Egg Detection and Manipulation System

Diego Zuñiga
Tecnológico de Monterrey
a01701853@tec.mx

Habib Wejebe
Tecnológico de Monterrey
a01708105@tec.mx

Antonio Leon
Tecnológico de Monterrey
a01708215@tec.mx

Alexander Carrizalez
Tecnológico de Monterrey
a01024901@tec.mx

***Abstract* — In the food industry, there are many important aspects to consider whenever manipulating food products. The process can't compromise the general quality of the product by causing damage to the product, such as deformations, bruises and bumps that could lead to the food being non-consumable. The sanitation needs to be a big priority as to avoid cross contamination with other products in the same system. An example of a hard product to work on is an egg, since the shell is fragile and can't be manipulated by a conventional gripper. The solution proposed involves the utilization of a custom gripper with soft actuators connected to a UR3e, which connects to a vision system in RoboDK that detects eggs and signals the UR3e to move over an egg and activate the pneumatic system that controls the actuators, which adapt to the eggs geometry with enough force to grab it and move it safely to another location. At the same time, a NodeRed instance runs in the background and keeps track of the UR3e status variables, such as the motherboard temperature, power state, operation mode and the state of the emergency stop.**

I. INTRODUCTION

The food industry is one of the biggest industries in modern society, so it's clear that if the quality and sanitation of the products are not constant or prioritized, it could potentially lead to bacteria and diseases being present in the food itself, risking the health of a large number of people [4]. With this information in mind, it's assumed that the egg production industry is not an exception. Since the eggshell is so fragile, any fracture or hole on it could lead to bacteria growing in the inside of the egg, infecting the yolk and causing diseases upon consumption.

The main focus of the system is to implement a vision system using a webcam that is able to detect an egg placed on a flat surface, along with a soft gripper system that can adapt to the egg's geometry to prevent ruptures on the shell. Said vision system includes an AI algorithm that can identify the egg's geometry and determine its orientation based on said geometry, which establishes the way the gripper needs to approach the egg in order to grab it without crushing the egg.

The soft actuators are designed so that, whenever air is pumped in them, a bending motion occurs and the actuators embrace the egg's geometry evenly, spreading the force along the area of contact with the eggshell. The actuators, which are made with a 2-part mix known as "Mold Star 15 Slow", are tough enough to resist deformation due to air pressure but also soft enough to adapt to an "irregular" geometry such as an egg.

Since the actuators need a constant air pressure to firmly grasp the egg and keep it from slipping, a pneumatic controller system had to be implemented. Using a solenoid electrovalve, the air pressure is regulated and maintained inside the actuators using a small air pump motor.

The different components and of the system will be explained thoroughly along this report, which has been divided into 7 sections to establish the differences between each part that contributes to the final result

II. MANUFACTURE AND DESIGN

As mentioned before, the gripper uses soft actuators to safely manipulate the eggs. Said actuators are hollow, as to allow a bending motion to happen in the actuator. In order to make said motion, the actuator has 5 air chambers on the top section, which inflate and cause the actuator to bend because the deformations caused by the air pressure collide with the other chambers, thus causing the bending motion.

The gripper only uses 2 identical actuators, so only one design was made. As shown in Fig. 1, the actuator design includes 5 hollow chambers for the bending motion, a small hose-like extrusion at the back of the actuator and a solid V-shaped figure at the front.

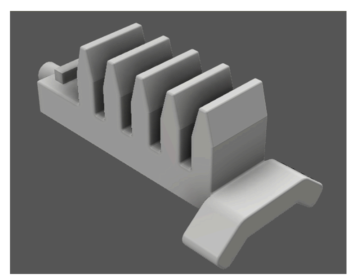

Fig. 1. 3D model of the actuator

The established number of 5 air chambers was ideal since the actuator is relatively small in size and did not need a large amount of pressure to bend and adapt to the geometry of the egg. Additionally, the front of the actuator was kept unmodified as it provided stability and grip when manipulating the egg, ensuring its safety while being transported.

In order to manufacture the actuators, a 3D model of the final actuator design had to be made in Fusion 360. After this, a mold was made based on the actuator design and divided into 2 sections: the upper section where the air chambers are, and the lower section. The molds were then 3D printed. The Mold Star 15 Slow mix that was mentioned before is poured into the molds in a casting process, let the mix sit for 4 hours so the mix could harden, and take both sections of the hardened actuator out of the molds. Then the same mix is used to join the two sections together in a single body. This process was made multiple times in order to have the second actuator along with spare ones in case of rupture or malfunction.

Regarding the gripper, it has a T-shape design to reduce the weight and allow for a fast installation on the cobot. As shown in Fig. 2, the lower section of the gripper has two rectangular holes, in which the actuators are fixed in place with a pair of screws on each side.

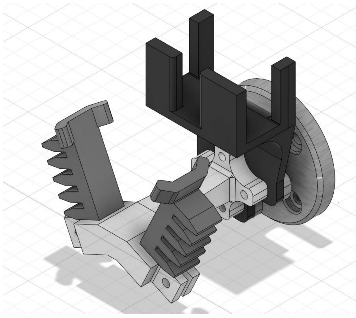

Fig. 2. 3D model of the gripper

III. CAMERA VISION SYSTEM

The camera vision system is tasked with identifying the eggs in a real world plane and assigning coordinates for the cobot to approach from above. To make this, a webcam needed to be calibrated, since its purpose is to determine its position in the simulation and the real world. The first part of the process was the camera-in-hand calibration, which is used to determine the position of the camera relative to the base of the cobot, as well as knowing the intrinsic parameters of the camera, such as the camera matrix and the distortion coefficients. Then, a program is executed that takes a photo and captures the position that the cobot has at that moment. It is then sent to the machine learning algorithm that identifies the eggs. Once the coordinates in pixels are defined, the coordinates of the centroid of each egg are saved in a “json” file.

Within the main program, the photo is analyzed and an OpenCV function is used and locates the 4 corners of the ArUco marker and the respective ID of the marker. A "solvepnp" function is used to determine the position of the reference frame of the ArUco and returns a translation and rotation vector in “rodrigues” format. Using this vector, a function is used that converts the vector into a matrix, which can be converted into a 4x4 homogeneous transformation matrix that represents the position of the ArUco marker relative to the camera. Then, the position of the cobot's flange relative to the base is obtained in a homogeneous transformation matrix. With the matrices defined, it is possible to obtain the pose of the ArUco relative to the base of the cobot. A function is created that allows obtaining the pose of multiple ArUco markers found in multiple photos, although in this case only a single marker is used. This function allows the algorithm to be more robust and allows the positioning to be more precise. Afterwards, the pixel coordinates of the eggs are inserted and a function is executed that allows finding the position vector of the eggs relative to the ArUco by using its translation and rotation vector. An identity matrix is added and the homogeneous transformation matrix of the egg relative to the ArUco is obtained, allowing for the transformation matrix of the egg relative to the base to be calculated. This entire process is carried out to assign a reference frame to the position of each egg that has been found and calculated in the program.

With the reference frame already found, RoboDK functions are used to move the cobot TCP over the egg. Because the reference frame was assigned to the egg with only one photo, there is a small position error (around 7 mm) which can affect the collection of the egg. For this, a proportional controller is applied for the angle and another one for the position of the centroid. Then, within the same python program, an MQTT instruction is sent to activate the actuators and close them, so that the egg can be grabbed and moved to the desired position.

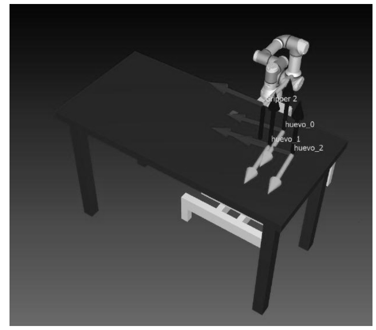

Fig. 3. Virtual workspace in RoboDK

IV. AI ALGORITHM AND TELEGRAM

A. Artificial Intelligence

In order to identify the objects in the workspace, object identification tools like “Roboflow”, which uses the YOLO V8 model to identify objects, were used. Yolo works using convolution and batches of images to train the network. Every image has its labels that allow the identification of pixels containing the desired objects in the image, and separate its trends using convolution to analyze the image. Once the bounding boxes of the objects present in the images are defined, it is easier to isolate the borders of the eggs using a binary image. This process returns a robust solution to the identification of the eggs since it doesn't depend on illumination because of the size of the batch of the model.

Using the bounding boxes of the objects, the image is isolated using the coordinates of the edges of the box as an input, then some image processing is done to smooth the image and reduce the possible noise before transforming the image to a binary matrix. The next step is to obtain the borders and moments of the image to detect the center of the egg and give this result as an output, adding the positions in a “json” file to use in the vision algorithm.

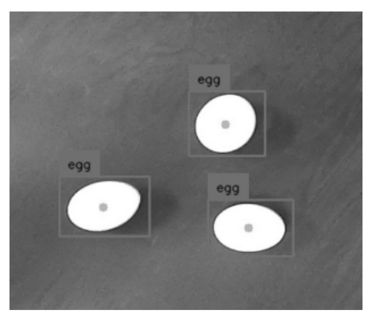

Fig. 4. Yolo object detection.

Remarkable features of this technology is the capability to separate objects if needed, therefore ignoring possible obstacles or in different situations identifying them, granting us an extra in case we have not desired items in the workspace, that way we'll find easier the way to maneuver in the table.

B. Telegram and NodeRed

The initial proposal for variable monitoring was to implement a dashboard in NodeRed for the operator, as well as a Telegram bot that could send requested data to a chat for the client to see. This proposal was modified because the NodeRed dashboard would show exactly the same data, so it was decided that the Telegram chat would be a "dashboard" for both the operator and the client.

To do this, an "exec" node in NodeRed runs a Python script that retrieves information directly from the Universal Robots servers and prints out the desired variables along with their values. In this case, 6 variables were considered relevant in the monitoring process: the cobot's timestamp, the state of the cobot's energy, the operation mode of the cobot, the masterboard's temperature, and both the states of the protective stop and the emergency stop. After initializing the flow, the desired variables and values are stored in the payload and can be used in a separate section of the flow, in this case a Telegram callback function.

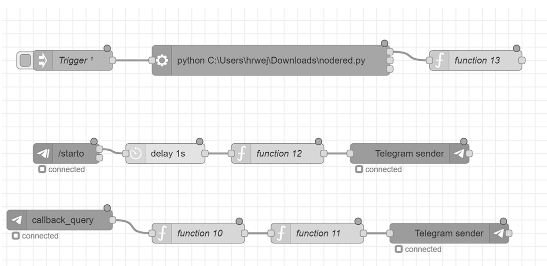

Fig. 5. NodeRed flow.

In the Telegram chat, the "/starto" command needs to be sent in order to initialize the bot, which then sends a greeting message and an options menu with the available actions the bot can perform. Whenever the "Huevos" button is pressed, the bot retrieves a counter value from an MQTT broker and displays the number of eggs that have been placed. Whenever the "Data" button is pressed, the bot retrieves and displays the variables stored from the python file.

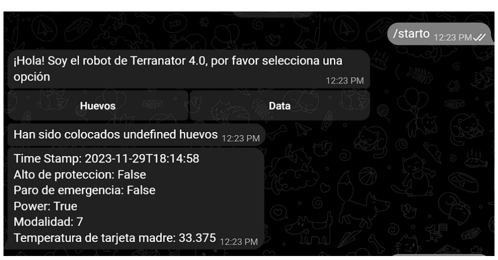

Fig. 6. Telegram Bot chat.

This function helps monitor the cobot's status without the need of being present in the cobot's workspace. This same function could be used, for example, to send a warning message to the operator whenever the cobot experiences a collision without the need to be present in the same room.

V. PNEUMATIC CONTROL

For the desired bending of the actuators that allow the desired manipulation of the eggs, an adequate pneumatic pressure is required to be constant at all times when the collection process is carried out. Any change in pressure can disrupt the clamping process and damage the material.

To provide constant pressure to the system, a pneumatic and electronic design was implemented that allows the pressures to be measured and controlled at all times.

The system contains a list of essential components for its execution:

- 12 Volt D/C Compressor
- 12 Volt Solenoid Electrovalve
- HL293D Module
- Pressure Sensor MPRLS
- L7805 Voltage Regulator
- ESP32 Module

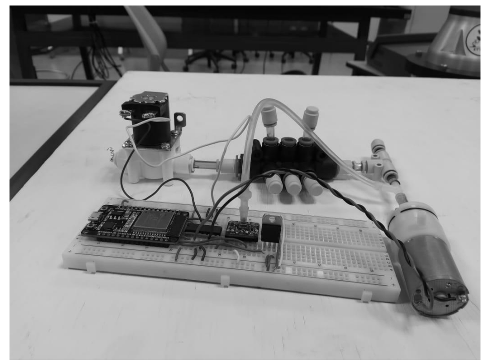

Fig. 7. Pneumatic control circuit.

Using MQTT communication, it reaches the ESP32 Module through a broker, which activates the compressor with a desired pressure, while the solenoid valve remains de-energized.

To provide the intended pressure standards, a PI (Proportional and Integrative) controller was implemented. The controller receives the pressure readings from the MPRLS Sensor in a measurement in Hectopascals (HPa). Then, the error between the desired pressure and the current pressure is determined, then multiplied by a proportional gain. This causes the control signal to approach the reference signal in the shortest possible time. For the integrative part, the previous error value from the last iteration is used by applying an integral gain and added along with the proportional part. The reason why the integrating part is used is to increase the control signal and reduce the error.

After providing a control signal, a PWM is implemented to send the required power to the compressor. In the project, it is observed that the PWM ranges had to be limited from 140 to 230 considering that the compressor had precision problems. With this configuration, the response time could be accelerated to reach the desired pressure. To interpret the controller pressure as a PWM value, a straight line equation was integrated that converts the PWM value that depends on the pressure variable as a function.

$$Y = m * x + b \tag{1}$$

$$PWM = \frac{Max.PWM - minimum\ PWM}{Maximum\ pressure - Minimum\ pressure} * Current\ pressure + minimum\ PWM \tag{2}$$

When obtaining the PWM to control the Power of the compressor, a device was needed that could send the signal and power the compressor and the solenoid valve, considering that the ESP32 could not output the necessary voltage. An H293D module was integrated, allowing the receiving of the PWM signals from the ESP32 and sending them to the compressor with a voltage between 9 and 12 volts.

To power the ESP32, the pressure sensor and the H293D module, an L7805 voltage regulator was used, which allows a voltage of 5 Volts to be sent to its output.

To release the system pressure when the actuator needs to release the egg at the end of the process, the solenoid electrovalve was used to be energized only when signaled via MQTT.

The pneumatic system also sends messages to the MQTT broker. The first message is transmitted to notify that the egg has already been collected. The interpretation used to send the message was to count 5 seconds where the controller signal was within a range of 50 hectopascals of difference from the desired pressure. If a pressure was maintained within this limited time, a message is sent to the cobot to continue with the process.

The second message was sent when the egg had to be released from the actuator. When the message to release the pressure arrived, the program activated the valve and turned off the compressor. After a 3 second delay, a message was sent to the broker to confirm that the collection process could be repeated.

A desired pressure of 30 KPa was required in the actuators to hold the eggs. The proportional gain to have an adequate error response was 0.7 and an integral gain 0.1.

The Serial monitor, as seen in Fig. 8, shows the relative percentage error of the pressure with the desired pressure, as well as the pressure in hectopascals and kilopascals. Finally, the PWM value that is being used in the compressor is shown as well.

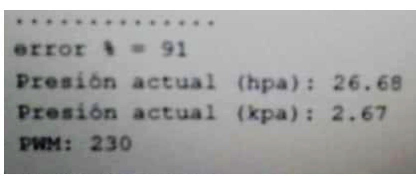

Fig. 8. Pressure and PWM values.

The results obtained were error-free communication when using the laboratory's local network to avoid errors with the MQTT protocol. The PI controller was extremely effective with errors averaging 6% at the desired pressure. During the experimentation and integration, there were no leaks in the system or damage with the eggs or the actuators.

As an improvement, the MQTT communication protocol could be changed considering that the response time with the cobot is slow with communication delays of 4 seconds on average, so it is considered optimal to have serial communication.

VI. PROJECT MANAGEMENT

During the whole project, there were different tasks that contributed to the final product. To keep track of how much time was being used up, 2 tools were used to regularly update and monitor the duration and importance of the tasks: a Gantt chart and a Planned/Earned Value Chart.

The Gantt chart helps assign time frames to work on each assignment, organizing the tasks and displaying visually how much time it requires to complete the tasks. Additionally, team members are assigned to different tasks, which can help with multitasking and making a faster progress on the project rather than just doing one task at a time.

The Planned/Earned Value chart keeps track of the importance of each task by estimating the amount of hours it requires to complete a single task. Depending on the number of people assigned, a task can take up many or little hours to complete. The whole process can be used to keep track of the general progress of the team and to verify that the real progress of the project is going according to the estimated progress of the project, referring to starting and finishing tasks on time.

All of the corresponding data can be found at the Google Drive folder in the “Annex” section

VII. RESULTS

Regarding object identification using AI, it is a robust solution because it makes the algorithm independent of the conditions that could be present in the work environment, directing the focus on the representation in the real world.

Of course, testing was due to determine the capabilities of the gripper, specifically the maximum payload the gripper could carry. To sort it out, different egg-shaped objects were used besides a real egg, like empty eggshells and Kinder chocolate eggs. The following table shows the different objects used in the experimentation and their respective weights.

Object	Weight
Regular White Egg	56.8g
Kinder Chocolate Egg	32.2g
Eggshell	9g

Fig. 9. Table of objects and weights

After the experimentation, it was concluded that the gripper could hold the 3 objects with no issues. Due to its relatively heavy weight, the regular white egg was the hardest object to manipulate, both because of its weight and the yolk moving around freely inside the egg. But putting that aside, the actuators firmly grasped the three objects and little to no movement was seen from the egg, proving that the design of the actuator fulfilled its purpose regarding stability and grip. Additional results regarding the gripper include the versatility of the handling. The design and location of the TCP allowed for the manipulation of the egg in different orientations even before the angle controller was activated, making it easier to move the egg around once the actuator was on.

One of the most polished aspects of the project was the calculation of reference frames using the ArUco markers which gave an accuracy of approximately 2 centimeters. The only issue came with the photos and the position of the eggs, which could compromise the algorithm if the photos aren’t clear enough

In the pneumatic control, results were promising due to the testing in the actuator alone. When integrating everything, it worked correctly most of the time, yet the time it took to communicate the devices (cobot and microcontroller) wouldn't allow faster even if the actuator didn't take much to fully expand.

VIII. DISCUSSION

The system has great potential in object position detection and recognition, considering that eggs have a complex shape and composition to find their centroid. The system can create frames of the position of the object very precisely and the movement control system of the robotic arm is suitable for delicate materials in order not to damage them. Its application can be done in other products that do not have a clear reference in their composition and that need to be collected in a specific space.

Something that could be improved is the process time. By having to wait for the video camera to activate to start the process, the code could be optimized to reduce the camera's response time. The control of movement towards the egg could be faster and the actuators could be redesigned to hold the eggs with more contact area considering that only empty shells could be used.

The proposal has to improve in many areas to compete in the food collection process and specifically eggs. Time can be reduced with more time spent debugging and optimizing the code. If the type of communication between the different subsystems is changed, collection would be improved. Also having more sophisticated processing and vision equipment could result in a competitive system in the industry.

Something that has to be fixed and that would give a notable improvement is to unify all the codes that were in place, since the system was divided into many codes that slowed down the process.

A larger robot could be used to have a range of movement and expand the work area.

The project has a lot to improve, but the foundations are in place to develop a very interesting system for the collection of complex and delicate objects that can help different industries and processes.

VIII. REFERENCES

[1] "¿Qué es MQTT? El protocolo de comunicación para IoT", solectroshop.com. https://solectroshop.com/es/blog/que-es-mqtt-el-protocolo-de-comunicacion-para-iot-n117 (accessed Nov. 28, 2023)

[2] "Robots Colaborativos en el Sector Alimentario", universal-robots.com. . https://www.universal-robots.com/es/blog/robots-colaborativos-sector-alimentario/ (accessed Nov. 28, 2023)

[3] HiSoUR Arte Cultura Historia, "Robótica suave", hisoure.com. https://www.hisour.com/es/soft-robotics-43130/ (accessed Nov. 28, 2023)

[4] the Food Tech, "La inocuidad de los alimentos, ¿Qué es y por qué es importante?", thefoodtech.com. https://thefoodtech.com/seguridad-alimentaria/la-inocuidad-de-los-alimentos-que-es-y-por-que-es-importante/

IX. ANNEX

[1] https://drive.google.com/drive/folders/1DOTdTrrrhWBA3TW2ROmb8jn4CYbqhrDi?usp=sharing