An application of swarm intelligence for the design of IIR digital filters

Ranjit Singh Chauhan*
Department of Electronics and Communication Engineering, JMIT Radaur, Dist. Yamuna Nagar, Haryana 135133, India
E-mail: chauhan443@yahoo.co.in
*Corresponding author

Sandeep Kumar Arya
Department of Electronics and Communication Engineering, Guru Jambheshwar University of Science and Technology, Hisar, Haryana 125001, India
E-mail: aryasandeep@rediffmail.com

Abstract: This paper presents the swarm intelligence (SI)-based particle swarm optimisation (PSO) in context of designing infinite-impulse response (IIR) digital filters. IIR filters are the part of digital filters with recursive responses. Since the error surface of IIR digital filters is generally non-linear and multimodal, global optimisation techniques are required in order to avoid local minima. The particle swarm optimisation algorithm is then applied for calculating optimal coefficients of IIR digital filters and comparison is done preferably with filter design tool and other heuristic techniques. The simulation results of benchmark filter are discussed in this paper and can be efficiently used for IIR digital filter design.

Keywords: digital signal processing; DSP; digital filter; infinite-impulse response; IIR; genetic algorithm; GA; simulated annealing; SA; swarm intelligence; SI; particle swarm optimisation; PSO.

Reference to this paper should be made as follows: Chauhan, R.S. and Arya, S.K. (2013) ‘An application of swarm intelligence for the design of IIR digital filters’, Int. J. Swarm Intelligence, Vol. 1, No. 1, pp.3–18.

Biographical notes: Ranjit Singh Chauhan is an Assistant Professor in the Department of Electronics and Communication Engineering, JMIT Radaur, Haryana, India. He received his Instrumentation and Control Engineering degree from the Regional Engineering College, Jalandhar, Punjab in 1999, and MTech in Electronic and Communication Engineering from G.N.E. Ludhiana, Punjab in 2007. He is currently pursuing his PhD in Electronics and Communication Engineering from Guru Jambheshwar University of Science and Technology, Hisar. He has 13 years experience in teaching and research. He is member of IACSIT and senior member of IASET. His research interests include artificial intelligence, evolutionary computation and digital signal processing.
1 Introduction

Swarm intelligence (SI) is a new technique of artificial intelligence (AI) which is inspired by real world insects. There comes a lot of swarms under this category the social insects such as ants, termites, bees, and wasps and by swarming, flocking, herding, and shoaling phenomena in vertebrates. SI is recently invented high performance technique for optimisation which is objective of all the engineering application. Beni and Wang introduced the expression ‘SI’ in 1989, in the context of cellular robotic systems. SI, one of the most recent sub fields of artificial life, can be defined as any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies (Tarasewich and McMullen, 2002). SI is an innovative computational and behavioural metaphor for solving distributed problems that originally took its inspiration from the biological. While nature remains a fundamental source of inspiration for researchers in SI, new ideas originating from the most different areas in engineering and computer science are emerging and strongly influencing the field. Despite this continuous evolving of the SI definition, key principles such as self-organisation, distributions, parallelism, and exploitation of local communication mechanisms among relatively simple individuals are emerging as invariants of this innovative computational and behavioural metaphor. Examples of systems like this can be found abundant in nature, including ant colonies, bird flocking, animal herding, honey bees, bacteria, and many more. SI models have many features in common with evolutionary algorithms (EAs). Like EA, SI models are population-based (Zhou and He, 2007). The system is initialised with a population of individuals (i.e., potential solutions). These individuals are then manipulated over many generations by ways of mimicking social behaviours of insects or animals, in an effort to find the optima. Unlike EA, SI models do not use evolutionary operators such as crossover and mutation. A potential solution simply ‘flies’ through the search space by modifying itself according to its relationship with other individuals in the population and the environment.

Digital filtering is one of the most important tools of digital signal processing (DSP). Digital filters are capable of performance specifications that would, at best, be extremely difficult, if not impossible, to achieve with the help of an analogue implementation. Furthermore, the characteristics of a digital filter can be easily changed under software control. Digital filter can be broadly classified into two groups: recursive [infinite-impulse response (IIR)] and non-recursive [finite-impulse response (FIR)] (Hong et al., 2009). An IIR filter can provide a much better performance than the FIR
An application of swarm intelligence for the design of IIR digital filters

filters having the same number of coefficients (Proakis and Manolakis, 2007; Ingle and Proakis, 2004). However, IIR filters might have a multi-modal error surface. Therefore, a reliable design method proposed for IIR filters must be based on a global search procedure. Digital IIR filters are widely used in the fields of automatic control, telecommunications, speech processing etc. There are two main methods for IIR digital filters design. As the error surface of IIR digital filters is usually non-linear and multimodal, conventional gradient-based design methods may easily get stuck in the local minima of error surface (Antoniou, 2005). Recently EA (Goldberg, 2005; Engelbrecht, 2005; Dai et al., 2010), such as artificial bee colony (ABC), particle swarm optimisation (PSO), ant colony optimisation (ACO), genetic algorithm (GA), simulated annealing (SA), tabu search (TS), etc., have attracted a lot attention for global optimisation problems.

Analytical or simple iterative methods normally lead to sub-optimal designs. Consequently, there is a great need of optimisation methods (heuristic type) that can be used to design digital filters that would satisfy prescribed specifications (Skaf and Stephen, 2008; Vaccaro and Harrison, 1996; Zhang and Iwakura, 1996; Argenti and Del Re, 1998). SI has become a research interest to many research scientists from various areas in recent years. The SI can be defined as any attempt for designing algorithms or distributed problem-solving devices inspired by the collective behaviour of insects and other animal societies. PSO, which was originally proposed by Kennedy and Eberhart (Fang et al., 2006), is a novel algorithm totally inspired by birds flocking in the sky or fish schooling in the sea. Benvenuto et al. (1992) described the salient features of using a SA algorithm in the context of designing digital filters with linear phase digital filter. The algorithm is then applied to the design of FIR filter. The result was not impressive. Moreover, it is computationally very expensive. Liang et al. (2003) used GA to design 1-D IIR filter with canonical-signed-digit coefficients restricted to low pass (LP) filter. Ahmad and Antoniou (2006) gave FIR filters and equalisers through the use of GA. Consequently GA requires a long computation. Oliveira et al. (2007) presented an innovative approach for designing linear FIR filters by using non-linear stochastic global optimisation based on SA techniques. Jung et al. (2008) found the design method of a linear phase finite word length finite-duration impulse response (FIR) filter through SA. Apaydin (2009) created a development of reduced-delay FIR digital filters with amplitude requirements using equiripple passband, peak constrained least squared stopband, and nearly constant group delay in passband for real time applications. Yong and Chen (2010) proposed a numerical optimisation algorithm for FIR digital filter based upon the SA paradigm and its implementation is known as fuzzy adaptive SA. Chen and Luk (2010) provided a powerful approach for solving a variety of practical signal processing problems. This contribution involves system identification application. Gao and Liao (2011) explored a novel approach of tuning PID controller which is proposed by the improved PSO. Coelho et al. (2012) concluded that the term ‘SI’ is used to describe algorithms and distributed problem solvers inspired by the collective behaviour of insect colonies and other animal societies. The basic drawback of preceding design methods is that the large amount of computation time is required. It is clearly indicated that all the above methods can mainly be used to design FIR digital filters. The outline of this paper is to introduce SI as an adaptive learning tool for the design of IIR digital filters. The proposed algorithm has strong search capability and is superior to the SA and GA. To test the optimisation procedure, the proposed algorithm is implemented in MATLAB and results are found to be very encouraging. In this paper, articles are organised as follows:
In Section 2, a brief introduction to SI approach is reported. PSO related to IIR digital filter design is given in Section 3. Two designed examples are used to verify the proposed method and simulation results of experiments are illustrated in Section 4. The conclusion and future scope is discussed in end of this paper.

2 Swarm intelligence

SI is an AI technique that focuses on studying the collective behaviour of a decentralised system made up by a population of simple agents interacting locally with each other and with the environment. Although there is typically no centralised control dictating the behaviour of the agents, local interactions among the agents often cause a global pattern to emerge. Two of the most successful SI techniques currently in existence are ACO and PSO. ACO is a metaheuristic optimisation algorithm that can be used to find approximate solutions to difficult combinatorial optimisation problems. The ant system is a general-purpose heuristic algorithm, which can be used to solve diverse combinatorial optimisation problems. This work has been lead by Marco Dorigo at the Politecnico di Milano (Tarasewich and McMullen, 2002; Gueret et al., 2012). Ant algorithm is multi agent system in which the behaviour of the each single agent is called artificial ant, is inspired by the behaviour of real ants. ACO is a paradigm for designing metaheuristic algorithms for combinatorial optimisation problems. PSO is a global minimisation technique for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space.

PSO is an EA developed by Eberhart and Kennedy in 1995 (Luitel and Venayagamoorthy, 2008). It is a population-based search algorithm and is inspired by the observation of natural habits of bird flocking and fish schooling. PSO is a flexible, robust population-based stochastic search/optimisation technique with implicit parallelism, which can easily handle with non-differentiable objective functions, unlike traditional optimisation methods (Yao et al., 1999). It is developed through simulation of bird flocking in multidimensional space. Bird flocking optimises a certain objective function. Each particle knows its best value so far (pbest). This information corresponds to personal experiences of each particle. Moreover, each particle knows the best value so far in the group (gbest) among pbests. Namely, each particle tries to modify its position using the following information:

- the distance between the current position and pbest
- the distance between the current position and gbest.

Each individual is treated as a volume-less particle in the D-dimensional space, with the position and velocity of the particle represented as \( X_i = (X_{i1}, X_{i2}, \ldots, X_{id}) \) and \( V_i = (V_{i1}, V_{i2}, \ldots, V_{id}) \). The particles move according to the following equation:

\[
\begin{align*}
V_{id} &= w * V_{id} + c_1 * r_1 *(P_{id} - X_{id}) + c_2 * r_2 *(P_{g} - X_{id}) \\
X_{id} &= X_{id} + V_{id}
\end{align*}
\]

(1)

where \( c_1 \) and \( c_2 \) are acceleration coefficients, \( r_1 \) and \( r_2 \) are two random numbers. Vector \( P_i = (P_{i1}, P_{i2}, \ldots, P_{id}) \) is the best previous position (the position giving the best fitness
An application of swarm intelligence for the design of IIR digital filters

value) of particle $i$ called $p_{best}$, and vector $P_g = (P_{g1}, P_{g2}, \ldots, P_{gD})$ is the position of the best particle among all the particles in the population and called $g_{best}$. Parameter $w$ is the inertia weight introduced to accelerate the convergence speed of the PSO.

PSO is similar to GA in that the system is initialised with a population of random solutions. It is unlike a GA, however, in that each potential solution is also assigned a random velocity, and the potential solutions, called particles, are then ‘flown’ through the problem space (William and Miller, 2004; Rout et al., 2012). Each particle keeps track of its coordinates in the problem space which are associated with fitness (best solution) it has achieved so far. The fitness value is also stored which is called $p_{best}$. Another ‘best’ value that is tracked by the global version of the particle swarm optimiser is the overall best value, and its location, obtained so far by any particle in the population. This location is called $g_{best}$. The process for implementing PSO is as follow:

- Initialise a population (array) of particles with random position and velocities on $d$ dimensions in the problem space.
- For each particle, evaluate the desired optimisation fitness function in $d$ variables.
- Compare particle’s fitness evaluation with particle’s $p_{best}$. If current value is better then $p_{best}$, then set $p_{best}$ location equal to the current location in $d$–dimension space.
- Compare fitness evaluation with the population’s overall previous best. If current value is better then $g_{best}$, then reset $g_{best}$ to the current particle’s array index and value.
- Change the value and position of the particle according to above equations.
- Loop to step 2, until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generation).

The pseudo code for PSO is represented in Figure 1.

![Figure 1](image)

The advancement in integrated circuit technology results from the increasing computing power, the simulation of evolutionary systems is becoming more and more tractable.
Most applications of EAs are to the solution of static problems. Many real-world systems, however, change state frequency. These system state changes result in a requirement for frequent, sometimes almost continuous, reoptimisation. It has been demonstrated that PSO can be successfully applied to tracking and optimising dynamic systems (Zhou and He, 2007; Lai and Lin, 2012).

3 PSO for designing IIR digital filters

In general, digital filters are classified as recursive and non-recursive filters. The response of recursive or IIR filters is dependent on one or more of its past output. IIR digital filters have lot of advantages over non-recursive filters and found to be more effective. It has a much sharper transition characteristic for a given filter order. Like analogue filters with poles, an IIR filter usually has non-linear phase characteristics. If such filter subjected to an impulse then its output need not necessarily become zero. The IIR of such a filter implies the ability of the filter to have an IIR. This indicates that the system is prone to feedback and instability (Tang et al., 1998; Abdesselam, 2000; Tseng and Pei, 2001).

Consider the IIR filter with the input-output relationship governed by:

\[ y(k) + \sum_{i=1}^{M} b_i y(k-i) = \sum_{i=0}^{L} a_i x(k-i) \]  \hspace{1cm} (3)

where \( x(k) \) and \( y(k) \) are the filter’s input and output, respectively, \( M (\geq L) \) is the filter order. The transfer function of this IIR filter can be determined as:

\[ H(z) = \frac{A(z)}{B(z)} = \frac{\sum_{i=0}^{L} a_i z^{-i}}{1 + \sum_{i=1}^{M} b_i z^{-i}} \]  \hspace{1cm} (4)

These parameters \( a_0, a_1, a_2, \ldots, a_L, b_1, b_2, \ldots, b_M \) appearing in equation (3) and equation (4) are the filter coefficients. These coefficients determine the characteristics of the IIR digital filters. The digital filters have various steps in their design. In the first step, specifications of the designed filter are to be mentioned. The objective of second step is to calculate the filter coefficients for transfer function \( H(z) \), approximating or satisfying a given frequency response specification. In third step, transfer function will be realised to a suitable filter network or filter structure. Input data and filter coefficients are being quantised, or rounded off due to finite word length limitation of the processor on which filtering process is taking place in fourth step. If analysis does not meet, we have to re-realise (by using other structure), re-calculate (use other technique) and re-specify (changing sampling frequency). This description is then transformed in the last step. The flow chart for the design of digital filter is shown in Figure 2 (Skaf and Stephen, 2008; Cao, 2010).

An important task for the designer is to find values of \( a_i \) and \( b_i \) such that the frequency response of the filter approximates a desired characteristic while preserving the stability of the designed filter.
Hence, the design of IIR digital filter can be considered as an optimisation problem (Kacekenga et al., 1990; Benvenuto and Marchesi, 1992) of cost function $J(w)$ stated as $\min J(w)$ where $w = [a_0, a_1, a_2, \ldots, a_L, b_1, b_2, \ldots, b_M]$ is filter coefficient vector. The aim is to minimise the cost function $J(w)$ by adjusting $w$. The cost function is usually expressed as the time-averaged cost function defined by:

$$J(w) = \frac{1}{N} \sum_{k=1}^{N} (d(k) - y(k))^2$$  \hspace{1cm} (5)

where $d(k)$ and $y(k)$ are the desired and actual responses of the filter, respectively and $N$ is the number of samples used for the calculation of cost function. The fitness value of a solution $i$ in the population is determined by using fitness formula given as:

$$fit(i) = \frac{1}{k + J(w)i}$$  \hspace{1cm} (6)

where $J(w)$ is the cost function value computed for $i$ and $k$ is the number of poles outside the unit circle.

The error surface of digital IIR filters is generally non-linear and multimodal, so global optimisation techniques are required in order to avoid local minima. In designing IIR digital filter, the values of $a_i$ and $b_i$ must be such that the magnitude response of the filter approximates a desired characteristic while preserving the stability of the designed filter. Relatively little work has been published so far on PSOs applied to analogues filters. A number of practical issues are important in analogues filter design. One of them is the choice of component values (Cousseau et al., 2007; Zheng, 2003).
PSO is used to perform the design of IIR digital filter. PSO-based infinite-impulse response (PSOIIR) digital filter is implemented by MATLAB and flow chart is shown in Figure 3. The benefit of such operation is to restore the lost genetic values when the population converges too fast. The filter coefficients belonging to PSO were achieved using following parameters: with acceleration constants $c_1$ and $c_2$ each equal to 2, and the population size of 50 was assumed. As originally developed $w$ often is decreased linearly from 1 to 0.1 in 500 iterations. The PSOIIR produces filter’s coefficients that satisfy both magnitude and phase templates.

Figure 3 Flow chart of PSOIIR design

4 Results and discussion

Experiments are carried out for well known IIR digital filters (see Appendix), which have been used by many authors as a ‘benchmark filter’ for comparison. To compare the performance of proposed method, the results of *fda* and *SA*, GA methods (under same conditions) are also obtained through simulations (Ahn et al., 2011; Guruswamy et al., 2011). The examples were performed on Pentium IV, 2.80 GHz CPU and 1 GB of RAM. Two performance measures, i.e., mean square error (MSE) and mean standard deviation (MSD) are explored to compare the performance of proposed method, GA, SA and mean square error (*fda*)-based approaches. Simulation study is carried out in MATLAB to demonstrate the potentiality of PSO for the design of IIR digital filters. The magnitude and phase response of first example is shown in Figure 4 and Figure 5.
In Figure 6, we have depicted pole-zero behaviour of LP filter. It can be seen that the poles-zeros location of designed filter falls with in unit circle. This shows that the designed filter is also stable using PSOIIR.
The MSE and MSD are indicated in Table 1. It shows that the performance of PSOIIR gives optimal values of MSE and MSD. Figures 7 and 8 provide the qualitative measure of the performance using four algorithms.

### Table 1 Results of MSE and MSD of LP filter

<table>
<thead>
<tr>
<th>Name of method</th>
<th>Order of filter</th>
<th>MSE</th>
<th>MSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>fda tool</td>
<td>2</td>
<td>0.3296</td>
<td>0.3140</td>
</tr>
<tr>
<td>SA-based</td>
<td>2</td>
<td>0.3284</td>
<td>0.3129</td>
</tr>
<tr>
<td>GA-based</td>
<td>2</td>
<td>0.3275</td>
<td>0.3105</td>
</tr>
<tr>
<td>PSOIIR-based</td>
<td>2</td>
<td>0.3195</td>
<td>0.3101</td>
</tr>
</tbody>
</table>

### Figure 7 Comparison of MSE using various techniques (see online version for colours)
An application of swarm intelligence for the design of IIR digital filters

Figure 8  Comparison of MSD using various techniques (see online version for colours)

![Comparison of MSD](image)

Table 2 provides a comparison of coefficients with the *fda* tool, SA and GA.

<table>
<thead>
<tr>
<th>Coeff.</th>
<th><em>fda</em> tool</th>
<th>SA-based</th>
<th>GA-based</th>
<th>PSO-IIR-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0$</td>
<td>0.3752</td>
<td>0.3749</td>
<td>0.3739</td>
<td>0.3659</td>
</tr>
<tr>
<td>$a_1$</td>
<td>0.7504</td>
<td>0.7489</td>
<td>0.7459</td>
<td>0.7313</td>
</tr>
<tr>
<td>$a_2$</td>
<td>0.3752</td>
<td>0.3749</td>
<td>0.3739</td>
<td>0.3659</td>
</tr>
<tr>
<td>$b_1$</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>$b_2$</td>
<td>0.3120</td>
<td>0.3109</td>
<td>0.3116</td>
<td>0.2916</td>
</tr>
<tr>
<td>$b_3$</td>
<td>0.1888</td>
<td>0.1790</td>
<td>0.1789</td>
<td>0.1723</td>
</tr>
</tbody>
</table>

Figure 9 and Figure 10 illustrate the magnitude and phase response of second example.

Figure 9  Magnitude response of HP filter using PSO (see online version for colours)
Figure 10  Phase response of HP filter using PSO (see online version for colours)

Figure 11 summarised pole-zero position of high pass (HP) filter using PSO. It is clearly seen that the poles are inside the circle and zeros placement exist inside or on the verge of unit circle gives us stable HP filter.

Figure 11  Pole-zero position of HP filter using PSO (see online version for colours)

Table 3 evaluates the values of MSE and MSD for second example under same conditions. Moreover, the MSE and MSD of second filter are shown in Figure 12 and
Figure 13, respectively. Table 4 gives the coefficients of HP filter under same specifications of PSOIIR, with the order of designed filter. The designed example is compared with the other methods under same conditions (fda tool, SA and GA approaches). On comparison, it is found that the PSOIIR algorithm gives optimal coefficients for HP filter and least MSE and MSD.

Table 3 Results of MSE of HP filter

<table>
<thead>
<tr>
<th>Name of method</th>
<th>Order of filter</th>
<th>MSE</th>
<th>MSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>fda tool</td>
<td>4</td>
<td>3.0589</td>
<td>1.4682</td>
</tr>
<tr>
<td>SA-based</td>
<td>4</td>
<td>2.9967</td>
<td>1.4534</td>
</tr>
<tr>
<td>GA-based</td>
<td>4</td>
<td>2.9913</td>
<td>1.4529</td>
</tr>
<tr>
<td>PSOIIR-based</td>
<td>4</td>
<td>2.9817</td>
<td>1.4503</td>
</tr>
</tbody>
</table>

Figure 12 Comparison of MSE for HP using various techniques (see online version for colours)

Figure 13 Comparison of MSD for HP filter using various techniques (see online version for colours)
Table 4  Coefficients of HP filter designed with fourth order

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>fda tool</th>
<th>SA-based</th>
<th>GA-based</th>
<th>PSOIR-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0$</td>
<td>0.0004218886</td>
<td>0.0004218886</td>
<td>0.0004218886</td>
<td>0.0004218886</td>
</tr>
<tr>
<td>$a_1$</td>
<td>-0.0016875540</td>
<td>-0.001610240</td>
<td>-0.0016152230</td>
<td>-0.0016468845</td>
</tr>
<tr>
<td>$a_2$</td>
<td>0.0025313316</td>
<td>0.0022993300</td>
<td>0.0023143520</td>
<td>0.0024093298</td>
</tr>
<tr>
<td>$a_3$</td>
<td>-0.0016875540</td>
<td>-0.001455500</td>
<td>-0.0014705890</td>
<td>-0.0015655560</td>
</tr>
<tr>
<td>$a_4$</td>
<td>0.0004218886</td>
<td>0.000344561</td>
<td>0.0003495715</td>
<td>0.00038122653</td>
</tr>
<tr>
<td>$b_1$</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
</tr>
<tr>
<td>$b_2$</td>
<td>3.177811000</td>
<td>3.15992000</td>
<td>3.1654356000</td>
<td>3.1604679000</td>
</tr>
<tr>
<td>$b_3$</td>
<td>3.854654400</td>
<td>3.80985000</td>
<td>3.8067268000</td>
<td>3.7993156200</td>
</tr>
<tr>
<td>$b_4$</td>
<td>2.107306220</td>
<td>2.06993900</td>
<td>2.0556433000</td>
<td>2.0534704500</td>
</tr>
<tr>
<td>$b_5$</td>
<td>0.437002640</td>
<td>0.42670300</td>
<td>0.4196924000</td>
<td>0.4204985600</td>
</tr>
</tbody>
</table>

5 Conclusions

In this paper, we developed a SI method for the design of IIR digital filters. It is concluded that PSO is a global optimisation technique for IIR digital filters, and the benefits of SI for designing digital filter have been studied. The simulation results show that PSOIR has better, or at least equivalent, global search ability and convergence speed than others. The examples demonstrate the versatility of the proposed approach. The performance of proposed method has been compared with fda tool, SA and GA technique. Better MSE and MSD are calculated from the proposed method. The position of pole-zero of the designed filters using optimal coefficients provides us stable filters. Thus, it is believed that the PSOIR algorithm is capable of quick and high performance. The proposed method can be extended to arbitrary magnitude response specifications and multi-band. The memetic algorithm or the other EAs can be the thrust area for the design of IIR digital filters.

References

An application of swarm intelligence for the design of IIR digital filters


**Appendix**

**Example 1**

*Design a low-pass filter with following specifications: pass/stop band ripples 1 dB/15 dB, and band edges 200 Hz/400 Hz and a sampling frequency of 1,000 Hz.*

**Example 2**

*This example is taken for design of a HP filter with following specifications: pass/stop band ripples 1 dB/75 dB, and band edges 700 Hz/300 Hz with sampling frequency of 1,500 Hz.*