Filter Design Strategy Using Optimization Techniques

Vaibhav Srivastava, Ramapati Mishra
Department of Electronics & Communication
IET, Dr. RMLAU, Ayodhya (U.P.)

Abstract: Nowadays we are used to perform many filtering tasks which in the not so distant past were performed almost exclusively by analog filters and are replacing the traditional analog filters in many applications. Along with the advantages, such as, high accuracy and reliability, small physical size and reduced sensitivity to component tolerances or drift, digital implementations helps to achieve certain characteristics not possible with analog designs such as exact linear phase and multirate operation. They are applied to very low frequency signals, such as those occurring in biomedical and seismic applications very efficiently. In addition, the characteristics of digital filters can be changed or adapted by simply changing the content of a finitenite number of registers, thus multiple filtering tasks can be performed by one programmable digital filter without the need to replicate the hardware. With the ever increasing number of applications involving digital filters, the variety of requirements that have to be met by them is increased. Consequently design techniques that are capable of satisfying sophisticated design requirements are becoming an impregnable necessity.

Keywords-Pass band, Stop band ripples, PSO, Velocity formula.

1. Introduction:

A filter is a frequency selective circuit that allows a certain frequency to pass while attenuating the others. Filters could be analog or digital. Analog filters use electronic components such as resistor, capacitor, transistor etc. to perform the filtering operations. These are mostly used in communication for noise reduction, video/audio signal enhancement etc. In contrast, digital filters use digital processors which perform mathematical calculations on the sampled values of the signal in order to perform the filter operation. A computer or a dedicated digital signal processor may be used for implementing digital filters. Filters mostly find their use in communication for noise reduction, audio/video signal enhancement etc.

Any time varying signal C=x(t) sampled at a sampling interval of h has input signals x0, x1, x2, x3,…………, xn in intervals 0, h, 2h, 3h, ……….. , nh. These inputs have corresponding outputs y0, y1, y2, y3, …………, yn depending upon the kind of operation performed. Thus, the order of the filter is determined by the number of the previous input terms used to calculate the current output. The a0, a1, a2 terms appearing in the following equations are called the filter coefficients and determine the operation of the filter. These determine the characteristics of the filter. Various filter parameters which come into picture are the stopband and passband normalized frequencies (ωs, ωp), the passband and stopband ripple (δp) and (δs), the stopband attenuation and the transition width. This has been shown in Fig. 1.

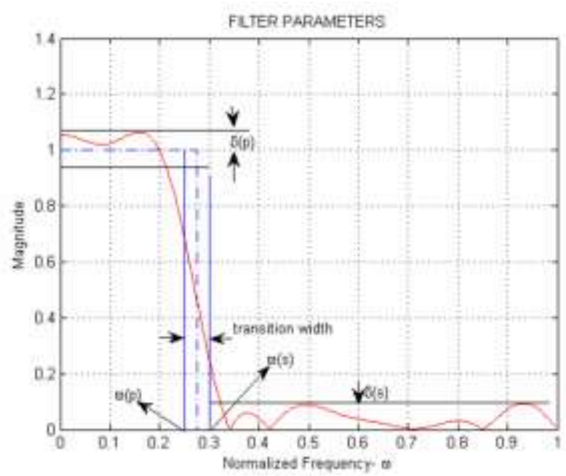

Figure 1. Illustration of filter parameters.

2. Related Work:

Several investigators have attempted to adapt PSO parameters in response to information from the environment. Techniques from evolutionary computation and other methods have been borrowed by particle swarm researchers as well.

Angeline [2] produced one of the first intentionally hybridized particle swarms. In his model, selection was applied to the particle population; "good" particles were reproduced with mutation, and "bad" particles were eliminated. Angeline's results showed that PSO could benefit from this modification.

Miranda and Fonseca (2002) borrowed an idea from evolution strategies. In that paradigm, points are perturbed by the addition of random values distributed around a mean of zero; the variance of the distribution is evolved along with function parameters. Those researchers used Gaussian random values to perturb χ, φ1, and φ2, as well as the position of the neighborhood best—but not the individual best—using selection to adapt the variance. The evolutionary self-adapting particle swarm optimization method, a hybrid of PSO and evolutionary methods, has shown excellent performance in comparison to some standard particle swarm methods. Miranda has used it for the manufacture of optical filters as well as in the optimization of power systems (Miranda and Fonseca 2002).

Loøvbjerg et al. (2001) use "breeding", borrowed from genetic algorithms, in a recent particle swarm study. Some selected particles were paired at random, and both positions and velocities were calculated from weighted arithmetic averages of the selected particles' parameters. Those researchers also divided the particle swarm into subpopulations in order to increase diversity, with some probability that individuals would breed within their own subpopulation or with a member of another. Results were encouraging, though the model as reported was not clearly superior to standard PSO or GA.

Wei et al. (2002) took a different tack, embedding velocity information in an evolutionary algorithm. They replaced Cauchy mutation with a version of PSO velocity in a fast evolutionary programming (FEP) algorithm, to give the FEP population direction. Their published results indicate that the approach is very successful on a range of functions; the new algorithm found global optima in tens of iterations, compared to thousands for the FEP versions tested.

Robinson et al. (2002), trying to optimize a profiled corrugated horn antenna, noted that a GA improved faster early in the run, and PSO improved later. As a consequence of this observation, they hybridized the two algorithms by switching from one to the other after several hundred iterations. They found the best horn by going from PSO to GA (PSO-GA) and noted that the particle swarm by itself outperformed both the GA by itself and the GA-PSO hybrid, though the PSO-GA hybrid performed best of all. It appears from their result that PSO more effectively explores the search space for the best region, while GA is effective at finding the best point once the population has converged on a single region; this is consistent with other findings.

Krink and Loøvbjerg (2002) similarly alternated among several methods, but they allowed individuals in a population to choose whether to belong to a population of a genetic algorithm, a particle swarm, or to become solitary hill-climbers. In their self-adaptive search method, an individual changed its stage after 50 iterations with no improvement. The population was initialized as PSO particles; the "LifeCycle" algorithm outperformed all three of the methods that comprised it. Krink and Loøvbjerg's graphs show interesting changes in the proportion of individuals in each state for various problems.

A hybrid between a PSO and a hill-climber was proposed by Poli and Stephens (2004) who considered swarms of particles sliding on a fitness landscape rather than flying over it. The method uses particles without memory and requires no book-keeping of personal best. Instead it uses the physics of masses and forces to guide the exploration of fitness landscapes. Forces include: gravity, springs, and friction. Gravity provides the ability to seek minima. Springs provide exploration. Friction slows down the search and focuses it.

Clerc's recent experiments (Clerc 2006b) have shown that adaptation of the constriction factor, population size, and number of neighbors can produce improved results. His studies found that best performance were obtained when all three of these factors are adapted during the course of the run. Clerc used three rules: (a) Suicide and generation: a particle kills itself when it is the worst in its neighborhood and generates a new copy of itself when it is the best; (b)Modifying the coefficient: good local improvement caused an increase in the constriction coefficient, while poor improvement caused its decrease; (c) Change in neighborhood: the locally best particle could reduce the number of its neighbors, while poorly performing particles could increase theirs. Adaptive changes were not made on every iteration, but only occasionally.

3. Methodology:

Particle Swarm Optimization (PSO) is an evolutionary computation technique, which is inspired by flocks of birds and shoals of fish (Kennedy and Eberhart, 1995). In PSO, a number of simple entities (the particles) are placed in the space of some problem and each evaluates its fitness as its current location. Each particle determines its movement through the space by considering the particle which had the best fitness and the history of its own, then it moves with a velocity. Finally, the swarm is likely to move close to the best location. The velocity and position of each particle is adjusted by the following formulas:

$$V_{id}=w\times V_{id}+c_1\times rand()\times(P_{id}-X_{id}) +c_2\times Rand()\times(P_{gd}-X_{id}) \quad (1)$$

$$X_{id}= X_{id}+V_{id} \quad (2)$$

where c1 and c2 are termed the cognitive and social learning rates. These two parameters control the relative importance of the memory of the particle itself to the memory of the neighborhood. The variable rand() and Rand() are two random functions that is uniformly distributed in the range [0,1]. Xi = (Xi1, Xi2, … , XiD) represents the ith particle. Pi = (Pi1, Pi2, …, PiD) represents the best previous position of the ith particle. The symbol g represents the index of the best particle among all the particles. Vi = (Vi1, Vi2, … , ViD) represents the velocity of the ith particle. Variable is the inertia weight. The general process of PSO is as follows.

Calculate fitness of particle

Update pbest if the current fitness is better than pbest

Determine nbest for each particle: choose the particle with the best fitness value of all the neighbors as the nbest

For each particle Calculate particle velocity according to (1)

Update particle position according to (2)

While maximum iterations or minimum criteria is not attained

Since the introduction of the PSO algorithm, several improvements have been suggested. In 1998, inertia weight was first proposed by Shi and Eberhart [11]. The function of inertia weight is to balance global exploration and local exploitation. In the following year, Clerc proposed the constriction factors to ensure the convergence of PSO [12]. Eberhart and Shi compared inertia weight with constriction factors and found that the constriction factors was better convergence than inertia weight [12].

In DAPSO2, if there were many particles far away from the global best position, then the velocities should be given a larger value. If there were many particles near from the global best position, then the velocities should be given a smaller value. DAPSO1 only adjusts the velocity of the certain particle, but in DAPSO2, the velocities of all particles are adjusted together.

The general flow of DAPSOs and the flowchart of DAPSO are shown as follows.

Step 1. Initialization of a population of particles with random positions and velocities

Step 2. Evaluation of particles.

Step 3. Calculate the distance from each particle to the global best position and save the farthest distance in the memory.
Step 4. Adjust particle's velocity according to its distance from itself to the global best position.
Step 5. Update particle's position by the adjusted velocity.
Step 6. Repeat Step.2~Step.5 until termination criteria are met.

The frequency response of the FIR digital filter can be calculated as,

$$H(e^{jwk})=\sum_{n=0}^{N} h(n)e-jwkn \qquad (3)$$

Where $\omega_k=\frac{2\pi k}{N}$; $H(e^{jwk})$ is the Fourier transform complex vector. This is the FIR filter's frequency response. The frequency is sampled in $[0, \pi]$ with N points. Different kinds of error fitness functions have been used in different literatures. An error function given by (3) is the approximate error used in PM algorithm for filter design [13].

$$E(w)=G(w)[H_d(e^{jw})- H_i(e^{jw})] \qquad (4)$$

where $Hde^{(j\omega)}$ is the frequency response of the designed approximate filter; $Hie^{(j\omega)}$ is the frequency response of the ideal filter; $G(\omega)$ is the weighting function used to provide different weights for the approximate errors in different frequency bands. For ideal LP filter, $Hie^{(j\omega)}$ is given as,

$$H_i(e^{jw}) = \begin{cases} 1 & \text{for } 0\leq \omega \leq \omega_c ; \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

where ω_C is the cut-off frequency. The major drawback of PM algorithm is that the ratio of δ_p/ δ_s is fixed. To improve the flexibility in the error function to be minimized, so that the desired level of δ_P and δ_S may be specified, the error function given in (5) has been considered as fitness function in many literatures [14] [15]. The error fitness to be minimized using the evolutionary algorithms, is defined as:

$$J_3=\max_{\omega\leq\omega_P}(|E(\omega)|-\delta_P)+ \max_{\omega\geq\omega_S}(|E(\omega)|-\delta_S) \qquad (6)$$

where δ_P and δ_S are the ripples in the pass band and stop band, respectively, and p ω and s ω are pass band and stop band normalized cut-off frequencies, respectively. Since the coefficients of the linear phase positive symmetric even order filter are matched, the dimension of the problem is halved. This greatly reduces the computational burdens of the algorithms. In this paper, a novel error fitness function given by (5) has been adopted in order to achieve higher stop band attenuation and to have better control on the transition width. By using (6), it is found that the proposed filter deign approach results in considerable improvement over the PM and other optimization techniques.

$$J_4=\sum abs[abs(|H_d(w)|-1)-\delta p]+\sum[abs(|H_d(w)|-\delta s] \qquad (7)$$

For the first term of (7), ω ∈pass band including a portion of the transition band and for the second term of (3.68), ω ∈stop band including the rest portion of the transition band. The portions of the transition band chosen depend on pass band edge and stop band edge frequencies.

4. Result and Discussion:

In this section we have presented the simulations results performed in MATLAB for the design of FIR LP filter. The filter order (N) is taken as 20, which results in the number of coefficients as 21. The sampling frequency is taken to be fs = 1Hz. The number of frequency samples is 128. Each algorithm is run for several times to obtain its best results.

PSOP1 is classical method named as P1.
PSOP2 is modified diversity control, dynamic& adjustable pso algorithm.
These algoritm are applied to get optimized solution of filter design. We have used four test function named as J1 J2 J3 J4. The parameters of the filter to be designed using the NPSO are: pass band ripple $\delta_P = 0.1$, stop band ripple $\delta_s = 0.01$. For the LP filter, pass band (normalized) edge frequency ωp= 0.45; stop band (normalized) edge frequency ωs = 0.55; transitiowidth=0.1. The filter has order 20.
The parameters that are selected for pso algoritm are given
Populationsize-75,
iterationcycle =500,C_1=2.05,C_2=2.05,
Vimin=0.01,Vimax=1.0,
Wmax=1.0,Wmix=0.4,
Z=100,
We have taken both PSO algorithm PSOP1 & PSOP2 one by one for above mentioned filter parameters. The response are represented in term of four kind of plots. These plots are Convergence plot of PSO, Impulse response of filter, normalized frequency response & magnitude response of filter.

(a) Convergence plot-This plot is drawn in term of error value(cost) versus no. of iterations. for a good optimization method, this plot monotonically decreases & after a particular iteration, it become saturated. Lower the no. of iteration indicates, higher the speed of convergence of optimization technique. We have taken 500 iteration in above PSO algorithm.

(b) Impulse Response-In this plot we are presented filter coefficient with respect to their weight in filter design differential equation.Since we have used a filter order of 20,there are 21 samples of filter coefficient.This plot is in between h(n) versus n.

(c) Normalized frequency Response-This plot is the 20log of the fourier transform of h(n).It indicates normalized frequency.It is in between 20log|H(jω)|db versus f Hz.In this plot we have also shown pass band & stop band ripples additionally.

(d) Magnitude Response-This plot represents pass band & stop band magnitude & ripples.It is in between H(jω) & frequency.
In both plot (c) & (d) X axis is frequency,since we have taken f_s as 1,that is why the frequency on X- axis from 0 to 1.

4.1. Case1-

The PSO approach is PSOP1 and the test functions are J1,J2,J3,J4 taken one by one in this case & their results are discussed below.

(A) PSO=PSOP1,Test function=J1

The figure 1 shown below represents the result better obtain after running simuiation for several times but here shown only best 1.

Figure 2(a)-It is convergence plot .In this figure we can see that error or cost function. Y-axis represents error in desired response. This error reduces as the no. of iteration increases. Initially the error is 3.5.After 400 iteration it is almost converges to zero. In this case we found error in between response obtain by PM algorithm & our desired response. This case is only to check the performance of our PSO technique. In fig 2(b),there are impulse response of both h_{pm} & h_{p1} ,where h_{pm} is desired response obtained on applying PSOP1& $h_{p1}$is optimized response obtained by PSOP1.However there are two plots of $h_{pm}[n]$ & $h_{p1}[n]$ but since our PSO has completely superimposed on desired response, so we are getting only single plot.

This has occurred because in J1 test function we did not take any constraints on pass band & stop band. This indicates that PSO algorithm is capable of giving similar results as given by PM algorithm. Similar results are also found for figure 2(c) & 2(d).

We can see that Gain is high near about 0db or 1 for frequency less than 0.45 that is our passband in fig 2(c) & fig 2(d).

Stop band is obtained after 0.55Hz in both figure. The middle plot of figure 2(c) represents stop band ripples. These stop band ripples range is 0.05.We want to get the stop band in range of 0.01.

Since the PM algorithm cannot minimize the result but in further cases we will minimize the result using constraints over the stop band for getting the optimized solution closer to ideal filter response.

Table 1 shows value of filter coefficients obtain by PM algorithm. Since we have taken order of 20, due to this h(1) to h(10) is replicated as h(12) to h(21).

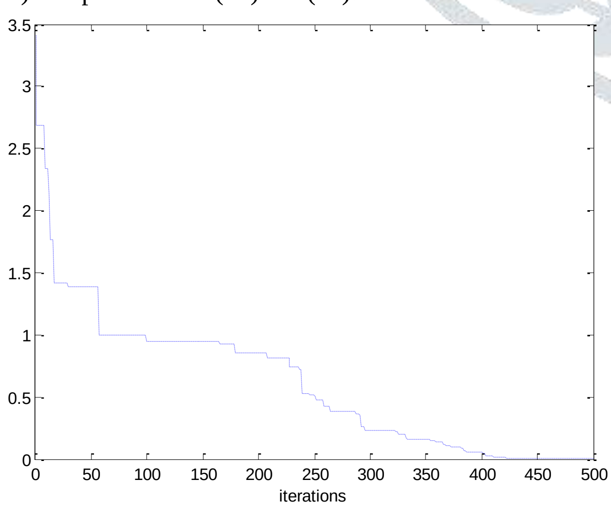

Fig1(a): Converge plot for PSOP1 and test function J1

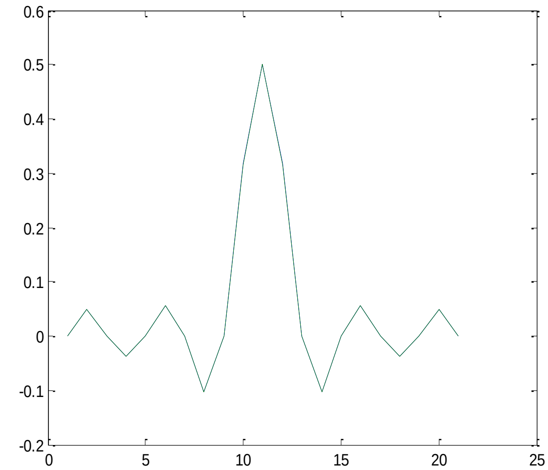

Fig 1(b): Impulse response for PSOP1 and test function J1

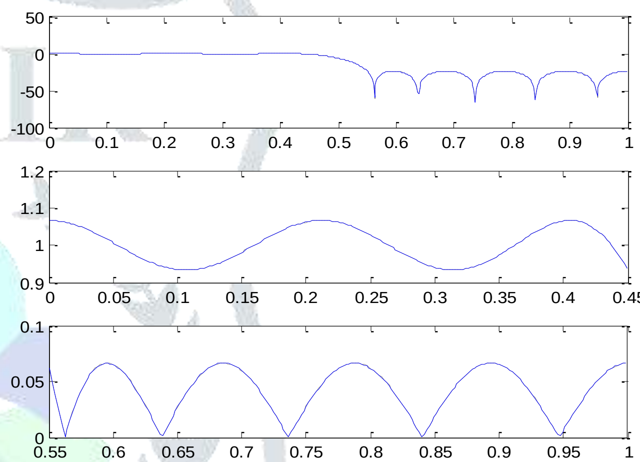

Fig 1(c): Normalized frequency response(dB)for PSOP1 and test Function J1, (above) passband, (below) stopband, Black-PM, Blue- PSOP1

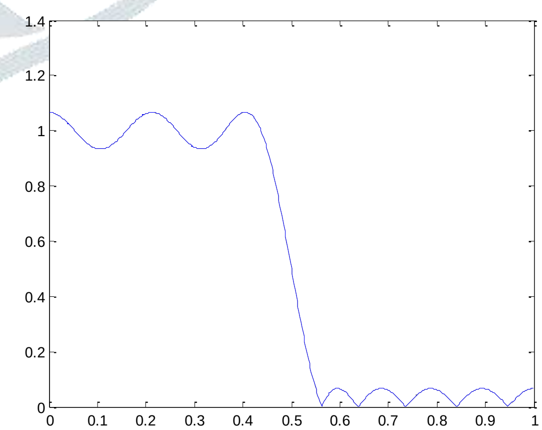

Fig 1(d): Magnitude response for PSOP1 and test function J1, Black-PM, Blue- PSOP1

Table 1: value of filter coefficients

S.NO	h(n)	PSOP1 with test function J1

1	h(1)	0.0000
2	h(2)	0.0481
3	h(3)	-0.0000
4	h(4)	-0.0369
5	h(5)	-0.0000
6	h(6)	0.0573
7	h(7)	-0.0000
8	h(8)	-0.1022
9	h(9)	-0.0000
10	h(10)	0.3170
11	h(11)	0.5001

5. Conclusion:

In this work we have presented an optimal design of linear phase digital low pass finite impulse response (FIR) filter using Dynamic and Adjustable Particle Swarm Optimization(DAPSO) . DAPSO is an improved particle swarm optimization (PSO) that proposes a new definition for the velocity vector and swarm updating and hence the solution quality is improved. The distance from each particle to the global best position is calculated in order to adjust the velocity suitably of each particle. It is revealed that DAPSO has the ability to converge to the best quality near optimal solution and possesses the best convergence characteristics in much less execution times among the algorithms. The simulation results clearly indicate that DAPSO demonstrates the best performance in terms of magnitude response, minimum stop band ripple and maximum stop band attenuation with the narrowest transition width. It can be used as a good optimizer for obtaining the optimal filter coefficients in any practical digital filter design problem of digital signal processing systems.

References:

[1] J. H. McClellan and T. W. Parks, "A unified approach to the design of optimum FIR linear phase digital filters,' IEEE Trans. Circuit Theory, vol. CT-20, pp. 697-701, Nov. 1973.

[2] E. W: Cheney, Introduction to Approximation Theory. New York: McGraw-Hill, 1966, pp. 72-100.

[3] E. Ya. Remez, "General computational methods of Tchebycheff approximation," Kiev, 1957 (Atomic Energy Translation 4491, pp. 1-85).

[4] B. Gold, and K. L. Jordan, "A direct search procedure for designing finite duration impulse response filters," IEEE Trans. Audio Electroacoust., vol. AU-17, pp. 33- 36, Mar. 1969.

[5] J. H. McClellan, "On the design of one-dimensional and two-dimensional FIR digital filters," Ph.D. dissertation, Rice Univ., Houston, Tex., Apr. 1973.

[6] P. J. Angeline. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, Evolutionary Programming VII, pp. 601–610. Springer, 1998.

[7] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks, Vol. 4, pp. 1942–1948. IEEE Press, 1995.

[8] J. S. Vesterstrøm and J. Riget. Particle swarms: Extensions for improved local, multi-modal, and dynamic search in numerical optimization. Master's thesis, EVALife, Dept. of Computer Science, University of Aarhus, Denmark, 2002.

[9] J. Riget and J. S. Vesterstrøm. A diversity-guided particle swarm optimizer – the arPSO. Technical report, EVALife, Dept. of Computer Science, University of Aarhus, Denmark, 2002.

[10] Y. Shi and R.C. Eberhart. A modified particle swarm optimizer. In Proceedings of the IEEE Conference on Evolutionary Computation, pp. 69–73. IEEE Press, 1998

[11] Eberhart, R. C. and Shi, Y., A Modified Particle Swarm Optimization, Proceedings of IEEE International Conference on Evolutionary Computation, 1998, pp.69-73.

[12] Eberhart, R. C. and Shi, Y., Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, Proceedings of the 2000 Congress on Evolutionary Computation, Vol. 1, 2000, pp.84-88.

[13] T.W. Parks, J.H. McClellan, Chebyshev approximation for non recursive digital filters with linear phase, IEEE Trans. Circuits Theory, CT-19, 1972, pp. 189–194.

[14] J.I. Ababneh, M. H. Bataineh, Linear phase FIR filter design using particle swarm optimization and genetic algorithms, Digital Signal Processing, 18, 657–668, 2008.

[15] A. Sarangi, R.K. Mahapatra, S.P. Panigrahi, DEPSO and PSO-QI in digital filter design, Expert Systems with Applications, 2011, vol. 38, No.9, pp.10966-10973.