RoBorregos Milker Robot – Team Description Paper

Emilio Chávez, Clara Gutiérrez, Alejandro Garza, Néstor Maldonado, Javier Escamilla and Diego Cardozo

Instituto Tecnológico y de Estudios Superiores de Monterrey
Av. Eugenio Garza Sada 2501 Sur, Tecnológico, PC. 64849, Monterrey, N.L.

Email: contacto@roborregos.com

Abstract— The following article presents the methodology proposed by team RoBorregos to overcome the Latin American Robotics Competition Open Challenge 2017. It describes the integration of electronic components, mechanical designs, computer vision and navigation algorithms applied in the elaboration of the milker robot. A modular design is implemented for an easy and effective scalability of the robot.

Keywords— Autonomous robotics, Milking robot, LARC IEEE Open

I. Introduction

In order to complete the tasks listed on Latin American Robotics Competition (LARC) IEEE Open, and based on our previous work, a second version of an autonomous cow-milking robot is being developed. The robot must be able to locate the simulated-cows on the field, get empty terrines, milk a cow, and take the full terrine to the milk tank. This could be accomplished by a single robot or n-robots working simultaneously.

Throughout our society filled with consumerism, a high demand of alimentary products has required the development of automatic processes. Such methodology has been no exception for farm products, since the milking procedure consists in the translation of the cows from outdoors to the milking station in order to set them up to the machines. This operation consumes various resources such as time and labor since the cows must move by their own will. "In order for automatic milking systems to work, cows must go to the milking station voluntarily". (Henriksson, 2008). However the procedure can be ineffective in the case of a cow wandering out of range or one unable to walk, as discovered by Gitau (1996) which correlated pasturation and its quality with cow's health, and determined that not allowing them to pasture gives them 2.9 higher odds of lameness and a higher probability of swollen knees. Hence the implementation of a system, such as a robot, which is able to milk in the open field without the need of translating a cow, would assure an advantage in optimization of resources, aside from guaranteeing a dignified treatment for the livestock animals.

To the best of our knowledge, no autonomous mobile milking system has been developed. Nevertheless, different milking systems have been patented. In modern systems, a robotic arm attaches suction cylinders to each teat of the cow. Cows are then fed and a vacuum is induced in pulses in each cylinder (Torgerson et al, 2015).

Given that last year RoBorregos participated on LARC IEEE Open, improvements needed to be made. To accomplish this, weaknesses and areas of opportunity were defined before starting the robot's second version. These include: the robot was big and reducing its size would permit a better mobility on the field; the vision system needed to become more efficient to make the cow's search faster; and connectivity problems needed to be solved. In the following sections, the specifications of RoBorregos' 2017 robot will be explained. The field strategy is first described. Next, the robot's main mechanisms will be presented. Last, the robot's on board processing will be analyzed, including its hardware, applied software for control systems and computer vision.

II. Strategy

The strategy followed is based on the rules and specifications of this year's challenge. As the robot accomplishes the different tasks, such as reaching a cow, data will be stored to make upcoming decisions. The field wall will be used as reference for position.

As a start location, the robot will be placed between the gravel and the grass zone. This with the purpose of knowing the robot's route during the round, from starting to ending position. As the terrine zone is near, the robot will check both sides for it. Once the zone is found, the robot will proceed to grab a terrine. Terrines are detected with sensors and captured with the help of a claw. Location of the terrine zone will be saved for following terrine gathering.

After possessing a terrine, the closest and most accessible cow will be found in order to approximate to it and milk. In order for the robot to position itself under the selected cow and start the milking process, an optimal path will be calculated. A reverse-construction technique is used to decide the path to follow. This consists on analyzing the position and the orientation of the cow and comparing it to the robot's actual position and orientation. The robot will advance while adjusting itself as it gets closer to the cow. If the cow is touched, the control algorithm will re-adjust the robot's path.

Once the robot is correctly positioned to milk the cow, the process will begin, filling approximately three quarters of the terrine. Subsequently, it will move out of the cow and direct itself to the gate. In order to get there, the nearest section of the wall will be reached and followed towards the gate. The gate will then be crossed, for the vision system to look for the

1

tank. The direction and approximate distance of this object will be obtained, and the robot will move forward until certain distance is detected. As it gets closer, the robot will turn itself to get its claw facing towards the tank.

The claw will carefully pour the liquid into the tank, to proceed its way to the exchange zone to leave the terrine and restart the circuit. All detected cows will be milked until they have the minimum required milk for the milking system to work. One a cow is out of milk, the vision system will select another one based on its accessibility and distance. As for the terrines, in case there is only one left, the last cup will be reused until the round's time is up. In case of a lack of progress, the robot will start on an unknown position, therefore its first task would be to locate the field's gate to start the circuit. Figure 1 describes the robot's logic by steps.

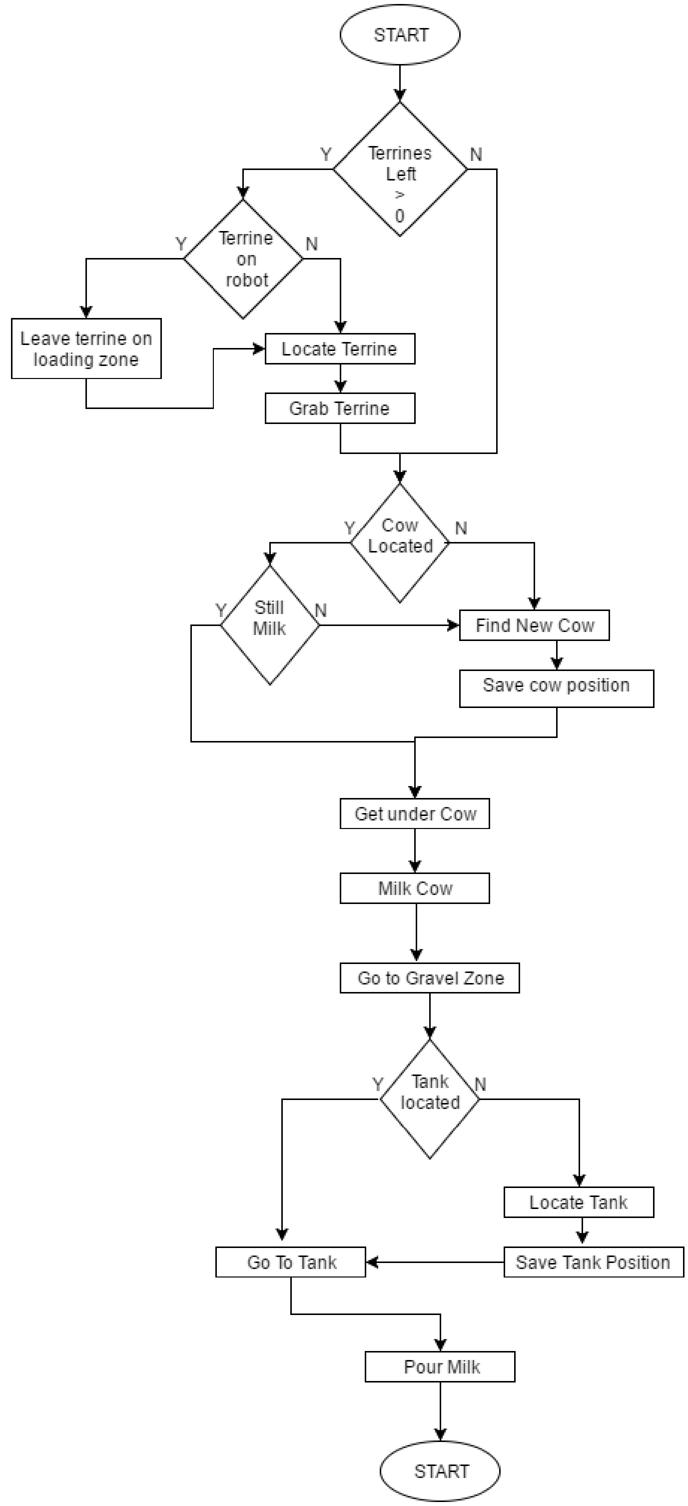

Figure 1. Strategy

III. Mechanics

Most of this robot is made out of 3D printing and Laser Cut Acrylic. For example, wheels were printed with PetG and NinjaFlex filaments to have a custom size and design for our robot. Milking and Terrine mechanisms work and move independently from each other and are described next.

A. *Milking*

It is definitely different milking a cow and milking a glove. Because of this, we analyzed and tried using different mechanism ideas. The most effective one, is to create a pressure drop, which would allow water to flow out of each tip. To generate this pressure on the glove, a two-phased mechanism was used. The first phase consists of a hermetic stamp, with the help of magnets. This phase prevents water flowing upwards into the glove. The second phase consists of a rectangular piece that constantly strikes the glove. Both movements are controlled by a 3D printed directional gear system, which can be seen in Figure 2.

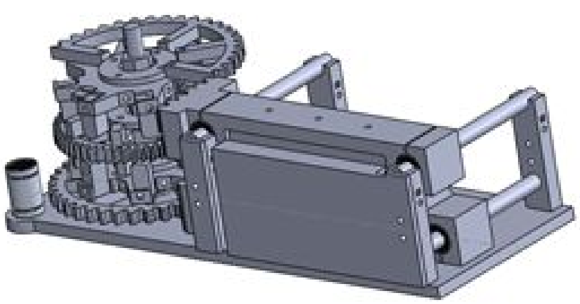

Figure 2. Milking mechanism

The directional gear system consists on three gears. The central gear, which has teeth in three directions (up, down and towards the sides) directs the movement in favour of one of the two other gears. The top and low gear are identical, but move towards different directions. When the central gear turns right, the top gear will move, allowing the hermetic stamp to open for the glove to come into the mechanism. The stamp will then close, and the central gear will start turning the other way around. When this happens, the second actuator will start moving front and back, pressing the globe several times to get water out of the glove. This sequence will be calibrated for better results.

B. *Terrines*

An optimal mechanism for terrine capture is a claw. The grasp of this mechanism can be modified to have enough strength to hold a cup firmly, with or without water. This way, terrines will not break nor will be dropped. To grab cups effectively, the claw will be supplied with distance sensors to locate the cup, stop in front of it, and confirm its capture. The mechanism has different movements: in and out of the robot, up and down to achieve different heights, and rotate to pour the liquid on the milk tank.

IV. On-board Processing

A. *Hardware and Software Specs*

In this section, the hardware of the milking robot are analyzed. The first item to mention is the camera for the vision system, the “Creative Senz3D” was selected because the 720p HD webcam and the capability of getting depth information. The second device is the Raspberry Pi 3 B which is a single-board computer with 1.2 GHz 64-bit quad-core ARMv8 CPU, 1GB RAM and Bluetooth 4.1, the Raspberry was selected because of it processing power and the capability of running the full range of ARM GNU/Linux distributions. Raspbian, OpenCV 2.4.13 and Python 2.7.13 is used with the main task to take images from the arena and locate the principal objectives as the cows, the tank and the terrines and exchange zone. Finally an Arduino Mega 2560 was chosen because of it 16 MHz crystal oscillator, 5V logic and capability of serial communication, the microcontroller will be handling the motors, sensors and servos. A BNO055 absolute orientation sensor was chosen because of it nine degrees of freedom that allow the sensor to give absolute orientation on euler vectors at a speed of 100 Hz; for distance the VL53L0X time of flight sensor was chosen, the sensor can handle distances from 50mm up to 2 meters depending on the ambient lighting, surface color and distance.

B. *Computer Vision*

This section will discuss the analysis of the computer vision developed using OpenCV and Python-Numpy. Since the environment is widely open to any conditions of light and background noise, the algorithm to detect the cow should be robust enough to to handle these uncertainty conditions. The computer vision analysis is fundamental for future decision making.

For now on, any analysis explained will be referred to a single frame; hence, the overall process is discrete. The first goal is to find the cow, the approach is based on the chess pattern that the objective has. The black rectangles that conform the cow will be found and processed to obtain relevant information such as the distance from the robot, the orientation and a set of coordinates to create a posterior trajectory to the objective.

The black rectangles will be associated with each other by building a structure that we will name *tissue.* In order to build the *tissue,* the image has to previously go through some steps, these are: binarization, erosion, image filtering, histogram equalization and contours search.

The first step is binarization. The input must be a gray scale image, taken by the robot frontal camera. If a pixel’s value is greater or less than the threshold, it is assigned either black or white respectively. For instance, the output is an image with only black or white pixels. It is thought that white is most present than black on a random image, so the threshold is iterated from 5% to 60 % of the total resolution, this iteration is needed due to the unknow light conditions this process is named *adaptive thresholding*.

The erosion of the image has the unique purpose of separating the edges of the squares in the cow so that they can be processed individually. Gaussian blurring is useful to eliminate black background noise. At this point the image is ready to obtain its contours using a routine provided by OpenCV that will return a numpy lists of all contours in the image (Mordvintsev, 2013).

Once the contours are found, the black rectangles will be obtained from them. The criteria to obtain black rectangles is the extent of the contour; this is, the number obtained by dividing the area of the object by the area of the minimum enclosing rectangle. If the quotient is close to 1, then, the contour must be a rectangle. In Figure 3, the green rectangle represents the minimum enclosing rectangle, and the red perimeter encloses the contour area.

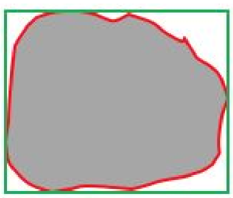

Figure 3. Extent of contours to obtain rectangles.

Hence:

$$if\ \left(\frac{contour\ area}{min.\ enclosing\ rectangle\ area} > \varepsilon\right) \Rightarrow contour\ is\ a\ rectangle$$

where ε is around 0.8

After the execution of this step, the information obtained results in a list of all the black squares detected in the frame, however since the competition is in an open zone, black squares not belonging to the cow but to the environment can be easily detected as well. This is considered as *noise*.

Since the list of black squares is not reliable to detect the cow, the implementation of another algorithm is needed. Such algorithm returns the *tissue*, which represents a list of black squares which belong to an individual cow. The algorithm iterates through every black square of the frame and searches for adjacent squares in the corners and sides of every square. By applying this in a recursive algorithm, all adjacent squares between each other are obtained and form the *tissue*, which provides the information needed to know the position and orientation of the cow towards the robot.

In Figure 4, the picture was taken in Mexico's national competition, and it has been processed by our computer vision algorithm. Blue squares represent black squares in the frame, green squares represent squares belonging to the tissue and red lines represent the side and upper limits of the detected cow.

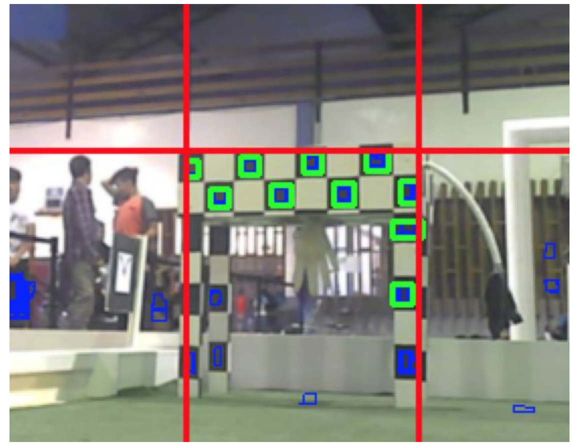

Figure 4. Execution of the code that detects the cow through computer vision

Two more approaches to detect the cow are being developed in order to qualify its reliability, which if after testing they are rated as trustworthy they will be implemented in the actual algorithm in order to obtain better results and eliminate noise as much as possible.

The first new approach is training a software through positive and negative samples with OpenCV, by introducing different samples of the cow (positive), and different samples of the possible environments the vision can be exposed (negative), a software can be teached to differentiate between a cow and not a cow, this generates a file that can be read by a classifier with the use of the OpenCV library. This would mean a more precise and faster way of detecting the cow.

The second still in development approach is to detect depth through epipolar geometry, by using two cameras in different sides, the two frames can be analysed and gain a tridimensional perception of the front side, such perception would be helpful to recognize near objects such as the cow. It would also be helpful to filter objects that are far away, such method would also reduce the noise in the environment.

C. Navigation System

The navigation system should be robust enough to be able to move throughout an unknown field that might present obstacles. Since the competition is *time-based,* it is intended to find the fastest paths to any objective. This will be achieved by the generation of an internal map over a graph data structure.

The graph should be able to expand and correct itself, it will be initialized empty and nodes will be added as the robot moves around the field. Objectives will be determined with the computer vision and the distance sensors, a combination of both will be used to reduce the error range once the objective is identified it will be marked in the graph.

The first objective the robot will find is the terrines zone, this zone will be the first node the graph. Once the terrines zone is mapped, the computer vision will run its process to find the cow. The next node in the graph will be placed referring to the distance and rotation of the cow, a proper route will then be calculated from the current position to the cow using A* algorithm (Muños, 2014). Similar process will be followed for rest of the objects in the field: the tank, the exchange zone and the gate.

In case of a lack of progress, the graph will be saved and the robot will use it again once it is located in the terrines zone. The previous graph will be softly compared to the new data from the sensors and the computer vision, if the data does not match, there will be a second round to verify to fix this mismatch.

D. Overall Operation

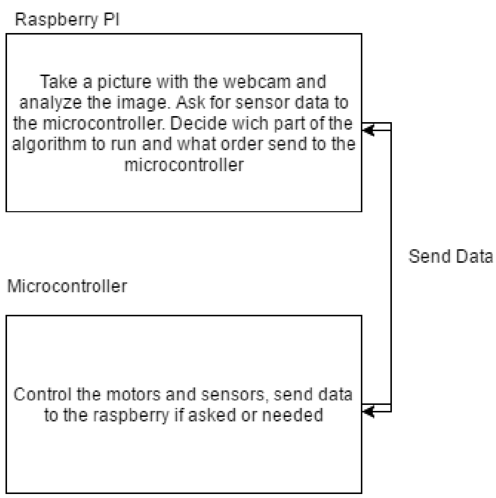

Figure 5. Diagram that explains the interaction between the two main cores of the robot

As we stated before, the first action taken by the robot is going to the terrines zone,. There, the grip will take the terrine and move it to a safe position inside the robot. Once the terrine is secured, the Raspberry Pi will take a photo and analyse it to find a cow in the given photo. If a cow is found, then the navigation system will generate the route based on the information given and this will be send to the microcontroller to decode the route. Otherwise, if the cow is not found; the robot will follow a recognition route and the process of identifying the cow will run again until the cow is found.

The computer vision will find a centered position under the cow and the robot will rely on his distance sensors to approximate itself and get in the milking proper position.

The milking system will start working after the glove's fingers are detected. It will rise and milk until there has been enough liquid extracted from the cow. After filling the terrine, the system will return to its original position and will head to the gate, where the vision system will start locating the tank.

Once the position of the tank is known, the robot will advance through the gravel without losing its direction with the help of a P correction in the motors. This is based on the angles detected by the robot's compass. When close to the tank, an actuator will lift the grip and pour the liquid inside. Afterwards, the robot may return to its original position and head to the exchange zone.

As the first complete circuit is made, the robot is capable of knowing several field locations, making it easier for the robot to find terrines, cows and the milk tank. The time required to get from one zone to another could be reduced on each new circuit the robot accomplishes. The vision system will be used even when certain locations are already known, since it is important to verify the saved data. It is also important to prevent crashing. Therefore, the robot has two distance sensors on each side and a rotating one at the center. In case a crash occurs, the robot can detect it with the use of its limit switches and its 9 axis of IMU freedom.

V. Conclusion

It is important to emphasize that the robot that will be used for this year's LARC IEEE Open competition is being developed with the best technology within reach. The strategy that will be carried out during the rounds is designed considering all possible cases that could occur while competing. All the work is being carried out with the intention of adding another robot to achieve better results in the overall operation. Having one robot at the gravel zone and another one at the grass zone would improve the time required to complete the challenge, but the two robots should be able to solve the challenge individually. Hence, all our development must be made modular, having in mind the possible improvements to be designed.

VI. Acknowledgment

All the team members are grateful with the Instituto Tecnológico y de Estudios Superiores de Monterrey for giving all their support to RoBorregos and giving us the opportunity to participate on this competition. Also the team express their sincere gratitude to our advisors Luis Ricardo Salgado, Juan M. Hinojosa and Vianney Lara for all their support on the team initiative and their reviews on this document. Ultimately, to team sponsors (Neitek, CSoftMty, Nuevo León Government, Yazaki, Transpais, Dextra, John Deere, Ventus, Lionchip and Akky) that believe in the development of technology and project as this kind.

References

[1] Henriksson, R. (2008). Pasture and Automatic Milking (Rep.). Retrieved June 10, 2016, from http://www.valacta.com/FR/Nos-publications/Documents/

Pasture and Automatic Milking.pdf
[2] Haan, M., Stuart, D., & Schewe, B. (n.d.). Challenges and benefits of adopting robotic milking on Michigan dairy farms. Michigan Dairy Review. Retrieved July 5, 2016, from https://msu.edu/~mdr/vol17no3/challenges.html
[3] Mordvintsev, A., & K., A. (n.d.). OpenCV-Python Tutorials. Retrieved July 6, 2016, from https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html?highlight=contoursl
[4] Torgerson, K.L., Hedlung, N., Stuessel, M.J. (2015). Patent No. 9,072,273. Holmen, WI: United States Patent
[5] Muñoz A.. (2014). Generación global de trayectorias para robots móviles, basada en curvas betaspline. 2017, de Dep. Ingeniería de Sistemas y Automática Escuela Técnica Superior de Ingeniería Universidad de Sevilla Sitio web: http://bibing.us.es/proyectos/abreproy/90085/fichero/TFG_Alejandro_Munoz_Cueva.pdf