APPLICATION OF ADVANCED SEARCH-METHODS FOR AUTOMOTIVE DATA-BUS SYSTEM SIGNAL INTEGRITY OPTIMIZATION

Harald Günther[1], Stephan Frei[1], Thomas Wenzel[2], Wolfgang Mickisch[2]

[1]Technische Universität Dortmund, Arbeitsgebiet Bordsysteme
Friedrich-Wöhler-Weg 4, 44227 Dortmund, Germany
[2]TÜV Nord IFM, Elektronik & IT
Adlerstr. 7, 45307 Essen, Germany

harald.guenther@tu-dortmund.de (Harald Günther)

Abstract

Automotive bus systems are used to connect control units, intelligent sensors and actors in vehicles. From economical point of view extended cable networks are desirable. Systems are often operated close to their specification limits. To provide safety critical applications under these circumstances, the functionality of a bus system must be ensured with sophisticated methods.
An approach to detect problematic behaviour and its causes is the computational investigation of the transmitted signal quality (signal integrity) of the physical layer of the bus system. Models for any component of a bus system have to be developed. Accuracy of models and methods has to be ensured by measurements with realistic systems.
However, many system parameters are not fixed, e.g. tolerances of devices, variations in the topology, external electromagnetic influences etc., so that there is a nearly unmanageable number of combinations. Validation of safe function becomes difficult. By simulation with parameterized models powerful search methodologies can be applied to find the most critical parameter combinations in the operation of a bus and to validate a system design this way. The application of several modern search methods is investigated and the results are compared to classical Monte Carlo Analysis.

Keywords: Bus System, Signal Integrity, Simulation, Search Methods, Validation.

Presenting Author's biography

Harald Günther studied computer science with electrical engineering at Technische Universität Dortmund. Currently he is research assistant and PhD student at Arbeitsgebiet Bordsysteme. His main interest of research is modeling of bus system physical layer components and simulation based bus system investigations.

1 Introduction and Motivation

Automotive bus systems are used to connect control units, intelligent sensors and actors in vehicles. From economical point of view extended cable networks are desirable. Systems are often operated close to their specification limits. To provide safety critical applications under these circumstances, the functionality of a bus system must be ensured with sophisticated methods.

An approach to detect problematic behaviour and its causes is the computational investigation of the transmitted signal quality (signal integrity or SI) of the physical layer of the bus system. Models for any component of a bus system have to be developed. Accuracy of models and methods has to be ensured by measurements with realistic systems.

However, many system parameters are not fixed, e.g. tolerances of devices, variations in the topology, external electromagnetic influences etc., so that there is a nearly unmanageable number of combinations. Validation of safe function becomes difficult. By simulation with parameterized models powerful search methodologies can be applied to find the most critical parameter combinations in the operation of a bus and to validate a system design this way.

2 Simulation Based Bus System Signal Integrity Optimization

Many parameters influence the signal integrity of automotive bus systems. To ensure maximum safety, possible worst cases in operation of bus systems and parameter values causing them have to be investigated. Interesting parameters to look more closely at within bus systems are parameters causing signal disturbances: device tolerances (for example asymmetries in termination resistors of the differential bus line or shifts in voltage level of sending or in detection level of receiving transceivers), disturbances in supply voltage and ground connections, or, within more extensive investigations, temperature differences between nodes influencing their behaviour or signal integrity of the system when branches in the topology are added, removed or even interrupted.

In [1] it is described how simulation models in VHDL-AMS (Very High Speed Hardware Description Language – Analogue and Mixed Signals) for the different components of bus systems can be designed and used in physical layer simulations. A transceiver model describing the signal integrity behaviour of a transceiver device was created using a physical approach and a transmission line model based on analytical approaches [2] is used. Having the models available search methodologies can be combined with physical layer simulation to find critical cases in bus system operation. Monte Carlo and sensitivity analysis, which are commonly used in circuit design, produce acceptable results [3], [4], [5]. But in the last years many new methodologies were developed which promise to produce better results in much shorter time [6], [7]. The application of modern search methodologies for the analysis of signal integrity of bus systems is investigated, especially the following methods: Tabu Search, Simulated Annealing, Genetic Algorithms and Swarm Optimization [8]. Results about the applicability to bus system investigation, adaptability to this problem, optimization run time and convergence to optima (worst cases) of those methods are presented.

In the following sections optimization results of the different methodologies are given to analyze the applicability and the efficiency of the algorithms in detail. Investigations about run time reduction and convergence within local optima are presented. Also a comparison between the applied modern search methodologies and the classical Monte Carlo Analysis concerning bus system optimization is provided. An application example in section 4 shows the necessity of using advanced search methods in practice.

3 Bus System Optimization with Search-Methods

For comparison of search methods, two test cases have been defined. The test cases are rather small, because first their worst case parameter set is obtained by analyzing all possible parameter combinations (with some practical restrictions, see below). Later those worst case parameter values are compared to the ones found by the different search methods.

3.1 Test Cases

Test case 1 consists of a bus system network with 3 nodes which are connected like a chain with cables of length 1 m (see Fig. 1). The nodes at the two ends, 1 and 2, are terminated with an adapted termination in order to avoid reflections, R_{BP} and R_{BM} have the value of 47 Ω. Node 3 in the middle is high ohmically terminated, R_{BP} and R_{BM} have the value of 1.3 kΩ. Sender is node 1.

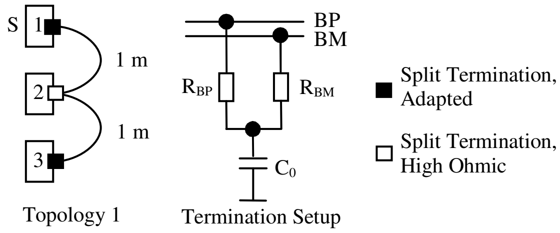
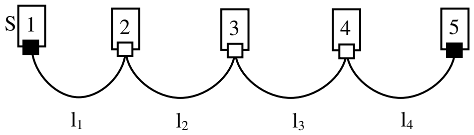


Fig. 1 Setup of Test Case 1

In the scenario of test case 1 the differential termination resistors R_{BP} and R_{BM} are the parameters to be varied, so that there are 6 variable parameters. This is of special interest because they have to be ideally symmetrically to ensure correct differential bus signal behavior.

Test case 2 consists of five nodes connected by transmission lines with lengths l_1 to l_4 (see Fig. 2). As in Test case 1 the two nodes at the ends, 1 and 5, are terminated with an adapted termination, the others in the middle are high ohmically. The values of the termination resistors R_{BP} and R_{BM} are fixed in this setup. Sender is as before node 1.



Topology 2 (Termination Symbols see Fig. 1)

Fig. 2 Setup of Test Case 2

In this test case the line length parameters are varied, so that there are 4 free parameters. Variation of line length may cause problematic bus signal behaviour because the line lengths directly influence the overlapping of reflected signals.

3.2 Worst Cases of the Test Cases

To have a comparison base for the parameter values found by the search methods, the worst case parameters values for the test cases have to be obtained. To achieve this all possible parameter combinations for the test cases have to be evaluated.

In test case 1 the termination resistance values are varied within a 10% range (see. Tab. 1).

Tab. 1 Parameter Variation Ranges for Test Case 1

	Min. Value	Init. Value	Max. Value
R_{BP1}	42.3 Ω	47 Ω	51.7 Ω
R_{BM1}	42.3 Ω	47 Ω	51.7 Ω
R_{BP2}	1.17 k Ω	1.3 kΩ	1.43 kΩ
R_{BM2}	1.17 k Ω	1.3 kΩ	1.43 kΩ
R_{BP3}	42.3 Ω	47 Ω	51.7 Ω
R_{BM3}	42.3 Ω	47 Ω	51.7 Ω

If the variation of the six resistors would be done in small steps, for example 1 Ω steps, then there is a great amount of combinations, in the example approximately $10^4 \cdot 260^2 = 676 \cdot 10^6$, that has to be checked. To reduce this extremely high amount of simulations to a reasonable number, the amount of values within the parameter range will be reduced. For each parameter there will be 5 different values: The edges of the interval, the initial value and the two values exactly in the middle between the initial and the edge value. So that in case of a resistor with 47 Ω initial value and a tolerance range of 10 % there will be the values 42.3 Ω, 44.65 Ω, 47 Ω, 49.35 Ω and 51.7 Ω. This reduces the amount of simulations to $5^6 = 15625$ which can be run in a reasonable amount of time.

The found worst case for test case 1 is listed in Tab. 2.

Tab. 2 Worst Case Parameter Values for Test Case 1

	Worst Case	Min. Value	Max. Value
R_{BP1}	42.3 Ω	42.3 Ω	51.7 Ω
R_{BM1}	42.3 Ω	42.3 Ω	51.7 Ω
R_{BP2}	1.17 kΩ	1.17 kΩ	1.43 kΩ
R_{BM2}	1.17 kΩ	1.17 kΩ	1.43 kΩ
R_{BP3}	42.3 Ω	42.3 Ω	51.7 Ω
R_{BM3}	42.3 Ω	42.3 Ω	51.7 Ω

The worst case values appear to be for all varied parameters exactly the minimum allowed values. A comparison between the differential bus signals with initial and worst case parameter values at the node with the worst quality value is shown in Fig. 3.

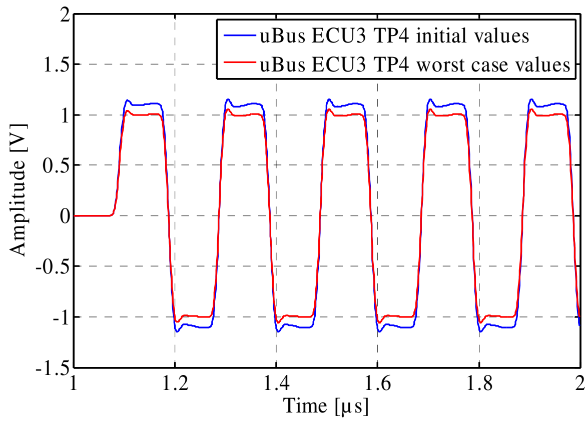


Fig. 3 Worst Case Signal for Test Case 1

It can be seen that the amplitude of the differential bus signal in Fig. 3 with worst case values is lower than the amplitude of the signal with initial values. Thus the quality value of the signal with worst case values is higher since it has greater potential to injure eye diagram mask borders because of its lower amplitude and thus is more interesting concerning worst case optimization.

The quality value of a signal is calculated by comparison of the signal with an eye diagram mask and determining weather the signal violates the borders of the mask. In case of violation the area of intersection between mask and signal is taken as quality value, in case of no intersection the area between signal and mask lines is proportional to the quality value of the signal to evaluate its potential risk of violating the mask. For details about automated signal quality evaluation see [1].

In test case 2 the parameters to vary are the lengths of the transmission lines l_1 to l_4 (see. Fig. 2) connecting the nodes. The line lengths are varied between 0.5 m and 5 m. In order to reduce the amount of simulations to check the number of parameter value combinations is reduced similar to test case 1. Here the line lengths are varied within the given interval in steps of 0.5 m so that there are ten different values for each parameter to be checked. This gives a total amount of simulations of 10^4 which have to be checked. Tab. 3 lists the parameter values of the three found worst cases together with the quality value of the corresponding parameter value combination.

Tab. 3 Worst Case Parameter Values for Test Case 2

	Worst Case	2nd Worst Case	3rd Worst Case
l_1	5.0 m	4.5 m	5.0 m
l_2	4.5 m	4.5 m	5.0 m
l_3	4.5 m	4.5 m	5.0 m
l_4	4.5 m	5.0 m	4.5 m
Quality	0.1014	0.0807	0.0782

Tab. 3 shows that the worst case parameter value combinations are not located at the border of the value intervals as in test case 1. A bad signal integrity in a chain topology like it is in test case 2 occurs when the line lengths are of approximately the same length, so that the reflections caused by the connected nodes overlap in a very disadvantageous way. Fig. 4 shows the differential bus signals for the initial and the worst case.

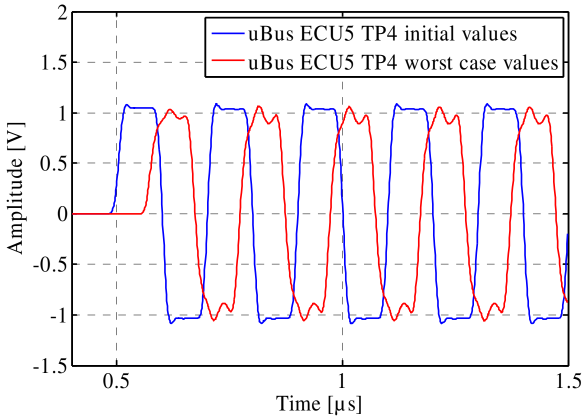


Fig. 4 Worst Case Signal for Test Case 2

It can be seen that the worst case signal has more ringing in the amplitude caused by disadvantageously overlapping reflections and a greater time delay caused by increased line lengths.

Due to the reflection overlapping at similar line lengths there are many local optima in this test case which will be a task to master for the search methods.

In the next sections the results of common search methods like Tabu Search (TS), Genetic Algorithms (GA), Simulated Annealing (SA) and Particle Swarm Optimization (PSO) on these two test cases will be presented. They will be compared to the results of a Monte Carlo Analysis (MC). The tables in the following sections list the average parameter values of the found worst case for the different search methods, and give their relative deviation from the found worst case values from this section.

3.3 Tabu Search

The results for test case 1 are listed in Tab. 4. The first row of the table lists the average worst case parameter values found by Tabu Search, the second row gives their relative deviation from those worst case values found in the previous section.

For test case 1 Tabu Search manages to find the exact worst case parameter values within 214 simulation runs. This is a rather good value, but the test case is friendly without local optima.

Tab. 4 Average TS Results for Test Case 1

Sim. Runs	R_{BP1} [Ω]	R_{BM1} [Ω]	R_{BP2} [kΩ]	R_{BM2} [kΩ]	R_{BP3} [Ω]	R_{BM3} [Ω]
214	42.3	42.3	1.17	1.17	42.3	42.3
(Dev.)	0 %	0 %	0 %	0 %	0 %	0 %

Tab. 5 shows the results for test case 2.

Tab. 5 Average TS Results for Test Case 2

Sim. Runs	Qual.	l_1	l_2	l_3	l_4
1131	0.0218	0.96 m	1.06 m	0.97 m	0.83 m
(Dev.)	79 %	81 %	76 %	78 %	82 %

For test case 2 Tabu Search does not give good results. Here it was not able to find a combination with a quality above 0.022 (worst: 0.1014) within more than 2500 simulation runs. The values of the found worst case were still in close range of the initial values of 1 m of line length.

Since Tabu Search is a search method that operates very locally, it has problems with local optima. This is proved here with test case 2, where Tabu Search is not able to overcome the first local optimum it reaches.

3.4 Genetic Algorithm

Genetic Algorithm shows an indifferent behaviour on the two test cases. The results for test case 1 are listed in Tab. 6.

GA manages to find the optimal configuration already in the initial population, so that actually no optimization is necessary. A probable reason for this is the small range of parameter variation.

Tab. 6 Average GA Results for Test Case 1

Sim. Runs	R_{BP1} [Ω]	R_{BM1} [Ω]	R_{BP2} [kΩ]	R_{BM2} [kΩ]	R_{BP3} [Ω]	R_{BM3} [Ω]
0	42.3	42.3	1.17	1.17	42.3	42.3
(Dev.)	0 %	0 %	0 %	0 %	0 %	0 %

The results for test case 2 are listed in Tab. 7.

Tab. 7 Average GA Results for Test Case 2

Sim. Runs	Qual.	l_1	l_2	l_3	l_4
308	0.0247	1.10 m	1.23 m	0.91 m	1.06 m
(Dev.)	75.7 %	77.9 %	72.7 %	79.7 %	76.5 %

In the second test case GA is not able to perform well. Even in optimization runs with more than 2000 simulations the found parameter values still lie within the range of the initial values. This seems to be a rather strong local maximum, since either Tabu Search nore the in general more powerful GA are not able to leave it. A possible approach to improve the results for GA here are to perform modifications within the mutation and crossover settings. But these settings can be very problem specific as test case 1 shows where the worst case was easily found with the same settings that fail for test case 2. So as already proposed in [1], GA is able to produce good results with significant efforts that have to be spent in adaptation work, which limits its practical use.

3.5 Simulated Annealing

Simulated Annealing is able to find the worst case for both test cases. Tab. 8 shows the results for test case 1.

Tab. 8 Average SA Results for Test Case 1

Sim. Runs	R_{BP1} [Ω]	R_{BM1} [Ω]	R_{BP2} [kΩ]	R_{BM2} [kΩ]	R_{BP3} [Ω]	R_{BM3} [Ω]
1353	42.3	42.3	1.17	1.17	42.3	42.3
(Dev.)	0 %	0 %	0.1 %	0 %	0 %	0 %

In test case 1 SA manages to reach the worst case parameter values within very little deviation. Tab. 9 shows the results for test case 2.

Tab. 9 Average SA Results for Test Case 2

Sim. Runs	Qual.	l_1 [m]	l_2 [m]	l_3 [m]	l_4 [m]
1015	0.1006	4.74	4.50	4.91	4.55
(Dev.)	0.8 %	5.1 %	0.0 %	9.1 %	1.1 %

In test case 2 the parameter values do not fit exactly to those obtained in section 3.2. But since there the values were fixed to certain steps here with simulated annealing the values are free to vary within the given range. All the found values lie in average between 4.5 m and 5 m with a deviation of less than 10 % from the worst case values so that it can be said that the worst case with respect to its definition from section 3.2 was found.

The rather high amount of simulation runs compared to Tabu Search, which is comparatively quick in test case 1 and without success in test case 2, can be explained as follows: Simulated Annealing is based on a randomized approach of neighborhood search and does not evaluate the complete neighborhood of the current search point like Tabu Search. In addition SA accepts value combinations with worse quality from time to time to be able to avoid getting stuck in local optima. Thus there are more simulation runs with parameter values in SA that have to be rejected than in Tabu Search. That SA needs for test case 2 less simulation runs than for test case 1, although the initial parameter values are located nearer to the worst case values in test case 1 than in test case 2, can be explained with the smaller number of parameters in test case 2 (4 compared to 6 in test case 1, so 50% less parameters have to be varied).

Because the parameter values are free to vary in SA it was even able to find a case that has higher quality than the worst case where the parameter values were bound to fixed steps to reduce the amount of combinations. The values of the found worst case by SA are listed in Tab. 10.

Tab. 10 Worst Case Found by SA for Test Case 2

Sim. Runs	Qual.	l_1 [m]	l_2 [m]	l_3 [m]	l_4 [m]
590	0.1046	4.90	4.91	4.80	4.91

3.6 Particle Swarm Optimization

Particle Swarm Optimization performs best out of the compared search methods on both test cases: Within comparatively fewest simulation runs it was able to find the worst cases for both test cases with only small deviation. Like with SA the parameter values in PSO were also free to vary and not bound to fixed steps. The results of test case 1 are listed in Tab. 11.

Tab. 11 Average PSO Results for Test Case 1

Sim. Runs	R_{BP1} [Ω]	R_{BM1} [Ω]	R_{BP2} [kΩ]	R_{BM2} [kΩ]	R_{BP3} [Ω]	R_{BM3} [Ω]
160	42.3	42.3	1.21	1.20	42.3	42.3
(Dev.)	0 %	0 %	3.3 %	3.0 %	0 %	0 %

The values of the adapted termination resistors of node 1 and node 3 are matched exactly. Those of the high ohmic terminated resistors of node 2 have a little deviation of about 3 %. The results for test case 2 are listed in Tab. 12.

Tab. 12 Average PSO Results for Test Case 2

Sim. Runs	Qual.	l_1 [m]	l_2 [m]	l_3 [m]	l_4 [m]
94	0.0962	4.70	4.07	4.74	4.98
(Dev.)	5.1 %	5.9 %	9.5 %	5.3 %	10.6 %

Compared to SA the results of PSO for test case 2 have a little higher deviation, but still match the found worst case from section 3.2 good. PSO needs much less simulation runs to reach the worst case than other methods. And like SA it was also able to find parameter values that have a worse quality than the worst case from section 3.2. They are display in Tab. 13.

Tab. 13 Worst Case Found by PSO for Test Case 2

Sim. Runs	Qual.	l_1 [m]	l_2 [m]	l_3 [m]	l_4 [m]
160	0.1025	4.77	4.87	4.68	5.00

So it can be said, that PSO gives the best performance of the compared search methods for the test cases. A comparative overview about the presented results of the search methods will be given in section 3.8.

3.7 Monte Carlo

After analyzing the modern search methods in the previous sections the results will be compared to classic Monte Carlo analysis as it can be found in many common circuit design and analysis programs. Monte Carlo creates parameter value combinations randomly and analyses them, so that after some amount of simulation runs the worst case is found with some probability. Here the maximum threshold was set to 1000 simulation runs to keep comparability. If MC will need more than 1000 simulation runs it will definitely have less good performance than the other methods.

The results for test case 1 are listed in Tab. 14.

Tab. 14 Average Monte Carlo Results for Test Case 1

Sim. Runs	R_{BP1} [Ω]	R_{BM1} [Ω]	R_{BP2} [kΩ]	R_{BM2} [kΩ]	R_{BP3} [Ω]	R_{BM3} [Ω]
515	42.97	43.02	1.255	1.286	43.42	43.81
(Dev.)	1.6 %	1.7 %	7.3 %	9.9 %	2.6 %	3.6 %

It can be seen that within 1000 simulation runs for test case 1 MC is only able to find the worst case parameter values with some deviation. It does not reach the final worst case values, though all parameter values are within 10 % deviation of the worst case values. The amount of simulation runs needed to find the worst parameter combination varies for the different Monte Carlo runs as expected: The fastest run managed to find the worst case within 77 simulation runs, the slowest one needed 899 simulations.

The results for test case 2 are listed in Tab. 15.

Tab. 15 Average Monte Carlo Results for Test Case 2

Sim. Runs	Qual.	l_1 [m]	l_2 [m]	l_3 [m]	l_4 [m]
312	0.0887	4.29	3.67	4.21	4.09
(Dev.)	12.5 %	14.2 %	18.4 %	6.5 %	9.1 %

For test case 2 the deviation between the found worst case parameter values and the actual worst case values is much higher than for test case 1. Here the parameter ranges were bigger than in test case 1 so that there are less parameters to vary, the probability of finding value combinations close to the worst case is smaller.

3.8 Comparison of Methods

After presenting the results of the different search methods for the test cases the methods will be compared concerning found quality and needed amount of simulation runs. Fig. 5 shows the comparison for test case 1.

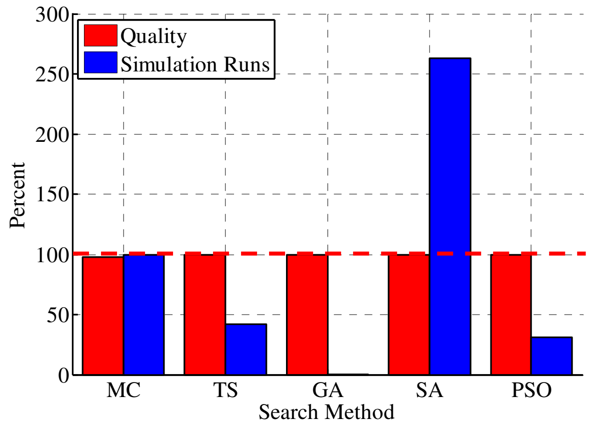


Fig. 5 Comparison of Methods for Test Case 1

The shown qualities are displayed in percent of the worst case quality, the amount of simulation runs is referring to the Monte Carlo value. All search methods are able to find the worst case for test case 1 within small deviation. The amount of needed simulation runs varies strongly: GA performs best because the worst case combination is already found in the initial generation, TS and PSO need clearly less than half of the simulations of MC, SA clearly more than twice.

The results for test case 2 are shown in Fig. 6.

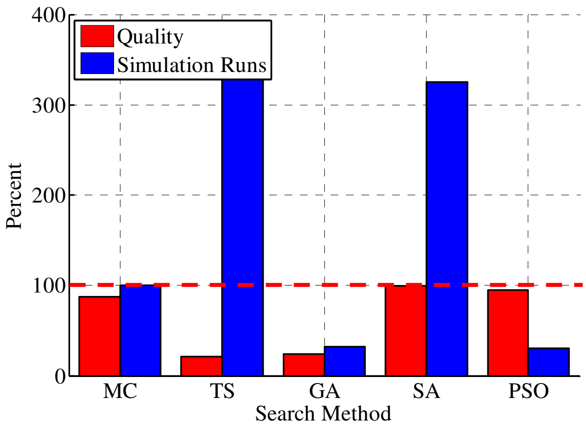


Fig. 6 Comparison of Methods for Test Case 2

For the more complex test case 2 only SA and with a little deviation PSO were able to find the worst case. With a deviation of about 10 % also MC succeeded. TS and GA got stuck in the initial local optimum. From the three successful methods PSO clearly performs best, it only needs a third of the simulation runs that MC needs. TS needs more than three times more than MC.

Summarized it is obvious that PSO clearly gives the best performance over the two test cases. In both cases PSO was able to find the worst case within small deviation needing comparatively few simulation runs. No other of the investigated search method showed a comparable performance on both test cases.

4 Application Example

After extensive comparison of different search methods on two small test cases the practical use of applying search methods will be shown in an application example. The topology in Fig. 7 consisting of 7 nodes will be analyzed using Particle Swarm Optimization.

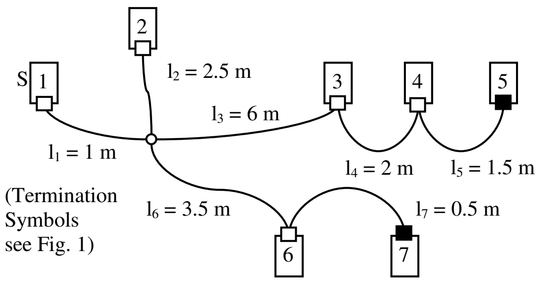


Fig. 7 Application Example Topology

The topology with the given line lengths in Fig. 7 already shows bad signal integrity due to many reflections (see Fig. 8) but should be still operational. The aim of the worst case optimization is to check weather changing the line lengths can cause worse reflections und thus influence the signal integrity in a negative way so that the bus is probably not operational any more. To achieve this aim the line length given in Fig. 7 are varied for some meters around their initial values. PSO manages to find a configuration with much worse signal integrity, the comparison of 4 transferred bits within the differential bus signals is shown in Fig. 8.

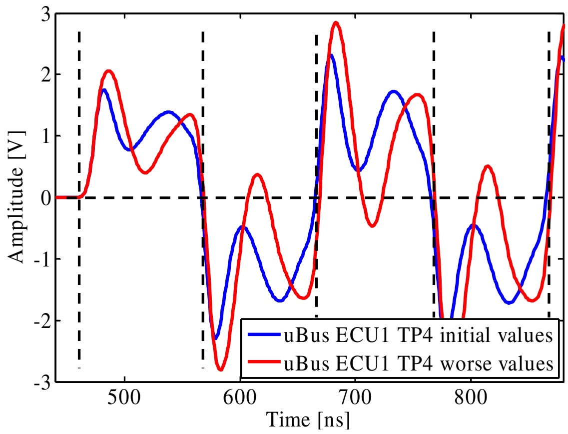


Fig. 8 Signal Integrity in Application Example

It can be seen clearly that the changed line lengths cause a heavy disturbance at the middle of each transferred bit. The disturbance even crosses the zero line, so that most probably the bit will not be detected properly by the receiver and thus the bus will not be functional. The line lengths which cause such a problematic signal integrity are listed in Tab. 16.

Tab. 16 Worst Values for Application Example

l_1 [m]	l_2 [m]	l_3 [m]	l_4 [m]	l_5 [m]	l_6 [m]	l_7 [m]
1.9	5.0	8.5	0.5	4.0	1.0	0.3

With this example could be shown, that only the design of a topology, even without consideration of parameter tolerances or external influences, can cause serious problems with signal integrity. Thus simulation based bus system topology validation is an important issue since such failure cases would be hard to discover only by measurement.

5 Conclusion

To ensure correct functionality of automotive bus systems sophisticated methods are needed. An important validation step is analyzing the bus system behaviour with computational investigation of the signal integrity within the physical layer. Models for each component of bus systems were developed and their accuracy verified by measurements with realistic systems.

The resulting comparison of the modern search methods with Monte Carlo demonstrates that Particle Swarm Optimization shows the best performance on the test cases, while other methods have problems with local optima or long optimization times. This knowledge is used to analyze a realistic application example containing a complex bus system with several nodes. With the help of simulation based worst case analysis it is explained, that the design of a bus system topology has to be done carefully in order to avoid malfunctions.

6 References

[1] H. Günther, S. Frei, T. Wenzel: Simulation Methods for Signal Integrity of Automotive Bus Systems, APEMC 2010.

[2] K. Siebert, H. Günther, S. Frei, W. Mickisch: Modeling of Frequency Dependent Losses of Transmission Lines with VHDL-AMS in Time Domain, International Zurich EMC Symposium 2009.

[3] J. G. Amar: The Monte Carlo Method in Science and Engineering, Computing in Science and Engineering March/April 2006 Vol. 8 No. 2 pp. 9-19.

[4] A. Singhee, R. A. Rutenbar: From Finance to Flip Flops: A Study of Fast Quasi-Monte-Carlo Methods from Computational Finance Applied to Statistical Circuit Analysis, ISQED 2007.

[5] J. Ogrodzki: Circiut Simulation Methods and Algorithms, CRC Press 1994.

[6] I. Wegener: Simulated Annealing Beats Metropolis in Combinatorial Optimization, ICALP 2005.

[7] F. Neumann, I. Wegener: Randomized Search, Evolutionary Algorithms, and the Minimum Spanning Tree Problem, GECCO 2004.

[8] E. K. Burke, G. Kendall (Hrgs.): „Search Methodologies“, Springer, 2005.