Journal of Systemics, Cybernetics and Informatics (2022) 20(4), 45-52
https://doi.org/10.54808/JSCI.20.04.45

Robot Arm Control System for Assisted Feeding of People with Disabilities in their Upper Limbs

Daniella ARNÁEZ
School of Electronic Engineering, Universidad Peruana de Ciencias Aplicadas
Lima 15023, Peru

Fiorela MANCO
School of Electronic Engineering, Universidad Peruana de Ciencias Aplicadas
Lima 15023, Peru

José OLIDEN
School of Electronic Engineering, Universidad Peruana de Ciencias Aplicadas
Lima 15023, Peru

Guillermo KEMPER
School of Electronic Engineering, Universidad Peruana de Ciencias Aplicadas
Lima 15023, Peru

ABSTRACT

This work proposes a robot arm control system for assisted feeding of people with reduced functionality in their upper limbs. It aims at improving their quality of life by helping users recover their independence when feeding, aided by the proposed system. Previous research presents solutions that often lack functionality to meet the user's needs, such as a lack of emergency functions or the use of passive feeding techniques, due to the absence of adequate human-robot interaction. The proposed solution involves the design of an interface adapter between the robot arm and the spoon, for the correct transport and positioning of the food. Moreover, a PD-type electronic controller is implemented for the robot arm; it includes gravity compensation and trajectories defined from the detection of the user's position. Additionally, the system has two safety features: an emergency button and a proximity warning that triggers when undesired objects are too close to the robot arm. The proposed system was validated through position tests and interaction with people using rice and oatmeal. When carrying out the tests with rice, 80% success was obtained, while in the case of oatmeal, 98.9% success was achieved.

Keywords: robot arm, feeding assistance, control system, active feeding, disability, upper limbs.

1. INTRODUCTION

According to [1] the World Health Organization, in the United States about 42% of non-elderly adults with disabilities reported not being able to perform daily activities such as getting out of bed or toileting. Likewise, 36.9% of people with disabilities in Europe depend on another person's care and support to carry out their daily activities [2].

Among these activities, not being able to feed themselves negatively affects a person's autonomy. Research on robot-arm-assisted feeding aims to help people with disabilities. However, commercial solutions and those that have been proposed in the state-of-the-art present limited functions and, in some cases, are not adaptable for people who completely lack movement in their upper limbs.

For example, in [3] Candeias et al. propose a system based on artificial vision to determine the trajectory of movements; however, the accuracy of the user's head position obtained by the Discriminative Optimization (DO) method is not very accurate. The authors recognize that the method needs improvements when generalizing to different users, since it was programmed for a given 3D face model.

In [4] Li et al., propose a platform in conjunction with a robot arm, which collects food from different plates by voice interaction. Its master-slave control structure allows the speed of the arm to be varied according to the function indicated by the user. However, speech recognition increases the computational load, which could affect the smoothness of the execution of the trajectories and increase cost of this system in order to avoid it.

In [5] Park et al., couple a system to the PR2 robot, using visually guided movements to autonomously pick up food and deliver it into the user's mouth using active feeding techniques. The disadvantage is that the characteristics of the PR2 robot do not allow a friendly human-robot interaction for the user, either because of its size or its speed.

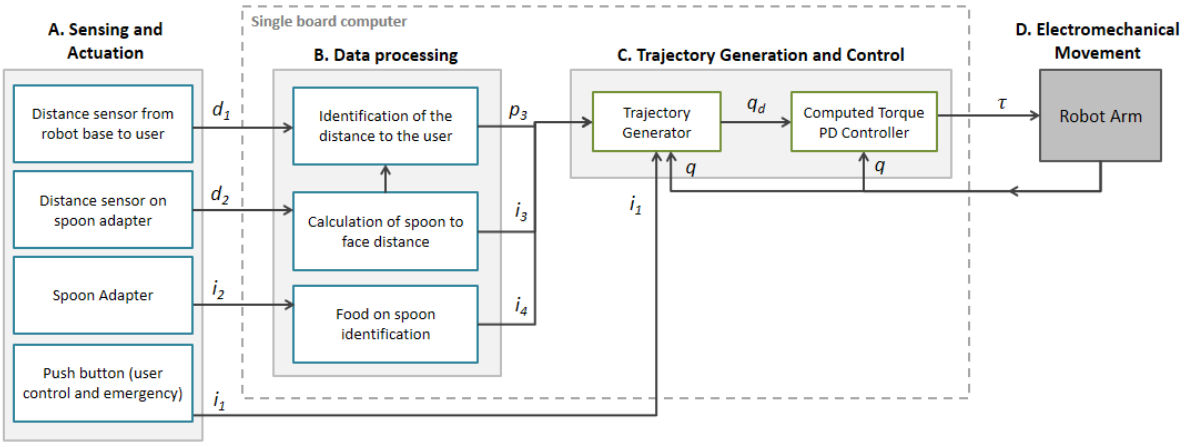

Figure 1. Block diagram of the proposed control system.

On the other hand, in [6] Liu et al., use a method that consists of demonstration learning programming to generate a feeding trajectory. It also uses a visual interaction method based on Deep Learning to control the robot through signs or gestures such as closing the eyes or opening the mouth. However, the system requires that it be taught the robotic arm's scoop delivery position by dragging and guiding the robotic arm. Encoders are used to obtain and record the angles of each axis and form the feeding trajectories.

The solution in [7] Haidar et al. consists of controlling the robot arm by blinking at a camera positioned in front of the person. Computer Vision techniques are used for this recognition, but although the system detects an eye of the user, it does not identify the position of the mouth.

Finally, in [8] Guo et al. use a mechanical control model by simulating the movement of the robotic arm, which allows observing the angles necessary for the robot to carry the food and pick up the spoon. However, the predefined mechanical movements in the control system are limited due to the lack of degrees of freedom (DOF). According to [5] Park et al., the causes that robots provide limited meal assistance is due to a low DOF arms, and limited sensing capabilities.

As has been shown, the different proposals, although they partially solve the problem, have limitations and deficiencies such as the lack of safety features for the user, the lack of versatility in the use of the system in multiple environments and the lack of methods that facilitate the interaction between the person and the robot.

In this context, the present work proposes a solution based on the use of a 5 DOF robot arm with rotational movements, which incorporates the use of actuators such as a user button, which is used to request a new spoon with food or to stop the robot arm in case of an emergency. Proximity sensors are also used to calculate the distance between the robot arm and the person. In this way the robot arm does not invade the user's personal space or collide with them. These peripherals and actuators are part of the computational control system based on a Raspberry Pi single board computer.

The proposed system was validated through position tests using rice and oatmeal, due to the consistency of these foods. A total of 35 tests were performed per food, since this is an approximate number of tablespoons that are required to finish the plate of food. The tests consist of evaluating different events during the execution of the system. When carrying out these evaluations with rice, 80% success was obtained in the 35 tests carried out, while in the case of oatmeal, 98.9% success was achieved. The details of the proposed control system will be described in the following sections along with the tests and obtained results.

2. DESCRIPTION OF THE CONTROL SYSTEM

The proposed control system is illustrated in the block diagram shown in Figure 1. The system generates and executes a trajectory for the movement of the arm to feed the user, fulfilling certain logical conditions and safety measures.

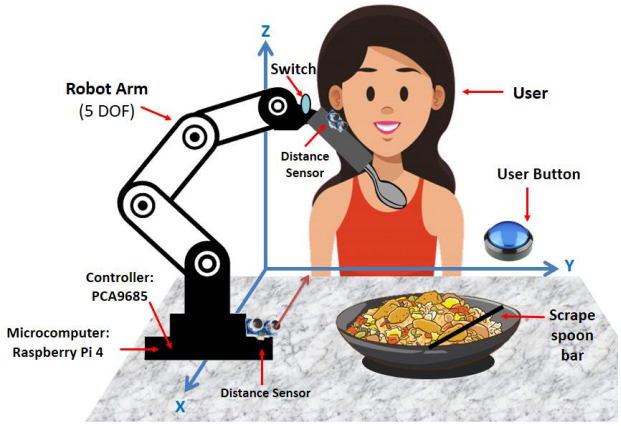

Figure 2. Pictorial representation of the robot arm interacting with the user.

The system is made up of 4 main parts: sensing and actuation, data processing, trajectory generation, and electromechanical movement and control. Figure 2 shows a pictorial representation of the robot arm interacting with the user.

2.1. Sensing and actuation

The sensing and actuation initially aim to collect data regarding the state of the user button (activated or not activated) and at the same time obtain information provided by the distance sensors to determine the distance between the base of the robot arm and the user, and between the face of the person and the spoon.

Prompts are entered into the system, through the user/emergency control button, depending on the running state of the system. A large button was chosen because it is easier for users to press it.

The distance sensor located in the spoon is used to identify if there is a person within the threshold considered dangerous during the execution of the trajectory. If this distance is less than 15cm, it indicates the appearance of an obstacle while the robot arm is in motion. Upon this detection, a movement halt order will be sent to the robot, and it will stop until a safe distance is detected again. In this way, the user's safety is not put at risk. It should be noted that the distance from the robot's base to the user will be identified in an uncontrolled environment with the distance sensors.

In Figure 3, a schematic of the implemented electronic system is presented. It consists of sensors and actuators that are used as input signals to the system. The measurement of the distances from the robot arm to the user is carried out by two ultrasound-type distance sensors. These are located at the base and at the last link of the robot arm. The HC-SR04 ultrasound sensor was chosen because it has sufficient sensitivity for the developed application and for the Raspberry Pi to perform constant measurements without requiring many computing resources.

A 3D printed adapter in the spoon senses if it contains food or not, to determine the next trajectory to execute. Figure 4 shows the 3D model of the adapter for the spoon and the adapter located on the last link of the robot arm. This adapter uses a contact switch, which is activated by the spoon when it is lifted slightly by the weight of the food, making it easy to decide whether to stay on the trajectory to serve the spoon or to move to the next trajectory in the execution of the program, during data processing. This adapter meets the requirements that the system needs, since it has a space to place the spoon and adjust it so that it does not become uncoupled during the movement of the robot arm. The contact switch sends a 5V voltage signal to indicate that there is food on the spoon. This was chosen because its shape is suitable to be pressed with the lever movement made by the spoon in the designed adapter. In addition, it has a space on one side to place the contact switch, so that during the movement of the last degree of freedom the position of the spoon with respect to the switch is not lost. The adapter will fit with a circular mechanical piece in the robot arm.

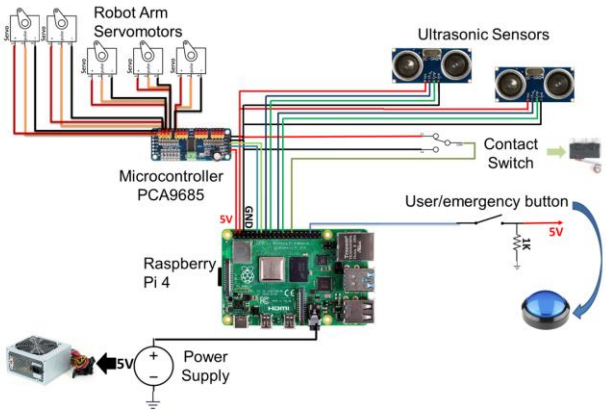

Figure 3. Schematic of the electronic circuit implemented.

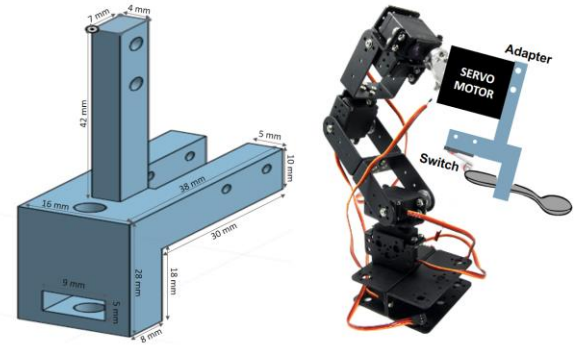

Figure 4. Adapter 3D model.

2.2. Data processing

Data processing consists of interpreting input signals and sensor readings to execute the required stage. The processing is done on a Raspberry Pi 4, which can run multiple programs simultaneously. The presented diagrams reflect the importance of structuring the logic of the program in an orderly manner.

Step 1: The first trajectory will be executed when the user button is pressed, and the robot arm is in its initial position.

Step 2: To execute the spoon transfer sequence from the plate to the user, the adapter switch is pressed when serving the spoon with food.

Step 3: To emit an emergency signal, the button is pressed during the execution of the trajectories. So, the robot stops and then returns to the starting position.

Step 4: The distance between the user and the spoon that supports the robot arm is calculated, as a safety measure.

Step 5: The distance between the user and the base of the robot arm is calculated. The scoop is moved between 8 and 15 cm from the base depending on the range corresponding to the distance, to bring the entire scoop closer to the user's face.

Step 6: When the button is pressed after the user picks up the food, the robot arm returns to its initial position and the trajectory is restarted.

2.3. Trajectory generation and control

The sequence of trajectories of the robot arm begins when it is in the initial position. The first trajectory starts in this position until the spoon is placed over the plate, where an action is performed to serve the spoon. This action is repeated until the adapter detects that the spoon has been served. The second trajectory starts over the plate and ends in front of the user. By executing this trajectory, the underside of the spoon is wiped against a bar on the plate to remove excess food and prevent spillage. During the transfer from the plate to the person, the distance from the spoon is sensed to indicate the appearance of an obstacle within the threshold considered dangerous for the user and to stop the execution of this trajectory. The third trajectory has the objective of reaching the user's feeding point, sensing the distance between the user and the base of the robot arm in real time and bringing the spoon closer. After the user has eaten, they must press the user button again if they want another spoon with food, executing the return trajectory to the initial position and repeating the sequence. At any time during the execution of the trajectories, the button can be pressed to interrupt the process and the robot arm returns to its initial position, away from the user due to the emergency signal.

Previous research, like Mišeikis et al. [10] has identified the kinematic model of the robot to apply decoupled control. In this project, the control of the robot is proposed, using both the kinematic model and its dynamic model. With dynamic modeling it will be possible to control the forces of the robot with greater efficiency. Then, the planning of trajectories by position in coordinates will be carried out, which will be performed by servos. Finally, the dynamic modeling will be carried out considering exclusively the force of gravity because it is a small robot. In this way, computed torque PD control with gravity compensation can be performed.

2.3.1. Kinematic, Dynamic Modeling and Trajectory Generation

The development of the proposal begins with the kinematic modeling of the robot arm with 5 degrees of freedom to be used. The robot kinematics determine the positions and coordinates of the robot links. Using the Denavit-Hartenberg algorithm, it is possible to determine the relationship of coordinates in space between the links [9] in Arnáez.

Step 1: The coordinate system in Figure 5 and the parameters in Table 1 result from the Denavit-Hartenberg algorithm.

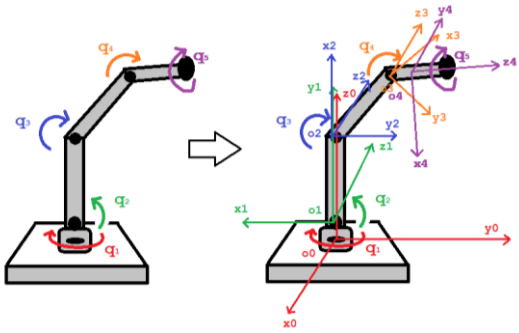

Figure 5. Robot arm coordinate system.

For the calculations below, the length to the first joint is 1.5 cm, the length of the first link to be 10.5 cm, the second link to be 10 cm, and the third link to be 6 cm. The centers of mass are 293.6gr, 44.8gr, 120gr and 180.8gr for m_1 to m_4, correspondingly.

Table 1. Denavit-Hartenberg parameters.

Joint\ Parameters	a_i	d_i	α_i	θ_i
1	0	1.5cm	$\pi/2$	q_1
2	10.5cm	0	0	q_2
3	10cm	0	0	q_3
4	0	0	$\pi/2$	q_4
5	0	6cm	0	q_5

Where:

- a_i: distance from the intersection of x_i with z_{i-1} to o_i, measured by x_i. (cm).
- d_i: distance from o_{i-1} to the intersection of x_i with z_{i-1}, along z_{i-1}.(cm).
- α_i: angle between z_{i-1} and z_i, measured around x_i. (rad).
- θ_i: angle between x_{i-1} and x_i, measured around z_{i-1}. (rad).
- q_1, q_2, q_3, q_4, q_5: joint variables (rad).

Step 2: The coordinate transformation matrix A_i, is assembled for each link from the parameters obtained, through the following expressions Eq. (1), (2), (3), (4) and (5):

$$A_1 = \begin{bmatrix} \cos(q_1) & 0 & \sin(q_1) & 0 \\ \sin(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & 0.015 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$A_2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & 0.105\cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & 0.105\sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_3 = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & \cos(q_3)/10 \\ \sin(q_3) & \cos(q_3) & 0 & \sin(q_3)/10 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_4 = \begin{bmatrix} \cos(q_4) & 0 & \sin(q_4) & 0 \\ \sin(q_4) & 0 & -\cos(q_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_5 = \begin{bmatrix} \cos(q_5) & -\sin(q_5) & 0 & 0 \\ \sin(q_5) & \cos(q_5) & 0 & 0 \\ 0 & 0 & 1 & 0.06 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

These matrices are used to calculate the generalized coordinate transformation matrices T_i for each center of mass of the system [9] in Arnáez, which can be expressed as Eq. (6), (7), (8) and (9):

$$T_1 = A_1, \text{ considering } l_1/2 \quad (6)$$

$$T_2 = A_1 A_2, \text{ considering } l_2/2 \quad (7)$$

$$T_3 = A_1 A_2 A_3, \text{ considering } l_3/2 \quad (8)$$

$$T_4 = A_1 A_2 A_3 A_4 A_5, \text{ considering } l_4/2 \quad (9)$$

Step 3: The dynamic modeling of the system is developed. From the kinematic modeling, the elements T_{1i}, T_{2i} and T_{3i} are extracted from each matrix T_i for each center of mass, which correspond to the system's equations to obtain the Cartesian coordinates x, y, and z of the center of mass points. These calculations are used to determine the squared instantaneous velocity of each center of mass. Due to the structure of the system used, there are 4 centers of mass. Then, the total kinetic energy K is calculated by adding the kinetic energies of each mass in Eq. (10):

$$K_i = 0.5\, m_i\, v_i^2 \quad (10)$$

Where K_i: kinetic energy of the mass i (J), m_i the mass i (gr) and v_i the linear speed (m/s^2).

Step 4: We clear the inertia matrix M from K, in Eq.(11):

$$M = 2q^{-T} K \dot{q} \quad (11)$$

where:
- M: inertia matrix.
- q: joint position variable.
- $\dot{q}$: joint velocity variable.

Since the matrix M resulting from this system is extensive, the matrix for the values of the position of the simulated user's mouth is presented, where q_1 is π+0.05 rad, q_2 is 0 rad, q_3 is π/2 and q_4 is 2π/5, in Eq. (12). As can be seen, the joint variable q_5 is not controlled; therefore, during the simulation the matrix M is reduced to the first 4 columns and 4 rows.

$$M = \begin{bmatrix} 3.8073 & 0 & 0 & 0 & 0 \\ 0 & 7.0942 & 3.4784 & 0.8546 & 0 \\ 0 & 3.4784 & 3.3024 & 0.6786 & 0 \\ 0 & 0.8546 & 0.6786 & 0.1627 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12)$$

Step 5: The matrix C of centripetal and Coriolis forces is determined, using the Christoffel terms to the matrix M, in Eq. (13):

$$c_{kj} = \sum_{i=1}^{n} \frac{1}{2}\left[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right] \dot{q}_l \quad (13)$$

where:
- c_{kj}: centripetal and Coriolis force matrix elements C.
- m_{kj}, m_{ki}, m_{ij}: elements of the inertia matrix M.
- q_i, q_j, q_k: joint variable for the number of elements of M.
- $\dot{q}_l$: derivative from joint variable.
- n: number of joints.
- i: iteration for each link i in the formula.

Step 6: Calculate the total potential energy P in Eq.(14):

$$P_i = m_i\, g\, z_i \quad (14)$$

where:
- P_i: potential energy of mass i (J).
- m_i: mass i (kg).
- g: gravity (m/s^2).
- z_i: height (m).
- i: iteration for each link i in the formula.

Step 7: Calculate the gravity vector G in Eq. (15),

$$G = \begin{bmatrix} 0 \\ 53.2\sin(q_2+q_3+q_4)) + (236.2\cos(q_2+q_3)) + (332.9\cos(q_2) \\ 53.2094\sin(q_2+q_3+q_4)) + (236.2\cos(q_2+q_3) \\ 53.2094\sin(q_2+q_3+q_4) \\ 0 \end{bmatrix} \quad (15)$$

where:
- q_1, q_2, q_3, q_4, q_5: joint variable (rad).

Step 8: The dynamical model in Eq. (16) is obtained from matrices M, C and G:

$$\tau = M\ddot{q} + C\dot{q} + G \quad (16)$$

where:
- τ: control input vector, torques required at each joint.
- M: inertial matrix.
- C: matrix of centripetal and Coriolis force.
- G: gravity vector.
- $\dot{q}$: joint velocity variable.
- $\ddot{q}$: joint acceleration variable.

Step 9: The trajectories of the robot arm are defined, determining the angular values of the start and end points of each trajectory, in radians:

- Start position: $p_0 = (\pi/2, 0, \pi/2, \pi/2)$
- Plate position: $p_1 = (\pi/2, \pi/4, 3\pi/4, 3\pi/4)$
- Point close to person: $p_2 = (\pi, 0, \pi/2, \pi/2)$
- Mouth location simulation: $p_3 = (\pi + 0.05, 0, \pi/2, 2\pi/5)$
- Start/End position: $p_f = (\pi/2, 0.05, \pi/2, \pi/2)$

Each position is designed by determining the angles from the initial position of the robot arm. The trajectories of each joint are generated to carry out this route in the shortest possible time.

Step 10: Using "Feedback linearization", the computed torque PD control with gravity compensation is designed, corresponding to the control signal, which is expressed as follows Eq. (17):

$$\tau_c = \ddot{q_d} + 50\dot{e} + 50000e + G \quad (17)$$

where:
- τ_c: computed torque, control signal.
- e: joint position error.
- $\dot{e}$: joint velocity error.
- $\ddot{q_d}$: desired joint acceleration.
- G: gravity vector.

Step 11: The following linear expression Eq. (18) is obtained for the control of the robot arm,

$$\ddot{x} = M^{-1}(\ddot{q_d} + 50\dot{e} + 50000e + G) - C\dot{x} \quad (18)$$

where:
- $\ddot{x}$: system equation.
- $\dot{x}$: derivative of the system state vector.

2.3.2. Simulation results

Figure 6 shows the joint error identified by the PD controller simulation of computed torque with gravity compensation over time, where each color represents the angle position error of each joint. The largest error found in the simulation is 0.0015 degrees (2.6216e-05 rad), which is considered acceptable for this system.

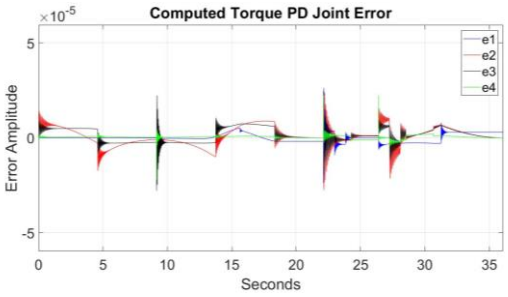

Figure 6. Gravity compensated computed torque PD controller joint error (blue: 1st joint, red: 2nd joint, black: 3rd joint, green: 4th joint).

Figure 7 shows the simulated trajectory followed by the robot arm and the desired path seen in 3D with Cartesian coordinates. Because of the used mathematical modelling, the forces and coordinate systems of the controlled system can be known, and it can be demonstrated that it is controllable with the resulting control signal. Likewise, the implementation of the system with servomotors enables both the tracking of trajectories defined with the coordinate system and the application of the control for the difference in joint positions.

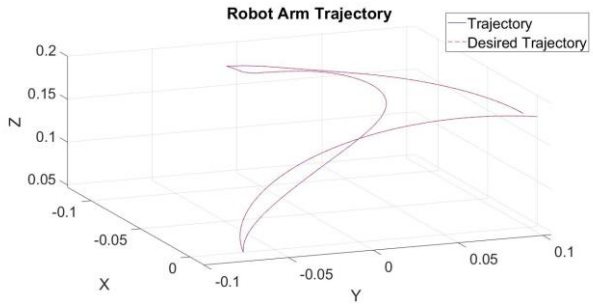

Figure 7. Trajectory followed by the robot arm seen in 3D (blue: actual trajectory, red: desired trajectory).

2.4. Electromechanical movement

An electronic system is implemented in a robotic arm that has already been designed, which has five degrees of freedom and is robust enough to be firm in the movement of the spoon. The system has a PCA9685 microcontroller to control the DS3218 servomotors of the robot arm with 5 degrees of freedom. Finally, the Raspberry Pi 4 and the microcontroller are connected to a 5V power supply. The robot arm chosen for this system has 5 degrees of freedom since according to [5] Park et al., robot arms with low DOF, provide limited meal assistance.

3. RESULTS

3.1. Validation of the Implemented System

The validation and performance evaluation of the implemented system is carried out through evaluation tests of the execution sequence of the complete system, in an uncontrolled environment. For these tests, the implemented electronic system, cooked oatmeal, cooked

rice, the system plate, and a spoon were used. A person without any disability participated in the study.

The system sequence was evaluated through events during its execution. In Figure 8, a diagram of the sequence executed by the system is shown. From the results, the percentage error of the sequence was calculated.

Sequence
Step 1. A spoon of food is ordered.

Step 2. Spoon is served.
Event A: Was the food picked up? The number of attempts is evaluated, less than 5 attempts is considered correct. Since a higher number indicates a lower probability that the spoon will be filled, and the time consumed during 5 attempts is reasonable.
Event B: Is the switch button pressed? Success is determined when the spoon is detected as served through the switch on the spoon adapter and goes to the transfer trajectory.

Step 3. Food is transferred.
Event C: Was the spoon cleaned properly? If it does not drip under the spoon during the transfer, it is correct.
Event D: Was it transferred without any inconvenience? If no food falls out of the spoon, it is OK.

Step 4. The spoon is brought closer to the user.
Event E: Was the food able to be picked up by the user? It is OK when it reaches the user position using the distance sensor at the base of the robot arm.

Step 5. The spoon is asked to be returned.

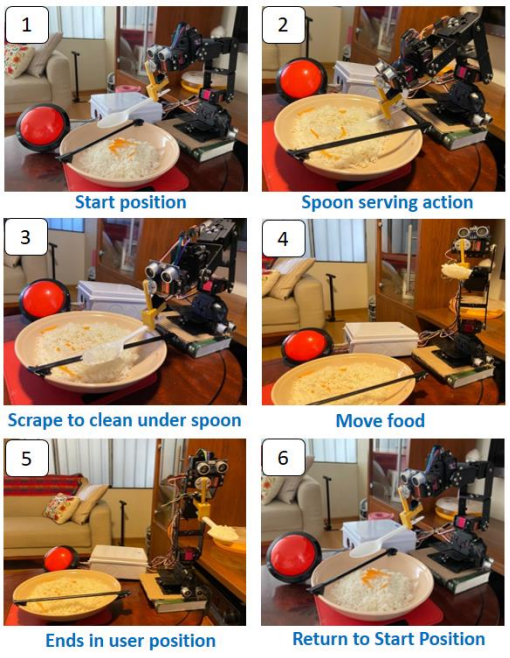

Figure 8. Sequence executed by the system.

The system performance measurement evaluates whether all events are met for each of the 70 tests. Then, the percentage of error is calculated, which is acceptable when it is less than 5%.

3.2. Graphic validation results

Figure 9 shows a summary of the results of the validations of the system with rice. A total of 35 tests are carried out with rice, since this is an approximate number of tablespoons that are required to finish the plate of food. The event of pressing the switch on the adapter when the spoon contains food failed validations with rice. However, the other events were completely successful.

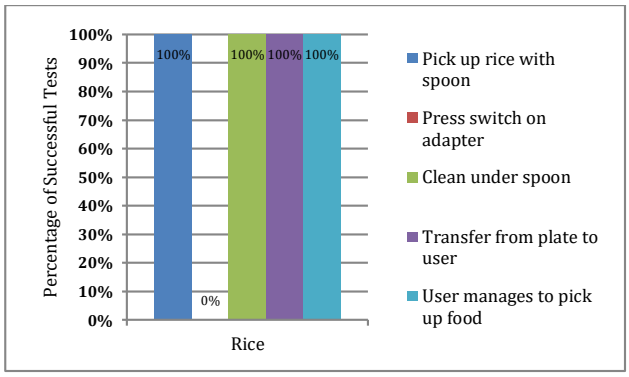

Figure 9. Summary of results of the validations of the system with rice.

In Figure 10, a summary with the data of the validations with oats is observed, in the same way 35 tests are carried out. In this case, all events were successful in validation.

3.3. Discussion of the results

Testing with rice shows that the spoon switch adapter is not sensitive enough to detect rice on the spoon. However, the events "pick up rice", "clean under spoon", "move towards user" and "pick up food by user" work correctly. On the other hand, when carrying out the tests with oatmeal, all events were successful. It should be noted that the event of cleaning under the spoon achieved a 94.29% of success. This is due to the varying weight of the oatmeal on the scoop, which may cause the cleaning bar to not be used properly.

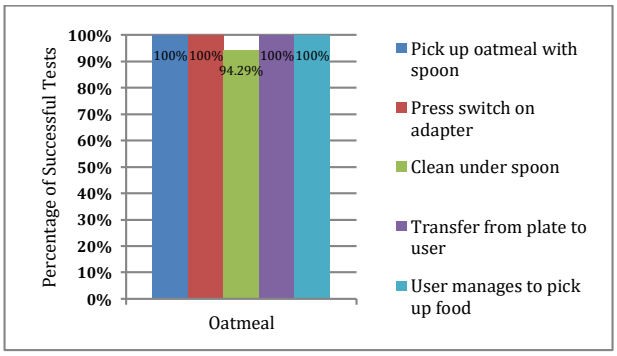

Figure 10. Summary of results of the validations of the system with oatmeal.

4. CONCLUSIONS

This work overcame human-robot interaction limitations by using actuators and sensors for the user's control over the system, as well as a robotic arm with a comfortable size for the user. Furthermore, implemented safety methods such as the emergency button and the proximity warning were successfully tested to prevent accidents during the robot arm movement. Additionally, it is proved that a robot arm of 5 DOF with rotational movements can successfully be programed to fulfill this type of system application's requirements, overcoming the limitations due to the lack of degrees of freedom.

Since the system doesn't require speech or face recognition, the computational load is lower than some previous works. Also, the system isn't designed for a specific 3D face model given it doesn't identify the user's facial characteristic, but his proximity to the system's robotic arm instead. The system achieves active feeding by approaching the user according to his proximity and not requiring any preview setup in the environment to deliver the food. Still, it is recommended that for future work the system is aware of the location of the user's mouth, instead of only the face.

Results of at least 90% success were achieved in validation events for both oatmeal and rice. However, certain differences were found for both cases. The weight of the collected food causes the trajectory to vary slightly when removing the excess food from under the spoon.

Future works should improve the identification of whether the spoon has picked up the food correctly. In this case, the contact switch used in the adapter is not sensitive enough to detect lighter foods such as rice. It is also important to note that this system works only with thick food, like oatmeal and rice. Solid food is not adequate to use since it requires a different procedure. Additionally, other measurements will be considered, such as food temperature or movement speed.

5. ACKNOWLEDGMENT

It is with deeply felt gratitude that we acknowledge Professor Erwin Dianderas and Giorgio L. Morales for their comprehensive and detailed peer reviews of this paper, as well as for pointing out the issues we overlooked when writing this document.

The authors also thank to the Dirección de Investigacion of Universidad Peruana de Ciencias Aplicadas for funding and logistical support with Code UPC-D-2022.

6. REFERENCES

[1] World Health Organization. (2011). **World Report on Disability.** Retrieved October 2, 2021 from https://www.who.int/disabilities/world_report/2011/summary_es.pdf

[2] Eurostat (2018) **Archive: Disability statistics - need for assistance.** Retrieved on February 7, 2022 from https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Disability_statistics_-_need_for_assistance&oldid=411113

[3] Candeias, A., Rhodes, T., Marques, M., Costeiral, J.P. & Veloso, M. (2019) **Vision Augmented Robot Feeding.** European Conference on Computer Vision 2018

[4] Liu, F., Yu, H., Wei, W. & Qin, C. (2020) **I-feed: A robotic platform of an assistive feeding robot for the disabled elderly population.** Technology and Health Care. 28(4)

[5] Park. D, Hoshi. Y. Mahajan. P, Kim, H. Erickson. Z, Rogers. W. & Kemp, C. (2020) **Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned.** Robotics and Autonomous Systems. 124 (6)

[6] Liu, F., Xu, P. & Yu, H. (2020) **Robot-assisted feeding: A technical application that combines learning from demonstration and visual interaction.** Technology and Health Care. 29(1)

[7] Haidar, G.A., Moussawi, H., Saad, G.A. & Chalhoub, A. (2019**) Robotic Feeder for Disabled People (RFDP).** 2019 Fifth International Conference on Advances in Biomedical Engineering (ICABME)

[8] Guo, M., Shi, P. & Yu, H. (2017) **Development a Feeding Assistive Robot for Eating Assist.** 2nd Asia-Pacific Conference on Intelligent Robot Systems

[9] Arnáez Braschi, E. (2015). **Enfoque práctico de la teoría de robots.** Lima: Universidad Peruana de Ciencias Aplicadas.

[10] Mišeikis, J., Caroni, P., Duchamp, P., Gasser, A., Marko, R., Mišeikienė, N., & Früh, H. (2020). **Lio-a personal robot assistant for human-robot interaction and care applications.** IEEE Robotics and Automation Letters, 5(4), 5339-5346.