Decision Threshold Control Method for the Optical Receiver of a WDM PON using PSO

C. Rajeswari
M.E student, Anna University of Technology, Trichy, India

P. Maheswaravenkatesh
Asst Prof/ECE, Anna University of Technology, Trichy, India

ABSTRACT:
An attempt is made to improve the performance of an optical receiver with beat noises by adjusting the threshold level [1] automatically according to the detected average power using MATLAB/SIMULINK model. Observations are made at different noise power levels, number of iterations, values of bit error rate, gain and error count within time elapsed in seconds. Optical communication systems have good speed, accuracy and efficiency but accuracy of high speed communication is unstable due to internal noise. This paper attempts to focus on eliminating one of the major internal noise known as Beat noise. The proposed method is useful for the optical receiver using WDM-PON. When compared to TDM-PON, WDM-PON provides point to point connectivity and pair of wavelengths for a single user. Thus a WDM-PON network is suitable for present and future generation networks.

General Terms:
Swarm optimization, genetic algorithm

Keywords:
Wavelength division multiplexing passive optical networks (WDM PON), Particle swarm optimization (PSO), Beat noise

1. INTRODUCTION
Beat noise is caused due to loosely coupled components. The possibilities of Beat noises in the optical receiver occurs also due to back reflections [5] caused by amplifiers and optical filters. Optical communication network supports good speed, high data rate and secure communication. In optically amplified transmission systems, the components act directly and provide signal photon multiplication by several order of magnitude than the incoming signal intensity.

1.1 Existing system:
In the existing system, the decision threshold level is adjusted automatically according to the detected average optical power. It consists of a conventional receiver, a decision control part and a power monitoring part as shown in fig 1. A power comparator circuit with reference power (VDC) will be compared with received power and the reference power is used to predict the threshold value to further reduce the received power.

1.2 Proposed system
In the proposed system the reference power is unknown and the decision is purely based on particle swarm optimization. PSO analyser is introduced in the decision threshold circuit as shown in the fig 2.

1.3. Particle Swarm optimization
Particle swarm optimization (PSO) [8], is a population-based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. It is
based on the notion that simple local interactions often lead to complex
global behaviors. In a PSO algorithm, each member of the population
(swarm) is called a particle. Each particle has a position and a velocity. The
position of the particle represents a candidate solution to the problem being
addressed. The velocity is used to move the particle from one position to
another position. PSO algorithms start by initializing all the particles in the
swarm. A fitness function is used to quantize the quality of the solution
represented by each particle. The particle having the best fitness value in
the swarm is marked as the global-best particle (gbest). The particle
having best fitness value in each neighborhood is marked as the local-best
particle (lbest). In a single iteration, for each particle, new velocity is
computed based on the positions of the global-best (or local-best) particles.

In each iteration, the particles move through the problem space
by following the current optimum particles. PSO, the potential solutions, called particles, fly through the
problem space by following the current optimum particles.

Fig3. Velocity computation and position updating of a
particle \( P_x \).

The magnitude of this move is a function of the current position of
the particle and the distance between itself and the best particle in the neighborhood (local-best / global-best particle).
Particles continue to move around the problem search space
trying to better themselves in comparison with their own
performance and that of their neighbors. This process continues
until either the whole swarm converges or till the given number of
iterations completes. In PSO, each particle keeps a record of
the best position it has traversed over the problem space, so far.
This position is called the personal-best position (pbest) of
the particle. This way, the particle not only does its own search, it
also learns from the search done by the particle having the best
fitness value in the swarm (or sub-swarm). The classical PSO
equations have the position and velocity represents physical
attributes of the particles.

PSO shares many similarities with evolutionary computation
techniques such as Genetic Algorithms (GA). The system
is initialized with a population of random solutions and searches
for optima by updating generations. However, unlike GA, PSO
has no evolution operators such as cross over and mutation. In
PSO, the potential solutions, called particles, fly through the
problem space by following the current optimum particles.

Each particle keeps track of its coordinates in the problem space
which are associated with the best solution (fitness) it has
achieved so far. (The fitness value is also stored). This
value is called pbest. Another “best” value that is tracked by the
particle swarm optimizer is the

Best value, obtained so far by any particle in the neighbors of
the particle. This location is called lbest. When a particle takes
all the population as its topological neighbors, the best value is a
global best and is called gbest.

The particle swarm optimization concept consists of, at each
time step, changing the velocity of (accelerating) each particle
toward its pbest and lbest locations (local version of PSO).
Acceleration is weighted by a random term, with separate
random numbers being generated for acceleration toward pbest
and lbest locations.

In past several years, PSO has been successfully applied in
many research and applications areas. It is demonstrated that
PSO gets better results in a faster, cheaper way compared with
other methods. Another reason that PSO is attractive is that there
are few parameters to adjust. One version, with slight variations,
works well in a wide variety of applications. Particle swarm
optimization has been used for approaches that can be used
across a wide range of application, as well as for specific
applications focused on a specific requirement. PSO has
successfully been used to solve many industrial and engineering
optimization problems in the diverse areas including biomedical,
communication networks, prediction, neural network, graphics
and visualization, signal processing, electronics, antenna design,
modeling, fuzzy and neuro-fuzzy logic, prediction and
forecasting, scheduling, robotics etc. Some of the advantages of
using PSO algorithm are that it is an easy implementation of a
problem search algorithm Possesses fewer algorithmic
parameters to adjust than other evolutionary algorithms like
genetic algorithms and robust in terms of controlling parameters
is computationally efficient.

2. EXPERIMENTAL SETUP

The unknown receiver power is predicted for the minimum bit
error rate value. The accuracy of the threshold increases by
increasing the iteration level. For each iteration our circuit
generates a random new threshold model and filter the received
optical signal with the predicted threshold at first iteration, the
calculated bit error rate is stored. The generation of optical signal
using simulink, Improper coupling results in back reflections, the best noise increase the actual power of the generated
signal. The voltage control circuit of the exisiting system is
replaced using a particle swarm optimizer.

3. ITERATION VS ERROR COUNT

Error count is the number of symbols mismatched with the
actual pattern, increasing the iteration (time taken for predicting
the threshold level) will reduce the magnitude of error count.
Various readings are tabulated between error count Vs iteration
by keeping the noise power constant.

By this graph as shown in fig 4, we can grasp the relation
between the error count and the time (iteration), thus by
analyzing this graph we can recommend an optimum iteration
cycle for various noise power in real time optical
communication system
4. EXPERIMENTAL RESULTS
Noise power is kept constant, as the number of iteration is increased, the bit error rate is reduced. The following graph denotes the optical back reflections and predicts results for various iteration levels. The observations are tabulated in Table 1. The process can be repeated for different values of noise power for several number of iterations.

Table 1: Comparison between several iterations and noise power levels

<table>
<thead>
<tr>
<th>No of iterations</th>
<th>Noise power in percentage</th>
<th>Bit error rate</th>
<th>Time elapsed in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>10</td>
<td>0.101</td>
<td>5.6</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>0.051</td>
<td>7.8</td>
</tr>
<tr>
<td>150</td>
<td>10</td>
<td>0.034</td>
<td>12.2</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>0.024</td>
<td>14.6</td>
</tr>
<tr>
<td>250</td>
<td>10</td>
<td>0.024</td>
<td>16.1</td>
</tr>
<tr>
<td>300</td>
<td>10</td>
<td>0.017</td>
<td>19.4</td>
</tr>
</tbody>
</table>

From the above results, we conclude that for maximum number of iterations the bit error rate is reduced to a minimum value.

5. ACKNOWLEDGMENT
I would like to thank Mr. Maheshwaravenkatesh, assistant professor, Anna University, Anna University of Technology, Trichy, for his guidance and support for the successful completion of the paper.

6. REFERENCES