in real-time and uploaded on a cloud service.

.

References

[1] C.-J. Liang, W. McGee, C. Menassa, and V. Kamat, "Bi-Directional Communication Bridge for State Synchronization between Digital Twin Simulations and Physical Construction Robots," in *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*, Oct. 2020, pp. 1480–1487. doi: 10.22260/ISARC2020/0205.

[2] https://www.ibm.com "How does a digital twin work"

[3] ROS.org, "ROS Documentation."

[4] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, Aug. 2009, vol. 3.

[5] W. Qian *et al.*, "Manipulation Task Simulation using ROS and Gazebo," Aug. 2014. doi: 10.1109/ROBIO.2014.7090732.

[6] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a Digital Twin for Real Time Remote Control Over Mobile Networks: Application of Remote Surgery," *IEEE Access*, vol. PP, p. 1, Aug. 2019, doi: 10.1109/ACCESS.2019.2897018.

[7] V. A. Siris, N. Fotiou, A. Mertzianis, and G. C. Polyzos, "Smart application-aware IoT data collection," *Journal of Reliable Intelligent Environments*, vol. 5, no. 1, pp. 17–28, 2019, doi: 10.1007/s40860-019-00077-y.

[8] W. M. S. Forum, "Mirobot ROS packages." 2021. [Online]. Available: https://github.com/wlkata/ROSForMirobot-master.git

[9] D. Sharma, K. Samuel, K. Ramoutar, T. Lowe, and I. David, "Raspberry Pi Based Real-Time Data Acquisition Node for Environmental Data Collection," *Journal of Basic and Applied Engineering Research*, vol. 4, pp. 307–312, Aug. 2017.