Chapter 4

A Signal Processing Model of a Product Development Process Phase

4.1 Introduction

Multiple perspectives can facilitate understanding a model, its limitations and uses, and the system it describes. In this chapter I describe my application of an alternative modeling approach to a portion of the Product Development Project Model. This work extends the investigation of model behavior and illustrates a different and valuable means of understanding system behavior. This improved understanding can aid in the design of rigorous policies, practices, and systems.

I built a signal processing model of some of the most important parts of the Development Tasks sector (Figure 3-8) for a single phase. The system was analyzed to determine the conditions required for stable system behavior and system-induced oscillation. The model was used to generate simulations across a range of parameter values to improve understanding of the impacts of individual parameters on system behavior. Finally, the limitations, use, and potential applications of the signal processing modeling approach is discussed.

4.2 System Descriptions

The system dynamics and signal processing representations of the system modeled in this chapter are shown in Figures 4-1 and 4-2.

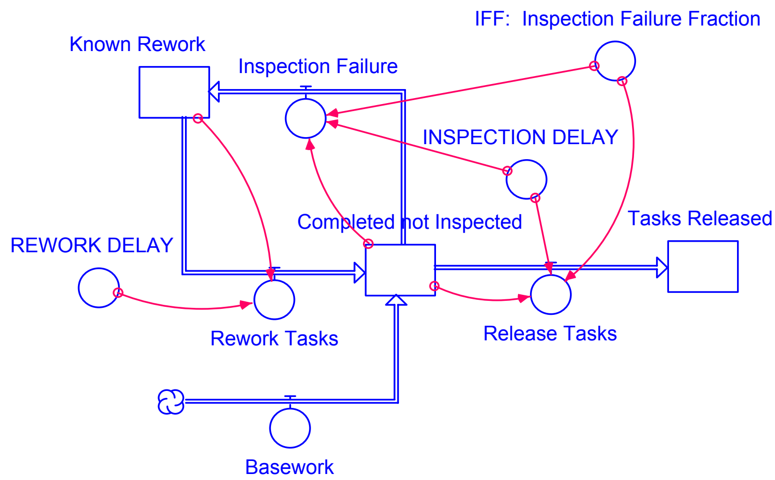

Figure 4-1: A System Dynamics Representation of a Product Development Process Phase Model

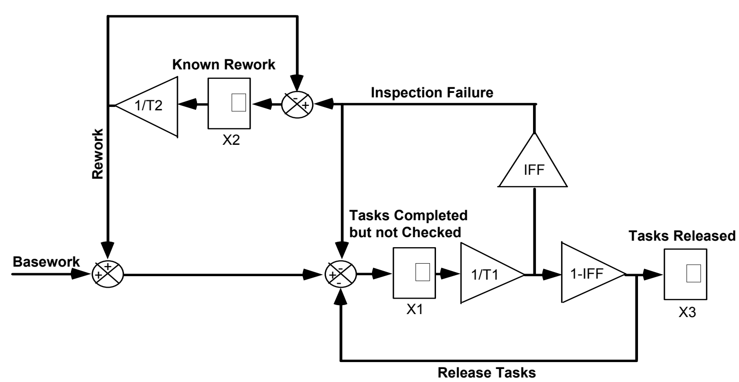

Figure 4-2: A Signal Processing Representation of a Product Development Process Phase Model

These two system representations are mathematically equivalent. The following differential equations describe the system:

The State Variables:

$$CT_i = CT_{i0} + \int_0^T (BW_i + RW_i - IF_i - RT_i)\, dt$$

where

i = development phase
T = time, weeks
CT = Completed but not Checked, tasks
BW = Basework rate, tasks per week
RW = Rework rate, tasks per week
IF = Rework due to Inspection Failures, tasks per week
RT = Release Tasks rate, tasks per week

$$KRW_i = KRW_{i0} + \int_0^T (IF_i - RW_i) \, d$$

where
KRW = Know Rework, tasks
IF = Rework due to Inspection Failures, tasks per week
RW = Rework rate, tasks per week

$$TR_i = TR_{i0} + \int_0^T RT_i \, dt$$

where
TR = Tasks Released, tasks
RT = Release Tasks rate, tasks per week

The Transfer Equations:

$$RW_i = KRW / RWDur$$

where
RW = Rework rate, tasks per week
KRW = Know Rework, tasks
RWDur = Minimum Rework Task Duration, weeks

$$IF_i = IFF * (CT / QADur)$$

where
IF = Rework due to Inspection Failures, tasks per week
IFF = Inspection Failure Fraction, dimensionless

CT = Completed, not Checked stock, tasks
QADur = Quality Assurance Minimum Duration, week

$$RT_i = (1 - IFF) * (CT / QADur)$$

where
RT = Release Tasks rate, tasks per week
CT = Completed, not Checked stock, tasks
QADur = Quality Assurance Minimum Duration, week
IFF = Inspection Failure Fraction, dimensionless

The basis for these equations is described in chapter 3. The equations describing the signal processing model (Figure 4-2) can be rearranged into a system description in the standard matrix form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}*x + \mathbf{B}*f$$

$$y = \mathbf{C}*\mathbf{x} + \mathbf{D}*f$$

where
x - a vector of state variables
$\frac{dx}{dt}$ - the rate of change of the state variables over time
y - output signal, taken to be the Tasks Released state variable
f - input (forcing) signal, taken to be the Basework rate
A - system matrix
B, C, D - matrices describing the relationships among the input signal and rate of change function or output signal

For the model shown the matrices are:

$$A = \begin{bmatrix} -1/T1 & 1/T2 & 0 \\ IFF/T1 & -1/T2 & 0 \\ (1-IFF)/T1 & 0 & 0 \end{bmatrix}$$

B = [BW 0 0]
C = [0 0 1]
D = [0 0 0]

The preceding equations were programmed into a signal processing simulation package (Burras et al., 1994) with forcing functions (input signals) and system responses (outputs) for analysis and simulation. The code is shown in Appendix 4.1

4.3 System Behavior

4.3.1 Introduction

Knowing the parameter values which generate stability of system behavior is important for understanding a system. A large body of systems analysis methodology exists for studying systems described in the signal processing form above (e.g. see Dorf and Bishop, 1995; Ogata, 1989; Oppenheim and Schafer, 1989). Many methods of investigating behavioral stability are available from this domain. A fundamental approach is used for the illustrative purposes of this chapter. The conditions for system stability were investigated using eigenvalues of the system matrix. This approach has previously been successfully applied to a system dynamics model of an economy (Forrester, 1982). The single phase process system has three eigenvalues:

0 (input determines behavior)

$$\left(\frac{1}{2}\right) * \frac{-T2 - T1 + \sqrt{T2^{2} - 2*T1*T2 + T1^{2} + 4*T1*T2*IFF}}{T1 * T2}$$

$$\left(\frac{1}{2}\right) * \frac{-T2 - T1 - \sqrt{T2^{2} - 2*T1*T2 + T1^{2} + 4*T1*T2*IFF}}{T1 * T2}$$

4.3.2 Conditions for Stability

System eigenvalues define a system's poles in the frequency domain. Those poles can be used to describe system behavior by plotting them in a root locus plot in the S-plane. The S-plane is a Cartesian coordinate system with real components plotted on along the abscissa and imaginary components plotted along the ordinate. Positive real system pole components correspond to unstable system behavior. Negative real pole components correspond to stable system behavior.

A more detailed discussion of the use of the frequency domain is available in Karu (1995) or Oppenheim and Schafer (1989).

The two non-zero system poles can be simplified for analysis into the sum of or difference between two expressions, referred to here as Expr1 and Expr2:

$$\underbrace{\frac{-T2-T1}{2*T1*T2}}_{\text{Expr1}} \pm \underbrace{\frac{\sqrt{T2^2-2*T1*T2+T1^2+4*T1*T2*IFF}}{2*T1*T2}}_{\text{Expr2}}$$

The conditions for potential system instability can be shown by plotting Expr1 and Expr2 on the S-plane (Figure 4-3).

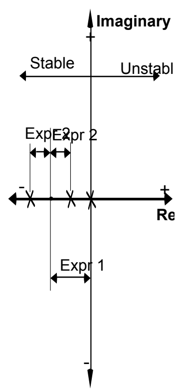

Figure 4-3: The System Poles Plotted in the S-Plane

From Figure 4-3 the two potential causes of system unstable behavior are:

- Expr1 > 0
- Expr2 > (-1) * Expr1

The first potential condition for instability reduces to

$$\text{Expr1} = \frac{-T2-T1}{2*T1*T2} = \frac{-1}{2} * \frac{T2+T1}{T1*T2} > 0 \text{ (unstable system)}$$

or

$$\frac{T2+T1}{T1*T2} < 0$$

By inspection the preceding inequality can only occur when $T1 < 0$ or $T2 < 0$.

A central part of the traditional signal processing approach to system evaluation is the consideration of system behavior under all parameter values regardless of the impossibility or reasonableness of the values and the subsequent limiting of the model or parameter values based on the results of the evaluation. This evaluate-all-then-limit approach distinguishes the signal processing approach from the system dynamics approach which limits parameter values to possible and reasonable ranges and then evaluates system behavior within those limits.

An understanding of the meaning and significance of parameter values which can cause instability will allow reasonable restrictions on parameter ranges. The first conditions to be considered are the conditions just established for system in stability, when $T1 < 0$ or $T2 < 0$. T1 and T2 represent smoothing and delays of inspection and rework activities, respectively. Values of $T1 < 0$ or $T2 < 0$ would represent infeasible system conditions and cause impossible movement of development tasks such as reversing the inspection process, causing Known Rework to become "un-inspected" and return to the Competed but not Inspected stock. The results of such conditions can be shown with a simulation of the system response to a pulse loading at time t = 5 and T1 = -10, as shown in Figure 4-4. A single phase project system producing a negative number of Tasks Released is clearly infeasible.

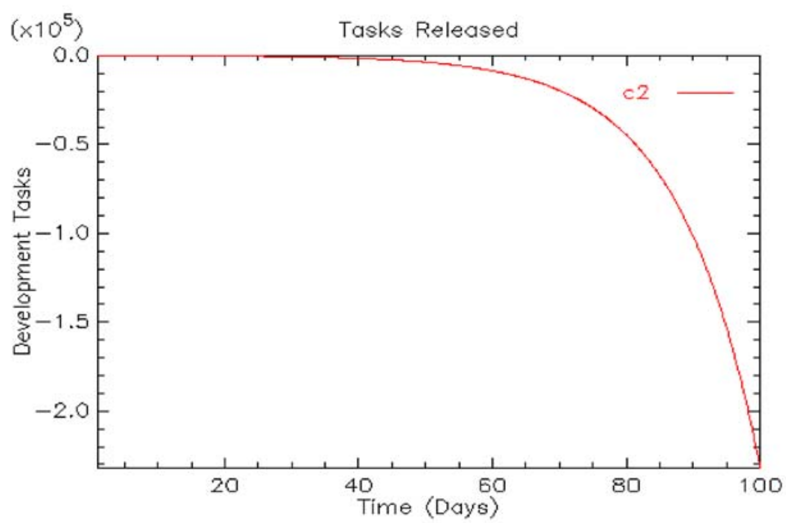

Figure 4-4: System Response (Tasks Released) to Pulse Signal with T1 = -10

Similar reasoning can be applied to the conditions in which T1 = 0 or T2 = 0. The condition T1 = 0 implies that inspection begins instantaneously upon completion of Basework or Rework and takes no time to perform. This is not feasible in real systems. Likewise when T2 = 0 Rework begins instantaneously upon identification of errors and takes no time to perform. This is also not feasible in real systems.

Based on the preceding the following restrictions and conclusions can be drawn concerning parameter values for T1 and T2 based on the first condition for stability:

- For a stable system behavior restrict Parameter Values for T1 and T2 to T1 > 0 and T2 > 0.
- No feasible system descriptions are eliminated by these restrictions.
- For T1 > 0 and T2 > 0 the system never meets the first condition for instability.

From Figure 4-3 the second potential cause of system instability is:

Expr2 > (-1) * Expr1

This condition can be reduced to

$$\frac{\sqrt{T2^2 - 2*T1*T2 + T1^2 + 4*T1*T2*IFF}}{2*T1*T2} > (-1) * \frac{-1}{2} * \frac{T2+T1}{T1*T2}$$

or

$$\frac{\sqrt{T2^2 - 2*T1*T2 + T1^2 + 4*T1*T2*IFF}}{T2+T1} > 1$$

This condition uses the Inspection Failure Fraction (IFF). Three possible ranges of values are possible for IFF:

IFF < 0
0<= IFF <= 1
IFF > 1

An evaluation of the meaning and significance of IFF at the first and third ranges simplify the analysis. By definition IFF is a fractional amount of a non negative quantity (tasks inspected) and remains within the range 0 - 1. Therefore IFF < 0 or IFF > 1 is meaningless and will not occur in actual systems.

Therefore the following restrictions and conclusions can be drawn concerning the values of IFF:

- Restrict Parameter Values for IFF to 0 <= IFF <= 0
- No feasible system descriptions are eliminated by these restrictions.

The second condition for instability can be evaluated using the prescribed range for IFF.

When IFF = 0 then system is unstable if $\frac{T2 - T1}{T2 + T1} > 1$

By inspection this inequality can never be true within the prescribed ranges for T1 and T2 (T1 > 0 and T2 > 0).

When IFF = 1 then $\frac{T2 + T1}{T2 + T1} = 1 > 1$

This inequality can never be true.

In the range 0 < IFF < 1 the left hand side of the inequality increases from

$\frac{T2 - T1}{T2 + T1}$ to $\frac{T2 + T1}{T2 + T1} = 1.$

Therefore this range of values for IFF will never generate a true inequality when T1 > 0 and T2 > 0.

Based on the preceding the following conclusions can be drawn concerning the second condition for system instability.

- Restrict Parameter Values for IFF to 0 <= IFF <= 1.
- No feasible system descriptions are eliminated by this restriction.
- For 0 <= IFF <= 0 the system never meets the second condition for instability.

By combining the conclusions concerning the conditions for system instability the following conclusions can be drawn:

- The System is stable within parameter ranges T1 > 0, T2 > 0, 0 <= IFF <= 1.
- No feasible system descriptions are excluded by these parameter restrictions.

4.3.3 System Changes Which Impact Stability

The analysis of system stability prompts the question "What system changes could cause instability?" Two possibilities are suggested.

- **Increase system gain with IFF** > 1: If the assumption of tasks as atomic pieces of work and therefore completely correct or completely flawed is relaxed then more than one error could be generated in each task. This could be represented by IFF > 1. This could generate more rework than tasks, potentially increasing total work infinitely.

- **Add exogenous work:** If errors generated rework from beyond the project scope additional work could enter the system directly into the Known Rework stock and increase the work infinitely.

- **Describe the system with pure delays:** If the first order exponential delays currently used to describe the demand-based flows are replaced with infinite order delays which shift conditions without changing their values the system will behave significantly differently. Although this change is not sufficient alone to cause instability, it may contribute to instability when combined with other changes.

Identifying system changes which could cause important behavioral changes can expand the understanding of the model by investigating the impacts of stretching the model to describe new or different systems and conditions. Considering how the model behaves when stretched beyond its original envelope of assumptions and parameter values improves the understanding of the model within its original envelope.

4.3.4 Conditions for System Induced Oscillation

A second system behavior of possible interest is oscillatory behavior generated by the system. A system which induces oscillatory behavior is described in the frequency domain on the S-plane (Figure 4-3) by system poles plotted off the real component axis. This condition requires the eigenvalues to have imaginary components. This is true when:

$$T2^2 - (2*T1*T2) + T1^2 + (4*T1*T2*IFF) < 0$$

By inspection the minimum value of the left side of the inequality occurs when IFF = 0. Under this condition the inequality reduces to:

$$(T2 - T1)^2 < 0$$

This inequality can never be true since T2 and T1 are real numbers. Because the inequality can never be true the system eigenvalues cannot have imaginary components. Therefore:

- The system will not generate oscillation under feasible parameter values.

4.3.5 Typical System Behavior

Visualizing the system behavior within the prescribed parameter value ranges can increase understanding of system behavior. Two types of simulations will be used to illustrate system behavior:

- Simulation of the behavior of several system components over time under single specific parameter values. This helps improve understanding of the interaction of system components.
- Simulation of a single system component behavior over time across a range of parameter values. This helps improve understanding of the impacts of specific parameters on system behavior.

The use of the Release Tasks rate as a primary system output can facilitate improved understanding of system behavior. This is because as the rate of final completion of tasks the Release Tasks rate is directly analogous and comparable to the input signal, Basework, which is the initial completion of tasks.

4.3.5.1 System Response to Impulse Signal

The impulse signal is a standard test of system behavior and produces the system's response to a single shock. Figures 4-5, 4-6, 4-7, and 4-8 show the behavior of several system components over time for an impulse input signal.

Input signal: pulse at time t = 5
T1 = 5
T2 = 8
IFF = 25%

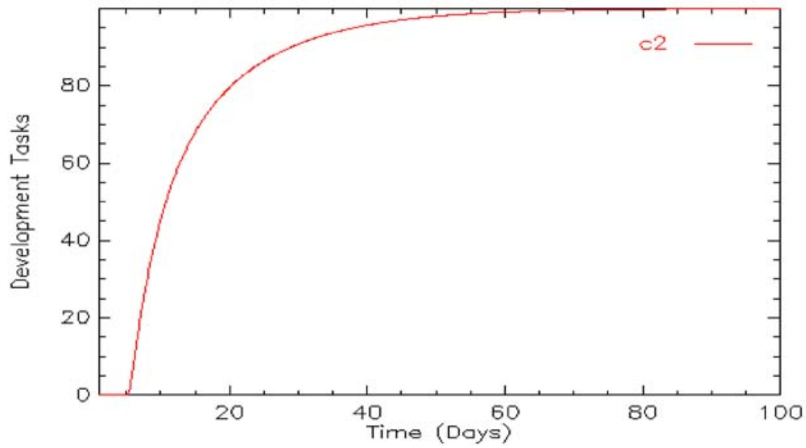

Figure 4-5: System Response (Tasks Released) to Impulse Signal

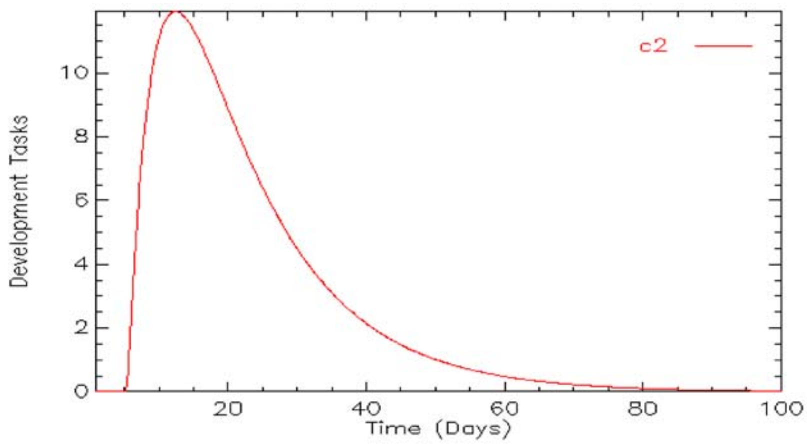

Figure 4-6: System Response (Known Rework) to Impulse Signal

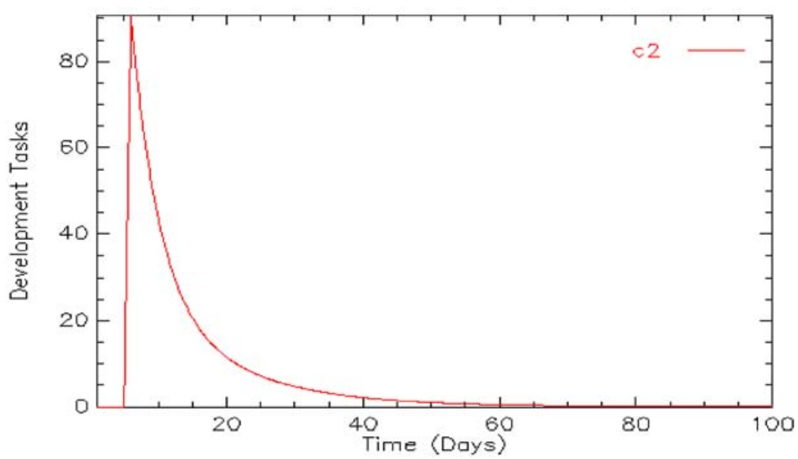

Figure 4-7: System Response (Tasks Completed but not Checked) to Impulse Signal

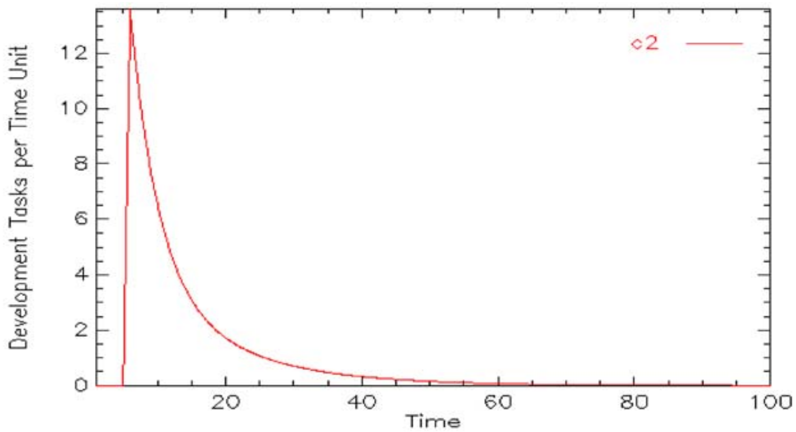

Figure 4-8: System Response (Task Release Rate) to Impulse Signal

4.3.5.2 System Response to Block Signal

Another common test signal is the step signal, which changes instantaneously from one stable amplitude to another stable amplitude. However the pure step differs from all possible basework signals for a project model in at least one important way: their cumulative size. The step signal continues infinitely and therefore has an infinite cumulative size. But the initial completion of project tasks eventually returns to and remains zero. This limits the project scope to some real finite size.[4.1] A similar but more realistic and useful signal for analyzing the behavior of a project model is a block signal. A block signal steps up from zero to a stable amplitude for a specific period of time and then steps down to zero again. The block signal is useful for testing and understanding a project model because it represents the limited scope of the project, whereas a pure step signal has no such limit. The block signal can represent an idealized project with a steady rate of initial task completion. The eventual return of the input signal (initial completions) to zero raises interesting and potentially important questions about projects:

- How does the system affect project performance after the signal has stopped?
- How long after the signal has stopped does it take for the project to end?

[4.1] The end to the basework signal and resulting existence of a limit on size is a fundamental distinction between projects, which end, and continuous operations, which are assumed to not end. In other words, a project must eventually come to an end or it is not a project.

These and other questions can be investigated with a block test signal more easily than with a pure step signal. An additional tool for developing system understanding has been added to the investigation of system responses to a block signal: the visualization of changes in system responses as parameters change. Figure 4-9 illustrates this model investigation tool by showing the system response to a block signal over a range of values for the Inspection Time Constant, T1. The input signal and parameter values are:

Input signal: step from 0 to 10 at time t = 5 and step from 10 to 0 at time t = 45
T1 varies from 1 to 10
T2 = 8
IFF = 25%

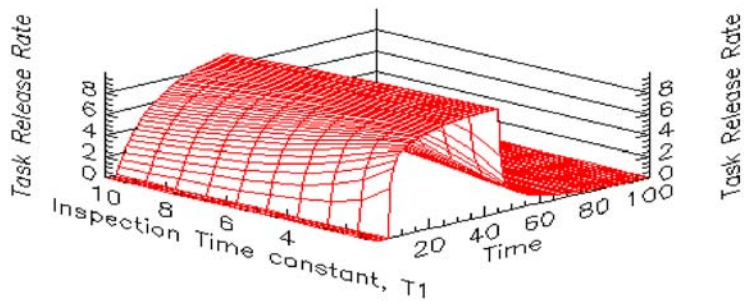

Figure 4-9: System Response (Task Release Rate) to Block Signal
Impact of Inspection Time Constant

The increase in the time required for the Task Release Rate to reach steady conditions as the Inspection Time Constant increases is clear in figure 4-9. In contrast changes in the system response to the same signal as the Inspection Failure Fraction varies across its range from 0 to 1 is shown in Figure 4-10. The input signal and parameter values are:

Input signal: step from 0 to 10 at time t = 5 and step from 10 to 0 at time t = 45
T1 = 5
T2 = 8
IFF varies from 0 to 1

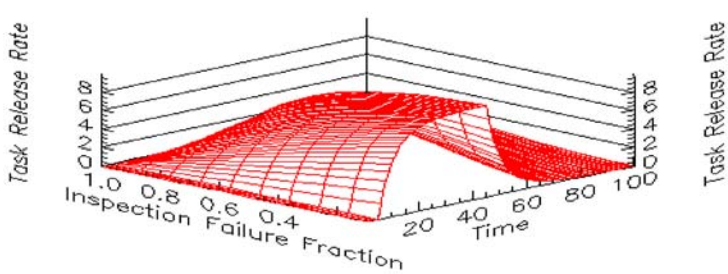

Figure 4-10: System Response (Task Release Rate) to Block Signal
Impact of Inspection Failure Fraction (IFF)

Figures 4-9 and 4-10 illustrate the different impacts of parameters on the system's behavior and how the signal processing model facilitates understanding those impacts. This knowledge can be useful in the design of system changes which would alter parameter values and generate improved system responses.

4.4 Discussion of the Signal Processing Modeling Approach

4.4.1 Alternative System Descriptions and Analysis Tools

The illustrative examples herein and other applications of classical control theory to system dynamics (e.g. Kampmann, 1992, N. Forrester, 1982) demonstrate the potential benefits of applying a signal processing perspective to the Product Development Project Model. Those benefits include:

- Methods for the exhaustive exploration of system behavior under all parameter value ranges
- Analysis tools for the study of system behavior, particularly stability
- Tools for simulating and describing system behavior in very flexible ways which facilitate improved understanding of model behavior

4.4.2 Applicability of Linearity Assumptions

All modeling approaches are limited by their fundamental assumptions. One of the most important assumptions used in the signal processing approach is the use of only linear relationships. The domain of applicability of the approach is affected by this assumption.

Linear relationships appear reasonable and useful descriptions of several project decision rules. For example the linear equations describing the three demand-based flows of development tasks among the Completed but not Checked, Known Rework, and Tasks Released stocks appear reasonable. Linear descriptions may also be reasonable for some additional project process relationships. For example work availability constraints within a project phase may be linear in nature. An example is the first Internal Precedence Relationship described in Chapter 3 which describes the linear progression of installing the steel structure for a building.

However some important relationships in projects are nonlinear[4.2]. For example the minimum of the work available and labor available limits as a basis for development activity rates is a fundamentally nonlinear relationship. Changes in the degree of concurrence of work within a phase as the phase progressed can cause nonlinear work-availability constraints. Inter-phase constraints may also be nonlinear. Other examples can be found in the relationships which describe the impacts of project conditions on humans such as the effect of schedule pressure on workweek. A graph of this relationship approaches asymptotic limits of the impacts and are nonlinear in their approaches. The nonlinearities in these relationships may be important determinants of system behavior. Describing some of these relationships with linear approximations may degrade the model's ability to describe the system's behavior (Forrester, 1992).

In conclusion the linear assumption used by the signal processing approach appears reasonable for many project relationships but limits the breadth of its applicability to modeling development projects. Careful consideration of each relationship between system components and the restrictions placed on parameter ranges allows the beneficial application of the signal processing approach to many portions of the Product Development Project Model.

4.4.3 Tractability of Analyzing Larger Systems

Another important issue is the tractability of analyzing larger systems with the signal processing approach. While systems significantly larger than the example given here can be analyzed with this approach the difficulty of analysis increases with system size and with the degree to which system components are linked. Closed form analysis becomes particularly challenging. This challenge leads to the use of two approaches: system decomposition and simulation. System decomposition seeks to model the system in separable linked pieces, each which can be more easily analyzed (Homer, 1983). The Product Development Project Model uses this approach by modeling a single project with a linked set of generic development phase structures. Hierarchical decomposition can extend this approach to analyzing models of larger systems. Simulation allows the investigation of models without the exhaustive analysis of each parameter (Clemson et al., 1995). These modeling approaches are useful for the analysis of models such as the complete Product Development Project Model.

[4.2] See Chapter 5 The Python Development Project for examples of nonlinear relationships.

4.5 Signal Processing Modeling Approach Summary

The signal processing approach was applied to a portion of the Product Development Project Model to investigate an alternative perspective to the traditional system dynamics modeling approach. Conditions for stable and oscillatory model behavior were studied using the system eigenvalues. The portion of the system studied was found to respond with stable non oscillatory behavior when restricted to parameter values which describe feasible project systems. System behavior was also investigated with simulations. These simulations illustrate system responses to an impulse signal and a block signal which can represent an idealized project. The use of visual displays of system responses as important parameter values vary was illustrated as one of the advantages of the signal processing approach. Issues raised by the application of the signal processing modeling approach were discussed, including the linear relationship assumption and the tractability of analyzing larger systems. The signal processing approach was found to provide several advantages and improve the analysis and understanding of the Product Development Project Model.