Sensitivity Analysis of a State Variable Model of the Software Test Process

João W. Cangussu[1], Ray A. DeCarlo and Aditya P. Mathur[2]

Department of Computer Sciences, Purdue University
Department of Electrical and Computer Engineering, Purdue University
Department of Computer Sciences, Purdue University
West Layette, IN, USA, 47907-1398.

Abstract
This paper reports results of a sensitivity analysis of a state variable model of the Software Test Process (STP). Given a state model of the STP, a sensitivity matrix is calculated using tensor algebra. The sensitivity matrix allows computation of output variations to small perturbations in the model parameters. The results confirm that the model behaves in a manner very similar to what one might expect from a software test process. Results of this analysis also suggest changes and enhancements in the model to improve its accuracy in predicting the behavior of the software test process.

keywords: Software process, feedback control, sensitivity analysis, modeling, software test process, state model.

1 Introduction

The use of state variable methods [1, 2] to model and control the system test phase of the Software Test Process (STP) has shown strong promise and applicability [3, 4]. The model, referred to here as `Model S`, was applied with encouraging accuracy to the control of a commercial STP [4]. A distinguishing characteristic of `Model S` is the presence of a feedback control loop similar to the one found in a large number of engineering applications such as in cruise control, rolling mill control, and satellite tracking. By evaluating response curves with respect to deadlines, etc., one can use parametric control to adjust decay rates to better meet management objectives.

The use of a sound mathematical approach to modeling and the presence of the feedback control loop distinguishes our approach to the control of the STP from a multitude of other models and approaches such as those found in Total Quality Management and the ANSI standard for software V &V plans.[5, 6]

Obviously, the accuracy of any model, such as `Model S`, depends on the goodness of parameter estimates. Hence it is critical to understand how sensitive is the behavior of `Model S` to changes in STP parameters. We therefore conducted a sensitivity analysis of `Model S` by first computing a sensitivity matrix and then using this matrix to study the model behavior to small variations in parameters.

The remainder of this paper is organized as follows. Section 2 is an overview of `Model S`. The method used to compute the sensitivity matrix for the STP model is described in Section 3. A discussion of the results obtained from computations done with the sensitivity matrix appears in Section 3.2. Section 4 summarizes this work and our conclusions.

2 Overview of the State Variable Model of the STP

A linear model of the STP is based on three assumptions. The assumptions and the corresponding equations are presented below [4, 7]. Note that these assumptions are based on an analogy of the STP with the physical process typified by a spring-mass-dashpot system. The analogy is outside the scope of this paper and is found in [4].

Assumption 1: The magnitude of the rate at which the remaining errors are decreasing is proportional to the net applied effort for the test

[1]Financial support by CAPES and UFMS
[2]Financial support in part by SERC and NSF

phase and inversely proportional to the software complexity.

$$\ddot{r} = \frac{e_n}{s_c} \quad \Rightarrow \quad e_n = \ddot{r}\ s_c \tag{1}$$

This is analogous to $F = m \times a$ where s_c is analogous to m.

Assumption 2: The magnitude of the effective test effort is proportional to the product of applied work force and the number of remaining errors, i.e., for an appropriate ζ.

$$e_{et} = \zeta\ w_f\ r \tag{2}$$

Here the effective test effort is seen to be a fraction (ζ) of the possible encounters between a member of the test team and an error ($w_f \times r$).

Assumption 3: The error reduction resistance is proportional to the error reduction velocity and inversely proportional to the overall quality of the test phase, for an appropriate constant ξ.

$$e_r = \xi\ \frac{1}{\gamma}\ \dot{r} \tag{3}$$

Combining Eqs. 1, 2, and 3 and organizing in a State Variable format ($\dot{x} = Ax + Bu$) produces the following system.

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{\zeta\ w_f}{s_c} & -\frac{\xi}{\gamma\ s_c} \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{s_c} \end{bmatrix} F_d \tag{4}$$

where F_d represents unforeseen disturbances such as hardware failures.

3 Sensitivity Analysis

The first step is to construct a sensitivity matrix for `Model S`. In the second step the sensitivity matrix is used as a computational tool to investigate the sensitivity of the number of remaining errors in the software product to each of the five parameters in the state model of the STP.

3.1 Computation of the sensitivity matrix

For a sensitivity analysis with respect to parameter variations, F_d is identically zero. Our objectives then is to characterize variations in the trajectories $x(t)$ with respect to variations in the five parameters of the 2×2 A matrix in Eq. 4. A vector G of parameters that characterize the STP in `Model S` is defined as

$$G_{p\times q} = \begin{bmatrix} s_c \\ \xi \\ \zeta \\ w_f \\ \gamma \end{bmatrix} \qquad for\ p = 5\ and\ q = 1 \tag{5}$$

Considering the state variables $x(t) = [r(t)\ \dot{r}(t)]^T$ and G the vector of parameters, the sensitivity of x to changes in G is denoted by $\frac{\partial x(t)}{\partial G}$. This derivative is the sensitivity matrix defined by Brewer [8] as:

$$\frac{\partial x(t)}{\partial G} = \frac{\partial \Phi}{\partial G}(I_q \otimes x(0)) + I_p \otimes \Phi \frac{\partial x(0)}{\partial G} + \int_0^t \left[\frac{\partial \Phi(t-\tau)}{\partial G}(I_q \otimes BF_d(\tau)) + (I_p \otimes \Phi)\frac{\partial}{\partial G}(BF_d(\tau)) \right] d\tau \tag{6}$$

where $\Phi = e^{At}$ is the state transition matrix and I_m is an $m \times m$ identity matrix. The symbol $\otimes$ represents the Kronecker product [9].

The last two terms of Eq. 6 are zero. The integral term is zero because we considered no perturbations ($F_d = 0$) and the second term is zero because the initial condition ($x(0)$) is not dependent on any parameter of G and hence the partial $\frac{\partial x(0)}{\partial G}$ is zero.

Let $x(0) = [r_0\ v_0]^T$ be the initial condition. Thus, the term $(I_q \otimes x(0))$ reduces to x(0). Next we need to compute $\frac{\partial \Phi}{\partial G}$. This is done by the use of Eq. 7 below [8].

$$\frac{\partial \Phi}{\partial G} = \sum_{i=0}^{n-1} \sum_{h=0}^{n-1} (I_p \otimes A^i)\ \frac{\partial A}{\partial G}\ (I_q \otimes A^h)\ q_{ih}(t) \tag{7}$$

The terms $(I_p \otimes A^i)$ and $(I_q \otimes A^h)$ can be computed easily. The derivative $\frac{\partial A}{\partial G}$ results in a 10×2 matrix consisting of five 2×2 partitions. Each partition $(i-k)$ is defined as $\frac{\partial A}{\partial g_{ik}}$, where g_{ik} is an element of vector G.

The term $q_{ih}(t)$ is defined as the inverse Laplace transform of $\beta_i(s)\beta_h(s)$ where

$$\beta_k(s) = \sum_{r=0}^{n-k-1} \frac{\Delta_{k+r+1} s^r}{\Delta(s)} \tag{8}$$

and where $\Delta(s)$ is the characteristic polynomial of A defined as

$$\Delta(s) = \sum_{j=0}^{n} \Delta_j \; s^j = det\,[sI \; - \; A] \qquad (9)$$

Computation of Eq. 6 generates a 10×1 matrix with 5 partitions of 2×1 matrices.

Rearranging the matrix produced by Eq. 6 in a 2×5 format results in the sensitivity matrix denoted here by SM. The matrix SM can now be used to compute variations in the state variables in response to perturbations in specific parameters or combinations thereof as per Eq. 10:

$$\begin{bmatrix} \Delta r(t) \\ \\ \Delta \dot{r}(t) \end{bmatrix} \quad \cong \quad SM|_{G_{nom}} \times \Delta_m \qquad (10)$$

where $\Delta_m = [\Delta s_c \; \Delta\xi \; \Delta\zeta \; \Delta w_f \; \Delta\gamma]^T$ and G_{nom} is the nominal parameter vector.

The variations introduced in one or more parameters are represented in the vector Δ_m and will reflect changes in $r(t)$ and $\dot{r}(t)$ observed by the variations in Δ_r and $\Delta_{\dot{r}}$.

3.2 Using the sensitivity matrix

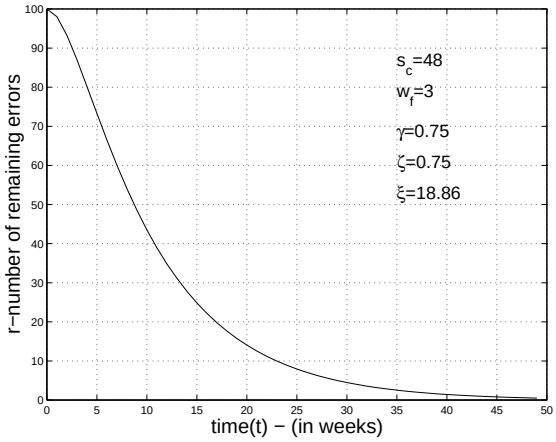

Figure 1: Expected behavior of the process under consideration.

In this section we use the sensitivity matrix to analyze the sensitivity of the state variables of `Model S` to variations in its parameters. Unless stated otherwise, the analysis is conducted using the parameters characterizing the STP of the *transformer* project conducted at Razorfish [4]. The values of the parameters are: $s_c = 48$, $w_f = 3$, $\gamma = 0.75$, $\zeta = 0.75$ and $\xi = 18.86$. The behavior of such process, characterized by the normalized number of remaining errors[3] over time and computed using `Model S`, is depicted in Figure 1.

To obtain the variations for individual or combined changes in parameters we need to set Δ_m to the desired value. For example, to analyze the effects of a 5% change in s_c and a -10% change in γ we only need to set Δ_m to $[0.05 s_c \; 0 \; 0 \; 0 \; -0.1\gamma]^T$. The results are a function of $\Delta r(t)$. $\Delta r(t') > 0$ at time $t = t'$ implies an increase in the number of remaining errors causing a decrease in the decay rate of errors indicating a slow down in the process. $\Delta r(t') < 0$ at time $t = t'$ implies a decrease in the number of remaining errors (as expected) causing an increase in the decay rate and an acceleration in the process.

Sensitivity of Δr to changes in w_f

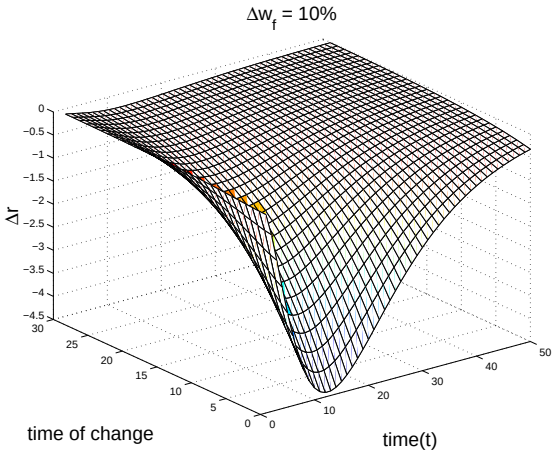

Figure 2: Sensitivity of the change in remaining errors (Δr) to a 10% change in w_f.

Figure 2 shows the variation of the fraction of remaining errors (Δr) in the presence of a 10% (Δw_f) variation of the nominal w_f. To obtain the results in Figure 2, we set the matrix Δ_m zero in all positions except Δw_f which is set to $\Delta w_f = 0.1 \; w_f$. As is observed from Figure 2, increasing w_f by a factor $\alpha > 1$ does not decrease $r(t)$ by α. A further examination of Figure 2 reveals that changing w_f at the beginning of the test phase is much more effective than changing it near the end, as expected again. This consistency suggests a high confidence level in the accuracy of the model.

Equivalently, as the number of remaining errors reduces, an increase in w_f is not as effective as it is at the beginning of the process. Indeed, if communication overhead were accounted for in

[3]The actual number of remaining errors is proprietary data of Razorfish.

our model, an increase in w_f close to the end of the process would delay the realization of the STP quality objective. Although not yet explicitly modeled, communication overhead and learning can be simulated by increasing w_f and decreasing the quality of the process (γ). As is observed from Figure 3 these changes will cause a decrease in the rate of decay and consequently delay the realization of the STP objective. The delay becomes increasingly acute as the STP gets closer to its end. The behavior of `Model S` observed from the sensitivity analysis conforms to the long held belief popularly known as the Brook's Law. [10].

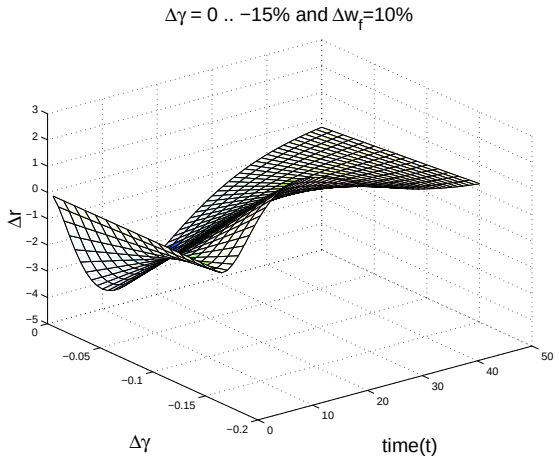

Figure 3: Sensitivity of the change in remaining errors (Δr) to a 10% change in w_f and a 0 to -15% change in γ. The negative change in the process quality (γ) represents the communication overhead.

Sensitivity of Δr to changes in s_c

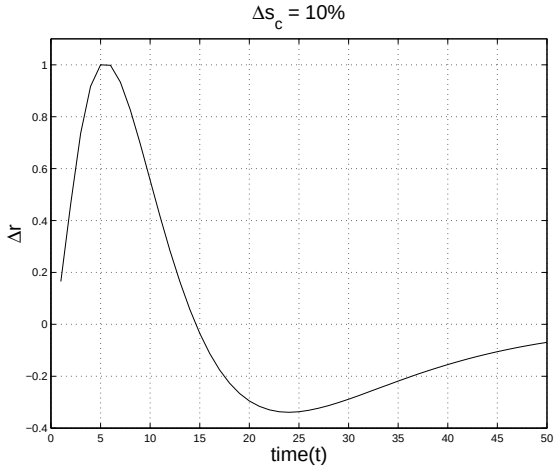

Figure 4: Sensitivity of the change in remaining errors (Δr) to a 10% change in s_c.

An increase of 10% in s_c represents only an increase of at most 1% in the number of remaining errors as can be observed in Figure 4. Although presenting the correct behavior (slow down) at the beginning, a speed up in the process is observed after 15 weeks. This behavior is unexpected. Certainly, increases in software complexity intuitively lead to greater difficulty in identifying and fixing errors, thereby slowing down the test process. One can think of an increase in software complexity as an increase in "frictional forces" opposing testing. This type of frictional force is not directly accounted for in `Model S`. Hence the sensitivity analysis suggests that `Model S` [4] needs to more fully utilize s_c in the underlying assumptions. This is an important consequence of the sensitivity analysis because s_c is presumed to be set without any error. In practice when two or more metrics are combined in a convex combination as suggested in [7], a range of values for s_c is possible for a specific test product. For example, suppose, for the same product, s_c is computed using KLOC(s_c1) and the average Cyclomatic Complexity(s_c2) [11] and that $s_c1 = 48$ and $s_c2 = 43.2$. Thus `Model S` must better incorporate s_c and be able to predict a realistic sensitivity with respect to changes in s_c.

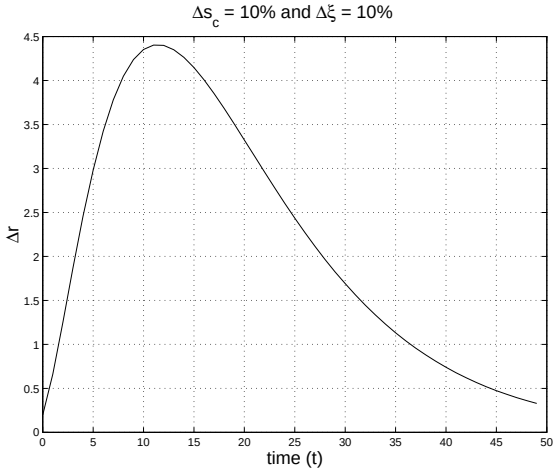

Figure 5: Sensitivity of the change in remaining errors (Δr) to a 10% change in s_c and a 10% change in ξ.

Nevertheless, within the context of `Model S`, it is possible to simulate an increased frictional force by either reducing the effective work force or by increase ξ. Such a simulation for example, i.e., a 10% increasing in s_c with a 10% "simulated" increase in ξ leads to the plot of Figure 5 which has the expected behavior. However, we must be very cautious about drawing conclusions from the

simulated frictional force.

To date we have assumed that the complexity of the product under test is computed at the beginning of the test process and remains constant for the entire process. Certainly, we understand that this might not be true in all cases and especially when the removal of an error leads to a significant change in the product under test. However, in the case where there is a significant change in s_c during the STP, use of `Model S` allows the new value of s_c to be computed and plugged in for future computations.

Sensitivity of Δr to changes in γ

Similar to the variation in w_f, changes in the quality of the process (γ) are more effective at the early stages of the process as observed in Figure 6. Analysis also reveals that under the conditions of a low quality process and a large work force, increasing γ has more effect on Δr than increasing w_f. In a opposite situation, increasing w_f is a better alternative.

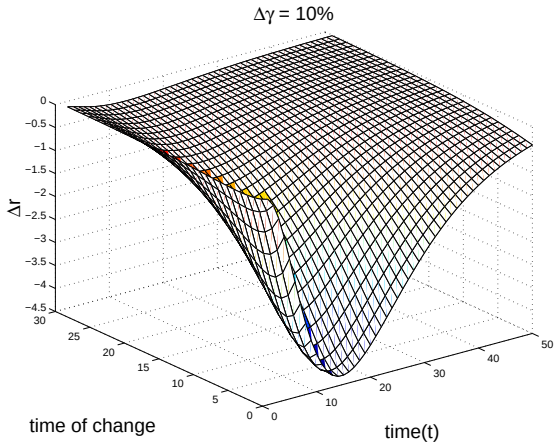

Figure 6: Sensitivity of the change in remaining errors (Δr) to a 10% change in γ.

The quality of a test process can be increased in many ways such as those listed below.

1. Improve the quality of test cases.
2. Improve the test plan.
3. Acquire and use a better test tool.
4. Increase the quality of the work force by exchanging a tester by one with more experience.

Items 1 and 2, in general, will not impose an extra effort in communication or learning, while items 3 and 4 will. Therefore, the increase in the overall quality of the test process (γ) must be analyzed jointly with the side effects of its change.

Sensitivity of Δr to changes in ζ and changes in ξ

The parameters ξ and ζ are obtained by a weighted least square [12] solution based on observed data. If the collection of this data is not precise, noise is introduced and can affect the computation of these parameters. For example, noise could be represented by unreported errors, by a missing or wrong date when the error was found, etc.. Therefore, the presence of noise justifies the sensitivity analysis of `Model S` for variations in ξ and ζ.

As with the other parameters, changes in Δr are more sensitive to changes in ζ at the beginning of the process and represent, like the changes in w_f, as much as a 4.5% increase in the rate of decay.

The sensitivity of the `Model S` to changes in ξ is similar to that for changes in ζ, but in the opposite direction. That is, increasing ξ slows the STP. The observations made regarding the effect of changes in ζ apply to changes in ξ also. As is observed in Figure 7, the variations in ξ and ζ can account for as much as 9% acceleration in the process (bottom layer in Figure 7) to the same slow down variation (top layer in Figure 7).

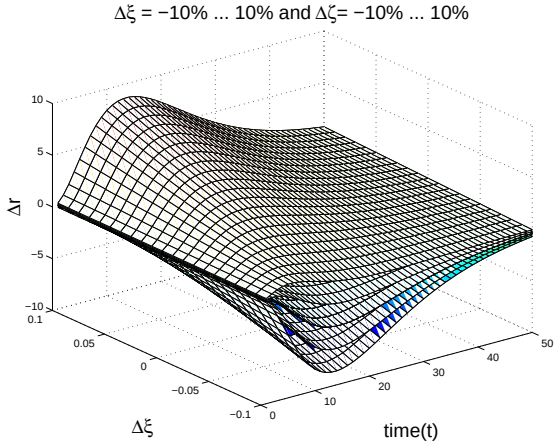

Figure 7: Sensitivity of the change in remaining errors (Δr) to a -10% to 10% change in ξ. The layers represent variations in $\Delta\zeta$ from -10% (top layer) from 10% (bottom layer).

If data from similar projects is available, ξ and ζ are estimated at the start of the STP, otherwise one needs to wait until data from the first few observations of the current project became available.

Indeed it is important to update these parameters in the course of the testing process. Combining this with the observation that the process is more sensitive at the beginning imposes a requirement of a good initial estimate for the parameters ξ and ζ.

4 Summary and Conclusions

Modeling the STP using a formal technique borrowed from the theory of automatic control allowed us to apply an analytical tool, namely the sensitivity matrix, to analyze the behavior of `Model S`. The advantages of using an analytical and not a simulation based approach, are in efficiency and precision. The approach is more efficient than for example, the simulation based approach, because it provides us with closed form solutions to the equations that characterize the model that are easy to manipulate using widely available tools such as MATLAB. The approach is more accurate because the closed form solutions are not quantitative approximations that result when using a simulation-based approach. The results of the sensitivity analysis conducted in Section 3.2 serve two main purposes: an improvement in our understanding of the behavior of `Model S` and the generation of ideas for the improvement of the model.

The sensitivity analysis of `Model S` suggests that

- changes in the process's parameters are more effective at early stages of the process;
- under certain conditions, late changes can make the process slow down instead of speed up (Brook's law); and
- in some cases improving the quality of the process is better alternative than increasing the size of the test team.

Another lesson learned from the sensitivity analysis is that good parameter estimates at the beginning of the process are important. This is due to the higher sensitivity of `Model S` at the beginning of the process to variations in ξ and ζ. Finally, the sensitivity analysis exposed weaknesses in `Model S`. These weaknesses are related to the sensitivity of `Model S` output to changes in s_c and to the lack of explicit accounting for learning and communication overhead in `Model S`.

References

[1] R. A. DeCarlo, *Linear systems : a state variable approach with numerical implementation.* Upper Saddle River, New York: Prentice-Hall, 1989.

[2] D. G. Luenberger, *Introduction to Dynamic Systems: theory, models and applications.* John Wiley & Sons, 1979.

[3] J. W. Cangussu, R. DeCarlo, and A. Mathur, "A state variable model for the software test process," in *Proceedings of 13th International Conference on Software & Systems Engineering and their Applications(ICSSEA)*, (Paris-France), December 2000.

[4] J. W. Cangussu, R. DeCarlo, and A. Mathur, "A formal model for the software test process," Tech. Rep. SERC-TR-176-P, Purdue University-SERC, March 2001.

[5] J. J. Marciniak, "Total quality management in software development," in *Encyclopedia of Software Engineering*, vol. 2, pp. 1359–1376, Whiley Interscience, 1994.

[6] J. J. Marciniak, "Verification and validation," in *Encyclopedia of Software Engineering*, vol. 2, pp. 1409–1433, Whiley Interscience, 1994.

[7] J. W. Cangussu, R. DeCarlo, and A. Mathur, "A state model for the software test process with automated parameter identification," in *Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference (SMC 2001)*, (Tucson-Arizona), September 2001.

[8] J. W. Brewer, "Matrix calculus and the sensitivity analysis of linear dynamic systems," *IEEE Transactions on Automatic Control*, vol. 23, pp. 748–751, August 1978.

[9] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on Circuits and Systems*, vol. 25, pp. 772–781, September 1978.

[10] F. P. Brooks, *The Mythical Man-Month.* Addison Wesley, 1995.

[11] T. H. McCabe, "A complexity measure," *IEEE Transactions on Software Engineering*, vol. 2, no. 6, pp. 308–320, 1976.

[12] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems.* Society for Industrial and Applied Mathematics (SIAM), 1995.