Digital IIR filter design using particle swarm optimisation

Sheng Chen*

School of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ, UK
E-mail: sqc@ecs.soton.ac.uk
*Corresponding author

Bing L. Luk

Department of Manufacturing Engineering and Engineering Management,
City University of Hong Kong,
Kowloon Tong, Hong Kong
E-mail: meblluk@cityu.edu.hk

Abstract: Adaptive infinite-impulse-response (IIR) filtering provides a powerful approach for solving a variety of practical signal processing problems. Because the error surface of IIR filters is typically multimodal, global optimisation techniques are generally required in order to avoid local minima. This contribution applies the particle swarm optimisation (PSO) to digital IIR filter design in a realistic time domain setting where the desired filter output is corrupted by noise. PSO as global optimisation techniques offers advantages of simplicity in implementation, ability to quickly converge to a reasonably good solution and robustness against local minima. Our simulation study involving system identification application confirms that the proposed approach is accurate and has a fast convergence rate and the results obtained demonstrate that the PSO offers a viable tool to design digital IIR filters. We also apply the quantum-behaved particle swarm optimisation (QPSO) algorithm to the same digital IIR filter design and our results do not show any performance advantage of the QPSO algorithm over the PSO, although the former does have fewer algorithmic parameters that require tuning.

Keywords: IIR filter; global optimisation; particle swarm optimisation; PSO; system identification; quantum-behaved particle swarm optimisation; QPSO.

Biographical notes: Sheng Chen received his BEng from Huadong Petroleum Institute, Dongying, China in 1982 and his PhD from the City University, London, UK in 1986, both in Control Engineering. He obtained his Doctor of Sciences (DSc) from the University of Southampton, Southampton, UK in 2005. He has been with the School of Electronics and Computer Science, the University of Southampton since 1999. His research interests include wireless communications, machine learning, finite-precision digital controller design and evolutionary computation.

Bing L. Luk received his BSc in Electrical and Electronic Engineering from Portsmouth Polytechnic, UK in 1985, his MSc in Digital Computer Systems from Brunel University, UK in 1986 and his PhD in Robotics from the University of Portsmouth, UK in 1991. He has been with the Department of Manufacturing Engineering and Engineering Management at City University of Hong Kong, Hong Kong, China since 2000. His research interests include mobile robotics, telemedicine research, non-destructive test methods, machine learning and evolutionary computation.

1 Introduction

Adaptive infinite-impulse-response (IIR) filtering has been an active area of research for many years and many properties of IIR filters are well-known (Widrow and Stearns, 1985; Shynk, 1989). Despite the fact that the digital IIR filter design is a well-researched area, major difficulties still exist in practice. This is because the error surface or the cost function of IIR filters is generally multimodal with respect to the filter coefficients. Thus, gradient-based algorithms can easily be stuck at local minima. In

order to achieve a global minimum solution, global optimisation techniques are needed, which require extensive computations. Despite of this drawback, applying global optimisation methods to IIR filter design is attractive, since in many applications a global optimal design offers much better solution than local optimal ones. The genetic algorithm (GA) (Goldberg, 1989; Man et al., 1998) as a global optimisation method has attracted considerable attention in application to digital IIR filter design (Nambiar et al., 1992; Wilson and Macleod, 1993; White and Flockton, 1994; Ng et al., 1996). An alternative global optimisation technique known as the adaptive simulated annealing (ASA) (Ingber, 1996; Chen and Luk, 1999) has also been applied to design IIR filters (Chen et al., 2001). More recently, the work of Chen et al. (2005) developed a simple yet efficient global search method, referred to as the repeated weighted boost search (RWBS) algorithm and demonstrated its application in digital IIR filter design.

The particle swarm optimisation (PSO) is a population-based stochastic optimisation technique (Kennedy and Eberhart, 1995, 2001) inspired by social behaviour of bird flocking or fish schooling. The algorithm starts with a random initialisation of a swarm of individuals, referred to as particles, within the problem search space. It then endeavours to find a global optimal solution by simply adjusting the trajectory of each individual toward its own best location visited so far and toward the best position of the entire swarm at each evolutionary optimisation step. The attractions of the PSO method include its simplicity in implementation, ability to quickly converge to a reasonably good solution and its robustness against local minima. The PSO technique has been applied to wide-ranging practical optimisation problems successfully (Kennedy and Eberhart, 2001; Ratnaweera et al., 2004; Guru et al., 2005; Sun et al., 2006, 2008; Feng, 2006; El-Metwally et al., 2006; Soo et al., 2007; Awadallah and Soliman, 2008; Guerra and Coelho, 2008; Leong and Yen, 2008; Soliman et al., 2008; Yao et al., 2009).

There exist some works applying the PSO to IIR filter designs. A quantum-behaved particle swarm optimisation (QPSO) algorithm was employed to design IIR filter (Fang et al., 2006), while the work (Das and Konar, 2007) applied the PSO algorithm to design two-dimensional IIR filters. These works, however, were developed for the synthesis of IIR filters in the frequency domain where a set of noise-free exact frequency response points are known for the IIR filter to match. In this contribution, we propose to apply the PSO algorithm for designing digital IIR filters in a realistic time domain setting where the desired filter output is corrupted by noise. System identification application is used to demonstrate the proposed PSO approach. Compared with the results obtained using the GA, the ASA and the RWBS methods for IIR filtering available in the literature, the efficiency and solution quality of the PSO-based method appear to be slightly better. This suggests that the PSO technique offers a viable alternative to digital IIR filter design. It is believed that the QPSO algorithm offers performance advantages over the PSO

algorithm (Sun et al., 2004, 2005, 2006; Fang et al., 2006). We also apply the QPSO to the same digital IIR filter design problem. However, our experimental results do not show any performance advantage of the QPSO algorithm over the PSO algorithm, although the former does have fewer algorithmic parameters that require tuning than the latter.

2   The PSO algorithm for digital IIR filter design

We consider the digital IIR filter with the input-output relationship governed by the following difference equation:

$$y(k) + \sum_{i=1}^{M} b_i y(k-i) = \sum_{i=0}^{L} a_i x(k-i), \qquad (1)$$

where $x(k)$ and $y(k)$ are the filter's input and output, respectively and $M (\geq L)$ is the filter order. The transfer function of this IIR filter is expressed by:

$$H_M(z) = \frac{A(z)}{B(z)} = \frac{\sum_{i=0}^{L} a_i z^{-i}}{1 + \sum_{i=1}^{M} b_i z^{-i}}. \qquad (2)$$

The most commonly used approach to IIR filter design is to formulate the problem as an optimisation problem with the mean square error (MSE) as the cost function:

$$J(\mathbf{w}_H) = \mathrm{E}\left[e^2(k)\right] = \mathrm{E}\left[(d(k) - y(k))^2\right], \qquad (3)$$

where $d(k)$ is the filter's desired response, $e(k) = d(k) - y(k)$ is the filter's error signal and:

$$\mathbf{w}_H = \begin{bmatrix} \mathbf{a}^T & \mathbf{b}^T \end{bmatrix}^T = [a_0 \, a_1 \cdots a_L \ b_1 \cdots b_M]^T \qquad (4)$$

denotes the filter coefficient vector. The design goal is to minimise the MSE (3) by adjusting $\mathbf{w}_H$. In practice, ensemble operation is difficult to realise and the cost function (3) is usually substituted by the time average cost function:

$$J_N(\mathbf{w}_H) = \frac{1}{N} \sum_{k=1}^{N} e^2(k). \qquad (5)$$

During the adaptive process, the stability of the IIR filter must always be maintained. The IIR filter (1) is in its direct form. An efficient way of maintaining stability of an IIR filter is to convert the direct form to the lattice form (Gray and Markel, 1973) and to make sure that all the reflection coefficients of the IIR filter, k_i for $0 \leq i \leq M-1$, have magnitudes less than one. This approach is adopted in our design to guarantee the stability of the IIR filter during adaptation. Thus, the actual filter coefficient vector used in optimisation is:

$$\mathbf{w} = [a_0 \, a_1 \cdots a_L \ k_0 \cdots k_{M-1}]^T = [w_1 \, w_2 \cdots w_D]^T, \qquad (6)$$

where $D = M + L + 1$ is the dimension of the filter coefficient vector. For the notational convenience, the cost

function will still be denoted as J or J_N. Converting the reflection coefficients back to the direct-form coefficients b_i, $1 \le i \le M$, is straightforward (Gray and Markel, 1973). For example, for the second-order ($M = 2$) IIR filter:

$$\left.\begin{aligned} b_2 &= k_1, \\ b_1 &= k_0(1+k_1), \end{aligned}\right\} \tag{7}$$

while for the third-order ($M = 3$) IIR filter:

$$\left.\begin{aligned} b_3 &= k_2, \\ b_2 &= k_0(1+k_1)k_2 + k_1, \\ b_1 &= k_0(1+k_1) + k_1 k_2. \end{aligned}\right\} \tag{8}$$

2.1 The PSO algorithm

As described previously, the digital IIR filter design is posed as the following optimisation task:

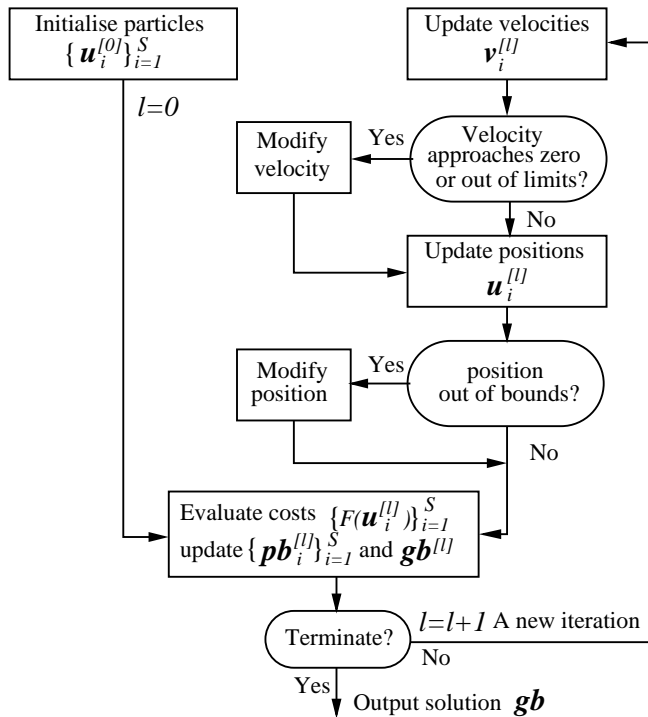$$\mathbf{w}_{\text{opt}} = \arg \min_{\mathbf{w}} F(\mathbf{w}) \tag{9}$$

$$\text{s.t.} \quad \mathbf{w} \in \mathsf{W}^D \triangleq \prod_{j=1}^{D} \mathsf{W}_j \tag{10}$$

where the cost function $F(\mathbf{w})$ is defined by $J(\mathbf{w})$ or $J_N(\mathbf{w})$ and:

$$\mathsf{W}_j \triangleq [-W_{\max,j}, \quad W_{\max,j}] \tag{11}$$

specifies the search range for w_j. The flowchart of the proposed PSO-aided IIR filter design is given in Figure 1.

Figure 1 Flowchart of the PSO algorithm

A swarm of particles, $\left\{\mathbf{u}_i^{[l]}\right\}_{i=1}^{S}$, that represent potential solutions are evolved in the search space W^D, where S is the swarm size and index l denotes the iteration step.

a   *The swarm initialisation.* Set $l = 0$ and randomly generate the initial particles, $\left\{\mathbf{u}_i^{[l]}\right\}_{i=1}^{S}$, in the search space W^D.

b   *The swarm evaluation.* Each particle $\mathbf{u}_i^{[l]}$ has a cost $F\left(\mathbf{u}_i^{[l]}\right)$ associated with it. Each particle $\mathbf{u}_i^{[l]}$ remembers its best position visited so far, denoted as $\mathbf{pb}_i^{[l]}$, which provides the cognitive information. Every particle also knows the best position visited so far among the entire swarm, denoted as $\mathbf{gb}^{[l]}$, which provides the social information. The cognitive information $\left\{\mathbf{pb}_i^{[l]}\right\}_{i=1}^{S}$ and the social information $\mathbf{gb}^{[l]}$ are updated at each iteration:

For $(i+1; i \le S; i{+}{+})$

   If $\left(F\left(\mathbf{u}_i^{[l]}\right) < F\left(\mathbf{pb}_i^{[l]}\right)\right)$ $\mathbf{pb}_i^{[l]} = \mathbf{u}_i^{[l]}$;

End for;

$i^* = \arg\min_{1 \le i \le S} F\left(\mathbf{pb}_i^{[l]}\right)$;

If $\left(F\left(\mathbf{pb}_{i^*}^{[l]}\right) < F\left(\mathbf{gb}^{[l]}\right)\right)$ $\mathbf{gb}^{[l]} = \mathbf{pb}_{i^*}^{[l]}$;

---

c   *The swarm update.* Each particle $\mathbf{u}_i^{[l]}$ has a velocity, denoted as $\mathbf{v}_i^{[l]}$, to direct its 'flying'. The velocity and position of the ith particle are updated in each iteration according to:

$$\begin{aligned} \mathbf{v}_i^{[l+1]} = {}& \xi * \mathbf{v}_i^{[l]} + c_1 * \varphi_1 * \left(\mathbf{pb}_i^{[l]} - \mathbf{u}_i^{[l]}\right) \\ & + c_2 * \varphi_2 * \left(\mathbf{gb}^{[l]} - \mathbf{u}_i^{[l]}\right), \end{aligned} \tag{12}$$

$$\mathbf{u}_i^{[l+1]} = \mathbf{u}_i^{[l]} + \mathbf{v}_i^{[l+1]}, \tag{13}$$

where ξ is the inertia weight, c_1 and c_2 are the two acceleration coefficients, while $\varphi_1 = rand()$ and $\varphi_2 = rand()$ denotes the two random variables uniformly distributed in (0, 1).

In order to avoid excessive roaming of particles beyond the search space (Guru et al., 2005), a velocity space:

$$\mathsf{V}^D \triangleq \prod_{j=1}^{D} \mathsf{V}_j = \prod_{j=1}^{D} [-V_{\max,j}, \quad V_{\max,j}] \tag{14}$$

is imposed so that:

If $\left(\mathbf{v}_i^{[l+1]}\big|_j > V_{\max,j}\right)$ $\mathbf{v}_i^{[l+1]}\big|_j = V_{\max,j}$;

If $\left(\mathbf{v}_i^{[l+1]}\big|_j < -V_{\max,j}\right)$ $\mathbf{v}_i^{[l+1]}\big|_j = -V_{\max,j}$;

where $\mathbf{v}|_j$ denotes the jth element of $\mathbf{v}$. Moreover, if $\mathbf{v}_i^{[l+1]}$ approaches zero, it is reinitialised according to:

If $\left(\mathbf{v}_i^{[l+1]}\big|_j == 0 \right)$

   If $(rand() < 0.5)$

      $\mathbf{v}_i^{[l+1]}\big|_j = \varphi_v * \gamma * V_{\max,j};$

   Else

      $\mathbf{v}_i^{[l+1]}\big|_j = -\varphi_v * \gamma * V_{\max,j};$

   End if;

End if;

---

where $\varphi_v = rand()$ is another uniform random variable in (0, 1) and γ a small positive constant.

Similarly, each $\mathbf{u}_i^{[l+1]}$ is checked to ensure that it stays inside the search space W^D. Specifically:

If $\left(\mathbf{u}_i^{[l+1]}\big|_j > W_{\max,j} \right)$

   $\mathbf{u}_i^{[l+1]}\big|_j = rand() * W_{\max,j};$

If $\left(\mathbf{u}_i^{[l+1]}\big|_j < -W_{\max,j} \right)$

   $\mathbf{u}_i^{[l+1]}\big|_j = -rand() * W_{\max,j};$

That is, if a particle is outside the search space, it is moved back inside the search space randomly, rather than forcing it to stay at the border. This is similar to the checking procedure given in Guru et al. (2005).

d   *Termination condition check.* If the maximum number of iterations, $I_{\max}$, is reached, terminate the algorithm with the solution $\mathbf{w}_{\mathrm{opt}} = \mathbf{gb}^{[I_{\max}]}$; otherwise, set $l = l+1$ and go to Step b.

The time-varying acceleration coefficient (TVAC) reported in Ratnaweera et al. (2004) is known to enhance the performance of PSO. The reason is that at the initial stages, a large cognitive component and a small social component help particles to wander around or exploit better the search space and to avoid local minima. In the later stages, a small cognitive component and a large social component help particles to converge quickly to a global minimum. This TVAC as suggested in Ratnaweera et al. (2004) is adopted, in which c_1 for the cognitive component is reduced from 2.5 to 0.5 and c_2 for the social component varies from 0.5 to 2.5 during the iterative procedure according to:

$$c_1 = 2.5 - (2.0 * l) / (1.0 * I_{\max}) \tag{15}$$

and

$$c_2 = 0.5 + (2.0 * l) / (1.0 * I_{\max}), \tag{16}$$

respectively. Our empirical experience also suggests that using a random inertia weight:

$$\xi = rand() \tag{17}$$

achieves better performance than using $\xi = 0$ or a constant ξ. An appropriate value of γ in reinitialising zero velocity found empirically for our IIR filter design application is $\gamma = 0.7$. The swarm size S depends on how hard the optimisation problem (9) is. For our IIR filter design problem, choosing the maximum number of iterations as $I_{\max} = 20$ is often adequate. A typical choice of the maximum velocity bound is $V_{\max,j} = 2W_{\max,j}$. The search space is specified by the design problem. In particular, $W_{\max,j}$ is smaller than but close to 1.0, for $L + 2 \le j \le D$, due to the IIR filter stability consideration.

2.2 The QPSO algorithm

The QPSO algorithm (Sun et al., 2004, 2005, 2006) is also applied to solve the IIR filter design problem (9) within the search space (10) and the flowchart of the QPSO-based IIR filter design is shown in Figure 2.

Figure 2 Flowchart of the QPSO algorithm
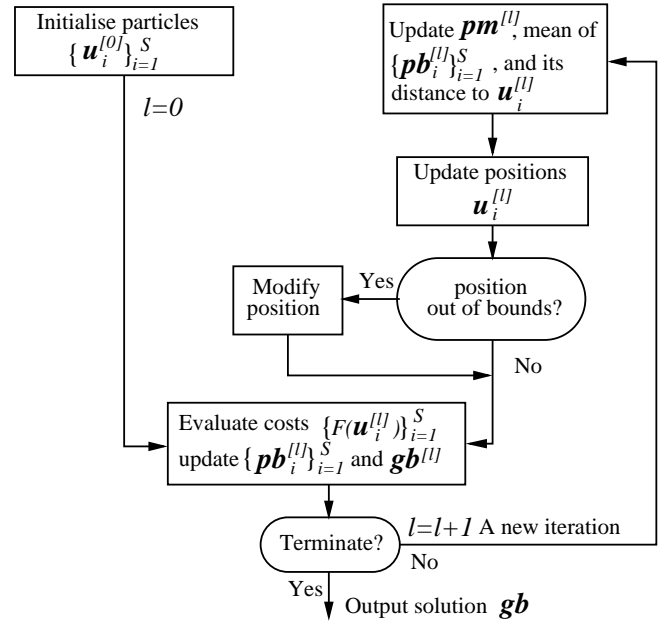
a   *The swarm initialisation.* This is identical to Step a of the PSO algorithm.

b   *The swarm evaluation.* This is also identical to Step b of the PSO algorithm.

c   *The swarm update.* The mean position of $\left\{ \mathbf{pb}_i^{[l]} \right\}_{i=1}^{S}$ is calculated as:

$$\mathbf{pm}^{[l]} = \frac{1}{S} \sum_{i=1}^{S} \mathbf{pb}_i^{[l]}, \tag{18}$$

and the particles are updated according to[1]:

For $(i+1; i \leq S; i++)$

$\quad \varphi_1 = rand();$

$\quad \varphi_2 = rand();$

$\quad \varphi_u = rand();$

$\quad$ For $(j=1; j \leq D; j++)$

$\qquad p = \left(\varphi_1 * \mathbf{pb}_i^{[l]} \big|_j + \varphi_2 * \mathbf{gb}^{[l]} \big|_j \right) \big/ (\varphi_1 + \varphi_2);$

$\qquad u = c_e * \left| \mathbf{pm}^{[l]} \big|_j - \mathbf{u}^{[l]} \big|_j \right|;$

$\qquad$ If $(rand() > 0.5)$

$\qquad\quad \mathbf{u}_i^{[l+1]} \big|_j = p - u * \log(1/\varphi_u);$

$\qquad$ Else

$\qquad\quad \mathbf{u}_i^{[l+1]} \big|_j = p + u * \log(1/\varphi_u);$

$\quad$ End for;

End for;

where c_e is the contraction-expansion coefficient.

Each $\mathbf{u}_i^{[l+1]}$ is then checked to ensure that it stays inside the search space W^D. Specifically, if a particle is outside the search space, it is moved back inside the search space randomly, as in the case of the PSO algorithm.

d *Termination condition check.* As in the PSO algorithm, if the maximum number of iterations, $I_{\max}$, is reached, the algorithm is terminated with the solution $\mathbf{w}_{\mathrm{opt}} = \mathbf{gb}^{[I_{\max}]}$; otherwise, it sets $l = l+1$ and goes to Step b.

The QPSO algorithm has fewer algorithmic parameters that require tuning than the PSO algorithm. The contraction-expansion coefficient c_e is critical to the performance of the algorithm and it is typically determined by experiment. The empirical formula for computing c_e (Sun et al., 2004, 2005, 2006):

$$c_e = c_{\max} - (c_{\max} - c_{\min}) * l / I_{\max} \tag{19}$$

is used in our experiment, where appropriate values for $c_{\max}$ and $c_{\min}$ can only be found empirically.

3 System identification application

System identification application based on adaptive IIR filter, as depicted in Figure 3, is used in the experiment. In this configuration, the unknown plant has a transfer function $H_S(z)$ and the PSO algorithm described in the previous section is employed to adjust the IIR filter that is used to model the system. When the filter order M is smaller than the system order, local minima problems can be encountered (Shynk, 1989) and this is used to simulate a

multimodal environment. The signal-to-noise (SNR) ratio of the system is defined as:

$$\mathrm{SNR} = \sigma_d^2 / \sigma_n^2. \tag{20}$$

Here, σ_n^2 is the noise variance and the system signal variance σ_d^2 is given by:

$$\sigma_d^2 = \frac{\sigma_x^2}{\sqrt{-1}} \oint H_S(z) H_S(z^{-1}) \frac{dz}{z}, \tag{21}$$

where σ_x^2 is the input signal variance. A numerical evaluation of the filter power $\oint H_S(z) H_S(z^{-1}) \frac{dz}{z}$ can be found in (Åström, 1970). The search ranges for the filter coefficients are $|a_i| \leq 1.0$ and $|\kappa_i| \leq 0.99$. Thus, the search space (10) is specified by $W_{\max, j} = 1.0$ for $1 \leq j \leq L+1$ and $W_{\max, j} = 0.99$ for $2 + L \leq j \leq D$. The results obtained by the PSO as well as QPSO-based IIR filter designs are compared with those obtained using the GA (White and Flockton, 1994), the ASA (Chen et al., 2001) and the RWBS (Chen et al., 2005).

Figure 3 Schematic of adaptive IIR filter for system identification configuration (see online version for colours)
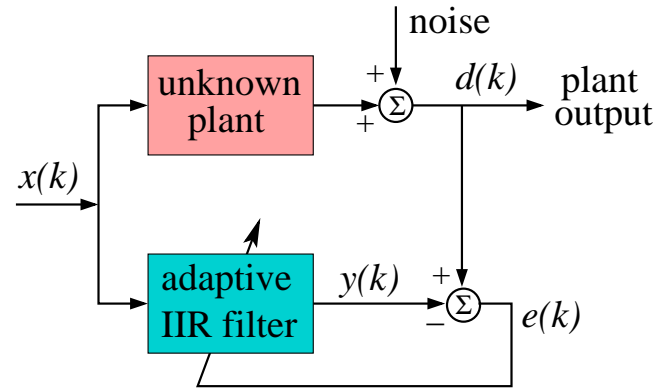
Example one

This example was taken from Shynk (1989). The system and filter transfer functions respectively are:

$$H_S(z) = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}} \tag{22}$$

and

$$H_M(z) = \frac{a_0}{1 + b_1 z^{-1}}. \tag{23}$$

The analytical cost function J in this case is known when the input is a white sequence and $\sigma_n^2 = 0$. The cost function has a global minimum at $\mathbf{w}_{\mathrm{global}} = [-0.311 \ -0.906]^T$ with the value of the normalised cost function $J(\mathbf{w}_{\mathrm{global}})/\sigma_d^2 = 0.2772$ and a local minimum at $\mathbf{w}_{\mathrm{local}} = [0.114 \ \ 0.519]^T$.

Figure 4   Convergence performance averaged over 100 random experiments for example one obtained using, (a) the PSO and QPSO (b) the ASA (c) the RWBS

(a)

(b)

(c)
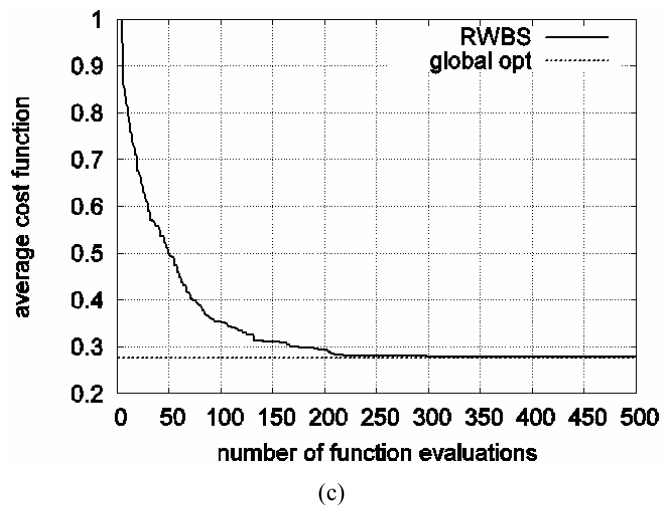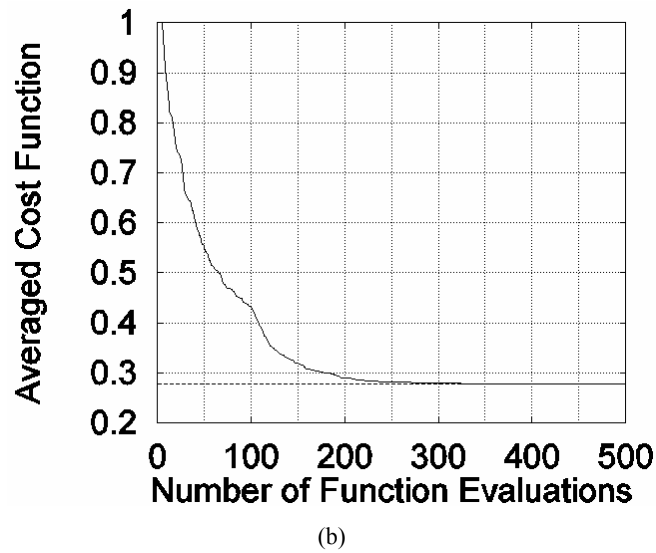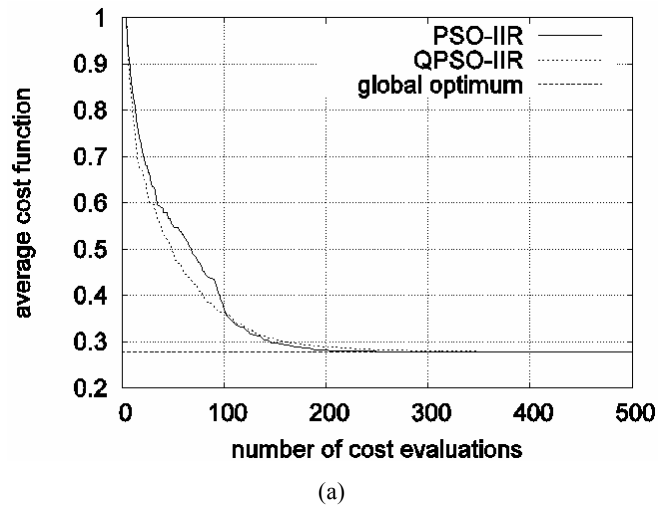
Notes: The dashed line indicates the global minimum.
       The ASA result is quoted from Chen et al. (2001)
       and the RWBS result from Chen et al. (2005).

Figure 5   Distribution of the solutions, (a_0, b_1) shown as small circles, obtained in the 100 random experiments for example one using, (a) the PSO (b) the RWBS (c) the QPSO

(a)

(b)

(c)
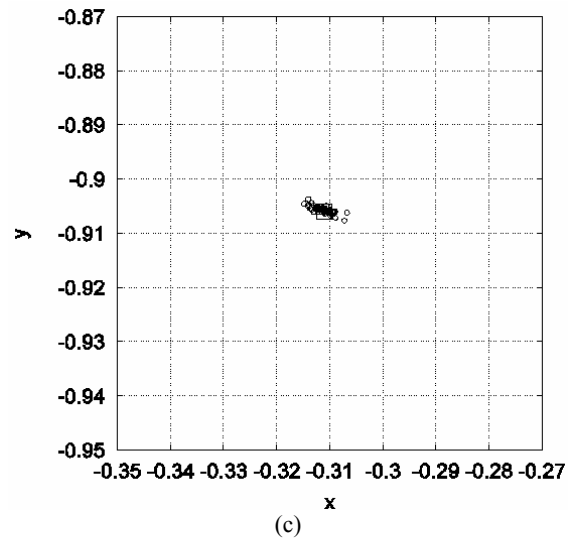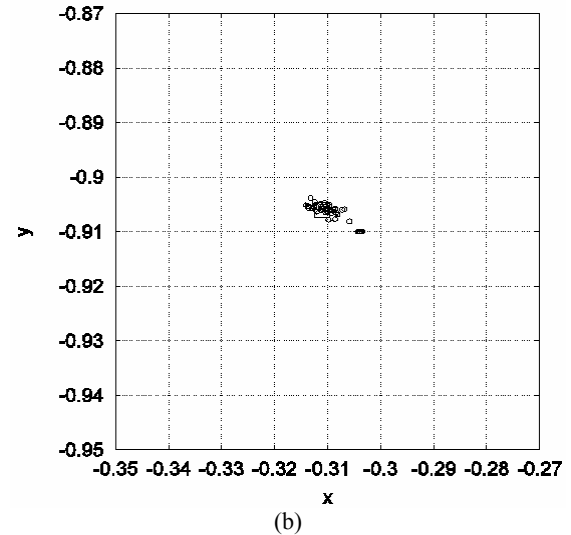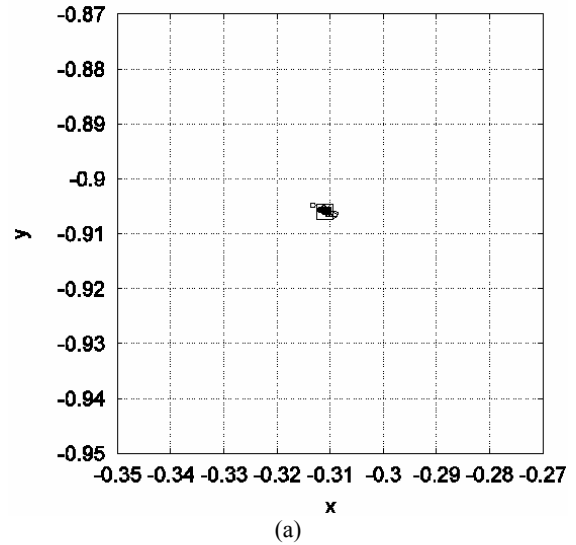
Note: The large square indicates the global minimum.

For the PSO, the swarm size was set to $S = 30$ and the maximum number of iterations to $I_{max} = 20$. Figure 4(a) depicts the evolution of the normalised cost function averaged over 100 different random runs obtained using the PSO. Under the identical experimental conditions, the ASA (Chen et al., 2001) and the RWBS (Chen et al., 2005) were also applied to this example and the results obtained are reproduced in Figures 4(b) and 4(c), respectively. It can be seen from Figure 4 that both the ASA and RWBS had a similar convergence speed, requiring on average 300 cost function evaluations to converge to the global optimal solution while the PSO had a slightly faster convergence speed as it required a slightly fewer cost function evaluations, about 250 on average, to converge. The work of White and Flockton (1994) applied a GA to the same example. The result given in White and Flockton (1994) shows that the GA was slower to converge to the global minimum, requiring an average of 600 cost function evaluations to do so.

For the QPSO with the swarm size $S = 30$ and the maximum number of iterations $I_{max} = 20$, appropriate values for c_{max} and c_{min} were found empirically to be $c_{max} = 1.4$ and $c_{min} = 0.6$. The learning curve of the QPSO is also depicted in Figure 4(a) in comparison with that obtained by the PSO. For this example, we do not see any convergence advantage of the QPSO over the PSO. From Figure 4(a), it can be seen that initially the QPSO algorithm converged faster than the PSO algorithm but after about 100 cost evaluations it became slower than the PSO. In fact, the QPSO algorithm took on average 300 cost function evaluations to converge to the global optimal solution. The distribution of the solutions obtained in the 100 experiments by the PSO algorithm is shown in Figure 5, in comparison with the solution distributions obtained by the RWBS algorithm and the QPSO algorithm, which confirms that the solution quality of the PSO algorithm was better than the other two algorithms.
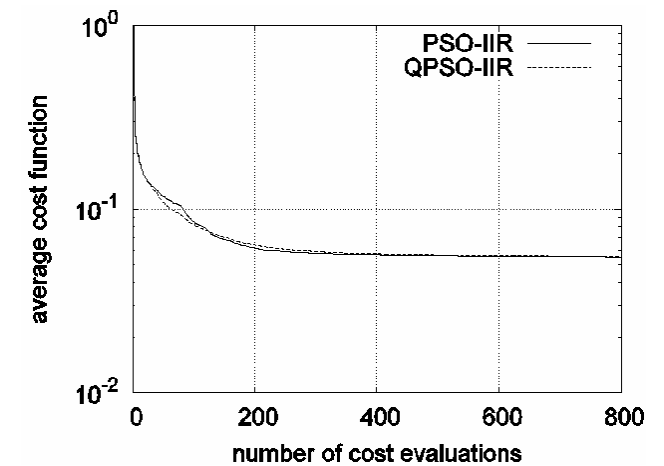
Example two

This was a third-order system with the transfer function given by:

$$H_S(z) = \frac{-0.3 + 0.4z^{-1} - 0.5z^{-2}}{1 - 1.2z^{-1} + 0.5z^{-2} - 0.1z^{-3}}. \tag{24}$$

In the simulation, the system input $x(k)$ was a uniformly distributed white sequence, taking values from $(-1, 1)$ and the SNR = 30 dB. The data length used to calculate the MSE (5) was $N = 2,000$. When a reduced-order filter with $M = 2$ and $L = 1$ was used, the MSE became multimodal and the gradient-based IIR filter design performed poorly as was demonstrated in Chen et al. (2001). It was also clearly shown in Chen et al. (2005) that there were many global optimal solutions.
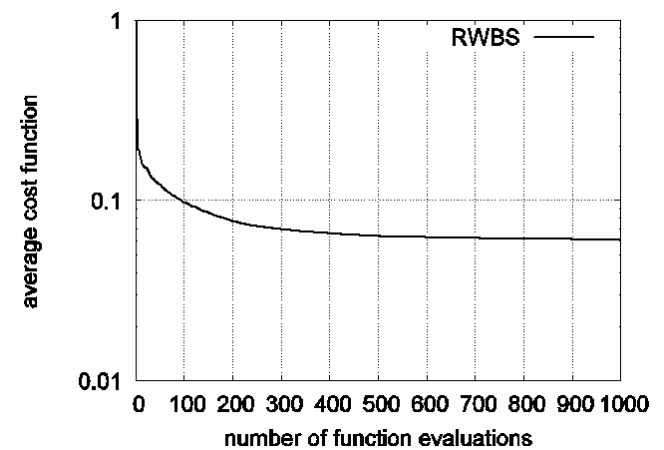
Figure 6 Convergence performance averaged over 500 random experiments for example two obtained using, (a) the PSO and QPSO (b) the ASA (c) the RWBS
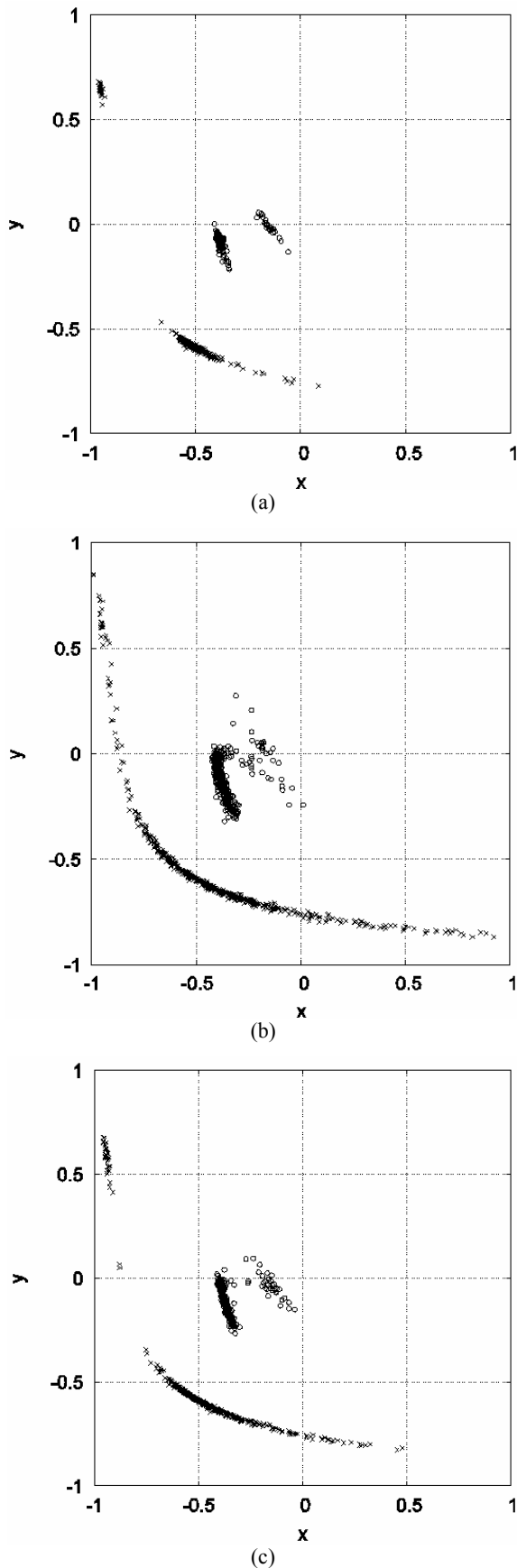

(a)
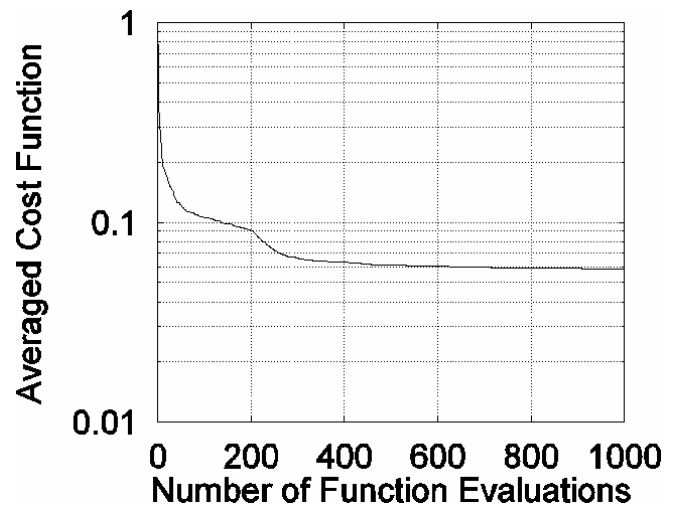
(b)

(c)

Note: The ASA result is quoted from Chen et al. (2001) and the RWBS result from Chen et al. (2005).

Figure 7   Distribution of the solutions, (a_0, a_1) shown as circles and (κ_0, κ_1) shown as crosses, obtained in the 500 random experiments for example two using, (a) the PSO (b) the RWBS (c) the QPSO

(a)

(b)

(c)

The swarm size $S = 40$ and the maximum number of iterations $I_{max} = 20$ were used for both the PSO and QPSO algorithms, while $c_{max} = 1.0$ and $c_{min} = 0.5$ were adopted by the QPSO algorithm. The convergence performance of the four algorithms, the PSO, QPSO, ASA and RWBS, averaged over 500 experiments are depicted in Figure 6. The results of the ASA and RWBS are reproduced from Chen et al. (2001, 2005), respectively. Again, the ASA and RWBS algorithms are seen to have a similar convergence speed. However, the PSO converged faster than the ASA and RWBS algorithms. From Figure 6(a), it can also be seen that the PSO algorithm converged slightly faster than the QPSO algorithm to a global minimum, although the latter had slightly faster initial convergence speed. The distribution of the solutions obtained in the 500 random experiments by the PSO algorithm is illustrated in Figure 7, in comparison with the solution distributions obtained by the RWBS and QPSO algorithms.

4   Conclusions

This contribution has applied a popular global optimisation algorithm, known as the PSO, to the digital IIR filter design. Simulation study involving system identification application has demonstrated that the PSO is easy to implement, is robust against local minimum problem and has a fast convergence speed. In particular, compared with the results of using other global optimisation techniques for adaptive IIR filtering available in the literature, the efficiency and the solution quality of the PSO appear to be slightly better. Thus, this study has confirmed that the PSO offers a viable alternative design approach for IIR filtering. We have also implemented the QPSO algorithm to the same digital IIR filter design problem. But our results obtained do not show any performance advantage of the QPSO algorithm over the PSO, although the QPSO does have fewer algorithmic parameters that require tuning than the PSO algorithm.

References

Åström, K.J. (1970) *Introduction to Stochastic Control Theory*, Academic Press, New York.

Awadallah, M.A. and Soliman, H.M. (2008) 'An adaptive power system stabiliser based on fuzzy and swarm intelligence', *Int. J. Modelling, Identification and Control*, Vol. 5, No. 1, pp.55–65.

Chen, S. and Luk, B.L. (1999) 'Adaptive simulated annealing for optimization in signal processing applications', *Signal Processing*, Vol. 79, No. 1, pp.117–128.

Chen, S., Istepanian, R. and Luk, B.L. (2001) 'Digital IIR filter design using adaptive simulated annealing', *Digital Signal Processing*, Vol. 11, No. 3, pp.241–251.

Chen, S., Wang, X.X. and Harris, C.J. (2005) 'Experiments with repeating weighted boosting search for optimization in signal processing applications', *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol. 35, No. 4, pp.682–693.

Das, S. and Konar, A. (2007) 'A swarm intelligence approach to the synthesis of two-dimensional IIR filters', *Engineering Applications of Artificial Intelligence*, Vol. 20, No. 8, pp.1086–1096.

El-Metwally, K.A., Elshafei, A-L. and Soliman, H.M. (2006) 'A robust power-system stabiliser design using swarm optimisation', *Int. J. Modelling, Identification and Control*, Vol. 1, No. 4, pp.263–271.

Fang, W., Sun, J. and Xu, W-B. (2006) 'Design IIR digital filters using quantum-behaved particle swarm optimization', *Proc. 2nd Int. Conf. Natural Computation*, Xian, China, 24–28 September, Part II, pp.637–640.

Feng, H-M. (2006) 'Self-generation RBFNs using evolutional PSO learning', *Neurocomputing*, Vol. 70, Nos. 1–3, pp.41–251.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Gray, A.H. Jr. and Markel, J.D. (1973) 'Digital lattice and ladder filter synthesis', *IEEE Trans. Audio and Electroacoustics*, Vol. AU-21, pp.491–500.

Guerra, F.A. and Coelho, L.S. (2008) 'Multi-step ahead nonlinear identification of Lorenz's chaotic system using radial basis function neural network with learning by clustering and particle swarm optimisation', *Chaos, Solitons and Fractals*, Vol. 35, No. 5, pp.967–979.

Guru, S.M., Halgamuge, S.K. and Fernando, S. (2005) 'Particle swarm optimisers for cluster formation in wireless sensor networks,' *Proc. 2005 Int. Conf. Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, Australia, 5–8 December, pp.319–324.

Ingber, L. (1996) 'Adaptive simulated annealing (ASA): lessons learned', *J. Control and Cybernetics*, Vol. 25, pp.33–54.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Proc. 1995 IEEE Int. Conf. Neural Networks*, Perth, Australia, 27 November–1 December, Vol. 4, pp.1942–1948.

Kennedy, J. and Eberhart, R. (2001) *Swarm Intelligence*, Morgan Kaufmann.

Leong, W-F. and Yen, G.G. (2008) 'PSO-based multiobjective optimization with dynamic population size and adaptive local archives', *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol. 38, No. 5, pp.1270–1293.

Man, K.F., Tang, K.S. and Kwong, S. (1998) *Genetic Algorithms: Concepts and Design*, Springer-Verlag, London.

Nambiar, R., Tang, C.K.K. and Mars, P. (1992) 'Genetic and learning automata algorithms for adaptive digital filters', *Proc. ICASSP 1992*, Vol. 4, pp.41–44.

Ng, S.C., Leung, S.H., Chung, C.Y., Luk, A. and Lau, W.H. (1996) 'The genetic search approach: a new learning algorithm for adaptive IIR filtering', *IEEE Signal Processing Magazine*, November Issue, pp.38–46.

Ratnaweera, A., Halgamuge, S.K. and Watson, H.C. (2004) 'Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients', *IEEE Trans. Evolutionary Computation*, Vol. 8, No. 3, pp.240–255.

Shynk, J.J. (1989) 'Adaptive IIR filtering', *IEEE ASSP Magazine*, April Issue, pp.4–21.

Soliman, H.M., Awadallah, M.A. and Nadim Emira, M. (2008) 'Robust controller design for active suspensions using particle swarm optimisation', *Int. J. Modelling, Identification and Control*, Vol. 5, No. 1, pp.66–76.

Soo, K.K., Siu, Y.M., Chan, W.S., Yang, L. and Chen, R.S. (2007) 'Particle-swarm-optimization-based multiuser detector for CDMA communications', *IEEE Trans. Vehicular Technology*, Vol. 56, No. 5, pp.3006–3013.

Sun, J., Xu, W-B. and Feng, B. (2004) 'A global search strategy of quantum-behaved particle swarm optimization', *Proc. 2004 IEEE Conf. Cybernetics and Intelligent Systems*, Singapore, 1–3 December, pp.111–116.

Sun, J., Xu, W-B. and Feng, B. (2005) 'Adaptive parameter control for quantum-behaved particle swarm optimization on individual level', *Proc. 2005 IEEE Int. Conf. Systems, Man and Cybernetics*, Big Island, Hawaii, 10–12 October, Vol. 4, pp.3049–3054.

Sun, J., Xu, W-B. and Liu, J. (2006) 'Training RBF neural network via quantum-behaved particle swarm optimization', *Proc. ICONIP 2006*, Hong Kong, China, 3–6 October, pp.1156–1163.

Sun, T-Y., Liu, C-C., Tsai, T-Y. and Hsieh, S-T. (2008) 'Adequate determination of a band of wavelet threshold for noise cancellation using particle swarm optimization', *Proc. CEC 2008*, Hong Kong, China, 1–6 June, pp.1168–1175.

White, M.S. and Flockton, S.J. (1994) 'Genetic algorithms for digital signal processing', *Lecture Notes in Computing Science*, Vol. 865, pp.291–303.

Widrow, B. and Stearns, S.D. (1985) *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ.

Wilson, P.B. and Macleod, M.D. (1993) 'Low implementation cost IIR digital filter design using genetic algorithms', *Workshop on Natural Algorithms in Signal Processing*, Chelmsford, Essex, UK, pp.4/1–4/8.

Yao, W., Chen, S., Tan, S. and Hanzo, L. (2009) 'Particle swarm optimisation aided minimum bit error rate multiuser transmission', *Proc. ICC 2009*, Dresden, Germany, 14–18 June, 5 pages.

Notes

1 In the original QPSO (Sun et al., 2004, 2005, 2006), the uniform random variables φ_1, φ_2 and φ_u are called at the element-level inside the inner loop for j but we find that the performance is better by calling them at the vector-level outside the inner loop for j.