Modeling, Analysis and Design of Complex Quality Testing Systems using a Hierarchical Simulation framework

Abhishek K. Shrivastava, Yu Ding, Jon Coody, Feng Niu, and Darek Ceglarek

Abstract— **This paper describes a hierarchical simulation framework for modeling complex systems. We describe the implementation of the framework as a two-stage model in the specific case of a phone quality testing process; this is a real-life example based on a study for a major cell phone manufacturer. The framework is applicable when certain factors are difficult to integrate into the simulation model and the system can be studied conditional to these factors. The hierarchical model can be used for studying system behavior for all the factors; and inferring important / useful relationships for system design / parameter selection. A simple design parameter selection problem for the phone quality testing process is considered to exemplify this.**

I. INTRODUCTION

COMPUTER simulations are an important tool for studying complex systems and have been used extensively in practice ([1]–[4]). The basic intent is to observe the behavior of the system for changes in different factor levels. This requires the simulation model to be built in a manner that the various factor levels can be varied. In certain situations, due to the complexity and / or existence of large number of factors, it may not be possible to build a single-stage simulation model addressing all the complexities for the entire system. In such situations, it is still possible to develop a model describing the system with certain factors fixed at certain levels. This paper describes a hierarchical simulation framework for modeling such complex systems. In particular, we describe a two-stage implementation of the framework for a phone quality testing system.

The phone quality testing system studied here is a real-life example based on a study for a major cell phone manufacturer. The testing process being studied is the final stage of the production process, after complete assembly of the phone. Phones are mounted on fixtures and go through a number of test blocks. The testing equipments are shared among the fixtures. The communication between fixtures and equipments is through buses which are also shared between various equipments. The sharing of resources, along with constraints on sequencing of test blocks, make the testing process fairly complex. The testing process is controlled by a computer controller which sends commands to the various equipments, buses, etc. for executing the testing process. The first objective of the study was to develop a simulation tool providing the design engineer with the ability to simulate the process for a given system configuration before actually implementing it. The current simulation capability in use by the company is inadequate in meeting these needs.

The simulation model developed is a two-stage model based on the hierarchical framework proposed. The top-level model can be seen as a general simulation model with the ability to cater to changes in all factor levels. The second-level model is a conditional simulation model with certain factors fixed at given levels. When studying the system for a given set of factor levels, the top-level model generates a second-level model with certain factors fixed at specified levels. This second-level model can then be simulated at the specified levels (or any other levels) of the remaining factors. The top-level model is thus a subroutine that outputs the conditional simulation model for the system, i.e., the second-level model. Specifically, for the phone testing system, the second-level model has the test sequence embedded in it and can be used for evaluating the same test sequence with different rack configurations. So, for a given phone model a second-level model is enough to analyze and even design the testing setup.

The hierarchical framework in an *n*-stage implementation will contain a set of factors, fixed at specified levels, at each stage. Thus, the simulation model generated at this level will be conditional to these factors and the factors fixed in the preceding stages. It must be noted that the choice of the factors being fixed at a particular stage directly impacts the ease of use of the framework. A factor which needs to be changed rarely is a more suitable candidate at a preceding stage, like the top-level model above, as compared to the one which is frequently changed. This would allow the analyst to work with the generated conditional model for a longer period before generating another one.

This paper is organized as follows. Section 2 describes the phone quality testing system. Section 3 describes the hierarchical framework implemented as a two-stage simulation model for the phone quality testing system. Section 4 discusses a testing scenario and provides results from the simulation model. In section 5, we consider the example of a simple test rack design problem to explain how the framework can be used for designing the phone testing

---

This work was supported in part by the National Science Foundation under Grant No. DMI-0217481.

Abhishek K. Shrivastava and Yu Ding are with the Industrial and Systems Engineering Department, Texas A&M University, College Station, TX 77843-3131 USA (e-mail: akshriv@tamu.edu; yu-ding@tamu.edu).

Jon Coody is with the Motorola iDen Subscriber Group, Plantation, FL 33322 USA (e-mail: Jon.Coody@motorola.com).

Feng Niu is with the Motorola Labs, Plantation, FL 33322 USA (e-mail: F.Niu@motorola.com).

Darek Ceglarek is with the Industrial and Systems Engineering Department, University of Wisconsin-Madison, Madison, WI 53706 USA (e-mail: darek@engr.wisc.edu).

1-4244-0065-1/06/$20.00 ©2006 IEEE

station. Finally, in section 6 we present our conclusions.

II. PHONE QUALITY TESTING SYSTEM

The testing system is defined by the test rack configuration, the test sequence and the number of operators at the testing station.

The test rack configuration is defined by the number of fixtures, the number of buses, the number of units of all equipments, and a mapping between the equipments, buses and fixtures. Equipments are required for conducting tests on the phones; the fixtures hold the phones during the testing process, and the buses provide communication between fixtures and equipments during the testing process. The mapping between the equipments, buses and fixtures describes which particular units of an equipment can be used by a particular fixture, and which bus needs to be used for communicating with a particular unit of an equipment.

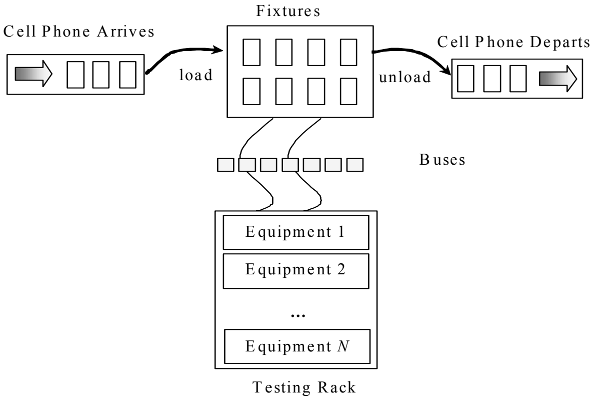

Fig. 1. A schematic representation of the phone quality testing procedure

The test sequence gives the set of test actions to be performed on each phone. These test actions have deterministic time length and are grouped into test blocks. The sequence in which the test blocks are executed on a phone is flexible. There is a preferred sequence of the test blocks, but a test block may be skipped as long as it satisfies the constraints set on the sequence. This may happen due to the unavailability of the equipment units required for running the block on the particular fixture, or due to a skipped block which is required to be completed before executing the current block. Any block which is skipped is said to be 'floating'. After reaching the end of the sequence, any blocks that are floating are revisited in the order in which they appear in the preferred test sequence. The constraints governing the floating of blocks are termed floating constraints and the sequence is termed as floating sequence. The constraints specify the range of the test blocks within which a test block may be floated and its dependency on any other blocks.

Fig. 1 is a schematic representation of the testing procedure. The testing procedure starts with the arrival of a phone at the testing station. Operators load and unload phones onto the fixtures. There could be one or more operators and they may have a specified operator to fixture mapping for both loading and unloading operations. The phone is loaded onto an available fixture or is put in a queue until a fixture becomes available. The operators being human, the loading and unloading processes have a stochastic nature. Once the phones are loaded, the execution of the testing logic is an automated process run by a controller. The equipment units required for the execution of a test block are requested by the fixture at the beginning of the test block and are held for the duration of the test block. That is, these units of equipments are not available to any other fixture for the duration of the block. The particular bus required for a test action is held by the fixture only for the duration of the test action requiring it, unlike the equipments. After all the test blocks have been completed, the phone is unloaded from the fixture by an operator and it exits the system.

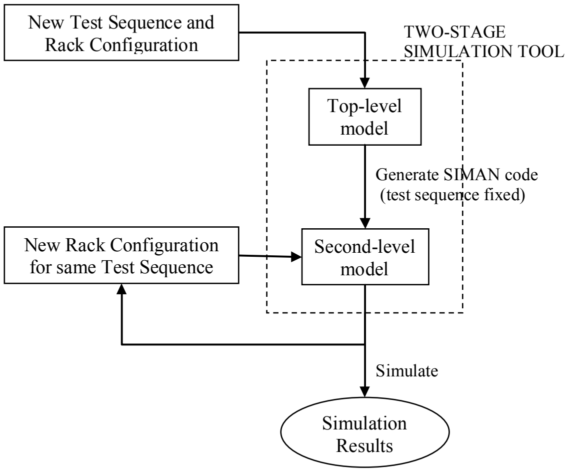

Fig. 2. Two-stage simulation model for the phone quality testing system.

The testing process is fairly complex and is tough to study through analytical procedures. The complexity is due to the complex sharing of equipment and buses, two-step method of accessing the equipment, the rules for holding equipments and buses by a fixture, and the logic of floating test blocks. The size of the real systems, that is, the number of fixtures, the number of test blocks, the number of equipments and their units, and the number of buses, makes the problem even more difficult to pursue analytically. The phone testing station studied in this paper had 8 fixtures, 29 test blocks, 8 equipments with number of units varying between 1 to 8, and 14 buses.

III. TWO-STAGE SIMULATION FRAMEWORK FOR THE PHONE QUALITY TESTING SYSTEM

The simulation tool developed is a two-stage model based on the hierarchical framework. In the first stage, the top-level model generates the second-level model for a fixed test sequence. The top-level model is implemented in C++ that takes the test sequence and a test rack configuration as input, and gives the corresponding secondary model as output. The second-level model can be used for studying the different test rack configurations for a particular test sequence as it has the test sequence embedded in it. This second-level model is a simulation model written in SIMAN ([5]), a process-oriented simulation language. Thus, the top-level model is actually a code generation routine which generates the SIMAN code for a given test sequence. Fig. 2 gives a flow diagram of the two-stage simulation model implemented for the phone quality testing system.

The second-level model can be used for studying the testing process of phone models undergoing the same set of tests and with the same testing logic. The second-level model takes as its input the test rack configuration. It can thus accommodate changes in the number of fixtures, number of buses, number of units of all equipment, and the mapping between the equipments, buses and fixtures. The output of the simulation (second-level model) is a set of performance measures of the system. The statistics that are observed during the simulation runs described in Table I.

The factor that is fixed in the top-level model is the test sequence. There are two reasons for this choosing this factor to fix in the top-level model. Firstly, this simplifies the development of the second-level model. The test sequence is generally very long with more than 20-25 test blocks, each of which comprising of a few tens of test actions. If the test sequence is not fixed for the second-level model, the model would become too complex with a large number of input variables to handle; this is because the entire test sequence would need to be considered as an input, which would take a large number of variables to define. Secondly, since the test sequence is fixed for a particular phone model, this choice allows repetitively using the same less complex simulation model for studying the testing process of the same phone. This also reduces the simulation time as compared to what a general model will take due to the reduced complexity (lesser input variables) while analyzing the testing process of a particular phone model.

TABLE I
STATISTICS OBSERVED DURING A SIMULATION RUN OF THE SECOND-LEVEL MODEL

Statistic	Description
Cycle time without loading	Time spent by a phone in the system, starting after it has been loaded onto the fixture and ending after the execution of the last test block in test sequence
Cycle time with loading	Time spent by a phone in the system, starting when the loading onto the fixture begins and ending after the unloading from the fixture is complete
Fixture utilization (for each fixture)	Ratio of the total time when the fixture is occupied to the total simulation time; fixtures are not considered as occupied during loading and unloading operations
Bus utilization (for each bus)	Ratio of the total time when the bus is in use to the total simulation time
Equipment utilization (for each unit of each equipment)	Ratio of the total time when the equipment unit is busy (i.e. held by a fixture) to the total simulation time

IV. TESTING STATION ANALYSIS USING THE SIMULATION TOOL

The results presented here are for a test scenario actually used by the cell phone manufacturer for a particular phone model. The testing station has one operator who performs both loading and unloading of phones onto the eight fixtures. The loading and unloading times have been assumed to follow a normal distribution with mean 10 seconds and standard deviation 5/3 seconds. There are eight types of equipments in the test rack. We will omit the actual names of these equipments here and refer to them by names EQ1, EQ2, …, EQ8. The number of units of these equipments is 8, 8, 8, 4, 4, 3, 1 and 8, respectively. The equipment units are connected with the fixtures through 14 buses. Each unit of equipment EQ8 requires a separate bus and caters to a different fixture. That is, the buses do not connect a unit of any other equipment to any fixture. Thus, eight buses are dedicated to EQ8, and rest six buses are used for connecting the other equipments with the fixtures.

The operator loading description, test sequence and the rack configuration are passed to the top-level model which generates the second-level model. The second-level model is a SIMAN program which is simulated to analyze the system. The results obtained from this simulation program are presented below.

Table II shows the average cycle times, both with and without loading time, and the fixture utilizations for each fixture. The difference between the cycle times with and without loading is roughly 22 to 25 seconds. The loading and unloading operations for a particular phone sum up to about 20 seconds. The extra 2 to 5 seconds are then the total waiting times for the operator for loading and unloading operations. The sum of the deterministic test times is 150.66 seconds. The difference between cycle times without loading and the deterministic total test time is due to the waiting times for equipments and buses. The fixture utilizations decrease from about 85% to 70% from fixtures 1 to 8. This is due to the fact that the operator is assumed to have a preference for loading and unloading fixture 1 as compared fixture 2 and so on, whenever two or more fixtures are

TABLE II
AVERAGE CYCLE TIMES AND FIXTURE UTILIZATIONS FOR TEST SCENARIO IN SECTION 4

Fixture number	Cycle Time without loading (in seconds)	Cycle time with loading (in seconds)	Fixture utilization
1	153.43	175.88	0.85487
2	153.15	175.72	0.85251
3	153.28	175.76	0.85110
4	153.15	176.13	0.84295
5	153.48	176.57	0.83138
6	153.32	176.54	0.81097
7	153.47	177.72	0.76593
8	153.76	178.89	0.69562

waiting for the operator. Consequently, the difference between cycle times with and without loading are also higher for fixture 8 as compared to fixture 1.

Table III gives the equipment utilizations for each unit of all equipments. The equipment utilizations for some equipment units, like unit 3 of EQ6, are low. If cost reduction were an objective it would be recommendable to study the system performance after removing unit 3 of EQ6. Utilizations for EQ4 are also low and one of EQ4 units would possibly be the next better candidate for removal.

Table IV gives the utilizations for all the buses. The bus utilization for buses 6 to 14 is quite high compared to the buses 1 to 5. The buses 7 to 14 are dedicated to one unit each of EQ8. Bus 6 communicates with the only unit of EQ7, which is shared by all the fixtures. The utilization for bus 1 is relatively much low. Moving some high utilization equipment units from buses 2 to 5 to bus 1 may reduce the waiting time and thereby reduce the cycle times.

The phone throughput rate, i.e., number of phones tested in unit time, from the testing station is 3.84 per 100 seconds. The above results show the possibility of improving the throughput and / or reducing the cost of testing. The throughput could be improved by adding more units of shared equipments and buses, as these would reduce the waiting times; but this would increase the cost of the system. Changes in the mapping between equipments, buses and fixtures, like allocating some equipment units, currently on buses 2 to 5, to bus 1, may improve the throughput rate. Equipment units with low utilization could be removed from the system. This would reduce the cost of the system while having small increment in the throughput rate.

TABLE III
EQUIPMENT UTILIZATIONS FOR TEST SCENARIO IN SECTION 4

Equip ment name	Utilization, for unit number							
	1	2	3	4	5	6	7	8
EQ1	0.82	0.81	0.81	0.81	0.79	0.77	0.73	0.66
EQ2	0.35	0.35	0.35	0.35	0.34	0.33	0.32	0.29
EQ3	0.69	0.62	0.69	0.67	0.68	0.56	0.69	0.65
EQ4	0.11	0.11	0.10	0.11	-	-	-	-
EQ5	0.27	0.29	0.26	0.28	-	-	-	-
EQ6	0.66	0.44	0.07	-	-	-	-	-
EQ7	0.42	-	-	-	-	-	-	-
EQ8	0.85	0.85	0.85	0.84	0.83	0.81	0.76	0.69

V. A SIMPLE TEST RACK DESIGN PROBLEM USING THE SIMULATION TOOL

An important use of the simulation tool is in designing the testing station. We will present a simple example to illustrate how the tool can be utilized for this purpose. The process optimization approach used is a classical response surface method ([6], [7]). We start with a screening experiment and analyze it using a linear model. This helps in selecting few important factors. We then perform experiments by varying the levels of these few factors and analyze using a second-order model to search for the optimum.

The test rack design problem is to determine the test rack configuration that maximizes the throughput of tested phones from the testing station and minimizes the cost of testing, for a given test sequence. The decision parameter is the test rack configuration. The minimization of cost restricts the number of units of all equipment available in the system. We get rid of the second objective by considering constraints on the number of units of each of the equipments. This is not an equivalent formulation but represents the problem fairly well.

TABLE IV
BUS UTILIZATIONS FOR TEST SCENARIO IN SECTION 44

Bus Number	Utilization	Bus Number	Utilization
1	0.06657	8	0.33381
2	0.20811	9	0.33308
3	0.20807	10	0.33020
4	0.19598	11	0.32480
5	0.20666	12	0.31717
6	0.39071	13	0.29934
7	0.33426	14	0.27134

The optimization problem considered here is to maximize the throughput rate given the test sequence, the maximum number of fixtures available, the maximum number of units available for all equipment and the number of buses available. In this example, we have considered the same test sequence as in section 4. Thus, we use the same second-level model generated in section 4 to run the simulations. From the knowledge of the process, it is known that the number of units of equipment EQ8 is same as the number of fixtures. Therefore, we do not consider it as a decision variable. The maximum number of fixtures available is 8, the maximum number of available units for equipments EQ1, EQ2 and EQ3 is 8, for equipments EQ4 and EQ5 is 6, for equipment EQ6 is 5 and for equipment EQ7 is 2. The maximum number of buses available is 14, of which 8 are used by

TABLE V
FACTORS AND LEVELS

Factor	Level		
	0	1	2
Fix	1	5	8
EQ1	1	5	8
EQ2	1	5	8
EQ3	1	5	8
EQ4	1	4	6
EQ5	1	4	6
EQ6	1	3	5
EQ7	1	2	-

EQ8. Since the cost of the buses is much less when compared to the cost of the equipments, we do not consider the number of buses as a decision variable and fix it to its maximum. The results from Section 4 suggest that the mapping between equipments, buses and fixtures used in the example performs fairly well. Using this knowledge, we construct an ad hoc rule for generating the mapping for a given number of fixtures, equipment units and buses. The ad hoc rule is to generate the new mapping by adding or deleting equipment units or fixtures from the mapping in Section 4.

TABLE VI
SCREENING EXPERIMENT: DESIGN MATRIX

Run	Utilization, for unit number								Response, $\hat{y}$ (Throughput per 100s)
	Fix	EQ1	EQ2	EQ3	EQ4	EQ5	EQ6	EQ7	
1	0	0	0	0	0	0	0	0	0.56
2	0	1	1	1	1	1	1	0	0.56
3	0	2	2	2	2	2	2	0	0.56
4	1	0	0	1	1	2	2	0	0.56
5	1	1	1	2	2	0	0	0	0.72
6	1	2	2	0	0	1	1	0	0.72
7	2	0	1	0	2	1	2	0	0.64
8	2	1	2	1	0	2	0	0	0.64
9	2	2	0	2	1	0	1	0	0.68
10	0	0	2	2	1	1	0	1	0.56
11	0	1	0	0	2	2	1	1	0.56
12	0	2	1	1	0	0	2	1	0.56
13	1	0	1	2	0	2	1	1	0.64
14	1	1	2	0	1	0	2	1	0.72
15	1	2	0	1	2	1	0	1	1.52
16	2	0	2	1	2	0	1	1	0.64
17	2	1	0	2	0	1	2	1	0.48
18	2	2	1	0	1	2	0	1	0.68

For the problem on hand, we have 8 decision variables. As the first step of optimization procedure, we run a screening experiment to identify the critical factors. We use an 18 run orthogonal array $2^1 3^7$ ([8]) of strength 2 as a screening experiment. The factor levels for the experiment are given in Table V. The levels are coded as 0, 1 and 2. The design matrix for the screening experiment, in coded variables, and the response values (throughput per 100 seconds) are shown in Table VI. We fit the data to a linear model with no interaction terms. The results indicate that Fixture, EQ1 and EQ6 are the only significant factors. The p-values corresponding to the t-statistics for these factors are 0.531, 0.2158 and 0.1831, respectively. The p-values are higher than the popular standards for levels of significance. We do not reject EQ1 and EQ6 because the p-values for these factors are the least compared to the other factors. The factor Fixture is accepted since the engineering knowledge of the process suggests that the number of fixtures should have an impact on the throughput rate. The fitted linear model is

$$\hat{y} = 0.4923 + 0.0124 * Fix + 0.0254 * EQ1 \\ - 0.0483 * EQ6 \tag{1}$$

Since the screening experiment does not give us a good idea about the optimum, we perform a second-order experiment with a 3^3 full factorial design. The factors EQ2, EQ3, EQ4, EQ5 and EQ7 are fixed at levels 8, 4, 4, 8 and 1, respectively. The factor levels for the experiment for Fix, EQ1 and EQ6 are the same as in Table V. The design matrix for the screening experiment in coded variables and the response values (throughput per 100 seconds) are shown in Table VII.

We fit the data from the full factorial experiment to a second-order model. The results show that only Fixture and EQ1 are critical factors with their linear, quadratic and interaction terms being significant. The p-values corresponding to the t-statistics for all these effects are less than 0.02. The second-order model fitted is

TABLE VII
FULL FACTORIAL 3^3 EXPERIMENT: DESIGN MATRIX

Run	Utilization, for unit number			Response, $\hat{y}$ (Throughput per 100s)	Run	Utilization, for unit number			Response, $\hat{y}$ (Throughput per 100s)
	Fix	EQ1	EQ6			Fix	EQ1	EQ6	
1	0	0	0	0.56	15	1	1	2	2.72
2	0	0	1	0.56	16	1	2	0	2.56
3	0	0	2	0.56	17	1	2	1	2.72
4	0	1	0	0.56	18	1	2	2	2.72
5	0	1	1	0.56	19	2	0	0	0.64
6	0	1	2	0.56	20	2	0	1	0.64
7	0	2	0	0.56	21	2	0	2	0.64
8	0	2	1	0.56	22	2	1	0	2.64
9	0	2	2	0.56	23	2	1	1	2.72
10	1	0	0	0.64	24	2	1	2	2.8
11	1	0	1	0.64	25	2	2	0	3.36
12	1	0	2	0.64	26	2	2	1	3.84
13	1	1	0	2.56	27	2	2	2	3.88
14	1	1	1	2.72					

$$\hat{y} = -0.1121 + 0.2647 * Fix + 0.2459 * EQ1 \\ - 0.034 * (Fix)^2 - 0.0334 * (EQ1)^2 \\ + 0.0636 * (Fix \times EQ1) \tag{2}$$

The second-order relation (2) can be used to determine the optimum levels for Fixture and EQ1. Since (2) is second order quadratic, the solution to the maximization problem is either a stationary point or a point on the boundary. The stationary point can be obtained by equating the first partial derivatives to zero. The stationary point for (2) is (66.98536507, 67.45762303), which is outside the region of interest.

The estimated response surface from (2) is shown in Fig. 3. From Fig. 3, the response is maximized in the top right corner region of the solution space. The estimated response for design candidates in this region are given in Table VIII. The response is maximized at Fixture=8 and EQ1=8. Therefore, the throughput rate is maximized by 8 fixtures and 8 units of EQ1. The estimated optimal throughput is 3.73 per 100 seconds.

Considering that the level for EQ6 is not specified, the example in Section 4 is an optimal solution to the design problem. The value of EQ6 was 3 in the example. The throughput rate was 3.84 per 100 seconds. Simulation runs for EQ6=2 and EQ6=4 at the optimal Fixture and EQ1 levels give throughput rates of 3.80 and 3.88 per 100 seconds, respectively. The throughput values being close to each other confirm that the EQ6 has little effect on the response. Thus, EQ6 can be reduced by one unit, in the example, to 2 without much loss in the throughput rate.

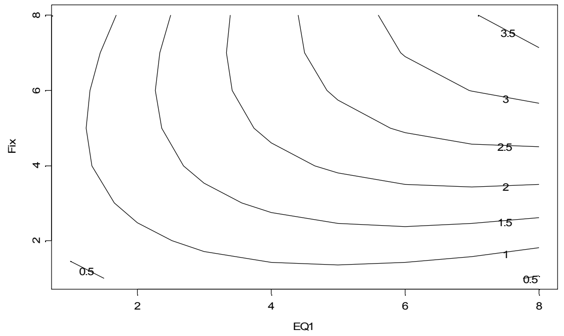

Fig. 3. Estimated response surface: second-order experiment.

VI. CONCLUSIONS

This paper describes a hierarchical simulation framework for modeling complex quality testing systems. The framework helps in reducing the complexity of the simulation tool and enhancing the speed of the simulation runs when working with the second-level model. The choice of the factors fixed in the top-level model dictates how well the potential of the framework is realized. The factors requiring changes are good candidates to be fixed in the top level model. The idea is to keep the complexity minimum at the second-level model and at the same time maximize the length of use of a second-level model before needing to generate a new one. We presented a real-life example of a phone quality testing station explaining how the framework had been implemented successfully in a real life environment.

TABLE VIII

ESTIMATED RESPONSE FOR SELECTED DESIGN POINTS FROM RESPONSE SURFACE

Factor		Estimated Response, $\hat{y}$ (Throughput per 100s)
Fix	EQ1	
7	8	3.47
8	7	3.48
8	8	3.73

We also show how the simulation tool can be used for design parameter optimization. The current research results also suggest that much more needs to be done in the future. A screening experiment that captures some higher order effects and interactions might be more appropriate. As seen in the second phase of the optimization, only the non-linear terms seem to be significant; but for bigger problems seen in practice, like when also optimizing the equipment, bus and fixture mapping, selecting a large number of design points might lead to a large number of simulation runs that could require an impractically long total simulation time. Thus, a trade-off needs to be achieved for the number of design points for screening experiments and the order of the model used for analyzing it.

REFERENCES

[1] Jeffrey S. Smith, Yali Li, and Jason Gjesvold, “Simulation-based complex printed circuit board testing process,” in *Proceedings of the 2002 Winter Simulation Conference*, pp. 993–998.

[2] Mats Jackson, and Christer Johansson, “Real time discrete event simulation of a PCB production system for operational support,” in *Proceedings of the 1997 Winter Simulation Conference*, pp. 832–837.

[3] James J. Swain, and Phillip A. Farrington, “Designing simulation experiments for evaluating manufacturing systems,” in *Proceedings of the 1994 Winter Simulation Conference*, pp. 69–76.

[4] G. R. Drake, J. S. Smith, and B. A. Peters, “Simulation as a planning and scheduling tool for flexible manufacturing systems,” in *Proceedings of the 1995 Winter Simulation Conference*, pp. 805–812.

[5] Dennis C. Pegden, Robert E. Shannon, and Randall P. Sadowski, *Introduction to Simulation Using SIMAN* (2nd edition). Blacklick, OH: McGraw-Hill, Inc., 1995.

[6] C. F. Jeff Wu, and Michael Hamada, *Experiments: Planning, Analysis, and Parameter Design Optimization*. New York, NY: Wiley-Interscience, Inc., 2000.

[7] Douglas C. Montgomery, *Design and Analysis of Experiments* (5th edition). New York, NY: John Wiley, Inc., 2001.

[8] A. S. Hedayat, N. J. A. Sloane, and John Stufken, *Orthogonal Arrays: Theory and Applications*. New York, NY: Springer-Verlag, Inc., 1999.