Roser, Christoph, Masaru Nakano, and Minoru Tanaka. “Throughput Sensitivity Analysis Using a Single Simulation.” In Winter Simulation Conference, edited by Enver Yucesan, C. -H Chen, J. L Snowdon, and John M Charnes, 1087–94. San Diego, CA, USA, 2002.

# THROUGHPUT SENSITIVITY ANALYSIS USING A SINGLE SIMULATION

Christoph Roser
Masaru Nakano
Minoru Tanaka

Software Science Laboratory
Toyota Central Research and Development Laboratories
Nagakute, Aichi, 480-1192, JAPAN

## ABSTRACT

This paper describes a novel method of calculating the sensitivity of the manufacturing system throughput to the variables of the machines. The sensitivity analysis needs only a single simulation, yet is easy to use and provides accurate results. This sensitivity analysis is then used to predict the change in the system throughput due to a change of the variables of the machines provided that the system change does not significantly change the bottleneck. These predictions can be used for a local optimization, allowing the use of a steepest descent optimization algorithm. The method is based on improving the momentary shifting bottlenecks. The shifting bottlenecks are detected using the shifting bottleneck detection method based on the active duration, i.e., the time a machine is active without interruption. The method is easy to understand and easy to implement in existing simulation software.

## 1 INTRODUCTION

This paper describes a novel method for the sensitivity analysis of the throughput of manufacturing systems. The throughput is an important performance measure of these systems, also known as the system capacity or production rate, usually measured as a mean time between the completion of two parts, or as the number of parts produced in a certain time. In many manufacturing system optimizations, the goal is to improve the system throughput. However, the optimization of these systems is a complex task. For a summary of the vast literature on simulation optimization techniques please see (Andradottir 1998; Fu 2001; Swisher et al. 2000). Future trends and developments are discussed in (Boesel et al. 2001; Fu et al. 2000).

Many optimization methods are developed around a gradient estimator or a sensitivity analysis. The proposed method determines the sensitivity of the variables of the machines to the throughput using only a single simulation, allowing the use of gradient-based optimization methods. There are a number of gradient estimation approaches described in the literature. Perturbation analysis is a widely researched gradient estimation method (Ho and Cao 1991; Simmonds and Mann 1997). Unfortunately, this method is complicated to apply for complex systems, where the algorithm has to be redeveloped for each application. (Glynn 1990) and (Kleijnen and Rubinstein 1996) use a likelihood ratio estimator, having milder assumptions than perturbation theory, but also a possible higher variation. (Bettonvil and Kleijnen 1998) uses a method based on binary search techniques. A technique related to gradient-based methods is design of experiments and regression analysis (Schmidt and Launsby 1994),(Myers and Montgomery 1995). However, a large number of replications are needed to establish a valid model. Furthermore, the interpolated functions may behave differently than the true system, causing an optimization method to move away from the true optimum. (Law and Kelton 2000) also lists a number of references for sensitivity analysis and optimization. Overall, these methods are either too complex or require too many replications as desired by the industry, and therefore are used only infrequently. Rather, a method of “educated guesswork” is frequently applied to improve a manufacturing system.

The proposed method, however, uses a very intuitive and straightforward method to determine the effects of the variables of the machines on the throughput. Only a single simulation is needed to provide accurate and reliable results. The method focuses on the sensitivity of the variables of the machines to the throughput, and provides an easy to use and easy to implement sensitivity analysis method. As the throughput is based on the bottleneck(s) of the system, it is necessary to find the bottlenecks of the system in order to improve the throughput. The prediction of the method is valid as long as there is no significant change in the bottleneck of the changed system. This sensitivity analysis is based on and originates from the shifting bottleneck detection method (Roser, Nakano, and Tanaka 2001). Thus, the shifting bottleneck detection method will be explained below before the sensitivity analysis is discussed in detail.

## 2 SHIFTING BOTTLENECK DETECTION

The shifting bottleneck detection method determines the temporary bottleneck based on the duration the machines are active without interruption. This method is a continued development and improvement based on the method of the average active duration (Roser, Nakano, and Tanaka 2001), expanding the theory of constraints (Blackstone 2001; Goldratt 1992) into momentary and shifting bottlenecks (Lawrence and Buss 1994), (Moss and Yu 1999).

### 2.1 The Active Duration

The presented method is based on the duration a processing machine is active without interruption. A state is active whenever the machine may cause other machines to wait. For example working on one part may cause a subsequent idle machine to wait for the completion of the part, or a machine under repair may block previous machines. A state is inactive if the associated machine is not active but instead waiting for the completion of another task, for example the arrival of a part or service, or for the removal of a part. Table 1 shows a possible list of selected active and inactive states for different entities of a production system.

Table 1: Active – Inactive States for Different Machines

| Index | Description | Machine Type | Active |
|---|---|---|---|
| 1 | Working | Processing Machine | Yes |
| 2 | Starving | Processing Machine | No |
| 3 | Blocked | Processing Machine | No |
| 4 | Repaired | Processing Machine | Yes |
| 5 | Tool Change | Processing Machine | Yes |
| 6 | Moving to pickup | AGV | Yes |
| 7 | Moving to drop off | AGV | Yes |
| 8 | Waiting | AGV | No |
| 9 | Repaired | AGV | Yes |
| 10 | Recharging | AGV | Yes |
| 11 | Working | Factory Worker | Yes |
| 12 | Rest | Factory Worker | No |

Figure 1 shows an example of the active (work, repair, tool change) and inactive (waiting) states of one machine during a brief period of a simulation. The bottleneck detection method compares the durations of the active periods of the different machines.

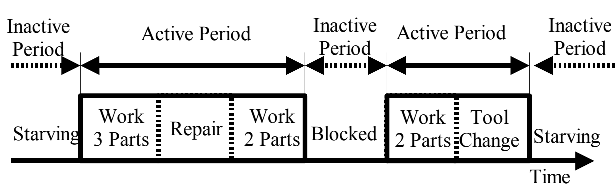


Figure 1: Active Periods of Machine During Simulation

### 2.2 The Momentary Bottleneck

The underlying idea of the method is that at any given time the machine with the longest uninterrupted active period is the momentary bottleneck at this time. In an interconnected production system, machines block and starve each other. If a machine is active, it is neither starved nor blocked. The longer a machine is active without interruption, the more likely it is that this machine blocks or starves other machines in the production system. The machine with the longest uninterrupted active period therefore has the biggest impact onto starving or blocking the other machines, therefore being the largest constraint a.k.a. the largest bottleneck. The overlap of the active period of a bottleneck with the previous or subsequent bottleneck represents the shifting of the bottleneck from one machine to another machine. During the shifting periods it is not entirely clear which machine is responsible for the limitation of the throughput, as either machine may be the bottleneck. The following method describes how to determine which machine of a production system is the sole bottleneck or part of a shifting bottlenecks at any time *t*.

If at time *t* no machines are active, then there is no bottleneck. If one or more machines are active at the time *t*, the machine with the longest active period at the time *t* is the momentary bottleneck machine, and the active period of this machine is the current bottleneck period. If the current bottleneck period ends, it is necessary to find the next bottleneck by determining the machine with longest active period after the current bottleneck period ended. The shifting of the bottleneck from the current bottleneck machine to the subsequent bottleneck machine happens during the overlap of the current and the subsequent bottleneck periods. During the overlaps between the bottleneck periods no machine is the sole bottleneck, instead the bottleneck shifts between the two machines. If a bottleneck machine is not shifting, then this machine is the sole and only bottleneck at this time.

Using this method, it can be determined at any given time if a machine is a non-bottleneck, a shifting bottleneck, or a sole bottleneck. This method allows the detection of the momentary bottleneck, where and when the previous bottleneck was shifting to the current bottleneck, and where and when the current bottleneck is shifting to the next bottleneck.

Figure 2 illustrates the method using a simple example consisting of only two machines. The figure shows the active periods of the machines over a short period of time. At the selected time *t*, both machines M1 and M2 are active. Yet, as M1 has the longer active period, M1 is the bottleneck machine for the time *t*. At the end of the bottleneck period, M2 is active and has the longest active period. Therefore the subsequent bottleneck machine is M2. During the overlap between the current bottleneck period and the subsequent bottleneck period the temporary bottleneck shifts from M1 to M2. Now, M2 is the bottleneck machine. Processing all available data using this method shows at what time which machine is the momentary bottleneck machine, when the bottleneck is shifting, and when there is no bottleneck at all. Therefore it is possible to detect and monitor the momentary bottleneck at all times.

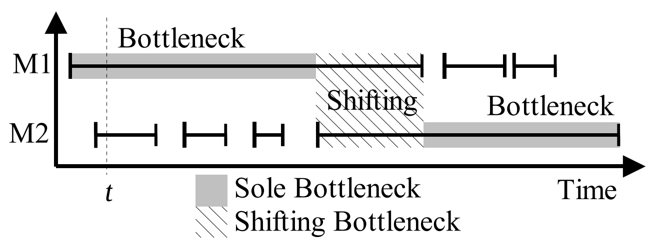

Figure 2: Shifting Bottlenecks

The shifting bottleneck detection method allows the detection and monitoring of the momentary bottleneck throughout the simulation. The shifting bottleneck detection method can also be expanded to evaluate the probability of a machine being a bottleneck. However, for the sensitivity analysis it is only necessary to know when a machine is the sole or shifting bottleneck, allowing a detailed analysis of the variables of the machines during the bottleneck periods.

## 3 SENSITIVITY ANALYSIS

The shifting bottleneck detection method as described above determined the sole and shifting bottlenecks at any given time during the simulation. The sensitivity analysis enhances this approach by analyzing the events of which the bottleneck periods consist of. Figure 3 shows the detailed per event analysis of the example used in Figure 2. The example includes three types of events, namely machine M1 working, machine M2 working, and machine M2 under repair. Each of the active periods shown in Figure 2 consists of one or more of these events. The sole and shifting bottleneck periods are underlined grey and hatched respectively, while non-bottleneck periods are greyed out, as they do not affect the throughput.

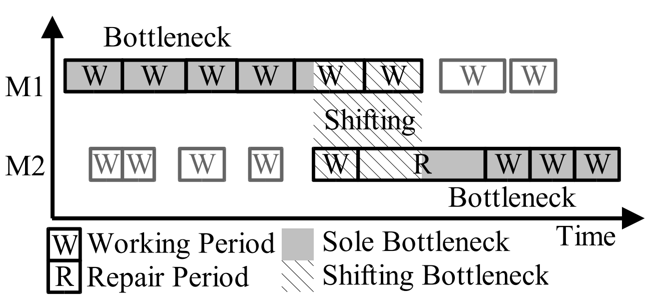

Figure 3: Bottleneck Events

The bottleneck periods limit the overall manufacturing system throughput, and the bottleneck periods consist of the different actions of the machines. Therefore, the actions of the bottleneck machines during the bottleneck periods determine the overall system throughput. Knowing the sole and shifting bottleneck periods and the events therein, the percentage contribution of the variables of the machines to the throughput can be calculated easily. Equation (1) shows the calculation of the percentage effect of state *j* of machine *i* due to the sole $P_{i,j}^{Sole}$ and shifting $P_{i,j}^{Shifting}$ bottleneck, where the time *t* is integrated if machine *i* is both in state *j* and the sole or shifting bottleneck respectively and divided by the total analyzed time, defined by the starting and ending times $t_{Start}$ and $t_{End}$. For examples of the different states, please refer to Table 1.

$$P_{i,j}^{Sole} = \frac{1}{t_{End} - t_{Start}} \cdot \int_{t_{Start}}^{t_{End}} \begin{cases} 1 & \begin{cases} \text{if } Status(i) = j \\ \text{and } i = Sole\,Bottleneck \end{cases} \\ 0 & else \end{cases}$$

$$P_{i,j}^{Shift} = \frac{1}{t_{End} - t_{Start}} \cdot \int_{t_{Start}}^{t_{End}} \begin{cases} 1 & \begin{cases} \text{if } Status(i) = j \\ \text{and } i = Shifting\ Bottleneck \end{cases} \\ 0 & else \end{cases} \qquad (1)$$

Figure 4 shows the percentages of the time each of the three events was a sole or shifting bottleneck for the example shown in Figure 3. Machine M1 working contributed with 45% sole and 20% shifting bottlenecks the largest part of all sole and shifting bottleneck periods, and therefore has the largest effect onto the throughput. Machine M2 working and machine M2 repair contributed smaller percentages, and therefore the throughput is less sensitive to these two variables. These values represent the relative effect of a change in the variables towards the overall throughput. For example if machine M1 Working would be improved by a small amount, between 45 and 65% of this improvement would benefit the overall system throughput. Therefore, these sensitivity values allow the prediction of the system performance of a changed system as described in the next section.

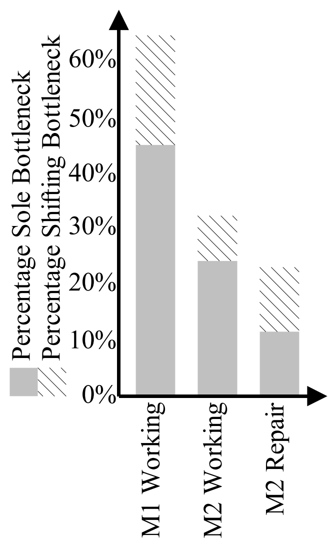


Figure 4: Sensitivity Analysis

## 4 PERFORMANCE PREDICTION

The sensitivity analysis using the shifting bottleneck method determines the percentage effect of each machine state (working, repair, tool change,) onto the overall throughput. This allows the prediction of the effect of a change in a machine variable (working time, repair time, tool change time …) onto the overall throughput. Note that the method distinguishes between the effect due to sole bottlenecks and due to shifting bottlenecks. A sole bottleneck is the only bottleneck at this time in the system, and an improvement of the sole bottleneck events will improve the throughput. However, if there is a shifting bottleneck, then it is not sure which machine actually is the true bottleneck, and an improvement of the shifting bottleneck events may or may not improve the overall throughput. Therefore, the lower and upper limits $\Delta P_{Low}$ and $\Delta P_{High}$ of the expected percentage change of the system performance can be calculated based on the percentage change of the state *j* of machine *i* for all machine variables $P_{i,j}^{Change}$ and the effects $P_{i,j}^{Sole}$ and $P_{i,j}^{Shift}$ of the variables of the machines as shown in equation (2). The change of the state $j$ of machine $i$ $P_{i,j}^{Change}$ represents the improvement of the machine within this state, e.g., if the original system produced in average one part every 100 seconds, then an improved system requiring only 80s per part would represent a $P_{i,j}^{Change}$ to 80% of the previous value.

$$\Delta P_{Low} = \sum_i \sum_j P_{i,j}^{Change} \cdot P_{i,j}^{Sole}$$
$$\Delta P_{High} = \sum_i \sum_j P_{i,j}^{Change} \cdot \left(P_{i,j}^{Sole} + P_{i,j}^{Shift}\right) \qquad (2)$$

This method is best explained using a numerical example. Assume that machine variable M1 working contributes 65% of the sole bottlenecks and an additional 20% of the shifting bottlenecks, and the overall system has an average production rate of one part every 100s. Therefore, between 65s and 85s of the average time between parts are due to M1 working Reducing the working time of M1 to 90s, i.e., a $P_{i,j}^{Change}$ of 10% would reduce the overall bottleneck periods at least 10% * 65% = 6.5% (Effect of sole bottlenecks $P_{i,j}^{Sole}$) and a possible additional 10% * 20% = 2% (Effect of shifting bottlenecks $P_{i,j}^{Shift}$). Therefore the overall reduction of the time between parts would be between $\Delta P_{Low}$ =6.5% and $\Delta P_{High}$ =8.5% as shown in Equation (3).

$$\Delta P_{Low} = 10\% \cdot 65\% = 6.5\%$$
$$\Delta P_{High} = 10\% \cdot (65\% + 20\%) = 8.5\% \qquad (3)$$

The subsequent expected average production rate would be between 91.5s and 93.5s for each part. Therefore using equation (2) allows the rapid calculation of the throughput of a large number of alternative design changes based on the sensitivity analysis of the original manufacturing system.

However, one shortcoming of sensitivity analysis and gradient-based methods in general is that they are only strictly true at the system for which the sensitivity has been measured. As the system variables change, the system changes, and subsequently the sensitivity changes. The larger the system changes the larger the uncertainty of the prediction. This is illustrated in Figure 5 for the above example. As the machine M1 working contributes between 65 and 85% of the bottlenecks, a reduction of the working time of M1 to zero would theoretically reduce the mean time between parts by 65 to 85%. However, it is to be expected that as the working time of M1 decreases, M1 becomes less likely to be a bottleneck and other machines will become a bottleneck, and the true performance improvement will be less than the expected performance improvement for larger changes.

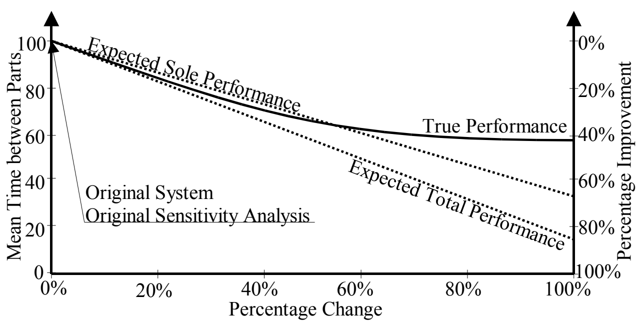


Figure 5: Performance Prediction

This sensitivity analysis and performance prediction can then be used to form the base of a manufacturing system optimization to allow the rapid evaluation of manufacturing system alternatives for a local optimization. For an overview of optimization techniques please see (Nemhauser, Rinnooy Kan, and Todd 1994) for general optimization techniques, and (Andradottir 1998; Fu 2001; Swisher et al. 2000) for simulation optimization methods.

## 5 VERIFICATION

The sensitivity analysis and prediction methods have been verified using a complex simulation example, consisting of a branched manufacturing system with seven machines and two different part types as shown in Figure 6. The buffer size for the different machines ranges from zero (no buffer at all) to five, depending on the buffer location. The simulation was performed using the GAROPS simulation software as shown in (Kubota, Sato, and Nakano 1999) and (Nakano et al. 1994). The method was implemented in an automatic software tool GAROPS ANALYZER for analyzing the log files of the GAROPS simulation software and automatically creating a report of the simulation performance data in MS Excel. The simulation time was 600 days to ensure sufficient accuracy of the results. The average time between the production of two parts was 54.0s, or about 66.7 parts per hour.

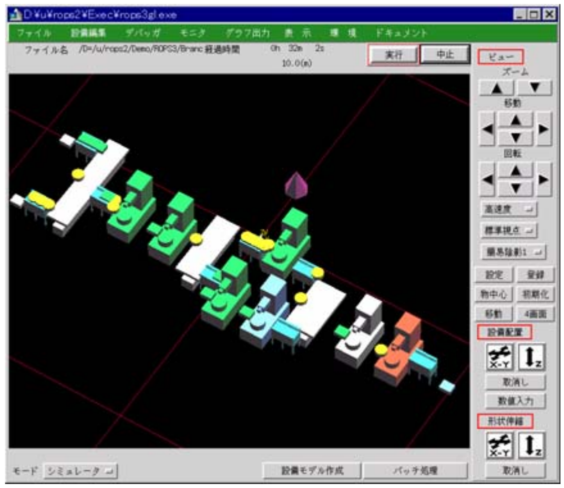

Figure 6: Simulation Model

### 5.1 Sensitivity Analysis

The initial simulation was analyzed and the sensitivity of the throughput to the variables of the machines was determined. Table 2 shows the results of the throughput sensitivity analysis for all machine variables. The effect due to the sole and shifting bottlenecks and the total effect is shown. The values are sorted according to the total effect. Figure 7 shows the results in graphical form for the six largest effects. It shows clearly, that the working rate of machine M2 has the largest effect onto the throughput, with a relative effect between 68% (sole bottlenecks) and 87% (sole and shifting bottlenecks). All other machine working and repair times have only minor effects of 10% or less. Therefore, in order to improve the throughput of the manufacturing system the working time of machine M2 has to be improved.

Table 2: Throughput Sensitivity Analysis Results

| **Name** | **Sole** | **Shifting** | **Total** |
|---|---|---|---|
| M2 Working | 67.78% | 18.85% | 86.63% |
| M7 Working | 2.51% | 7.56% | 10.06% |
| M3 Working | 1.42% | 7.25% | 8.67% |
| M5 Working | 0.70% | 4.82% | 5.52% |
| M2 Repair | 3.67% | 0.79% | 4.45% |
| M7 Repair | 1.29% | 1.87% | 3.17% |
| M6 Repair | 0.66% | 0.68% | 1.35% |
| M5 Repair | 0.41% | 0.68% | 1.09% |
| M3 Repair | 0.19% | 0.73% | 0.92% |
| M6 Working | 0.00% | 0.18% | 0.18% |
| M4 Repair | 0.01% | 0.03% | 0.04% |
| M4 Working | 0.00% | 0.02% | 0.02% |
| M1 Working | 0.00% | 0.00% | 0.00% |

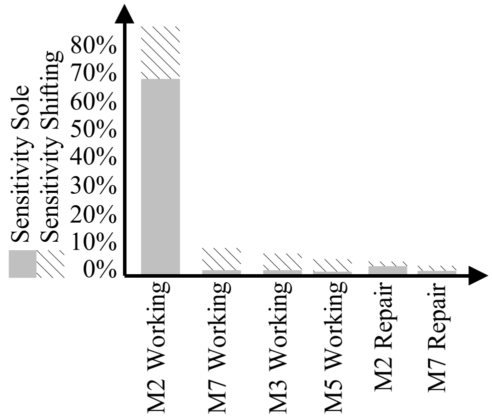

Figure 7: Throughput Sensitivity Analysis Results

### 5.2 Performance Prediction Verification

The performance prediction has been verified by comparing the predicted performance with the actual measured simulation performances for a number of changed alternative designs. The working time of machine M2 had the largest effect on the throughput of between 68% and 87%. Therefore, a reduction of the working time of M2 by 5% would according to equation (2) improve the time between parts between 3.4 and 4.3%. For an initial time between parts of 54.0s, the expected mean time between parts of the changed system was predicted to be between 51.7 and 52.2s. The expected improvement has been verified, with the actual mean time between parts of the improved system being between 52.1 and 52.3s, where the range of the actual improvement is based on the 95% confidence interval of the verification simulation. This is illustrated in Figure 8, where the predicted improvement of 3.4 ~ 4.3% is compared to the actual improvement of 3.1 ~ 3.5%. It can be seen, that the predicted performance and the actual performance matches very well.

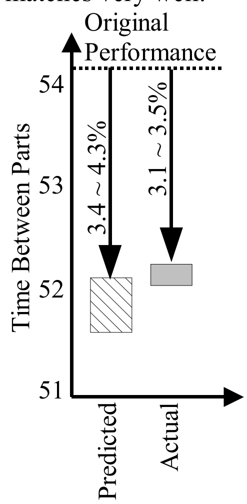

Figure 8: Change M2 Working Predicted vs. Actual Results

Table 3 and Figure 9 show the predicted and the actual results of the verified variables. The change in M2 working time has been described above. M7 working and M3 working are the variables with the second- and third most effect. The range of the prediction of a 10% change of M7 working fits the actual measured performance change very well. For M3 working the prediction is also quite close to the actual measured results.

Table 3: Predicted vs. Actual Change

| | **Predicted** | | **Actual** | |
|---|---|---|---|---|
| **Variable Change** | **Low** | **High** | **Low** | **High** |
| M2Work-5% | 3.39% | 4.33% | 3.08% | 3.48% |
| M7Work-10% | 0.25% | 1.01% | 0.55% | 1.00% |

| | | | | |
|---|---|---|---|---|
| M3Work-10% | 0.14% | 0.87% | -0.23% | 0.22% |
| M2Repair-10% | 0.37% | 0.45% | 0.23% | 0.68% |
| M4Working-90% | 0.00% | 0.02% | -0.30% | 0.15% |

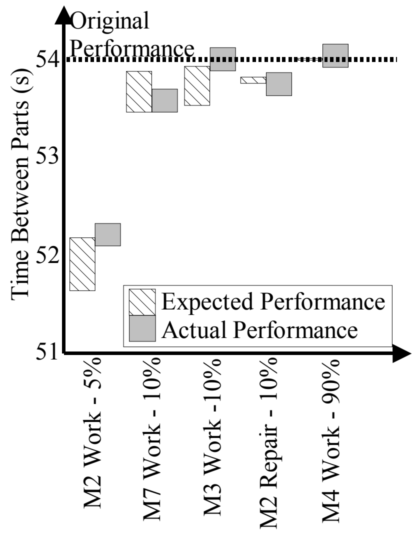


Figure 9: Predicted vs. Actual Results

Additionally, M2 repair time has also been changed by 10% to include the effect of a less frequently occurring event. In this case, the prediction and the actual performance are also a very good match. Finally, the working time of M4 has been changed by 90% to verify that a variable with an extremely small effect according to the sensitivity analysis (a total of 0.0023%) indeed does not affect the overall system performance. Overall, the predicted changes and the actual changes match very well, indicating that the performance prediction is valid for small changes.

### 5.3 Prediction Accuracy

Theoretically, a sensitivity analysis is only valid for the analyzed system. If the system changes, the sensitivity may change, too. With respect to the manufacturing system, the sensitivity analysis determines the effect of the machines onto the throughput, where the machine with the main effect constitutes the bottleneck. If for example, the manufacturing system is improved by improving the main bottleneck, another machine may become the main bottleneck, i.e., have the largest effect on the throughput, and the actual performance improvement may be less than the predicted performance improvement.

In the presented example, machine M2 working has the largest effect onto the throughput of 68 ~ 87% as shown in Table 2, and machine M2 is the main bottleneck. Theoretically, if the working time of machine M2 is set to zero, the time between parts would improve by 68 ~ 87%. Practically, of course, another machine becomes the main bottleneck and the improvement is less than expected.

The change of the performance of the system due to a change in the working time of machine M2 has been predicted and measured for a wide range of changes from 0% to 40%. Figure 10 compares the range of the expected performance with the measured true performance of the system. For small changes, the predicted time between parts is very close to the measured mean time between parts. However, as the working time of machine M2 is improved, machine M2 is less and less likely to be the bottleneck. Therefore, the actual change becomes less than the predicted change, until a further improvement has no effect on the system performance.

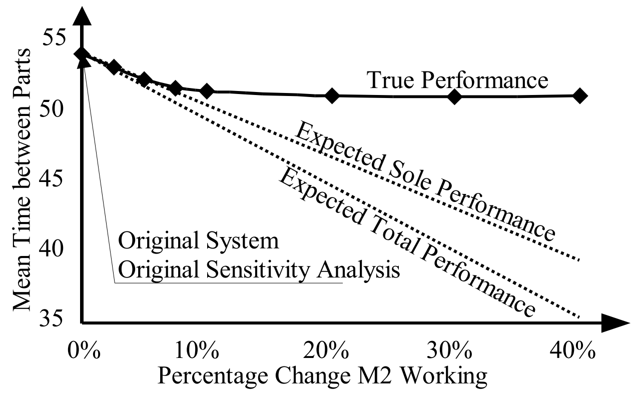

Figure 10: Prediction Error

This can also be seen in Figure 11. Figure 11 shows the results of the sensitivity analyses for an improved working time of M2. While at the beginning M2 working is the main effect, this effect gradually decreases as the working time of M2 is improved. Instead, the working time of machine M3 becomes increasingly significant, until the working time of M3 is the main effect, and the working time of M2 is all but insignificant. This can also be compared to Figure 10, where the prediction becomes less accurate as the effect of M2 working changes. In summary, the main bottleneck gradually changes from machine M2 to M3, with the two machines having an equal effect if M2 working is reduced by approximately 10%.

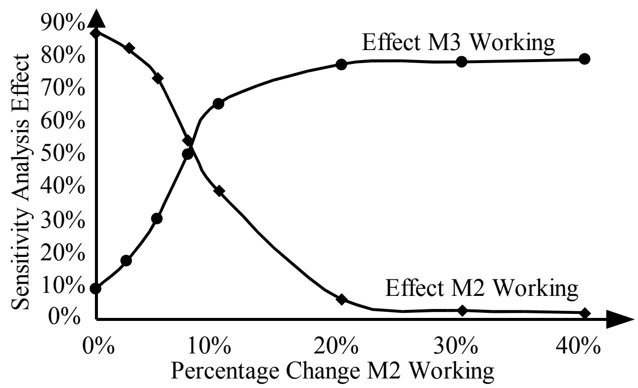

Figure 11: Changing Bottleneck

Note that the switchover point may be different for different systems, depending on how fast another machine becomes the main bottleneck. Subsequently, the predictions are only valid within the local area of the analyzed system. Therefore, in order to use the prediction for a manufacturing system optimization, it is necessary to reevaluate the system as the optimization moves away from the initial system design.

## 6 SUMMARY

In summary, the above sensitivity method is able to accurately detect the effect of the variables of the machines onto the throughput of the manufacturing system using only a single simulation. The method is very intuitively and easy to understand, and the mathematical analysis is straightforward and reliable, giving clear results. Using the sensitivity analysis, it is possible to make predictions of the system performance based on changes in the variables of the machines. This allows a fast and easy search for local optima's as part of an optimization of the manufacturing system.

Further research includes the sensitivity analysis of variables other than machine variables as for example buffer sizes. In addition, the change of the main bottleneck will be researched in more detail in order to predict when and where the bottleneck will change in response to a system change, allowing a more accurate prediction over a wider range of the system variables.

## REFERENCES


Andradottir, S. 1998. A Review of Simulation Optimization Techniques. In *Winter Simulation Conference*, ed., 151-158, Washington, DC, USA.

Bettonvil, B., and Kleijnen, J. P. C. 1998. Identifying the Important Factors in Simulation Models with Many Factors. In *Winter Simulation Conference*, ed., 713-718, Washington, DC, USA.

Blackstone, J. H. 2001. Theory of constraints - a status report. *International Journal of Production Research*, 39(6), 1053-1080.

Boesel, J., Jr., R. O. B., Glover, F., Kelly, J. P., and Westwig, E. 2001. Future of Simulation Optimization. In *Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 1466-1469, Arlington, Virginia, USA: Institute of Electrical and Electronics Engineers.

Fu, M., Andradottir, S., Carson, J. S., Glover, F., Harrell, C. R., Ho, Y.-C., Kelly, J. P., and Robinson, S. M. 2000. Integrating Optimization and Simulation: Research and Practice. In *Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. R. Barton, 610-615, Orlando, Florida, USA.

Fu, M. C. 2001. Simulation Optimization. In *Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, Arlington, Virginia, USA: Institute of Electrical and Electronics Engineers.

Glynn, P. W. 1990. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10), 75-84.

Goldratt, E. M. 1992. *The Goal: A Process of Ongoing Improvement.* North River Press.

Ho, Y.-C., and Cao, X.-R. 1991. *Perturbation Analysis of Discrete Event Dynamic Systems.* Kluwer Academic Publisher.

Kleijnen, J. P. C., and Rubinstein, R. Y. 1996. Optimization and sensitivity analysis of computer simulation models by the score function method. *European Journal of Operational Research*, 88(3), 413-427.

Kubota, F., Sato, S., and Nakano, M. 1999. Enterprise Modeling and Simulation Platform Integrated Manufacturing System Design and Supply Chain. In *IEEE Conference on Systems, Man, and Cybernetics*, ed., 511-515, Tokyo, Japan.

Law, A. M., and Kelton, D. W. 2000. *Simulation Modeling & Analysis.* McGraw Hill.

Lawrence, S. R., and Buss, A. H. 1994. Shifting Production Bottlenecks: Causes, Cures, and Conundrums. *Journal of Production and Operations Management*, 3(1), 21-37.

Moss, H. K., and Yu, W. B. 1999. Toward the Estimation of Bottleneck Shiftiness in a Manufacturing Operation. *Production and Inventory Management Journal*, 40(2), 53-58.

Myers, R. H., and Montgomery, D. C. 1995. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments.* New York: John Wiley & Sons.

Nakano, M., Sugiura, N., Tanaka, M., and Kuno, T. 1994. ROPSII: Agent Oriented Manufacturing Simulator on the basis of Robot Simulator. In *Japan-USA Symposium on Flexible Automation*, ed., 201-208, Kobe, Japan.

Nemhauser, G. L., Rinnooy Kan, A. H. G., and Todd, M. J. 1994. *Optimization.* Amsterdam: Elsevier Science.

Roser, C., Nakano, M., and Tanaka, M. 2001. A Practical Bottleneck Detection Method. In *Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 949-953, Arlington, Virginia, USA: Institute of Electrical and Electronics Engineers.

Schmidt, S. R., and Launsby, R. G. 1994. *Understanding Industrial Designed Experiments.* Colorado Springs, Colorado: Air Academy Press.

Simmonds, J. G., and Mann, J. E. J. 1997. *A first look at perturbation theory.* Dover Publications.

Swisher, J. R., Hyden, P. D., Jacobson, S. H., and Schruben, L. W. 2000. A Survey of Simulation Optimization Techniques and Procedures. In *Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. R. Barton, 119-128, Orlando, Florida, USA.


## AUTHOR BIOGRAPHIES


**CHRISTOPH ROSER** is a researcher at the Toyota Central Research and Development Laboratories, Nagoya, Japan. Current research topics are simulation output analysis, bottleneck detection and risk and sensitivity analysis. He graduated from the Fachhochschule Ulm, Germany with a diploma in automation engineering, and received his Ph.D. in Mechanical Engineering at the University of Massachusetts, Amherst, studying flexible and robust design methods. His email address is <croser@robotics.tytlabs.co.jp>

**MASARU NAKANO** is a senior researcher at Toyota central Research and Development Laboratories, Inc. (TCRDL) in Japan. He joined TCRDL in 1980 and since then has studied in the field of robotics, computer vision, and manufacturing system design. He is currently research leader and manager of the Digital Engineering Research Domain. He received his B.S. and M.S. in operations research from Kyoto University, and his Dr. Eng. in manufacturing systems from the Nagoya Institute of technology. His email address is <nakano@robotics.tytlabs.co.jp>

**MINORU TANAKA** is an associate researcher at the Toyota Central Research and Development Laboratories, Nagoya, Japan. He graduated from Matsue Technical College, joined TCRDL in 1985, and specialized in the field of robotics, manufacturing system modeling, and discrete event simulation. He is a member of The Robotics Society of Japan. His email address is
<tanaka@robotics.tytlabs.co.jp>