

Automated Design of Robust PID Controller using Genetic Algorithm

Patel Iftekar

Department of Electronics
Engineering
PIIT, New Panvel.

Monika Bhagwat

Department of Electronics
Engineering
PIIT, New Panvel.

Ujwal Harode

Department of Electronics
Engineering
PIIT, New Panvel.

ABSTRACT

The paper deals with the design of robust controller for uncertain SISO systems using Quantitative Feedback Theory (QFT) and optimization of controller being done with the help of Genetic Algorithm (GA). Quantitative Feedback Theory (QFT) technique is a robust control design based on frequency domain methodology. It is useful for practical design of feedback system in ensuring plant's stability by reducing the sensitivity to parameter variation and attenuates the effect of disturbances. Parameter variation or physical changes to the plant is taken into account in the QFT controller's design.

Quantitative Feedback Theory (QFT) can provide robust control for the plant with large uncertainties. The manual design with the help of QFT toolbox in Matlab is complicated and even unsolvable. The existing automatic design methods are limited in optimization. Based on the genetic algorithm (GA), a more effective automatic design methodology of QFT robust controller is proposed. Some new optimization indexes like IAE, ISE, ITAE and MSE are adopted, so the design method is more mature. To obtain good performance of the controller in a relatively short time, the manual design and the automatic design are combined. Compared with the results from the manual design method, the performance of the QFT controller based on genetic algorithms is better and the efficiency of searching scheme is the best. An illustrative example which compares manual loop shaping with automatic loop shaping is given.

Index Terms

Robust controller, Quantitative Feedback Theory (QFT), Genetic Algorithm (GA), Uncertainties, stability, Manual loop shaping, Automatic loop shaping, Optimization index, Matlab, Mean of the Squared Error (MSE), Integral of Time multiplied by Absolute Error (ITAE), Integral of Absolute Magnitude of the Error (IAE), Integral of the Squared Error (ISE).

1. INTRODUCTION

There are two general control systems. (i) Conventional control, and (ii) Robust control, which typically involves worst case design approach for family of plants using a single fixed controller [1]. If the design performs well for substantial variations in the dynamics of the plant from the design values, then the design is robust. Robust control deals explicitly with uncertainty in its approach to controller design. Controllers designed using robust control methods tend to be able to cope with small differences between the true system and the nominal model used for design. [2]

Quantitative Feedback Theory (QFT) is robust control method which deals with the effects of uncertainty systematically. QFT is a graphical loop shaping procedure used for the control design

of either SISO (Single Input Single Output) or MIMO (Multiple Input Multiple Output) uncertain systems including the nonlinear and time varying cases. QFT, developed by Isaac Horowitz is a frequency domain technique utilizing the Nichols chart (NC) in order to achieve a desired robust design over a specified region of plant uncertainty. In comparison to other robust control methods, QFT offers a number of advantages. For the purpose of QFT, the feedback system is normally described by the two-degrees-of freedom structure. [3]

A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism widely used in industrial control systems. PID controllers have dominated the process control industry over the decades owing to its associated simplicity and easiness in implementation. The design of PID controllers, tuning involves selecting the amounts of Proportional, Integral and Derivative components required at the output of the controller. Since the design of PID controllers involves obtaining the P, I and D components there always occurs a compromise in the design. The design of the optimum values for the PID controller parameters has always been challenging. Many new tuning techniques have been developed for the design of PID controllers, however then still exists a scope for better tuning method. Various control strategies are used for design of control system depending on plant model. [4]

One of the most frequently used on-line tuning methods is the Genetic Algorithm (GA). The genetic algorithm is very effective at finding optimal solutions to a variety of problems. This innovative technique performs especially well when solving complex problems because it doesn't impose many limitations of traditional techniques. Due to its evolutionary nature, a GA will search for solutions without regard to the specific inner working of the problem. This ability lets the same general purpose GA routine perform well on large, complex scheduling problems. There is now considerable evidence that genetic algorithms are useful for global function optimization.

2. LITERATURE REVIEW

Quantitative Feedback Theory (QFT) has been applied in many engineering systems successfully since it was developed by Horowitz. Several researchers have investigated the design automation of QFT.

In [7], Qingwei Wang, Zhenghua Liu, Lianjie Er presented automatic design method of QFT controller based on GA and proposed several initial optimization indexes to make the controller more practical. They proved that GA based automatic design method of QFT controller is very effective and optimization method can achieve both high performance and increased efficiency.

In [8], A Zolotas and G.D Halikias proposed an optimisation algorithm for designing PID controllers, which minimises the asymptotic open-loop gain of a system, subject to appropriate robust stability and performance QFT constraints. The algorithm is simple and can be used to automate the loop-shaping step of the QFT design procedure. The effectiveness of the method is illustrated with an example.

In [6], Wenhua Chen and Donald J. Balance addressed design of robust controllers for uncertain and non minimum phase and Unstable plant using QFT technique. Benchmark examples were used to illustrate the improved method

In [11], Mario Garcia-Sanz In the first part of the paper summarized the main concepts of the Quantitative Feedback Theory (QFT) and presented a wide set of references related to the principal areas of research. In the second part of the paper introduced a method to design non-diagonal QFT controllers for MIMO systems. Finally the paper ends presenting some real-world applications of the technique, carried out by the author: an industrial SCARA robot manipulator, a wastewater treatment plant of 5000 m³/hour, a variable speed pitch controlled multipolar wind turbine of 1.65 MW and an industrial furnace of 40 metres and 1 MW.

In [10], O. Yaniv and M. Nagurka examined an analytically-based algorithm for finding low order controllers that satisfy closed loop gain and phase margin constraints and a bound on the sensitivity.

In [14], K. Krishna Kumar and D. E. Goldberg proposed an approach which improves the quality and speed of manual loop shaping; Genetic Algorithm was used to provide global optimization.

In [16], P.S.V.NATRAJ Proposed an algorithm for generating QFT bounds to achieve robust tracking specifications. The proposed algorithm can generate tracking bounds over intervals of controller phase values, as opposed to discrete controller phase values in existing algorithms

3. BASICS OF PID CONTROL SYSTEM

A PID controller consists of the three terms: proportional (P) integral (I), and derivative (D). Its behaviour can be roughly interpreted as the sum of the three term actions The P term gives a rapid control response and a possible steady state error, the I term eliminates the steady state error and the D term improves the behaviour of the control system during transients.[4]

The Transfer function of PID controller is given by

$$T.F.1 = K_p * \left(\frac{T_i * T_d * s^2 + T_i * s + 1}{T_i * s} \right) \quad (1)$$

Where K_p = Gain of the system.

T_i = Integral time constant.

T_d = Derivative time constant

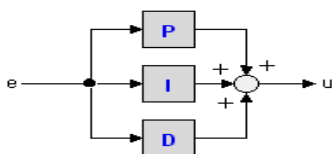


Fig 1: parallel form of PID

4. QFT BASIC CONCEPTS

QFT is a very powerful control system design method when the Plant parameters vary over a broad range of operating conditions. The basic concept of QFT is to define and take into account, along the control design process, the quantitative Relation between the amount of uncertainty to deal with and the amount of control effort to use. The Quantitative Feedback Theory (QFT) method offers, frequency-domain based design approach for tackling feedback control problems with robust performance objectives.[3] In this approach, the plant Dynamics may be described by frequency response data, or by a transfer function with mixed (parametric and nonparametric) Uncertainty models. The basic idea in QFT is to convert design specifications at closed loop and plant uncertainties into robust stability and performance bounds on open loop transmission of nominal system and then design controller by using loop shaping [5]. The two-degree-freedom feedback system configuration of QFT is given in Fig.1

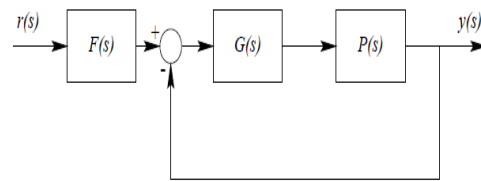


Fig 2 QFT Configuration

Where $G(s)$ -Controller

$F(s)$ -prefilter

$P(s)$ -uncertain plant which belongs to a given plant family 'P'.

5. CONTROLLER DESIGN PROCEDURE USING QFT

Design Steps for PID Controller are as follows [6].

1. Translation of TDS to FDS (Time Domain Specification to Frequency Domain Specification)
2. Generation of the Plant Set.
3. Generation of the Template.
4. Selection of Nominal Plant.
5. Generation of Stability Bounds.
6. Generation of Tracking Bounds.
7. Grouping of Bounds.
8. Intersections of Bounds.
9. Loop shaping (Design of a Controller).
10. Prefilter Designs.

5.1 Design Example

A Benchmark example from QFT toolbox is considered in order to explain the various design steps QFT controller.

Consider a plant with parametric uncertainty given by

$$P(s) = \frac{k}{s(s+a)} : k \in [1, 10], a \in [1, 10]$$

The stability index is given by

$$\left| \frac{PG}{1+PG}(j\omega) \right| \leq 1.2, \omega > 0,$$

And upper and lower boundaries of tracking index are,

$$T_U(\omega) \leq \left| F \frac{PG}{1+PG}(j\omega) \right| \leq T_L(\omega)$$

$$\text{Where } T_U(\omega) = \left| \frac{0.6854(j\omega+30)}{(j\omega)^2+4(j\omega)+19.752} \right| \text{ and}$$

$$T_L(\omega) = \left| \frac{120}{(j\omega)^2+17(j\omega)^2+828(j\omega)+120} \right|$$

And output disturbance rejection bound is given by

$$\left| \frac{Y}{D}(j\omega) \right| \leq \left| 0.02 \frac{(j\omega)^3 + 64(j\omega)^2 + 748(j\omega) + 2400}{(j\omega)^2 + 14.4(j\omega) + 169} \right|$$

5.2 Translation of TDS to FDS (Time Domain Specification to Frequency Domain Specification)

Since a benchmark example from QFT TOOLBOX is considered in this paper, the specifications are readily available in frequency domain.

5.3 Plant Templates

Templates are plot of magnitude verses phase of plant sets for various frequencies. Generation of templates at specified frequencies that pictorially describe region of plant parameter uncertainty on the Nichols chart defines the structured plant parameter uncertainty. The frequency array for plotting the templates must be chosen based on the performance bandwidth and shape of the templates. Margin bounds should be computed up to the frequency where the shape of the plant template becomes invariant to frequency. [5] Here, at approximately $\omega = 100$ rad/sec, the template's shape becomes fixed, a vertical line. Our array includes several frequencies from $\omega = 0.1$ rad/sec to $\omega = 100$ rad/sec.

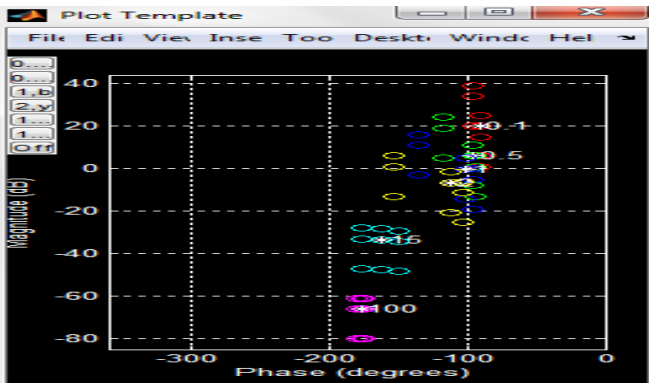


Fig 3 Plant Templates.

5.4 Stability bounds

It is well known that magnitude of $T(j\omega) \leq ML$ establishes a circle in the Nichols Chart. The specifications on system performance in the time domain and in the frequency domain identify a minimum Damping ratio for the dominant roots of the closed-loop system which corresponds to a bound on the value of $M_p = M_m$. On the NC this bound on $M_p = ML$ establishes a region which must not be penetrated by the templates and the loop transmission functions for all frequencies. The boundary of this region is referred to as the universal high-frequency boundary (UHFB) or stability bound, the U-contour [5].

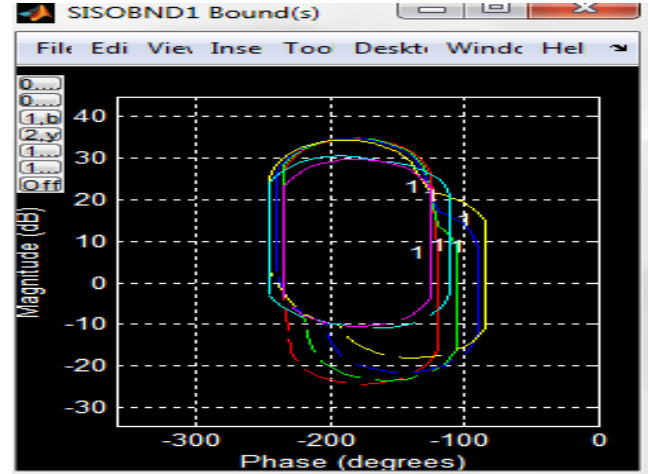


Fig 4 Stability bounds

5.5 Tracking Bounds

Depending upon the functions based on upper and lower limit curves, tracking Bounds are generated at various frequencies and different plant sets. Tracking bounds are generated by using MATLAB QFT TOOLBOX command. [5]

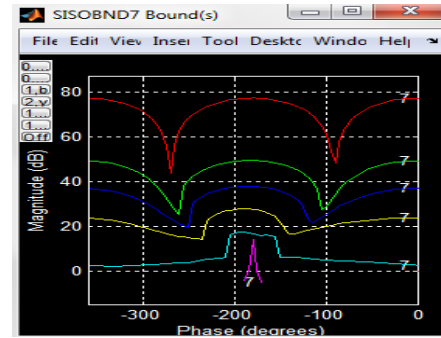


Fig 5 Tracking bounds

5.6 Output Disturbance Rejection Bound

Depending upon output disturbance rejection transfer function given in the specifications output disturbance bounds are generated at various frequencies and for different plants. Disturbance-bounds are generated by using MATLAB QFT TOOLBOX COMMAND. [5]

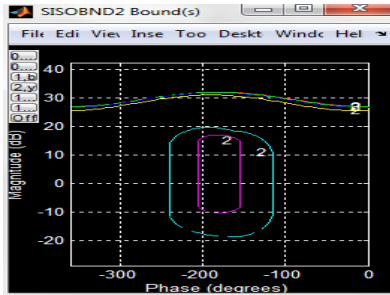


Fig 6 Disturbance Rejection bounds

5.7 Grouping Of Bounds

All the bounds i.e. stability, Tracking and disturbance bounds are grouped together in order to calculate the worst case possibilities. The dominant ones of all the three are retained. Grouped bounds are generated by using MATLAB QFT TOOLBOX COMMAND. For each frequency there is a dominant bound and is shown in intersection bound window of Matlab. [5]

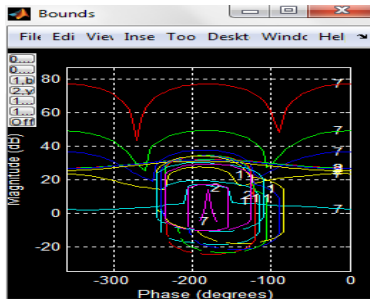


Fig 7 Group bounds

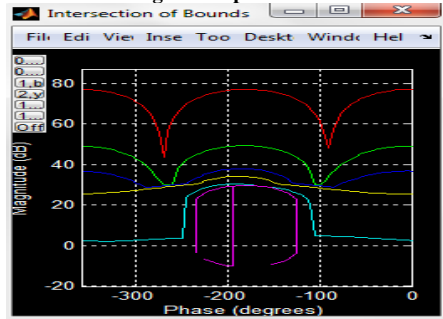


Fig 8 Dominant bounds

In this way specifications of closed loop systems for all the plants specified are translated in to that of open loop nominal case.

5.8 Loop Shaping

Loop shaping is a design method where it is attempted to choose a controller such that the loop transfer function obtains the desired shape. It is a key design step and it consists of shaping of the open loop function to set the boundaries given by the design specifications. Manual loop shaping can be done with use of QFT Matlab toolbox. The controller design is undertaken on the NC considering the frequency constraints and the nominal loop $L_0(s)$ of the system. At this point, the designer begins to introduce controller functions $G(s)$ and tune their parameters, a process called Loop Shaping, until the best possible controller is reached without violation of the frequency constraints the loop shaping with nominal plant is as shown in figure. [7]

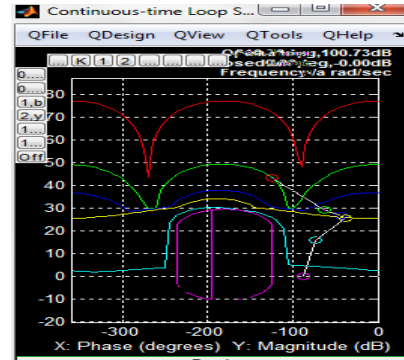


Fig 9 Loop Shaping

In this paper an initial controller is considered which is given by

$G(s) = \frac{20s^2 + 12s + 1}{s}$. Although the controller is easy to implement; it does not satisfy all the bounds of the QFT as shown in the fig. In order to obtain the best Controller, Manual loop shaping has to be performed with the help of CAD environment of QFT toolbox in Matlab.

5.9 Filter Design

Once the best controller is designed by manual loop shaping process a prefilter can be easily designed in order to keep desired tracking performance specifications, Using Matlab QFT toolbox prefilter is designed.

6. PROBLEMS ASSOCIATED WITH THE MANUAL LOOP SHAPING OF QFT CONTROLLER

The main advantage of QFT controller is that the design Experiences can be used and the design procedure is transparent to the designer. The designer can consider many factors which might be difficult to represent by analytical expressions or quantitatively. However when the plant has unstable zeros or poles or complicated characteristics it may be difficult to design a stabilizing controller manually. In addition, whether or not the design is successful mainly depends on designer's experience applied to trials. Even if manual design is done narrowly, the results are not always satisfying. Also manually designed results are not unique. Usually there are several controllers satisfying with boundary constraints. How to select the best controller from the candidates, i.e., how to realize the optimization of QFT controller, is worthy of investigation. Thus it is necessary to study more automatic QFT controllers. [7] Loop shaping may be considered as the most crucial step of designing a QFT controller. Thus only this particular step is considered in this paper.

7. PROPOSED TECHNIQUE FOR TUNING OF QFT CONTROLLER

Compared with above traditional QFT automatic design methods, the GA based loop-shaping technique has many advantages [6]. GA is the most efficient optimization method and can implement multi objective optimization. The controller can be automatically designed by using the computer searching program based on GA to make loop-shaping instead of using manual adjustment. We improve the technique of GA based automatic QFT, and introduce new optimization indexes to make the controller more practical. Compared with results of the manual design and other automatic design methods, the performance of the QFT controller designed by GA is better.

The problem that the designer must have experience for loop-shaping is overcome. And the problem that it is not easy to design controllers for complex plants is also solved.

Although the GA based automatic design method has many advantages mentioned above, many designers who have certain experiences also want to get the best controller in a short time. This is the optimization of QFT controller. What is called optimization of QFT controller is that based on an initial QFT controller, the parameters of the controller are optimized and the performance of the controller is improved. While the automatic design of QFT controller does not have reference information and the parameters region for searching is big. Compared with the automatic design, the searching region is optimal design is much smaller. Thus the searching time is reduced. We adopt a manual controller as the initial controller for optimization, and then a local optimization is performed on GA, so we can get the optimal QFT controller in a short time.

8. GENETIC ALGORITHM

Genetic Algorithms (GA's) are a stochastic global search method that mimics the process of natural evolution. The genetic algorithm starts with no knowledge of the correct solution and depends entirely on responses from its environment and evolution operators (i.e. reproduction, crossover and mutation) to arrive at the best solution. By starting at several independent points and searching in parallel, the algorithm avoids local minima and converging to sub optimal solutions. In this way, GAs have been shown to be capable of locating high performance areas in complex domains without experiencing the difficulties associated with high dimensionality, as may occur with gradient decent techniques or methods that rely on derivative information. A genetic algorithm is typically initialised with a random population consisting of between 20-100 individuals. This population (mating pool) is usually represented by a real-valued number or a binary string called a chromosome. For illustrative purposes, the rest of this section represents each chromosome as a binary string. How well an individual performs a task is measured is assessed by the objective function. The objective function assigns each individual a corresponding number called its fitness. The fitness of each chromosome is assessed and a survival of the fittest strategy is applied. In this project, the magnitude of the error will be used to assess the fitness of each chromosome. [14]

There are three main stages of a genetic algorithm; these are known as Reproduction, Crossover and Mutation.

8.1 REPRODUCTION

During the reproduction phase the fitness value of each chromosome is assessed. This value is used in the selection process to provide bias towards fitter individuals. Just like in natural evolution, a fit chromosome has a higher probability of being selected for reproduction.

Four common methods for selection are:

1. Roulette Wheel selection
2. Stochastic Universal sampling
3. Normalised geometric selection
4. Tournament selection

8.2 CROSSOVER

Once the selection process is complete, the crossover algorithm is initiated. The crossover operation swaps certain parts of the two selected strings in a bid to capture the good parts of old chromosomes and create better new ones. Genetic operators manipulate the characters of a chromosome directly, using the assumption that certain individual's gene codes, on average,

produce fitter individuals. Following are the three Crossover techniques

1. Single Point Crossover.
2. Multi Point Crossover.
3. Uniform Crossover

8.3 MUTATION

Using selection and crossover on their own will generate a large amount of different strings. However there are two main problems with this:

1. Depending on the initial population chosen, there may not be enough diversity in the initial strings to ensure the GA searches the entire problem space.
2. The GA may converge on sub-optimum strings due to a bad choice of initial population.

These problems may be overcome by the introduction of a mutation operator into the GA. Mutation is the occasional random alteration of a value of a string position. It is considered a background operator in the genetic algorithm. The probability of mutation is normally low because a high mutation rate would destroy fit strings and degenerate the genetic algorithm into a random search. Mutation probability values of around 0.1% or 0.01% are common, these values represent the probability that a certain string will be selected for mutation i.e. for a probability of 0.1%; one string in one thousand will be selected for mutation.

8.4 ELITISM

With crossover and mutation taking place, there is a high risk that the optimum solution could be lost as there is no guarantee that these operators will preserve the fittest string. To counteract this, elitist models are often used. In an elitist model, the best individual from a population is saved before any of these operations take place. After the new population is formed and evaluated, it is examined to see if this best structure has been preserved. If not, the saved copy is reinserted back into the population. The GA then continues on as normal []

The steps involved in creating and implementing a genetic algorithm are as follows:

1. Generate an initial, random population of individuals for a fixed size.
2. Evaluate their fitness.
3. Select the fittest members of the population.
4. Reproduce using a probabilistic method (e.g., roulette wheel).
5. Implement crossover operation on the reproduced chromosomes (Choosing probabilistically both the crossover site and the 'mates').
6. Execute mutation operation with low probability.
7. Repeat step 2 until a predefined convergence criterion is met.

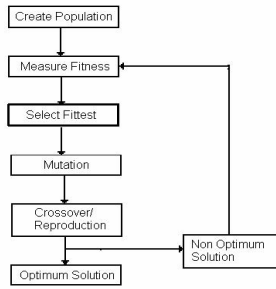


Fig 10 Genetic Algorithm Process

8.5 TERMINATION

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

1. A solution is found that satisfies minimum criteria.
2. Fixed number of generations reached.
3. Allocated budget (computation time/money) reached.
4. The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results.
5. Manual inspection.
6. Combinations of the above.

9. USING THE OPTIMIZATION TOOL

To open the Optimization Tool, enter Optimtool ('ga') at the command line, or enter optimtool and then choose ga from the Solver menu. [16]

To use the Optimization Tool, you must first enter the following information:

9.1 FITNESS FUNCTION:

The objective function you want to minimize. Enter the fitness function in the form @fitnessfun, where fitnessfun.m is a file that computes the fitness function. The @ sign creates a function handle to fitnessfun. Writing an objective function is the most difficult part of creating a genetic algorithm. In this paper, the objective function is required to evaluate the best PID controller for the DC Motor. An objective function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or quickest settling time but in order to combine all of these objectives it was decided to design an objective function that will minimise the error of the controlled system. Each chromosome in the population is passed into the objective function one at a time. The chromosome is then evaluated and assigned a number to represent its fitness, the bigger its number the better its fitness. The genetic algorithm uses the chromosome's fitness value to create a new population consisting of the fittest members.

In this paper the objective function is defined by various Error Performance Criterion such as MSE, ITAE, ISE, IAE. The controlled system is given a step input and the error is assessed using an appropriate error performance criterion i.e. ITAE, ISE, IAE or MSE. The chromosome is assigned an overall fitness value according to the magnitude of the error, the smaller the error the larger the fitness value. The above mentioned objective functions are specified with the help of a code in a Matlab file and it is then called appropriately from the objective functions pane.

9.2 Number of variables

The length of the input vector to the fitness function is defined by number of variables. In this paper the number of variables are 3 i.e. Kp, Z1, Z2.

9.3 CONVERSION OF QFT BOUNDS IN TO QUADRATIC INEQUALITIES OR GA CONSTRAINTS

You can enter constraints or a nonlinear constraint function for the problem in the Constraints pane. If the problem is unconstrained, leave these fields blank.

- (1) Stability bound given by

$$\left| F \frac{PGH}{1 + PGH} \right| \leq Ws1$$

is converted to a quadratic inequality given by

$$P^2 \left(1 - \frac{1}{\delta_1^2} \right) g^2 + 2P \cos(\theta + \theta) g + 1 \geq 0$$

- (2) Disturbance bound given by

$$\left| F \frac{H}{1 + PGH} \right| \leq Ws2$$

is converted to a quadratic inequality given by

$$P^2 g^2 + 2P \cos(\theta + \theta) g + \left(1 - \frac{1}{\delta_2^2} \right) \geq 0$$

- (3) Tracking bound given by

$$Ws3 \leq \left| F \frac{PG}{1 + PGH} \right| \leq Ws4$$

is converted to a quadratic inequality given by

$$p_e^2 p_d^2 (1 - 1/\delta_3^2) g^2 + 2 p_e p_d (p_e \cos(\theta + \theta_d) - p_d/\delta_3^2 \cos(\theta + \theta_e)) g + (p_e^2 - p_d^2/\delta_3^2) \geq 0$$

$$\text{WHERE } p = |p| \text{ \& " } \theta = \angle p \text{ \& } g = |g| \text{ \& } \theta = \angle g$$

The above derived constraints are specified with the help of a code in a Matlab file and it is then called appropriately from the constraints pane.

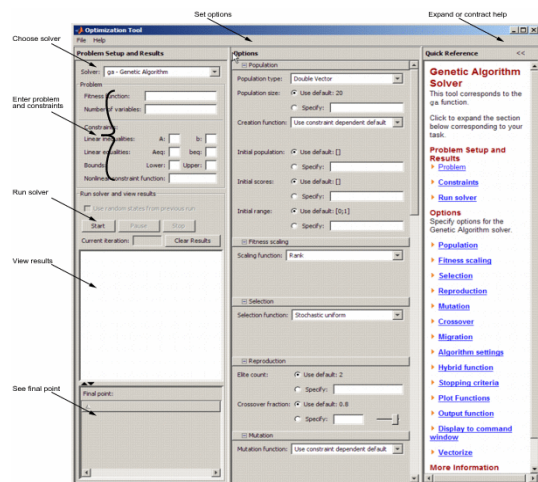


Fig 11 GA Solver window in Matlab

To run the genetic algorithm, click the Start button. The tool displays the results of the optimization in the Run solver and view results pane.

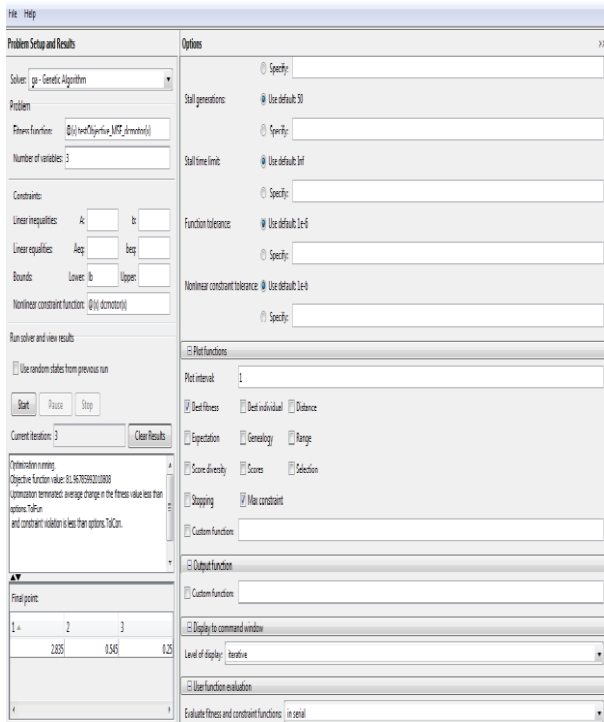


Fig 12 GA Solver displaying results

After running Genetic Algorithm the optimized values of K_p , Z_1 , Z_2 for different Error Performance Criterion are shown below:

IAE

$K_p=3.422736908487951$;
 $z1=0.504455566406250$;
 $z2=0.323009751215281$;

ITAE

$K_p=0.833933617167150$;
 $z1=1.259402420027586$;
 $z2=0.0349251846221$;

ISE

$K_p=3.270617093138372$;
 $z1=0.685546875000000$;
 $z2=0.233036539926829$;

MSE

$K_p=2.834755580119764$;
 $z1=0.544967338442802$;
 $z2=0.250000230446162$;

The above tuned parameters of PID Controller provide automatic loop shaping of open loop plant Transfer function which satisfies all the bounds of QFT, Thus providing us the best of the controller.

The shaped response of control system is shown below

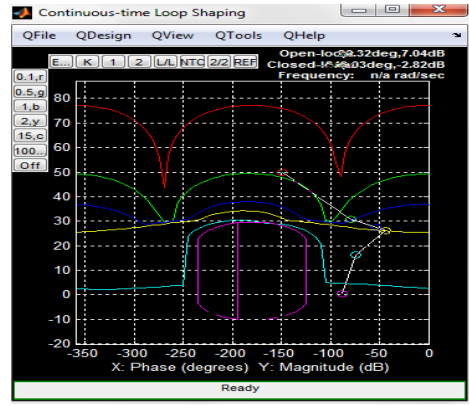


Fig 13 Shaped Response Of Control System

The above control system can be verified by applying a step input to it and then observing its output response. The response so obtained is very close to step input. So GA based parameter optimization method for QFT controller can accurately adjust parameters on the basis of the results from manual design method, which can obtain good QFT controller in a short time. Thus it is good combination of manual and automatic design.

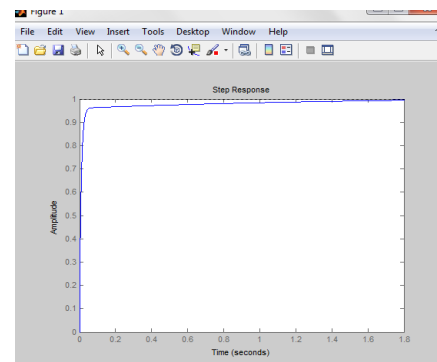


Fig 14 Step Response Of Control System verifying the Results in the paper

10. CONCLUSION

QFT can provide robust control for the plant with large uncertainties. But for the complicated plant, it is hard to get the controller by manual design with the help of QFT TOOLBOX in Matlab. Traditional automatic design has limited capability with such multiple objective optimization problems as what appears in QFT CONTROLLER design. But GA is a more efficacious global optimization method; it can make optimization aiming to multiple goals. So in this paper, we present the automatic design method of QFT controller based on GA, and propose several optimization indices to make controller more practical. Then the paper expatiates on optimal design methods which combines the manual design and automatic design. The experiment shows that the GA based automatic design method of QFT controller is very effective and optimal design method can achieve both high performance and increased searching efficiency.

11. REFERENCES

- [1] I. J. Nagrath and M. Gopal , “Control Systems Engineering.”New Age International (P) Limited.
- [2] R. C. Dorf, Modern Control Systems. Addison-Wesley publishing Company, 5th edition, 1990.
- [3] Houpis, C.H., S. J. Rasmussen and M. Garcia-Sanz Quantitative Feedback Theory. Fundamentals and Applications. 2nd Edition (Taylor & Francis, 2006).
- [4] B. C. Kuo, Automatic Control Systems. Prentice-Hall International Editions, 1991.
- [5] Borghesani, C., Y. Chait, O. Yaniv, Quantitative Feedback Theory Toolbox v2.0 - For use with MATLAB (Terasoft, 2003).
- [6] Chen, W., D.J. Balance and Y. Li, “Automatic loop-shaping in QFT using genetic algorithms,” Proceedings of the 3rd Asia-Pacific Conference on Control and Measurement, pp. 63-67, 1998.
- [7] Qingwei Wang, Zhenghua Liu, Lianjie Er, School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics. Beijing 100083
- [8] BRYANT G.F. and HALIKIAS G.D., 1995, Optimal loop shaping for systems with large parameter uncertainty via linear programming. International Journal of Control, 62, 557-568.
- [9] Horowitz, I. M., “Survey of quantitative feedback theory (QFT),” International Journal of Control, 53 (2), 1991, pp. 255-291.
- [10] [10] Chait Y, and Yaniv O. (1993), Multi-input/single-output computer-aided control design using the Quantitative Feedback Theory. Int. J. Robust Nonlinear Control, 1993, No.3, pp. 47-54.
- [11] Mario Garcia-Sanz, Automatic Control and Computer Science Department, Public University of Navarra. Campus, Arrosadia, 31006 Pamplona, SPAIN.
- [12] Horowitz I.M., “Optimum loop transfer function in single-loop minimum-phase feedback systems,” *Int. J. Control*, vol. 18, no. 1, pp. 97-113, 1973.
- [13] Ibtissem Chiha,1 Nouredine Liouane,2 and Pierre Borne3, Tuning PID Controller Using Multiobjective Ant Colony Optimization.
- [14] K. Krishnakumar and D. E. Goldberg, “Control System Optimization Using Genetic Algorithms”, *Journal of Guidance, Control and Dynamics*, Vol. 15, No. 3, pp. 735-740, 1992.
- [15] Help files of Matlab 7.10
- [16] Nataraj, P.S.V. (2002). Interval QFT: a mathematical and computational enhancement of QFT.