FUZZY GENETIC CONTROLLERS FOR THE AUTONOMOUS RENDEZVOUS AND DOCKING PROBLEM

Vijayarangan Gopalan, Abdollah Homaifar, M. Reza Salami,
North Carolina A&T State University
Bijan Sayyarrodsari
Stanford University
and
R.W. Dabney
Marshall Space Flight Center

Keywords: Transportation, Docking, Fuzzy, Genetic Algorithm

ABSTRACT

Autonomous rendezvous and docking has been defined as one of the primary goals in today's space technology. Autonomous operation of an unmanned space vehicle in a real world environment poses a series of problems. The knowledge about the environment is in general incomplete, uncertain and approximate. Perceptually acquired information is not precise, sensor's noise introduces uncertainty and imprecision, sensor's limited range and visibility introduces incompleteness. In this study, fuzzy logic and genetic algorithm (GA) have been applied to this problem in order to perform better in the case of all these problems. Fuzzy and GA combination imitates the role of human in the decision process.

Background Information On Autonomous Rendezvous And Docking Technology

The methodology presented in this research can be applied to any transportation problem. Some of the problems include decision making and evaluation of transportation system, transportation network design and traffic scheduling. Decision making and evaluation of a transportation system comprise of achieving multiple objectives using one of the alternate methods. All of these methods cannot satisfy all the objectives. Human decision making is required to weigh all the methods and choose the right alternative. Fuzzy logic with its role of human expert and GA being a strong search and optimization algorithm can mimic the human expert's role. The problem can be formulated and GA-fuzzy method can be applied to find the method that satisfies all the objectives optimally. In the transportation network design, GA-fuzzy method can be applied to find the shortest path or routes to be traversed between the different nodes (for example, bus terminus). GA has been used to solve the traveling salesman problem (TSP), gas pipeline and other combinatorial problems. GA and fuzzy have also been used to find the optimal solution for a lot of scheduling problems.

Autonomous rendezvous and docking (ARD) is one of the most challenging aerospace problems. ARD is required for tasks such as refuelling and resupply of a spacecraft. The rendezvous and docking can be accomplished by manned control or autonomous control. Manned control can be from space or from the ground. ARD saves money and risks to the astronaut. ARD will be needed since manned control of the later phases of rendezvous from the ground or even from the space gets extremely tough or impossible because of astronaut reaction and time delays. It also reduces the ground station workload. The rendezvous mission is generally divided into six mission phases. They are launch, approach phase, terminal rendezvous phase, stationkeeping phase, docking approach phase and docking linkup phase. All these phases do not have to be included in a rendezvous mission and not necessarily in that order.

Autonomous Rendezvous And Docking

The autonomous rendezvous and docking is one of the space transportation's interesting problems and numerous techniques [9] have been suggested for solving it. The study uses a straightforward concept based on the use of a standard video camera and a Remote Manipulator System (RMS) docking target. This concept was originally tested at Marshall Space Flight Center (MSFC). The system consists of a chase and a target spacecrafts. The target spacecraft has a three-dimensional target attached to it near its docking port. The chase spacecraft has to be maneuvered towards the target. A video camera of charge coupled device type acquires the images of the target, which is illuminated by two different wavelengths of laser diodes. Then Inverse Perspective Transformation is performed, which involves the computation of three-dimensional relative position and attitude from the two-dimensional image frame coordinates of the three retroreflector images. This is the exact reverse of the camera operation, which is creation of two-dimensional image of a three-dimensional target. This transformation can be performed deterministically, but the resulting equations are very complex and time-consuming when executed on computer. Therefore this transformation was done by using neural networks. The neural networks are used to calculate the range, azimuth, elevation and the relative roll, pitch and yaw alignment. These are the six degrees of freedom. This information is used to drive the system's fuzzy Proportional derivative (PD) controlled autopilot. The autopilot output consists of net delta-velocity and delta-omega (acceleration) maneuver commands which are executed by the reaction control system (RCS) during the next sample interval. Autopilot commands are then passed to a thruster selection algorithm, which chooses the appropriate combination of thrusters to fire and calculates the length of time each is to burn. Further information on Docking and Space technology can be obtained from [1], [9], [10].

Fuzzy PD controlled autopilot is a controller designed using Genetic algorithm and Fuzzy logic. Fuzzy logic was developed by Zadeh in the 1960's in an attempt to deal with complex systems which are beyond exactness and precise description. Fuzzy logic uses linguistic descriptions for the system variables. Genetic Algorithm is a search and optimization technique based on the mechanics of natural selection and natural genetics [2]. Genetic algorithm has been used to design the fuzzy rules and combining this with fuzzy membership functions, an initial value of PD controller coefficients were obtained. Then these coefficients were adaptively adjusted with respect to output errors till the chase spacecraft docks near the target.

Genetic Algorithm

Traditional optimization techniques can be divided into three main categories: calculus-based, enumerative and random. Calculus-based methods rely on derivative information to find local extrema. In most real world problems, derivatives may not exist, or the functions may be discontinuous, thus rendering these techniques useless. These methods tend to be around the peaks of local extrema as opposed to global extrema. Enumerative techniques evaluate every discretized point in the search space. It is obvious that this technique is extremely inefficient and impractical for large search spaces. Random techniques are no better than enumerative methods in the long run.

Genetic Algorithm (GA) was developed by John Holland and his colleagues in an attempt to model the processes of natural selection and survival of the fittest. GA has a probabilistic nature; while it uses a structured yet randomized search procedure, it exploits historical information to find improved search points. Holland modeled the mechanics of GA after nature's evolutionary process, including chromosomes, selection, reproduction, and mutation. Living beings are defined by their genes and chromosomes. How well a creature survives will depend on the traits found in the chromosomes. Stronger beings are more likely to be selected by nature to reproduce, thereby mixing the stronger chromosomes to form strong offspring. Mutation of chromosome information may occur with small probability to create an individual somewhat different from the parents. Genetic Algorithm requires the problem being evaluated to be formulated in such a way that a performance measure can be stated as a cost (objective) function. The variables in the problem are encoded in a low cardinality string, and a population of these strings are generated. The GA works to either maximize or minimize the cost function based on the performance of each individual. Selection of individuals for reproduction is based on the performance measure of the strings; strings with higher performance are more likely to be selected for reproduction. A new population is then formed by the reproductive process on the selected individuals. Several selection schemes have been developed [2], with the more typical ones used being roulette wheel selection and tournament selection. The reproductive process involves crossover and mutation. Mutation is a process whereby individual bits are changed with a low probability . Mutation serves to introduce new search points in the search space, and add diversity to the population. Crossover is an operation whereby a portion of one string is combined with a portion of a second string. The resulting string is then placed in a new population. There are different crossover operators discussed in Goldberg [2].

Fuzzy Logic

Fuzzy logic can be thought of as a generalization of set theory. Conventional set theory requires that an object be completely in a set, or not in the set at all, while fuzzy sets allow an object to partially belong to the set. Therefore, every fuzzy set should be characterized by a membership (characteristic) function which associates with each element a real number in the interval [0,1], representing its degree of membership in that fuzzy set. A set of operators, such as union, intersection and product, are appropriately defined for fuzzy sets via their membership functions. Using these operators, fuzzy implication inference rules are defined utilizing the concept of linguistic variables [12]. This provides a rigid basis for a type of reasoning commonly known as approximate reasoning. Fuzzy logic controllers have been developed as an application of the above mentioned conceptual framework. A fuzzy controller consists of four principal components [7]: fuzzification interface, fuzzy knowledge base (containing membership function, linguistic control rule), fuzzy inference engine and a defuzzification interface.

The fuzzification is a process by which the crisp, real world input values are converted into fuzzy linguistic values. These fuzzy values can be labels for fuzzy sets such as Positive Medium (PM), Negative Small (NS), etc. The membership functions reflect expert's knowledge about the application domain and the way they are defined substantially affects the performance of fuzzy controller. Fuzzy rules consists of a premise with one or more antecedents, and a conclusion with one or more consequences. The individual rules in the set are connected through the operator "also". Given the fuzzy rule base and the input values, the fuzzy controller then applies some type of inference operation. The inference engine performs two functions: determination of applicable rules for a set of inputs, and inference of output fuzzy set(s). Several inference operators have been developed, but the two most common types are Min of Mamdani and Larsen's product [7]. The Min operator takes the minimum of all fuzzy membership values in the "if_side" for the rule being evaluated, and clips the corresponding output membership at this level. The product operator scales the output membership as opposed to clipping them. Interpreting "also" as a Max operator combines the individual output membership functions, generated by each rule, and produces the final output fuzzy set. The result of the inference engine must now be defuzzified. Defuzzification serves to provide crisp, numeric output to the process being controlled. Two methods of defuzzification are used the most: Maximum membership and centroid or center of gravity [6]. The former chooses the output value corresponding to the maximum degree of membership in the output fuzzy set. One problem associated with this method is that typically many output values will have the same membership level, particularly when using the Min_Max inference. The second problem with maximum criterion method is that much of the information in the output set is ignored and lost. The centroid is the most commonly used method. An average, or weighted sum of the output values is calculated, yielding a single crisp value.

Integration Of Fuzzy Logic And Genetic Algorithm

The application of GA to fuzzy logic controllers holds a great deal of promise in overcoming two of the major problems in fuzzy controller design, design time and design optimality. Previous work has been done mainly in two areas: learning the fuzzy rules and learning membership functions. The GA's robustness enables it to cover a complex search space in a relatively short period of time while ensuring an optimal or near-optimal solution. Because of this capability, GA is a natural match for fuzzy controllers. Thrift [11] examined the feasibility of using GA to find fuzzy rules. Karr [5] examined the feasibility of using a GA to find high performance membership function for a controller for the pole-cart system. While all these methodologies have provided improvements in fuzzy controller design, they have a major limitation; how can an optimal design be obtained when one of the two main components is designed using a non-optimizing method. Logically, to obtain an optimal rule set and set of membership functions, the two must be designed together so the links between them can be fully exploited. There is lot of research activity in this area. Homaifar and McCormick [4] examined the initial applicability of GA to solving the cart-centering problem and laid the foundation for this more in-depth study. The way the authors decoded the string for the membership functions deserves a mention. For the cart controller using a triangular membership function, they kept the center or peak of the triangle fixed. One bit per triangle was reserved for the base width of the triangle and decoding was done in such a way that minimum overlap between the membership functions was possible. Knowing the base width and the peak, the membership of both position and velocity could be determined. Moreover the string contained representation for the rules. GA was used to optimize a string of rules and membership functions. The controller thus designed, obtained very good results. The allele representation in each string was integer based as opposed to binary bits. Hogans et. al. [3] used GA to design fuzzy membership function and inference rules for variable structure control. Nomura et. al. [8] examined using a GA to determine both the membership function and optimum number of rules for a single input, single output nonlinear system. These examples show that by using GA to design both simultaneously, the two elements of fuzzy controllers can be fully integrated to deliver a more finely tuned, high performance controller. In this study, GA has been used to design rules and the endpoints of the membership functions.

Methodology And Simulation Results

Autopilot controller is based on a PD controller. The PD coefficients have to be designed and adaptively varied for effective docking. Typical example of the controller is as follows:

"output error = Kp*(Actual azimuth-desired azimuth)-
Kr*(actual azimuth rate-desired azimuth rate)" .. (1)

where Kp and Kr are proportional and rate coefficients respectively. Kp and Kr have been designed using GA and Fuzzy logic.

Initial Conditions

To find a satisfactory controller, the controller must be able to operate over the entire range of input spaces. For a GA to properly design fuzzy rules and membership functions, this fact must be integrated into function evaluation. This was done by using multiple initial conditions in the evaluation of each member of the population. If a single initial condition was used, then the GA would find a controller which would work well around that particular point but may fail elsewhere. This makes the choice of initial conditions an important consideration. The points must be chosen to sufficiently cover the input spaces, but at the same time, more initial conditions leads to increased run time for the program. Eleven initial conditions were considered for the simulation of each controller. In evaluating each member of the population, the total fitness of the individual was the sum of the fitnesses at each initial condition.

Fitness Function

The process was divided into two stages, an evolution stage and a refinement stage. In the evolution stage, the GA was used to find satisfactory controllers, while in the refinement stage, the GA used the previously developed controllers and attempted to get the optimized controller.

The fitness function associated with the two stages are as follows:

$$\begin{array}{c} \textit{Evolution Stage Fitness Function} \\ fitness = k \,/\, \sqrt{(e_1^2 + \dot{e}_1^2)} \\ \textit{Refinement Stage Fitness Function} \\ fitness = k * \sqrt{(e_1^2 + \dot{e}_1^2)} \end{array} \quad .. 2$$

where e_1 and $\dot{e}_1$ are position and rate errors and k is an arbitrary constant that will enable good convergence to an optimal solution. Fitness function derivation was one of the most crucial portion of the research. The fitness function had to incorporate the ability to produce a controller capable of docking the spacecraft successfully and reduce the noise to a desirable level. Furthermore, the fitness function must be formulated to discriminate between different individuals. The evolution stage lasted until generation 40 and the refinement stage was from generation 41 through generation 100. The fitness functions were obtained through experimentation. It was derived in such a way that it matched the performance of PD controller by the MSFC.

String or Chromosome Representation

The objective of the GA was to simultaneously design fuzzy rules and endpoints of the membership function of the inputs and the outputs. The GA-fuzzy combination has to design an initial value of proportional (Kp) and rate (Kr) coefficients for the PD controller of MSFC. Hence the outputs of GA-fuzzy are Kp and Kr. The inputs are error and error rate. The input and output spaces were divided into five fuzzy sets each. Hence, the controller could be termed as 5555 controller. Two inputs (error and error rate) were considered to be of five alleles each. Hence the rules came to be 25 alleles long for a single output. 25 rules for Kp and 25 rules for Kr. Chromosome is divided into small parts called as allele. The outputs (Kp and Kr) were also considered to be of five alleles each. Hence, the chromosome length was designed to be 70 (5+5+25+25+5+5) alleles. GA used was floating point version of genetic algorithm. The alleles in the string were of floating point representation. The inputs and outputs were partitioned into different fuzzy sets (negative medium (NM), negative small (NS), zero (ZE), positive small (PS) and positive medium (PM)). GA optimizes the rules, membership functions simultaneously and also optimizes the overlap between the different fuzzy sets. For the rules, the alleles were interpreted into different fuzzy sets as follows:

Range	*Fuzzy Sets*
$0 \le X < 1$	*NM*
$1 \le X < 2$	*NS*
$2 \le X < 3$	*ZE*
$3 \le X < 4$	*PS*
$4 \le X \le 5$	*PM*

.. 3

The rules, thus interpreted are as listed below:

"If error N M and rate error PM Then Kp PS"

"If error N M and rate error PM Then Kr PS" .. 4

Fuzzy logic Parameters

Membership function used was of triangular in nature. Rules used were the ones designed by GA. GA designed rules and the endpoints were used with the fixed center (or the peaks of the triangle) to obtain crisp value of Kp and Kr. The defuzzification operator used was centroid. Kp and Kr designed this way was taken as the initial guess and was adaptively varied subsequently. The adaptive variation was done as follows:

If output error>previous output error then Kp=Kp-constant
If output error<previous output error then Kp=Kp+constant
If output error>previous output error then Kr=Kr-constant
If output error<previous output error then Kr=Kr+constant

.. *(5)*

RESULT COMPARISON

Five of the six degree of freedom controllers were designed. The block diagram of the controller is shown in Fig. 1. It can be observed from the figure that the MSFC designed PD controller is the base and GA and fuzzy logic are placed on top of it. GA designs the optimal membership functions and rules. These rules are then evaluated and a crisp Kp and Kr values are determined by defuzzification. The Kp and Kr designed in such a way plugged into the controller equation. The output error and the autopilot command are used to select the appropriate thrusters to fire. The subsequent Kp and Kr values are adaptively adjusted according to equation (5). The results of two of the five controllers are shown in Fig. 2-4. The GA-fuzzy tuned PD controllers were compared with the PD controller designed at MSFC. Fig. 2 and 3 describe the elevation controllers. The controller designed in Fig. 3 has reduced the noise and final error has been reduced as compared to the one in Fig. 2. The elevation controller in Fig. 3 has brought the chase spacecraft to within 0.01 feet of the target while MSFC controller was 0.07 feet from the target on the elevation plane. The chase spacecraft is very near to the target on the elevation plane than the controller designed by MSFC. Fig. 4 describe the pitch controllers. The controller designed in this study is smoother and has less output error on the pitch axis from the target than the one designed by MSFC. The pitch controller designed in this study ends has the final output error of -5e-05 as compared to -5.8e-04 for the MSFC controller. The other three controllers (Azimuth, Yaw and Roll) have also shown noticeable improvement. The chase spacecraft is docked very close to the target spacecraft. The errors on all the five controllers indicate that there exists a docking misalignment with the target spacecraft and some modifications in the controller design have to be done.

Even though the performance of GA-fuzzy adaptive PD controller has improved performance, there is still considerable error on all the five degrees of the freedom control. The present design has to be modified to reduce these errors. The concept of hierarchy of controllers is going to be considered to improve the performance of the present design.

CONCLUSION

The five one degree freedom controller has been designed and has been shown to have an improved performance than the MSFC designed controllers. Further modifications will be done in the controller design and the concept of hierarchy will be considered. Present research in hierarchy has shown encouraging results.

ACKNOWLEDGEMENT

The work was made possible due to NASA (grant number 48057) and NSF (grant number 446009).

ADDRESS

Abdollah Homaifar, North Carolina A&T State University, Greensboro, NC 27405. Email: homaifar@ncat.edu

REFERENCES

[1] Dabney R.W., 1992, Application of Neural Networks to Autonomous Rendezvous and Docking of Space Vehicles, AIAA Space Programs And Technologies Conference, Huntsville, Alabama.

[2] Goldberg, D.E., 1989, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA, Addison-Wesley.

[3] Hogans, J., E., Homaifar, A., Sayyarrodsari, B., 1994, Fuzzy Inference for Variable Structure Control, J. of Intelligent and Fuzzy Systems.

[4] Homaifar, A., and McCormick, V.E., 1992, Full Design of Fuzzy Controllers using Genetic Algorithms, SPIE Vol. 1766 Neural and Stochastic Methods in Image and Signal Processing, pp. 393-404.

[5] Karr, C.L., 1991, Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm, Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA, pp. 450-457.

[6] Kosko, B., 1992, Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, Englewood Cliffs, NJ 07632

[7] Lee, C.C., 1990, Fuzzy logic in control systems: fuzzy logic controller-Part I+II, IEEE Trans. on Systems, Man & Cybernetics, Vol. 20, no. 2.

[8] Nomura, H., Hayashi, I., and Wakami, N., 1992, A Self-Tuning Method of Fuzzy Reasoning by Genetic Algorithm, Proceedings of the International Fuzzy Systems and Intelligent Control Conference (IFSICC '92), pp. 236-245, Louisville, KY.

[9] Schappell R.T., Vandenberg F.A., Hughes C.A., 1979, Study of Automated Rendezvous and Docking Technology, final report of contract JPL-955363, October .

[10] Teitz J.C., Development of an Autonomous Video Rendezvous and Docking System,1981,1982,1984, Phase 1,2, 3 final report of contract NAS8-34679, January.

[11] Thrift, P., 1991, Fuzzy Logic Synthesis with Genetic Algorithms, Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA, pp. 509-513.

[12] Zadeh L.A., 1975, The concept of a linguistic variable and its application to approximate reasoning I,II,III, Informat. Sci. vol 8, pp. 199-251, pp. 301-357; vol.9, pp. 43-80.

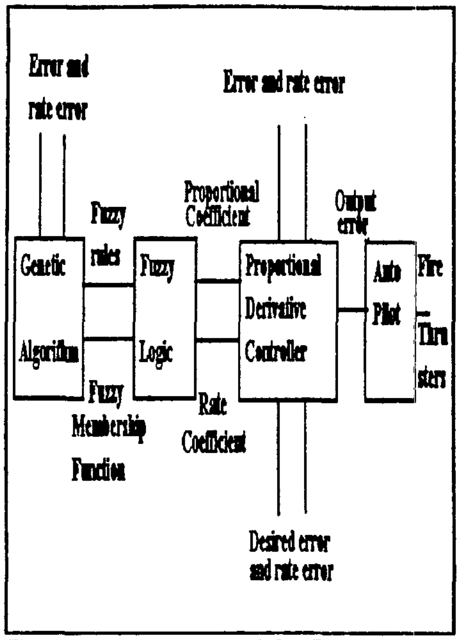

Fig 1. Block Diagram of GA-Fuzzy Proportional Derivative Controller

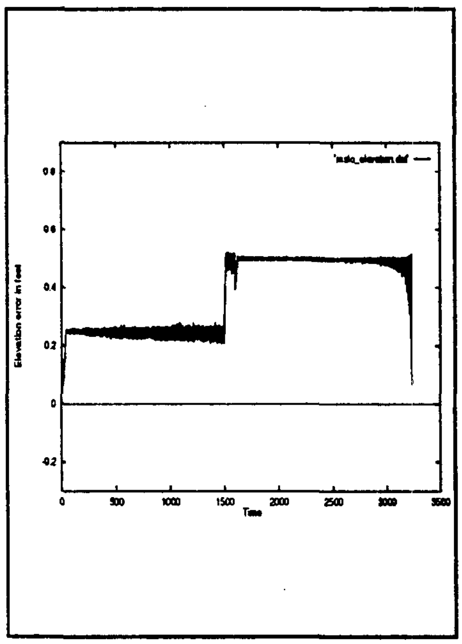

Fig. 2 Performance Of Marshall Space Proportional Derivative Controller For Elevation Control

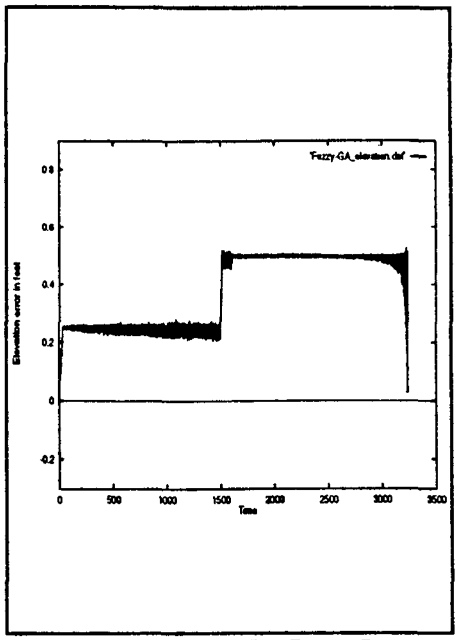

Fig. 3 Performance Of GA-Fuzzy Controller For Elevation Control

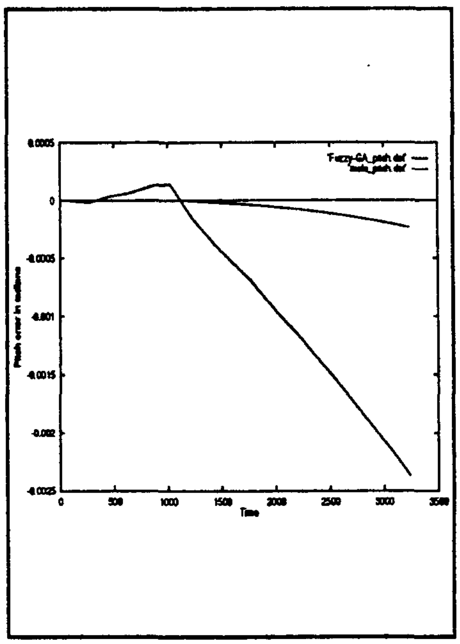

Fig. 4 Performance Comparison Of GA-Fuzzy Controller With Marshall Space Proportional Derivative Controller For Pitch Control