Energy Conversion and Management 84 (2014) 390–404

Contents lists available at ScienceDirect

Energy Conversion and Management

journal homepage: www.elsevier.com/locate/enconman

Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II

Zhihuan Chen, Xiaohui Yuan *, Bin Ji, Pengtao Wang, Hao Tian

School of Hydropower and Information Engineering, Huazhong University of Science and Technology, 430074 Wuhan, China

ARTICLE INFO

Article history:
Received 2 January 2014
Accepted 16 April 2014

Keywords:
Fractional-order PID controller
NSGAII
Hydraulic turbine regulating system
Iterative chaotic map with infinite collapses (ICMIC)

ABSTRACT

Fractional-order PID (FOPID) controller is a generalization of traditional PID controller using fractional calculus. Compared to the traditional PID controller, in FOPID controller, the order of derivative portion and integral portion is not integer, which provides more flexibility in achieving control objectives. Design stage of such an FOPID controller consists of determining five parameters, i.e. proportional, integral and derivative gains {*Kp*, *Ki*, *Kd*}, and extra integration and differentiation orders $\{\lambda, \mu\}$, which has a large difference comparing with the conventional PID tuning rules, thus a suitable optimization algorithm is essential to the parameters tuning of FOPID controller. This paper focuses on the design of the FOPID controller using chaotic non-dominated sorting genetic algorithm II (NSGAII) for hydraulic turbine regulating system (HTRS). The parameters chosen of the FOPID controller is formulated as a multi-objective optimization problem, in which the objective functions are composed by the integral of the squared error (ISE) and integral of the time multiplied squared error (ITSE). The chaotic NSGAII algorithm, which is an incorporation of chaotic behaviors into NSGAII, is used as the optimizer to search true Pareto-front of the FOPID controller and designers can implement each of them based on objective functions priority. The designed chaotic NSGAII based FOPID controller procedure is applied to a HTRS system. A comparison study between the optimum integer order PID controller and optimum fractional order PID controller is presented in the paper. The simulation and some experimental results validate the superiority of the fractional order controllers over the integer controllers.

1. Introduction

Hydraulic turbine regulating system (HTRS) is a complex system which is influenced by hydraulic, mechanical and electrical factors. In this system, turbine governor is the controller and hydroelectric generating unit (HGU) is the controlled object. Thus the close loop control system's performance will be directly related to the safety of the power plant and the power quality sending to the grid.

Traditionally the proportional–integral–derivative (PID) controller has been the most common used in HTRS system loop due to its simplicity and ease of implementation [1–6]. Recently the increasing interests in enhancing the performance of PID controller by using the concept of fractional calculus has led to considerable attention toward to the fractional order PID controller (FOPID or $PI^{\lambda}D^{u}$ controller), in which the order of derivatives and integrals is non-integer, i.e. λ and μ are the integrative and derivative orders and they are not limited to integer number but fractional number. This new type controller proposed by Podlubny [7] is characterized by five parameters namely the proportional gain, the integration gain, the derivative gain, the integration order and the derivative order. Over the past decades, the FOPID controllers have been applied by researchers in different fields of engineering and industries around the world, such as in irrigation canal control [8], motion control of DC motor [9–11], automatic voltage regulator [12–14], aerospace designing control system [15], weapon system [16], wind energy system [17], and nuclear reactor [18], and most of the above mentioned research results show that the FOPID controller has a better performance and robustness than the traditional PID controller. Although it is so, the parameter optimization of FOPID controller is still an important and challenge issue until now. Comparing to the conventional used parallel PID controller, the FOPID controller has two extra parameters. On one hand, it enables more degrees of freedom to design the controller, and on the other hand, it means that there is more complex in the

* Corresponding author. Tel.: +86 18907198929.
E-mail address: yxh71@163.com (X. Yuan).

http://dx.doi.org/10.1016/j.enconman.2014.04.052
0196-8904/

Nomenclature

FOPID:

Kp	the proportional gain of FOPID controller
Ki	the integral gain of FOPID controller
Kd	the differential gain of FOPID controller
λ	the integration order of FOPID controller
μ	the differentiation order of FOPID controller
s	the Laplace operator

HTRS:

c	turbine speed relative deviation, p.u.
σ	the output of control signal, p.u.
y	wicket gate stroke relative deviation, p.u.
x	turbine speed relative deviation, p.u.
mt	turbine torque relative deviation, p.u.
mg	load torque relative deviation, p.u.
q	flow rate relative deviation, p.u.
h	water head relative deviation, p.u.
Ty	major relay connecter response time, second
Ta	generator mechanical time, second
ex	first-order partial derivative value of torque with respect to turbine speed
$c(s)$	Laplace transform of c, p.u.
$\sigma(s)$	Laplace transform of σ, p.u.
$y(s)$	Laplace transform of y, p.u.
$x(s)$	Laplace transform of x, p.u.
$mt(s)$	Laplace transform of mt, p.u.
$mg(s)$	Laplace transform of mg, p.u.
$q(s)$	Laplace transform of q, p.u
$h(s)$	Laplace transform of h, p.u.
Tw	water starting time constant, second
eg	generator load self-regulation parameters
ey	first-order partial derivative value of torque with respect to wicket gate
eh	first-order partial derivative value of torque with respect to water head
eqx	first-order partial derivative value of flow rate with respect to turbine speed
eqy	first-order partial derivative value of flow rate with respect to wicket gate
eqh	first-order partial derivative value of flow rate with respect to water head

The Chaotic NSGAII:

NSGAII	non-dominated sorting genetic algorithm II
SBX	simulated binary recombination (crossover) operator
ICMIC	iterative chaotic map with infinite collapses
N	maximum population size of the algorithm
η_c	given cross probability of the algorithm
DR	domination rank operator
CD	crowding distance operator
PM	polynomial mutation operator
Maxgen	maximum generation of the algorithm
η_m	given mutation probability of the algorithm

synthesis of the controller. A number of approaches have been tested and documented to tune the parameters of the FOPID controller. In [19], a kind of analytic method is proposed to design the FOPID controller by expanding the control loop signal and reference model input and output over a piecewise orthogonal functions. In [20], a set of tuning rules were devised based on a first order plus dead-time model of the process by minimizing the integrated absolute error with a constraint on the maximum sensitivity. In [21], the author proposed to optimize the parameters of FOPID controller by taking into account five conditions about phase and gains margins specifications and constrains over the sensitivity objectives. Intelligence evolutionary algorithms have also been introduced into solving the optimization problem of FOPID controller. In [14], the parameters adjustment of fractional order PID controller has been obtained based on particle swarm optimization (PSO) method by searching the defined available space. In [22], an improved electromagnetism-like algorithm mechanism with genetic algorithm (IEMGA) technique is used for the gains design of FOPID controller throughout minimizing the objective errors. In [23], the FOPID controller has been designed based on the root locus method with an improved differential evolution algorithm version.

However, a major of existing researches are focused only on a single objective designing, but in a practical control system, design multiple objectives need to be addressed for the reason that various optimal desired indicators of control system output are always conflicting and there need a trade-off between these objectives. In [12,13,24], the authors tried to apply the multi-objective optimization methodology to balance the contradictory indicators and results show that there obtained a very good result. Inspired by this thought, an improved evolutionary multi-objective intelligence algorithm, the non-dominated sorting genetic algorithm II (NSGAII) [25], augmented with the iterative chaotic map with infinite collapses (ICMIC) [26], is used in this paper to design the FOPID controller in HTRS system with contradictory objectives. As aforementioned, the dynamic response performance of the HTRS is crucial for the stability and robustness of power grid system, which is often subjected to the severe frequency disruption and load disturbance, thus fast and accurate desired-point tracking and strong robustness are the key factors in determining performance objective functions. In the present paper, two performance objective functions, namely the integral of squared error (ISE) and the integral of the time multiplied squared error (ITSE) are considered, in which ISE-based system always has a response with a small overshoot percentage but long settling time and ITSE-based system has a response with a shorter settling time but without stability margin [14]. The trade-offs between two chosen objectives generates a set of best compromise solutions. This set of solutions give the designer an idea of what he can expect out of the controller with respect to different performance indices prior. When making the final decision choice for the controller, he has to choose one controller parameter series from the set depending on which objective function is more important in the current design. Thus in a particular case, e.g. let's say that strong robustness is essential and there can tolerate a long settling time. In such a case, the system designer can choose a solution on the Pareto front which might go for a more "safer" design than better desired-point tracking. This in effect is actually assigning more importance (weight percentage) to the objective of ISE than ITSE.

The rest of paper is organized as follows: Section 2 briefly introduces the concept of fractional calculus and the application of it in FOPID controller. In Section 3, the need of control system for multi-objective optimization, the fundamentals of multi-objective optimization, the description of HTRS system, and the chosen multi-objective functions is discussed in detail. The chaotic NSGAII is constructed in Section 4. Section 5 illustrates the simulation results along with a few discussions. The conclusions are summarized in the Section 6. Finally, acknowledgements are given.

2. Fractional calculus and the fractional order PID (FOPID)

2.1. Theory of fractional calculus

Fractional calculus is a branch of mathematical analysis that studies the possibility of taking real number power of the differential operator and integration operator, which extends the integer order differentiation and integration to the entire available real number order fundamental operator ${}_aD_t^\alpha$. The continuous integro-differential operator is defined as follows:

$$ {}_aD_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha}, & \alpha > 0 \\ 1, & \alpha = 0 \\ \int_a^t (d\tau)^\alpha, & \alpha < 0 \end{cases} \tag{1} $$

where a and t are the lower and upper limits and $\alpha(\alpha \in R)$ is the order of the operation. There are several definitions for fractional derivatives. The most usual definition is introduced by Caputo [12] that generalizes the following definitions corresponding to integer orders:

$$ {}_aD_t^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)} \int_a^t \frac{f^m(\tau)}{(t-\tau)^{\alpha+1-m}} d\tau, \quad \alpha \in R, m \in Z, m-1 \leqslant \alpha < m \tag{2} $$

The Laplace transform of Caputo fractional derivative equation is:

$$ \int_0^\infty e^{-st} D^\alpha f(t) dt = s^\alpha F(s) - \sum_{k=0}^{m-1} s^{\alpha-k-1} f^k(0) \tag{3} $$

where m is the smallest integer which is larger than α, $D^\alpha = {}_0D_t^\alpha$ is the Caputo's fractional derivative of order α, $\Gamma(\alpha) = \int_0^t e^{-t} t^{\alpha-1} dt$ is the Gamma function and $F(s) = \int_0^\infty e^{-st} f(t) dt$ is the Laplace transform of $f(t)$.

To implement the fractional order transfer functions in simulation or practical studies, one most common way is to approximate them with integer order transfer functions. To perfectly approximate a fractional transfer function with an integer order number, the integer order transfer function has to include an infinite numbers of zeroes and poles.

One of the well-known approximations is caused by Oustaloup [27] who uses recursive distribution of N pole points and N zero points of the following form, representing a higher order analog filter.

$$ s^\alpha = K \prod_{k=-N}^{N} \frac{s + \omega_k'}{s + \omega_k} \tag{4} $$

where the poles, zeros, and gain of the filter can be recursively evaluated as:

$$ \omega_k = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+N+\frac{1}{2}(1+\alpha)}{2N+1}}, \quad \omega_k' = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+N+\frac{1}{2}(1-\alpha)}{2N+1}}, \quad K = \omega_h^\alpha \tag{5} $$

where α is the order of the differ-integration, $(2N+1)$ is the order of the filter and (ω_b, ω_h) is the expected fitting range. Any signal $f(t)$ can be passed through the filter (4) and the output of the filter can be regarded as an approximation to fractionally differentiated or integrated signal $D^\alpha f(t)$.

In the present paper, the Oustaloup's band-limited frequency domain approximation technique is used because of its easy implement in hardware using higher order infinite impulse response (IIR) analog or digital filters. An 5*th* order Oustaloup's approximation is done for the integro-differential operators within a

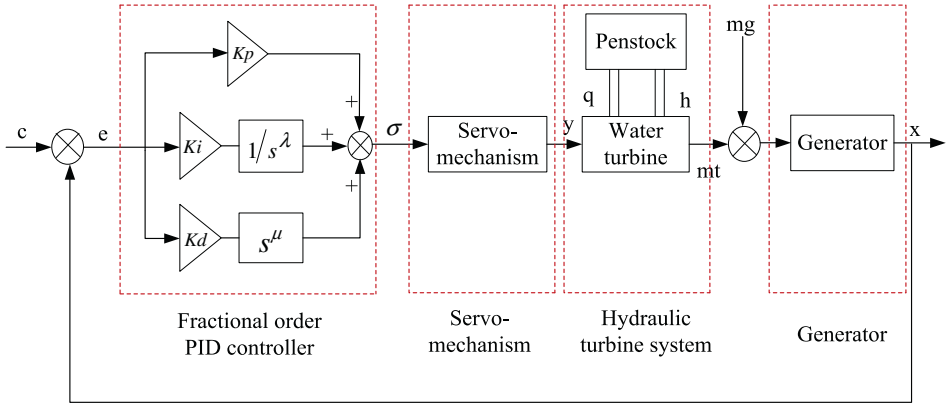

Fig. 1. Structure of hydraulic turbine governing system (HTRS) with FOPID controller.

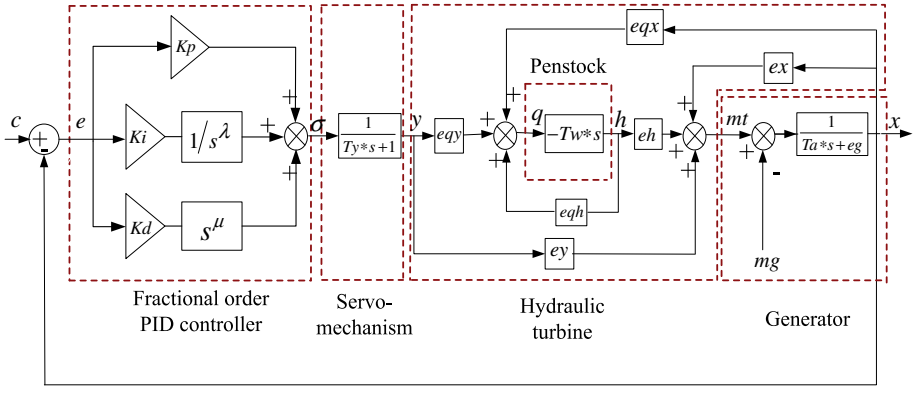

Fig. 2. The frame diagram of transfer functions of HTRS system with FOPPID controller.

Table 1
Transfer parameters in water turbine and generator under two running condition.

Running condition	Transmission coefficients in the water turbine and generator system							
	ex	*ey*	*eh*	*eqx*	*eqy*	*eqh*	*Ta*	*eg*
Unload	−1.0567	0.9080	1.4191	−0.0574	0.7887	0.4571	12	0.45
Load	−1.2481	1.3130	1.3028	−0.1035	1.0045	0.3843	8.5	0.65

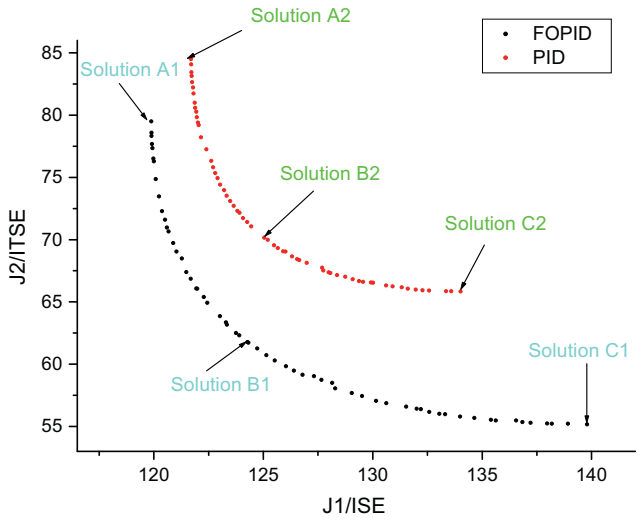

Fig. 3. Pareto front of the objectives J_1 and J_2 for the controllers under unload condition.

frequency band of the constant phase elements (CPEs) as $w \in \{10^{-2}, 10^{2}\}$ rad/s.

2.2. Basic concepts of FOPID controller

FOPID controller is an application of fractional calculus theory in PID controller, the differential equation of it in time domain is described by [23]:

$$u(t) = Kp * e(t) + Ki * D_i^{-\lambda} e(t) + Kd * D_i^{\mu} e(t) \tag{6}$$

The continuous transfer function of the fraction order PID controller is obtained through the Laplace transform is shown as follows [13]:

$$C(s) = Kp + \frac{Ki}{s^{\lambda}} + Kd * s^{\mu} \tag{7}$$

It is obvious that the FOPID controller nor only contains conventional proportional, integral and derivative gains {*Kp*, *Ki*, *Kd*}, but also owns additional integration and differentiation orders $\{\lambda, \mu\}$, which is adjustable parameters that gives more possibility to realize the desired control performance. While $\lambda = 1$ and $\mu = 1$, the FOPID controller structure is reduced to the classical PID controller.

3. Multi-objective optimization framework for FOPID controller design in HTRS system

3.1. The need for multi-objective optimization in controller design

The question of why multi-objective optimization is required for controller designing problems is enunciated explicitly in [28], it states that the key concept of different synthesis techniques, like the H_2, H_∞ or L_1 control is that the design objectives can be satisfied by minimizing a suitable weighted norm of some characteristic closed-loop properties in the system. However, each norm has its own particular feature and only minimizing that norm to ensure that the control system satisfies that criteria well, but it does not say anything about the other design specifications. For example, minimizing H_∞ norm gives closed loop robust stability, but this norm only working in frequency domain method and cannot deal with time domain specifications; minimizing H_2 norm implies closed loop stabilization in the presence of disturbances, but the obtained controller always present an arbitrary robustness. In a practical control system design, designers should design a system which ideally should have both properties to some extent. Hence there need a variety of objective functions to reflect the different performance specifications and a suitable multi-objective evolutionary algorithm is essential for the controller design.

3.2. Fundamentals of multi-objective optimization

Generally, the multi-objective optimization problems contain several objectives to be minimized and (optional) constraints to be satisfied. In this case, optimization problems consist of optimizing a vector of functions which differs from a single-objective optimization problem and shown as [29]:

$$\begin{aligned} &\text{Minimize} && F(x) = (f_1(x), \quad f_2(x), \cdots, f_k(x)) \\ &\text{Subject to}: && g_i(x) \leqslant 0 \qquad \forall i \in [1, p] \\ &&& h_j(x) = 0 \qquad \forall j \in [1, q] \end{aligned} \tag{8}$$

where $x \in \Omega$ and Ω is the decision space, R^n is the objective space, $F{:}\Omega \to R^n$ consists of k real valued objective functions, $f_i(x)$ is the *ith* objective function. $g_i(\cdot)$ and $h_j(\cdot)$ are the optional p number of inequality and q is number of equality constraints on the problem.

These functions $f_1(x), f_2(x), \cdots, f_k(x)$, usually in conflict with each other, are a mathematical description of performance criteria. There is few encounters that a vector of decision variables that optimizes all the objectives simultaneously. So the concept of Pareto optimality is used [30].

Table 2
Representative solutions on the Pareto front under unload condition.

Controller structure	Solution number	J_1/ISE	J_2/ITSE	K_p	K_i	K_d	λ	μ
FOPID	A1	119.87870	79.49121	6.90148	0.42649	3.72009	1.21260	1.20213
	B1	124.27184	61.78834	7.89817	0.52561	4.52804	1.18724	1.18203
	C1	139.79517	55.16297	9.19952	0.58708	5.13400	1.19846	1.18461
PID	A2	121.69633	84.49785	5.56815	0.74778	4.33028	–	–
	B2	124.44038	71.06710	6.30182	0.84016	4.92944	–	–
	C2	134.00850	65.83614	7.01479	0.95068	5.59556	–	–

Definition 1 (*Pareto dominance*). A solution vector $x^{(1)}$ is said to dominate the other solution vector $x^{(2)}$, i.e. $x^{(1)} \prec x^{(2)}$, if both statement below are satisfied.

(1) The solution vector $x^{(1)}$ is no worse than vector $x^{(2)}$ in all objectives, or $f_j(x^{(1)}) \leqslant f_j(x^{(2)})$ for all $j \in \{1, 2, \cdots, k\}$.

(2) The solution vector $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective, or $f_j(x^{(1)}) < f_j(x^{(2)})$ for at least one $j \in \{1, 2, \cdots, k\}$.

Definition 2 (*Pareto optimality*). For a given multi-objective problem, a solution $\vec{f}^* \in \Omega$ is the Pareto optimality if, and only if there is no $\vec{f} \in X$ that dominates $\vec{f}^*$.

Definition 3 (*Pareto set*). A Pareto set P^* is a set in the decision variable space consisting of all the Pareto optimality vectors, $P^* = \{x \in \Omega | \neg \exists x' \in \Omega : F(x') \prec F(x)\}$. In other words, there is no other x' in Ω that dominates any $x \in P^*$.

Definition 4 (*Pareto front*). The plot of the objective function whose non-dominated vectors are in the Pareto set is called the Pareto front. This implies that no other feasible objective vector exists in Ω which can improve one objective function in $F(x)$ without simultaneous worsening of some other objective functions in $F(x)$.

3.3. Description of the HTRS system

HTRS is a complicated system, mainly contains four parts, namely speed governor controller, electro-hydraulic

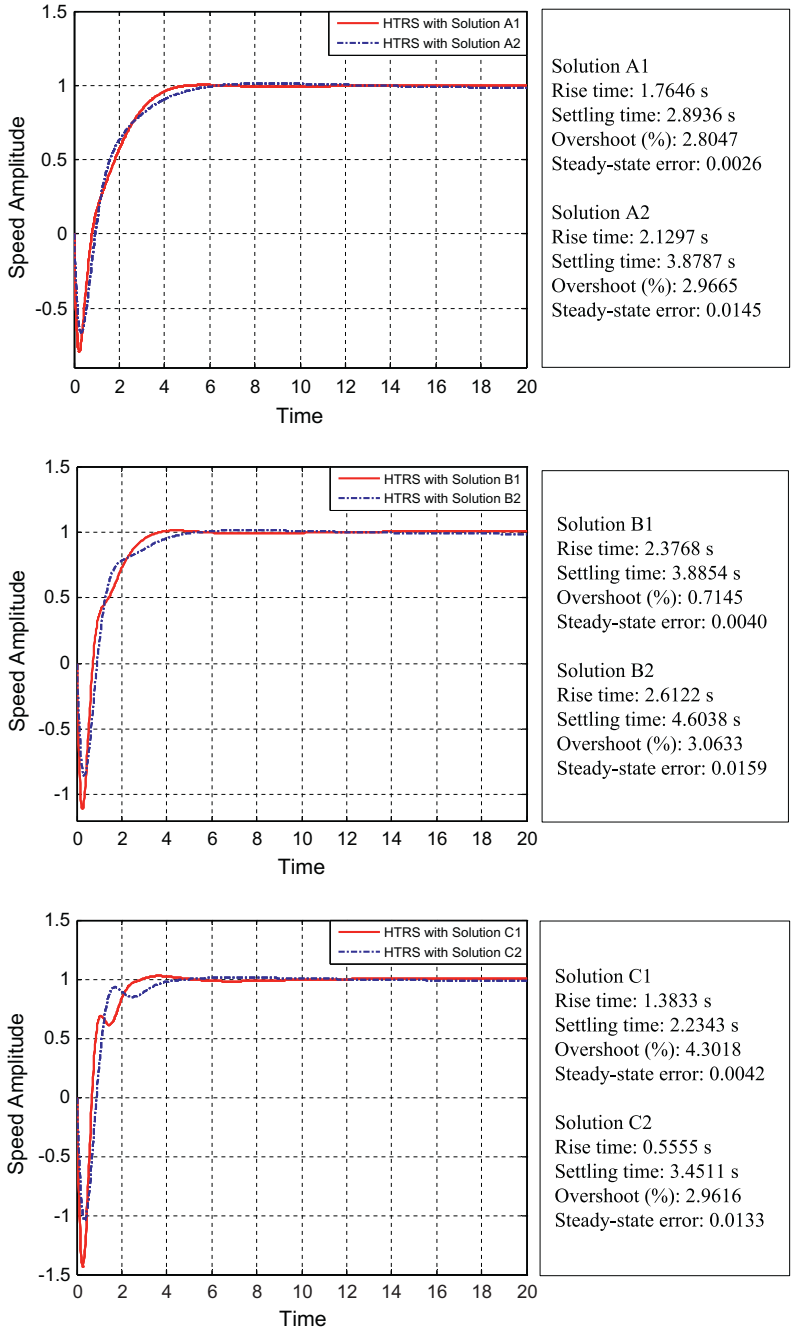

Fig. 4. Set point tracking response for representative solutions as reported in Table 2.

servomechanism, hydraulic turbine and penstock, and generator. Fig. 1 shows the schematic diagram of the HTRS system with the fractional order PID controller. As seen in this figure, HTRS is a control system of hydropower generator sets that governs the speed of turbine according to the set point of output power and set point of speed. The main task of HTRS is to adjust the power output to grid and to track the frequency of grid, thus in order to ensure the safety of power grid, an excellent dynamic performance of HTRS system is essential. To analyze the dynamic performance of HTRS system, transfer functions of these components are represented as follows [1,2,31,32]:

(1) Electro-hydraulic servomechanism model: servomechanism is the actuator of hydraulic turbine. It is made up by the major relay connecter and auxiliary relay connecter in HTRS. Generally, the parameter of auxiliary relay connecter is far less than the parameter of major relay connecter. So the model can be simplified as a one-order system and the transfer function is shown as follows:

$$G_y(s) = \frac{y(s)}{\sigma(s)} = \frac{1}{Ty * s + 1} \tag{9}$$

(2) Hydraulic turbine system model: hydraulic turbine system is the key component in HTRS, and it is a very complicated system, there is not any analytic expression to describe this system until now. In generally, it is divided into two subsystems, namely water turbine and penstock pipeline. As flow rate is nonlinear to pressure, it is difficult to describe the movement laws of fluid in the penstock. However, neglect-

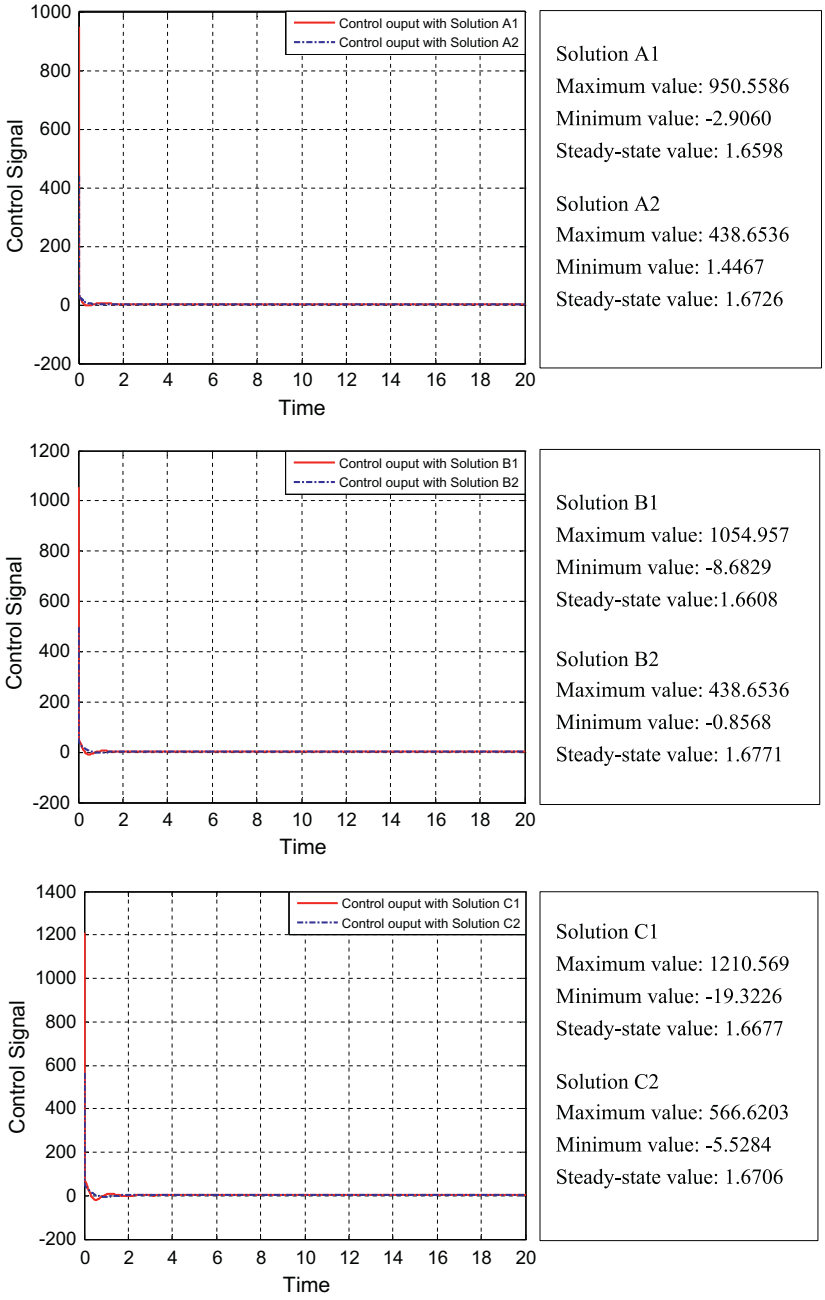

Fig. 5. The control output for representative solutions as reported in Table 2.

ing plant parameter changes and water column elasticity effect in the penstock, the transfer function of inelastic water hammer could be expressed as:

$$G_h(s) = \frac{h(s)}{q(s)} = -Tw * s \tag{10}$$

The nonlinear characteristic of Francis-turbine is described as follows:

$$\begin{cases} mt = f_1(x,\ y,\ h) \\ q = f_2(x,\ y,\ h) \end{cases} \tag{11}$$

For a small variation around some steady-state working conditions, it often uses the following equations approximating the model:

$$\begin{cases} mt = ex * x + ey * y + eh * h \\ q = eqx * x + eqy * y + eqh * h \end{cases} \tag{12}$$

Six transmission coefficients are defined as follows:

$$\begin{aligned} ex &= \tfrac{\partial mt}{\partial x}, \quad ey = \tfrac{\partial mt}{\partial y}, \quad eh = \tfrac{\partial mt}{\partial h} \\ eqx &= \tfrac{\partial q}{\partial x}, \quad eqy = \tfrac{\partial q}{\partial y}, \quad eqh = \tfrac{\partial q}{\partial h} \end{aligned} \tag{13}$$

(1) Generator system model: the generator model used in the simulation is the well-known Park first-order model and the transfer function is as follows:

$$G_g(s) = \frac{x(s)}{mt(s) - mg(s)} = \frac{1}{Ta * s + eg} \tag{14}$$

At present, PID controller is the most often used governor in HTRS system. In this paper, the FOPID controller is used in HTRS system instead of PID controller. The transfer functions of HTRS compensated with FOPID controller is shown in Fig. 2. It is noted that the hydraulic turbine system and generator system are all time-varying systems, in which the parameters associated are varying under different running conditions. In this paper, the parameters in hydraulic turbine system and generator system under unload and load running conditions in a Chinese hydropower plant have been measured and used in the later simulation experiments. The effectiveness of FOPID controller for HTRS system will be verified throughout experimental results.

3.4. Conflicting objectives: Trade-off between different performance criteria

Optimization of FOPID controllers firstly needs to design the optimization goal. It is desirable to shorten settling time and damp oscillations of the response in HTRS simultaneously with the power of FOPID controller, thus two different objective functions are considered for the optimization of controller parameters in the system. First is the Integral of the Squared Error (ISE) (J_1) and second is the Integral of the Time multiplied Squared Error (ITSE) (J_2), which are defined as follows:

$$J_1 = ISE = \int_0^\infty e^2(t)dt \tag{15}$$

$$J_2 = ITSE = \int_0^\infty te^2(t)dt \tag{16}$$

The first objective function J_1 tries to ensure fast tracking of desired set-point with a relatively small overshoot, and the second objective function J_2 gives heavy penalty to the errors occurring at later stages ensuring a shorter settling time. Both J_1 and J_2 must be minimized for effective operation of the control loop. It is well known that lower value of ISE makes the control system to act faster. Also with increase in speed, the accuracy becomes low, which implies there is prone to have oscillatory time response. These oscillations or overshoot is characterized by the presence of a higher ITSE value and vice versa.

4. Chaotic non-dominated sorting genetic algorithm II

Evolutionary algorithms have been widely used for multi-objective optimization due to that their parallel or population-based search approach properties are suited for these types of problems. The Pareto-based approach of NSGAII [25], as a kind of the most popular and excellent multi-objective evolutionary algorithms, has been used in a wide range of engineering multi-objective problems because of its simple and efficient non-dominance ranking procedure in yielding different levels of Pareto frontiers [33–35]. However, some latest researches have shown that although NSGAII has a better sorting scheme and incorporates elitism mechanism than the first NSGA version, it still falls short in maintaining lateral diversity and obtaining the Pareto-front with high uniformity [36], thus in the paper, chaotic map technique is incorporated to construct the chaotic NSGAII algorithm, intending to overcome these disadvantages. The hybrid algorithm is outlined by **Algorithm 1**.

Algorithm 1 The chaotic NSGAII algorithm

1. **[Start]** Generate a random population of N chromosomes within feasible search space.
2. **[Fitness]** Evaluate multiple fitness of each chromosome in the population.
3. **[Rank]** Rank the population by the following steps:
   3.1. **[Domination rank]** Rank individuals in population by using **Algorithm 2**.
   3.2. **[Crowding distance]** Calculate the crowding distance by using **Algorithm 3**.
4. **[New Population]** Create a new population by repeating the following steps:
   4.1. **[Selection]** Select parent chromosomes from previous population based on the crowding selection operator using **Algorithm 4**.
   4.2. **[Crossover]** Using simulated binary crossover operator and specified cross probability, do crossover of parents to form new offspring (children) by using **Algorithm 5**.
   4.3. **[Mutation]** Using polynomial mutation operator and specified mutation probability, do mutation of new offspring at each locus by using **Algorithm 6**.
   4.4. **[Chaos operation]** Executing chaotic mapping iteration of new offspring in **Algorithm 7**.
   4.5. **[Acceptance]** Place new offspring in the new population.
5. **[Replace]** Replace the old population by new generated population and continue.
6. **[Test]** If the end condition is satisfied (e.g. reaches a constant number of generations *Maxgen* in this paper), return Pareto set of solutions from current population and stop.
7. **[Loop]** Go to step 2.

In **Algorithm 1**, the algorithm of finding non-dominated solutions, the calculation of crowding distance, crowding selection operator, crossover operator, mutation operator and chaos operator are introduced below.

4.1. Domination rank (DR)

The main idea of DR assignment in algorithm can be summarized as follows [12]: the solution set not dominated by any other solutions in population is designated as the first front and are given the highest fitness value and rank number. These solutions are then excluded out and the second non-dominated front from the remaining individuals in population is created and ascribed the second highest fitness. This methodology is iterated until all the

solutions are assigned a fitness value. The law of DR assignment as required is shown in **Algorithm 2**.

Algorithm 2 DR assignment

1. Let rank counter r be zero.
2. Increase: $r = r + 1$.
3. Find the non-dominated individuals from population P based on the definition of domination.
4. Assign rank r to these individuals.
5. Remove these individuals from P and continue.
6. If P is empty then stop, else go to step 2.

4.2. Crowding distance (CD)

The CD rule is a measurement of the density of obtained solutions. The value of the CD presents an estimate of relative density of solutions surrounding a particular solution in the neighborhood [13]. Let a number of non-dominated solutions in Ω of size Z be given along with a number of objective functions f_k, $k = 1,2, \cdots \Psi$, where Ψ is the number of objective functions. d_i is the value of CD rule for the solution *ith*, Then the calculation method of CD is shown as follows by **Algorithm 3**.

Algorithm 3 CD calculation

1. Let $d_i = 0$ for $i = 1, 2, \cdots, Z$.
2. For each objective function f_k, $k = 1,2, \cdots \Psi$, sort the set in ascending order.
3. Let d_1 and d_Z be maximum values, e.g. $d_1 = d_Z = \infty$.
4. For $j = 2$ to $(Z - 1)$, set $d_j = d_j + \sum_{k=1}^{\Psi} \frac{f_k(j+1) - f_k(j-1)}{f_k(Z) - f_k(1)}$.

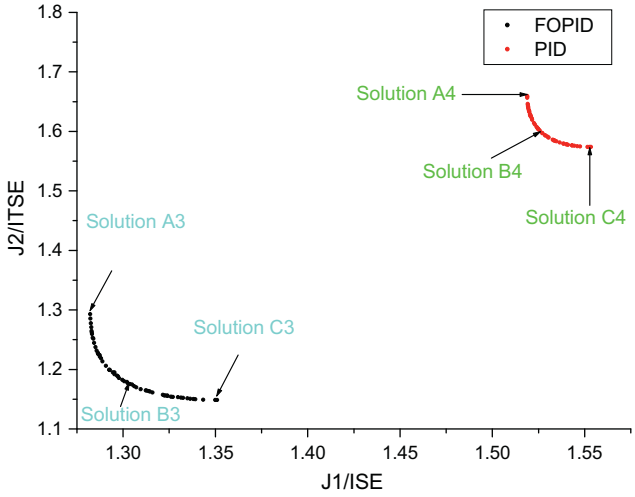

Fig. 6. Pareto front of the objectives J_1 and J_2 for the controllers under load condition.

4.3. Selection operator

In order to compare two individuals *xth* and *yth*, domination relationship operator (crowding tournament selection operator) $\succ$ is redefined. The solution *xth* dominates the solution *yth*, if either of the two conditions given below is satisfied [13].

I. Domination rank of *xth* is smaller than *yth*.
II. Domination ranks are equal and crowding distance of *xth* is larger than *yth*.

The chaotic NSGAII comparison criteria can be written as follows:

Algorithm 4 Crowding Selection Operator

$x \succ y$ iff $r_x < r_y$
or $r_x = r_y$ and $d_x > d_y$

4.4. Crossover operator

Simulated binary recombination (crossover) operator, i.e. SBX operator [37], is used in the study to combine two selected chromosomes in parents and creating two new children chromosomes. This operator is similar to one-point cut crossover in the binary data and is the most often used operator in the real-coded data of genetic algorithm (GA). The procedure of computing children offspring $children_1$ and $children_2$ from the chosen parents $parent_1$ and $parent_2$ is described by **Algorithm 5**:

Algorithm 5 SBX Operator

1. Calculate the difference between the objective functions of parents and children β_i with given cross probability η_c as follows (μ_i is a uniform random number generated between [0 1]):

$$\beta_i = \begin{cases} (2\mu_i)^{\frac{1}{\eta_c+1}} & \text{if } \mu_i \leqslant 0.5 \\ \left(\frac{1}{2(1-\mu_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{17}$$

2. Assign the children's values by the following equations:

$$\begin{cases} children_1 = 0.5 * [(1+\beta_i)parent_1 + (1-\beta_i)parent_2] \\ children_2 = 0.5 * [(1-\beta_i)parent_1 + (1+\beta_i)parent_2] \end{cases} \tag{18}$$

3. Judge whether the obtained value located within the available range [*Varmin Varmax*], if the value beyond, reset the chromosome within the available range.

4.5. Mutation operator

After the reproduction and crossover operators are applied, a mutation operator is used for the population. Polynomial mutation (PM) is a common used mutation operator in the multi-objective researches, which is proposed by Deb [38] in 1996, the implement of PM operator is shown in **Algorithm 6**.

Table 3
Representative solutions on the Pareto front under load condition.

Controller structure	Solution number	J_1/ISE	J_2/ITSE	K_p	K_i	K_d	λ	μ
FOPID	A3	1.28220	1.29311	10.1212	2.28312	2.99588	1.21017	1.27931
	B3	1.29777	1.18571	10.0642	2.64951	3.10748	1.17796	1.25293
	C3	1.35096	1.14873	10.2943	2.95314	3.17704	1.16592	1.23885
PID	A4	1.51885	1.65973	6.62847	3.23558	3.88998	–	–
	B4	1.52689	1.59777	6.82785	3.34285	3.83597	–	–
	C4	1.55342	1.57393	7.01744	3.48903	3.78235	–	–

Algorithm 6 PM Operator

1. Calculate the perturbation value δ_i with given mutation probability η_m as follows (r_i is a uniform random number generated between [0 1]):

$$\delta_i = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1, & \text{if } r_i < 0.5 \\ 1 - [2(1-r_i)^{1/(\eta_m+1)}], & \text{if } r_i \geqslant 0.5 \end{cases} \quad (19)$$

2. Assign the children's values by the following equations:

$$children = parent + \delta_i \quad (20)$$

3. Judge whether the obtained value located within the available range [*Varmin Varmax*], if the value beyond, reset the chromosome within the available range.

4.6. Chaos operator

Chaos is a kind of characteristic of nonlinear dynamic system which exhibits bounded dynamic unstable and ergodic behavior depended on initial value and control parameter. Chaos does not mean a total disorder, but a kind of phenomenon with elaborate structure inside [36,39], which has been widely used in intelligence algorithms as a local search to against premature and local optimal tracking convergence [40–43]. There are several chaotic map frequently used for local search, such as Logistic map, Henon map, and iterative chaotic map with infinite collapses (ICMIC). In the paper, ICMIC map is studied, the chaotic local search is shown in **Algorithm 7**.

ICMIC is proposed in [26], and it has infinite fixed points in comparison with the finite collapse one-dimensional maps. The ICMIC map is described by following equation:

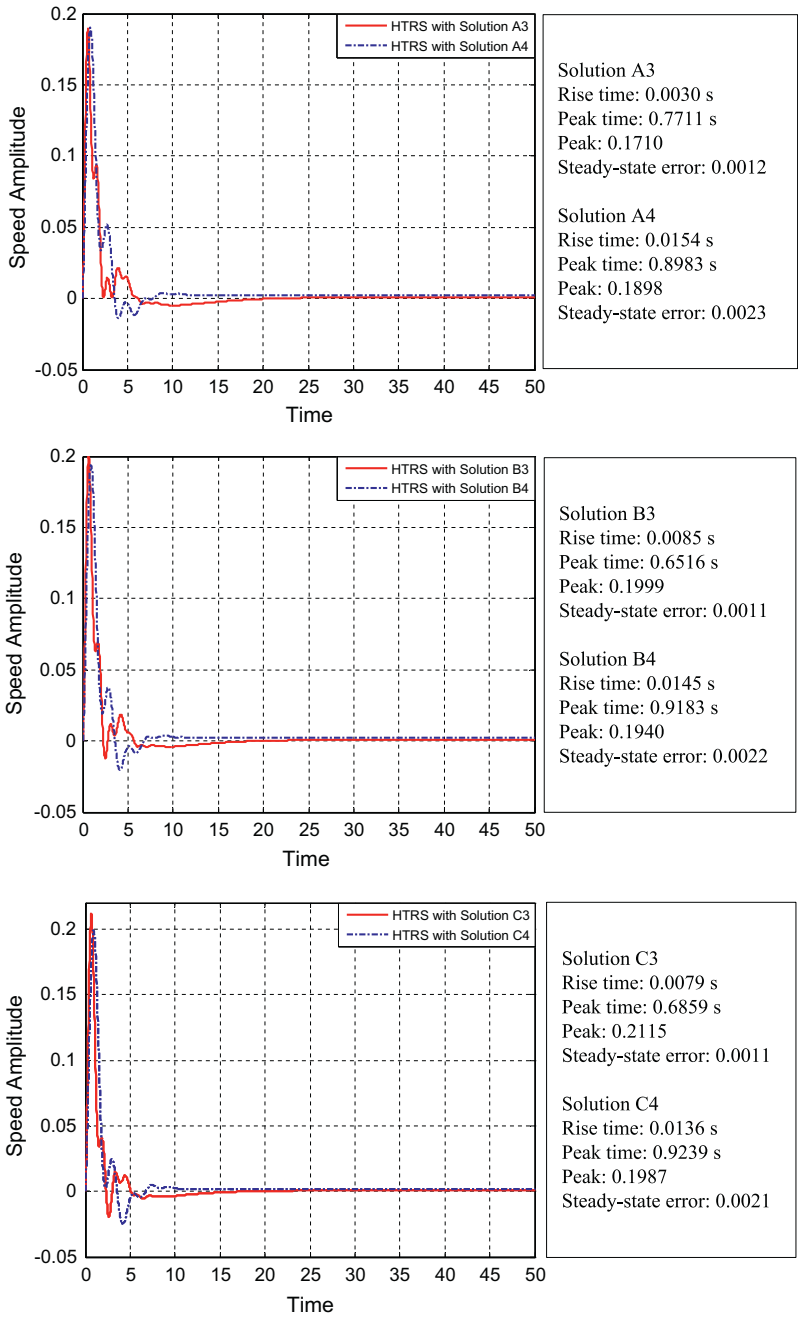

Fig. 7. Load disturbance rejection for representative solutions as reported in Table 3.

$$cx^{(k+1)} = sin\left(\frac{a}{cx^{(k)}}\right) \quad for\ a > 0,\ cx^{(k)} \in [-1 \quad 0) \cup (0 \quad 1] \tag{21}$$

where $cx^{(k)}$ means the parent chaos variables and $cx^{(k+1)}$ means the children chaos variables, a is a control variable, in this paper, $a = 2$.

As is known, the basic idea of searching optimum using chaos variables is: generating chaos variables with a kind of chaotic map, building chaos variables to optimize variables interval and then searching the optimal solution with the chaos variable using chaotic mapping. It is noted that the searching radius δ determines the searching scope, and a shrinking function with current iteration is always used to describe the acceleration of convergence as follows:

$$\delta(k+1) = \omega * \delta(k) \tag{22}$$

where ω is a positive number smaller than 1, in the present study, ω is set as 0.98.

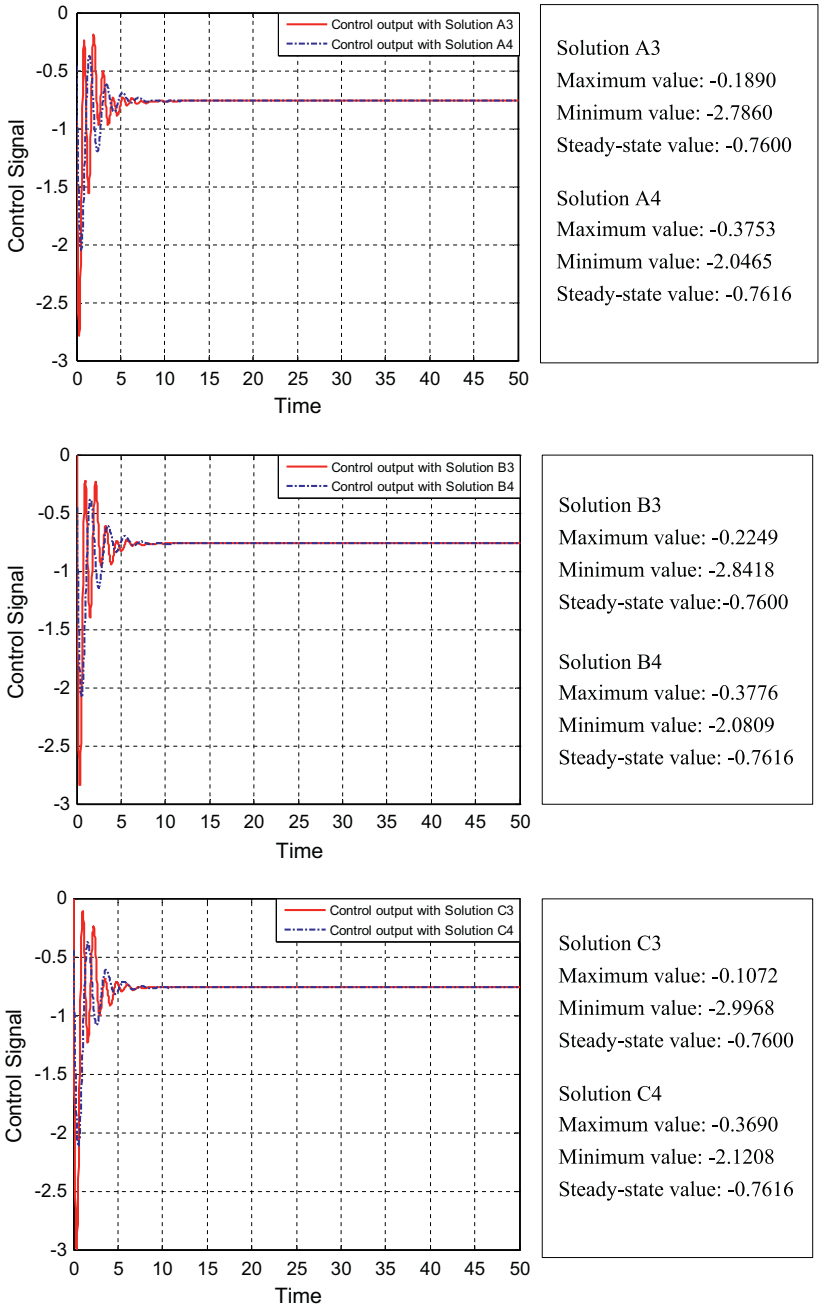

Fig. 8. The control output for representative solutions as reported in Table 3.

Algorithm 7 Chaotic Local Search

1. Let $k = 0$, randomly initialize n chaos variables $cx_i^{(k)} \in [-1 \quad 0) \cup (0 \quad 1]$, $i = 1, 2, \cdots, n$, Set the initial search radius $\delta(1)$ and variables interval $(x_{\min i} \quad x_{maxi})$, where $x_{\min i} = X_g - \delta_i/2$ and $x_{\max i} = X_g + \delta_i/2$ (X_g means the position of chosen chromosome).
2. Calculate the chaotic variables $cx_i^{(k+1)}$ for the next iteration according Eq. (17).
3. Build chaotic variables to optimize intervals $(x_{mini} \quad x_{maxi})$ and get optimization variables:

$$x_i^{(k+1)} = \frac{x_{maxi} + x_{\min i}}{2} + \frac{x_{maxi} - x_{\min i}}{2} cx_i^{(k)} \tag{23}$$

4. Evaluating the new solution $X_{(k+1)} = [x_1^{k+1}, \cdots, x_i^{k+1}, \cdots, x_n^{k+1}]$ by calculating the objective function value J_{k+1}.
5. If $J_{k+1} < F_g$, then replace the chosen chromosome with $X_{(k+1)}$ and J_{k+1}. (F_g means the fitness of selected chromosome) and decrease the searching radius with Eq. (18).
6. If the terminal criterion is reached, go out of the local search, otherwise go back to step 2.

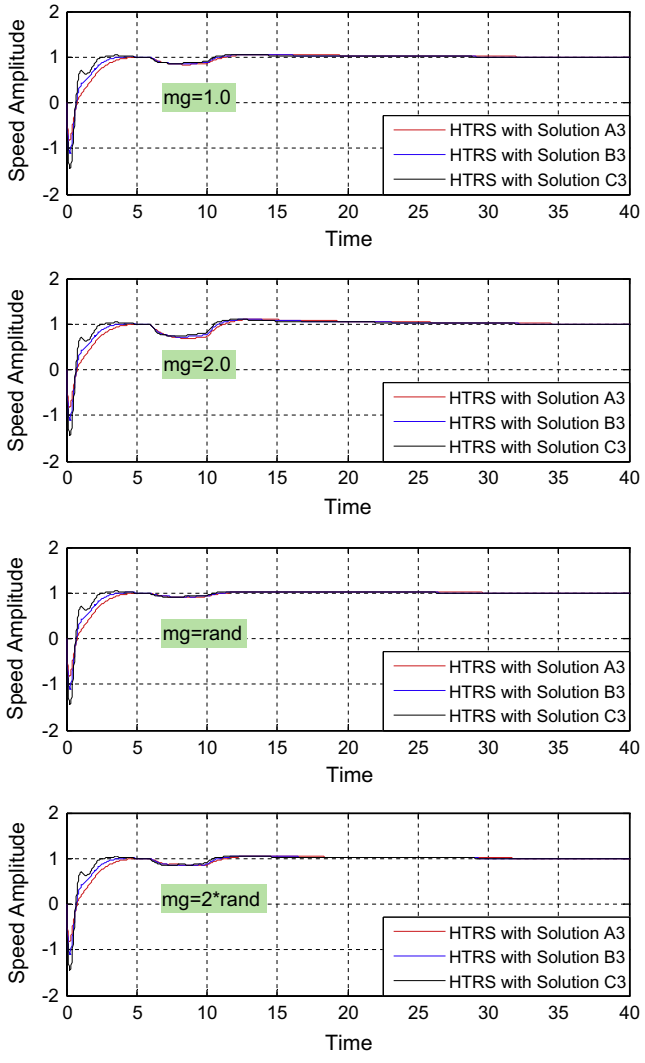

Fig. 9. Robustness of FOPID controller for sudden-load disturbance (unload condition).

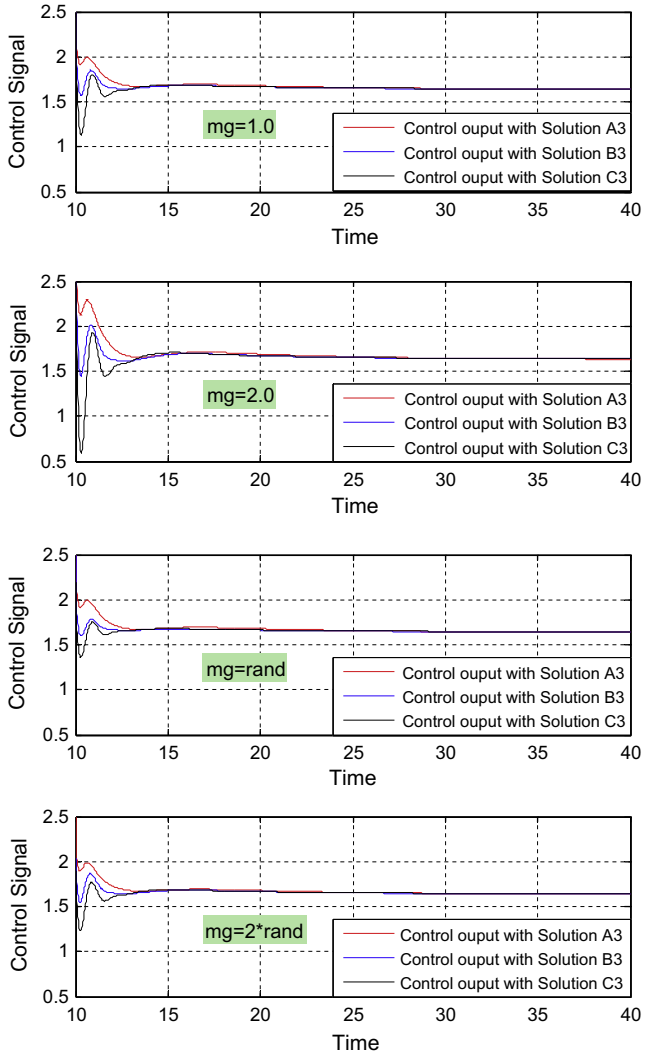

Fig. 10. The control output of FOPID controller for sudden-load disturbance (unload condition).

5. Numerical simulation and results

In this section, the HTRS is simulated in MATLAB, the proposed chaotic NSGAII is applied to optimize the parameters of FOPID controller and the model of the HTRS is illustrated in Fig. 2. Simulation experiments respect to the HTRS is conducted with unload and load running conditions, respectively. Under unload running condition, step disturbance of given speed is adopted to excite this system, and under load running condition, step disturbance of load is used to excite this system. The parameters respect to the hydraulic turbine system and the generator system under different running conditions are given and shown in Table 1. The other parameters in HTRS system are measured as follows: $Ty = 0.1$, $Tw = 1.5$.

In simulations, the population size of chaotic NSGAII is considered to be 60 and the number of generations is chosen as 100 (i.e.$N = 60$, $Maxgen = 100$). The intermediate crossover probability is taken 0.8, and the mutation ratio is taken 0.2 (i.e.$\eta_c = 0.8$, $\eta_m = 0.2$). The elitism count, which represents the number of fitness individuals which are directly copied over to the next generation, is assigned to 60. The maximum iteration for the chaotic map augmented NSGAII is taken as 200. For the optimization problem, the searching ranges of the controller parameters $\{Kp, Ki, Kd\}$ are chosen as [0, 15] and those of the fractional orders $\{\lambda, \mu\}$ are chosen as [0, 2].

5.1. Study of the trade-off between different objectives under various running conditions

5.1.1. Comparison of FOPID and PID controller under unload running condition

In this part of experiments, FOPID controller and the traditional PID controller are employed to act on the HTRS system for a better dynamic performance with a step perturbation of given speed under unload running condition. The simulation speed signal input of HTRS is illustrated in Fig. 2 and experiment is run for a finite time horizon of 20s. The obtained Pareto fronts for HTRS system assisted with PID and FOPID controller considering two contradictory objective functions J_1 and J_2 are shown in Fig. 3. Some representative solutions on the Pareto front are reported in Table 2 for both the PID and the FOPID controllers. The two extreme solutions and the median solution on the Pareto front are chosen as representative cases.

As can be observed from Fig. 3, the Pareto front of the FOPID controller totally lies inside the concave portion of the Pareto front of PID controller, which means that the FOPID controller outperforms the PID controller for all the cases under unload running condition. Figs. 4 and 5 show the set desired-point tracking and control output for the representative solutions as reported in Table 2. The result from the Pareto front shown in Fig. 3 is also verified by these figures. It is seen that the FOPID controller have a better dynamic performance than the optimal PID controller for HTRS system in terms of overall obtained indicators, especially in steady-state error aspects.

5.1.2. Comparison of FOPID and PID controller under load running condition

Fig. 6 shows the Pareto fronts for the PID and the FOPID controller with a step perturbation of given load under load running condition. Since the load disturbance rejection of controllers does not settle within 20s, thus for this case the simulations are run for a finite time horizon of 50s. Here it can be seen that there is a much lower multi-objective function values obtained by the FOPID controller while compared with PID controller. Table 3 shows some representative solutions on the Pareto fronts for both PID and FOPID cases. These representative solutions are chosen as the ones on the extreme ends and the median solution as before.

Figs. 7 and 8 show the load disturbance rejection and the control output under load running condition for the representative solutions as reported in Table 3. As the steady-state output of this running condition is equal to zero, peak time and peak value is used to instead of rise time and overshoot. It is obvious that this system with FOPID controller quickly recovers from a unit load disturbance, but the PID controller takes a long time to recover under a load disturbance.

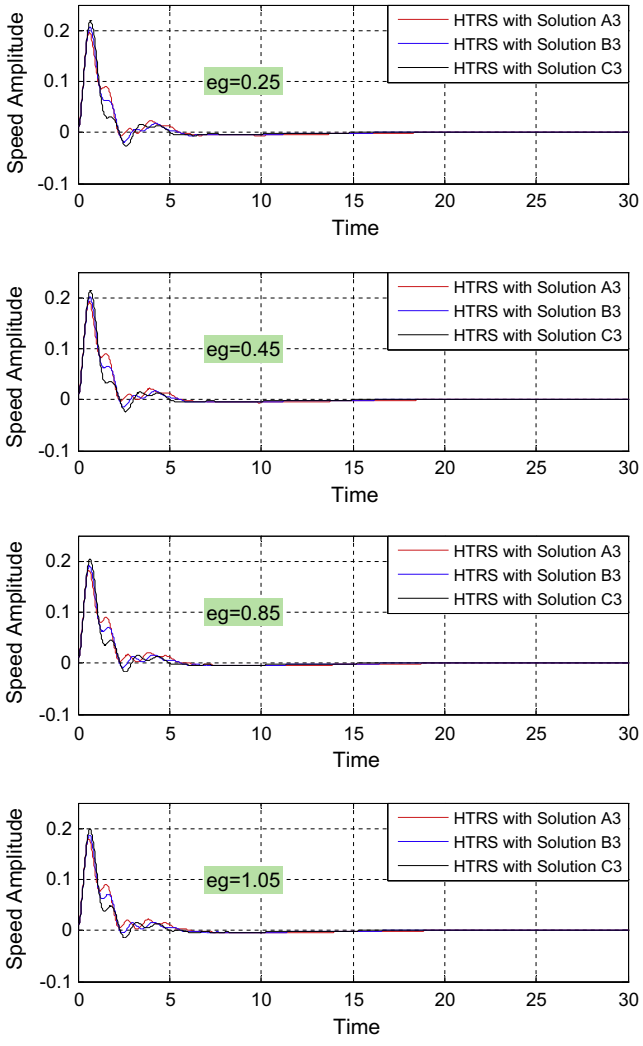

Fig. 11. Robustness of FOPID controller for variation in load-self regulation factor.

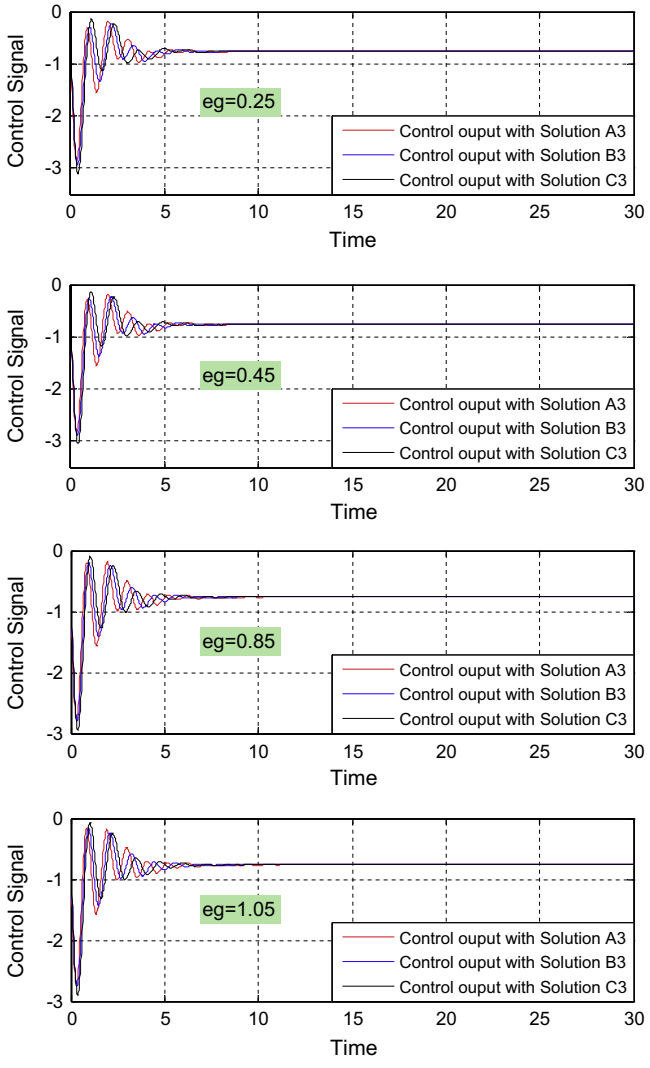

Fig. 12. The control output of FOPID controller for variation in load-self regulation factor.

From the above discussion, it can be seen that no matter whether HTRS system is working under unload or load condition, there is a more desirable dynamic output obtained by the action of the FOPID controller compared with classical PID controller. Hence when these two contradictory objectives are considered in the design framework, the FOPID controller should be preferred.

5.2. Robustness analysis of the obtained solutions

In the previous experiments, there has discussed the FOPID controllers designed for the nominal operating conditions. In this part experiments, some other simulations which test the effect of tuned controller working for sudden-changed operating conditions, i.e. the robustness ability to change in system parameters, are conducted. To illustrate the effect of the variation in system parameter on the obtained solution, there should be discussed in two common encounter situations, namely load disturbance during the unload running condition and load change during the load running condition.

Figs. 9 and 10 show the effect that various load perturbations, which happened in the time range [6s 10s], disrupting this system (under unload condition), for the representative solutions on the Pareto front. With the action of the FOPID controller, it is obvious that the extremes and the median solutions are all own sufficient robustness to against the change in load, which verifies that FOPID controller has a strong robustness against load disturbance during the unload condition.

Figs. 11–14 show the effect of increase and decrease in the gain of the system (under the load condition). Due to the load changes which occur frequently in the system, the load self-regulation factor and water starting time constant are often varied, thus in the present study, the robustness analysis simulation is carried on based on the variation of *eg* and *Tw* during the load condition.

It is seen that no matter how decrease or increase of load self-regulation factor *eg* and *Tw* in Figs. 11–14, the FOPID controller is capable of tolerating these changes for the representative solutions A3, B3, C3. As the robustness of other solutions on the Pareto front which lie in between the representative cases of these solutions have similar characteristics, it is clear that all solutions show sufficient robustness to parameter variation of the system during the load condition.

5.3. Comparison of design trade-offs with different chaotic map augmented NSGAII algorithm

As reported in [44], different chaotic maps augmented over the original evolutionary algorithm contribute different performance improvement percentage for the given problem. It is difficult to

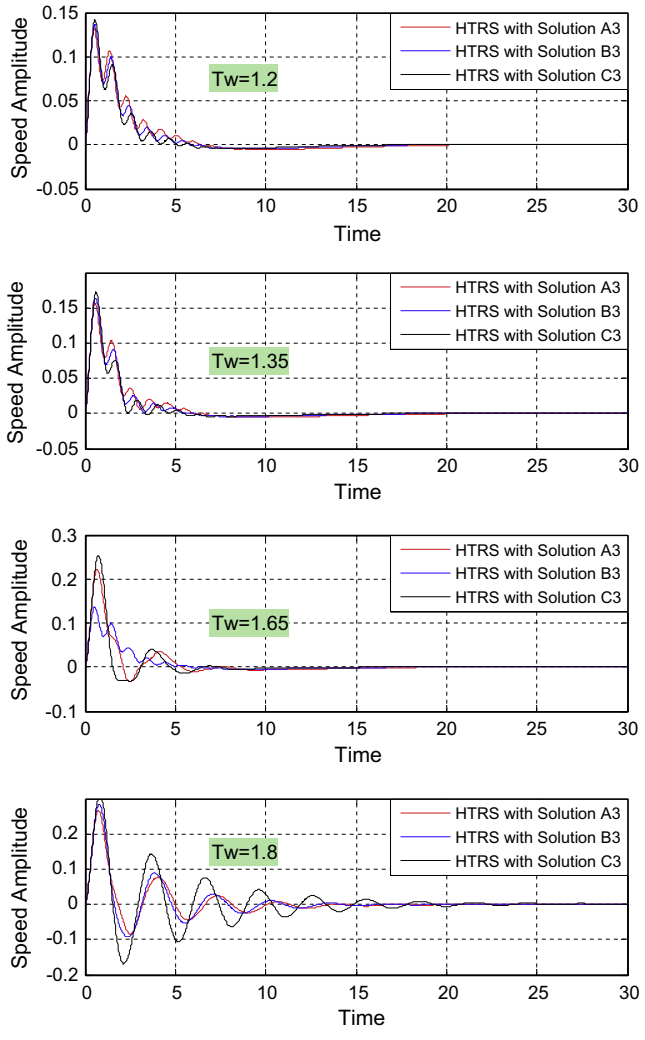

Fig. 13. Robustness of FOPID controller for variation in water starting time constant.

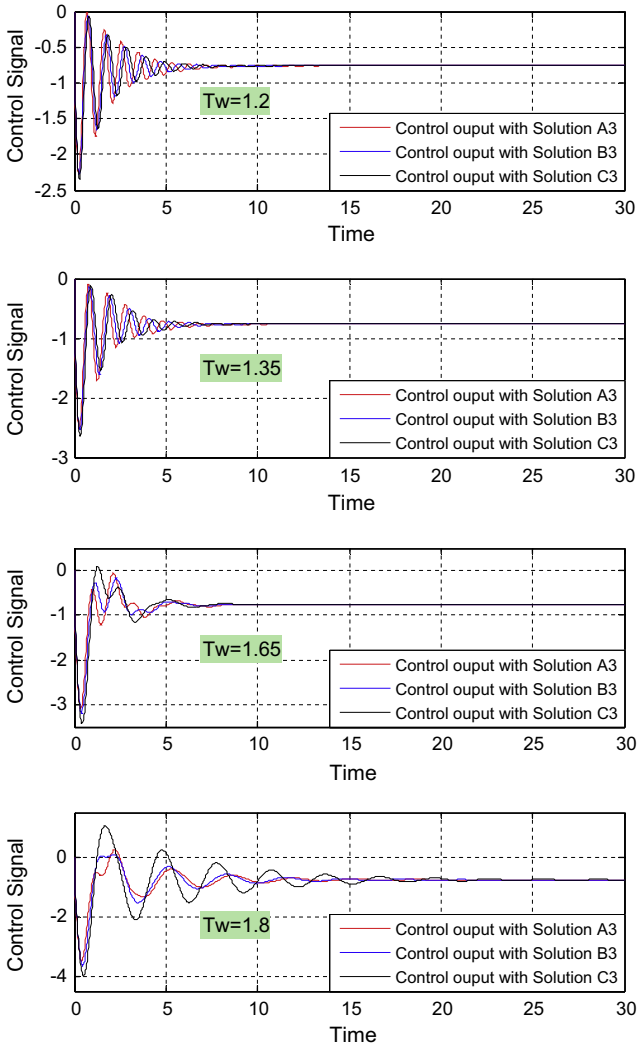

Fig. 14. The control output of FOPID controller for variation in water starting time constant.

identify which chaotic map can give the best assistant to improve the performance of the algorithm in solving the given problem unless with an exhaustive simulation study of all existence mapping, but that would be an ideal cloud-castle and digression from the main focus of the present paper. However, for the sake of completeness, in this paper, another well-known mapping rule, i.e. the Logistic map is also augmented into the classical NSGAII. The one dimensional Logistic map is given as follows:

$$cx^{(k+1)} = a * cx^{k} * (1 - cx^{k}) \quad for \quad a > 0, cx^{(k)} \in (0 \quad 1) \tag{24}$$

where u is a control variable in which $u = 4$, the mapping is in full state [2], the initial condition of Logistic map in Eq. (19) has been chosen a uniform random number between (01) but excluding some special values such as 0.25, 0.5, 0.75 while $u = 4$.

In order to reflect the effective of different chaotic map augmented over the original algorithm, the classical NSGAII, NSGAII assisted with the ICMIC map and NSGAII assisted with the Logistic map are used for multi-objective FOPID controller optimization in HTRS with unload and load running condition, respectively. For the purpose that getting a fair comparison, the maximum iteration for the chaotic map augmented NSGAII is taken 200 for three versions.

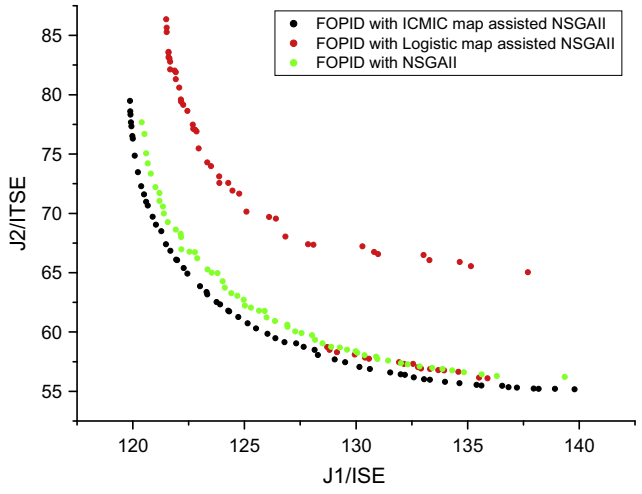

Fig. 15. Comparison of Pareto fronts for FOPID controller under unload running condition.

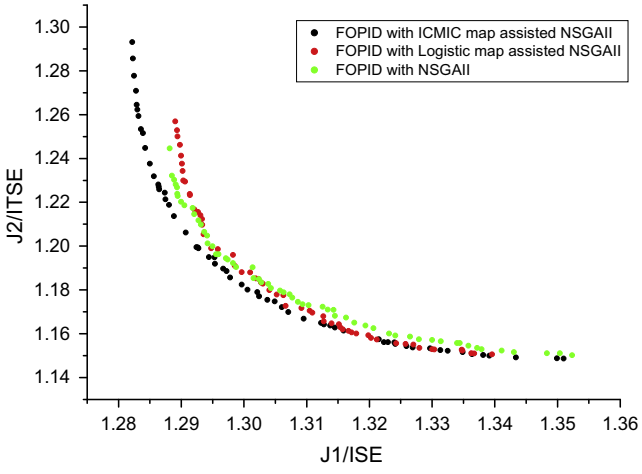

Fig. 16. Comparison of Pareto fronts for FOPID controller under load running condition.

Figs. 15 and 16 show the results of the trade-off design for FOPID controller under unload and load running conditions.

From Figs. 15 and 16, it is clear that the Pareto fronts obtained by two chaotic versions are better than the original NSGAII, which proves the effective of chaotic system incorporated into NSGAII by means of utilizing the chaotic search as a local search procedure. The best result of three versions for FOPID controller has been found is the ICMIC map assisted NSGAII, which improves the results in terms of spread and diversity of Pareto front as well as maximum achievable value of the conflicting objectives in solving the tuning problem of FOPID controller in HTRS.

6. Conclusions

In this paper, a fractional order PID controller (FOPID or $PI^{\lambda}D^{\mu}$) is designed for the HTRS system with the consideration of conflicting performance objectives. An improved multi-objective evolutionary non-dominated sorting genetic algorithm II (NSGAII), augmented with the iterative chaotic map with infinite collapses (ICMIC), is used to balance the trade-offs of different design objectives. A comparative analysis is made between the PID and FOPID controller and it is shown that the latter outperforms the former and gives better designs. The inclusion of ICMIC map as a convergence and diversity retainer assisted for NSGAII outperforms than the original version and its chaotic Logistic map assisted version for simultaneous maximization of objectives with the FOPID controller. It is also shown that the stability and robustness of the design is also guaranteed unlike the other optimization based controller design methods, which means the proposed controller design technique may serve as an efficient alternative for the design of next-generation controllers.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 51379080) and Fundamental Research Funds for the Central Universities (No. 2014TS152).

References

[1] Fang H, Chen L, Shen Z. Application of an improved PSO algorithm to optimal tuning of PID gains for water turbine governor. Energy Convers Manage 2011;52:1763–70.
[2] Chen Z, Yuan X, Tian H, Ji B. Improved gravitational search algorithm for parameter identification of water turbine regulation system. Energy Convers Manage 2014;78:306–15.
[3] Fang JA, Zheng D, Ren Z. Computation of stabilizing PI and PID controllers by using Kronecker summation method. Energy Convers Manage 2009;50:1821–7.
[4] Jiang C, Ma Y, Wang C. PID controller parameters optimization of hydro-turbine governing systems using deterministic-chaotic-mutation evolutionary programming (DCMEP). Energy Convers Manage 2006;47:1222–30.
[5] Eker İ. Governors for hydro-turbine speed control in power generation: a SIMO robust design approach. Energy Convers Manage 2004;45:2207–21.
[6] Kishor N, Singh S, Raghuvanshi A. Dynamic simulations of hydro turbine and its state estimation based LQ control. Energy Convers Manage 2006;47:3119–37.
[7] Podlubny I. Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers. IEEE Trans Autom Control 1999;44:208–14.
[8] Domingues J, Valério D, da Costa JS. Rule-based fractional control of an irrigation canal. J Comput Nonlinear Dyn 2011;6:024503.
[9] Petráš I. Fractional-order feedback control of a DC motor. J Electr Eng 2009;60:117–28.
[10] Muresan CI, Folea S, Mois G, Dulf EH. Development and implementation of an FPGA based fractional order controller for a DC motor. Mechatronics 2013;23:798–804.
[11] Villagra J, Vinagre B, Tejado I. Data-driven fractional PID control: application to DC motors in flexible joints. In: IFAC conference on advances in PID control. Brescia, Italy; 2012.
[12] Pan I, Das S. Chaotic multi-objective optimization based design of fractional order $PI^{\lambda}D^{\mu}$ controller in AVR system. Int J Electr Power Energy Syst 2012;43:393–407.

[13] Pan I, Das S. Frequency domain design of fractional order PID controller for AVR system using chaotic multi-objective optimization. Int J Electr Power Energy Syst 2013;51:106–18.
[14] Zamani M, Karimi-Ghartemani M, Sadati N, Parniani M. Design of a fractional order PID controller for an AVR using particle swarm optimization. Control Eng Practice 2009;17:1380–7.
[15] Aboelela MA, Ahmed MF, Dorrah HT. Design of aerospace control systems using fractional PID controller. J Adv Res 2012;3:225–32.
[16] Jia L, Xiuyun M, Zaozhen L. Freestyle fractional order controller design with PSO for weapon system. Energy Procedia 2011;13:7577–82.
[17] Melício R, Mendes VMF, Catalão JPS. Fractional-order control and simulation of wind energy systems with PMSG/full-power converter topology. Energy Convers Manage 2010;51:1250–8.
[18] Das S, Pan I, Das S. Fractional order fuzzy control of nuclear reactor power with thermal-hydraulic effects in the presence of random network induced delay and sensor noise having long range dependence. Energy Convers Manage 2013;68:200–18.
[19] Bouafoura MK, Braiek NB. $PI^{\lambda}D^{\mu}$ controller design for integer and fractional plants using piecewise orthogonal functions. Commun Nonlinear Sci Numer Simul 2010;15:1267–78.
[20] Padula F, Visioli A. Tuning rules for optimal PID and fractional-order PID controllers. J Process Control 2011;21:69–81.
[21] Monje CA, Vinagre BM, Feliu V, Chen Y. Tuning and auto-tuning of fractional order controllers for industry applications. Control Eng Practice 2008;16:798–812.
[22] Lee CH, Chang FK. Fractional-order PID controller optimization via improved electromagnetism-like algorithm. Expert Syst Appl 2010;37:8871–8.
[23] Biswas A, Das S, Abraham A, Dasgupta S. Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution. Eng Appl Artif Intell 2009;22:343–50.
[24] Tehrani KA, Amirahmadi A, Rafiei S, Griva G, Barrandon L, Hamzaoui M, et al. Design of fractional order PID controller for boost converter based on multi-objective optimization. In: IEEE 14th international power electronics and motion control conference (EPE/PEMC). Ohrid, Macedonia; 2010, pp. T3-179–T3-85.
[25] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6:182–97.
[26] He D, He C, Jiang LG, Zhu HW, Hu GR. A chaotic map with infinite collapses. In: IEEE TENCON proceedings. Kuala Lumpur, Malaysia; 2000 pp. 95–9.
[27] Oustaloup A. La commande CRONE: commande robuste d'ordre non entier. Paris: Hermes; 1991.
[28] Herreros A, Baeyens E, Perán JR. Design of PID-type controllers using multiobjective genetic algorithms. ISA Trans 2002;41:457–72.
[29] Shabanpour-Haghighi A, Seifi AR, Niknam T. A modified teaching–learning based optimization for multi-objective optimal power flow problem. Energy Convers Manage 2014;77:597–607.
[30] Atashkari K, Nariman-Zadeh N, Gölcü M, Khalkhali A, Jamali A. Modelling and multi-objective optimization of a variable valve-timing spark-ignition engine using polynomial neural networks and evolutionary algorithms. Energy Convers Manage 2007;48:1029–41.
[31] Chen D, Ding C, Ma Z, Yuan P, Ba D. Nonlinear dynamical analysis of hydro-turbine governing system with a surge tank. Appl Math Model 2013;37:7611–23.
[32] De Jaeger E, Janssens N, Malfliet B, Van De Meulebroeke F. Hydro turbine model for system dynamic studies. IEEE Trans Power Syst 1994;9:1709–15.
[33] Tian H, Yuan X, Ji B, Chen Z. Multi-objective optimization of short-term hydrothermal scheduling using non-dominated sorting gravitational search algorithm with chaotic mutation. Energy Convers Manage 2014;81:504–19.
[34] Ahmadi MH, Dehghani S, Mohammadi AH, Feidt M, Barranco-Jimenez MA. Optimal design of a solar driven heat engine based on thermal and thermo-economic criteria. Energy Convers Manage 2013;75:635–42.
[35] Sağlam G, Tutum CC, Kurtulan S. A new PI tuning method for an industrial process: a case study from a micro-cogeneration system. Energy Convers Manage 2013;67:226–39.
[36] Guo D, Wang J, Huang J, Han R, Song M. Chaotic-NSGA-II: an effective algorithm to solve multi-objective optimization problems. In: IEEE international conference on intelligent computing and integrated systems (ICISS). Guilin, China; 2010. pp. 20–3.
[37] Deb K, Agrawal RB. Simulated binary crossover for continuous search space. Complex Syst 1994;9:1–34.
[38] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. Comp Sci Inform 1996;26:30–45.
[39] Yuan X, Cao B, Yang B, Yuan Y. Hydrothermal scheduling using chaotic hybrid differential evolution. Energy Convers Manage 2008;49:3627–33.
[40] Yuan X, Zhang Y, Yuan Y. Improved self-adaptive chaotic genetic algorithm for hydrogeneration scheduling. J Water Res Plann Manage-ASCE 2008;134:319–25.
[41] Yuan X, Wang Y, Xie J, Qi X, Nie H, Su A. Optimal self-scheduling of hydro producer in the electricity market. Energy Convers Manage 2010;51:2523–30.
[42] Yuan X, Su A, Nie H, Yuan Y, Wang L. Unit commitment problem using enhanced particle swarm optimization algorithm. Soft Comput 2011;15:139–48.
[43] Yuan X, Su A, Nie H, Yuan Y, Wang L. Application of enhanced discrete differential evolution approach to unit commitment problem. Energy Convers Manage 2009;50(9):2449–56.
[44] Bucolo M, Caponetto R, Fortuna L, Frasca M, Rizzo A. Does chaos work better than noise? IEEE Circuits Syst Mag 2002;2:4–19.