

DEVELOPMENT OF INTELLIGENT ALGORITHMS FOR UAV PLANNING AND CONTROL

Victor VLADAREANU^{*}, Elena-Corina BOSCOIANU^{**}, Ovidiu-Ilie SANDRU^{***},
Mircea BOSCOIANU^{****}

^{*}Institute of Solid Mechanics of the Romanian Academy, Bucharest, Romania
(vladareanuv@gmail.com)

^{**}National Institute for Aerospace Research "Elie Carafoli", Bucharest, Romania
^{***}Politehnica University, Bucharest, Romania (oisandru@yahoo.com)

^{****}"Henri Coandă" Air Force Academy, Braşov, Romania
(boscoianu.mircea@yahoo.com)

DOI: 10.19062/2247-3173.2016.18.1.29

Abstract: *The paper presents a review of intelligent techniques used for the optimization of an UAV mission plan using sensory input. The constrained problem is evaluated using evolutionary algorithms (such as GAs and PSO), while extenics logic is used to evaluate the sensor fusion state. The review is part of a innovative project dealing with swarm UAV missions and AirWing testing.*

Keywords: UAV, evolutionary algorithms, neutrosophy

1. INTRODUCTION

Beginning with the 1950s and 1960s, several independent research papers dealt with the possibility of mimicking the basic idea of evolution, as found in nature, to aid in the optimization of engineering problems (Mitchell, 1998) [1]. Thus the field of evolutionary computing was born. Today, population-based meta-heuristic methods such as genetic algorithms (evolutionary algorithm) and particle swarm optimization (swarm algorithm) are beginning to find a place in the optimization of controllers in industry applications [2-4].

In a most interesting and helpful paper, Thomas and Poongodi (2009) argue the benefits of using a genetic algorithm to tune the term gains of a controller in comparison with the classical Ziegler-Nichols scheme [2]. A genetic algorithm is a search algorithm that attempts to improve an initial population of 'chromosomes' (initial solutions) in regards to a user-defined fitness function. Based on the mechanisms of natural selection, the algorithm employs stages of 'selection', 'crossover' and 'mutation' to advance to the next population [1,2]. The authors run a series of simulations on both controllers and find that the GA-tuned system has a much faster response and less steady state error. An interesting idea put forth in the paper is using the classical values as the starting population for the genetic algorithm [2] (while GAs generally start from a random population, it can be specified by the user if so desired (Mitchell, 1998) [1]).

Khan, Abdulazeez et al (2008) delve deeper into the uses of genetic algorithms by attempting to design an optimal fuzzy logic controller [3]. They build upon the work of Byrne (2003) [5] in using a GA to tune the rule base of a fuzzy controller and compare it with a conventional controller. While the subject is treated superficially at times, it does

provide an interesting look into what could be the future of numerical processing as far as controllers are concerned. Byrne himself notes (2003) that classical control, in comparison to a GA-tuned fuzzy, is more prone to faults for a multiple-input multiple-output (MIMO) system [5].

Particle swarm optimization is a newer approach to evolutionary search algorithms which is derived from the natural movement of insects and other ‘social’ biological organisms (Kennedy, Eberhart, 1995) [6]. Gaing (2004) presents a novel approach using particle swarm optimization to determine the gains of a controller in an AVR system and quotes Fogel (2000) in claiming that ‘the premature convergence of genetic algorithms degrades their performance and reduces its search capability’ (Gaing, 2004) [7], [8]. He then goes on to compare it with a GA-tuned PID controller, arriving at the conclusion that it provides more robust stability and efficiency, while not suffering from the drawbacks of premature convergence and high computational requirements of the GA-tuned scheme, which obviously also makes it faster [7]. This is perhaps one of the best written papers referenced here and its findings should be of great interest for future research.

2. EVOLUTIONARY ALGORITHMS

Evolutionary Algorithms (EAs) are a type of optimization algorithms which are derived from the evolution of biological populations found in nature. As a search optimization algorithm, it seeks to minimize the value of an objective function supplied by the user in the context of user – defined constraints for the search space, the dimension of which equals the number of independent variables that are part of the objective function (Fogel, 2000) [8].

A Genetic Algorithm (GA) is the most popular type of EA. It mimics the process of natural evolution to obtain a heuristic result for the search problem. In the case of natural selection, constant improvement in a species is brought about through continuous exchange of the genetic makeup of its individuals [5]. This results in fitter, better adapted individuals over the course of a number of generations. A genetic algorithm adapts this process for use in numerical optimization problems.

Prospective solutions are coded as bit strings corresponding to the basic building blocks found in nature, from which they borrow the name of chromosomes. These individuals make up the population of the Genetic Algorithm. In cases where there is some initial knowledge or acceptable estimate of the solution points (for example when we would like to improve an existing heuristic solution), an initial population can be specified for the algorithm. Otherwise, it will be created randomly. Each individual’s fitness is tested against the general fitness of the population as part of the process of passing from one generation to the next. This is done by applying the fitness function, which is the function to be minimized, to each individual. A fitter individual will have a higher probability of being selected for reproduction (passing on its genetic information) [5].

The next stage of evolution can then be broken down into selection, crossover and mutation. Selection is the process whereby the individuals from the previous generation are selected probabilistically based on their fitness value into an intermediary population. Crossover is analogous to the act of parenting, producing new individuals from those previously selected by recombining their chromosomes. Mutation means randomly changing a bit within a chromosome with the aim of making the algorithm less susceptible to falsely converging on local minima. In genetic algorithms, mutation is viewed as a background process with a very low probability [5].

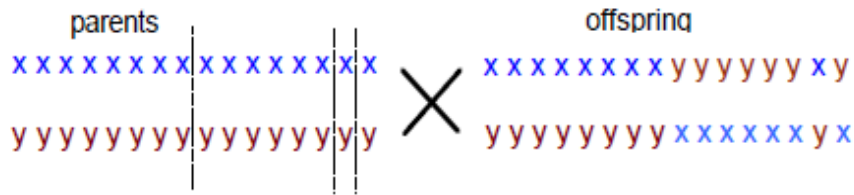


FIG.1. Crossover in a genetic algorithm

While there are a number of options for when a GA should terminate, the most common are specifying a minimum change in the overall fitness function that should be met in order to continue to the next generation, a fixed number of generations for which the program should be run, or both (whichever comes first).

Particle Swarm Optimization (PSO) is another evolutionary algorithm that has seen increased interest in recent years since being introduced by Kennedy and Eberhart in 1995 [6][7]. Instead of using genetic information to modify the individual, the PSO algorithm lets particles traverse a solution space at speeds which are dynamically adjusted according to their track history and other particle's track history. Each particle records a history of the best solution it has found, as well as the best solution found by the overall population. At each time, the velocity of each particle is dynamically changed towards those two bests achieved so far [6].

Although as of this writing there is no standard PSO Toolbox included in Matlab, quite a number of them are readily available on the developer's community website (Rapai, 2008) [9]. The toolbox's implementation and use in Matlab are very similar to those of the GA Toolbox.

There is a lot still to be discovered about evolutionary algorithms in general as well. As mentioned previously, the definition of the objective function is paramount for the use of an evolutionary algorithm, and can, in theory, be weighed differently depending on the preferred system behavior. More complex objective functions, the inner workings of the algorithm itself (mutation rate in GAs, for example) and the impact of various implementation options (such as maximum number of generations, initial population, etc.) amount to virtually limitless opportunities for research in one of the fields found at the forefront of artificial intelligence.

3. ARTIFICIAL NEURAL NETWORKS

Using artificial neural networks and regression implementations to model the information received from the sensor, an accurate reading of the measured sensor array can be obtained without the need for costly components. It also makes the scanner less susceptible to anomalous readings given by the interpolation of different wave signals. For a time-of-flight optical scanner, for instance, this is achieved by using the first three reception trigger signals for the returning wave to estimate the measured distance, rather than having to use a fourth trigger signal on the falling slope, which introduces dead time and may decrease performance due to wave interpolation [10].

There are a great number of methods which can be used to model such an approach. The first step is to test the assumption using linear regression and neural networks on the mean values generated for each point at each of the four triggers. The total sum of differences across all values is called the cost function (J).

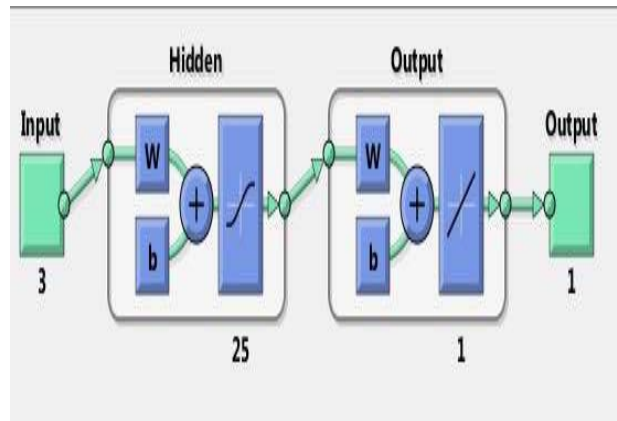


FIG. 2. Artificial neuronal network with one hidden layer (25 neurons)

An artificial neural network consists of a number of ‘hidden’ features (or neurons) associated with a network of weights, which are improved each iteration until their prognosis is within an accepted tolerance or the number of iterations expires. An example of the investigated network topographies is available in Figure 2.

From the available data, approximately 70% of the examples are used for the actual training of the network. Another 20% is used for cross-validation, whereby the weights are adjusted again based on the observed deviations. The remainder is used as a test for the obtained network, which provides a measure of its accuracy and of whether the network over-fits the available data.

By using the artificial neural network toolbox in Matlab with the obtained data and training a number of network configurations, as well as running through a variety of linear regression models, estimates can be obtained for comparison with the actual mean values derived from the experimental data.

The end result is the selection of an artificial neural network to be implemented into the sensor software with the trained weights obtained in the simulation. This will lead to faster and more robust results being obtained from the raw data, as well as the ability to run more tests using the sensor in actual situations (both static and dynamic).

4. EXTENICS

Extenics is a science whose stated aim is to deal with unsolvable problems. With applications in artificial intelligence, business, marketing, planning, design, control theory and image processing, to name just a few, it is one of the fastest developing new fields of study today.

In his 1983 paper, ‘Extension Set and Non – Compatible Problems’, Cai Wen put down some of the earlier concepts and established the foundation of what would later become Extenics Theory, a wide – spanning inter-disciplinary science [11](the reference is the 1990 translation). Early papers were few and far between, before the framework of Extenics was established and it became recognized as a field of scientific research in its own right. Further expansion brought important updates both in the formal expressions and the tools used in Extenics. Thus was developed a formalism which would facilitate working within the latest trends of technical processing and whose improvement is still on-going as an important direction in Extenics [12].

One of the first papers to discuss an involved engineering application was published in 1994 by Prof. Wang and entitled ‘Extenics control’ [13]. It provides a blueprint for the first Extenics controller, which has subsequently undergone numerous modifications and opens up the field of Extenics control proper.

Since the beginning of the new millennium, an expanded research group combined with the years of existing built-up expertise to produce new directions of research in the field. This outlined the beginning of their respective specialized work, making Extenics applications wide-spread in the academic and business world, especially in the Asian continent. In 2012 a monograph work entitled ‘Extenics Engineering’ was published in English [12] (the previous Chinese version had come out in 2007), marking a very important step to promoting Extenics world-wide.

Extenics is said to be a science combining Mathematics, Engineering and Philosophy [12]. With respect to its application and use it is a trans-disciplinary science, classified as belonging to the wider field of Artificial Intelligence. Extension Logic is purported by the authors to extend fuzzy logic, much in the same way that fuzzy logic extends the classical Cantor logic [11]. This is necessary for work with Extension Sets in contradictory problems and for Extension Strategy Generation Systems (ESGSs)[14]. An example of such strategies is shown in Figure 3.

The main goal of Extenics, as set forth in almost every paper that deals with the issue, as well as throughout its theoretical mainframe, is the generation of solutions to contradictory problems [11]. This is of course very important in and of itself, however in Extenics the process by which this goal is to be achieved is also of great scientific relevance. Contradictory problems, or seemingly impossible problems, have been solved throughout history by using human ingenuity. Extenics studies the creation of such innovative ideas and seeks to develop procedures for understanding and creating new, original thought with the help of modern day advances in computer science. It is partly for this reason that Extenics is classified as part of Artificial Intelligence and is one of the fastest developing new fields of study in the world today [15].

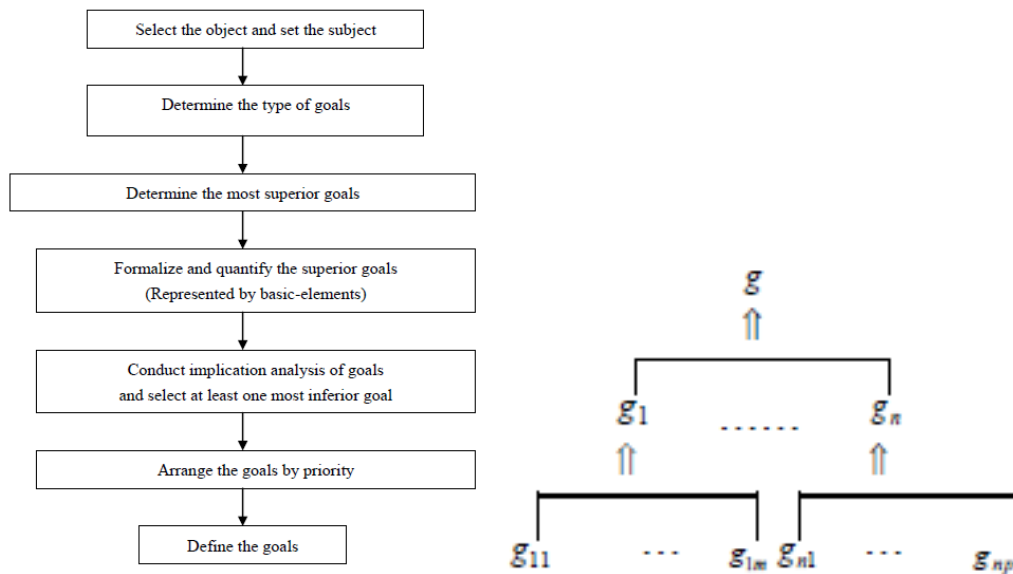


FIG.3. Extenics Strategy Generation and Goal Decomposition

Aside from Extension Sets and Logic, Extension Transformation is one of the core concepts in the theory [12]. It is said that the only constant in life is change, and Extenics emphasizes the need to adapt problem specifications to ever-evolving contexts. Transformation is at the basis of solutions to contradictory problems, both from a theoretical and a practical standpoint. Basic element representations are designed to

support dynamic modelling (as a type of parametric modelling) and solution algorithms consist, for the most part, of iterations of transforms and compound transforms.

CONCLUSIONS

The methods and algorithms presented throughout this paper have been thoroughly studied and simulated and will be used for the implementation of UAV motion strategies in swarm unmanned aviation projects, both in high-level decision algorithms and flight optimization procedures, as well as in lower-level implementations of control and sensor fusion architectures and hierarchies. Further implementation and simulation will also feature software-in-the-loop mission planning and intelligent formation control, of which the algorithms described in this paper will be an integral part.

ACKNOWLEDGMENT

This work was accomplished through the Partnerships Program in priority fields - PN II, developed with the support of MEN-UEFISCDI, PN-II-PT-PCCA-2013-4-1349, MASIM project no. 255/2014.

REFERENCES

- [1] Mitchell, M., *'An Introduction to Genetic Algorithms'*, First MIT Press, 1998
- [2] Thomas, N., Poongodi, P., *'Position Control of DC Motor Using Genetic Algorithm Based PID Controller'*, Proceedings of the World Congress on Engineering 2009, Vol. II, WCE 2009, Jul. 2009
- [3] Khan, S., Abdulazeez, S.F., Adetunji, L.W., Alam, Z., Salami, M.J., Hameed, S.A., Abdalla, A.H., Islam, M.R., *'Design and Implementation of an Optimal Fuzzy Logic Controller using Genetic Algorithm'*, Journal of Computer Science 4, 2008
- [4] Gaing, Z.L., *'A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System'*, IEEE Transactions on Energy Conversion, vol. 19, no. 2, Jun. 2004[3] T. R. Nail, *A disturbance-rejection problem for a 3-D airfoil exhibiting flutter*, Thesis, Virginia Tech., 2000;
- [5] Byrne, J.P., *'GA-optimization of a fuzzy logic controller'*, Master's Thesis, School of Electronic Engineering, Dublin City University, 2003
- [6] Kennedy, J., Eberhart, R., *'Particle Swarm Optimization'*, Proceedings of the IEEE International Conference on Neural Networks, vol. IV, 1995
- [7] Gaing, Z.L., *'A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System'*, IEEE Transactions on Energy Conversion, vol. 19, no. 2, Jun. 2004
- [8] Fogel, D.B., *'Evolutionary Computation: Toward a New Philosophy of Machine Intelligence'*, 2nd ed. IEEE Press, 2000
- [9] Rapai, M., *'Particle Swarm Optimization (PSO) Algorithm'* (Matlab code), Nov. 2008, <http://www.mathworks.com/matlabcentral/fileexchange/22228-particle-swarm-optimization-pso-algorithm>
- [10] Iliescu, M., Vladareanu, V., Serbanescu, M., Lazar, M., *'Product Development of a New 2D Time-of-Flight Scan Laser'*, Journal of Control Engineering and Applied Informatics, submitted for publication, Oct. 2015
- [11] Cai Wen. *Extension Set and Non-Compatible Problems [A]. Advances in Applied Mathematics and Mechanics in China [C]*. Peking: International Academic Publishers, 1990.1-21.
- [12] Yang Chunyan, Cai Wen, *'Extension Engineering'*, Science Press, Beijing, 2002
- [13] Wang Xingyu, Li Jian. „*Extension Control*” [J]. *Control theory & applications*, 1994, 11 (1) : 125-128
- [14] Li Lixi, Yang Chunyan, Li Huawen. *Extension Strategy Generation System [M]*. Science Press, 2006.
- [15] Boscoianu, M., Cioaca, C., Vladareanu, V., Boscoianu, C., *"An Active Support Instrument for Innovation in Deep Uncertainty – the Strategic Management Ingredients in Robotics and Mechatronics"*, Procedia Computer Science 65 (2015): 210 – 217. ELSEVIER, www.sciencedirect.com, doi:10.1016/j.procs.2015.09.112, www.elsevier.com/locate/procedia, ISSN 1877-0509, ELSEVIER