Project, Modeling and Gait Optimization in the Pneumatic Gecko Robot

Matheus Rodrigues Goebel* Jorge Luiz Fontanella**
Marco Antonio Meggiolaro* Giovane Quadrelli***

**Mechanical Engineering Department, Pontifical Catholic University of Rio de Janeiro, RJ, Brazil, (e-mails: matheusrgoebel@gmail.com, meggi@puc-rio.br).*
***Mechanical Engineering Department, Pontifical Catholic University of Rio de Janeiro, RJ, Brazil, (e-mail: jorge.fontanella@gmail.com).*
**** Electrical Engineering Department, Catholic University of Petrópolis, RJ, Brazil, (e-mail: giovane.quadrelli@ucp.br).*

Abstract: The present work focuses on the development of a pneumatic gecko robot capable of moving on surfaces with different inclinations, using linear actuators. The main applications are inspections, cleaning and repairs to buildings, ship hulls and large solar panel structures. The work relies on the design of vacuum claws using mechanical and pneumatic concepts to escape the high expense of compressed air provided by the common vacuum components on the market. In addition, the present work applies the neural network algorithm called backpropagation with tests on the prototype to obtain a computational learning of the robot's operation, after observing the behavior of the actuators through simulations with a kinematic model.

Keywords: Pneumatic Gecko; Gecko Robot; Climbing Robot; Pneumatic Robot; Vacuum Claw.

1. INTRODUCTION

The use of robots for the execution of daily services has grown every day, especially in those that offer some degree of risk to the lives of humans, such as inspection and painting on buildings and ship hulls, and inspection and cleaning of large structures of solar panels.

This work aims at the development of a pneumatic robot with prismatic joints capable of moving on surfaces with different angles in relation to the ground, using vacuum claws. The robot's claws were designed in such a way that the vacuum is generated mechanically, in order to optimize the energy expenditure during its use.

In addition to the robot design, kinematics modeling that includes pneumatic and mechanical part was made. Thus, it was possible to know the operation of the robot in a simplified way through the simulations. Based on these simulations, data from random inputs were generated for testing with the built prototype.

The tests were extremely important to obtain a solid database, given that the kinematic simulations do not encompass all system variables. This database was used to train a neural network using supervised learning and backpropagation, in the MATLAB® (MATLAB 2021) software, in order to have a computational learning of the projected robot's movement mechanics. This way, even if the hardware presented problems, the computer would learn to simulate it in an intelligent way to give a good answer about its functioning.

2. RELATED WORKS

There are some robots in the literature that share the idea of displacement on angled surfaces, however, that use different technologies for this.

Santos et al. (2008) developed a robot inspired by geckos that uses microstructured polymer feet to obtain adhesion with the desired surface.

Provancher et al. (2011) uses a system with a pendulum and hooks that form the robot's claws to be able to climb on very rugged surfaces, such as carpets.

Chen (2015) used electroadhesion technology for its robot to be able to climb on smooth surfaces.

Mir-Nasiri et al. (2018) projected a robot for cleaning glass windows using suction cups and vacuum generator valves that are activated through pneumatics.

Hossain et al. (2018) designed a robot with wheels capable of moving horizontally and vertically using propellers in its center, which when close to the surface on which the adhesion is desired, generate an area of low pressure. The force caused by the pressure difference of this area and the atmospheric pressure of its exterior, makes the robot able to move on inclined surfaces.

Shi et al. (2020) had designed a robot, where its suction chamber works with two fluids, water and air. The robot's claws use the swirling water to help maintain the suction generated by a vacuum pump, thus achieving grip on surfaces with different textures.

There are also works in the literature in the area of gait optimization, not necessarily used in robots that fit into the same category of this work.

Masuri et al. (2020) uses a generic algorithm for dynamic self-learning of the walking movement of a quadruped robot. The equations to simulate this movement are complicated, so the prototype was used to collect experimental data in order to optimize the robot's gait.

Alharbi et al. (2021) used the Levenberg-Marquardt method to work with predictions of human walking gait angles, considering fixed speeds. The study data shows a new approach to be used in projects of prosthetics and biomechanics rehabilitation.

3. MECHANICAL PROJECT

The first step of the project was the mechanical design, which aimed to develop a robot at the lowest possible cost, both in terms of energy expenditure, as in the manufacturing of parts.

The suction cups and vacuum generator valves represent a considerable expenditure of compressed air, since they use the Venturi principle (despite being the two most common elements on the market when it comes to vacuum claws). So, there was a need to design a claw without having these commercial elements. Therefore, the idea of generating the vacuum mechanically was put into practice.

Figure 1 is used to show the cross section of the claw designed in the SolidWorks® (SolidWorks 2020) CAD software. It can be seen that the claw has two pistons, i.e. two built-in actuators. The lower actuator is responsible for generating the vacuum mechanically when activated, since when the compressed air enters its chamber, the rubber is stretched and the vacuum is generated between its surface and the surface on which it is desired to adhere.

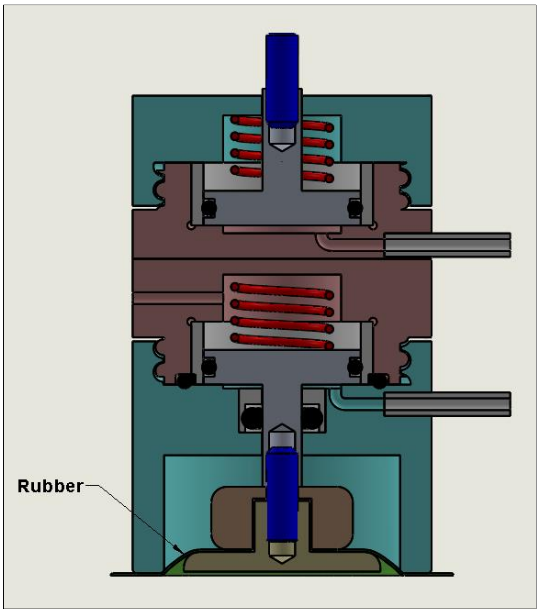

Figure 1 Designed claw: Sectional view.

The advantage of this type of suction cup over the commercial ones is that it consumes only the volume of air necessary for the actuator advance to occur. In other words, there is a fixed energy expenditure, no matter how long one wants to keep the piston actuated. Commercial components, on the other hand, need a constant flow while the vacuum is generated, thus, the air consumption is proportional to the working time.

The upper actuator serves to lift the entire structure of the robot while it moves, to avoid friction between the suction cups not used in a given step and the surface on which to adhere.

Figure 2 is used to show the assembly of the robot, indicating each actuator. Actuators 1 and 2 are the main cylinders, responsible for the movement of the gecko - these are commercial actuators from Festo® (Festo Corporate, Esslingen, Germany). Actuators 3, 4, 5, and 6 are the suction cups shown in Figure 1. Finally, actuators 7, 8, 9, and 10 are the lifters shown in Figure 1 by the upper piston.

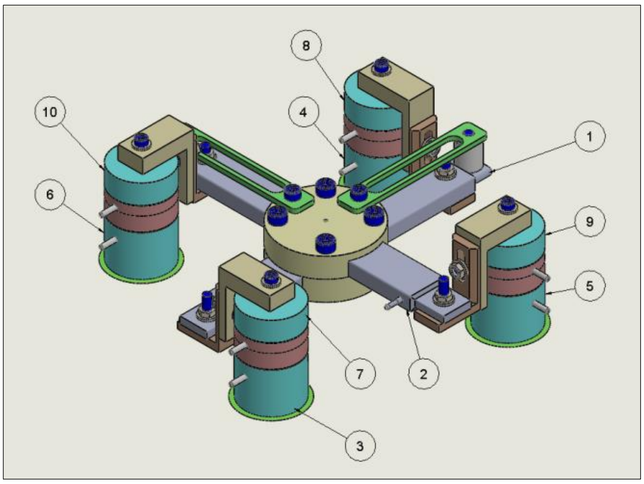

Figure 2 Complete assembly of the robot body.

4. KINEMATIC MODELING

During the execution of the project, the knowledge of the behavior of the actuators is needed, according to parameters that would later be used in the tests. For that purpose, the kinematic modeling of the system was developed, based on Kluever (2015).

For the pneumatic subsystem, the mass flow equations are first defined, which can be classified as having limited or unlimited flow. The flow is said to be limited when it reaches sonic conditions in a throat, a valve orifice, for example:

$$w = C_d A_0 P_1 \sqrt{\frac{2\gamma}{(\gamma-1)RT}\left[\left(\frac{P_2}{P_1}\right)^{\frac{2}{\gamma}} - \left(\frac{P_2}{P_1}\right)^{\frac{\gamma+1}{\gamma}}\right]} \tag{1}$$

$$\text{if } \frac{P_2}{P_1} > C_r$$

where γ is the ratio of the specific heats, C_d is the discharge coefficient, A_0 is the orifice area, P_1 and P_2 represent upstream and downstream pressure, respectively, R is the gas constant, T is the fluid temperature, and C_r is the critical ratio:

$$C_r = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{2}$$

ISSN: 2175-8905

DOI: 10.20906/sbai.v1i1.2717

When the flow is unlimited, the mass flow equation is given by:

$$w = C_d A_0 P_1 \sqrt{\frac{\gamma}{RT} C_r^{\frac{\gamma+1}{\gamma}}} \quad \text{if} \quad \frac{P_2}{P_1} \leq C_r \tag{3}$$

These equations are directly applied to the fundamental model of pneumatic systems:

$$\dot{P} = \frac{nRT}{V}\left(w - \frac{P}{RT}\dot{V}\right) \tag{4}$$

where P is the pressure in the actuator chamber, V is its volume, n is the polytropic exponent and R is the gas constant. To model the mechanical part of the system, the force equilibrium in the desired component is used.

Figure 3 represents a generic pneumatic piston with simple action, i.e. with spring return, with indications of the forces acting on it. In the figure, P is the pressure inside the cylinder chamber, P_{atm} is the atmospheric pressure, k is the spring constant, b is the viscous friction coefficient, x is the displacement of the piston, and F is a generic variable that represents some external force that acts against the piston rod. Depending on how the actuator is used, this force can change direction.

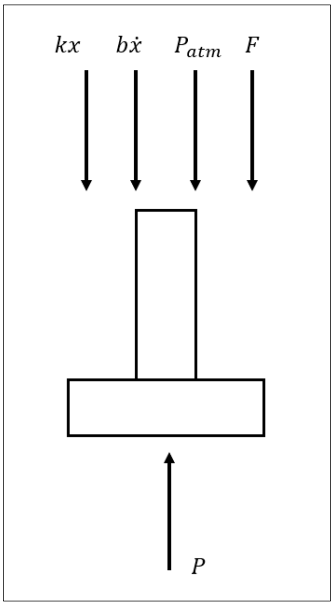

Figure 3 Free body diagram of a single acting pneumatic piston.

Knowing that the force coming from the compressed air, which acts on the piston, is the product of the air pressure and the piston area, the equation that mechanically represents this system is:

$$\ddot{x} = \frac{PA}{m} - \frac{P_{atm}A}{m} - \frac{kx}{m} - \frac{b\dot{x}}{m} - \frac{F}{m} \tag{5}$$

where A is the piston area and m is its mass.

With the mechanical design completed, the initial idea of moving a complete cycle on an axis of the robot is given by the path-step diagram shown in Figure 4. This diagram, also referred to as the robot *gait*, is a schematic way of representing the actuation order of each actuator, and also show when each actuator is forward or backward.

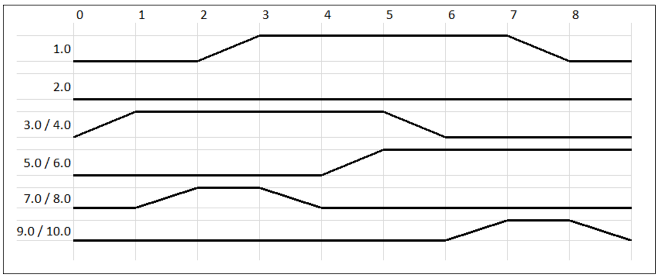

Figure 4 Path-step diagram of a movement cycle.

Based on this diagram, a simulation was generated with a lag of 0.4 seconds between each actuation signal of the valves that control the actuators. Figure 5 shows the graph of the generated signals, where the upper part shows the signals for the main cylinders, the middle part for the suction cups, and the lower part shows the signals for the lifters. Figures 6-8 show the displacements of the main actuators, the suction cups, and the lifters, respectively.

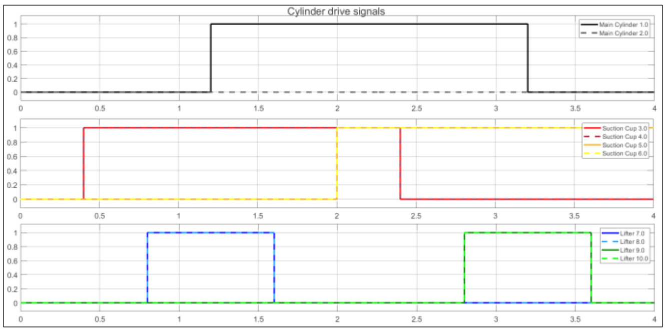

Figure 5 Graph with the trigger signals of the valves.

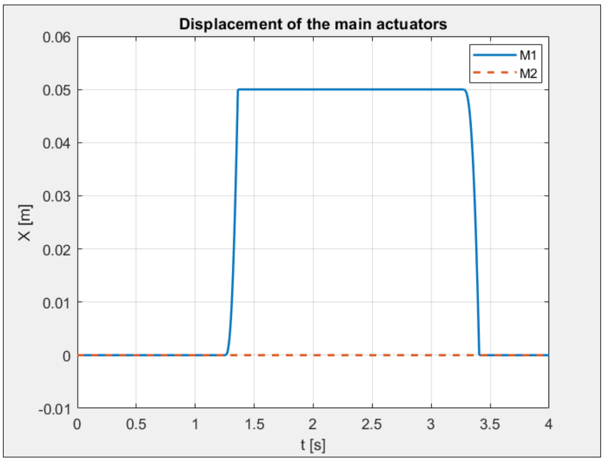

Figure 6 Graph of displacements of main cylinders 1 and 2.

This kinematic simulation allows the knowledge of the behavior of the actuators in a simplified way, so that it can be used as a base parameter for the neural networks. Thus, in the next sections, the computational learning approach to be applied to the robot is presented, for a generalized understanding of how the robot works.

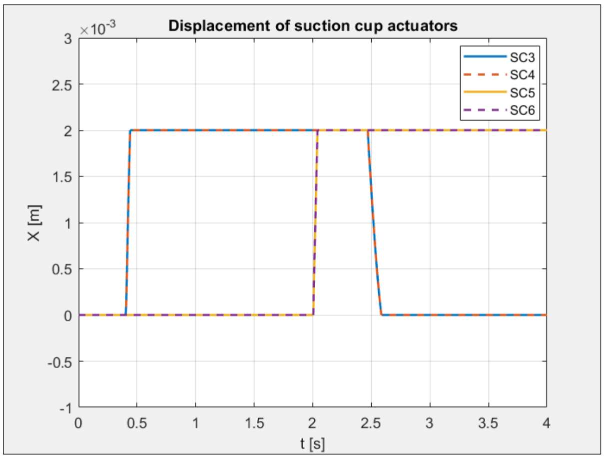

Figure 7 Graph of displacement of suction cups 3, 4, 5, and 6.

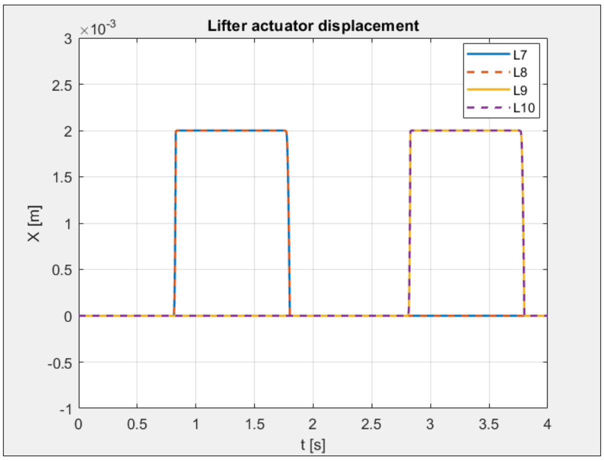

Figure 8 Graph of the displacements of lifters 7, 8, 9, and 10.

5. NEURAL NETWORKS

According to Lin (1996), neural networks are signal processing systems that have the ability to learn and generalize from the offered database, with several ways to be implemented. Neural networks are inspired by the neuron networks of human brains, so they are also called artificial neural networks.

A basic artificial neuron is formed around the signal processing part, which is where the activation function is located. It receives the values of the inputs that are connected to weights before processing it, generating an output that is passed onto another neuron. Each layer of a neural network can be composed of several neurons.

A neural network can have varying amounts and layer storage structures. These variations classify neural networks, and each variation has a different application, depending on the problem.

In this work, supervised learning was used with the backpropagation algorithm. Simply put, this algorithm works with networks that use multiple layers, in which the weights propagate forward according to the input data and activation functions. Arriving at the output layer, it compares the result with the desired output value and finds the error value; if it is not satisfactory, this value is propagated back to the previous layers, so that in this way they update their weights. This cycle continues until it finds a satisfactory error or stops at a point where training causes the network to begin to lose the ability to generalize.

Figure 9 is used to show schematically the backpropagation algorithm, where x_m are the inputs, y_n are the calculated outputs, d_n are the desired outputs and e_n are the calculated errors.

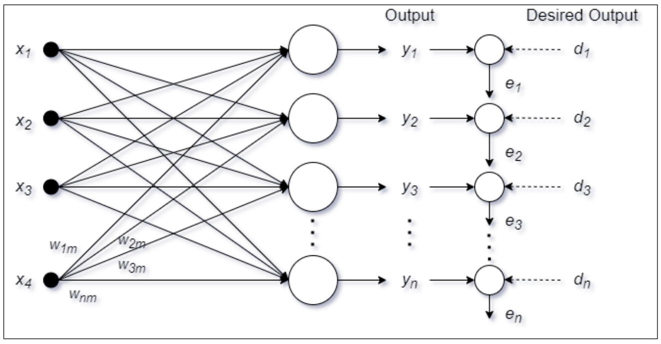

Figure 9 Backpropagation algorithm.

The algorithm works in a few steps, as follows:

Step 0 is the initialization, where small and random initial values are defined for the weights, in addition to defining the maximum error.

In step 1, the matrix with the input values is applied to the first layer of the network.

Step 2, which is called a forward step, is responsible for propagating the signal forward according to the activation functions.

The error measurement takes place in step 3, where the difference between the desired output and the one obtained by the neural network is calculated.

Step 4 is called backpropagation, where the gradient of the loss function is calculated and the weights of the previous layers are updated.

In step 5, which is the final step, it is checked whether the error is acceptable. If the error obtained is less than the maximum allowed error, the problem is over, otherwise the algorithm goes back to step 1 and a new epoch begins.

Among the various methods of applying backpropagation algorithms in MATLAB®, the Levenberg-Marquardt method was chosen, which is generally highly recommended as the first supervised learning algorithm option, in addition to being one of the fastest algorithms in the MATLAB® toolbox, according to The MathWorks (2021)

6. APPLICATION OF THE NEURAL NETWORK IN THE PROTOTYPE

The idea of the project is to make the neural network learn to fully simulate the dynamics of the robot, in order to adjust its control later, since the modeling does not cover all aspects that may influence the robot movement. So, even if there are mechanical problems in the prototype and some other variants or disturbances not foreseen in the modeling, the computer will

ISSN: 2175-8905

DOI: 10.20906/sbai.v1i1.2717

learn to simulate the actual system and optimize its gait. This is interesting, because depending on the application, different activation times may be desired for each actuator, and there may even be the possibility of two or more actuators being operated at the same time or with very little lag between them.

Most parts of the prototype, shown in Figure 10, were manufactured using the 3D printing method with a synthetic plastic polymer, in order to reduce cost and weight. Only the parts that needed better resistance, stiffness and finishing, due to the movement of the pistons, were made of aluminum.

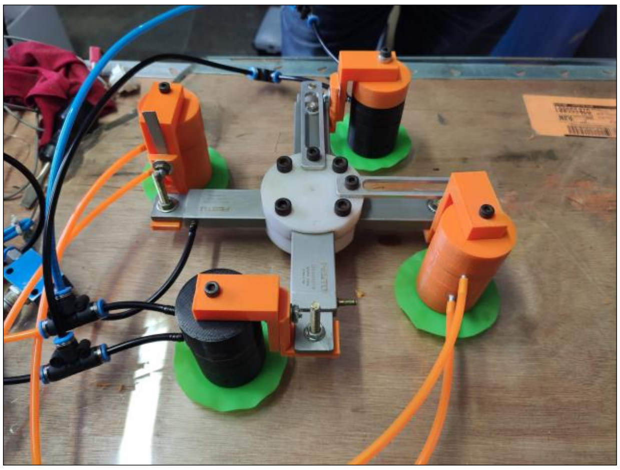

Figure 10 Prototype being prepared for testing.

To carry out the test, the prototype was assembled using commercial solenoid valves from Festo®, which received a signal from a programmable logic controller.

As the backpropagation algorithm requires input and output values, 250 sets of random values were generated for the input times, i.e. the times when each valve should be activated. These times were generated within the limits observed in the kinematic simulation, using the same gait shown in Figure 4, but with the activation times between each actuator varying between -0.1 and 0.5 seconds. That is, actuators 7 and 8 could have its advances activated a maximum of 0.5 seconds after, or 0.1 seconds before activating actuators 3 and 4, for example.

Each of the sets of inputs was implemented separately in the programmable logic controller, therefore, the controller had the exact times in which it would have to activate each valve to advance or retract a given actuator. With the programming complete, a trigger was needed to start the movement, so a retentive button was used.

The neural network used was a recurrent two-layer hidden network, with 5 neurons in each of the layers, using the unipolar sigmoid activation function.

The output data were obtained under human supervision, that is, with observation during the experiment. If the movement cycle was successful, it was noted that the output for this particular set of inputs was 1, in case there were any problems that caused movement failure, for example, the robot comes off the surface or slips due to poor suction cup attachment, the output for this set of inputs was 0.

The collected data were applied using the MATLAB® software with its training tools to execute the backpropagation algorithm.

7. RESULTS

With the application of the data in the backpropagation algorithm, so that the neural network could learn how the robot works, some graphs were generated for a better understanding of the results.

Figure 11 shows the graphs of the mean squared errors, over the periods during the training of the neural network. The blue line represents the training, the red line are tests that the MATLAB® tool itself performs by default, the green line represents the validation of the network, and the dashed line shows the best result obtained.

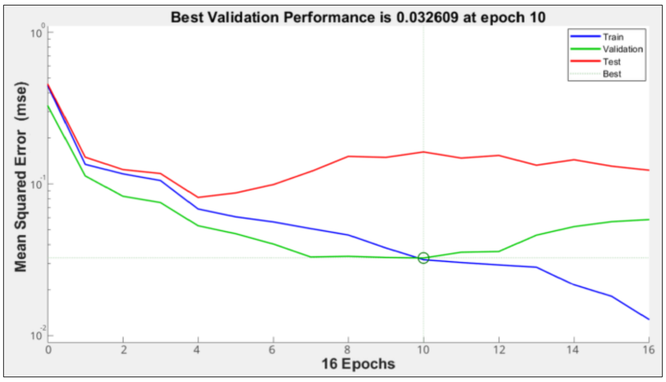

Figure 11 Graph of training performance of the neural network.

Although there are times when weights have improved training and have given a better result in the testing phase of the software itself, it is important to stop at the point where the validation had the smallest error, as it is responsible for telling if the neural network has a better or worse ability to generalize. In algorithm configuration, it was decided that if in a range of 6 epochs was not found a new minimum value for the validation, the algorithm ends. As a minimum value was found in epoch 10, the algorithm stopped running after epoch 16.

In addition to the 250 sets of times generated at random for the initial tests, a further 25 were generated for the execution of tests for the purpose of validation and comparison.

Figure 12 shows the output results obtained through the tests with the prototype (blue line), and the results calculated by the neural network (red line) for each of the 25 new sets of generated inputs (*n*).

Figure 13 shows the regression graph obtained with the results of the final tests compared to the values calculated by the neural network.

Figure 14 is used to show the number of correct answers and errors of the neural network considering a specific value of the maximum allowed calculation error. It is observed that the network correctly calculated twenty three outputs, which corresponds to 92 percent correctness, based on a maximum calculation error of 0.2 for the outputs, since the outputs have binary values.

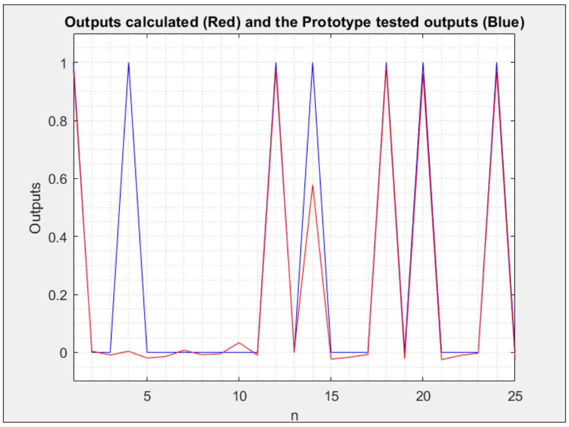

Figure 12 Comparison between tested and calculated values by the neural network.

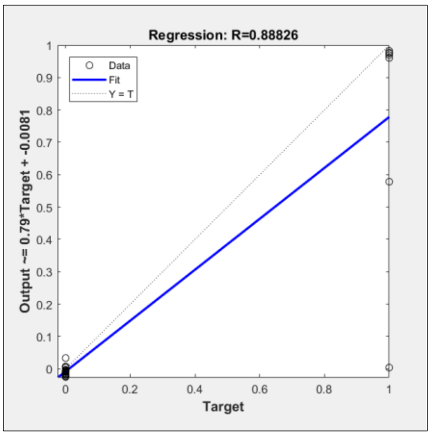

Figure 13 Final tests regression graph.

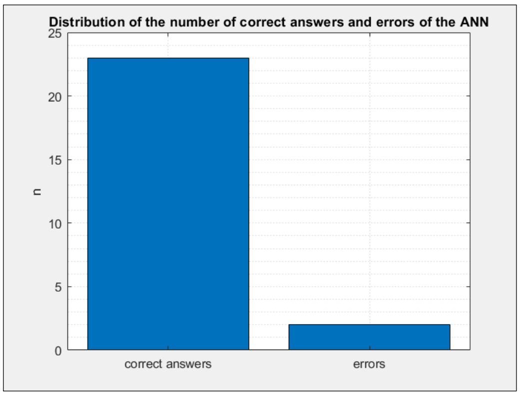

Figure 14 Graph of the neural network's successes and errors.

The mechanical part of the project worked well in tests, although its components were manufactured on a 3D printer, it presented stability and resistance to complete the desired movements. The main component developed in this work, the claw, proved to be effective in the tests, and can be a good alternative for conventional vacuum units and suction cups. Its manufacture is simple and its energy expenditure is limited to the volume provided by its base area and displacement of the piston.

8. CONCLUSIONS

In this work, the analysis of a pneumatic gecko robot was presented, driven by linear actuators and capable of moving on surfaces with different inclinations. Due to its 10 actuators, there are many possibilities for the order of activations and activation times. Such robot gaits were controlled using neural networks. Two hundred and fifty tests were performed to train the neural network. During the tests, only randomly generated values were used, within pre-established limits, so that there was less contamination of the database. Thus, the network's natural learning behavior could be observed in a more natural way. To improve the robot gait in future work, it is necessary to adopt a larger database or to have some sets of inputs and outputs applied manually for training the network, for a greater disversity of situations in the training.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

REFERENCES

Alharbi, Abdullah & Equbal, Kamran & Ahmad, Sultan & Rahman, Haseeb & Alyami, Hashem. (2021). Human Gait Analysis and Prediction Using the Levenberg-Marquardt Method. *Journal of Healthcare Engineering*. 2021. 1-11.

Chen, Rui. (2015). A Gecko-Inspired Electroadhesive Wall-Climbing Robot. *Potentials*, *IEEE*, 34, 15-19.

Festo. Esslingen, Germany: Festo Corporate.

Hossain, Riad & Chisty, Nafiz. (2018). Design and Implementation of a Wall Climbing Robot. *International Journal of Computer Applications*, 179, 1-5.

Kluever, C. (2015). *Dynamic Systems: Modeling, Simulation, and Control*, Wiley.

Lin, C-T & Lee, CSG (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall.

Masuri, Ariel & Medina, Oded & Hacohen, Shlomi & Shvalb, Nir. (2020). Gait and Trajectory Optimization by Self-Learning for Quadrupedal Robots with an Active Back Joint. *Journal of Robotics*. 2020. 1-7.

MATLAB version 9.10.0 (R2021a). (2021). Natick, Massachusetts: The MathWorks Inc.

Mir-Nasiri, Nazim & Siswoyo Jo, Hudyjaya & Ali, Md. (2018). Portable Autonomous Window Cleaning Robot. *Procedia Computer Science*, 133, 197-204.

Provancher, William & Jensen-Segal, Samuel & Fehlberg, Mark (2011). ROCR: An Energy-Efficient Dynamic Wall-Climbing Robot. Mechatronics, *IEEE/ASME Transactions on Mechatronics*, 16, 897-906.

Santos, Daniel & Heyneman, Barrett & Kim, Sangbae & Esparza, Noe & Cutkosky, Mark. (2018). Gecko-inspired climbing behaviors on vertical and overhanging surfaces. *Proceedings – IEEE International Conference on Robotics and Automation*, 1125-1131.

Shi, Kaige & Li, Xin (2020). Vacuum suction unit based on the zero pressure difference method. *Physics of Fluids*, 32. DOI: 10.1063/1.5129958.

SolidWorks version 2021. (2020). Waltham, Massachusetts: Dassault Systèmes SolidWorks Corporation.

The MathWorks, I. (2021). *Deep Learning Toolbox*. Natick, Massachusetts: The MathWorks Inc. Available at: https://www.mathworks.com/help/deeplearning/.