High-speed Interconnect Simulation Using Particle Swarm Optimization

Subas Bastola and Chen-yu Hsu

Intel Corporation

subas.bastola@intel.com, chen-yu.hsu@intel.com

***Abstract*—Particle Swarm Optimization (PSO) is proposed as an efficient algorithm for simulation of high speed interconnects used in today's digital applications. First, a generic methodology is proposed for high speed interconnects simulation using PSO and finally comparisons are made between the performance of PSO compared to traditional optimization techniques used in high-speed serial bus simulation.**

***Index Terms*—Particle Swarm Optimization, interconnect simulation, link optimization, response surface**

I. INTRODUCTION

PArticle Swarm Optimization has been proposed as a viable optimization technique for varieties of engineering problems[1]. Recently, PSO has been shown to be a good candidate for optimization in microwave and electromagnetic applications [2]. This algorithm has shown to be more efficient in solving some optimization problem than other evolutionary stochastic techniques like GA (genetic algorithm).

PSO is a population based stochastic algorithm developed by Dr Eberhart and Dr Kennedy in 1995. It is inspired by the social behaviour of a swarm of birds in search of optimum place for food; this algorithm builds group momentum towards the most optimum point in multi-dimentional solution space.

General algorithm for PSO is as follows:

1) Decide the particle population size N. Initialize the particle location and velocity randomly.

$P= (P_{0,} ....P_{i,} ...P_N)$
$V= (V_{0,} ....V_{i,} ...V_N)$

2) Find the system response each particle.

$F(P) = (F(P_0), \ldots F(P_i), \ldots F(P_N))$

3) Keep track of the locations of best fitness for each particle.

$P= (P_0(best)_, ...P_i(best)_, ...P_N(best))$

4) Continuously keep track of the position of the global best fitness.

$P_g(best) = \max_{p \varepsilon P}(F(P))$

5) Modify the particle velocity based on previous best and global best positions:

$V_i(new) = V_i + \varphi_1 (P_i(best) - P_i) + \varphi_2 (P_g(best) - P_i)$

6) Modify particle locations:

$P_i(new) = P_i + V_i(new)$

7) Go back to step 2 if terminating condition is not met.

As the core frequency of the CPUs continue rise each year, IO operation are also trying to keep pace by moving up higher in Gb/s (gigabits per second) regime [3]. Today, most of these high speed IO links are implemented using low voltage high bit rate differential links [4]. It's extremely important to properly model each components of the serial link and check the performance of the link for an acceptable bit error rate (BER) using computer simulations. A typical high speed serial link consists of few building blocks such as package, socket, motherboard traces, vias, connector etc. Each of these building blocks will have some parameters which are electrically significant and need to be included in the link simulations. Some examples of these parameters include impedance and length for transmission lines, silicon process corners for the driver and receivers etc. Unfortunately, these parameters can assume any value within a tolerance range and as such they can vary from system to system. A unique set of values for these parameters constitutes a corner case. One of the major problems for any link simulations has been the large number of possible corner cases, even for a relative small number of variables. For example, a 12 variables serial link with each variable having three distinct values will require 3^12 = 531441 distinct simulation cases if a grid-based simulation technique was used. To circumvent this difficulty, many alternatives techniques have been proposed and used, such as Monte Carlo, Response Surface Method (RSM) and neural networks [5]-[7].

1-4244-0708-7/07/$20.00 ©2007 IEEE

II. NEW HIGH SPEED SIMULATION METHODOLOGY

1) Identify key variables
2) Find the dynamic range of each variable
3) Identify the fitness function
4) Use PSO algorithm to find the corner condition which gives least desirable response (worst case corner condition)

1) The first step in this new methodology is to find the top level blocks that make up the high speed serial link (Fig 1). A typical high speed serial link consists of a driver buffer, driver package, motherboard, connector, backplane, connector, motherboard, receiver package and receiver buffer. (see diagram)

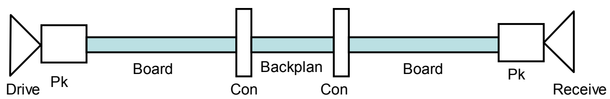

Fig 1. Typical components in a high speed bus

Each of the above components will have multiple variables that affect the performance of the link. For example, a driver can have rise-time/fall-time, output pad capacitance and termination values as variables. Furthermore, especially true for driver and receiver circuits used in today's high speed digital applications; there are other parameters, which are under user's control. For example, pre-emphasis tap settings of a driver can be tuned to optimize the link performance by the user if desired. Irrespective of a variable's controllability by the user, simulation engineer should note down all variables pertaining to each block.
Simulation engineer should then make some engineering judgment or preferably, use some effect screening method (such as RSM) to find out which are the significant variables that need to be swept in simulation and which can be held constant.

2) One of the most challenging parts of any simulation process is to find dynamic ranges for each of the variables that need to be swept in simulation. Some of the variables are governed by specification while others need to come from the component designer. For example, motherboard trace impedance is pretty much guaranteed by PCB fabrication house to be within certain percentage of the nominal value. Other times, it may not be obvious and the simulation engineer will have to make some engineering judgment.
One challenge that arises while using PSO for electrical simulation is the fact that some of the variables can only be swept in discrete steps. For example, even though motherboard impedance is continuous in nature, typically simulation engineers will only have discrete models associated with minimum, typical and maximum impedances. Since fundamental PSO algorithm treats each variable as continuous in range, it expects the simulation engineer to supply models for impedance that could lie anywhere in its dynamic range. To solve this problem, we propose the following scheme:

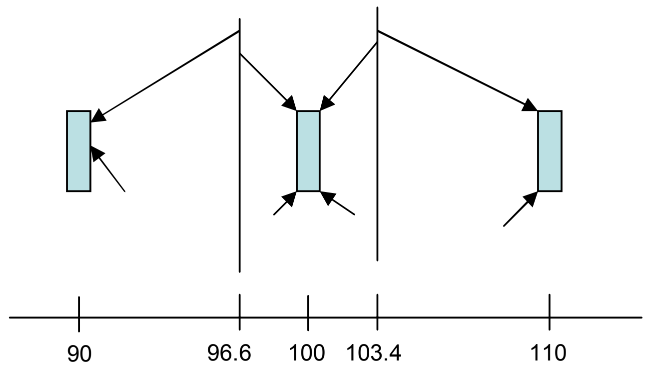

Fig 2. User has only 90, 100 and 110 ohms transmission line models so PSO will snap to one of these models based on the value of new impedance. Arrows indicate snapping action.

Let's assume that the nominal mother board differential impedance was 100 ohm and since the manufacturing house spec their tolerance to be +/- 10%, dynamic range of this variable is from 90 ohm to 110 ohm as shown in Fig 2. We then divide this dynamic range into three equal segments (90-96.6), (96.6-103.4) and (103.4-110). These three segments now dictate the PSO which of the three discrete transmission line models that PSO should pick. For example, if one of the particle in PSO wants to assume the value of 93 ohms for motherboard impedance, PSO algorithm checks which of the three regions the new value lies and since 93 ohms lies in 90-96.6 region, PSO will be forced to choose 90 ohm (minimum impedance corner) as the variable value and use 90 ohm model accordingly.

3) Since the aim of our simulation is to determine the worst possible response (for example, eye height or eye width), it's difficult to find an objective fit function. However, there are few methods in which we can specify the terminating conditions. One simple way is to specify a threshold value for the system response and as soon as the system response goes below the threshold, PSO will stop further processing. Another better approach to this problem is to keep track of the responses that PSO found for last N epochs and calculate how much change has happened since last N epochs. If the PSO best value stays same for last N epochs, then we can assume that PSO has found its best response and terminate the simulation (Fig 3). This second method was used in our simulations to stop PSO.

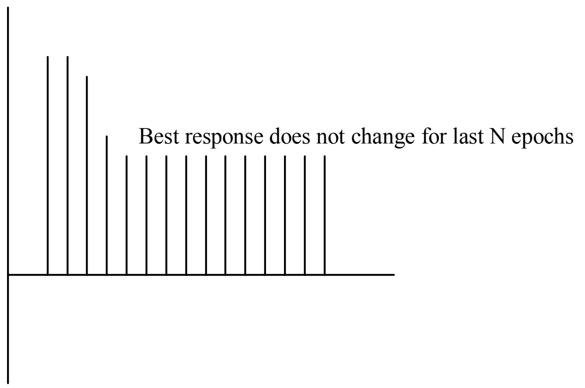

Fig 3. PSO will terminate if the response from last N epochs do not change

4) Using PSO algorithm simulation can now proceed until the terminating criterion is met.

Some of the parameters that were used in our simulations are:

Inertia: It was gradually swept from 0.9 to 0.4 as the epoch progressed. This parameter adjusts the behavior of the swarm by controlling how fast a particle should change its velocity and direction to match the global best response location.

Number of particles: Eighteen particles were used in our simulation exercise.

Number of epochs: Epochs refer to the each instance when all particles report their findings and a new global best location is calculated. After each epoch, particles, change their direction and velocity to be reflect the location of new global best. As a starting point, 100 epochs were allowed in the simulation. However, if the response from last 10 epochs did not change, we did not wait for the completions of 100 epochs. Instead PSO was simply terminated.

PSO can be implemented with different boundaries for the variables: reflecting boundary, absorbing boundary or invisible boundary (Fig 4). This research was based on the reflecting boundary for the variables. In reflecting boundary scheme, if a variable's new position (calculated from existing position plus the new correction) goes outside the dynamic range of the variable, it will simply reflect from the boundary it is trying to cross. This will make sure that the variable stays inside its dynamic range.

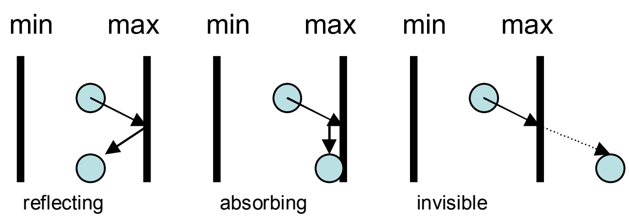

Fig 4. Different boundaries for variables values

III. RESULTS AND COMPARISON

Comparison was made between traditional RSM based worst case prediction and PSO based worst case prediction. A typical high speed channel with two connector three board topology was chosen. Total of 13 variables were picked and simulated using RSM approach. Same variables were swept in PSO approach as well. PSO Tool, written in Visual Basic was used for automating all the steps of PSO algorithm (Fig 5). This tool automatically launches SPICE simulations, waits for all the cases (particles) to complete their simulations, post-processes the waveform to extract eye height and eye width. It then calculates the global best position and finds new positions and velocities for all the variables. This loop continues until the terminating condition is met. Figure 5 shows how the global best converges to a stable value very quickly. For RSM, SPICE simulation was done for different corner cases as dictated by RSM and post-processing was done on the resulting waveforms to obtain the response, eye width in our case.
A three board two connector CSI link topology was used for the simulation. The link operated at 6.4 GB/s speed.

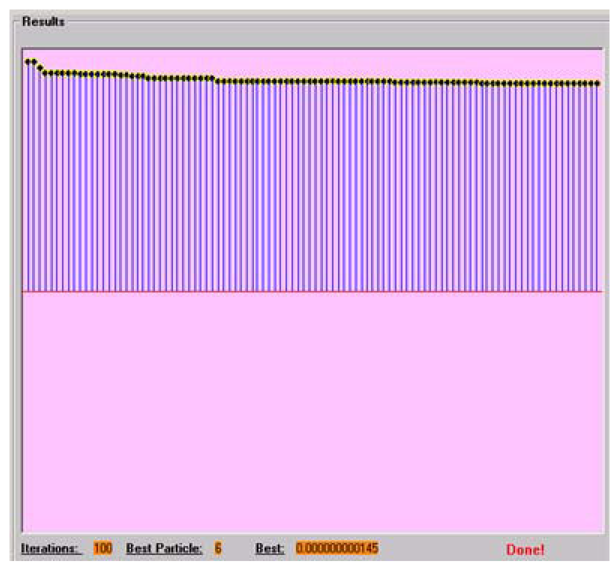

Fig 5. Screen shot of the PSO Tool which automated all steps of PSO algorithm.

Result I:

TABLE 1. WORST CASE RESPONSE FOUND FROM RSM AND PSO. ALSO SHOWS THE VALUE OF VARIABLES IN THE WORST CASE.

Variable	Min	Max	RSM	PSO
Driver IO cap(ps)	0.6	1	1	0.98
RX IO cap(ps)	0.4	0.7	0.7	0.67
TX termination(ohm)	88	112	112	110.9
RX termination(ohm)	88	112	106	109
Pkg length(inch)	0.3	1.25	1.25	0.9
MB1 length(inch)	4.2	6.2	6.2	4.3
MB2 length(inch)	4.8	6.8	6.8	5.5
Backplane length(inch)	4.9	6.9	6.9	5.76
Backplane Z(ohm)	82.7	100.8	82.7	82.7
MB1 Z(ohm)	81.3	98.9	98.9	98.9
MB2 Z(ohm)	82.6	100.4	100.4	100.4
Pkg1 Z(ohm)	80	101	101	101
Pkg2 Z(ohm)	80	101	101	101
Response				
Eye Width			150ps	145ps

As shown in Table 1, PSO predicted worse case eye width of 145ps whereas RSM techniques predicted worse case of 150ps. Also PSO not only predicted worse response than traditional RSM techniques but it also predicted different corners than RSM. While RSM predicted the extremes of the lengths as the worse case conditions, PSO found worse case somewhere in between the extremes of the length. This means that PSO is capable of taking into account resonances in the system. Also PSO output is based on actual simulation and not based on prediction modelling. So the worse case predicted by PSO is real whereas the prediction from RSM may not be accurate, all depending on the how well the RSM model fit is.

Result II-a

TABLE 2. WORST CASE RESPONSE FOUND FROM RSM AND PSO. ALSO SHOWS THE VALUE OF VARIABLES IN THE WORST CASE.

Variable	Min	Max	RSM	PSO (global)
Driver IO cap(ps)	0.6	1	1	0.99
RX IO cap(ps)	0.4	0.7	0.7	0.69
TX termination(ohm)	88	112	112	88.6
RX termination(ohm)	88	112	112	110.2
Pkg length(inch)	0.3	1.25	1.25	0.67
MB1 length(inch)	3.2	5.2	4.2	5.19
MB2 length(inch)	4.8	6.8	6.8	5.9
Backplane length(inch)	4.9	6.9	6.9	6.4
Backplane Z(ohm)	82.7	100.8	100.8	82.7
MB1 Z(ohm)	81.3	98.9	81.3	98.9
MB2 Z(ohm)	82.6	100.4	100.4	100.4
Pkg1 Z(ohm)	80	101	101	101
Pkg2 Z(ohm)	80	101	101	101
Response				
Eye Width			84ps	79ps

As shown in tables above, worst case eye width predicted by PSO is really worse than that predicted by RSM. One of the limitations of RSM technique is that is tries to fit a mathematical model based only on the minimum, typical and maximum values of a variable and as such any severe resonance conditions occurring at in-between values will be missed. PSO does not suffer from this problem.

Result II-b

We completed another set of simulation where the worst case parameter value predicted by RSM in Table 2 was used for all the parameters except the lengths. PSO was used only to sweep length parameter as opposed to sweeping all parameters, as shown in Results II-b.

TABLE 3. WORST CASE RESPONSE FROM PSO WHEN ONLY LENGTHS WERE SWEPT (OTHER VARIABLE VALUES WERE FIXED BASED ON RSM RESPONSE FROM TABLE 2).

Variable	Min	Max	RSM	PSO (RSM WC)
Driver IO cap(ps)	0.6	1	1	1
RX IO cap(ps)	0.4	0.7	0.7	0.7
TX termination(ohm)	88	112	112	112
RX termination(ohm)	88	112	112	112
Pkg length(inch)	0.3	1.25	1.25	1.25
MB1 length(inch)	3.2	5.2	4.2	3.4
MB2 length(inch)	4.8	6.8	6.8	6.1
Backplane length(inch)	4.9	6.9	6.9	5.0
Backplane Z(ohm)	82.7	100.8	100.8	100.8
MB1 Z(ohm)	81.3	98.9	81.3	81.3
MB2 Z(ohm)	82.6	100.4	100.4	100.4
Pkg1 Z(ohm)	80	101	101	101
Pkg2 Z(ohm)	80	101	101	101
Response				
Eye Width			84ps	75ps

When we fixed all the variables (except lengths) to RSM worst corner and only swept the PCB routing lengths in PSO, then we observed that PSO's final eye width number was smaller than global PSO worst case number. This shows that PSO can be an effective tool in capturing worst case corner in the presence of severe resonance arising at different lengths of motherboard traces.

RSM/PSO hybrid:

Since PSO is new in the area of signal integrity simulation for high speed busses, hybrid approach utilizing both PSO and traditional optimizations techniques seems a very reasonable approach. This approach uses traditional optimization techniques such as RSM to find the worst corners. Then it fixes all parameters that are less likely to contribute to any resonance conditions and run PSO for only those variables that might cause resonance. In our case, when we used RSM method to find the worst corner and use PSO to do length sweeps only, we found much worse eye width number than predicted by RSM alone.

Global versus local PSO:

As shown in the example above, PSO can be allowed to change all variables to find the worst case (global PSO) or we can fix most of the variables to some known settings and only allow PSO to vary few parameters (local PSO). However, having a large number of variables may take a longer time for PSO to converge successfully. Local PSO are very useful when we want to capture resonance conditions in a channel. For example, traditional optimization techniques usually do a good job in predicting the worst case corners for most of the parameters except transmission line lengths. Since traditional optimization techniques only take minimum, typical and maximum lengths into consideration when doing the curve fitting, any very high order resonance conditions happening in between these lengths can be missed. With PSO, one can fix all the parameters and only sweep the transmission line lengths and still capture the resonance conditions.

IV. Conclusion

PSO was used in determining worst case channel response and associated variable corner conditions for a very high speed serial link. Compared to traditional RSM based approach, PSO was able to predict a worse response at an entirely different variable corner conditions. Also PSO was able to predict worse resonant conditions from length sweep whereas traditional RSM approach missed those severe resonance conditions.

References

[1] J. Kennedy and R. Eberharrt, "Particle Swarm Optimization," Proceeding of IEEE International Conference on Neural Network, Perth, Australia , pp1942-1948, 1995

[2] J. Robinson and Yahya Rahmat-Samii, "Particle Swarm Optimization in Electromagnetics," IEEE Transactions on Antennas and Propagation, vol 52, No 2, February 2004

[3] Stephen Hall, Tao Liang, Howard Heck and David Shykind, "Modelling Requirements for Transmission Lines in Multi-Gigabit Systems," IEEE 13th Topical Meeting on Electrical Performance of Electronic Packaging, pp 67-70, 2004

[4] Igor S. Stievano, Ivan A. Maio, Flavio G. Canavero and Claudio Siviero, "Parametric Macromodels of Differential Drivers and Receivers," IEEE Transactions on Advanced Packaging, Vol. 28, No.2, May 2005

[5] Tawfik Arabi and Mike Tripp, "A Numerical Frequency Domain Technique for the Optimization of High Speed Multi-Conductor Transmission Line Circuits Based on the Worst Case Design Approach," Electrical Performance of Electronic Packaging, pp 7-9, 1995

[6] Adam Norman, David Shykind, Maynard Falconer and Kim Ruffer, "Application of Design of Experiments (DOE) Methods to High-speed Interconnect Validation," Electrical Performance of Electronic Packaging, pp 15-18, 2003

[7] Wendemagegnehu (Wendem) T. Beyene, "Application of Artificial Neural Networks to Statistical Analysis and Nonlinear Modeling of High-Speed Interconnect Systems," IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, Vol. 26, No. 1, January 2007