# Research on SVG DC-Side Voltage Control Based-on PSO Algorithm

**Yingwei Xiao**
School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, 230009, china
E-mail: xiaoyingwei8@126.com, xiaobenxian@126.com

*Abstract*—The operation flow of particle swarm optimization (PSO) is presented, at the same time the PSO algorithm and GA algorithm are used to find the optimal value of the standard function, simulation results show that the PSO algorithm has better global search performance and faster search efficiency. The inertia weight decreasing strategy of PSO algorithm is studied, the simulation results show that the concave function decreasing strategy can accelerate the convergence rate of the algorithm. The stability control of the DC-side voltage is very important for the static var generator (SVG) compensation, but the disadvantages of the traditional PI control are fixed parameters and poor adaptability of dynamic response, PSO algorithm is introduced to the optimization of PI parameters, so online PSO-PI control and off-line PSO-PI control are obtained, the SVG voltage loop transfer function is used as the controlled object. The simulation results show that the PSO-PI control can satisfy the time varying system of the controlled object with strong adaptability.

*Index Terms*—SVG, DC voltage stability control, PSO, off-line PSO-PI, online PSO-PI.

## I. INTRODUCTION

In recent years, the wide use of power electronic devices makes the semiconductor switching equipments increasingly diverse, these devices consume reactive power and bring additional burden for the power grid and adverse effects on the power quality. Therefore, it is important and urgent to control the reactive power in power grid in order to ensure the power transmission quality and ensure the power system to be effective, stable and normal operation.

SVG is a reactive power compensation device which composed of inverter, by controlling the on-off of the inverter switching device, continuous injection or absorption of reactive power to the power grid, not only can compensate reactive power, but also can compensate the harmonic[1-3].

At present, the efficiency of SVG in controlling reactive power, improving the power transmission quality and guarantying the power system stability and other aspects has been recognized. Because of its rapid development, it becomes one of the most important reactive power compensation devices, and is the focus of the research on the direction of reactive power compensation and harmonic control.

The main research contents of this paper mainly includes: (1) The operation flow of PSO algorithm is given, and the PSO algorithm and GA algorithm are applied to the optimization comparison of standard function extreme value[4-6]. (2) The inertia weight decreasing strategy of PSO is studied. (3) considering that the stability control of the DC side voltage is very important to guarantee the effect of SVG compensation, and considering the disadvantages of the traditional PI control with fixed parameters and the poor adaptability, the paper introduces the PSO algorithm with global search performance into the PI parameters optimization[7-9], online PSO-PI control and off-line PSO-PI control are obtained. (4) The paper takes the voltage loop of SVG as the controlled object, and the simulation verification is done for PSO-PI control.

## II. PRINCIPLE OF SVG

According to the different choices of DC energy storage device, the SVG circuit can be divided into the voltage type bridge circuit and the current type bridge circuit, in fact, so far the used SVG mostly adopts the voltage type circuit, because the voltage type circuit has a high running efficiency. Therefore, SVG often refers to the voltage type circuit as the inverter circuit of dynamic compensation devices. The basic circuit structure is shown in Fig. 1.
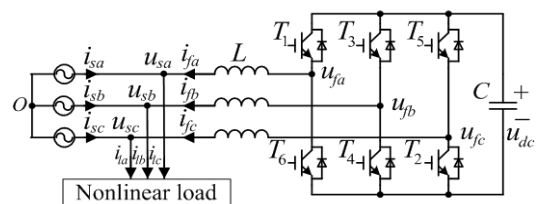


Fig.1. Basic circuit structure of SVG

The principle of SVG is that the AC side of inverter circuit can output the voltage $u_{fa}, u_{fb}, u_{fc}$ of preset amplitude and frequency through the on-off control of $T_1 \sim T_6$ six semiconductor switching devices shown in Figure 1. Under the action of the connecting reactance, compensating current $i_{fa}, i_{fb}, i_{fc}$ are converted, and offset reactive current in power grid, achieve the purpose of reactive power compensation.

In practice, the IGBT full control switch is chosen as the semiconductor switching devices. Therefore, the SVG can be equivalent to the controllable amplitude and phase voltage source connected with the power grid through the connection reactance.

## III. PSO ALGORITHM RESEARCH

In 1995, Eberhart and Kennedy used the bird population behavior model of biologist Hepper to propose a particle swarm optimization (PSO) algorithm. Each individual in the group follows a very simple three rules of conduct:

(1) Away from the nearest individual, to avoid a collision.
(2) Flight to the target.
(3) Flight to the center of the group.

Different from the model proposed by Hepper, Kennedy thought birds initially didn't know the target position, each individual follows a certain rule to evaluate their current position, and "local optimum" is used as the best position that it had flown, "globally optimal" is used as the best location that the entire population had flown, birds use the two optimal variables to guide their flight, share information with each other until the final flight target position.

Assuming there is a group of $M$ particles, each particle is in a $D$ dimensional search space, the speed of $i$ particle can be expressed as vector $v_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$, $i = 1, 2, \cdots M$, position is expressed as vector $x_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$. Then according to the pre-set fitness function, the merits of particles are evaluated for their current position, the position and speed of the particles are updated according to the fitness value, updated basis is two extreme values[10,11]: individual extremum and global extremum. The individual extremum is the best position of $i$ particle to be found at present, denoted as $pBest = (p_{i1}, p_{i2}, \cdots, p_{iD})$. The global extremum is the best position to be found for all the particles in the group, denoted as $gBest = (p_{g1}, p_{g2}, \cdots, p_{gD})$, ( $gBest$ is the optimal value of $pBest$ ). For the $k$ iteration, the update rule of particle speed and position in PSO is shown in the formula (1):

$$\begin{cases} v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 \left( p_{id} - x_{id}^k \right) + c_2 r_2 \left( p_{gd} - x_{id}^k \right) \\ x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \end{cases} \quad (1)$$

In the formula (1), $d = 1, 2, \cdots, D$, $d$ represents the current search space dimension, $i = 1, 2, \cdots, M$, $i$ represents the particle, $k$ represents the evolution algebra. $x_{id}^k$: the $d$ dimensional component of the position vector of particle $i$ by $k$ iterations. $v_{id}^k$: the $d$ dimensional component of the speed vector of particle $i$ by $k$

iterations. $p_{gd}$: the $d$ dimensional component of global extremum $gBest$. $p_{id}$: the $d$ dimensional component of particle $i$ individual extremum $pBest$. $c_1$, $c_2$: acceleration factor, $c_1$ is used to adjust the particles speed flying to the individual extremum. $c_2$ is used to adjust the particles speed flying to the global extremum. $\omega$: inertia weight, adjust its size and can change the strength of the particle swarm search ability[12], $r_1$, $r_2$ are the random number between [0,1].

### A. PSO operation flow

The operation flow of PSO algorithm can be summarized as:

*1. Encoding for the solution space of problem*

Common encoding way includes binary encoding and real encoding. Real encoding is more commonly used way of PSO algorithm encoding, and it directly searches the solution space.

*2. Determining the objective function of optimization problem*

The objective function is to reflect the characteristics of the problem to be optimized, that is, the fitness value of the particles. The fitness value of particles in the current position determines the update degree of the particle's position and speed.

*3. Initialization of the parameters in the algorithm*

(1) Determining the particle position and speed range, and initializing population size,
(2) Initializing the position information of each particle,
(3) Initializing the speed information of each particle,
(4) Initializing inertia factor, learning factor, iteration number and other parameters.

*4. Designing particle flying model*

In the PSO algorithm, the flying speed of particles directly affects the ability of the particles to search for the optimal solution, too fast flying speed makes the particles fly over the global extreme point, and too slow flying speed may make particles falling into the local extreme point. The design of flying model directly reflects the ability of the particles to use their own memory and share the social information.

*5. Determining the termination condition of algorithm*

The termination condition of PSO algorithm is usually that the maximum number of iterations is satisfied or the fitness of the particles reaches a certain extreme value, also multiple termination conditions can be set according to different problems.

*6. Code implementation*

According to the above steps, the PSO algorithm is programmed, and need to change one or a few parameters of the algorithm in the process of debugging in order to

search for the optimal solution.

Integrated the above steps, the PSO algorithm flow can be summarized as:

**Step1**: initialization of the algorithm parameters, initializing the population size, the number of iterations, the inertia weight, the learning factor, the particle position information and speed information et al.

**Step2**: according to the predetermined fitness function, the current fitness values of all particles are calculated.

**Step3**: compare the particle current position and individual extremum $pBest$, if the current position is better, the current position is used as the particle individual extremum $pBest$. Compare the current particle individual extremum $pBest$ and group global extremum $gBest$, if the current position of the particle is better, the individual extreme $pBest$ is used as the global extremum.

**Step4**: judging whether the termination of the algorithm is satisfied, if satisfied, the algorithm ends, if not satisfied, enter Step5.

**Step5**: each particle's position and speed are updated according to formula (1), continue to enter Step2.

### B. Analysis of standard function simulation

#### 1. Simulation model

In order to show the superiority of the PSO algorithm, the PSO algorithm and GA algorithm are simulated in the same condition, and the solving efficiency of the two algorithms is compared.

Seeking the maximum value of Rosenbrock function in a certain range:

$$\begin{cases} f\left(x_1, x_2\right) = 100\left(x_1^2 - x_2\right)^2 + \left(1 - x_1\right)^2 \\ -2.048 \le x_i \le 2.048 \left(i = 1, 2\right) \end{cases} \quad (2)$$

In the range of $-2.048 \le x_i \le 2.048\left(i = 1, 2\right)$, the function has two local extreme points, that is $f\left(2.048, -2.048\right) = 3897.7342$ and $f\left(-2.048, -2.048\right) = 3905.9262$ respectively, and the latter is the global maximum.

#### 2. Simulation parameter setting

The sampling time of simulation $t_s = 0.001s$, the individual fitness is the value of Rosenbrock function, the reciprocal of the individual fitness is the objective function. The population size of GA algorithm $Size = 500$, termination of evolutionary algebra $G = 200$, crossover probability $P_c = 0.90$, mutation probability $P_m = 0.10 - \left[1:1:Size\right] \times 0.01 / Size$, adaptive mutation probability, the smaller the adaptation degree, the greater the mutation probability, decimal encoding. PSO algorithm also uses the real encoding, group size, termination of the evolution algebra are same as GA algorithm, the particle speed range is [-0.1, 0.1].

### 3. Simulation results

Figure 2 shows the objective function values changes with the number of iterations for the GA algorithm and the PSO algorithm with linear decreasing strategy of inertia weight, it can be seen that the PSO algorithm has a faster convergence speed.
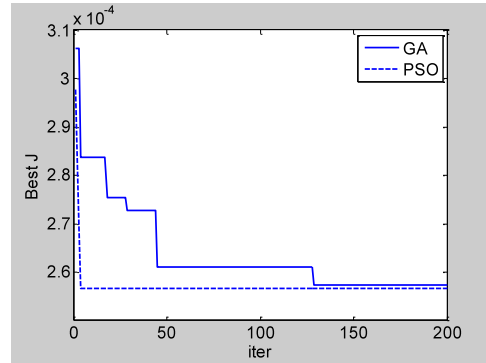


Fig.2. Optimization comparison between GA and PSO

According to the above simulation parameters, GA algorithm and PSO algorithm randomly run five times, the five results of GA are 3882.2210, 3860.4745, 3891.7978, 3878.3616, 3882.9877 respectively. While the five results of PSO are 3905.9262. It can be seen that the PSO algorithm has higher search accuracy and faster convergence speed, and has good stability.

In order to make the algorithm obtain the strong global search ability at the early, and the fast local convergence speed at the late, according to the formula (3) the inertia weight is reduced linearly. Among them, $\omega Min, \omega Max$ are the minimum and the maximum inertia weight, $iter$ is the number of the current iteration, $iterMax$ is the total number of iterations.

$$\omega = \omega Max - iter \times \frac{\omega Max - \omega Min}{iterMax} \quad (3)$$

In order to find a better balance between the global search ability and local search ability, two nonlinear inertia weight decreasing strategies are introduced into the paper, namely the concave function decreasing strategy and the convex function decreasing strategy, respectively, as shown in the formula (4) and formula (5):

$$\omega = \left(\omega Max - \omega Min\right)\left(iter / iterMax\right)^2 + \\ \left(\omega Min - \omega Max\right)\left(2iter / iterMax\right) + \omega Max \quad (4)$$

$$\omega = -\left(\omega Max - \omega Min\right)\left(iter / iterMax\right)^2 + \omega Max \quad (5)$$

In order to fully compare the influence on the global search ability and local search ability of PSO algorithm with three kinds of inertia weight decreasing strategies. Selecting, the group of particle swarm is 30, the number of iterations is 1000, the minimum value of the ten dimensional Rosenbrock function of formula (6) is searched, and the objective function is the function value.

$$\begin{cases} f\left(x_1, x_2, \cdots x_{11}\right) = \sum_{i=1}^{10}\left[100\left(x_i^2 - x_{i+1}\right)^2 + \left(1 - x_i\right)^2\right] & (6) \\ -100 \le x_j \le 100\left(j = 1, 2, \cdots, 11\right) \end{cases}$$

By using the convex function decreasing inertia weight and linear decreasing inertia weight and concave function decreasing inertia weight PSO algorithm, the minimum value of ten dimensional Rosenbrock function is searched respectively, and 50 times are done respectively. Three kinds of inertia weight decreasing strategies are used to obtain the minimum value of the function, the average value of 50 times is: 0.155331854735086, 0.123303231330160, 0.099328198916885. The fitness curve of PSO algorithm is shown in Figure 3 under three strategies. It can be seen that the PSO algorithm by using the concave function decreasing inertia weight has been accelerated, the inertia weight is reduced fast in the initial stage of the algorithm, and the convergence speed of the algorithm is accelerated, the optimization accuracy is the highest.
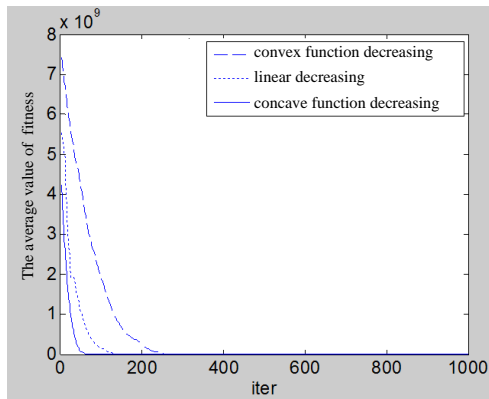


Fig.3. Comparison with three kinds of inertia weight decreasing optimization

## IV. PID Parameter Tuning based on PSO

The PSO algorithm is used for PID parameter tuning, according to the PSO algorithm tuning principle and the different ways of implementation, the tuning methods can be divided into off-line PSO-PID and on-line PSO-PID[13-18]. Figure 4 is the PSO-PID control system structure diagram.
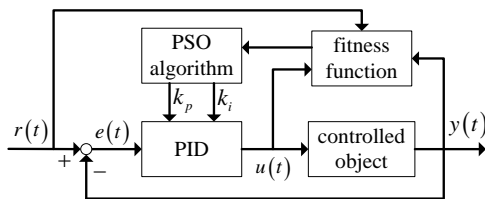


Fig.4. Control system structure of PSO-PID

### A. Off-line PSO-PID algorithm

In off-line PSO-PID, PSO algorithm firstly needs off-line learning. The position and speed of the particles are updated within the scope of space according to the size of the fitness function, and constantly search for the optimal PID parameters, after the completion of the search, the global best position is substituted into the control system. In this paper, the time integral performance index of the error absolute value is selected as the fitness function, in order to prevent the overshoot and ensure the fast system response, the fitness function is shown in the formula (7).

$$J = \int_0^{\infty}\left(\omega_1\left|e(t)\right| + \omega_2 u^2(t)\right)dt + \omega_3 t_u \qquad (7)$$

In the formula (7), $e(t)$ represents the error, $u(t)$ represents the output of PID controller, $t_u$ represents the rise time, $\omega_1$, $\omega_2$, $\omega_3$ represent the weight of the performance index of each part.

The penalty function is introduced into the fitness function to prevent the overshoot. Once the overshoot is generated, the overshoot is used as the most important one of the fitness function. At this time, the fitness function is:

$$if\ e(t) < 0\quad J = \int_0^{\infty}\left(\omega_1\left|e(t)\right| + \omega_2 u^2(t) + \omega_4\left|e(t)\right|\right)dt + \omega_3 t_u$$

(8)

In the formula (8), $\omega_4 \gg \omega_1$, in order to ensure that the adaptation value function is the largest proportion after the overshoot, that is, the current target is to suppress the overshoot.

After determining the fitness function, according to the PSO operation flow in MATLAB programming, the concave function decreasing strategy of the inertia weight is used, and the process of the off-line PSO-PID is as follows:

**Step 1**: initialization of algorithm parameters. The population size, the iteration number, the inertia weight, the learning factor, the initial values of PID parameters, the update speed, the change range of the position & speed, and each weight value in formula (7) and (8) are initialized.

**Step 2**: takes the each current particle PID parameters into the controlled object, gets the performance parameters of corresponding step response, calculates the fitness value of particles according to formula (7) and (8).

**Step 3**: comparison between the current particle position and individual extreme $pBest$, if the current position is better, take the current position as the particle individual extreme $pBest$. Compare the current particle individual extremum $pBest$ with the group global extremum $gBest$, if the current particle position is better, take the individual extreme $pBest$ as the global extremum value.

**Step 4**: judging whether the termination condition of the algorithm is satisfied. If satisfied, the algorithm ends. If not satisfied, enter Step5.

**Step 5**: according to the formula (1), the position and speed of each particle is updated, continue to enter Step2.

In order to verify the effectiveness of the off-line PSO-PID algorithm, the two order transfer function of the formula (9) is used as an example, and the results are compared with the off-line GA-PID tuning value and off-line PSO-PID tuning value.

Controlled object:

$$G(s) = \frac{400}{s^2 + 50s} \tag{9}$$

Setting simulation parameters, the crossover probability and mutation probability of GA algorithm are 0.9, 0.033 respectively. In formula (7) and (8), $\omega_1, \omega_2, \omega_3, \omega_4$ are 0.999, 0.001, 100, 2.0 respectively. The initial inertia weight of PSO algorithm is 0.95, the termination of inertia weight is 0.4, the concave function decreasing strategy is adopted, the acceleration factor $c_1 = c_2 = 2$. The range of proportional gain parameter of GA algorithm and PSO algorithm is [0,20], the integral gain parameter and the differential gain parameter are [0,1], the number of particles is 30, real encoding, 100 generations of iteration.
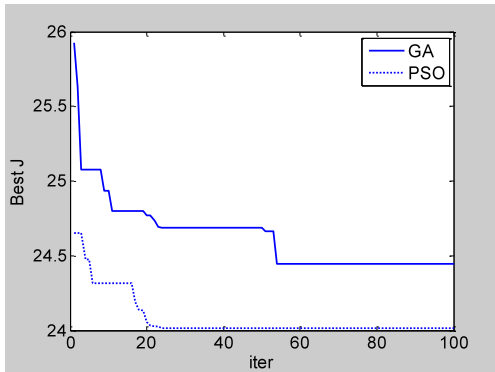


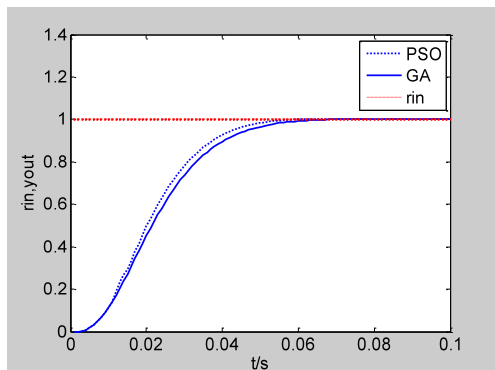Fig.5. Adaptive value iteration curve of GA and PSO



Fig.6. The step response of the system under the condition of off-line GA and off-line PSO

Figure 5 is the adaptive value iteration curve of GA and PSO, Figure 6 is the step response of controlled object by using two algorithms to find the optimal PID parameters. Can be seen that the PSO algorithm

converges quickly fast, the optimal value can be found in 25 generations, but GA algorithm needs 54 generations around to find the optimal value. From the optimization accuracy, the final fitness value of PSO algorithm is 24.0122, the final fitness value of GA algorithm is 24.4478.

According to Figure 5 and Figure 6, GA algorithm and PSO algorithm can obtain better effect in the off-line tuning of PID parameters, the step response curve is basically no overshoot, and has fast rise time. Compared with GA algorithm, PSO algorithm has higher searching accuracy and better convergence speed, and the off-line PSO-PID has better tuning effect.

*B. Online PSO-PID algorithm*

The so-called online PSO-PID tuning method is to tune PID parameters at each sampling period. For example in a sampling time, select enough particles, calculate each individual's fitness, according to the degree of fitness the magnitude and direction of the particle position and speed updating value are determined, after a certain number of iterations, the optimal PID parameters are obtained at this sampling time. In the next sampling time, the optimal PID parameters are searched by using the same method.

Firstly, the particle fitness function is constructed, and the absolute value of the error and the absolute value of error change rate are used as the fitness function of $i$ particle:

$$J(i) = \alpha_p \times |errori(i)| + \beta_p \times |de(i)| \tag{10}$$

In the formula (10), $\alpha_p, \beta_p$ are the weight, $errori(i)$ is the error of $i$ particle, $de(i)$ is the error change rate. The introduction of the penalty function is to prevent the overshoot:

$$if\ errori(i) < 0 \quad J(i) = J(i) + 100|errori(i)| \tag{11}$$

Online PSO-PID algorithm can be summarized as the following six steps:

**Step 1**: Setting sampling time, initializing the size of the population, the number of iterations, the inertia weight, the learning factor and the initial value of the PID parameters and the update speed and the weight of the formula (10).

**Step 2**: in the $k$ sampling time, take the PID parameters of current each particle into the controlled object, calculate the error and error change rate, then calculate the fitness values according to formula (10) and (11).

**Step 3**: compare the particle current position and individual extremum *pBest*, if the current position is better, the current position is used as the particle individual extremum *pBest*. Compare the current particle individual extreme *pBest* and the global extremum *gBest*, if the current particle position is better, the

individual extreme $pBest$ is used as global extremum. The current best position is used as the PID parameters in the $k$ sampling time.

**Step 4**: judging whether to meet the algorithm termination conditions in this sampling time, if satisfied, jump into Step6. If not satisfied, enter Step5.

**Step 5**: follow the formula (1) to update the particle position and speed, and then continue to enter Step2.

**Step 6**: judge whether the final time of algorithm is satisfied or not, if satisfied, the algorithm ends. If not satisfied, $k = k+1$, enter the next sampling time, enter Step2.

Also select formula (9) as the controlled object, the system input is the step signal, sampling time is 1ms, PD parameters are optimized on-line by PSO algorithm and GA algorithm. Proportional gain parameter range is [9.0, 12.0], differential gain parameter range is [0.2, 0.3], GA algorithm crossover probability $P_c = 0.90$, mutation probability $P_m = 0.20 - [1:1:Size] \times 0.01 / Size$, the initial inertia weight of PSO algorithm is 0.95, the termination of the inertia weight is 0.4, concave function decreasing strategy, acceleration factor $c_1 = c_2 = 2$. The population size of two algorithms is 120, the generation of evolutionary algebra is 10, use formula (10) and (11) as the fitness function, and $\alpha_p = 0.95$, $\beta_p = 0.05$.

Figure 7 is the step response curve by using on-line PSO-PD and on-line GA-PD to optimize PD parameters, figure 8 respectively represents the varying curve of two algorithms under the proportional gain $k_p$ and the differential gain $k_d$ versus time. Can be seen from Figure 8, $k_p$ and $k_d$ have similar time varying trend under two kinds of algorithms.
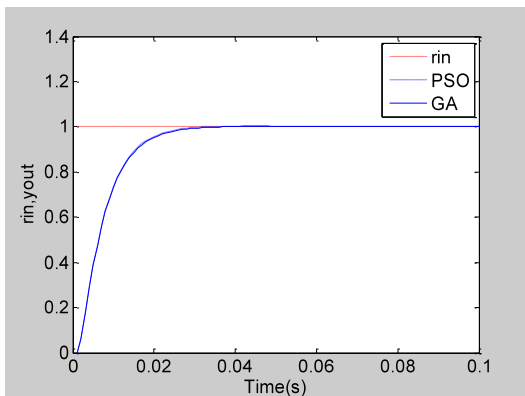


Fig.7. The system step response under the condition of online GA and online PSO

For the two cases of off-line PID tuning and on-line PID tuning, after a comprehensive comparison, the application of GA algorithm and PSO algorithm in PID parameters tuning can be concluded as follows:

(1) Compared with the off-line GA-PID, off-line PSO-PID has faster convergence speed and higher searching accuracy.

(2) The step response of the system is almost the same at on-line GA-PID and on-line PSO-PID tuning. Because the PSO algorithm has no selection, crossover and mutation operators in GA algorithm, it only needs to update the position and speed information according to the adaptive value, so the algorithm is simpler and faster.

(3) Compare figure 6 and figure 7, the step response of on-line tuning is better than that of off-line tuning.
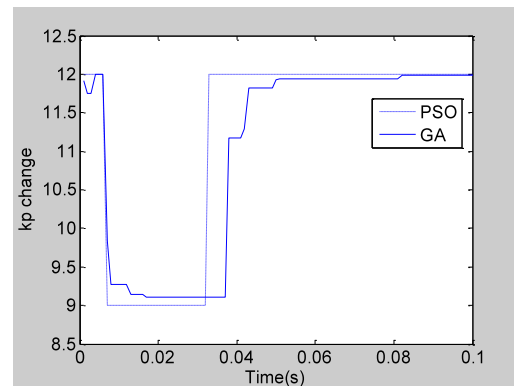


(a) Proportional gain parameter variation
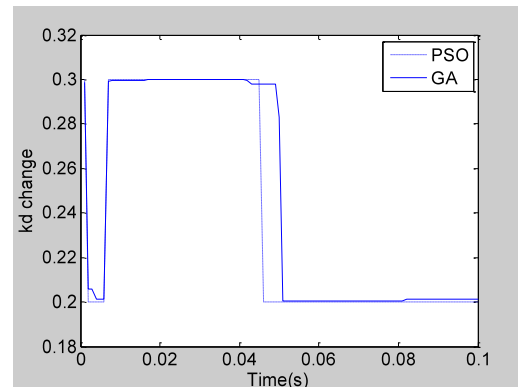


(b) Differential gain parameter variation

Fig.8. Parameters variation waveform of online GA and online PSO

## V. RESEARCH ON SVG DC-SIDE VOLTAGE CONTROL

In theory, SVG by using full control device, which is used to the reactive power compensation of the power grid, is not required to set up the energy storage element in the DC side. But in practice, considering the SVG converter circuit to absorb the fundamental current in addition, but also including harmonic current, which will cause the round return of part energy between the SVG and the power supply. Therefore, the existence of the DC energy storage elements can guarantee the stability of the converter circuit[19-23], actually the SVG has both reactive power compensation and low order harmonic compensation function.

## A. Energy exchange between SVG DC side and AC side

The following is an analysis of the energy exchange between the DC side and the AC side of SVG in the reactive power and harmonic compensation[24-28].

For SVG system, the instantaneous active power and instantaneous reactive power of the power side are represented by $p_s$ and $q_s$ respectively, $p_{SVG}$ and $q_{SVG}$ are used to represent the instantaneous active power and instantaneous reactive power of SVG output, the instantaneous active power and instantaneous reactive power of the load side are represented by $p_L$ and $q_L$. Considering that the load current contains a partial harmonic, $p_L$ and $q_L$ can be expressed as:

$$\begin{cases} p_L = \bar{p}_L + \tilde{p}_L \\ q_L = \bar{q}_L + \tilde{q}_L \end{cases} \tag{12}$$

In the formula (12), $\bar{p}_L$ and $\bar{q}_L$ represent the DC component of active power and reactive power in the load, $\tilde{p}_L$ and $\tilde{q}_L$ represent the AC component of active power and reactive power in the load.

When only reactive power compensation with SVG:

$$\begin{cases} p_{SVG} = 0 \\ q_{SVG} = -q_L \end{cases} \tag{13}$$

At this time:

$$\begin{cases} p_s = p_L + p_{SVG} = \bar{p}_L + \tilde{p}_L \\ q_s = q_L + q_{SVG} = 0 \end{cases} \tag{14}$$

There is no active energy exchange between the SVG DC side and the AC side, and the DC side voltage should be kept constant. In theory, the DC side does not need energy storage element at this time.

When the harmonic is only compensated:

$$\begin{cases} p_{SVG} = -\tilde{p}_L \\ q_{SVG} = -\tilde{q}_L \end{cases} \tag{15}$$

At this time:

$$\begin{cases} p_s = p_L + p_{SVG} = \bar{p}_L \\ q_s = q_L + q_{SVG} = \bar{q}_L \end{cases} \tag{16}$$

At this point, SVG is needed to provide active energy, the average value of instantaneous active power is zero, but there is the AC component in $p_{SVG}$, DC side voltage will fluctuate with $p_{SVG}$.

When the reactive power and harmonic are compensated at the same time:

$$\begin{cases} p_{SVG} = -\tilde{p}_L \\ q_{SVG} = -q_L \end{cases} \tag{17}$$

At this time:

$$\begin{cases} p_s = p_L + p_{SVG} = \bar{p}_L \\ q_s = q_L + q_{SVG} = 0 \end{cases} \tag{18}$$

At this point, the voltage of the DC side will fluctuate and is similar to that of the harmonic only when it is compensated. In fact, whether it is the compensation of harmonic or reactive power, the system itself has a loss. The power loss of SVG mainly includes the loss of the inverter switching device and the magnetic hysteresis loss of the output inductor, copper loss, etc. In the case of the switching frequency, the power loss of SVG increases with the increase of the DC side voltage. The above analysis shows that the DC side voltage will change in the working process of SVG, in order to guarantee the stability of the SVG compensation performance and low power loss, the DC side voltage must be controlled [29-31].

## B. Derivation of voltage loop transfer function

For the control of DC side voltage, at present, it is used to add a voltage outer loop on the basic of the current tracking control inner loop. Figure 1 is the SVG basic circuit structure, with this figure as the object, the transfer function of SVG DC side voltage loop is derived.

In figure 1, $u_{sa}, u_{sb}$ and $u_{sc}$ represent the power grid three-phase voltage, $i_{sa}, i_{sb}$ and $i_{sc}$ represent the grid side three-phase current, $i_{fa}, i_{fb}$ and $i_{fc}$ represent the SVG output three-phase current, $u_{fa}, u_{fb}$ and $u_{fc}$ represent the three-phase voltage of the PWM inverter output, $u_{dc}$ represents the capacitor voltage of SVG DC side. The connecting inductor is $L$ and the DC side capacitance is $C$.

Assuming the power grid is three phase equilibrium, the fundamental electric potential of three-phase power grid is:

$$\begin{cases} u_{sa} = E_m \cos(\omega t) \\ u_{sb} = E_m \cos(\omega t - 120°) \\ u_{sc} = E_m \cos(\omega t + 120°) \end{cases} \tag{19}$$

Considering that the switching frequency is high, it is much higher than the grid frequency, only considering the low frequency component of the switching function $s_k (k = a,\ b,\ c)$, the PWM harmonic components are ignored[20], then:

$$\begin{cases} s_a \approx 0.5m\cos(\omega t - \theta) + 0.5 \\ s_b \approx 0.5m\cos(\omega t - \theta - 120°) + 0.5 \\ s_c \approx 0.5m\cos(\omega t - \theta + 120°) + 0.5 \end{cases} \tag{20}$$

In the formula (20), $\theta$ represents the initial phase angle of the fundamental wave for the switching function, $m$ represents the PWM modulation ratio $(m \leq 1)$.

Three phases grid side current is:

$$\begin{cases} i_{sa} \approx I_m \cos(\omega t) \\ i_{sb} \approx I_m \cos(\omega t - 120°) \\ i_{sc} \approx I_m \cos(\omega t + 120°) \end{cases} \quad (21)$$

The other three phases DC current can be described as a switching function:

$$i_{dc} = s_a i_{sa} + s_b i_{sb} + s_c i_{sc} \quad (22)$$

Taking formula (20) and (21) into formula (22), the DC side current of the inverter circuit and the grid side current simplified model are obtained as follow:

$$i_{dc} \approx 0.75 m I_m \cos\theta \quad (23)$$

According to the formula (23), the SVG voltage loop structure diagram is shown in Figure 9. Without considering the load current $i_L$ disturbance, the voltage loop transfer function can be expressed as follow:

$$W_{ov}(s) = \frac{0.75 m \cos\theta}{(\tau_v + T_{ci})Cs^2 + Cs} \quad (24)$$

In the formula (24), $1/(1 + \tau_v s)$ is the small inertial delay for the sampling of DC side voltage, $W_{ci}(s) \approx 1/(1 + T_{ci}s)$ is the current inner loop equivalent function, $0.75 m \cos\theta$ is a time varying link that represents the relationship between the DC side current and the grid side current, and the controlled object is a time-varying system, in most references, $0.75 m \cos\theta$ is reduced to $0.75 (m \leq 1)$, the controlled object is simplified as constant system.

$$W_{ov}'(s) = \frac{0.75}{(\tau_v + T_{ci})Cs^2 + Cs} \quad (25)$$

In order to compare the performance of the traditional PI control system with the time varying system and the constant system, the time varying link $K = 1 + 0.1\sin 2\pi t$ is added in the constant system of formula (25).

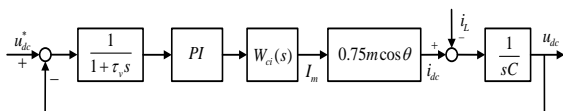$$W_{ov}''(s) = \frac{0.75K}{(\tau_v + T_{ci})Cs^2 + Cs} \quad (26)$$



Fig.9. SVG voltage loop structure diagram

## C. DC side voltage control based on PSO

In the paper, for the constant system of formula (25) and the time varying system of formula (26), simulation is done respectively to verify the control effect of the traditional PI control system and time varying system. In practice, $(\tau_v + T_{ci})$ takes 0.08s, DC side capacitance $C$ takes $0.0022 F$.
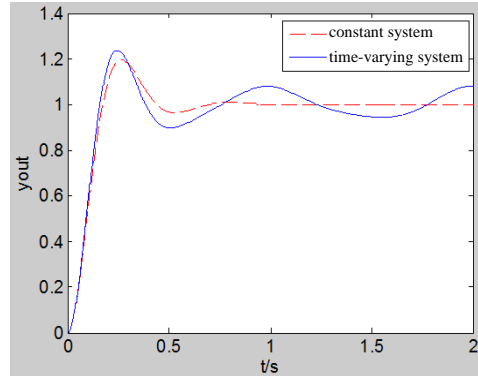


Fig.10. The step response of traditional PI control in constant system and time-varying system

Figure 10 is the unit step response of the traditional PI control for the constant system and time varying system, $k_p$ and $k_i$ are derived from the classical Z-N method, $k_p = 0.0446$, $k_i = 0.0021$. The unit step response index of the constant system of formula (25) in traditional PI control is that overshoot $= 20.8\%$, the rise time $= 0.1133s$, steady output y=1.005. The unit step response index of the time varying system of formula (26) in traditional PI control is that overshoot $= 22.5\%$, the rise time $= 0.1038s$, steady output $y = 0.9507 \sim 1.0963$. Obviously, traditional PI control for the constant system has better dynamic response characteristics, and the output basically has no steady-state error. However, when the system parameters are variable, the overshoot is obviously increased, and there is the output oscillation and steady-state error.

For the time varying system of formula (26), the PI parameters are optimized by using the off-line PSO and the on-line PSO algorithm respectively. The initial inertia weight of PSO algorithm is 0.95, the termination of the inertia weight is 0.4, the concave function decreasing strategy is used, acceleration factor $c_1 = c_2 = 2$, the algorithm population size is 120, the evolutionary algebra is the 10 generations, speed range is set to [-1,1], the range of $k_p$ is [0,10], the range of $k_i$ is [0,1]. Uses the formula (10) and (11) as the fitness function, and $\alpha_p = 0.95$, $\beta_p = 0.05$.

Figure 11 is the unit step response of time-varying system by using three control methods, and the corresponding response index is shown in table 1. Compared with the traditional PI control, the overshoot of off-line PSO-PI control is reduced, the rise time is faster, but its essence is still the PI control, so system output has oscillation, although steady-state error problem has eased, but has not yet been resolved. While the unit step

response of on-line PSO-PI control has a faster response time, almost no overshoot, no output oscillation, and steady error is almost zero, it has strong adaptability for time varying system.
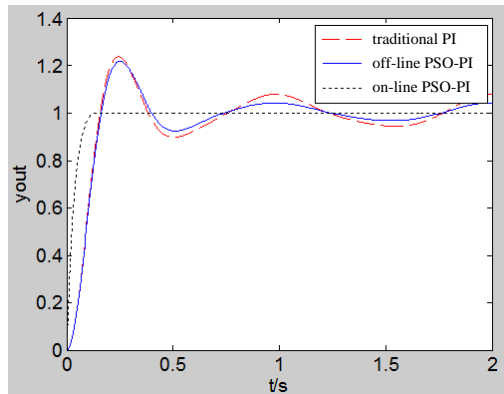


Fig.11. The step response of three control methods in time-varying system

Table 1. The step response performance index under three control modes

| | $k_p$ | $k_i$ | $\sigma$ /% | $t_r$ /s | $y$ |
|---|---|---|---|---|---|
| traditional PI | 0.0446 | 0.0021 | 22.5 | 0.1038 | 0.9507~1.0963 |
| off-line PSO-PI | 0.0421 | 0.0013 | 20.56 | 0.1094 | 0.9671~1.0354 |
| on-line PSO-PI | variation | variation | 0.2 | 0.0295 | 1.0003 |

## VI. CONCLUSIONS

With the development of power electronic devices, the load form in the power grid is becoming more and more diversified, which leads to the severe low power factor. In the field of reactive power compensation, SVG with fast response speed, smooth adjustment of reactive power, reliable operating characteristics and high precision becomes the new research hotspot in the field of reactive power compensation, represents the development direction of reactive power compensation device.

Because the stability control of the DC side voltage is very important for the SVG compensation effect, considering the disadvantages of the traditional PID control with fixed parameters and the poor adaptability[32-36], the paper introduces the PSO algorithm with global search performance into the PI parameters optimization. The paper takes the SVG voltage loop as the controlled object, and the voltage loop transfer function is deduced. Combined with PSO algorithm, on-line PSO-PI control and off-line PSO-PI control are adopted, and decreasing strategy of inertia weight function is used. The simulation results show that PSO-PI control can satisfy the situation of time varying system, and has strong adaptability.

### ACKNOWLEDGMENT

The authors wish to thank my graduate tutor Hongbin

Wu Professor and Benxian Xiao Professor and graduate student Qinglin Zhang and Peng Zhang.

REFERENCES

[1] Jie Tang, An Luo, Ke Zhou, "Design and Realization Voltage Control for Static Synchronous Compensator," *Transactions of China Electrotechnical Society*, vol.21, no.8, pp.103-106, Aug.2006.
[2] Qing Lu, Wei Ni, Xiaohui Wang, Liyun Zhuang, "Study on SVG for reactive power compensation and harmonic suppression based on active current separation," *Control and Decision Conference* (CCDC), Chinese, p. 3653-3657 2011.
[3] Hanson D.J., Woodhouse M.L., Horwill C., Monkhouse D.R., Osborne M.M., "STATCOM: a new era of reactive compensation," *Power Engineering Journal*, vol.16, no.3, pp.151-160, Mar.2002.
[4] MingFeng Yeh, MinShyang Leu, KaiMin Chen, "Design of PID Controller Using Hybrid Particle Swarm Optimization," *Fuzzy Theory and it's Applications, International Conference*, Taichung, p. 333-337, 2012.
[5] Liqing Xiao, Chengchun Han, Xiaoju Xu, Weiyong Huang, "Hybrid Genetic Algorithm and Application to PID Controllers," *Chinese Control and Decision Conference*, Xuzhou, p. 586-590, 2010.
[6] Neath M.J., Swain A.K., Madawala U.K., Thrimawithana D.J, "An Optimal PID Controller for a Bidirectional Inductive Power Transfer System Using Multi-objective Genetic Algorithm," *IEEE Transactions on Power Electronics*, vol.29, no.3, pp.1523-1531, Mar.2014.
[7] Ali Shahcheraghi, Farzin Piltan, Masoud Mokhtar, Omid Avatefipour, Alireza Khalilian, "Design a Novel SISO Off-line Tuning of Modified PID Fuzzy Sliding Mode Controller," *I.J. Information Technology and Computer Science*, vol.6, no.2, pp.72-83, Feb.2014.
[8] ZweLee Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Transactions on Energy Conversion*, vol.19, no.2, pp.384-391, Feb.2004.
[9] Wael M. Elmamlouk, Hossam E. Mostafa, Metwally A. El-Sharkawy, "PSO-Based PI Controller for Shunt APF in Distribution Network of Multi Harmonic Sources," *I.J. Intelligent Systems and Applications*, vol.5, no.8, pp.54-66, Aug.2013.
[10] Carlos A. Coello Coello, Gregorio Toscano Pulido, Maximino Salazar Lechuga, "Handling Multiple Objectives With Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol.8, no.3, pp.256-279, Mar.2004.
[11] Adel T., Abdelkader C., "A Particle Swarm Optimization Approach for Optimum Design of PID Controller for nonlinear systems," *Electrical Engineering and Software Applications, International Conference*, Hammamet, p.1-4, 2013.
[12] Aziz M.A.A., Taib M.N., Adnan R., "The Effect of Varying Inertia Weight on Particle Swarm Optimization(PSO) Algorithm in Optimizing PD Controller of Temperature Control System," *Signal Processing and its Applications*, IEEE 8th International Colloquium, Melaka, p. 545-547, 2012.
[13] Jalilvand A., Kimiyaghalam A., Ashouri A., Mahdavi M., "Advanced Particle Swarm Optimization-Based PID Controller Parameters Tuning," *Multitopic Conference, IEEE International Conference*, Karachi, p. 429-435, 2008.
[14] Khandani K., Jalali A.A., Alipoor M., "Particle Swarm

Optimization based design of disturbance rejection PID controller for time delay systems," *Intelligent Computing and Intelligent Systems, IEEE International Conference*, Shanghai, p. 862-866, 2009.

[15] Jianhong Qi ，Jinda Cai, "Error Modeling and Compensation of 3D Scanning Robot System Based on PSO-RBFNN," *International Journal on Smart Sensing and Intelligent Systems*, vol.7, no.2, pp.837-855, Feb.2014.

[16] RongJong Wai, JengDao Lee, KunLun Chuang, "Real-Time PID Control Strategy for Maglev Transportation System via Particle Swarm Optimization," *IEEE Transactions on Industrial Electronics*, vol.58, no.2, pp.629-646, Feb.2011.

[17] Zakiah Mohd Yusoff, Zuraida Muhammad, Mohd Hezri Fazalul Rahiman, Mohd Nasir Taib, "Real Time Steam Temperature Regulation Using Self-tuning Fuzzy PID Controller on Hydro-diffusion Essential Oil Extraction System," *International Journal on Smart Sensing and Intelligent System*, vol.6, no.5, pp.2055-2073, June,2013.

[18] Hidayat, Sasongko, P.H, Sarjiya, Suharyanto, "Performance Analysis of Hybrid PID-ANFIS for Speed Control of Brushless DC Motor Base on Identification Model System," *International Journal of Computer and Information Technology*, vol.2, no.3, pp.694-700, Mar.2013.

[19] Xiaobin Zhang, Yanru Zhong, "Optimal Dynamic Hierarchical Control of the STATCOM DC Side Voltage," *Proceedings of the CSEE*, vol.29, no.33, pp.60-67, Nov.2009.

[20] Qiaopo Xiong, An Luo, Xiao Wang, Xiaocong Li, Fujun Ma, Huagen Xiao, "DC Side Voltage Stability Analysis of Cascaded Static var Generator," *Automation of Electric Power Systems*, vol.38, no.3, pp.25-29, Mar.2014.

[21] Hanxiang Cheng, Xianggen Yin, ZhiHua Wang, "Operation analysis of dual three-point SVG," *Proceedings of the 12th IEEE Mediterranean, Electrotechnical Conference*, MELECON, vol.3, p.1141-1144, 2004.

[22] Li M., Chiasson J.N., Tolbert L.M., "Capacitor Voltage Control in a Cascaded Multilevel Inverter as a Static Var Generator," *CES/IEEE 5th International Conference on Power Electronics and Motion Control*, 2006, IPEMC, vol.3, p.1-5, 2006.

[23] Gang Yao, Luan He, Xin Wang, LiDan Zhou, HuaJun Yu, "The study of hybrid modulation based on cascade SVG and its DC control method," *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering* (CCECE), p. 680-685, 2015.

[24] Kusic G.L., Whyte I.A., "Three Phase, Steady-State Static var Generator Filter Design for Power Systems," *IEEE Transactions on Power Apparatus and Systems*, PAS-103(4), pp.811-818, Apr.1984.

[25] Dixon J., Moran L., Rodriguez J., "Domke R. Reactive Power Compensation Technologies: State-of-the-Art Review," *Proceedings of the IEEE*, vol.93, no.12, pp2144-2164, Dec.2005.

[26] Xiaobin Zhang, Yanru Zhong, "Research on the mathematical model and DC-voltage of static var generator," *4th IEEE Conference on Industrial Electronics and Applications*, 2009, ICIEA, p.3292-3297, 2009.

[27] Guopeng Zhao, Jinjun Liu, Xin Yang, Weiwei Gan, Zhaoan Wang, "Analysis and specification of DC side voltage in Static var Generator regarding compensation characteristics of generators," *Applied Power Electronics Conference and Exposition*, 2008, APEC 2008, Twenty-Third Annual IEEE, p.1235-1239, 2008.

[28] Zhang Z., Fahmi N.R., "Modeling and analysis of a cascade 11-level inverters-based SVG with control strategies for electric arc furnace (EAF) application," *IEE Proceedings, Generation, Transmission and Distribution*, vol.150, no.2, pp.217-223, Feb.2003.

[29] Xiaobin Zhang, Yanru Zhong, "Dynamic Model and Nonlinear Inverse System-PI Control Strategy of Static Var Generator," *Journal of System Simulation*, vol.21, no.11, pp. 3166-3170, Nov.2009.

[30] Guopeng Zhao, Minxiao Han, "Analysis of significant current tracking error and suppression of over-shoot current in grid-connected voltage source converter with abc frame control," *Power Electronics, IET*, vol.7, no.9, pp. 2347-2353, Sep.2014.

[31] Yoshida H., Fukuyama Y., Takayama S., Nakanishi Y., "A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems Considering Voltage Security Assessment," *Systems Man and Cybernetics, IEEE International Conference*, Tokyo, vol.6, p.497-502, 1999.

[32] Kiyong Kim, Rao P., Burnworth J.A., "Self-Tuning of the PID Controller for a Digital Excitation Control System," *IEEE Transactions on Industry Applications*, vol.46, no.4, pp. 1518-1524, Apr.2010.

[33] Keyu Li, "PID Tuning for Optimal Closed-Loop Performance with Specified Gain and Phase Margins," *IEEE Transactions on Control Systems Technology*, vol.23, no.3, pp. 1024-1030, Mar.2013.

[34] XingJian Jing, Li Cheng, "An Optimal PID Control Algorithm for Training Feedforward Neural Networks," *IEEE Transactions on Industrial Electronics*, vol.60, no.6, pp. 2273-2283, Jun.2013.

[35] Benxian Xiao, Jun Xiao, Rongbao Chen, "PID Controller Parameters Tuning Based-on Satisfaction for Superheated Steam Temperature of Power Station Boiler," *I.J. Information Technology and Computer Science*, vol.6, no.7, pp. 9-14, Jul.2014.

[36] Farahani, M., Ganjefar, S., Alizadeh, M., "PID controller adjustment using chaotic optimisation algorithm for multi-area load frequency control," *IET Control Theory & Applications*, vol.6, no.13, pp. 1984-1992, Jul.2012.

**Authors' Profiles**

**Yingwei Xiao** was born in Hefei, China in 1993. She has completed the B.Sc degree in electrical engineering and automation from Anhui Polytechnic University, Wuhu, China in 2015. Currently she is still pursuing her master degree in Hefei University of Technology, Hefei, China. Her research interests include automation of electric power system, micro grid power flow calculation.