

Paper

053_SUGIARTO_ICAMME21

by Indar Sugiarto

Submission date: 03-Apr-2023 08:25PM (UTC+0700)

Submission ID: 2054596674

File name: 053_SUGIARTO_ICAMME21.pdf (558.63K)

Word count: 4120

Character count: 18897

Mini-SPERO: Personal Servicing Robot During Self-Isolation

Indar Sugiarto^{1, a)}, Iwan H. Sahputra^{2, b)}, Hariyo P. S. Pratomo^{3, c)}, and Hilton Tnunay^{4, d)}

¹*Department of Electrical Engineering, Petra Christian University, Siwalankerto 121-131, Surabaya 60236, East Java, Indonesia*

²*Department of Industrial Engineering, Petra Christian University, Siwalankerto 121-131, Surabaya 60236, East Java, Indonesia*

³*Department of Mechanical Engineering, Petra Christian University, Siwalankerto 121-131, Surabaya 60236, East Java, Indonesia*

⁴*GIPSA Lab, CNRS, Université Grenoble Alpes
621 Av. Centrale, 38400 Saint-Martin-d'Hères, Grenoble, France*

^{a)} Corresponding author: indi@petra.ac.id

^{b)}iwanh@petra.ac.id, ^{c)}hariyo_p@petra.ac.id, ^{d)}ishak.tnunay@gipsa-lab.fr

Abstract. During the Covid-19 pandemic, it is important to have systems that support patient recovery, for example, a robot that can help in delivering various medical supplies to patients. This paper describes the process of designing a low-cost robot that can be used for helping patients during their self-isolation. The robot is composed of two main parts: a manipulator and a mobile base. The manipulator's joints were made using a 3D printer articulated by several DC servos, whereas the mobile base was constructed using three DC motors equipped with omni-wheels. Both parts were controlled using a single Raspberry Pi that provides all means for data acquisition, control, and communication. The manipulator was also designed with the ability to avoid fall over by adding a controllable beam that works as a balancer. On the other hand, the mobile base was also equipped with a proximity sensor that can be used to measure the distance of objects surrounding the robot. The manipulator can be controlled by using a PID or a Fuzzy controller. From the experiment, we observed that the Fuzzy controller has a slower response compared to the PID one with a difference between 0.23 and 1.07 seconds. For the mobile base, we used only an inverse kinematics proportional controller with odometry calculation that results in the smooth robot movement with error at about 9.5% on the x-axis and 8.1% on the y-axis.

INTRODUCTION

Coronavirus disease 19 (COVID-19) has swept through countries around the world and can be considered as one of the worst pandemic diseases in the 21st century [1]. Treatments against this pandemic have been performed at various levels, including the ones that determine the successful recovery of an infected person. In Indonesia, after a person passes the rapid test or swab test process and is declared positive, the person is asked to self-isolate. This self-isolation mechanism can be carried out in hospitals, in special hotels, or in private homes. Currently, almost all referral hospitals can no longer accommodate patients for self-isolation, so patients are advised to self-isolate in special hotels or at home. To reduce the risk of Covid-19 transmission during self-isolation, we propose a robot that we call mini-SPERO (Support in Pandemic and Epidemic Robot). The aim is to help reduce direct interactions between patients and their families to reduce the risk of transmitting Covid-19 in homes/hotels.

This mini-SPERO robot was designed with the following specifications:

- it uses a telepresence mechanism to reduce direct/physical interactions between nurses/family members and patients, thereby reducing the risk of transmission

- the robot has a semi-autonomous navigation capability for reducing the burden on the operator to continuously control the robot and reduce the chance of the robot crashing due to collision with objects
- the robot that can "talk" in the sense that it provides a communication channel between the patient and other family members that will make the patient more relaxed and entertained, thereby reducing the feeling of depression caused by Covid-19 suffered by the patient

In general, the mini-SPERO robot was designed as a mobile manipulator with a telepresence capability. Minsky [2] defines telepresence as a "sense of being in another environment". Telepresence robots are made to overcome the weakness of many conventional immobile teleconference systems. Usually, a telepresence robot is composed of a video conferencing system placed on a mobile robot base [3].

Teleoperation on robots is not always accurate, for example, due to network latency or limited bandwidth on the network. Thus, telepresence robots need to have autonomous or, at least, semi-autonomous navigation capabilities. Due to inaccurate sensors and robot actuators, as well as a lot of noise in the real world, a probabilistic approach to a reliable robot navigation system is needed. Thrun et al. [4] using a probabilistic approach to the Minerva robot whose job is to be a guide for museum visitors. They designed a system that can perform localization and mapping simultaneously (Simultaneous Localization and Mapping – SLAM) so that the robot can make a map and know its position on the map at the same time.

Relying on fully autonomous operation carries a lot of risks, due to the changing environment (e.g., because people pass by, people move objects, etc.). For this reason, there needs to be a combination of teleoperation and fully autonomous. This system is called semi-autonomous (or shared autonomy) which regulates the level of autonomy of the robot. This concept can be applied to various robotic platforms. For example, Leeper et al. [5] define four strategies for robotic grasping from the lowest to the highest level of autonomy as follows: direct control, waypoint following, grasp execution and grasp planning.

The semi-autonomous concept can also be applied to a telepresence robot. Takayama et al. [6] designed an assisted driving system where the robot will follow the operator's orders if it finds no obstacles around it based on the trajectory simulation. This system prevents the robot from hitting obstacles due to less accurate commands from the operator. Mini-SPERO also uses the semi-autonomous concept for its navigation system. This is an improvement to our previous robot called SPERO, which was a robot that can be controlled remotely for carrying medicines and food in a hospital. In this paper, we describe how the mini-SPERO robot was designed and tested.

METHODS

The mini-SPERO is composed of two parts: the upper body manipulator and the lower mobile platform. Figure 1(a) shows the fully constructed mini-SPERO, whereas Fig. 1(b) shows its drawing sketch.

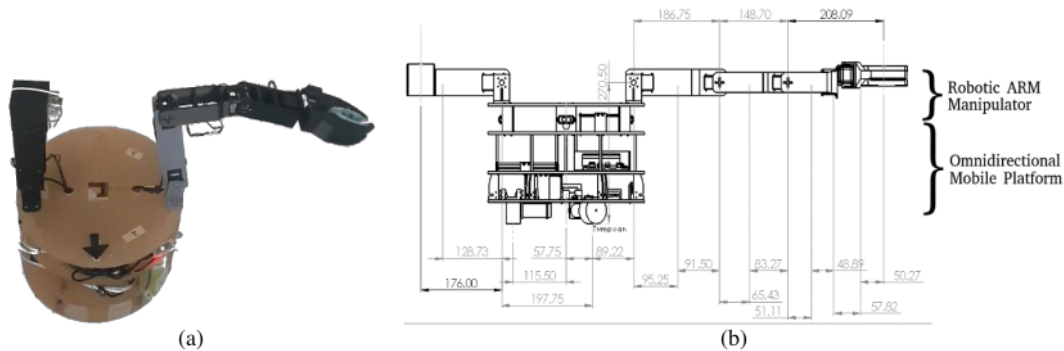


FIGURE 1. The physical appearance and the drawing sketch of mini-SPERO. All measured dimensions in the drawing sketch are in mm

Omni-Directional Mobile Platform

The omni-directional robot system was constructed using three omni wheels driven by three DC motors (which are equipped with an encoder). These actuators are controlled by a Raspberry Pi. The Raspberry Pi receives input from the user to determine the destination position and the motion direction of the robot. Limit switches will be placed at each point a total of 6 pieces and function as a detector of objects/obstacles around the robot. The robot is also equipped with a camera to detect the shape of an object. This camera will be controlled by a Jetson Nano. Figure 2 shows the block diagram of the robot control system.

The Raspberry Pi will drive the DC motors using inverse kinematics calculation based on odometry data provided by the encoder. The odometry calculation was used to determine the distance the robot has traveled. The input from the infrared sensor was used to detect the distance of an object from the robot. The output from the pi-camera will be processed by the Jetson Nano. The camera image processing was carried out on the Jetson Nano to determine the length and the width of the detected object. This information will be passed on to the manipulator so that the manipulator could grasp the object correctly. Both microcontrollers (Raspberry Pi and Jetson Nano) are connected using an ethernet switch. Figure 3 shows the kinematics configuration of the mobile platform.

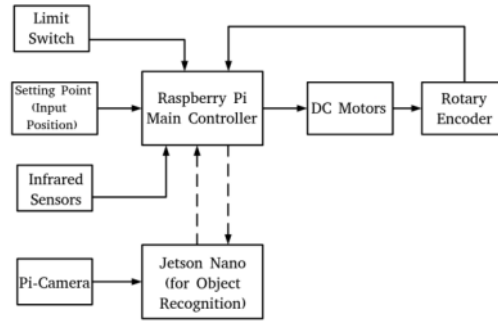


FIGURE 2. Block diagram of the mobile platform control system

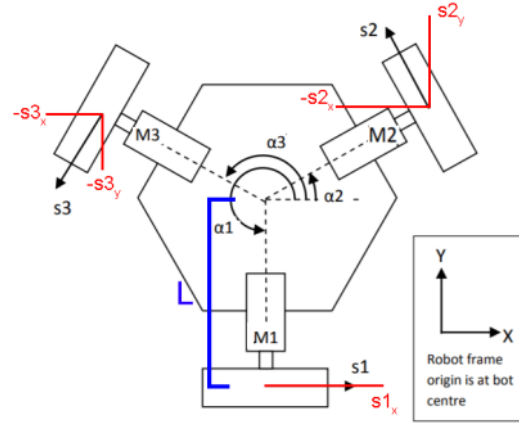


FIGURE 3. Kinematics configuration of the mobile platform. Three DC motors are used and arranged in a way to produce a complete circular motion

The robot's motion is determined by its kinematics. To control its motion, we have derived the kinematics equation of the robot as shown in (1) according to [7].

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} 2/3 & 0 & 1/3 \\ 0 & 1/\sqrt{3} & 1/3 \\ -1/3 & -1/\sqrt{3} & 1/3 \end{bmatrix} \begin{bmatrix} x/r \\ y/r \\ \theta \end{bmatrix} \quad (1)$$

where ϕ_i is the motion angle of each motor, r is the radius of each wheel (which is 0.0246 m), and L is the distance from the center of the robot to the center of the wheel (which is 0.12 m). This formula is used in a closed-loop proportional controller shown in Fig. 4. In general, a proportional controller will produce a fast response and is sufficient to be used in a simple robot architecture [8].

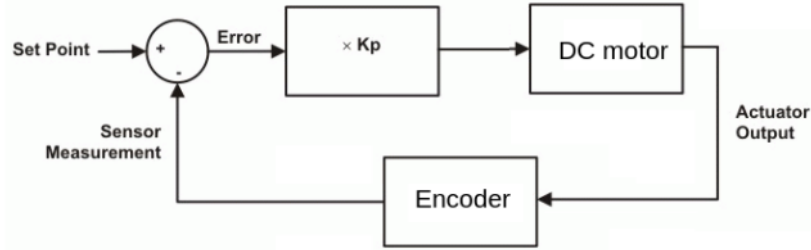


FIGURE 4. Proportional controller diagram for controlling the DC motors. The proportional coefficient K_p was determined as 0.7 which can be obtained from testing data

Robotic ARM Manipulator

The robotic arm manipulator was made using plastic materials and was designed using SolidWorks 2018 software. The minimum reach of the manipulator is 50 cm which is divided into three segments. The manipulator has 4-DOF (degree of freedom) with the end effector in the form of a gripper as shown in Fig. 2. The joints are implemented using several digital servos namely Dynamixel MX-64, MX-28 and AX-12A.

Moment calculations were carried out to determine the load from the balancer so that the robot does not fall over when picking up an object. The moment is calculated from the fulcrum which is the front wheel of the platform. The position of the fulcrum and servo can be seen also in Fig. 2. The mass of the arm is positioned at the center of gravity (CG) of each arm obtained from the SolidWorks 2018 software. The maximum load to be lifted is 1 kg.

In order to obtain the stability of the platform, the total moment should be equal to zero [9,10]. The balancer load, then, can be calculated as follows:

$$\Sigma \tau = 0$$

$$\tau_{right} - \tau_{left} = 0 \quad (2)$$

where $M_{right} = 8.988 \text{ Nm}$ and $M_{left} = 8.783 \text{ Nm}$, which can be obtained using the physical dimension and constraints of the arm and the balancer. The balance weight w_b then can be calculated using the following condition:

$$M_{left} = 3.777 + w_b * 0.37375 \quad (3)$$

which then produced $w_b = 1.42 \text{ kg}$. This balancer ensures the equilibrium state of the robot so that the platform will not fall over when lifting a load with a maximum weight of 1 kg.

The calculation of inverse kinematics of the manipulator was based on a geometric basis. It requires only two dimensions, namely x and y . The result of this inverse kinematics is the angle that composes the manipulator to reach the specified x and y coordinates. For controlling the manipulator, two types of controllers were used: a fuzzy controller and a PID controller. A fuzzy controller is a type of knowledge-based computing originating from fuzzy set theory with structure depicted in Fig. 5 [11].

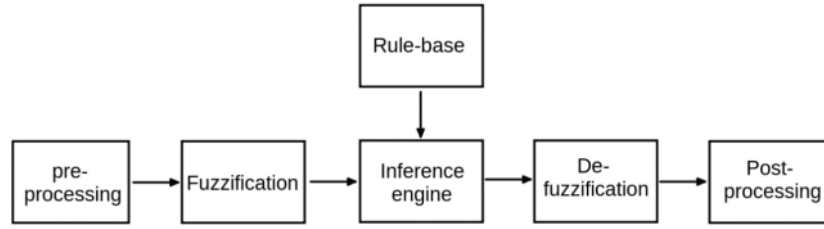


FIGURE 5. Generic fuzzy controller structure

The fuzzy controller was designed with inputs from platform angles, angular velocity (ω) of the falling platform, and the angular acceleration (α) of the falling platform. The output of the fuzzy control is the angular velocity as well as the angular position of the counter balance. Membership functions (Mfs) of the angle input can be seen in Fig. 6 with the universe from 0° to 90° .

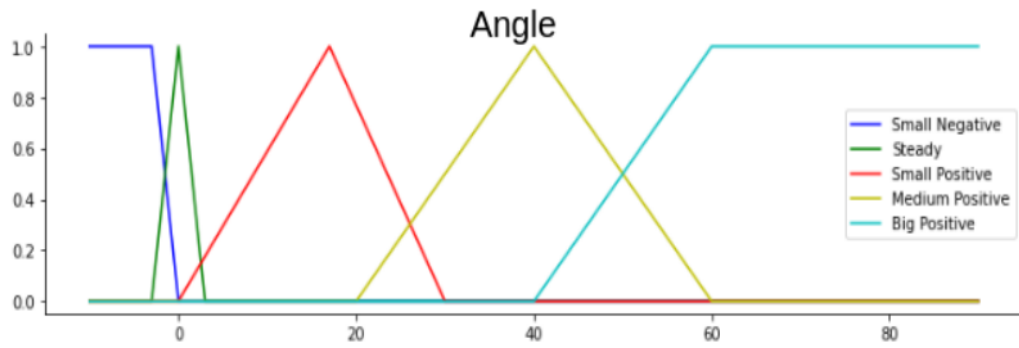


FIGURE 6. Membership functions for input angles

The membership function of ω can be seen in Fig. 7. This MF is divided into seven fuzzy sets as follows: very slow (VS), slow (SL), below average (BA), average (AV), above average (AA), fast (FS), and very fast (VF). This ω was set to have values within the range: $U_{\omega} = \{x \mid 0 \leq x \leq 270\}$

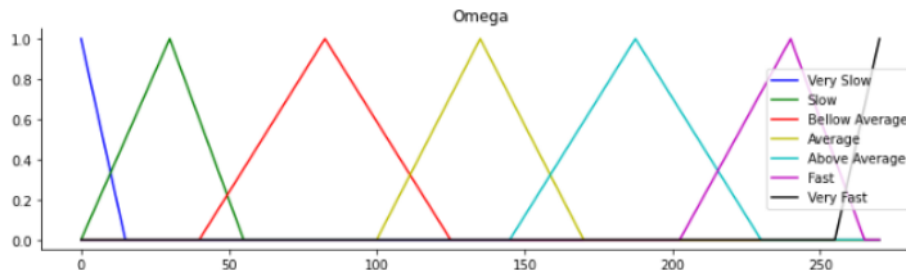


FIGURE 7. Membership functions for angular velocity ω

The angular acceleration membership functions can be seen in Fig. 8. It is divided into five fuzzy sets, namely very small (VS), small (SM), medium (MD), big (BG), and very big (VB), where $U_{\alpha} = \{x \mid 0 \leq x \leq 180\}$.

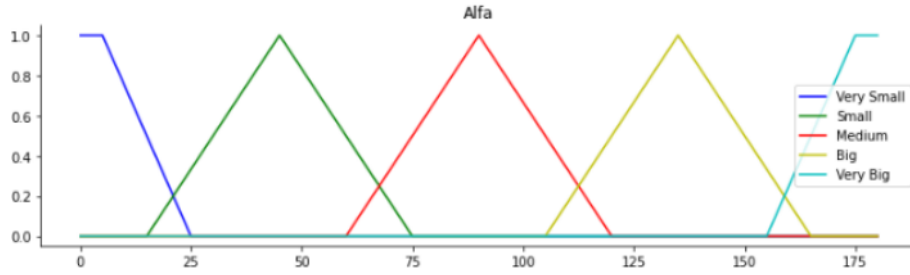


FIGURE 8. Membership functions for angular acceleration alpha

The three inputs above will be used as a rule to control the output of the controller. For the output, we defined five membership functions, namely: small negative (SN), steady (ST), small positive (SP), medium positive (MP), and big positive (BG). The outputs are the angular position of the balancer (θ_b) and the angular velocity of the balancer (ω_b). The rules used to produce the angular position output can be seen in Table 1, whereas the output for angular velocity can be seen in Table 2. From these, we used CoG (center of gravity) method for computing the fuzzy controller output.

A PID controller was also used in comparison to the fuzzy controller. The PID constant tuning is performed by using Simulink Matlab R2018a software. The PID controller was modelled in the Simulink of the Matlab software, which can be seen in Fig. 9.

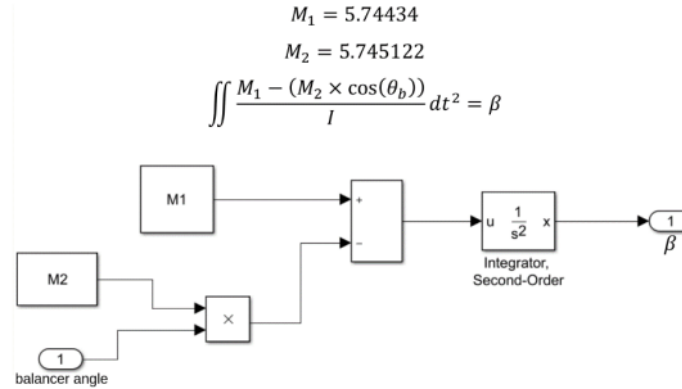


FIGURE 9. Simulink model for the PID-based controller for the manipulator balancer

RESULT AND DISCUSSION

We have performed initial experiments on the robot using several scenarios. Due to the complexity of the robot, we could not provide experimental results of the robot as one, but as separate parts.

Mobile Platform Experiments

The main objective of the mobile base platform is to carry the robotic arm manipulator and move it to a certain position and orientation. Thus, we are interested to evaluate the mobile platform performance in this direction. Table 3 and Table 4 show the experiment results on moving the mobile platform to a certain position.

For the cartesian movement (i.e., the robot moves only in y or x direction), we observe that the robot moves more reliably in Y direction. It produces the error at about 6.8% during Y movement; whereas in X movement, the robot experienced slippage that results in a 9.1% average deviation from the intended direction. For diagonal robot movements, there is an average error of 9.5% on the x-axis and 8.1% on the y-axis. All robots are set to reach the target position within 4 seconds, except the target position $x = 120, y = 120$ and $x = -120, y = -120$ where the robot's movement

time is set at 5 seconds because the motor 3's speed has exceeded the maximum (for 4 seconds). Looking at the data shown in Table 4, if the final position is small (in this test $x = \pm 40$, $y = \pm 40$), it causes an error that tends to be greater, because the required motor speed is below 20 rpm so that proportional control has difficulty in reaching that point.

TABLE 1. Fuzzy rules to produce the angular position and angular velocity outputs

Angle	Alpha Omega	Angular Position Rule					Angular Velocity Rule				
		VM	SM	MD	BG	VB	VM	SM	MD	BG	VB
SN	VS	SM	SM	SM	VM	VM	VS	VS	VS	VS	VS
	SL	SM	SM	VM	VM	VM	VS	VS	SL	SL	SL
	BA	VM	VM	VM	VM	VM	SL	SL	SL	SL	SL
	AV	VM	VM	VM	VM	VM	SL	SL	SL	AV	AV
	AA	VM	VM	VM	VM	VM	SL	SL	AV	AV	AV
	FS	VM	VM	VM	VM	VM	AV	AV	AV	AV	AV
	VF	VM	VM	VM	VM	VM	AV	AV	AV	AV	AV
ST	VS	VM	VM	VM	VM	SM	VS	VS	VS	SL	SL
	SL	VM	VM	SM	SM	SM	VS	VS	VS	SL	SL
	BA	SM	SM	SM	MD	MD	SL	SL	SL	SL	SL
	AV	MD	MD	MD	MD	MD	SL	SL	AV	AV	AV
	AA	MD	BG	BG	BG	VB	SL	AV	AV	AV	FS
	FS	BG	BG	BG	VB	VB	FS	FS	FS	VF	VF
	VF	BG	BG	VB	VB	VB	FS	FS	VF	VF	VF
SP	VS	MD	MD	MD	MD	BG	SL	SL	SL	SL	SL
	SL	MD	MD	MD	MD	BG	SL	SL	SL	SL	SL
	BA	MD	MD	MD	BG	BG	SL	AV	AV	AV	AV
	AV	MD	BG	BG	BG	BG	AV	AV	AV	AV	AV
	AA	BG	BG	BG	BG	BG	AV	FS	FS	FS	FS
	FS	BG	BG	VB	VB	VB	FS	FS	FS	VF	VF
	VF	BG	VB	VB	VB	VB	FS	VF	VF	VF	VF
MP	VS	MD	MD	MD	MD	BG	SL	AV	AV	AV	AV
	SL	MD	MD	BG	BG	BG	AV	AV	AV	AV	AV
	BA	MD	MD	BG	BG	BG	AV	AV	AV	AV	AV
	AV	BG	BG	BG	BG	BG	AV	AV	FS	FS	FS
	AA	BG	BG	BG	BG	BG	AV	FS	FS	FS	FS
	FS	BG	BG	BG	VB	VB	FS	FS	FS	FS	FS
	VF	BG	BG	VB	VB	VB	FS	VF	VF	VF	VF
BG	VS	BG	BG	BG	BG	BG	AV	AV	AV	AV	AV
	SL	BG	BG	BG	BG	BG	AV	AV	FS	FS	FS
	BA	BG	BG	BG	BG	VB	FS	FS	FS	FS	FS
	AV	BG	BG	BG	VB	VB	FS	FS	FS	FS	FS
	AA	BG	BG	VB	VB	VB	FS	FS	FS	FS	FS
	FS	VB	VB	VB	VB	VB	FS	VF	VF	VF	VF

Angle	Alpha Omega	Angular Position Rule				Angular Velocity Rule					
		VM	SM	MD	BG	VB	VM	SM	MD	BG	VB
		VF	VB	VB	VB	VB	VF	VF	VF	VF	VF

Input (cm)		End Position (cm)		Error (cm)	
X	Y	X	Y	X	Y
0	40	8	40	8	0
0	80	14	78	14	2
0	120	7	125	7	5
40	0	38	16	2	16
80	0	86	3	6	3
120	0	135	5	15	5
0	-40	-7	-35	7	5
0	-80	-20	-77	20	3
0	-120	-15	-116	15	4
-40	0	-38	-15	2	15
-80	0	-82	3	2	3
-120	0	-131	-20	11	20
Average Error				9.1	6.8

Input (cm)		End Position (cm)		Error (cm)	
X	Y	X	Y	X	Y
40	40	35	38	12.5	5
80	80	73	65	8.7	18.7
120	120	125	102	4.1	15
40	-40	38	-37	5	7.5
80	-80	72	-75	10	6.2
120	-120	115	-110	4.1	8.3
-40	40	-33	37	17.5	7.5
-80	80	-70	74	12.5	7.5
-120	120	-110	111	8.3	7.5
-40	-40	-35	-36	12.5	10
-80	-80	-73	-78	8.8	2.5
-120	-120	-108	-118	10	1.7
Average Error				9.5	8.1

Manipulator with Balancer

The experiments on the manipulator with balancer were performed by first removing the mobile base from the robot. This was to make sure that the performance of the manipulator controllers was not affected by the jittery movement of the mobile base. Two controllers are used for the manipulator: the fuzzy controller and the PID controller. Hence, we obtained two results from the experiments.

The experiments were conducted using two different loads: 200 grams and 500 grams scales. The test is carried out by moving the end effector at the coordinates (400,200) to pick up the object and then return to the initial position. The position and the width of the object (grabber opening) are obtained from sensors on the mobile platform.

In the first experiment, the manipulator tried to lift an object weighing 200 grams. The load was lifted in 5.02 seconds and stabilized after 6.18 seconds; thus, the time needed to make the stabilization is 1.16 seconds. The balancer was successfully positioned at 49.57° and the speed was recorded at 51.68 rpm.

The second experiment was carried out by lifting a weight of 500 grams. The manipulator started to lift at 4.14 seconds and stabilized at 8.21 seconds; thus, it needed 4.07 seconds to stabilize. The output from the fuzzy controller for the balancer showed an angle of 67.5° and the balancer speed was 22.8 rpm.

The experiment with the PID controller was performed after the test with the fuzzy controller. The values of PID constants were obtained with the help of the Matlab software, which are $K_p = 0.19185$, $K_i = 0.021142$, $K_d = 0.42852$. In the first PID experiment by lifting a weight of 200 grams, the time needed to stabilize the platform was 0.93 seconds. The output of the omega angle position is 90° . The angular speed of the balancer is kept constant at 34.2 rpm. In the second PID experiment by lifting a load of 500 grams, the time needed to stabilize the platform was 3 seconds. The output of the controller is 90° with a constant speed set of 34.2 rpm. The results of the aforementioned experiments are presented in Table 5.

TABLE 5. Comparison of fuzzy vs PID controller used by the manipulator.

Experiments	I		II	
	Fuzzy	PID	Fuzzy	PID
Stabilization time (s)	1.16	0.93	4.07	3
Balancer Angle ($^\circ$)	49.5	90	67.5	90
Balancer Angular Speed (rpm)	51.68	34.2	22.8	34.2

From the experiments, we observed that the stabilization time of the PID is faster than the fuzzy controller. This is because the angular position of the balancer that is removed from the controller, immediately touches the maximum, so that when there is a small error, correction is immediately carried out.

On the other hand, the fuzzy controller is simpler in design due to its use in linguistic rules. It was found that the response from the fuzzy control was 0.23 seconds slower for the experiment I. In the experiment II, it was 1.07 seconds slower than the PID controller. However, the fuzzy controller has shown a good control result.

CONCLUSION

This paper describes the process of designing a mobile manipulator that consists of a 3-wheel omni-directional robot as the mobile base and a 4-DOF robotic arm manipulator at the upper body. This robot is intended to be used as a personal servicing robot that will find itself very useful in the self-isolation scenario during the Covid-19 pandemic. The first part of the robot, which is a 3-wheel omni-directional mobile robot, is controlled by a Raspberry Pi using inverse kinematics for determining the direction and speed of the robot. The odometry was used to estimate the movement of the robot's position. From the experiments, we observed that the error rate of the robot movement is about 9.51% on the x-axis and 8.12% on the y-axis. This is mainly influenced by accuracy and the fast response of the proportional controller used during the inverse kinematics computation. The second part of the robot, which is the upper body manipulator, shows promising results on the use of both the PID controller and the fuzzy controller. From the designer's perspective, the fuzzy controller is easier to design but the response is a bit slower than the PID controller. We found the delay at about 1.16 seconds to 4.07 seconds for the fuzzy controller to stabilize the platform, whereas, for the PID controller, the delay is only at about 0.9 seconds to 3 seconds. For our future work, we intend to test the overall robot performance in a real situation involving patients of Covid-19 being treated in a self-isolation situation.

ACKNOWLEDGMENTS

This research was supported partially by Lembaga Penelitian dan Pengabdian kepada Masyarakat Universitas Kristen Petra (grant no: 02/HBK-PENELITIAN/LPPM-UKP/XI/2020). The authors would like to thank E. O'neil and Y. K. A. Sarumaha for the initial robot design and collecting the experimental data.

REFERENCES

1. J. Feehan, and V. Apostolopoulos, *Maturitas* **149**, 56-58 (2021).
2. M. Minsky, Telepresence, *OMNI Magazine*, (1980), pp. 44-52.
3. A. Kristoffersson, S. Coradeschi, and A. Loutfi, *Advances in Human-Computer Interaction* **2013**, 902316 1-17 (2013).
4. S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, and J. Schulte, *The International Journal of Robotics Research* **19**(11), 972-999 (2000).
5. A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 1-8 (2012).
6. L. Takayama, E. Marder-Eppstein, H. Harris, and J. M. Beer, *2011 International Conference on Robotics and Automation (ICRA)*, 1883-1889 (2011).
7. H. Taheri, and C. X. Zhao, *Mechanism and Machine Theory* **153**, 103958 (2020).
8. A. Izadbakhsh and S. Khorashadizadeh, *Journal of the Brazilian Society of Mechanical Sciences and Engineering* **42**(91), 1-12 (2020).
9. E. F. Kececi, *Mechatronic Components: Roadmap to Design*, (Elsevier Science, 2018), pp.145-154.
10. M. Popovic and M. Bowers, *Biomechatronics* (Academic Press, Elsevier, 2019), pp.11- 43.
11. E. A. Muraveva, E. A. Shulaeva, P. N. Charikov, R. R. Kadyrov, M. I. Sharipov, A. V. Bondarev, and A. F. Shishkina, *Procedia Computer Science* **120**, 487-494 (2017).

Paper 053_SUGIARTO_ICAMME21

ORIGINALITY REPORT

7%

SIMILARITY INDEX

7%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

jirae.petra.ac.id

Internet Source

7%

Exclude quotes On

Exclude bibliography On

Exclude matches < 3%