AUTOMATIC DESIGN OF ROBUST OPTIMAL CONTROLLER FOR INTERVAL PLANTS USING GENETIC PROGRAMMING AND KHARITONOV THEOREM

PENG CHEN[*]
Department of Automation, Shanghai Jiaotong University
Shanghai, 200240, P. R. China
E-mail: pengchen@sjtu.edu.cn

YONG-ZAI LU
Department of Automation, Shanghai Jiaotong University
Shanghai, 200240, P. R. China
E-mail: y.lu@ieee.org

Received 11 March 2010
Accepted 19 May 2011

Abstract

This paper presents a novel approach to automatic design of a robust optimal controller for interval plants with Genetic Programming based on Kharitonov Theorem (KT), which provides a theoretical foundation in the design of robust controller for interval plants. The structure and parameters of the robust optimal controller for interval plants are optimized by Genetic Programming and the Generalized KT related stability criteria are integrated into the solution to guarantee the stability of the closed-loop system. Consequently, the evolved controller not only minimizes time-weighted absolute error (ITAE) of the closed-loop system, but also stabilizes the whole interval plant family robustly. Finally, the simulations on a benchmark problem show that the proposed method can effectively generate a robust optimal controller for interval plants.

Keywords: Genetic Programming; Interval plant; Kharitonov Theorem; Robust Optimal Controller

1 Introduction

During the last few decades, significant progresses have been made in the realm of robust stability and control for parameter uncertain systems.[1-6] The Kharitonov Theorem has been one of the most popular approaches to investigate stability of interval systems via four vortex polynomials with real-valued min-max bounds.[7] Based on Kharitonov Theorem, both Edge Theorems and Box Theorems suggest that the set of transfer functions generated by changing the perturbed coefficients in the parameter ranges correspond to a box in parameter space and are referred to "interval plants".[8,9] Consequently, the stability criterion of the entire family of infinite interval plants can be simplified to only check the stability with limited number of systems. Moreover, the number of finite systems to be checked has nothing to do with the dimension of the uncertain parameter space, thus the "dimension disaster" may be avoided. Many important results derived from Kharitonov Theorem have been reported in the close-loop system stability analysis[1,2,3,5,6,10] and the frequency domain properties of interval plants.[11-14]

However, the design of robust controller for interval plant family is still an attractive problem for academic and industrial communities, since most published results

[*] Corresponding author. Email: pengchen@sjtu.edu.cn

Published by Atlantis Press

in this area are restricted to analytical methods; only limited progresses on the controller design methods have been made.[4,15,16,17,18] On the other hand, the robust stability is not the only issue to be considered in the design of a robust control system, i.e., the desired static and dynamic criteria should also be included. All these issues together with system stability heavily depend on the design of controller structure and the parameters setting. However, these problems usually are hard to be expressed by a mathematical formula and solved by traditional optimization techniques.[19]

The applications of computational intelligence (CI) methods in control engineering have been a promising alternative to solve the above-mentioned problems, due to its superior features in population based intelligent search and optimization with ill-behaved cost landscapes as well as complicated constraints.[18-22] Among them, the applications of Genetic Programming (GP) in system modeling and controller design have been an attractive research area due to its unique functionality to simultaneously optimize structure and parameters of the problem to be solved. However, there are only limited number of published GP applications in control engineering field.[20,23,24,25,26,27] It should be emphasized that one of the major disadvantages preventing the GP applications in control engineering is the lack of mathematical foundation in stability, reliability and robustness under system parameter uncertainties.[20] In fact, all those issues are vitally critical when designing a robust controller with desired performances subject to the relevant constraints. Consequently, a GP based controller design method combining the knowledge from both computational intelligence and control theory could help the synthesis of innovative and high practically feasible control systems.

The objective of this study is to develop a novel robust optimal controller synthesis approach for interval plants in terms of GP associated with Kharitonov Theorem. The Generalized Kharitonov Theorem is integrated into the heuristic genetic programming method to guarantee that the evolved controller can strictly stabilize entire family of the interval plants and minimize the worst case Integral of Time-weighted Absolute Error (ITAE) in the mean time, which has never been studied before. Moreover, the functions and the terminals used in this paper are simple, hence some undesired intensive actions (such as high order derivative) are diminished, and the optimization time is reduced considerably compared with previous studies.

The rest of the paper is organized as follows: Section 2 describes the problem formulation of robust controller design for interval plants with parameter uncertainties. Section 3 recalls Kharitonov Theorem and some relevant results for close-loop system with interval plants. Section 4 introduces the fundamentals of GP briefly. The methodology and algorithm of designing robust optimal controller based on GP and Kharitonov Theorem are given in Section 5. The simulation study and its results on a benchmark interval plant are given to show the efficiency and effectiveness of the proposed method in Section 6. Finally, the concluding remarks are drawn in Section 7.

2 Problem Descriptions

Let the general form of a SISO (single-input single-output) interval plant be described by the following transfer function[5]:

$$G_p(s,\vec{a},\vec{b}) = \frac{b_0 + b_1 s + b_2 s^2 + \ldots + b_n s^n}{a_0 + a_1 s + a_2 s^2 + \ldots + a_m s^m} = \frac{\hat{N}(s)}{\hat{D}(s)} \quad (1)$$

where the coefficient vectors $\vec{a} = (a_0, a_1, \ldots, a_m)$ and $\vec{b} = (b_0, b_1, \ldots, b_n)$ lie in the m and n dimensional boxes respectively, $n \le m$.

$$A = \left\{ \vec{a} : a_i \in \left[a_i^-, a_i^+ \right], \forall i = 0,\ 1,\ 2,\ \ldots,\ m-1,\ m \right\} \quad (2)$$

and

$$B = \left\{ \vec{b} : b_i \in \left[b_i^-, b_i^+ \right], \forall i = 0,\ 1,\ 2,\ \ldots,\ n-1,\ n \right\} \quad (3)$$

Suppose that a robust optimal controller $C(s,\vec{p},\vec{q},l)$ for the interval plant $G_p(s,\vec{a},\vec{b})$ can be described as:

$$C(s,\vec{p},\vec{q},l) = \frac{F_1(s)}{F_2(s)} = \frac{q_0 + q_1 s + \ldots + q_{l-1} s^{l-1} + q_l s^l}{p_0 + p_1 s + \ldots + p_{l-1} s^{l-1} + p_l s^l} \quad (4)$$

where the $\vec{p} = (p_0, p_1, \ldots\ldots, p_l)$ and $\vec{q} = (q_0, q_1, \ldots\ldots, q_l)$ represent the vectors of the controller parameters in the controller parameter domain $\vec{p} \in \Re^l$ and $\vec{q} \in \Re^l$, respectively.[5] The parameter l in Eq. (4) denotes the order of the controller. The closed-loop control system in Fig. 1 should provide satisfactory performance when the parameters of the interval plant $G_p(s,\vec{a},\vec{b})$ are perturbed within predefined min-max bounds in Eq. (2) and Eq. (3).

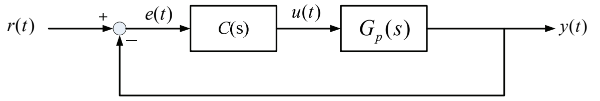

Fig.1. Closed-loop system structure of robust controller for interval plants

In this paper, the robust optimal controller design for interval plants is based on "worst-case" design principle. Conceptually, a robust optimal controller for the whole interval plant family can get satisfied performance on some "worst-cases" and meanwhile stabilize the other cases.[18]

For a given set of controller parameters $(\vec{p},\vec{q},l)$, the worst-case of the interval plants can be denoted by:

$$WorstJ = \underset{\substack{\vec{a}\in A \\ \vec{b}\in B}}{Max}(J(\vec{p},\vec{q},l,\vec{a},\vec{b})) \qquad (5)$$

The design of a robust optimal controller can thus be formulated as a min-max problem of searching for the best controller structure and parameters from all admissible space, which will minimize the $WorstJ$ in Eq. (5) as well as stabilize the whole interval plant family:

$$\begin{aligned}&\underset{\substack{\vec{p}\in\Re^l \\ \vec{q}\in\Re^l}}{Min}(\underset{\substack{\vec{a}\in A \\ \vec{b}\in B}}{Max}\, J(\vec{p},\vec{q},l,\vec{a},\vec{b})) \\ &Subject\ to: the\ interval\ plant\ family\ is\ stable.\end{aligned} \qquad (6)$$

More detailed discussions about J and $WorstJ$ will be introduced based on robust optimal design principle and Generalized Kharitonov Theorem in the following sections.

3 Kharitonov Theorem and Related Results on Interval Feedback Systems

Before applying Kharitonov Theorem to the design of robust optimal controller, the following definitions are given firstly:

Definition 1: Interval polynomial and Kharitonov Theorem

Let the characteristic polynomial of a closed-loop control system with parametric uncertainties be written as:

$$D(s) = d_0 + d_1 s + d_2 s^2 + d_3 s^3 + d_4 s^4 + ... + d_i s^i \qquad (7)$$

where $d_i \in [d_i^-, d_i^+], i = 0,1,2,3...$ are defined as the characteristic coefficients with min-max bounds. Eq. (7) is defined as an interval polynomial. The Kharitonov Theorem states that "*the stability of an interval polynomial can be determined by testing the stability of just four polynomials which can be easily obtained by using upper and lower levels of the unknown parameters*".[7] For example, $D(s)$ in Eq. (7) is stable if and only if the following four extreme polynomials (Kharitonov polynomials) are stable:

$$D_1(s) = d_0^- + d_1^- s + d_2^+ s^2 + d_3^+ s^3 + d_4^- s^4 + ... \qquad (8)$$

$$D_2(s) = d_0^- + d_1^+ s + d_2^+ s^2 + d_3^- s^3 + d_4^- s^4 + ... \qquad (9)$$

$$D_3(s) = d_0^+ + d_1^- s + d_2^- s^2 + d_3^+ s^3 + d_4^+ s^4 + ... \qquad (10)$$

$$D_4(s) = d_0^+ + d_1^+ s + d_2^- s^2 + d_3^- s^3 + d_4^+ s^4 + ... \qquad (11)$$

Definition 2: Kharitonov plants

Consider the interval plant family described in Eq. (1), let $G_{ik}(s)$ for $i,k = 1,2,3,4$ denote the so-called Kharitonov Plants:

$$G_{i,k}(s) = \frac{\hat{N}_i(s)}{\hat{D}_k(s)} \qquad (12)$$

where $\hat{N}_i(s)$ and $\hat{D}_k(s)$ for $i,k = 1,2,3,4$ are the Kharitonov polynomials associated with the transfer function in Eq. (1):

$$\hat{N}_1(s) = b_0^- + b_1^- s + b_2^+ s^2 + b_3^+ s^3 + b_4^- s^4 + ... \qquad (13)$$

$$\hat{N}_2(s) = b_0^- + b_1^+ s + b_2^+ s^2 + b_3^- s^3 + b_4^- s^4 + ... \qquad (14)$$

$$\hat{N}_3(s) = b_0^+ + b_1^- s + b_2^- s^2 + b_3^+ s^3 + b_4^+ s^4 + ... \qquad (15)$$

$$\hat{N}_4(s) = b_0^+ + b_1^+ s + b_2^- s^2 + b_3^- s^3 + b_4^+ s^4 + ... \qquad (16)$$

and

$$\hat{D}_1(s) = a_0^- + a_1^- s + a_2^+ s^2 + a_3^+ s^3 + a_4^- s^4 + ... \qquad (17)$$

$$\hat{D}_2(s) = a_0^- + a_1^+ s + a_2^+ s^2 + a_3^- s^3 + a_4^- s^4 + ... \qquad (18)$$

$$\hat{D}_3(s) = a_0^+ + a_1^- s + a_2^- s^2 + a_3^+ s^3 + a_4^+ s^4 + ... \qquad (19)$$

$$\hat{D}_4(s) = a_0^+ + a_1^+ s + a_2^- s^2 + a_3^- s^3 + a_4^+ s^4 + ... \qquad (20)$$

For a first order controller, such as

$$C(s) = K\frac{s-z}{s-p} \qquad (21)$$

The close-loop stability criterion of a feedback control system with interval plant family can then be described by:

Theorem 1. *A first order controller $C(s)$ can stabilize the interval plant family $G_p(s,\bar{a},\bar{b})$ if and only if $C(s)$ can stabilize the sixteen Kharitonov Plants $G_{i,k}(s)$.*[1,5]

Definition 3: 32 subsets of interval plant family

Based on the Kharitonov polynomials for the numerator and the denominator defined in Eq. (13)-(20), the following 32 subsets of interval plant family can be obtained:

$$G_E(s) = \frac{\hat{N}_k(s)}{(1-\lambda)\hat{D}_i(s)+\lambda\hat{D}_j(s)} \cup \frac{(1-\lambda)\hat{N}_i(s)+\lambda\hat{N}_j(s)}{\hat{D}_k(s)} \quad (22)$$

where $\lambda \in [0,1]$ and $(i,j) \in \{(1,2),(1,3),(2,4),(3,4)\}$.

Then the close-loop stability of a feedback control system with interval plant family and a higher order controller $C(s,\bar{p},\bar{q},l)$ can be solved:

Theorem 2. *A higher order controller $C(s,\bar{p},\bar{q},l)$ can stabilize the whole interval plant family if and only if it can stabilize the 32 subsets of interval plant family.*[2,5]

4 Introduction to Genetic Programming

In computational intelligence, Genetic Programming proposed by Koza[28] is a novel evolutionary approach to automatically generate human competitive results by computer technology for many engineering problems and has produced many novel and outstanding results in areas such as quantum computing, electronic design, game playing, sorting, searching and complex structure designs.[29] It starts with a set of randomly created computer programs which are composed of numerous predefined functions and terminals appropriate to the problem domain. The creation of this initial random population is, in fact, a blind random search on the search space of the problem represented by computer programs. A run of genetic programming is a competitive search among a diverse population of programs composed of the available functions and terminals. GP can often create novel designs because it is a probabilistic process that is not encumbered by the preconceptions that often channel human thinking down familiar paths.[26,28,29]

Before genetic programming is implemented to solve real world problems, the following issues should be specified: searching space of the Algorithm (including the terminals, the functions, etc), the fitness measure and control parameters for the evolution. Then the following three steps are executed by the algorithm in order to evolve the population.[26]

(i) An initial population of typically random possible solutions is generated.

(ii) The following cycle of steps is performed until the termination criterion is satisfied.

(a) Each program of the population is executed to determine its value of fitness.

(b) A new population is created by the application of the following operations. The individuals are selected with a probability function based on the fitness measure.

- *Reproduction*: a copy of the selected program is inserted into the new population.
- *Crossover*: a new offspring program is created by recombining parts of two other selected programs.
- *Mutation*: a new offspring program is created by mutating a randomly chosen part of the selected individual.
- *Architecture-altering operations:* a new offspring program is created by altering the program architecture of the selected individual.

(iii) The individual with the best fitness is chosen to represent the solution of the current generation.

5 Automatic Synthesis of Robust Controller Using Genetic Programming and Kharitonov Theorem

5.1 *GP solution for robust controller design*

When GP is applied to design a controller with desired criteria subjected to given constraints, the following aspects are vital:

(i) Introducing control domain knowledge (control and robustness criteria, relevant constraints, the feasible components and operators, etc.) in the design of new controller with GP will definitely make search process more effective and reasonable. The applications of GP in auto-design of control systems should not be a blind search, but a domain-driven or meta-heuristic search process.

(ii) The high computation costs are often a disadvantage when using GP-based auto-design for control systems. It will be helpful to improve the search efficiency and accelerate the optimization process if the infeasible or unfitted individuals are removed at the early stage of the evolution process by checking some relevant conditions (constraints or stability criteria).

The GP algorithms for auto-design of controller originally proposed by Koza were mainly from computer science point of view. Compared with traditional PID controller, the generated controller provides better performance for the reason that the controller evolved by GP contains second derivative blocks.[26] A detailed insight into Koza's controller shows that it uses the control variable in a more intensive way than the standard PID, saturated control and higher varying rate of the control variable are employed to achieve better performance.[20] However, from a control engineering point of view this is not approved because higher order differentiators may result in amplifying high frequency effects, such as noises. Moreover, the application of higher order differentiators should be limited, since the controller with higher order differentiators might work in a saturated region.[20] Most importantly, when GP was used to design a robust optimal controller for a system with parameter uncertainties (such as interval plant family), only two boundary systems of the interval plants were checked with upper and lower levels,[25,26] no strict mathematic theorem was given to guarantee the evolved controller can stabilize the whole interval plant family.

The GP algorithm to synthesize a robust optimal controller in this paper considers both sides of the coin. In computer control systems, controllers are usually discrete, because of the sampling required by computer implementations, and discrete time controllers are easier to implement and quicker to simulate. All the basic conclusions on close-loop system stability drawn in the earlier sections will still hold for discrete-time controller because the continuous transfer function of the controller can be transformed into an equivalent Z-transfer function by a given sample time interval without loss of any feature and precision. Based on the equivalent *Z*-transfer function, the discrete control law can be further described by a difference equation as follows:

$$\begin{aligned}U(k) = \alpha_1 * U(k-1) + \ldots + \alpha_i * U(k-i) + \\ + \beta_1 * Y(k-1) + \ldots + \beta_j * Y(k-j) + \\ + \omega_1 * e(k-1) + \ldots + \omega_l * e(k-l) + \\ + \theta_1 * r(k) + \ldots + \theta_l * r(k-m)\end{aligned} \tag{23}$$

where

$U(k-1), \ldots U(k-i)$
- Time shifted controller outputs,

$Y(k-1), \ldots Y(k-j)$
- Time shifted system outputs,

$e(k-1), \ldots e(k-l)$
- Time shifted error between Y and r ,

$r(k), \ldots r(k-m)$
- Reference trajectory of set-point,

$\alpha_1, \ldots \alpha_i, \beta_1, \ldots \beta_j, \omega_1, \ldots, \omega_l, \theta_1, \ldots \theta_m$
- The set of controller parameters,

In this paper, a discrete recursive feedback control law will be created from a terminal set T consisting solely of the reference inputs, system outputs and controller outputs shifted by time sample, i.e.

$$\begin{aligned}T = \{u(k-1), \ldots, u(k-i), Y(k-1), \ldots, Y(k-j), \\ e(k-1), \ldots, e(k-l), r(k), \ldots, r(k-m), \\ \alpha_1, \ldots \alpha_i, \beta_1, \ldots \beta_j, \omega_1, \ldots, \omega_l, \theta_1, \ldots \theta_m\}\end{aligned}$$

and the function set $F = \{Multiply, Add, Subtract\}$ is much simpler compared with previous studies.[25,26]

5.2 *Criteria and constraints*

Constraints:

The value of index i, j, l and the varying ranges of the coefficients $\alpha_1, \ldots \alpha_i, \beta_1, \ldots \beta_j, \omega_1, \ldots, \omega_l, \theta_1, \ldots \theta_m$ in Eq. (23) constitute constraints of the controller structure and the controller parameters.

Stability:

The individuals in the evolution (controller candidates) should first meet the requirement that it can stabilize the whole interval plant family; this is highly important and fundamental issue for a robust optimal controller design problem.

Overshoot:

The maximal deviation that the system output response proceeds beyond the desired value after applying a step to reference signal.

Rise time:

The duration from an ideal step input until the output of the system reaches 90% of its steady-state value for the first time.

Settling time:

The duration from an ideal step input until the output of the system enters and remains within a specified error band related to the amplitude of the pulse and the expected final settling value.

Control criteria:

ITAE:

The Integral of Time-weighted Absolute Error (ITAE) between reference and plant output. In the area of control engineering, ITAE has been used as a popular criterion that provides better optimum performance evaluation for various systems.[30]

All the constraints and performance indices discussed will be integrated into the Genetic Programming to guide the evolution of the controller.

5.3 *Robust optimal controller design based on generalized Kharitonov Theorem*

A block scheme of a GP-based robust optimal controller design used in this paper is shown in Fig. 2.

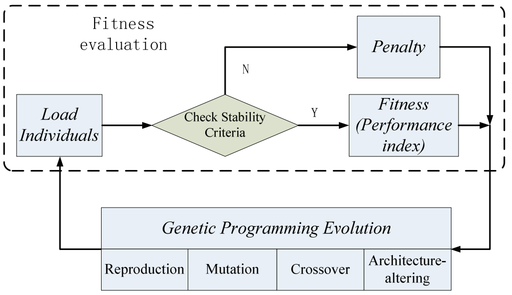

Fig.2. Block scheme of GP-based robust optimal controller design

The design procedure includes the following two key issues:

Issue One: Guarantee the robust stability of the close-loop system on the whole interval plant family. According to Theorem 2, this is achieved by checking the stability of 32 subsets defined in Eq. (22) for every solution in the beginning of each generation, as shown in Fig. 2. An individual which can't meet the stability criteria will be evaluated as an inferior solution with a large penalty.

Issue Two: Seek the best performance of the resulting closed-loop control system. This can be achieved by minimizing the maximal ITAE of the close-loop system during the evolution.

In this paper, a modified ITAE index J is used to evaluate individual related performance of the closed-loop system in Genetic Programming as defined in Eq. (24):

$$J(\vec{p},\vec{q},l,\vec{a},\vec{b}) = \int_0^t t\,|e(t)|\,\phi(|e(t)|,t)\,\psi(e(t))\,dt \quad (24)$$

where $e(t) = r(t) - y(t)$ is the error between reference input and system output, $\phi(\bullet,\bullet)$, $\psi(\bullet)$ are weighting functions of $|e(t)|$, t and $e(t)$ respectively, the function $\phi(|e(t)|,t)$ here is used to penalize long settling time and prevent oscillation:

$$\phi(|e(t)|,t) = \begin{cases} 1, \; if\ rise\ time\ t_r\ less\ than\ t_s \\ or\ |e(t)|\ \text{less}\ than\ \delta\ after\ rise\ time\ t_r, \\ 5, \; otherwise \end{cases} \quad (25)$$

where δ is a predefined parameter with a small value to prevent oscillation and t_s is a predefined parameter to penalize long settling time.

$\psi(e(t))$ is used to penalize the overshoot of the closed-loop system, defined as:

$$\psi(e(t)) = \begin{cases} 1, & if\ \text{overshoot}\ is\ \text{less}\ than\ \mu \\ 10, & otherwise \end{cases} \quad (26)$$

where μ is a predefined parameter with a small value to prevent overshoots.

Thus the constraints (*Stability, overshoot, settling time, rising time, etc.*) mentioned in section 5.2 can be processed by penalty structure, in which a penalty is imposed on the fitness function if the constraints are violated.

On the other side, there are some valuable extreme point results based on the generalized Kharitonov Theorem which can be used here to reveal the "worst-cases" of an interval plant family. It is shown that maximum H_∞ norm for an interval plant family is achieved at one of the 16 Kharitonov plants.[10,11] Other researchers find that large and critical portions of the Nyquist envelope are covered by the envelope deduced on the 16 Kharitonov plants.[12,13] A method was proposed by Bartlett [14] to find Bode envelopes for interval systems, and it was shown that Bode envelope of interval plants can be determined from the corresponding Kharitonov plants. All these results imply that the "worst case" of an interval plant family lies on or near the 16 Kharitonov plants. Consequently, the 16 Kharitonov plants are used as the "worst cases" of interval plant family to design a robust optimal controller in this paper.

To minimize the "worst-case" ITAE of the interval plant family, the sum of the ITAE on 16 Kharitonov plants is used as the fitness function to evaluate the performance of a given controller $G_c(z,\bar{p},\bar{q})$ on the whole interval plant family:

$$\begin{aligned} Fitness &= \underset{\substack{a\in A\\ b\in B}}{Max}\, J(G_c(z,\bar{p},\bar{q}),\bar{a},\bar{b}) \\ &\cong \sum_{i=1}^{4}\sum_{k=1}^{4} J(G_c(z,\bar{p},\bar{q}),\ \mathrm{G_{i,k}}(\bullet)\ ) \end{aligned} \qquad (27)$$

where $G_{i,k}(\bullet)$ refers to the 16 Kharitonov plants defined in Eq. (12).

The objective of our design problem is to find a robust optimal controller $G_c(z,\bar{p},\bar{q})$ that can minimize the fitness defined in Eq. (27) and stabilize the whole interval plant in the mean time:

$$\begin{aligned} & Min(\underset{\substack{a\in A\\ b\in B}}{Max}\, J(G_c(z,\bar{p},\bar{q}),\bar{a},\bar{b})) \\ & \textit{Subject to: the interval plant family is stable.} \\ & \cong Min(\sum_{i=1}^{4}\sum_{k=1}^{4} J(G_c(z,\bar{p},\bar{q}),\ \mathrm{G_{i,k}}(\bullet))) \\ & \textit{Subject to: the interval plant family is stable.} \end{aligned} \qquad (28)$$

6 Experimental Tests and Analytical Results

6.1 *Numerical examples:*

The example of interval plant family used in this paper was first published as a model of an oblique wing aircraft.[31] The model is given by

$$G(s,\bar{a},\bar{b}) = \frac{b_1 s + b_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \qquad (29)$$

in which its parameter uncertainties are specified by:

$$\begin{aligned} & 90 \le b_0 \le 166,\ 54 \le b_1 \le 74,\ -0.1 \le a_0 \le 0.1, \\ & 30.1 \le a_1 \le 33.9,\ 50.4 \le a_2 \le 80.8,\ 2.8 \le a_3 \le 4.6. \end{aligned} \qquad (30)$$

This model is also used in many published papers[1,18] as a benchmark example for robust controller design or system stability analysis.

6.2 *Experiment settings:*

Consider a discrete-time control law which has the form of Eq. (23), and the sampling time interval is set as 0.1s. The set-point of controlled variable is a square wave varying between +1 and -1 with a time period of 20 seconds, and the integral time for ITAE is from 0s to 20s.

The search space of the random constants is set between [-10 10]. The l,i,j,m in Eq. (23) is set to be 9, which means the order of the resulting controllers is equal or less than 9 according to the constraints described in section 5.2. The parameter δ in Eq. (25) is set to be 0.1 and t_s be 2 seconds, hence the individuals with settling time longer than 2 seconds and larger oscillation are penalized. The parameter μ in Eq. (26) is set to be 0.2.

The probabilities of genetic operators are fixed and set to be equal, in other words, both mutation and crossover probability during GP evolution process are set equal to 0.5. The population size = 200.

The program is executed on a PC with Pentium 4 2.8 G CPU, 512MB RAM and Windows XP operating system with Matlab 7.

6.3 *Simulation results and analysis:*

After 5000 generations, a controller with satisfied performance was automatically created. The computation time elapsed is 180 hours. Although the evolutionary process of Genetic Programming in our paper is still very time consuming at first glance; it has been reduced considerably compared with Koza's algorithm,[26] which cost 44.5 hours on a parallel cluster computer system consisting of 66 processing nodes (each containing a 533-MHz DEC Alpha microprocessor and 64 megabytes of RAM). Fig. 3 shows the evolution of best-individual fitness generation

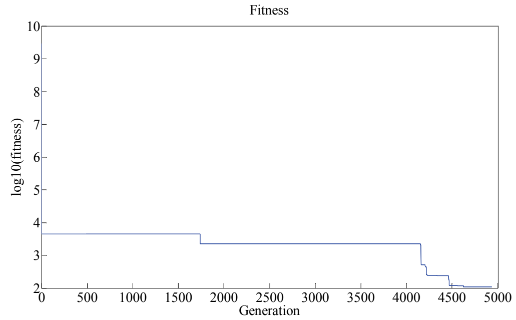

Fig.3. Fitness of best-individual generation by generation

Table.1 Best controller evolution generation by generation

Generation	Control Law U(k)	Fitness
1*	e(k-1)	4.5109e+03
1737*	y(k-6)-y(k-5)+e(k-1)	2.2700e+03
4154*	e(k-1)-0.75204e(k-3)	2.0781e+03
4159	e(k-1)-0.75204e(k-3)+y(k-6)-y(k-5)	509.2597
4206	y(k-7)-y(k-5)+ e(k-1)-0.38696e(k-3)	448.8759
4219	y(k-7)-y(k-5)+ e(k-1)-0.624226e(k-3)	256.8655
4227	e(k-1)-0.55483e(k-3)+y(k-7)-y(k-5)	246.8200
4338	e(k-1)-0.56441e(k-3)+y(k-7)-y(k-5)	239.2471
4465	e(k-1)-0.17746e(k-3)-0.38696e(k-5)+y(k-7)-y(k-5)-0.38696u(k-5)	192.2903
4466	e(k-1)-0.23722e(k-3)-0.38696e(k-5)+y(k-7)-y(k-5)-0.38696u(k-5)	186.0496
4467	e(k-1)-0.17746e(k-3) +y(k-7)-y(k-5) -0.38696u(k-5)	171.3529
4468	e(k-1)-0.62418e(k-3) +y(k-7)-y(k-5) -0.38696u(k-5)	121.7594
4471	1.14973e(k-1)-0.77392e(k-3) +y(k-7)-y(k-5) -0.38696u(k-5)	121.4193
4475	1.23118e(k-1)-0.77392e(k-3) +y(k-7)-y(k-5) -0.38696u(k-5)	120.9707
4559	1.20992e(k-1)-0.77392e(k-3) +y(k-7)-y(k-5) -0.38696u(k-5)	116.2026
4626	1.17075e(k-1)-0.77392e(k-3) +y(k-7)-y(k-5) -0.38696u(k-5)	110.2520
4767	1.21825e(k-1)-0.77392e(k-3) +y(k-7)-y(k-5) -0.38696u(k-5)	108.8732
5000	1.21825e(k-1)-0.77392e(k-3) +y(k-7)-y(k-5) -0.38696u(k-5)	108.8732

* The control law evolved before 4159th generation can't meet the constraints (stability/Overshoot/Rise time/Settling time), which is indicated by a large fitness. After 4159th generation, the GP found a feasible solution that can stabilize the whole interval plant system with fewer violations of other constraints; the best feasible solution is improved generation by generation until the termination criteria are met.

by generation. The best individual evolution history in GP is listed in Table. 1.

The evolved controller after 5000 generations can be written as follows:

$$\begin{aligned} U(k) = 0.38696U(k-5) + 1.2182R(k-1) - \\ - 0.77392R(k-3) - 1.2182Y(k-1) + \\ + 0.77392Y(k-3) - Y(k-5) + Y(k-7) \end{aligned} \tag{31}$$

For comparison purpose, the step responses of the closed-loop system with 16 Kharitonov plants with GP evolved controller and the optimal controllers obtained by genetic algorithm[18] are plotted in Fig. 4 and Fig. 5, respectively. It is obvious that the close-loop performance of the optimal controller designed by GP are better than the results by Hsu and Yu.[18] One can see that the settling time and the oscillations/overshoots in Fig. 4 are significantly reduced compared with that in

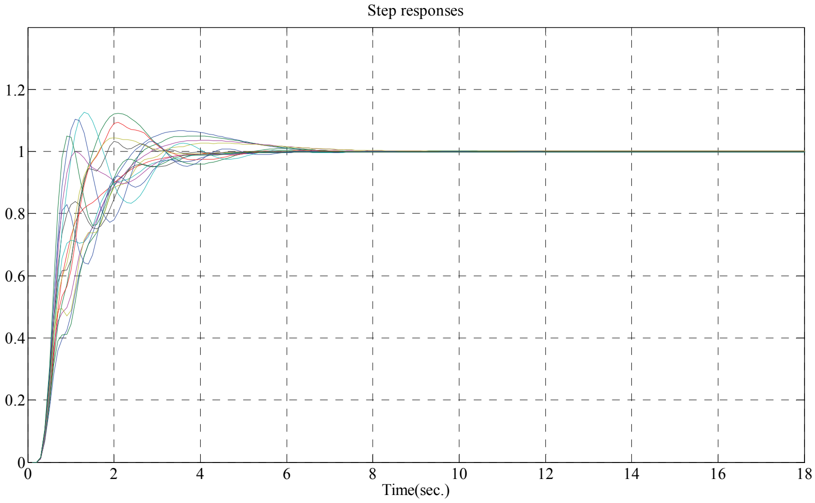

Fig.4. Step responses with all 16 Kharitonov plants by GP evolved controller

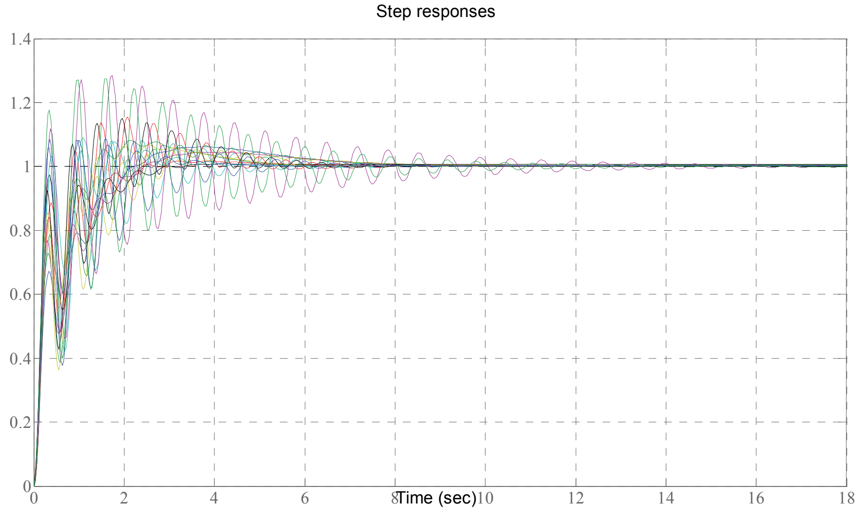

Fig.5. Step responses with all 16 Kharitonov plants by Hsu and Yu[18]

Table.2 Robust optimal controller performance and the worst-case plant related

Settling Time	Overshoot	Max ITAE	Related Worst-case System
≤5.1s	≤12.7%	1.5191	b_0=90, b_1=54, a_0=-0.1 , a_1=30.1, a_2=80.8, a_3=4.6,

Fig. 5. The modified ITAE index of close-loop system with evolved controller is listed in Table. 2.

To guarantee that the controller designed by GP can synchronously stabilize the whole interval plant family, the 32 subsets of interval plant family are integrated into GP as the stability criterion. Fig. 6 shows the roots distribution of the discrete system with the 32 subsets of interval plant family. All the roots in Fig. 6 are located within the unit circle, so the controller can synchronously stabilize the 32 subsets of interval plant family. According to the stability condition of Theorem 2, the robustness of the controller in Eq. (31) can be guaranteed.

Based on Monte-Carlo experiment, the robust performance of the evolved controller is further tested on 10000 randomly generated systems in the given parameter uncertainty regions described in Eq. (29) and Eq. (30). The step responses of closed-loop systems with the 10000 systems and the evolved controller are shown in Fig. 7. Interesting result appears that large and critical portions of the step-responses with the randomly generated systems are bounded by the 16 Kharitonov Plants in Fig. 4, which further validates that *the "worst case" of an interval plant family lies on or near the 16 Kharitonov plants*.

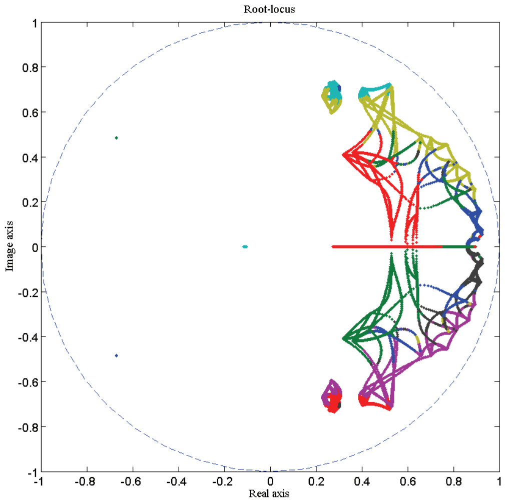

Fig.6. Roots distribution of the discrete system with the 32 subsets of interval plant family

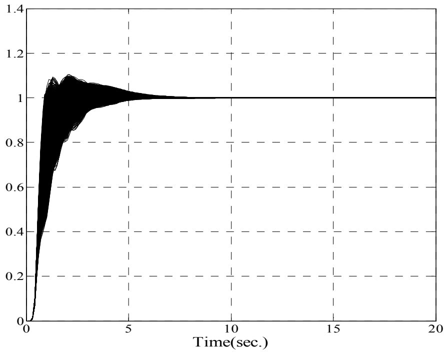

Fig.7. Closed-loop step responses of Monte-Carlo simulation

The Monte-Carlo simulation results also reveal that the robust optimal controller in Eq. (31) can stabilize the whole interval plant family while remaining satisfied performance.

7 Concluding Remarks

In this paper, a novel solution has been developed to automatically evolve a discrete robust optimal controller for interval plant family by integrating rigorous mathematical stability criteria into GP evolution, which is formulated as an optimization of time domain index on the worst cases revealed by Kharitonov Theorem. This solution extends previously published implementations of GPs with the following advanced features: (a) the generalized Kharitonov Theorem is integrated into the evolution to guarantee that the evolved controller can stabilize the whole interval plant family, which is never included in the previous work. (b) a discrete form controller, which reduces the function set considerably, is helpful to accelerate the evolution, and more importantly, prevent some undesired results like saturation caused by using complex operations (such as high order derivative blocks).

There is no restrictive condition under which the proposed approaches are developed. Conventional design constraints on the higher-order interval plants and controller order are therefore removed. Simulation results show that the resulting robust optimal controller can provide optimal performance while maintaining its robustness.

Although the evolution process is very time consuming for Genetic Programming compared with other computational intelligence methods, the capability of automatic design for the controller structure and the parameters makes it more attractive for academic researches and applications. Our work hints of several potential directions for future research, we believe that the results could be extended to multi-inputs multi-outputs (MIMO) systems with parameter uncertainty. There is a wide range of additional performance indices (time domain and frequency domain) that could be included in the GP framework for designing a robust optimal controller. It should be emphasized that the applications of control domain knowledge in automatic design of robust controllers with GP will be the key to success in our research and applications.

References

1. B. R. Barmish, C. V. Hollot, F. J. Kraus and R. Tempo, Extreme point results for robust stabilization of interval plants with first order compensators, *IEEE. Trans. Automat. Contr.* 37(6) (1992) 707-714.
2. B. R. Barmish and H. I. Kang, Extreme point results for robust stability of interval plants: beyond first order compensators, *Automatica*. 28(6) (1992) 1169-1180.
3. S. P. Bhattacharyya, H. Chapellat and L. Keel, *Robust Control-The Parametric Approach* (Prentice-Hall, New Jersey, 1995).
4. D. Henrion and O. Bachelier, Low-order robust controller design for interval plants, Int. J. Control. 74(1) (2001) 1-9.
5. N. Tan and D. P. Atherton, Stability and Performance Analysis in an Uncertain World, *Comput. Control. Eng. J.* 11(2) (2000) 91-101.
6. L. Grman, D. Rosinová, V. Veselý and A. Kozáková, Robust stability conditions for polytopic systems, *Int. J. Syst. Sci.* 36(15) (2005) 961-973.
7. V. L. Kharitonov, Asymptotic stability of an equilibrium position of a family of systems of linear differential equations, *Different. Equat.* 14 (1979) 1483-1485.
8. A. C. Bartlett, C. V. Hollot and H. Lin, Root Locations of an Entire Polytope of Polynomials: It Suffices to Check the Edges, *Math. Contr. Signals. Syst.* 1(1) (1988) 61-71.
9. S. P. Bhattacharyya and H. Chapellat, A generalization of Kharitonov's theorem: Robust stability of interval plants, *IEEE. Trans. Automat. Contr.* 34 (1989) 306-311.
10. H. Chapellat, M. Dahleh and S. P. Bhattacharyya, Robust stability under structured and unstructured perturbations, *IEEE. Trans. Automat. Contr.* 35(10) (1990) 1100-1108.
11. S. J. An, H. Lin and E. Wang, On the parametric H infinity problems of weighted interval plants, *IEEE. Trans. Automat. Contr.* 45(2) (2000) 332-335.
12. C. V. Hollot and R. Tempo, On the Nyquist envelope of an interval plant family, *IEEE. Trans. Automat. Contr.* 39(2) (1994) 391-396.
13. N. Tan, Robust phase margin, robust gain margin and Nyquist envelope of an interval plant family, *Comput. Electr. Eng.* 30(2) (2004) 153-165.

14. A. C. Bartlett, A. Tesi, and A. Vicino, Frequency response of uncertain systems with interval plants, *IEEE. Trans. Automat. Contr*. 38(6) (1993) 929-933.
15. K. B. Datta and V. V. Patel, H infinite-based synthesis for a robust controller of interval plants, *Automatica*. 32(11) (1996) 1575-1579.
16. Y. J. Huang and Y. J. Wang, Robust PID tuning strategy for uncertain plants based on the Kharitonov theorem, *ISA. T*. 39(4) (2000) 419-431.
17. C. C. Hsu, C. Y. Yu and S. C. Chang, Multiobjective Evolutionary Approach to the Design of Optimal Controllers for Interval Plants via Parallel Computation, *IEICE. Trans. Fund. Electron. Comm. Comput. Sci.* E89-A(9) (2006) 2363-2373.
18. C. C. Hsu and C. Y. Yu, Design of optimal controller for interval plant from signal energy point of view via evolutionary approaches, *IEEE. Trans. Syst. Man. Cybern. Part. B*. 34(3) (2004), 1609-1617.
19. P. J. Fleming and R. C. Purshouse, Evolutionary algorithms in control systems engineering: A survey, *Control. Eng. Pract*. 10(11) (2002) 1223-1241.
20. A. Soltoggio, A Comparison of Genetic Programming and Genetic Algorithms in the Design of a Robust Saturated Control System. *In Proc Int. Conf. Genetic and Evolutionary Computation,* 2004, 3103, pp.174-185.
21. C. C. Sun, H. Y. Chung and W. J. Chang, GA-based robust S-stability output feedback controller design algorithm with hierarchical fitness function structure, *In IEEE Int. Conf. Neural Networks and Signal Processing,* 2003, 1, pp.444-449.
22. P. Chen, Y. Z. Lu and Y. W. Chen, Extremal Optimization Combined with LM Gradient Search for MLP Network Learning, *Int. J. Comput. Intell. Syst.* 3(5) (2010) 622-631
23. K. Y. Chan, C. K. Kwong and T. C. Fogarty, Modeling manufacturing processes using a genetic programming-based fuzzy regression with detection of outliers, *Inform. Sciences*. 180(4) 2010 506-518.
24. B. Grosman and D. R. Lewin, Automated Nonlinear Model Predictive Control using Genetic Programming, *Comput. Chem. Eng*. 26 (2002) 631-640.
25. J. R. Koza, M. A. Keane, J. Yu, F. H. Bennett III, W. Mydlowec and O. Stiffelman, Automatic synthesis of both the topology and parameters for a robust controller for a non-minimal phase plant and a three-lag plant by means of genetic programming. *In Proc IEEE. Conf.* Decision and Control, 1999, 5, pp.5292-5300.
26. J. R. Koza, M. A. Keane, J. Yu, F. H. Bennett III and W. Mydlowec, Automatic creation of human-competitive programs and controllers by means of genetic programming, *Genetic Programming and Evolvable Machines*. 1 (2000) 121-164.
27. E. Alfaro-Cid, E. W. McGookin, D. J. Murray-Smith and T. I. Fossen, Genetic programming for the automatic design of controllers for a surface ship, *IEEE. T. Intell. Transp*. 9(2) (2008) 311-321.
28. J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (MIT Press, Cambridge, Massachusetts, 1992).
29. J. R. Koza, M. A. Keane and M. J. Streeter, What's AI Done for Me Lately? Genetic Programming's Human-Competitive Results, *IEEE. Intell. Syst*. 18(3) (2003) 25-31.
30. K. Ogatta, *Modern Control Engineering* (Prentice Hall, New Jersey, 1990).
31. R. C. Dorf, *Modern Control Systems* (Addison Wesley, Menlo Park, California, 1989).