Optimizing a One DOF Robot Without a Mathematical Model Using a Genetic Algorithm

Daniel Wilczak[1], Dylan Ballback[2], Rachel Secord[3], Dr. Matthew Verleger [4]

Abstract— **The purpose of the present study was to design a flight control system with no pre-determined mathematical model, but instead using a genetic algorithm to maintain the optimal altitude. The study is done through a quantitative empirical research method. In the process of conducting the research, we found that programming a genetic algorithm was cumbersome for novice users to implement. Due to this, we created and released an open source Python package called EasyGA. An initial population of 10 chromosomes and 5 generations were used during the trial. The throttle value of the device had an associated gene value of 1 second. When the trial of five generations was completed, the total increase percentage was 171 percent. Preliminary results showed that optimizing a one DOF device, in real-time, is possible without using a pre-determined mathematical model.**

I. INTRODUCTION

Genetic algorithms have become widely used to solve large search space problems. DOF problems are typically not solved using search space algorithms but rather a pre-determined mathematical model. The most common pre-determined mathematical model typically uses a PID to stabilize the devices state. Mathematical models work well if the device that is being optimized is used in the way that the creator of the mathematical model intended. No modifications or adjustments can be made to the device without it requiring the user to adjust the mathematical model manually. Genetic algorithms while training in real time builds a memory similar to a mathematical model that continuously optimizes itself.

Previous research has shown the real-world application of genetic algorithms and neural networks. Dhawan and Biagioni (2005), used genetic algorithms and neural networks with a relatively easy flying device, a one DOF flight controller. The purpose of Dhawan and Biagioni's research was to manage the thrust to a desired height, using a cost function to give feedback to the algorithm. The cost function feedback helps to optimize the neural network. Dhawan and Biagioni proved that using genetic algorithms and neural networks a self-learning one DOF device can be created and trained in real-life. The current study is like Dhawan and Biagioni's one DOF helicopter; however, a 54mm electrically ducted fan (EDF) is used, as seen in Figures 1-3 in Appendix A, and will only be controlled with a genetic algorithm. Previous research uses genetic algorithms and similar implementation methods to that of the current study.

The overall approach for this research will be to train genetic algorithms to learn from the prior iterations of failures from the one DOF device. The idea of genetic algorithms came from natural selection and evolution. Mitchell (1996), explains genetic algorithms as the following: Evolutionary computation that rules are typically "natural selection" with variations due to crossover and/or mutation; the hoped-for emergent behavior is the design of high-quality solutions to difficult problems and the ability to adapt these solutions in the face of a changing environment (p. 4).

All testing will be done using a one DOF robot as an agent in the algorithms. This project will test the boundaries of training algorithms to learn from real-life data sets. An underlying goal of this project is to make an "easy to learn from" project so that others can replicate and build onto it. The project only uses open source tools and technology. OpenSource.com defines open source as, "something people can modify and share because its design is publicly accessible" ("What is Open Source," 2019). Open source tools that the project uses are Arduino Uno, C++, Python, project code and CAD designs to 3D print. All code and CAD designs are hosted on github.com for public access. Most universities have the rest of these tools and technologies available to the student population.

The algorithms created will open a whole new window to control theory for researchers in computer science, aerospace engineering, and applied artificial intelligence fields. The research is useful because it can fundamentally change how we explain these algorithms with a simple platform to learn how to implement genetic algorithms in real-life scenarios with real-world results.

II. METHODOLOGY / APPROACH

A. Design

This study is based on a quantitative empirical research design. The researcher uses trials to test data sets, attempting to find the optimal value of a one DOF device. Observations are recorded during the trails and data is received from the genetic algorithms. A percentage is given by Python to determine the overall percentage increase of the total score.

B. Instruments

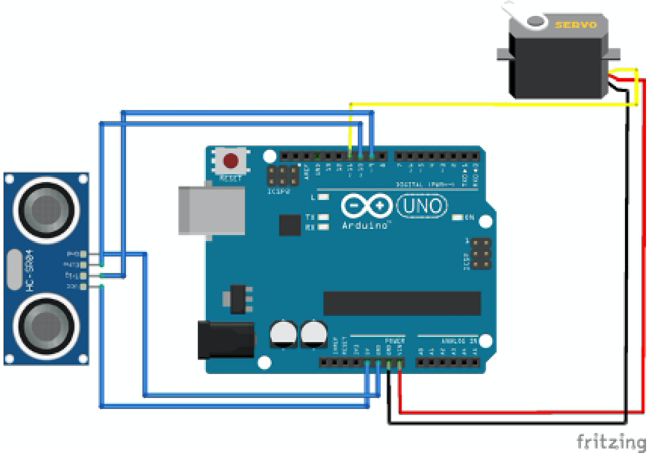
In order to run the genetic algorithms needed for this experiment to be successful, an Arduino Uno, programming languages, a laptop, an EDF, Electric Speed Controller (ESC), and an ultrasonic ranging module are required.

*This work was supported by Embry-Riddle Aeronautical Universities IGNITE program

[1]D. Wilczak is a computer Science student at Embry–Riddle Aeronautical University, 1 Aerospace Blvd, Daytona Beach, FL Wilczakd@my.erau.edu

[2]P. Misra is with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA p.misra at ieee.org

*1) **Ardunio Uno**:* The Arduino Uno is an open-source micro controller board. This micro controller board comes with digital and analog pins. The pins have different purposes. Some pins are used for sending 3.3V and 5V power to other devices, and other pins are used as input, and output pins, which send pulse width modulation (PWM) signals to other devices.



For this experiment, digital pins 9, 10, and 11 will be used to send PWM signals and receive sensory information from the ultrasonic range module. Digital pin 9 (TRIG) will be used for the signal output, pin 10 (ECHO) will be used for the signal input, and pin 11 will be used for PWM signals to the ESC.

*2) **C++**:* C++ is another programming language used to provide input and output procedures to the Arduino Uno.

*3) **Python**:* Python is a high-level programming language that allows for rapid prototyping of machine learning algorithms. For this experiment, Python will be used to send and receive serial data from the Arduino Uno enabling it to perform calculations for the genetic algorithm. Python is running on a laptop sending instructions to the Arduino Uno, allowing C++ to interpret and then perform the instructions.

*4) **SQL Database**:* A SQL database is used to store all data coming from muliple test platforms. Its also used to find and trends and query data.

*5) **One DOF Device**:* The one DOF device is an evolving altitude controller. The device is made up of an EDF fan, the USRM, a carbon fiber tube, assorted wiring, and two 3D printed parts, the intermediary connector, and platform. A visual of the one DOF device is shown in Appendix A, Figure 1. The platform for the device is designed to work functionally with all parts and pieces needed for the EDF to run. The EDF required that the 120mm x 20mm carbon fiber tube connect directly to the fan. CATIA software is used to design an intermediary connector that was 3D printed. The intermediary connector was designed to be wider than the platform. The connector is wider than the platform so that the ultrasonic range module has a wide enough space to measure the distance from the module to the device accurately. Appendix A, Figure 4, shows a visual of the connector "blocking" the USRM so the module can receive an effective height range. Appendix A, Figure 3, provides a visual of the EDF mounted inside the connector, secured with two screws. The intermediary connector allows the user to easily install and remove the fan from the connector and carbon fiber tube. The USRM is embedded directly below the EDF, as seen in Appendix A Figure 2. All the parts and pieces, except for the USRM, are connected to the carbon fiber tube which is connected to the platform by one pin acting as a hinge point. The ultrasonic range module measures the distance from the module to the device. A high-frequency ultrasonic sound signal is sent from the module to the device and acts as sonar returning to the module allowing it to calculate the distance.

III. Algorithm

A standard genetic algorithm is used to solve the search-based problem in this research study. The genetic algorithms are used to optimize the throttle values, sustaining a hover of 6cm above the USRM. Each throttle input lasts 1 second and is encoded as a gene. A gene represents one throttle value, lasting 1 second, inside the chromosome of 10 seconds. Ten genes make up one chromosome. The groups of chromosomes, which are made up of genes, are what makes up an entire population. Figure 2 shows a visual of how genes, chromosomes, and the population are connected.

The current study has a search space of 106 (ten million) possible chromosome configurations. Using a genetic algorithm allows for rapid search through the search space. Search space involves how big of an "area" the algorithm searches for the optimal value. Is the robot going to be able to train within a reasonable time frame? Genetic algorithms run into problems of being applied to an unreasonable sized search space, which causes the algorithms to search for a longer period. Jboss (n.d.) community documentations shows how pertinent search space is to the research when they explain, "An algorithm that checks every possible solution can easily run for billions of years on a single real-life problem… What we really want is to find the best solution in the limited time at our disposal" (p.1). Parameters will be used to limit the size of the search space. A smaller search space allows for the genetic algorithms to search more effectively. Whitley (n.d.) explains, "as long as the number of 'good solutions' to a problem is sparse with respect to the size of the search space, then random search or search by enumerations of a large search space is not a practical form of problem-solving" (p. 3)

There are five phases to the genetic algorithm, initial population, fitness function, selection, crossover, and mutation. Initial population is defined as the total number of chromosomes. The initial population for the current study was 10 chromosomes. The fitness function determines how well a chromosome does at a specific task. To test the fitness function in this study, chromosomes are encoded into a string and serially sent to the Arduino Uno. The Arduino Uno runs it and sends the data back to the Python script. The fitness function of the current study is to see how long the EDF can hold a position of 6cm above the USRM. If the fan hits 6cm during that exact second, that specific chromosome receives 1 point toward the total chromosome score. There is a max of 10 possible points. Selection is the next step; It is defined as the phase in which the fittest chromosome is selected for crossover. Elitism is used for selection criteria for this study. Elitism is defined as the selection of the best. This study will be selecting the 2 best or fittest chromosomes. Elitism was used for its simplicity so that new learners can understand a simplistic approach to the selection process. Crossover is the most prolific phase in a genetic algorithm. Crossover splits two chromosomes and crosses them over to create a "child." Think of each chromosome as a parent who passes their genes onto their child. Figure 3 shows a visual representation of crossover. Crossover was accomplished by selecting 5 random cut points between the 1st and 9th gene, creating two new chromosomes. The process of creating two new chromosomes is called reconstruction. The final phase is mutation. Mutation is defined as a low random probability that the gene mutates to a new or different value of the initial gene. In this study, there is a 1 in 10 chance, for each gene, that mutation will occur during the genetic algorithm phases. This means that there is a 1 in 15 chance that a gene will mutate into a new or different throttle value.

IV. Procedures

All the procedures for the study are done within Python. The only part of the procedure that is not is fitness function. The fitness function data is computed using the Arduino Uno and C++. Once the fitness function phase has been completed the data is sent back to the Python script. The study starts by generating the initial population. Next, it computes the fitness of each chromosome. Then, the generation is performed. A generation includes the next four phases in the genetic algorithm, selection, crossover, mutation, and computation of fitness scores. These four phases are repeated until the number of generations stipulated by the user has been completed. Once the generation has been completed Python generates a graph to represent the results from the trial.

V. Preliminary Results

A total of 5 generations were run. Graph 1 shows the total score of each generation's population. Each chromosome runs in a generation, and every time it reaches the desired altitude of 6cm, one point is added to the chromosome score. All the scores are added together to get the total score of the generation. Figure 4 shows an example of how generation zero's total score was obtained. Generation 3 had the highest total score. Generation 3's population obtained the highest total score by reaching the desired altitude of 6cm a total of 39 times.

GRAPHS

Graph 1 shows how Python generates the scores of a series of different results. The history of the scores from each generation is shown in a bar graph. The graph also shows the highest score achieved by the third generation. From the preliminary results of the graph, we have come to find out that the device learns quickly. Graph 1 shows after 5 generations that the device has made a 171

VI. Discussion

The current study tested to see if a flight-controlled system could be optimized with no pre-determined mathematical model. The results showed that it is possible to optimize a one DOF device only using a genetic algorithm tested in real-time. The results of the study indicated that after 5 generations the one DOF made a 171 percent increase in the total score. An analysis of the data in the graphs shows that generation 3 was the most successful in reaching the desired altitude of 6cm.

In order to maximize optimization, future studies could use a probability approach for use in the selection process. The reason for this is that it would give the chromosomes a better chance of finding the correct crossover match. Another method for future studies to improve upon the current study would be to do a comparative analysis of using a PID and a genetic algorithm. Johnson and Moradi (2005) explain that PID controllers use different algorithms and controls to operate (p.9). Johnson and Moradi make a great comparison of how different control methods that can be used other than the ones I am implementing in this research. My follow-on research will be comparing a PID controller, which uses a mathematical model to the current study, which uses a genetic algorithm to search for the optimal value.

In retrospect, the time it takes to test all the possible combination of chromosomes is time intensive. A better approach may be a combination of simulated trials followed by real-life testing and optimization. This two-way approach would have both benefits of running the initial generations in a simulation where trials are processed using computing power, followed by real-life trials and training to optimize the system further.

VII. CONCLUSIONS

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression, "One of us (R. B. G.) thanks . . ." Instead, try "R. B. G. thanks". Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[5] E. H. Miller, "A note on reflector arrays (Periodical styleÑAccepted for publication)," IEEE Trans. Antennas Propagat., to be publised.

[6] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical styleÑSubmitted for publication)," IEEE J. Quantum Electron., submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style)," IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740–741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[9] M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.

[10] J. U. Duncombe, "Infrared navigationÑPart I: An assessment of feasibility (Periodical style)," IEEE Trans. Electron Devices, vol. ED-11, pp. 34–39, Jan. 1959.

[11] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," IEEE Trans. Neural Networks, vol. 4, pp. 570–578, July 1993.

[12] R. W. Lucky, "Automatic equalization for digital communication," Bell Syst. Tech. J., vol. 44, no. 4, pp. 547–588, Apr. 1965.

[13] S. P. Bingulac, "On the compatibility of adaptive controllers (Published Conference Proceedings style)," in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 8–16.

[14] G. R. Faulhaber, "Design of service systems with priority reservation," in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 3–8.

[15] W. D. Doyle, "Magnetization reversal in films with biaxial anisotropy," in 1987 Proc. INTERMAG Conf., pp. 2.2-1–2.2-6.

[16] G. W. Juette and L. E. Zeffanella, "Radio noise currents n short sections on bundle conductors (Presented Conference Paper style)," presented at the IEEE Summer power Meeting, Dallas, TX, June 22–27, 1990, Paper 90 SM 690-0 PWRS.

[17] J. G. Kreifeldt, "An analysis of surface-detected EMG as an amplitude-modulated noise," presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[19] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[20] J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.