A Stochastic Control Model of the Software Test Process

João W. Cangussu
Department of Computer Science
University of Texas at Dallas
Richardson - TX 75083-0688, USA
cangussu@utdallas.edu

Abstract

The state variable model of the software test process has been shown to be a promising approach for the control of the system test phase of the Software Test Process. However, the presence of unforeseen perturbations and noise in the data collection process motivates the application of a stochastic approach. This paper explores a stochastic model construct upon the deterministic state variable model. The Capability Maturity Model is used to identify levels of disturbance and noise that can be associated with the maturity level of processes. Simulation results are presented as an indication of the applicability of the stochastic state model.

1. Introduction

The understanding and controlling of a process requires knowledge about the state of the process at a given time. The more information available regarding the states of the process the better the controllability level. The software development process has been modeled using different techniques ranging from finite state machine, process language, and simulation based models among others [1, 2, 3, 4, 5, 6]. Recently, a state variable approach has also been used to model and control the system test phase of the Software Test Process (STP) [7, 8]. A state variable is a set of differential equations organized in a matrix format allowing the prediction and control of the states of a process. This approach distinguishes from the others by presenting, under a control theory perspective, a closed feedback control loop.

Despite the difficulty in creating mathematical models, plausible accurate models have been derived for physical and non-physical systems [9, 10]. However, the difficulty level arises when modeling non-physical system due to the fact that observation/measurement of these processes is not accurate, more specifically, they are subject to disturbances and noise data. Under these circumstances, a stochastic rather than a deterministic model appears to be a better solution to represent the process. The software test process presents such characteristics and seems to be suitable for a stochastic approach.

Here, a stochastic model of the software test process is presented. The model is a variant of a previously specified deterministic control model, briefly described in Section 3. The new model accounts for foreseen and unforeseen perturbations as well as for noise in the data collection process. The level of disturbances can be adjusted according to the maturity level of the organization.

The remainder of this paper is organized as follow. Section 2 presents background material related to state model and the Capability Maturity Model [11]. A brief description of the deterministic model of the STP is presented in Section 3. The new stochastic model construct upon the deterministic version is presented in Section 4 along with a description of foreseen perturbations and the association of CMM levels to unforeseen disturbances and noise level. A result from the simulation runs is presented in Section 5. Section 6 presents the concluding remarks.

2 Background

In this section a brief description of two important topics related to the stochastic model of the software test process is presented. Section 2.1 presents a short description of state variable models whereas Section 2.2 define the maturity level of a organization based on the CMM levels. These levels are used later in Sections 4.2 and 4.3 to define disturbance and noise degrees expected in a specific process.

2.1 State Models

Linear state models have provided useful representations for a multitude of systems, ranging from engineering to biological and social processes [10]. The general format of a deterministic Linear Time Invariant (LTI) state model is presented in Eq. 1.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{1}$$

where $x(t)$ is the state vector, $u(t)$ is the input, and A, B, C, and D are coefficient matrices [9, 10]. The dominant variables that can properly characterize the states of a process compose the state vector $x(t)$. The availability a state model capturing the dominant aspects of a process allows the application of control theory. The fundamental control problem, as defined by Goodwin [9], is stated below:

> **Definition:** *the central problem in control is to find a technically feasible way to act on a given process so that the process adheres, as closely as possible to some desired behavior. Furthermore, this approximated behavior should be achieved in the face of uncertainty of the process and in the presence of uncontrollable external disturbances acting on the process.*

Based on the definition above it can be seen that the Software Development Process (SDP), more specifically the STP, presents all the ingredients for the application of control theory. The adherence to a desired behavior, the presence of uncertainty, and external disturbances are common place in any SDP and therefore justifies the investigation of the approach presented here.

2.2 Capability Maturity Model

The levels of the Capability Maturity Model are presented in Figure 1 [11]. A brief description [11] of each level is presented next to allow the association, in Sections 4.2 and 4.3, of CMM levels to disturbance and noise levels in organizations/groups with distinct process maturity.

Initial Level (1): this level can be characterized by presenting an “ad hoc” process completely dependent on the skills of the project manager. Coding and testing are the typical solutions for a crisis. The capability of organizations/groups at Level 1 is unpredictable, unstable and dependent on individual rather than an organizational level.

Repeatable Level (2): this level presents an improvement by establishing policies and management procedures/controls at organizational level. This allows successful practices, based on previous projects, to be repeated leading to more realistic expectations. Projects at Level 2 are more stable due to the discipline ensured by the established management controls.

Defined Level (3): the major difference between Levels 2 and 3 is the presence of software process standards at the latter. Not only standards but also verification mechanisms and completeness criteria are in place at the organizational level. Stability and consistence are two major characteristics of processes in organizations at Level 3.

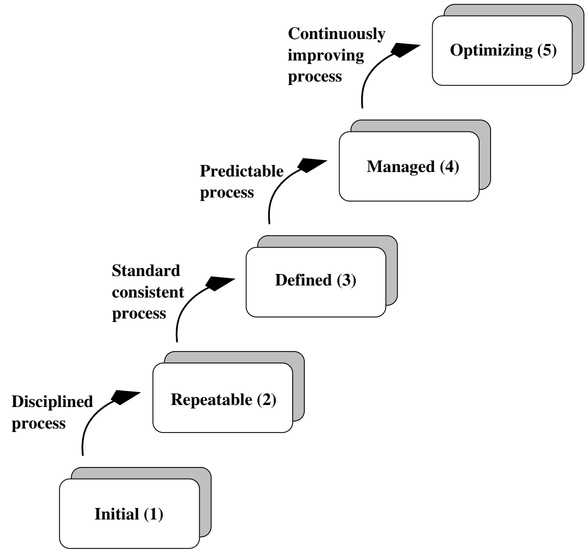

Figure 1. Levels of the Capability Maturity Model

Managed Level (4): qualitative and quantitative goals are established for both, the software product and process. Measurement plays a central role in achieving these goals. Predictability represents the major improvement achieved by organizations at Level 4.

Optimizing Level (5): a continuously process improvement characterizes organizations at Level 5. Analysis of the results of projects and new technologies combined with constant revision of the standards and control procedures lead to process improvement and consequently better process prediction capabilities.

The Managed Level (4) requires an environment where quality and productivity can be measured. With these two properties and others required at lower levels a direct use of a deterministic model is foreseen. The same will apply to Level 5, since it represents an improvement over Level 4. Although Level 3 has a well defined software development environment, the process characteristics are not completely appropriate to a deterministic approach due to lack of measurement. The lack of measurement at levels 1 and 2 increases making the use of a deterministic model more difficult, though not impossible.

3. Previous Results: Deterministic Model

The linear deterministic model of the STP is based upon three assumptions. These assumptions and the corresponding equations are presented below [7]. They are based on an analogy of the STP with the physical process typified by a spring-mass-dashpot system and also in Volterra's predator-prey model [10]. A description and justification of this analogy and the choice of a linear model is outside the scope of this paper [7]. The model has been validated using sets of data from testing projects [12] and also by means of a extremal case and sensitivity analysis [13, 14].

Assumption 1: *The rate at which the velocity of the remaining errors changes is directly proportional to the net applied effort (e_n) and inversely proportional to the complexity (s_c) of the program under test, i.e.,*

$$\ddot{r}(t) = \frac{e_n(t)}{s_c} \quad \Rightarrow \quad e_n(t) = \ddot{r}(t)\ s_c \tag{2}$$

Assumption 2: *The effective test effort (e_f) is proportional to the product of the applied work force (w_f) and the number of remaining errors (r), i.e., for an appropriate $\zeta(s_c)$,*

$$e_f(t) = \zeta(s_c)\ w_f\ r(t) \tag{3}$$

where $\zeta(s_c) = \dfrac{\zeta}{{s_c}^b}$ is a function of software complexity. Parameter b depends on the characteristics of the product under test.

Assumption 3: *The error reduction resistance (e_r) opposes, is proportional to the error reduction velocity ($\dot{r}$), and is inversely proportional to the overall quality (γ) of the test phase, i.e., for an appropriate constant ξ,*

$$e_r(t) = -\xi\ \frac{1}{\gamma}\ \dot{r} \tag{4}$$

Combining Eqs. 2, 3, and 4 in a force balance equation and organizing it in a State Variable format ($\dot{x} = Ax + Bu$) [9, 10] produces the following system of equations.

$$\begin{bmatrix} \dot{r}(t) \\ \ddot{r}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-\zeta\ w_f}{s_c^{(1+b)}} & \frac{-\xi}{\gamma\ s_c} \end{bmatrix} \begin{bmatrix} r(t) \\ \dot{r}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{s_c} \end{bmatrix} F_d(t) \tag{5}$$

$$\begin{bmatrix} r(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} r(t) \\ \dot{r}(t) \end{bmatrix} \tag{6}$$

F_d above is included in the model to account for unforeseen disturbances such as hardware failures, personnel illness or any event that slows down or even interrupts the continuation of the test process.

Along with the model in Eq. 5 an algorithm has been develop to calibrate the parameters of the model [15]. The fast convergence presented by the algorithm increases the model applicability and accuracy [16]. Finally, a parametric control procedure is used to compute required changes in the model in order to converge to desired results according to time constraints [7]. The input $u(t)$ is, in general, used to drive the system. However, the procedure used for the control of the system in Eq. 5 uses a parametric approach and the input F_d is used to account for unforeseen perturbations, as stated earlier.

4. Stochastic Model

The deterministic model described in Section 3 seems to be appropriated to control relatively well defined test processes, where the level of unpredictability is not critical. However, in practice, test processes are subject to a variety of external disturbances. In addition, collected data is often noisy and prediction of the intermediate states of the process may be compromised depending on the level of inaccuracy of the data. Under such circumstances, one alternative would be the use of a stochastic model as presented in Eq. 7. In this case, the prediction of the intermediate states of a system becomes an stochastic rather than a deterministic process.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + G\eta(t) \\ y(t) = Cx(t) + D\varphi(t) \end{cases} \tag{7}$$

where $x(t)$ is the state vector, $u(t)$ is the input, A, B, C, G, and D are coefficient matrices, and η and φ are two mutually independent noise sequences [17].

The inclusion of noise components represents the major difference between the deterministic and the stochastic model. The influence of randomly external disturbance is accounted for by the noise sequence $\eta(t)$ in Eq. 7 whereas the inaccuracy of the collected/measured data is represented by the noise sequence $\varphi(t)$ in the output part of the same equation.

4.1 Input Characterization

The input F_d in the deterministic model of Eq. 5 was used to account for unforeseen perturbations. However, F_d was included, periodically, as the average perturbation from the previous time period(s) and therefore still characterizes a deterministic process. F_d was also used to model common disturbances in the test process, such as a training period, a migration of the product under test from the developers to the users environment, the replacement of an already tested component, etc. [18]. Since unforeseen perturbations

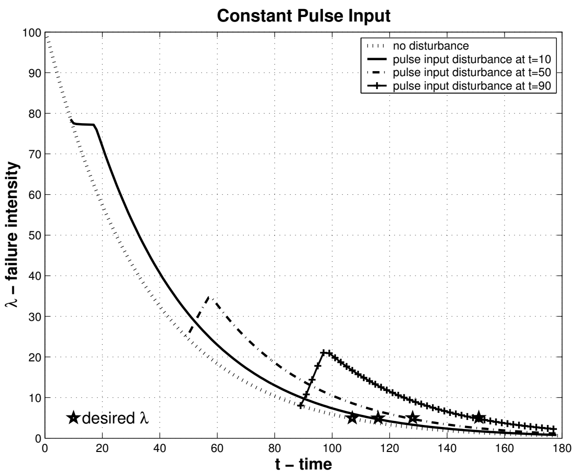

Figure 2. Results of the perturbation of the process due to the migration of the system from the development to the user's environment. A pulse input signal is used to represent the disturbance.

are accounted for by the stochastic processes $\eta(t)$ and $\varphi(t)$, F_d is used here account to for foreseen perturbations as the ones mentioned above [18]. This appears to be a reasonable approach due to the fact that these disturbances can be modeled and they are, in many situations, anticipated/expected. As before, the system is not driven by the input $u(t)$ and a parametric control technique can still be applied.

For example, assume the test process cannot be conducted, from the beginning, at the user's environment. The process starts at the developer's environment and the migration will be allowed at a known time period. Since the time is known in advance, the effect of the migration on the outcome of the test process can be predict. The prediction of the results at three different time periods can be seen in Figure 2. As expected, the later the migration is performed, the larger the effect of the disturbance [18]. More results on the characterization of the input can be found elsewhere [18].

4.2 Disturbance Characterization

As stated above, $\eta(t)$ and $\varphi(t)$ account for unforeseen perturbations. Both of these stochastic processes are represented here by uniform randomly distributed white noise sequences. The major concern becomes how to characterize the level of the disturbance to proper represent the ones found in different software test processes. The maturity of the organization/group where the process is being conducted is a good indication of the level of disturbance expected. The levels of the Capability Maturity Model [11] presented in Section 2.2 can be used for this purpose.

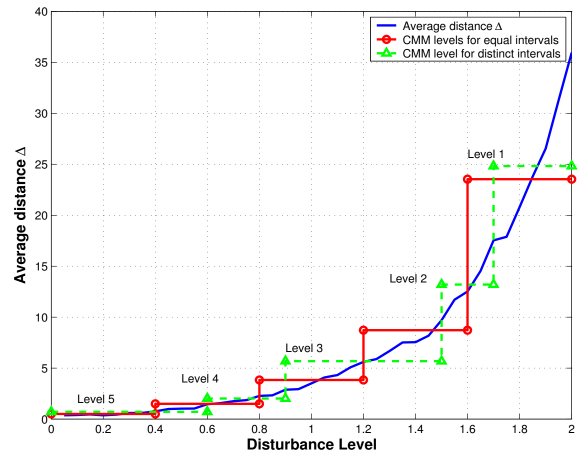

Figure 3. Average Disturbance Levels associated with each CMM level.

The matrix G in Eq. 7 determines the level of the disturbance inserted into the system. That is, $G = \begin{bmatrix} D_l \\ 0 \end{bmatrix}$, where D_l, ranging from zero to two, represents the disturbance level. The disturbance is inserted by perturbing the velocity component ($G_{1,1} = D_l$) of the model which will, in a chain reaction, perturb the acceleration component. In this case, the direct perturbation of the acceleration component is not necessary and therefore $G_{2,1} = 0$.

Now, let r_e be the expected decay of number of errors and r_d be the disturbed decay. The distance between these two curves is computed as $\Delta = |\, r_e - r_d \,|$. Figure 3 depicts how the average distance Δ changes as the disturbance level increases from 0 to 2 for 50 simulation runs. As can be observed, Δ increases exponentially as the disturbance level increases. Initially one would expect to set the disturbance level at equal intervals of 0.4 for each CMM level, i.e.,

$$(5 - L_i) \times 0.4 < D_l \leq (6 - L_i) \times 0.4$$

where $L_i = 1, 2, \ldots, 5$ represents the corresponding CMM levels. The distance Δ associated with each level under the equal interval assumption is observed in Figure 3.

At CMM Level 1 the ad hoc characteristics of the process does not allow a reasonable prediction of the results of a project and therefore the level of disturbance can be determined as high. A considerable improvement is expected as an organization moves from Level 1 to Level 2 and therefore a reasonable decrease in the disturbance level is expected. The exponential shape of the Δ curve captures this behavior. The improvement from Level 2 to Level 3, and consequently the decrease of the disturbance level, is supposed to be large but not as large as from Level 1 to 2. Again, the equal interval assumption appears to be a good solution. The problem arises when considering the

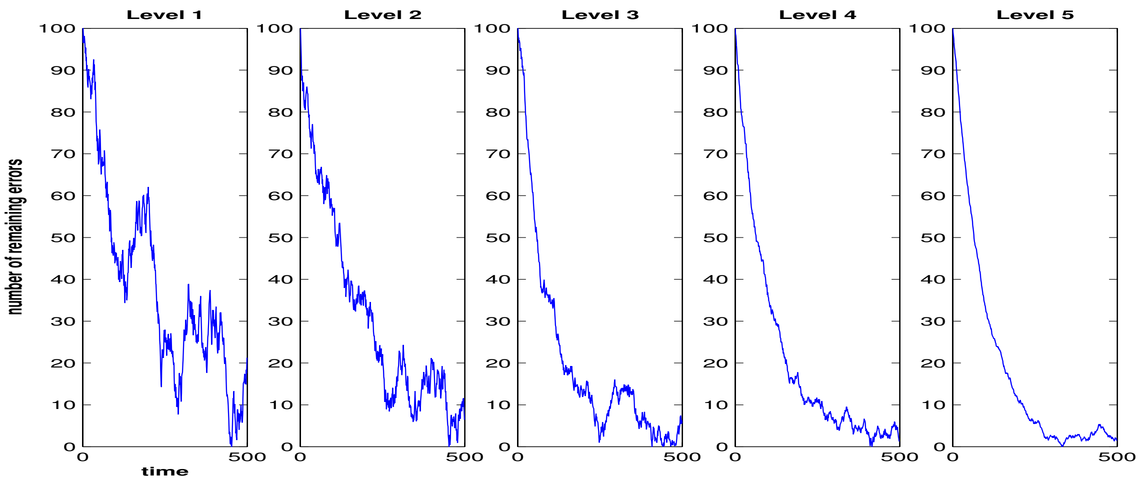

Figure 4. Unforeseen disturbance associated with each level of the Capability Maturity Model. The disturbance is generated by the randomly distributed white noise sequence η in Eq. 7.

improvement from Level 3 to 4. Due to the qualitative and quantitative goals and the measurement scheme to achieve them, a reasonable decrease in the disturbance level is expected and, as observed from Figure 3, the equal interval assumption does not properly represent this behavior.

Table 1. Disturbance and noise level associated with levels of the Capability Maturity Model used to generate the results in Figures 4 and 6.

CMM Level	Disturbance Level	Noise Level
1	$0 < D_l \leq 0.6$	$0 < N_l \leq 4$
2	$0.6 < D_l \leq 0.9$	$4 < N_l \leq 8$
3	$0.9 < D_l \leq 1.5$	$8 < N_l \leq 12$
4	$1.5 < D_l \leq 1.7$	$12 < N_l \leq 16$
5	$1.7 < D_l \leq 2$	$16 < N_l \leq 20$

A larger disturbance gap between Levels 3 and 4 is desired. A definition of the intervals as presented in Table 1 appears to achieve the expected differences as an organization moves between levels. As observed from Figure 3, a large difference exists between Levels 1 and 2 and also a larger gap is observed between Levels 3 and 4. Though some decrease in the disturbance level is expected from Level 4 to 5, the decrease is not foreseen to be large. Improvement in performance due to a constantly optimization of the process is the major impact at Level 5 and a large decrease in disturbance is not a consequence of this.

Figure 4 shows the results of applying disturbance levels within the ranges specified in Table 1 for each of the CMM levels. It can be observed that the behavior of the process moves from chaotic at Level 1 to a reasonably stable and predictable process at Level 5. The test process in the plots is characterized by the following parameter values: $w_f = 2$, $s_c = 30$, $\gamma = 0.4$, $b = 1.12$, $\xi = 100$, and $\zeta = 60$. A sensitivity analysis of the model under the presence of disturbance can provide information of how changes in the parameters affect the behavior of the model. However, such analysis is outside the scope of this paper.

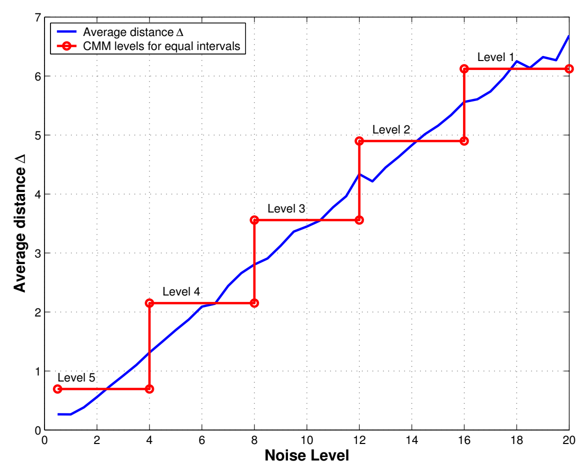

Figure 5. Average Noise Levels associated with each CMM level.

In addition, disturbances seem to have different levels, according to different time periods, for distinct project. For example, the disturbance level for a test process where the test team is not familiar with the product under test will

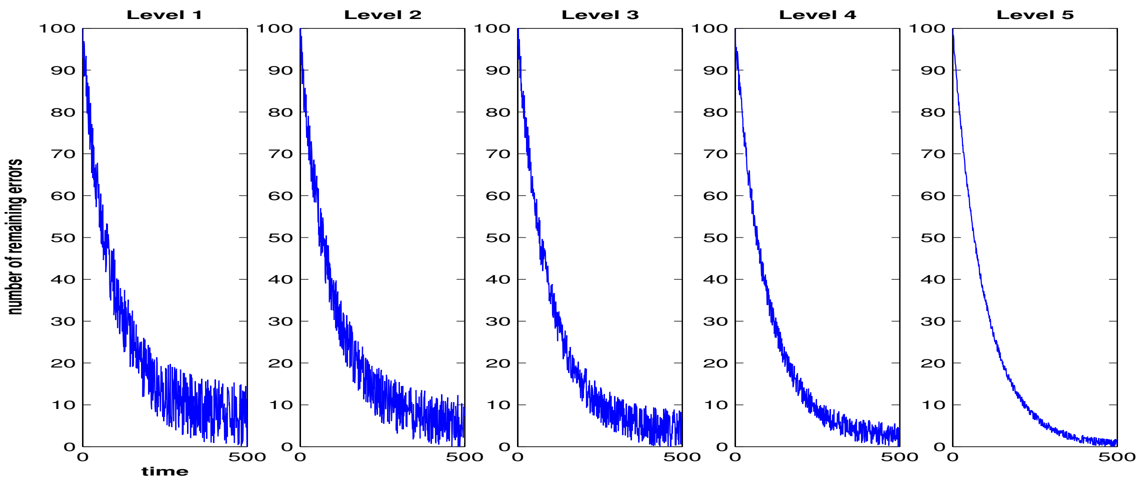

Figure 6. Noise in the data collection process associated with each level of the Capability Maturity Model. The noise is generated by a randomly distributed white noise sequence φ in Eq. 7.

differ from a process, within the same organization, where the test team has previous experience with similar products. The interval associated with each CMM Level can be used to represent/adjust these differences.

4.3 Noise Characterization

The stochastic process $\varphi(t)$ in Eq. 7 represents noise in the data collection process. Unreported errors, duplicated reports, and missing information such as date when the error was found are a few examples of problems/noise when collecting data. These are common problems that are likely present in any organization. However, the more mature the process and the organization, the less the likelihood and the frequency of occurrence of these problems. Again the CMM levels are used here in association with noise levels.

Matrix D in Eq. 7 determines the level of the noise associated when measuring the output of the system. One output variable $r(t)$ is present in Eq. 6 and therefore the dimension of matrix D is 1×1. In this case, the single element of D, referred hereafter as N_l, accounts for the noise level.

Unlike the disturbance, the average distance Δ (computed as before) related to the noise level does not present an exponential increase. From Figure 5 it is clear that the distance Δ increases linearly as the noise level increases.

Though it may not be true for organizations at CMM Level 1 or 2, let us assume that a data collection process is in place. As the maturity level of an organization or development group increases the presence of standards, control mechanisms, quantitative measurements, etc. leads to a reduction on the noise in the data collection. However, the noise associated with reporting errors or defects is not expected to have a large impact on the behavior of the model. This can be noticed by the true nominal value of the average distance Δ in Figure 5. In addition, the characteristics of the CMM levels do not give indication of a specific pattern in the increase of noise. Under these conditions, as shown in Figure 5 and Table 1, an equal interval for the noise associated with the CMM levels is assumed.

Figure 6 shows the results of inserting noise related to the data collection process in the output of the model. As can be observed, the perturbation of the output decreases linearly as the process moves from Level 1 up to Level 5. The process in the plot is characterized by the same parameters used in Figure 4. As before, a sensitivity analysis of the model can be used to collect information on the behavior of the model when noise is present. However, this issue is not addressed here.

The results of a combined insertion of data collection noise and unforeseen perturbations in a test process is depicted in Figure 7 along with the related CMM level. As expected, Figure 7 appears like a merge of Figures 4 and 6.

Here, the noise sequences η and φ are represented by a uniform randomly white noise sequence. However, under certain circumstances these sequences are likely to present not only a specific distribution but also specific trends. In certain cases, the effects of a disturbance are expected to increase as time progresses and such correlation needs to be taken into consideration when modeling the disturbance.

Another aspect to be considered is the distribution related to CMM levels. A uniform distribution may be appropriated to Levels 4 and 5 while a normal distribution may better represent the disturbance at Levels 1, 2, and 3. In the second case, the parameters μ and σ can be used to distinguish each CMM level. The study of probabilistic distributions that better represent the disturbance and noise se-

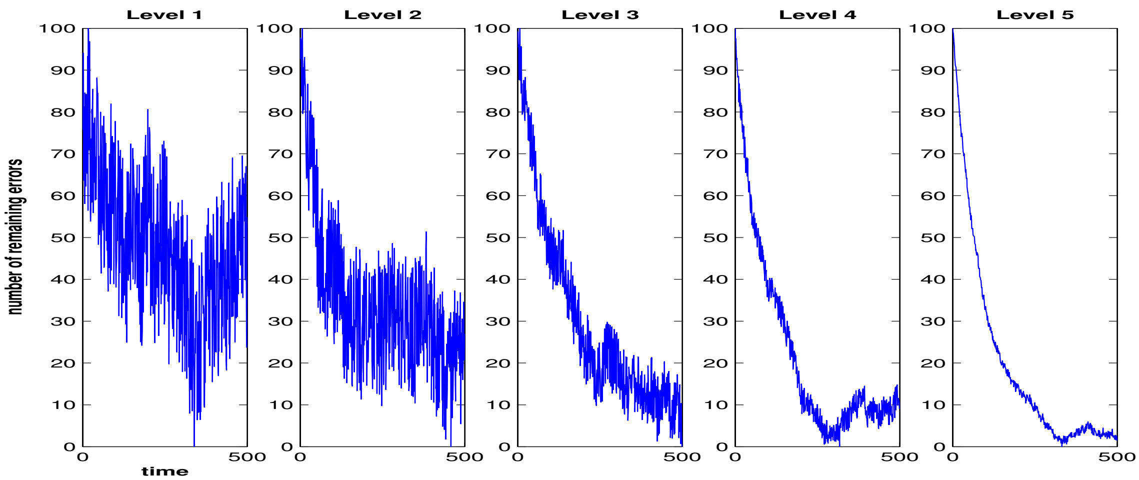

Figure 7. Combination of data collection noise and unforeseen disturbances associated with each CMM level.

quences of the stochastic model in Eq. 7 is a subject of future investigation.

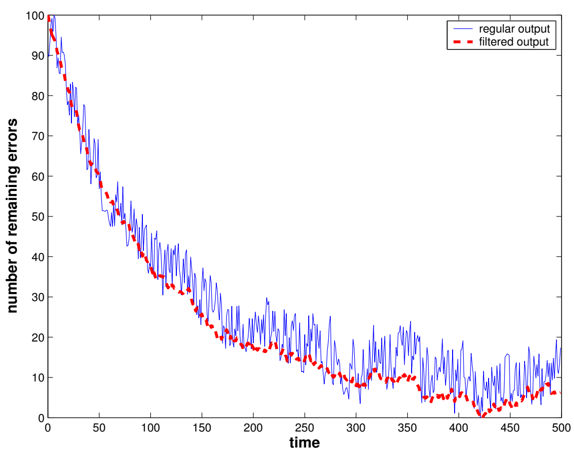

Figure 8. Application of a Kalman filter to a noisy test process.

4.4 Noise Filtering

The next step in improving the applicability of the model is the use of a Kalman filter. The filter allows a better prediction of the states of the system in the presence of noise. Grewal and Andrews state " ... it is not possible or desired to measure every variable you want to control, and the Kalman filter provides a mean for inferring the missing information from indirect (and noisy) measurements." [19]

The result of the application of a Kalman Filter [19] in the stochastic version of the state variable model of the software test process is presented in Figure 8. It can inferred from Figure 8 that the Kalman filter has a great impact on the noise associated with the data collection process but does not affect the unforeseen perturbations. This behavior of the filter properly represents the expected results, i.e., the noise is reduced but the effects of the perturbations are still considered. Changes in the overall structure of the feedback model of the STP are required for the application of the filter and a detailed explanation of these changes are beyond the scope of this paper.

5 Simulation Results

One of the results of the simulation runs is depicted in Figure 9. The process in this Figure is typified by the same parameters defined in Section 4.2. In Figure 9 we can observe the results of the undisturbed deterministic model (dashed-doted line) representing the expected decay of errors with a goal of reducing the number of remaining errors to 15% within 180 days. The result of the introduction of a disturbance sequence level $D_l = 1.3$ and a noise sequence level $N_l = 9$ is also shown in Figure 9. In this case, according to Table 1, a process at CMM Level 3 is characterized. As can be observed, the goal of the process cannot be achieved under the presence of disturbance.

Assume that the first checkpoint is at time $t = 50$, where it can be observed that the process is not proceeding according to the expected behavior. If we apply the control approach used for the deterministic model [7], an increase of $\Delta w_f = 0.5$ (a half time tester) places the eigenvalues of the system at the desired level. However, the disturbances are not accounted for and the process does not converge to

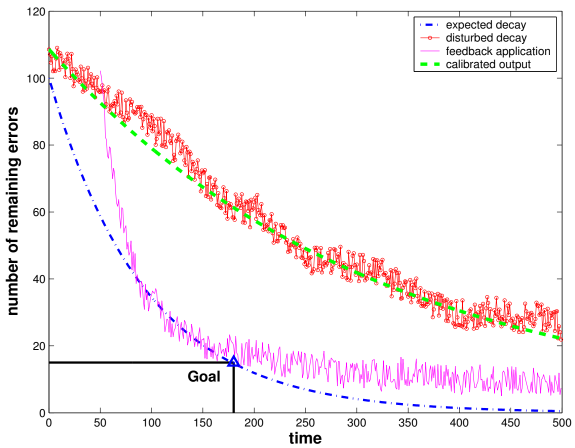

Figure 9. Result from feedback application of the Stochastic State Model for the Software Test Process.

the expected curve. A group/organization is unlikely to improve the CMM Level in the middle of a process and therefore disturbances are expected to continue slowing down the process.

The parametric control approach used here is done at two levels.The first step is to calibrate the model to incorporate the disturbance into the parameters ζ and ξ. The result of the calibration is represented by the dashed line in Figure 9. Then, the changes needed to place the eigenvalues of the calibrated model and make the results converge to the expected curve are applied not to the calibrated but to the disturbed model. The changes in the calibrated model are already accounting for disturbances and therefore will have a similar impact in the disturbed model, as can be observed in Figure 9. An increase of $\Delta w_f = 5.5$ is required to achieve the desired goal within the deadline. As one could argue, side effects of increasing w_f (Brook's Law [20]) are expected and are analyzed elsewhere [13].

Finally, the changes to drive the system to the desired behavior were presented here in terms of w_f. However, the same approach applies to changes in the quality of the test process γ or to combined changes of w_f and γ.

6 Conclusion

The introduction of noise sequences in the state variable model of the software test process extends its use to organizations in the lower CMM levels. Similarly to the deterministic model, it allows the computation of changes in process parameters to correct for schedule deviations in the process. In addition, statistical information allowing inferences on the accuracy of the predictions related to the level of perturbation injected into the model can be acquired.

The results presented here are an indication of the applicability of the stochastic state model of the STP. However, actual data is needed to validate the model. This data can be further used to analyze specific probabilistic distributions associated with the CMM levels.

References

[1] T. Abdel-Hamid and S. E. Madnick, *Software Project Dynamics: An Integrated Approach*. Upper Saddle River, New Jersey: Prentice Hall Software Series, 1991.

[2] G. Cugola and C. Ghezzi, "Software processes: A retrospective and a path to the future," *Software Process Improvement and Practice*, vol. 4, no. 2, pp. 101–123, 1998.

[3] G. Cugola, "Tolerating deviations in process support systems via flexible enactment of process models," *IEEE Transactions on Software Engineering*, vol. 24, November 1998.

[4] A. G. Cass, B. S. Lerner, E. K. McCall, L. J. Osterweil, S. M. Sutton, and A. Wise, "Little-jil/juliette: A process definition language and interpreter," in *Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000)*, (Limerick, Ireland), pp. 754–757, IEEE, June 2000.

[5] G. A. Hansen, "Simulating software development processes," *IEEE Computer*, vol. 29, pp. 73–77, January 1996.

[6] N. S. Eickelmann, "Applying simulation technologies to manage test practices," in *Proceedings of the 18th International Conference on Testing Computer Software*, (Bethesda, MD), June 18-22 2001.

[7] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, "A formal model for the software test process," *IEEE Transaction on Software Engineering*, vol. 28, pp. 782–796, August 2002.

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

[9] G. C. Goodwin, S. F. Graebe, and M. E. Salgado., *Control system design*. Upper Saddle River, New Jersey: Prentice Hall, 2001.

[10] D. G. Luenberger, *Introduction to Dynamic Systems: Theory, models and applications*. New York: John Wiley & Sons, 1979.

[11] M. C. Paulk and et. al., “Capability maturity model for software,” tech. rep., Software Engineering Institute, Carnegie Mellon University, 1993.

[12] J. W. Cangussu, *A Mathematical Foundation for Software Process Control*. PhD thesis, Purdue University, April 2002.

[13] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, “Using sensitivity analysis to validate a state variable model of the software test process,” *IEEE Transactions on Software Engineering*, vol. 29, pp. 430–443, May 2003.

[14] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, “Sensitivity analysis of a state variable model of the software test process,” in *Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference (SMC 2001)*, (Tucson, Arizona), pp. 712–717, October 2001.

[15] J. W. Cangussu, R. A. DeCarlo, and A. P. 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), pp. 706–711, October 2001.

[16] J. W. Cangussu, “Convergence assessment of the calibration algorithm for the state variable model of the software test process,” in *Proceeding of the IASTED International Conference on Software Engineering (SE’03)*, (Innsbruck, Austria), February, 10-13 2003.

[17] G. Chen, G. Chen, and S.-H. Hsu, *Linear Stochastic Control Systems*. Boca Raton, Florida: CRC Press, Inc., 1995.

[18] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, “Effect of distrubances on the convergence of failure intensity,” in *Prooceding of* 13^{th} *International Symposium on Software Reliability Engineering*, (Annapolis, MD, USA), November 12-15 2002.

[19] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*. New York, NY.: John Wiley & Sons, second ed., 2001.

[20] F. P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*. Reading, MA: Addison Wesley, 1995.