# Developing Automated Helicopter Models Using Simulated Annealing and Genetic Search

**Namir Aldawoodi**
PhD Candidate / Computer Engineer.
University of South Florida
Tampa, FL 33620

**Dr. Rafael Perez**
Professor & Graduate Program Director Computer Science Dept.
University of South Florida
Tampa, FL 33620

**Wendy Alvis**
Grad Student /Electrical Engineering Dept.
University of South Florida
Tampa, FL 33620

**Dr. Kimon Valavanis**
Deputy Director, CRASAR
Professor Computer Science Dept.
University of South Florida
Tampa, FL 33620

## Abstract

A heuristic technique is presented that applies simulated annealing search to derive mathematical equations that model a pilot for an X-CELL 60 helicopter. The technique uses a pre-defined alphabet of formulas and combines them to create a mathematical model of the system controller or pilot.

The proposed technique provides a new tool that can be used to develop an accurate helicopter pilot algorithm that can follow a trajectory – The method proposed generates accurate pilot formulas that are relatively easy to create. The proposed technique requires data that describes the actions taken by a pilot flying the helicopter under different conditions. This data was collected from a test a Proportional Integral Derivative (PID) controller (Simulating actions of a pilot on a change in altitude) against which the generated pilot algorithm was compared. The PID implantation simulates a portion of the flight of a helicopter model. Test results in a simulated environment show that the pilot formulas created using this method performed within 1% of the PID controllers.

The method proposed also addresses some shortcomings in controls development. It can be used to derive an accurate plant mode when none exist, this is especially useful if little is known about the internal function of the plant and second, the proposed method can be used to model the controller itself when the system is nonlinear and thus can not be modeled well with a PID. One such area where this method has great potential is in modeling the response of a human pilot flying a helicopter. A human response to the environment cannot be duplicated within a mathematical function that is normally used with PIDs – this is due to the limited range of these functions.

## 1 INTRODUCTION

The field of Autonomous Vehicle research and robotics is very active – this is due to the ability of Autonomous Vehicles to carry out Dirty, Dull and Dangerous work [1]. There are numerous applications where an automated vehicle is desirable. One such application is reconnaissance where automated pilot-less vehicles are desirable. The use of these crafts eliminates human-error due to fatigue and results in longer, more efficient missions. Another advantage lies in the fact that an automated craft is replaceable, a live pilot is not.

Automated vehicles rely on controllers to carry out tasks that the vehicle is asked to complete. For example a hover controller is tasked with keeping a helicopter in the same coordinates within an acceptable error margin (5 feet for example) – so long as the helicopter is within that 5-foot sphere radius then the module is working well. Many approaches can be used to construct a module and those include PID controllers, Neural Networks, Fuzzy Logic sets among others. Each of these controllers has their own specific limitations.

The method proposed would offer an alternative to modeling a controller or a helicopter (plant) when other approaches fall short or when an alternate model is needed to compare experimental results against. This method provides a mathematical formula from which analysis can be conducted or a model can be built or simulated. In other words, the method presented can apply as a primary algorithm to model a system or as a research tool to come up with alternate models to the same system.

## 2 BACKGROUND

Compared to regular airplanes, helicopters have many drawbacks. They are costly, slow, have a low payload capacity, and poor fuel efficiency. However, a helicopter offers two main advantages: first, the ability to land anywhere and; second, the ability to fly at a low altitudes very slowly (hover). Unlike airplanes, commercial autopilot technology for helicopters is limited to level flight for the most part; taking off and landing still pose a challenge. Currently, commercial helicopter autopilots can cost upwards of $250,000. There is no inexpensive commercial autopilot software that can be used for a small kit or pilot-less helicopters so research on automated helicopters may yield new and more sophisticated autopilot modules that can rival that of fixed wing aircraft.

Pilot-less helicopters are in use today; however, they are usually flown using a remote interface. The pilot would operate the craft from afar. There are two types of pilot less control, remote control and teleoperation. With remote control the plane is visible to the pilot. This is usually the case with kit planes that are flown in a limited area and are always visible to the operator. Teleoperation is where the plane is not visible to the pilot. In this case sensors on the plane (usually a camera) send information to the pilot (and crew) on the ground through an interface. This interface usually requires a team to run and also requires a large amount of bandwidth to relay data to the aircraft in a timely manner. A failure at the command center or anywhere in the communication chain can lead to loss of control and the aircraft.

Remote controlled autonomous vehicles have had reasonable success, and many have autopilot as a backup system should they lose communication with ground control. Hence, a helicopter with a fully automated controller offers many advantages amongst them the ability to fly for long periods of time without human intervention, the requirement for a team to run the autonomous aircraft can be eliminated and should human intervention be required, the operator is free to focus on more important tasks such as analyzing data collected by a camera or other sensors [1][5].

A completely autonomous helicopter may have been an ambitious project only a few years ago – current advances including GPS technology, advanced sensor suites and ultra fast and reliable hardware makes it possible to pack more power than ever before into a vehicle. Yet the challenge lies not in computing power, but more with information analysis. Parsing information and selecting relevant data is still difficult [2].

### 2.1 DESIGN CHALLANGES

One of the main difficulties in developing automated controllers for helicopters has to do with the complexity of the craft [8][9]. A helicopter model, unlike a conventional plane, has 6 degrees of freedom and a complex mechanical system to control its ascent and decent. More over, many of the formulas that govern the physical dynamics of a helicopter are interdependent and non-linear. Generally, there are 4 phases of helicopter flight that a viable controller design needs to consider: (1) Ascent – taking off and reaching a predetermined altitude. (2) Cruising – this is where the helicopter is on level flight covering a predetermined path. (3) Hovering – this is where the helicopter has to maintain a constant height with as little forward velocity as possible and finally, decent, this is where the helicopter must land at a controlled rate in order to prevent damage to the aircraft.

### 2.2 BACKGROUND OF AUTOMATED HELICOPTERS

Research into autonomous helicopter controllers started in the late 1980’s at the University of Tokyo – that controller model was based on fuzzy logic and was tested on a Yamaha R-50 agricultural-work helicopter. The goal of the project was to design a controller without having to worry about the modeling and dynamics of the helicopter. After several years of research, the helicopter was still limited in what it could do autonomously. In 1991, the Aeronautics Directorate of NASA Ames Research Center became involved in researching autonomous helicopters but used the "dynamical modeling and identification" approach. The Robotics Institute of Carnegie Mellon University (CMU) also started work on its own (independent) flight project using an R-50 helicopter.

Later research at both Carnegie Mellon University and MIT began to focus on frequency responses modeling methods in order to develop a linearized mathematical model of the craft. In 2001 Gavrilets, et al, [9] published a paper introducing a partially linearized model of a X-Cell-60-SE helicopter that had been developed by using frequency response methods in order to derive several of the plant parameters for hover and slow flight conditions. This mathematical model was built in Simulink and results derived from it were used in this paper.

In 2003, Mettler published a book introducing a fully linearized model of both the Yamaha R-50 and the X-Cell. This model is linearized around three operating points, hover flight, slow flight and cruise flight. Frequency response methods were used in this development as well, but this time were implemented with CIFER, Comprehensive Identification from

Frequency Responses. CIFER was developed by NASA and is software that extracts a mathematical description of a vehicle from test data using frequency response modeled from actual flight data. CIFER works much like reverse-simulation. Simulation requires that assumptions be made ahead of time (a-priori) to allow for the derivation of equations related to the model, on the other hand, system identification starts with measured vehicle motion and measured responses in order to develop a model that reflects the measured data accurately. CIFER allows designers to skip modeling and get straight to a model that reflects the data collected from a real system. The result was a linearized helicopter model derived from an inherently nonlinear system. A Matlab model of the information presented in [10] is presently being developed and will be used for future research.

### 2.3 CURRENT RESEARCH METHODS

Autonomous helicopters have been designed using many approaches – some of the more common methodology includes the following: Proportional Integral Derivative (PID) Controllers, Fuzzy Logic Controllers, Neural Networks and Genetic Algorithms. PID controllers are advantageous in the sense that they are relatively easy to implement and simulate in software and have simple tuning procedures where the parameters are tuned until the desired performance is obtained. PIDs also offer an advantage when it comes to computational time. This is an important consideration when a robot must make a move to adjust its flight pattern. A slow response can make the difference between a machine that flies successfully and one that runs into the ground. The disadvantage of PIDs is that, without an accurate plant model, it takes trial and error to set the parameters correctly. In cases where a plant model is complex one then a model may be derived using step functions. However, if the plant cannot be modeled accurately the model will not perform well. In addition, if the control mechanism requires a nonlinear output then the PID controller will not control the plant. Hence PID controllers are a good fit to linear models or as stabilization controllers within other non-linear controllers.

Neural Networks have also been used to develop autonomous controllers. Neural Networks, unlike PIDs, are very good at dealing with nonlinear data, once trained they usually perform well provided that the training set was well selected [6]. However, the disadvantage is that it is difficult to convert a Neural Network to an equation for study and analysis. Also, if an equation is derived it is difficult to get it in a desired format [12]. This becomes necessary when modeling a plant where response time can be an issue and the programmer is never sure what features of the data the Neural Network has really learned. Hence, an algorithm might appear to have mastered the data only to fail later due to the Neural Network learning features that are relevant to the training set (but not the test set).

Fuzzy Logic based autonomous controllers have also been used, fuzzy logic offers advantages in that it can use linguistic variables, allows imprecision/contradiction in input data, conflicting objectives are reconciled, base rule (fuzzy sets) can be modified with ease and is easy to implement [4]. In addition fuzzy logic handles non-linear cases well and produces a smooth response. However, there are drawbacks such as difficulties in programming and difficulties in modeling with analytical methods; fuzzy sets have imprecise boundaries, results can be unexpected and hard to debug, they sometimes post additional computational load. There are reports of fuzzy logic based controllers (in industry) that did not work as well as a good PID controller.

On the other hand, genetic algorithms and simulated annealing offer advantages such as robustness to noise and independence of discontinuities in the data. Also they are not susceptible to getting stuck in local maximas [7] [13].

Also, the proposed method is easier and less expensive to implement than CIFER; where CIFER requires a proprietary software package and requires that information be recorded from several actual flights in order to collect enough data for the frequency response models to be constructed. Once the frequency response models are collected, the linear model can then be calculated using the software. Hence CIFER is costly and requires a large data set otherwise the system cannot be molded accurately. In addition, CIFER is complex and entails specific training to become familiar with the product and requires operator experience in order to be applied effectively. All of these facts indicate that CIFER can be an expensive and time intensive process.

## 3 PROBLEM DOMAIN

The goal of this paper is to study the feasibility of using the Advanced Formula Prediction algorithm to model the actions of a pilot. This algorithm searches for a formula from a set of mathematical primitives using genetic algorithms and simulated annealing [3]. In modeling a plant or the pilot the resulting functions (models) should have the capability of accurately reproducing the plant or pilot when provided with a set of input and output data collected from said system. The benchmark used in this paper is a straightforward PID controller that simulates a pilot acting on a plant (a Yamaha X-CELL 60 remote controlled helicopter). This controller simulates the helicopter undergoing a change in height and then maintaining a hover. Cruise flight and landing controllers are currently being developed and are not ready to benchmark against at this time. The PID controller (simulating the actions of a pilot) was tuned using trial and error until the model underwent a change in height of 10 meters and then stabilized. The PID was chosen because it offered a fast and efficient way to simulate a takeoff and the plant was modeled using semi-linear models. Figure 1 demonstrates a block diagram of this system in which the PID is simulating the function of a pilot controlling the helicopter directly. This setup will simulate a change in height and hovering at a fixed altitude.

While the PID controller is straightforward to implement, the goal of this paper is to test the application of a genetic search based algorithm to model a human pilot simulated by the PID. Once this is established, it is the intentions of the authors to further this research by deriving a controller for flight data collected directly from a human pilot following a trajectory that is much more complex than staking off and hovering. It is also important to mention, that this simulated helicopter model does not take into account ground affects on take-offs. That will be a part of future work. Also, we will further study the ability of the Simulated Annealing based search algorithm to model a plant that that is difficult to model using traditional approaches (such as step models). Mathematics behind the helicopter can be very complicated and nonlinear – hence the research done by Mettler using CIFER software to derive a linear model was a break-through. It allows researchers to derive new equations (linear for the most part) that can describe the helicopter model in a simpler fashion (mathematically). This linearization on frequency response formulas is the basis for the PID controller and much research in autonomous helicopters.

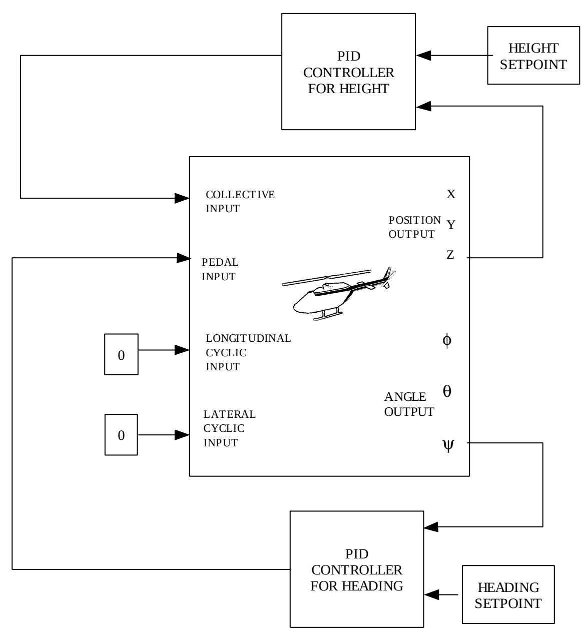



Figure 1: A PID controlling a helicopter plant.

### 3.1 DATA COLLECTION

The data collected represents a set of input and output values from a simulated human pilot flying an X-CELL 60 helicopter, as displayed in Figure 2. In this example, a pilot controlling an interface is shown to the left and the helicopter (plant) is shown to the right. Data is collected from the input and the output of the plant

The PID used has already been tuned and returns accurate data that simulate the actions of a pilot implementing a change in height and then holding the helicopter there. When the model runs, data is collected that show the input to the plant (collective control and rudder control, height and coordinates) and the output from the plan being height and power to that helicopters' main rotor.

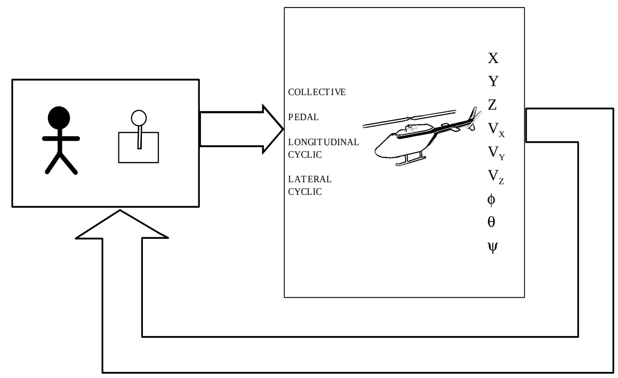



Figure 2: Collecting Data with a pilot.

A PID works well as a controller but can prove cumbersome in determining an exact setting of the parameters. Another approach is to evolve a pilot to control the plant. This technique has advantages in that little knowledge or no knowledge is needed about the plant other that input and output data. The method collects the data by watching another controller (a live pilot for example) run the plant and then studying the data to determine if that control behavior can be reproduced using the mathematical primitives that it has at its disposal.

By using flight data from a PID, a controller (representing the actions of a pilot) generates flight data from which a new mathematical model will be derived based solely on the data and not on the characteristics of the plant. It is assumed that the data collected represents a good representative sample. The resulting formula set will be compared to the PID to see how well the new function simulates a controller it knows nothing about other than what it has seen through observation.

A team at the Department of Electrical Engineering at the University of South Florida developed the PID used in this paper (some members of the team are listed as co-authors is this paper). This PID was developed and tuned to handle a pre-determined change in height and then to hold (hover) the helicopter once the desired height is achieved. It performs well in Matlab simulations and is currently being modified to undergo field testing. Neither model will consider path taken or obstacle avoidance at this time. Again, the algorithm presented is uses simulated annealing and genetic search methods to generate formulas that fit the data set by searching for patterns inherent in the data itself. The newly generated hypothesis equation is then tested to see how well it fits the test data.

### 3.2 ADVANTAGES OVER OTHER PATTERN MATCHING TECHNIUES

There are methods available to search for patterns in data. Such as regression analysis, which can be used for both linear and nonlinear systems. There are some problem classes that have presented a challenge to statistical regression analysis. These include problems in which there is little or no information about the function that generated data. In these cases, researchers typically use neural networks to learn more about the domain space of the function. Once the domain is learned then regression techniques can be applied more effectively.

The goal of a search algorithm is to learn something about the data set, detect patterns and then generalize what was learned into a set of rules that when applied can generate a function. Hence, a good learning algorithm has the ability to predict future values from past experience. The better the learning algorithm the less the error in future predictions provided the data sample is well distributed. In order to successfully classify data the following should be considered: (1) Data: Is it relevant to base a solution on? Is it noisy? Is distributed well? Is it a representative sample – where a representative sample is the minimum amount of data needed to represent the function? (2) Assumptions: Are there domain specific assumptions that must be taken into account? (3) Output form: What is the most useful output format? Will this format allow for further analysis using tools not related to the algorithm that generated the output? (4) Evaluation: Is the algorithm accurate? What is the error tolerance or, what is an acceptable level of error?

Learning based on past knowledge falls under the category of supervised learning. This type of learning assumes that observed data values and their corresponding output values are provided in the beginning of the search. The aim is to approximate the original function with a function that has an acceptable error level.

## 4 WHY SIMULATED ANNEALING AND GENETIC SEARCH

After considering the data collected and the goal of the experiment (to model a plant or controller) the choice of search strategy was narrowed down as follows: First classical programming techniques (if then else) where not a good fit for deriving a set of equations from data. This is due to the fact that no assumptions should be made as to the shape of the sample data, and if assumptions were made then the algorithm could be limited to considering solutions that were programmed for ahead of time.

For example, it would be difficult to write a voice recognition program using a classical if-then-else approach, because the program must make assumptions ahead of time as to what patterns map to a given word. This is made even more complicated by the fact that different people have a different way of pronouncing a spoken word.

A more effective approach for this type of problem would be with the use of a neural network that learns from the data directly. An effective neural net implementation should be able to generalize (given enough samples) the difference between two words that may sound very similar – for example: (her and here). Thus, in learning directly from data, some approaches tend to be a natural fit. Since the goal of this research is to learn directly from data, some of the methodologies that we considered include decision trees, genetic algorithms, regression splines and clustering.

In order to effectively produce a readily configurable function and return an answer made up of pre-selected mathematical primitives, a more direct approach was needed. A neural network was not chosen because it does not offer a practical way to back-solve the solution that it returns into a function (with an alphabet of our choosing).

Decision trees were also considered because of their relative speed (faster than version spaces) for a large concept space and disjunction easier to carry out; however, they were not considered flexible enough to produce the function formats required and would be too complex to adapt them for use in building formulas that fit a set of data. Also, a decision tree may not always expain its classification clearly. Another consideration was statistical (regression based approaches) – however, these do not do well when the final form of the function is not known and there is limited flexibility as to the mathematical primitives that can be used.

On the other hand, Simulated Annealing and Genetic search techniques were a good fit for many reasons; first, the way they search is not mathematically based. The latter means that no direct calculations on the data would be performed. As such, data discontinuities, noise and data inconsistencies would have little effect on the search strategy. Second, they search for a solution independently of what the data looks like. Hence, there is no significant bias in relation to what the data looks like and the algorithm is free to look for any pattern hidden within the data. Granted, sometimes a search bias can have advantages. For example if a researcher knows that the function that generated the data contains a given set of primitives. The algorithm can then be biased, through the selection of the alphabet, to search in that solution space. Even if it's a hunch that the function contained a certain mathematical operation, that hunch can be verified by applying the algorithm over and over using different sets of mathematical primitives.

Simulated annealing and genetic search strategies are also resistant to getting stuck in local maxiams [11] – a big advantage when searching for patterns within a dataset.

As such, the algorithm was implemented with simulated annealing and genetic search to look for a set of mathematical primitives that when combined would result in a function that maps the input data to the output data with as little error as possible.

### 4.1 THE SEARCH DOMAIN

The proposed algorithm manipulates the data directly and derives a solution based on the data alone. As such the algorithm does not need to know how a pilot thinks or how a PID controller (simulating the pilot) works internally – what is of importance is collecting input and output data. Figure 3 shows a system where input and output data values are known, but little is known as to how these values are being generated. For sufficiently large number of variables this problem can be NP hard.

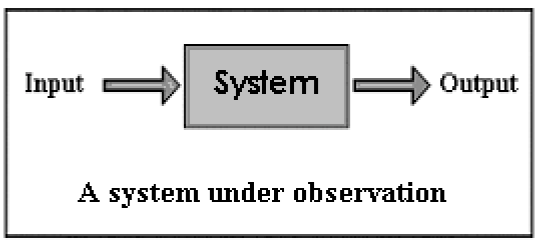



Figure 3: Collecting data from a system.

A search of the literature found no method that generates formulas the way proposed in this paper. There is one method, however, that has some parallels with the algorithm that we propose – that method is CIFER. However, CIFER works by measuring frequency response while the proposed method searches for combinations of mathematical primitives that model the data set. They both offer a way to derive a formula directly from data but they take a different path in generating the formulas.

## 5 THE PROPOSED METHOD

The technique described in this paper generates mathematical formulas from sets that hold mathematical primitives using genetic algorithms and simulated annealing. This method can be classified as a supervised regression learning approach that uses batch data. This paper investigates and quantifies the ability of genetic algorithms and simulated annealing to find formulas that describe the relationship among a set of observed data with little or no knowledge of the problem domain.

Genetic search techniques were selected because they offer advantages such as not getting stuck in local maxima by searching in parallel from a number of different solutions and they are not hindered by discontinuities in solutions expressed by mathematical formulas since they directly manipulate a string representing those formulas [11][12]. Likewise, simulated annealing is a robust searching method with similar properties [13]. This is especially useful when searching large, complex domains: the type that we expect to encounter when searching for unknown functions.

Thus, the proposed method would search for patterns in data and generate an estimator function. Performance is also important – both in terms of the time it takes to search and of the accuracy of the prediction. The way a search is conducted is also important because it has an impact on how quickly the search converges. The method proposed does not use statistical techniques – rather it uses genetic algorithms and simulated annealing because these methods may find patterns in the data that regression may miss due to bias built into the regression methodology itself.

However, simulated annealing and genetic search are not perfect and there are disadvantages in using them; mainly, the lack of absolute precision. These search methods are also not very efficient – especially if the data is well defined and the model that generated the data is well understood – in these instances regression analysis would probably be a better tool.

### 5.1 ADVANTAGES OF PROPOSED METHOD

The proposed method uses genetic search and simulated annealing to assemble a formula out of an alphabet of primitives supplied by the programmer. Since data may be improperly fitted by any one search strategy, the use of two search strategies to generate functions provides a better overall data fit. The algorithm returns a unique combination each time it is ran. Naturally, that depends on the number of maximas in the data. The algorithm proposed provides a versatile and powerful tool that can be used in investigating, verifying and analyzing problems where the relationship between the independent and dependent variables is not well understood or known. The results of this algorithm may be used to derive a function for a PID controller to use.

Another consideration was the calculation of error. The authors opted for a variation of MAPE to calculate error. While it is true that the method used to calculate error could bias the search one way or another – however, overall we felt that the error calculation method selected works well for the problem that we are solving.

## 6 THE ALGORITHM

The algorithm was designed to return a solution that use pre-provided mathematical primitives. The returned function must meet the minimum error required. This indicated by a fitness measure that shows how well the generated function approximates the observed data. The search is halted once the function maps the data within a predefined error level or when the allotted search time has expired.

### 6.1 DATA FORMAT

Given a system under observation, inputs to that system will be represented by the vector S and observed outputs of the system will be represented by the vector T. Figure 4 demonstrates a system with an observed input vector Si, and an observed output vector Ti. S and T cannot be empty.

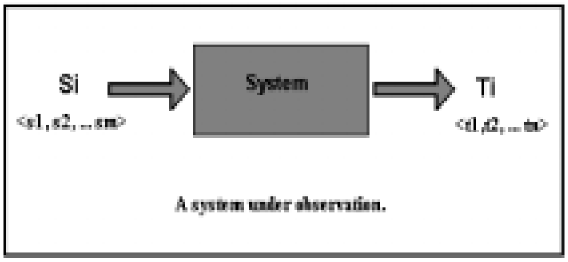



Figure 4: Inputs and outputs to a system

Each input vector **S** must have a corresponding output vector **T**. When collecting data from a system, a collection of input vectors is entered and a collection of output vectors is observed. The algorithm would take the **S** and **T** vectors and produce some function **F** that relates all the input vectors (**S1, S2 S3, … , SN)** to their respective output vectors (**T1, T2, T3, … , TN)** such that***: Tij = Fj(Si())***

where: 1 <= j <= n. and n is the size of **T**.

The average error rate e is calculated as shown in figure 5.

$$\text{Error} = \frac{\prod_{i=1}^{N} \left( \frac{\text{ABS } |F(Si()) - Tij| * 100}{\text{ABS } |Tij|} \right)}{N}$$

m = Number of elements in S    where 1 =< j =< n
n = Number of elements in T
N = Number of input/output sets    ABS = Absolute Value

Figure 5: Error Calculation

Note that the error calculation is applied to one output value at a time. Since **T** holds all the output values for a system, the algorithm produces a separate formula for each element of **T**. The output under consideration (represented by i) is selected from each output vector **T** and considered.

### 6.2 THE SOLUTION FORM

The algorithm generates formulas using the format shown below. This format is flexible and allows for fine-tuning of the returned function. The algorithm returns a solution that contains the same number of variables as the size of the input vector **S**. FPEG takes the members of the input vector **S** and applies to each member the operation, multiplier and property operators. The property operators are variables that may hold any of a number of different mathematical primitives. The values for the operators are derived from sets that will be referred to as the alphabet sets. Once the first set of operators is applied, the resulting (modified) variables are combined using the combination operator. When all these calculations are completed, a delta value is added or subtracted to the resulting sum. The final result is referred to as the calculated value. This technique searches within all the possible combinations of the above operators to find the best combination, that when applied to the input values in **S**, will yield a calculated value that has as little error as possible when compared to the actual value. If the algorithm were to search every combination then the problem would be unmanageable for large alphabets and/or large input sets. Genetic and simulated annealing search methodologies excel in searching large problem domain for combinations that produce the best fit.

### 6.3 OPERATOR DETAILS

The algorithm is flexible because the formula can be adjusted and changed as needed. Also the alphabet sets can contain any mathematical primitive or function and, as such, the length of the sets themselves can be varied. In the next section we will discuss the alphabet values that were chosen for this paper

### 6.4 CALCULATING THE FITNESS VALUE

The application of these operators to the vector **S** of input variables transforms these variables into a resulting calculated value. This resulting value is compared with the original output value stored in the corresponding result set **T**. As shown in figure 3 earlier, for some pair **Si():Tij**, the error value is calculated by comparing the calculated value with the actual value. The error value shows how close the function built by our algorithm comes to approximating the supplied data. The fitness value is defined as the average of all the error values for a given output. The resulting error value is a percentage that shows the output deviation of the generated function in relationship to where the output value should be (as provided by the data). The returned error value is referred to as the fitness of the function. The lower the value the better the overall fit of the generated function. Hence, fitness values have a range of 0.00 % error (indicating a perfect match) to large numbers (20,000 % for example) – the latter indicates a poor fit.

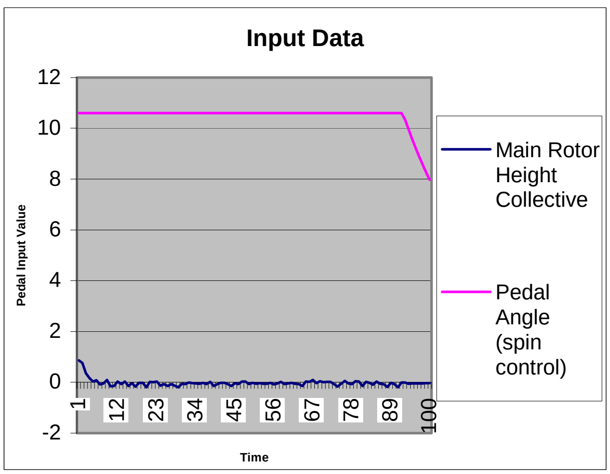
# 7 TESTING METHODOLOGY

## 7.1 TEST GOALS

The aim of this algorithm is to derive functions from a given set of data. We wanted to verify that the proposed method would be able to generate function that behaves much like a helicopter pilot (modeled by a PID controller) would.

## 7.2 INPUT DATA

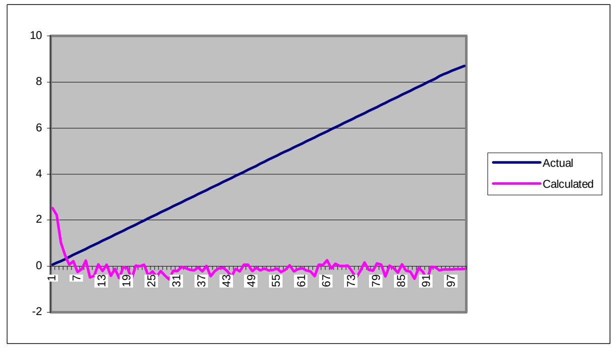
The graph below represents the input parameters collected over 4000 time slices and normalized to fit within 100 samples. They demonstrate that in order to achieve a height change the ‘Main Rotor Height Collective’ must be kept at a certain power setting and then reduced as the desired height is approached. The ‘Pedal Angle Spin Control’ oscillates to compensate for the plane’s tendency to spin left or right. This oscillation reflects the pilot’s efforts in keeping the plane stable. This data is shows in Graph 1.



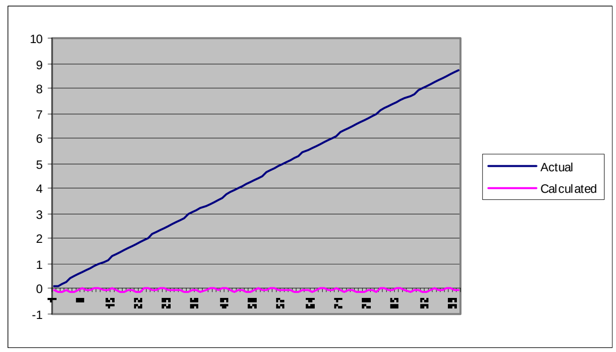

Graph 1: Input Data Plot showing Collective and Pedal Angle as time passes

The next four graphs (Graphs 2-5) show one of the actual output values (height) of the plant (helicopter) plotted against the predicted value generated by the hypothesis function. It is important to note that the predicted value starts out with a large error and then as the annealing temperature approaches 0 the generated function gets better and better at approximating the output. The annealing temperature used in the run that generated the graphs was 5000.

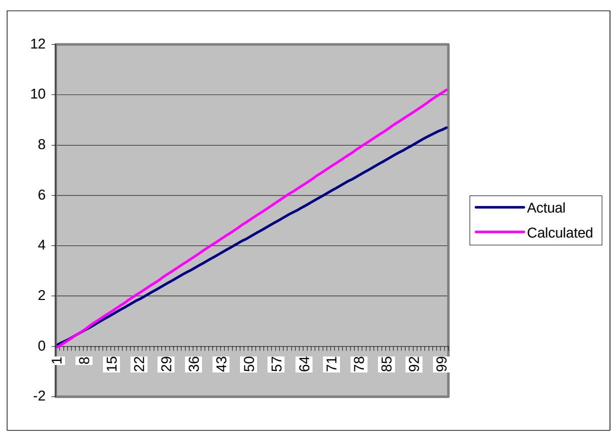
Note that although the output looks relatively simple – after all it is a semi-linear line – the true difficulty was not in modeling the line but rather associating inputs that stay flat (no value change) for a significant period of time to the output shown. None the less, the algorithm was able to map things perfectly well.



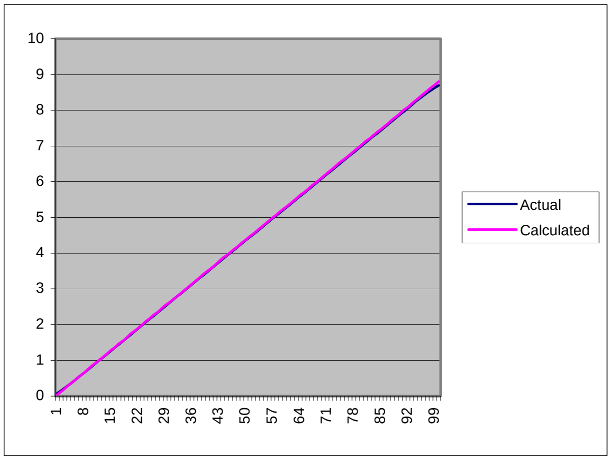

Graph 2: Actual vs. Function-Generated value – Annealing Temperature = 4500.




Graph 3: Actual vs. Function-Generated value – Annealing Temperature = 3000.




Graph 4: Actual vs. Function-Generated value – Annealing Temperature = 1500.




Graph 5: Actual vs. Function-Generated value – Annealing Temperature = 0.

The table below shows data collected over 50 runs – each run set covers 10 runs – the run set is indicated and the starting temperature is noted. Also, the number of layers is shown – the higher the layer number the more complex the function – the latter means more expensive computational time but a more powerful function. The performance average for each set of 10 runs is shown in the rightmost columns – with a best average set performance of 0.45% and a worst performance of 3.25 %.

Table 1: Test results for benchmarks

| **Run Set** | **Temperatu** | **Layers** | **Performance** |
|---|---|---|---|
| 1 | 3,000 | 2 | 3.25% |
| 2 | 4,000 | 2 | 3.02% |
| 3 | 5,000 | 3 | 0.55% |
| 4 | 6,000 | 3 | 0.86% |
| 5 | 10,000 | 4 | 0.45% |

# 8 CONCLUSIONS

We have demonstrated a new tool based on genetic-search that can be used to construct mathematical formulas from datasets collected from a system under observation. The derived functions model the actions of a pilot (simulated by a well tuned PID) very well. The results obtained demonstrate that the algorithm is capable of generating results that archive an accuracy level of 97% to 99% or higher. This tool has more than one application. First, it can be used to model a pilot flying a helicopter; next, it can also be used to model the plant (for cases where models of the plan can not be easily obtained). The technique can also be used to derive equivalent alternate methods to a controller to contrast against an existing controller design.

The method proposed generates accurate pilot formulas that are relatively easy to generate. The next step will be to model a more complex helicopter model. There is also new data collected from a live pilot flying the Simulink model of the R-50 helicopter via a USB port and the model helicopter controller– this data is interesting in the sense that the pilot was asked to take off and follow a path without any other information. It is the authors' intentions to use this new test data to create a more versatile controller. The authors also propose using recorded flight data from the helicopter itself to demonstrate that the methods in this paper have the ability to develop an accurate mathematical model of the helicopter with an easy to implement fashion.

## References

[1] R. Murphy (2000). Introduction to AI Robotics, The MIT Press.

[2] P. Spanoudakis, N. C. Tsourveloudis, K. P. Valavanis, "Technical Design Specifications for a Prototype Unmanned VTOL Vehicle", IEEE Transactions on SMC: Part B, (submitted).

[3] N. Aldawoodi, R. Perez (2003), Formula Prediction Using Genetic Algorithms, University of South Florida, Gecco AI Conferece

[4] L. Doitsidis, K. P. Valavanis, N. C. Tsourveloudis, M. Konte, "A Fuzzy Logic Based Control Architecture for Autonomous Navigation and Collision Avoidance of Skid-Steering Mobile Robots" Journal of Autonomous Robots (submitted).

[5] Srikanth Saripalli, Gaurav S. Sukhatme, and James F. Montgomery, An Experimental Study of the Autonomous Helicopter Landing Problem, Department of Computer Science, University of Southern California,Los Angeles, California, USA

[6] Q. Wang, T. Aoyama (2001), A Neural Network Solver for Differential Equations, Miyazaki University, Japan.

[7] T. Mitchell, (1997), Machine Learning, WBC/McGraw-Hill.

[8] H. Shim, TJ. Koo, F. Homann (2001), A Comprehensive Study of Control Design for an Autonomous Helicopter Robotics and Intelligent Machines, University of California at Berkeley.

[9] V. Gavrilets, B. Mettler and E. Feron, Nonlinear Model for a Small-Size Acrobatic Helicopter, AIAA Guidance, Navigation, and Control Conference and Exhibit, August, 2001.

[10] Bernard Mettler, Identification Modeling and Characterization of Miniature Rotorcraft, Kluwer Academic Publishers, 2003

[11] Q. Wang, T. Aoyama (2001), A Neural Network Solver for Differential Equations, Miyazaki University, Japan.

[12] Lawrence Davis, (1990), Genetic Algorithms and Simulated Annealing, Pitman, London.

[13] P. J. M. van Laarhoven and E. H. L. Aarts, (1987), Simulated Annealing Theory and Applications.