PREDICTION THEORY

FOR

CONTROL SYSTEMS

BY:
WILLIAM C. CAVE

PREDICTION SYSTEMS, INC.
PREDICTION & CONTROL SYSTEMS ENGINEERS

309 Morris Avenue – Suite J
Spring Lake, NJ 07762

☎ (732)449-6800   📄 (732)449-0897
📫 PSI@predictsys.com   🕸 www. predictsys.com

PREDICTION THEORY

FOR

CONTROL SYSTEMS

July 17, 2020

BY:

WILLIAM C. CAVE

PREDICTION SYSTEMS, INC.

PREDICTION & CONTROL SYSTEMS ENGINEERS

309 Morris Avenue
Spring Lake, NJ 07762

☎ (732)449-6800 🗎 (732)449-0897
📫 PSI@predictsys.com 🕸 www. predictsys.com

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

PREFACE

The ability to accurately predict future outcomes of complex systems, such as military operations, business markets and national economies, has been the goal of forecasters for decades. The field is rich with sophisticated methods taken from high technology disciplines, e.g., the Kalman filter taken from control theory, [19]. However, these methods have done little to replace forecasts using human judgment by people knowledgeable in their field. It is apparent that, if a computer could do better over a long period of time, the sophisticated methods would be highly recommended, and their developers would be wealthy. Evidently, this is not the case.

Engineering systems are certainly very complex. But to most engineers the major objective is building a control system. In this case, the imbedded prediction systems are rarely broken out as separate models. This is because typical engineering problems require tight-loop control, where the time constants are short between observations (typically sensor inputs) and control outputs. In this problem space, prediction is embodied in a single step model, handled implicitly by the solution to a set of differential equations. As we will show in the technical discussions below, explicit prediction models are left out and estimation techniques, e.g., Kalman Filtering are used to resolve the statistical error.

Single step models are sufficient when the system dynamics may be assumed stationary. However, this assumption may not be valid for problems that require predicting multiple steps into the future. In these cases, one finds that control theory does not deal very well (if at all) with multi-step prediction. However, as certain control systems become more sophisticated, one starts to realize that the embedded prediction system becomes the critical part of the solution. Based upon years of experience with such problems, it became apparent that an extension to the theory is needed. The extended theory provided here covers multi-step prediction, using a new approach to nonstationary systems, particularly those that heretofore depended upon human judgment, e.g., air traffic control or power grid control. This extension highlights the significant difference between estimation and prediction, a concern not normally found in engineering disciplines. This extension effectively elevates prediction theory as a distinct discipline.

Einstein did much work using the theories of physics, mathematics, and probability. He also understood the methods of statistics. He knew how and where each could be applied to help solve the problem of predicting an event quite far into the future. Most important, he had the ability to perceive the underlying structure of physical systems, and could represent these structures using models of their dynamics. And this ability, *to model the underlying structure of a system - to a sufficient level of accuracy* - lies at the heart of the prediction problem.

In order to increase accuracy of prediction, one must be able to apply additional information. This additional information does not have to come from observation data. In fact, it should come from knowledge of how a system operates internally, with all the inherent feedback loops, and the external factors that influence it. Specifically, it comes from knowing how factors act as leading forces that can be observed in advance of the system's response. This implies modeling how the inertial properties of one entity may affect those of another, and how the corresponding feedback effects can produce highly nonlinear responses. Without an approach that can characterize inertial properties whose time constants are sufficiently long, there is effectively no chance of predicting future responses with useful accuracy.

MODELING PHYSICAL SYSTEMS

One of the major differences between Einstein's approach to the theory of relativity and that of other researchers, e.g. Poincaré, was the selection of the space used to define and solve the problem. Most every other researcher was using a pair of spaces, e.g., free space and ether (to which Einstein did not relate), or dual coordinate systems, x, y, z, t and x', y', z', t'. Einstein derived his theory based on the physical properties of the system using a single x, y, z, t space.

According to Lorentz, Einstein used the physical interpretation of the Lorentz transformation the basis for a clear and simple discussion of the electrodynamics of moving bodies, whereas Poincaré's remarks on the physical interpretation of the Lorentz transformed quantities apparently struck Lorentz as inconsequential philosophical asides in expositions that otherwise closely followed his own. Lorentz found Einstein's physically very intuitive approach more appealing than Poincaré's rather abstract but mathematically elegant approach.

Prediction Systems, Inc. (PSI), has always used the expression "modeling along physical lines." This goes back to the days when we were competing with companies that were modeling similar - if not the same - communication systems. Some of these competitors used mathematical approaches, e.g., queuing theory based on probabilistic models. A few were using special simulation languages, e.g., SIMSCRIPT or GPSS, both of which used a *Discrete Event* approach, a significant theoretical improvement - but poorly implemented (each ran very slow). Additionally, neither were amenable to handling large numbers of mobile nodes, nor modeling radio systems where connectivity (who could talk to who) was constantly changing the network structure (which nodes were connected). Most engineering groups were using FORTRAN or languages based on FORTRAN. These ended up causing unwanted mathematical abstractions

The problem of building structural (versus statistical) models is currently being faced by practitioners who are trying to produce more accurate forecasts. The problem stems from the most difficult task of translating knowledge of a system's structure into a model, and the subsequent difficulties in verification and validation of executable computer code. Because of these difficulties, many forecasters fall back on statistical approaches, fitting the data with mathematical functions that get extrapolated into the future.

In contrast, PSI has always worked to directly model the physical elements of a system that were required to achieve sufficient accuracy of the measures of merit (performance or effectiveness) that had to be produced. This invariably leads to building models that follow the actual design or physics of the system being modeled. Models are built that represent sufficient detail to produce the level of accuracy required for the measures of merit. To run on parallel processors, independent modules must be designed that follow the inherent parallelism of the system being modeled. These models are then easy to understand, build, and change (one can add more detail as needed). By closely representing the physical system, models naturally run very fast (time-wasting mathematical abstractions and transformations are eliminated).

After studying existing simulation languages and their faults, and the requirement to run very fast on parallel processors, a new language environment was developed. Another requirement was the ability for subject area experts, e.g., communication engineers, to build the models directly. This implied that the language had to read like English. There is no relationship between the readability of the user language and the run-time speed. It does require extremely complex language translators and a run-time system that is generated automatically based upon the architecture of independent modules. But this burden falls on the computer.

So how does one teach and implement a structural modeling approach if it is so difficult to comprehend conceptually, and so difficult to apply in practice? There are no articles or textbooks to help people - even those with an excellent math background - to create such models successfully.

There are three sides to solving this problem. One side is the need for special education. The academic environment must look at prediction theory as an interdisciplinary field. Probably the most difficult problem to be overcome is assimilation of the model building process used by physical scientists doing worst-case design of complex systems. Many of the fallacies that econometricians invoked when attempting to apply control theory to economics are explained well by Athans and Kendrick, [2]. These problems were also pointed out quite vehemently by Kalman himself, [19]. It is this author's belief that attempts to apply control theory must be preceded by an understanding of discrete event systems. This is the most important ingredient to achieving accurate predictions of nonlinear nonstationary systems. Maybe then the academic community can gain a sufficiently deep understanding of the underlying aspects of complex systems to deal with prediction theory.

The second side of the problem is the need for a unified theory to support a scientific approach to evaluating and comparing prediction methods and techniques. This need is amplified by the numerous attempts to compare forecasting methods, and the amount of literature - whose validity is questionable - consumed by the ensuing argumentation. The foundation for such a theory must be based on broad principles that are widely accepted in the scientific community as representing invariant facts about the real world. One can argue that these principles already exist. In fact, it is the thesis of this book that, by properly unifying the relevant existing theories of the physical sciences, a much clearer picture of the problem and its general solutions can be painted. This should also pave the way for the educational process.

The third side of the problem is the availability of tools to automate the model building process in a way that makes the development of prediction systems significantly easier to implement. These tools must also afford the model builder the ability to keep pushing complexity into the background, as he verifies and validates pieces of his model. Since 1982, this has been a major goal of this author. The result is embodied in ***VisiSoft***, a Computer-Aided Design (CAD) system in which subject area experts can easily build very accurate discrete event, and discrete or continuous time - models and simulations, as well as complex software. VisiSoft contains many facilities for developing interactive software, high resolution graphical interfaces, file and communication channel interfaces, and libraries as well as nonlinear optimization facilities to develop complex models, simulations, and real-time control systems. The interactive CAD developer interface is the Visual Development Environment (VDE).

The resulting VisiSoft product is much more than a language. It is an environment that includes additional features required for complex simulations. It includes the Run-Time Graphics (RTG) system, where end users interact with systems graphically - while they are running. It also includes a non-linear optimization facility that is used to optimize parameters for accuracy of prediction and for adaptive real-time control systems.

A unified prediction theory, supported by advanced tools and a proper education is the only way that problems requiring multi-step prediction can be solved. The ability to accurately predict outcomes depends upon the inherent properties of the system itself. Given that these properties exist, people must be trained and armed to take full advantage of opportunities to accurately predict and control their future. This book is an attempt to set out in that direction.

1. INTRODUCTION

Figure 1-1 is a simple illustration of a control system. In the problems of interest, the control system may contain a large number of observable inputs provided by many sources. In certain cases, these inputs may be processed by human intelligence to help make decisions on controlling the system. Once a plan (a sequence of control inputs) is made, the corresponding control actions are promulgated down to the subordinate people or systems to be carried out.

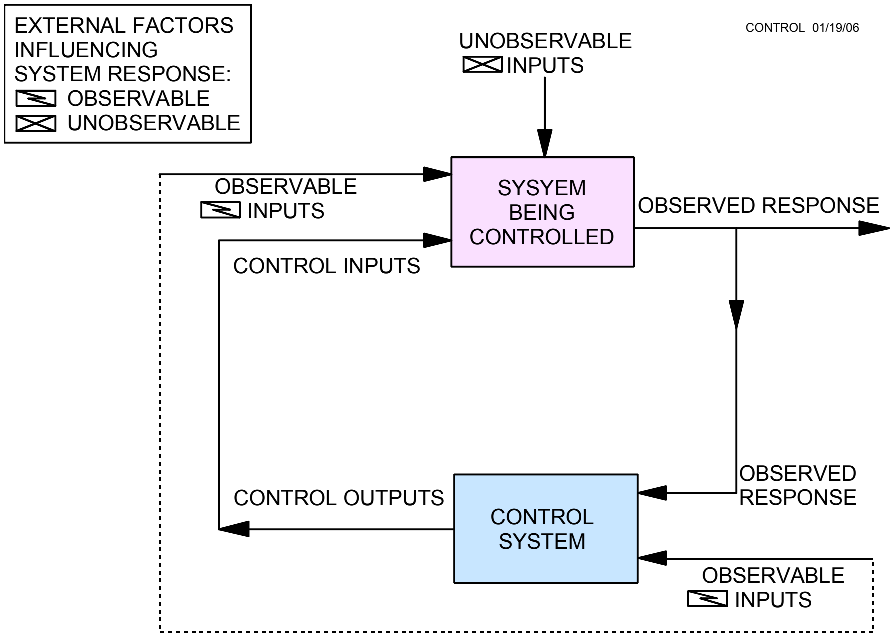


Figure 1-1. Simplified representation of a control system.

The plan that is promulgated is effectively the same as the optimal control sequence put out by the controller in a classical control system. Given desired objectives, the control system is constantly producing a sequence of parameters in real time that are used as controlling inputs to the system. However, in our case, the system may be distributed as well as require multiple prediction steps into the future.

The Embedded Prediction Component Of A Control System

The sophisticated part of most control systems is the embedded prediction subsystem. This is characterized generically in Figure 1-2. The prediction subsystem takes in a selected control sequence and observable inputs up to the current time T, and produces a prediction of the resulting system response out to some desired T+τ. To accomplish this, the prediction system must contain models that represent all of the complexities required to produce the predicted outcomes *with sufficient accuracy* to support the desired control inputs and desired system outputs. Our focus here is on nonstationary systems that require multi-step prediction.

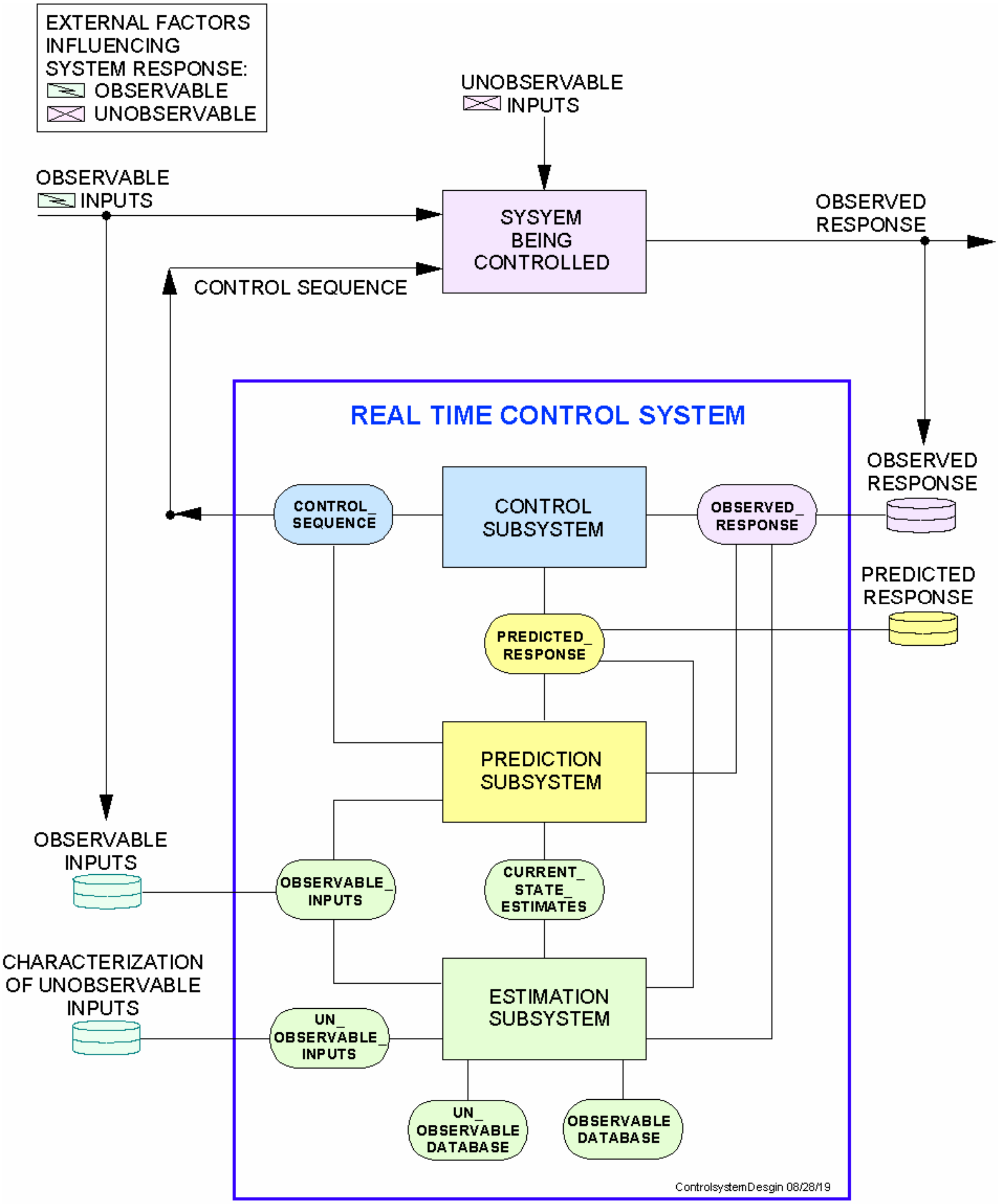


Figure 1-2. The embedded prediction component of a control system.

Sufficient accuracy of prediction depends upon the needs of the control system. For example, a household thermostat uses a simple estimate of the current temperature to turn on or turn off the heater. There is no prediction required. Similarly, a person planning a two-week trip to a distant part of the world wants to know the temperature range and typical rainfall during the period of interest so that the proper clothes can be packed to cover the range of possibilities.

This does not require a dynamic prediction model; a statistical estimate will suffice. However, a trip with no change of clothes may require an accurate prediction of weather. This is best obtained from a dynamic meteorological forecasting model for the area of interest. Such models can be quite complex.

Figure 1-2 provides a rough illustration of the required elements of a prediction model. The ovals represent the critical data spaces required to produce accurate estimates of prior, current and predicted states. It is the selection and representation of these complex spaces that support the design of algorithms that determine the accuracy of predicted responses. The importance of the design of these spaces cannot be over emphasized.

For nonstationary systems requiring accurate prediction models, one may use discrete event simulation and interactive graphics (a huge topic described elsewhere including many PSI and VSI documents). In this case, the control system produces sets of control sequences to the prediction system and gets back corresponding sets of predicted system responses. The optimal control problem is to come up with the control sequence that meets the *constraints* required of the system while optimizing some prescribed *objective* function. In the ensuing discussion, we will use the words *solution*, *control sequence*, and *desired outcome* or *desired output* interchangeably.

Prediction Versus Forecasting

When attempting to make decisions relative to best control actions, one wants to know what the outcomes would be for each potential control action selected. An example is tracking the seismic behavior of a volcano and trying to determine if and when to evacuate surrounding communities. Evacuation will cause a major disruption; but without an evacuation, many lives may be lost. When dealing with such problems, one will be trying to postulate the actions and reactions that will determine the best control actions to take. Predictions and forecasts are made to support the analysis and decision process that precedes control actions.

If sufficient data and time exist, then a *prediction* can be made with the accuracy characterized. If not, one must make a *forecast*. When decisions are critical, particularly if life and death are at stake, it is important to understand the difference between prediction and forecasting to avoid misleading statements and corresponding results.

As defined here, predictions can only be made when the accuracy of the prediction mechanism can be characterized in terms of historic data used to compare *a priori* predicted outcomes to the actual outcomes. A priori is italicized because once one has seen the outcomes, any changes to the prediction mechanism will generally require re-characterization of the error using data that has not been seen. This point is critical and is discussed further in the next section. If one cannot perform such a characterization, then one is making a forecast.

As defined here, the difference between prediction and forecasting is independent of the prediction mechanism. One may use human instincts to make predictions. As long as the error associated with the instinctive prediction mechanism can be characterized on a consistent basis statistically, confidence levels on the error can be produced. On the other hand, one can use large quantities of historic data to optimize coefficients in a sophisticated mathematical model that generates future outcomes without characterizing the error. This is a forecast based upon modeling error - further described below.

If there is no history data, one cannot characterize prediction error, and therefore one must make forecasts. This is true when new problems are being addressed that may not fit the existing prediction mechanism. In these cases, one must determine whether the changing situation still fits the prediction mechanism, or whether it is time to drop the error characterization and confidence statements and go with a forecast. This determination is relatively easy to do when the prediction mechanism is a mathematical model driven by mechanically quantified measured data. This becomes difficult when characterizing prediction error based upon human instinct.

In Chapter 7, we will indicate a method for combining forecasts and predictions to produce a prediction. This method will rely on the characterization of worst case outcomes, i.e., outcomes that occur based upon worst case conditions. In effect, we can condition probability statements using parametric worst cases.

The Prediction Problem

Prediction of future outcomes of systems must be couched in terms of probability statements. In fact, they are conditional probability statements. For example, one can predict whether or not it will rain tomorrow in Point Pleasant, on the coast of New Jersey, as follows:

> ***First Prediction:*** Given the historic data on rainy days in Point Pleasant for the past 20 years, one can predict the probability of rain by dividing the number of rainy days by the total number of days. If the number of rainy days for the past 20 years (7300 days) was 730, then the probability of rain tomorrow (or any day for that matter) is 10%. This is a statistical estimate based upon historic data.
>
> ***Second Prediction:*** Given the number of rainy days in each month in Point Pleasant for the past 20 years, one can make separate predictions of the probability of rain for each month. For example, one might say that - if the month is July, then the probability of rain is 5%; - if the month is November, then the probability of rain is 15%. This is also a statistical estimate based upon historic data.
>
> ***Third Prediction:*** Using knowledge of the weather patterns around the coast of New Jersey, one can generally rely upon the fronts moving from west to east. Given knowledge about a rainstorm heading toward Point Pleasant from Pennsylvania, one can predict the probability of rain over the next 24 hours in 6 hour increments. For example, one might say that the probability of rain is less than 2% over the next 6 hours. It is 25% for the following 6 hours. It continues to rise to 95% in the period 13 to 18 hours from now, and then falls off to 65% in 19 to 24 hours. This requires a dynamic prediction model.

All of these predictions may contain valid probability statements based upon historic measurements. However, the accuracy of each is obviously different. The difference in accuracy is determined by the conditioning of the probability statement. The first prediction is conditioned only upon the number of rainy days in a year, with no additional information. The second prediction is conditioned upon additional information, i.e., the number of rainy days in each month of the year. It will be a more accurate statement. The third prediction is conditioned upon a dynamic model of weather patterns. This model contains much more information than the other two, and is much more accurate.

Predictions are statements of probability of the outcome of a future event. In general, they are conditional probability statements, i.e., they are conditioned upon the information used to compute the probability. The more information one can use to condition the probability statement, the more accurate the prediction.\

Human Judgment Versus Automation

The general prediction problem is to produce a sufficiently accurate prediction given the time frame and resources at one's disposal. As indicated above, predictions are probability statements that are conditioned on all of the information one can obtain. Mathematical formulation is not as important as having additional information and correctly quantifying the prediction error.

The classic example of additional information is that of the salesman who knows little about mathematics and uses a computer spread sheet to organize his forecasts of sales volumes of product lines for the next quarter. The numbers come from his head. The marketing department gets independent sales forecasts from a set of PhD statisticians who use various sophisticated statistical approaches and historic data to forecast the same sales volumes. Why does the salesman consistently come up with a much more accurate forecast? He has more information about what's going on in the market!

As indicated above, predictions are conditioned probability statements. Modelers that incorporate more information into their model will produce more accurate predictions. This information need not be in the form of historic data. It is likely that the most important information is knowledge about the structure of the system. That's why the salesman does better. He knows what is happening in the market (his system). If he's good, he has intelligence on what's changing. Are some new stores opening in two months that will be buying? Are some existing clients about to shut down? He has a more accurate model in his head than the statistician who is manipulating historic data with time-series models.

This does not imply that we cannot build a model on the computer that incorporates the salesman's knowledge. In fact, we can generate probability statements conditioned on that knowledge. If we had 100 territories each with a salesman, we could build one model with 100 instances and get them to enter their knowledge and then roll up the results - automatically. Can we get them to cooperate? Yes, if we can improve their accuracy and still make it easy for them to enter their knowledge. These are the practical problems that must be dealt with, and the questions that must be answered.

Drawing The Line Between Human Judgment And Automation

How do we decide what's best for a human to do versus using a computer? This problem has been addressed for many years in the field of CAD. One must answer the questions: What processes depend upon - or are best left to - human judgment? Where are the break points where computers do better? The answer to this question generally comes down to time. Given the requirement to achieve a given quality of results and objectives, e.g., being able to meet specified accuracy requirements, how much time will it take to get to the desired accuracy?

When building tools to help people solve design problems or make complex planning decisions, time enters into the picture in two major ways.

- Development Time - the time it takes to develop a tool that can be used to automate what was heretofore done using human judgment. This does not necessarily eliminate human judgment as an override.
- Solution Time - the time it takes to get useful solutions from an automated tool. For systems of interest here, this generally implies getting a fast response - in real time.

Both of these times depend upon the state of technology. However, there are general principles that apply when trying to decide upon an approach. These principles assume agreement upon the quality or reliability requirements, as well as the development time and solution time requirements. For complex applications, particularly those involving extreme reliability requirements, one must go through a careful learning process to establish accurate assessments of these times. This often involves an evolutionary approach, where parts of a system are automated while others evolve more slowly. This usually saves time in the end.

Figure 1-3 provides an illustration of how the level of automation achieved tends to grow over time in various applications. This is due to the process of learning about the *unknown* unknowns as well as the *known* unknowns. Some applications have achieved a high degree of automation quickly. These tend to have a high degree of rote functions. Some have to wait for technology to catch up to be practical. Others have clear limits in terms of % automation, at least with foreseeable technology.

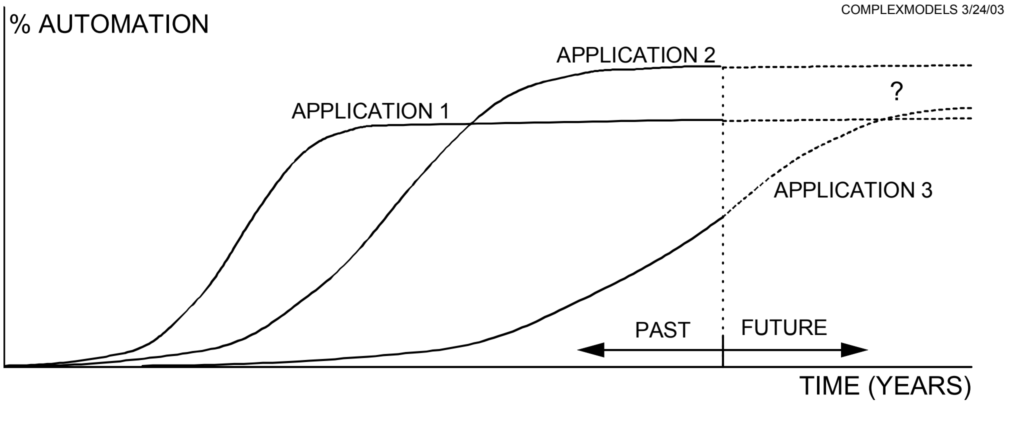


Figure 1-3. Level of automation achieved (past) and predicted (future) for various applications.

In order to make these decisions, one must clearly define the problem. This sounds obvious, but when it comes to automation of systems heretofore dependent upon human judgment, it is imperative to carefully define the problem, since human intelligence may not be around to take care of the mistakes that may take years to uncover.

Defining The Prediction Problem

Figure 1-4 provides an overview of an actual system and a corresponding model to be used for prediction. When building models to predict the future, it is most important to clearly perceive the differences between the inherent properties of *real world systems*, their related *observation data*, and the *models* which people use to describe them. As is often the case, such an obvious conclusion tends to be ignored. We will emphasize these differences throughout this presentation, and also the corresponding differences between *prediction* and *estimation*. As normally taught in statistics courses, estimation assumes that the real world system can be described as a population. This is certainly true if our concern is characterizing all that is known about a system to date. However, as soon as we look toward the future, we cannot use "standard" estimation theory unless the system is stationary by "standard" definitions, as we shall show in Chapter 6.

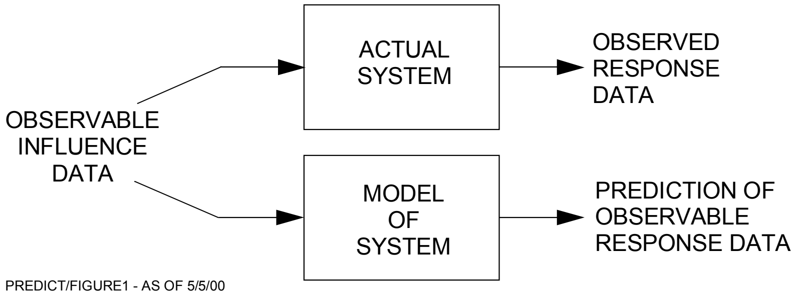

ACTUAL SYSTEM
OBSERVED RESPONSE DATA
OBSERVABLE INFLUENCE DATA
MODEL OF SYSTEM
PREDICTION OF OBSERVABLE RESPONSE DATA
PREDICT/FIGURE1 - AS OF 5/5/00

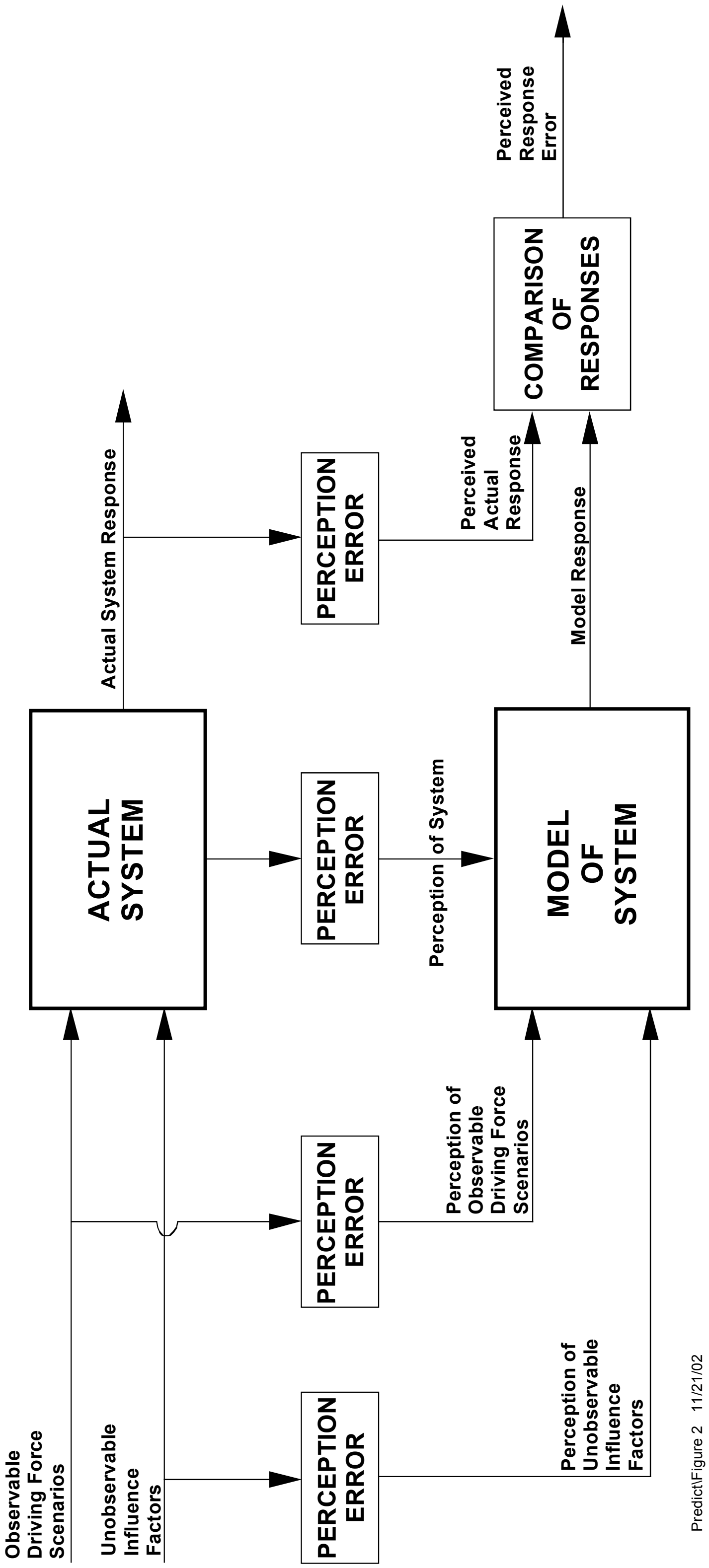
Figure 1-4. General form of a prediction model.

When trying to build models of real world systems to predict their future responses, we must use any *causal* properties (properties that relate cause and effect) that we can derive from knowledge of the system. In doing so, we are seeking delays between external causal factors that we can observe, and their effect on system response. This approach is not to be confused with that of finding statistical correlation between some observable time-series and the system response. *Correlation does not imply causality.* Furthermore, if we restrict our correlation tests to *linear* relations, we will likely not be able to uncover the correlations between causal factors and the response of a *nonlinear* system.

When pursuing the search for models that properly relate cause and effect, one must also be aware of the errors in perception that typically occur. These are categorized in Figure 1-5. At first, the possibilities for error may look exaggerated. They are not. It is up to the modeler to make as much use of observations as possible, while minimizing the likelihood of perception errors. This is a real challenge.

From the above observations, it should be clear that building models of real world systems is a very difficult endeavor. As the systems we try to model become more complex, and particularly if they are nonlinear, the challenge is great. Models of business and economic markets are difficult to build and validate. The intent of this work is to lay the foundation upon which to build a knowledge base for successful modeling of these types of systems.

CATEGORIZATION OF PERCEPTION ERRORS



Observable Driving Force Scenarios
Unobservable Influence Factors
ACTUAL SYSTEM
Actual System Response
PERCEPTION ERROR
PERCEPTION ERROR
PERCEPTION ERROR
PERCEPTION ERROR
Perceived Actual Response
Perception of System
Perception of Observable Driving Force Scenarios
Perception of Unobservable Influence Factors
MODEL OF SYSTEM
Model Response
COMPARISON OF RESPONSES
Perceived Response Error
Predict\Figure 2 11/21/02

Figure 1-5. Categorization of possible perception errors.

To provide the foundation for building models to predict the future, a set of definitions is proposed that leads to a general formulation of the prediction problem for a broad class of applications. These definitions are based on a measure of prediction accuracy, defined in Chapters 2 and 3, which is independent of methods used to produce predictions. *Real-time prediction error* (that encountered as live data becomes available) is contrasted to the *estimation error* encountered during model identification (parameter estimation using historic data). It is clear that typical measures for estimation error do not apply directly for prediction error. This will be seen in Chapter 8 when we compare the conditional probabilities before and after the observed response data becomes available. Validating statements about prediction accuracy represents a more difficult problem, requiring careful attention to "hiding" data from the modeler, a concept which is clearly at odds with the idea of characterizing a population. This topic is very briefly addressed in Chapter 4.

In Chapters 5 through 7, definitions are examined for distinguishing between stationarity of deterministic and statistical functions, consistent with those used in physics and engineering. The definitions provided here are also consistent with those accepted in the forecasting literature, but they have been devised to extend their usefulness in defining the prediction problem.

Statistical approaches to prediction, as described by Box and Jenkins, [4], Harrison and Stevens, [16], and others, depend upon stationarity of the data to be predicted. Relaxed forms of stationarity are also admissible such as with approaches using time-varying coefficients, see for example Mehra, [21], and Rosenkranz, [26]. But these "quasi" stationary forms are shown to also depend on stationarity assumptions. Definitions are provided for classifying these approaches in Chapter 6. Underlying methods are described in Chapters 7 and 8 for constructing models to predict responses whose history data is nonstationary, apparently random, and need not be characterized statistically, i.e., the distribution functions are considered unknown but bounded, as in Fisher, [11], and Schweppe [27]. Chapter 9 provides a theoretical definition of the prediction problem, and the requirement for ensuring that the data used to measure prediction error is properly used.

Chapter 10 provides some practical suggestions for building prediction models that take maximum advantage of the available information. including that based upon human judgment. Commonly used models are discussed and compared relative to their usefulness and shortcomings. Also covered are the use of optimization and estimation techniques to determine model parameters that maximize prediction accuracy.

Examples are used throughout to demonstrate how knowledge of the structure and dynamics of a particular system can be used to build models which introduce additional information above and beyond that available from the observation data. This additional information can serve to further condition probability statements, and increase the accuracy of predictions. In fact, it is this additional information, that is not available in the normal "observation data," which allows us to more accurately predict the future responses of a nonstationary system.

Stochastic Nature Of The Problem

There are various levels of planning and prediction in a large complex control system. For example, one may have to determine the "best" approach to moving vehicles or supplies from one location to another. This can be treated as a classic transportation problem. Given the knowledge of available transportation facilities, air, sea, and ground routes, and the myriad of other factors affecting the time and energy required to complete the move, one can apply standard techniques, e.g., Linear Programming (LP), to come up with the *best* solution. But is *best* good enough? If the potential outcomes can cause mid-air collisions or power grid failures, then one may be dealing with six-sigma (or greater) probabilities.

Dealing With Variations

An LP approach may be excellent for coming up with answers to a problem as posed. However, by itself, it may not deal with the stochastic nature of such a problem. In dynamic systems, every attribute may be subject to variations. In the transportation problem, these variations can be due to traffic, weather, breakdowns, etc. These variations can be taken into account in various ways. For example, traffic may be predictable based upon day-of-week and time-of-day. Even so, traffic can get tied up due to special events. The time to get from point A to point B can be adjusted by a traffic variable. The traffic variable can be a function of the calendar and clock. Actual traffic can also vary around a mean value due to effects that appear random, and are therefore unpredictable. All such variations must be accounted for when determining whether the solution meets the time constraints.

To generalize the approach to characterizing traffic, each route may have variations in time that can be broken into two categories, those that are predictable based upon observable attributes, and those that appear to be random. If we can develop relationships between the predictable variations and the observable attributes, they can be applied to adjust the mean value. This serves as additional information to reduce the prediction error.

There are approaches for characterizing the effects due to the random variations. The approach used most often is Monte Carlo Analysis. In this case, distributions are postulated for all of the random variations. Then a simulation is run with random samples drawn from these distributions each time an event occurs requiring a value for the variation. Depending on the scenario, one runs enough simulations to characterize the distributions of the resulting measures of performance. For example, total time to move the trucks from A to B may involve many traversals of many routes. When these individual traversals are simulated, they are subject to the variations determined by the random samples. If a new random number seed is used for each simulation, different results will occur for the total time measure.

After enough simulations are run, a histogram can be used to characterize the distribution of total time. Consider Figure 1-6 as the resulting distribution representing the time to move trucks from A to B. If the simulations took into account all of the variations present in the real environment, then one can derive a probability statement about the range of time. For example, if Tmax is 20 hours, and the area under distribution D1 up to Tmax is 95% of the total, one can state that trucks can be moved from A to B in 20 hours with a 95% probability.

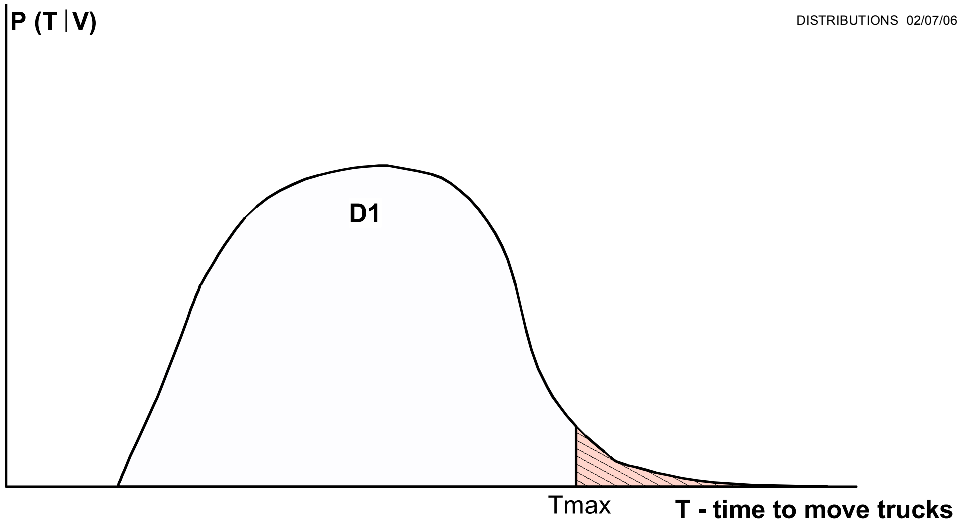


Figure 1-6. Example of a measure of performance characterized by a distribution.

The probability statement comes directly from a mathematical calculation based upon the distribution. If the distribution represents the real world perfectly, then the probability statement is correct. One must ask how accurately the distribution represents the real world. This is answered by providing a confidence level in the distribution relative to the calculations being used. This is also described in [11]. In addition, there is a more direct way to get to a solution without using Monte Carlo. This is described below in Accounting For Constraints.

Dealing With Large Decision Trees In Time

Complex planning and control systems require a large number of decisions to be made over time. In many cases, many decisions must be made to start or continue an operation before any results can be seen. Decisions may involve selection of an approach from many choices. One decision may lead to another next decision level where more selections must be made. Just considering the sequence of decisions coming from each level in a control hierarchy, one could envision a very complex picture of this process.

As observations come in, decision makers, with the help of their staffs, must assess changes in a situation, and make corresponding changes in plans and follow on decisions. At each step along the way, at different levels in the decision hierarchy, the characterization of effects achieved can be represented by a distribution as in Figure 1-7.

Although Figure 1-7 looks like a *normal* distribution, the large number of variations that one may be faced with may not be characterized. To use a Monte Carlo approach, or the worst case design approach defined below, the distributions can be unknown but bounded. If the Tmax boundary in Figure 1-6 is known, i.e., we know where the 95% point lies, that's all we need to know. We need not know the shape of the distribution. But in many cases we don't even know Tmax and must provide an estimate.

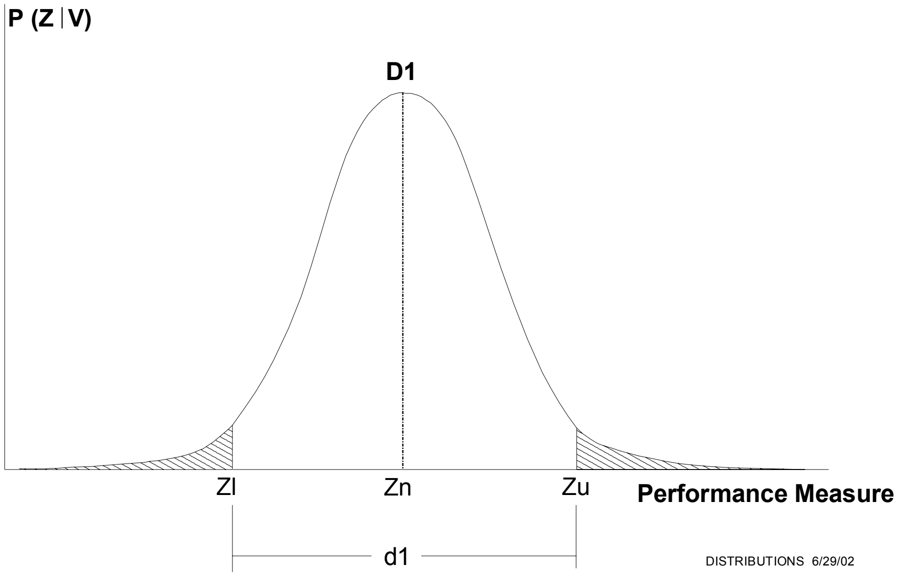


Figure 1-7. Desired effects characterized by a distribution.

Considering all of the possible variables and characterizations, one may feel overwhelmed by what would appear to be unpredictable chaos. But operations do unfold according to rules. The rules may be changing, but some level of rules and coordination is required to achieve a desired level of effectiveness.

Accounting For Constraints

The need for rules and coordination in operations imposes constraints on behavior. This need increases with the tempo of operations. In addition, real world systems are nonlinear, imposing additional boundaries of constraint. Behaviors never get to infinity. Something breaks down first. In addition, there are different levels of constraint violation and corresponding actions that must be taken, and these can be bounded in terms of their outcomes in time. Figure 1-8 illustrates a (rose colored) envelope generated by six sigma points of successive distributions in time.

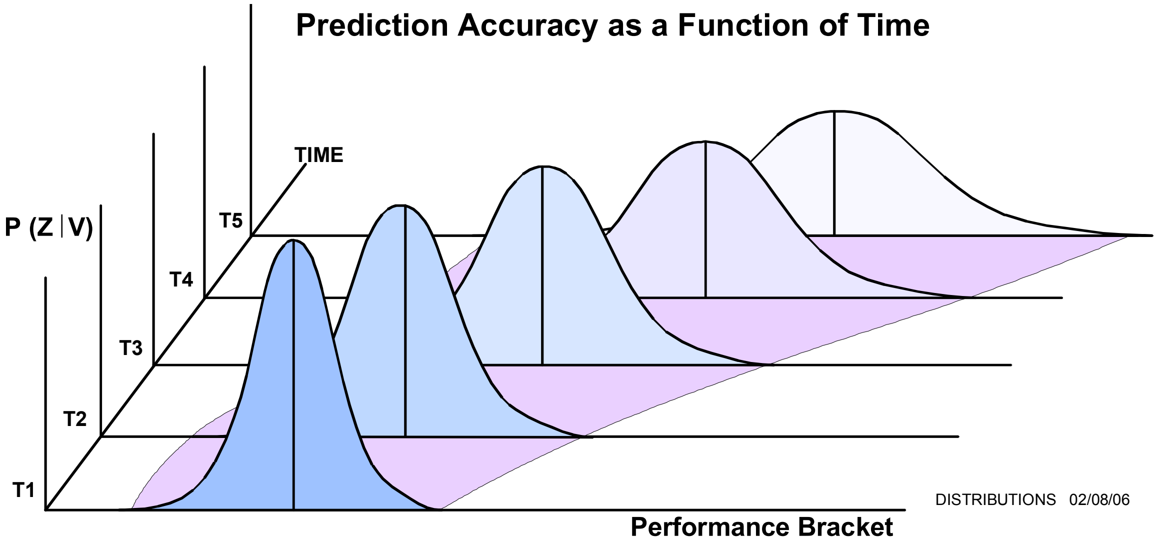


Figure 1-8. Prediction of effects characterized by a distribution envelope.

Analyzing the potential trajectory that events may follow as they unfold in time, and the way they may contribute to variations in potential outcomes, provides an improved understanding of how one may want to proceed to reduce the risk of a failure or catastrophe. Operators can lay down operational constraints, e.g., *this action must not take more than some specified amount of time*; or, *this flight must land before that flight can take off*. All of these constraints serve to bound the problem.

2. MEASURING PREDICTION ERROR

When building models to predict the future responses of a system, a number of questions arise that can lead to confusion when comparing the accuracy of one prediction to another. The following discussion is an attempt to surface and resolve these questions in a manner consistent with standard statistical practices. In this discussion, we will use t_c to denote the current time.

Probably the most important distinction between the problem of prediction (determining the system response at $t > t_c$) versus the problems of filtering (determining the system response at $t = t_c$) and smoothing (determining the system response at $t < t_c$) is the means for measuring "optimality" with respect to the data, and a corresponding error criterion. Referring to Figure 1-4, the only "true" test of a prediction model is to drive it with observable influence factor data currently available (at $t \leq t_c$), make a prediction, and then compare the predicted and observed outcomes in "real time" as they occur. This must be performed in such a way that future outcomes cannot be used to influence the error. If a model is influenced by the modeler after having seen the "future" data, any measures of *prediction* error are subject to "contamination," i.e., the measure of error may not fairly represent the ability of the model to predict the *real* future.

Note that this approach is quite different from that of many forecasting techniques that use influence factor data predicted by another source to forecast what will happen in the future. Using this approach, models are built which simulate what will occur given the factors predicted by others. We will refer to this technique as *simulation,* since the model *simulates* what would happen if the predicted data accurately represented future outcomes. However, the accuracy of this approach clearly depends on the predicted data as well as the simulation model which uses it. The error measurement used to characterize the accuracy of such a simulation model is usually performed using the real values of the predicted variables based on actual past history. These models can be very "accurate" if the predicted factor data is accurate. However, if the predicted factor data upon which the model depends is very inaccurate, so is the resulting prediction.

For convenience, we define *prediction error* as the error encountered when making predictions in "real time," or the equivalent thereof, without interaction by the modeler to readjust prior predictions. In other words, prediction error, as defined here, can only be measured using data for $t > t_c$ which was not available to the modeler. If any link exists which allows information to be derived by the modeler from the future data set ($t > t_c$) upon which error is to be measured, then prediction error cannot be fairly measured. *Modeling error*, on the other hand, is the error measured when using history data ($t \leq t_c$) to estimate model parameters during the model identification process. This definition of prediction error does not impair the use of adaptive models, since they only use currently available data to predict future responses.

It is worthwhile to remark that, once a modeler has seen "future" test data, it is difficult to determine that he will not in some way use this information during a model identification process that precedes the measure of prediction accuracy on the same "future" data set. From practical experience, there can exist a significant difference between modeling error encountered when "fitting" the history, and prediction error. It is not uncommon for prediction error to be twice as big. This is the case when using the simulation technique described above. The simulation modeler can say his model is very accurate, and that it was the data that came from the prediction source which caused the error. However, the management who made decisions based on this model knows only that the forecasted result was anything but accurate.

As we can see from this example, the definitions used to characterize prediction accuracy are critical to our understanding of the causes of error in our models. It must be emphasized that the particular *measures* of error used for estimation (model identification) and prediction may be identical. It's the difference in the data sets that cause the difference in the errors. The data set used for prediction cannot contain information beyond the current time (i.e., $t > t_c$), else we are performing simulation. However, when performing model identification, we will certainly be predicting values for which we already have answers, else we cannot measure the model error.

It is clear that typical measures of estimation error do not encounter the constraints imposed upon the measure of prediction error. To eliminate confusion, a distinction will be made throughout this book between estimation and prediction. This distinction will be based upon the conditional probabilities used for estimation and prediction. In the case of estimation, the probability of the value estimated is conditioned upon *all* available data, i.e., the population concept is valid. No concern exists about the separation of "future" data upon which the accuracy of the estimator is to be tested. In the case of prediction, the probability of the value to be predicted can only be conditioned upon data up to the *current time*. "Future" data, to be used for measuring prediction error, must not enter into the conditioning of the probability. These probability statements are addressed in the next chapter.

3. CHARACTERIZING PREDICTION ACCURACY

In addition to the above problem of measuring prediction error, characterizing prediction accuracy for the purpose of comparing two models presents a further complication. This results from the need to introduce probability statements about the accuracy of prediction. To illustrate this problem, we offer the following example. If we were buying predictions from two commercial services, and each makes "point" predictions (i.e., they provide us with a single number), we have no way to tell who is going to be best without keeping our own measurements over time. For example, if one service predicts the future response will be 14, and the other predicts it will be 18, there is no way to compare the accuracy of their predictions before the response actually occurs, since no measure is provided with the prediction.

Specification of a Prediction Envelope

The solution to this basic problem is best addressed by a more detailed example. Assume we want to compare two prediction services, A and B, who provide weekly predictions of money supply (M1). Each service provides predictions over a 12 week future horizon using an 80% probability *prediction envelope*. An example of such an envelope is shown in Figure 3-1. It is composed of a sequence of intervals for each of the prediction horizons $\tau_p = 1, 2, ..., 12$. Each service claims that "80% of the time," future values of M1 will fall within their envelope.

Service A points out that B is not meeting its probability criteria since over the last 6 months (26 time steps), the actual values of M1 have fallen outside of its $\tau_p = 12$ prediction interval (farthest out horizon) 6 times. Refer to Figure 3-2. Therefore, it should have been called a 77% envelope (at best) since actuals were outside slightly more than 23% of the time.

B counters by saying that 26 weeks is an insufficient time period to characterize the probability. B then points to its ten year track record which shows that actuals have been inside the 12 step prediction interval better than 80% of the time. In fact, at $\tau_p = 12$, they have been in 81.5% of the time.

A states that B is riding on its old laurels. That, in fact, it had a great model 5 years ago but, over the past few years, its accuracy has degraded. B immediately recognizes that the marketplace is most concerned about the current history. Everyone knows that the basic structure of markets can change, so it decides to research the problem. The first decision to be made is what "looking back" horizon into the past, τ_b, to use to characterize its probability statement. Obviously, the shorter the horizon, the more appealing to the marketplace. After much thought, B concludes that it must consider horizons on a quarterly basis, and that a single quarter might be watched, but that two quarters (26 weeks) is probably the shortest realistic time period from a "statistical" standpoint.

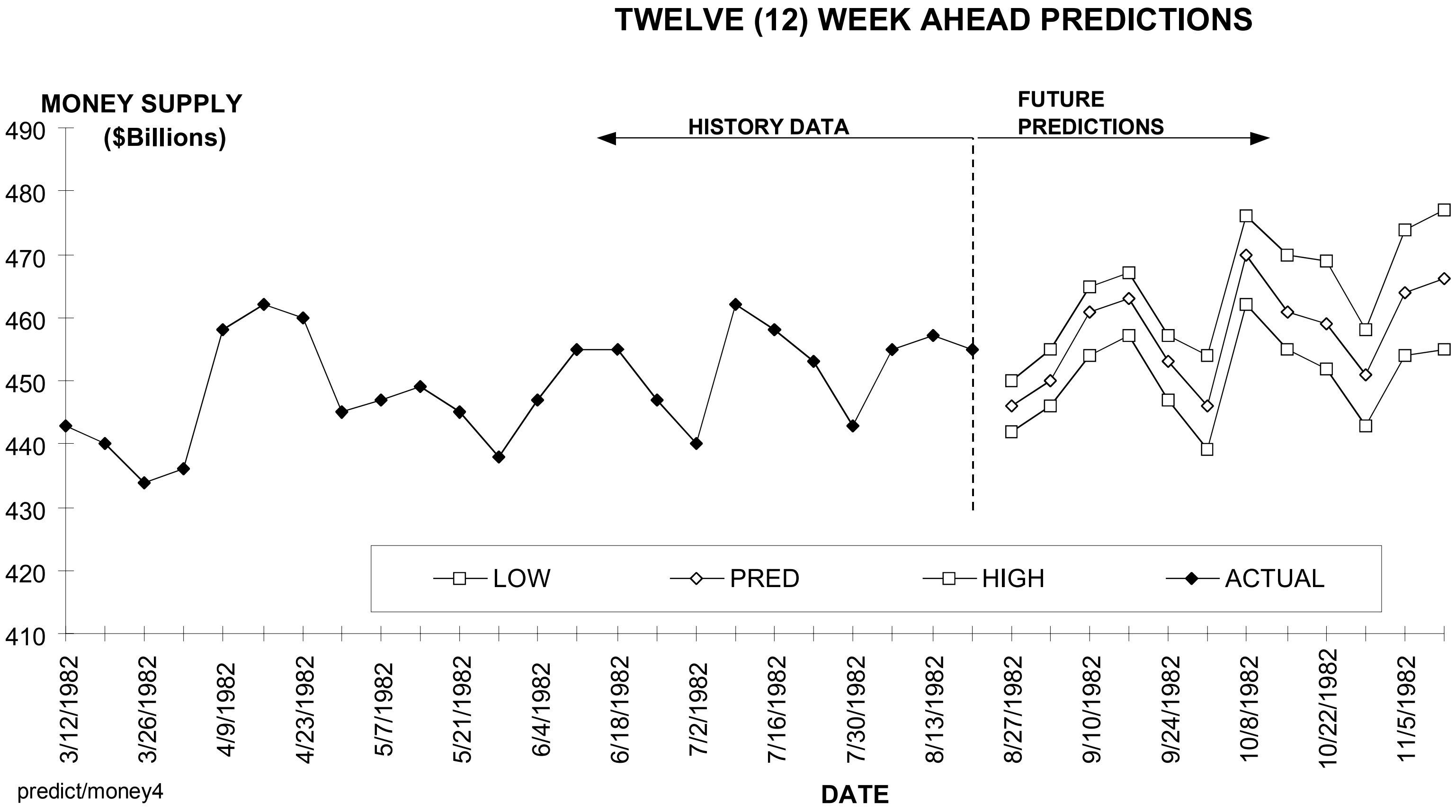


Figure 3-1. Twelve week ahead predictions of M-1, not seasonally adjusted.

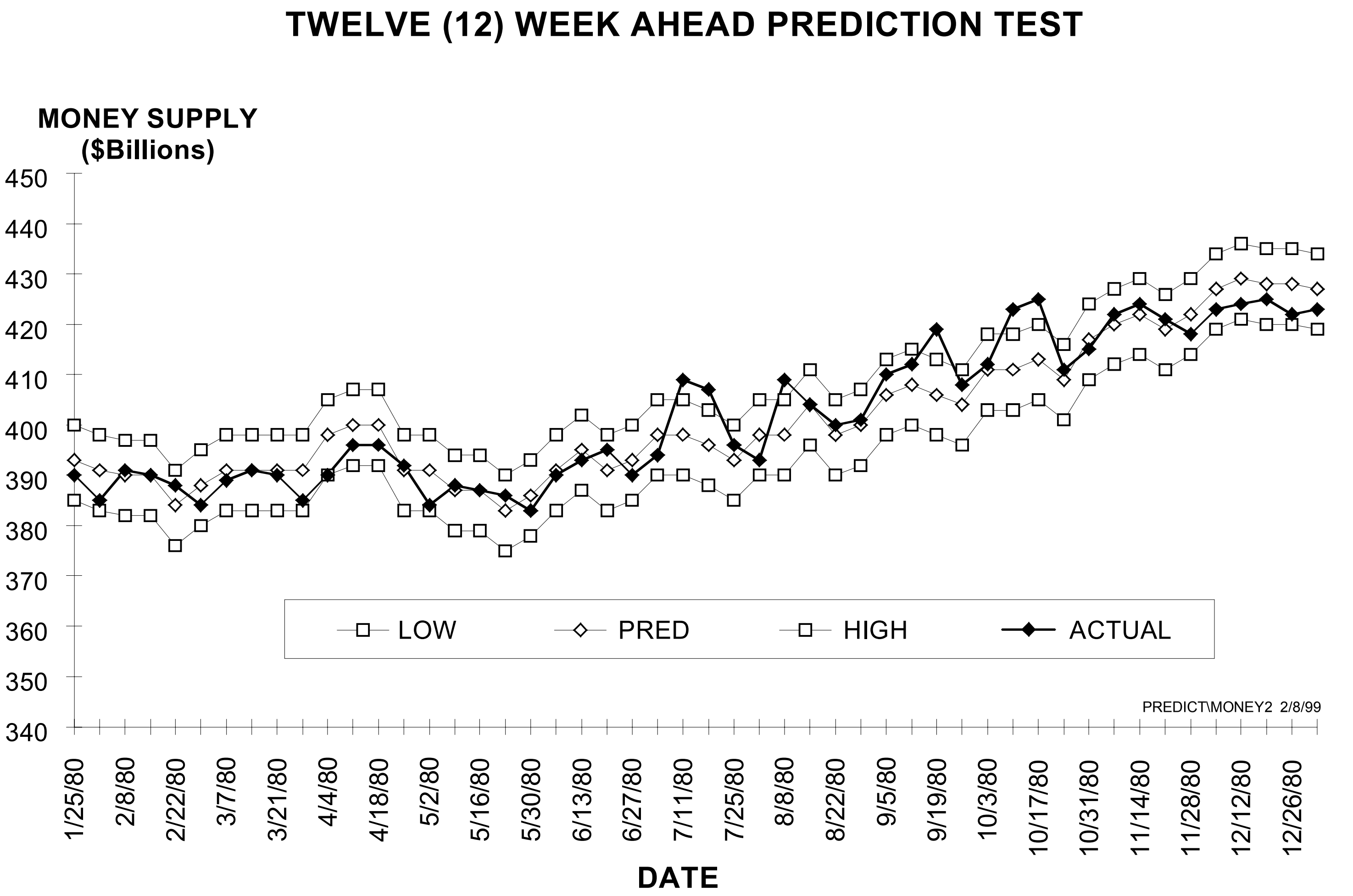


Figure 3-2. Twelve week ahead prediction for 52 weeks.

To characterize the statistics for the above problem, the following definitions are offered.

τ_p - is the number of future time steps from the current time step to the future time horizon for the which system response is being predicted.

τ_b - is the number of past time steps from, and including, the current time step to the looking back horizon, used to define the probability statements.

n - is the number of mutually exclusive "τ_b" sample sets (ensembles) of history data available for testing the probability statement.

In other words if N is the total number of sample points (weeks) of history data, then

$$n = \frac{N}{\tau_b}$$

As B modified its model to ensure the truth of its 80% probability statement, it determined that there were certain sample sets of τ_b weeks for which it was very difficult to support the 80% level. Upon checking A's predictions, it was determined that they too were "out of bounds" during these periods. In fact, A was now outside the envelopes more than 20% of the time. And, this probability increased as τ_b became smaller.

Measuring Confidence in the Prediction Envelope

From the above sample problems we derive the following conclusions. When making statements about the probability that future outcomes will lie within a given envelope, we must pick a specific looking back horizon, τ_b, to test the probability statement. Next, we must consider all possible sample sets from the history data which contain τ_b *contiguous* samples. (There will be N - τ_b + 1.) We can then plot the distribution of the number of times the actual values fell inside the envelope for a given horizon. See Figure 3-3.

Assuming this distribution is representative of the future, we can compute the probability that the actuals will fall inside the envelope at least 80% of the time. This provides a confidence statement about the 80% probability envelope. For example, we might conclude from Figure 3-3. that

$$P\{X \geq 80\%\} = 0.95.$$

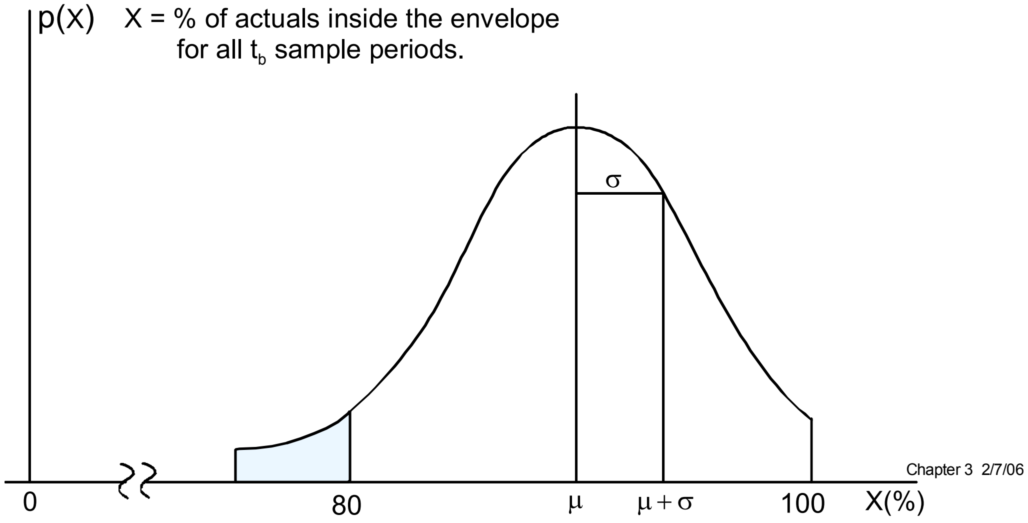


Figure 3-3. Statistical distribution of the number of times the predictions fall inside the envelope.

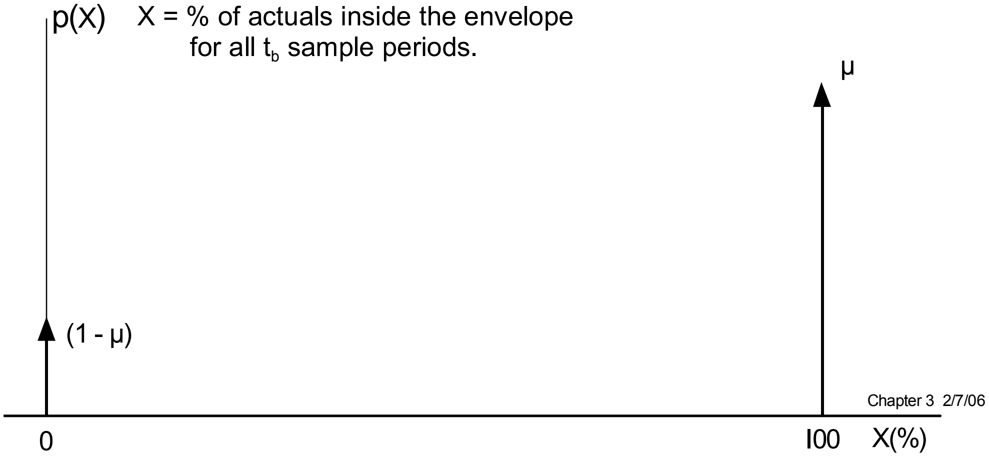


Figure 3-4. Distribution when the looking back horizon, τ_b, equals one.

We note that as $\tau_b \rightarrow N$, $\sigma \rightarrow 0$, and μ represents the probability statement that would be perfectly correct for the entire history. Conversely, as $\tau_b \rightarrow 1$, σ expands so that the distribution has finite probabilities at 0 and 100%, and zero probability everywhere else, refer to Figure 3-4. Ideally, for a τ_b of reasonable size, we would like to see the standard deviation as small as possible. A small standard deviation would indicate that the probability statement varied little from time period to time period. However, to achieve this may require a large value for τ_b, which the market for predictions may question.

Our goal is to develop measures of accuracy that also serve to measure consistency of the model for small looking back horizons over long periods of history. This can be accomplished using confidence intervals about the prediction envelope boundaries for a given τ_b. In general, for any given τ_b, we can determine the confidence level (e.g., 95%) for which we will be inside the (80%) envelope. Assuming that the distribution in Figure 3-3 were normal, then maximum consistency can be achieved by minimizing the variance, or the mean absolute deviation, given a desired looking back horizon, τ_b, and probability prediction envelope, e.g., 80% .

A Measure Of Prediction Quality

Using the above definitions, we can now pose a measure of *quality of prediction* that accounts for the actual width of the envelope for a given probability (e.g., 80%). The following measure is offered for a particular forward prediction horizon, τ_p, and looking back horizon, τ_b.

$$Q(\tau_p, \tau_b) = \frac{C*P}{1 + W}$$

where: - Q is the measure of prediction quality,

- P is the probability that future values will fall within the envelope at a given τ_p (80% in the above examples),

- C is the confidence in the value of the probability statement for a given τ_b (95% in the above examples),

- W is the mean normalized width of the envelope, relative to the actual value, at τ_p.

Using this measure, quality improves (degrades) with increasing (decreasing) probability of being inside the envelope, and with increasing (decreasing) confidence in the probability. It also improves (degrades) as the width of the envelope grows smaller (larger). As the statement of probability of being inside the envelope approaches unity (100%) and the confidence in the statement approaches unity (100%), and the width of the envelope approaches zero, quality approaches unity, and predictions approach certainty.

4. OPTIMIZING AND VALIDATING MODELS OF SYSTEMS TO PREDICT THEIR FUTURE RESPONSES

If prediction accuracy cannot be measured using data available to the modeler, then how can the modeler optimize his model to maximize prediction accuracy? This question involves the relationship between optimization and validation of models to predict the future. To answer this question, we must understand that the modeler's *mathematical measure* may be the same for both optimization and validation. However, the *data set* available to him during model identification only allows him to minimize model error. In other words, having identified his model by minimizing the difference between model prediction and *known* "future" data (an error measure), he has conditioned his probability statement on that known data. He must use a new *unseen* data set for measurement of prediction error.

The model identification process can be achieved using a deterministic approach, a statistical approach, or a combination of both. The next sections describe these approaches, and their fundamental differences. In practice, a combination of the two will likely be best. However, the order in which these are approached is important, as should be apparent from the following sections. In general, one should build a model structure from deterministic knowledge of the mechanics of the system. After this knowledge is exhausted, one usually resorts to a statistical approach to optimize internal model parameters (coefficients) which cannot be obtained deterministically. Given that the modeler has knowledge of the workings of the system, and the skill to build the model structure, scarcity of data for model validation remains a great cause for concern.

The following sections provide concepts for building and validating models to minimize prediction error.

Statistical Models for Prediction

We will start with the statistical approach to building models to predict the future. This will serve to emphasize the importance of taking the deterministic approach as far as possible before resorting to the statistical approach. We assume that a model structure has been developed using the deterministic approach which effectively takes the form of mathematical functions or rules that relate the observable influence factors to the future response of the system. Using the history data of both the influence factors and the response, one can try to find values of prescribed model coefficients, that have been left as unknown parameters, to minimize the difference between model predictions and observed responses. Typical performance measures for modeling error are the mean absolute deviation and mean square deviation. These same measures can be used for validating the model to determine its prediction error. However, as stated above, a new *unseen* data set must be used to perform the prediction error tests. Thus, the model cannot be optimized while measuring prediction error.

It is therefore necessary that the modeler establish correlation between modeling error and prediction error. Otherwise, he has no measure for deciding on ways to improve his model, and could not expect improvements in prediction accuracy on other than a "random" basis. This correlation can only be ascertained by successive experiments in real time, or with sufficient history which has been "hidden" from the modeler. The problem is further complicated because the correlation measure must depend, in general, upon model parameters being optimized and, by definition, correlation must be done after the optimization has been completed. Once prediction accuracy is measured over a given data set, that data set cannot be used again to measure prediction accuracy by the same modeler. Only the correlation can be used.

From the above facts, one sees the difficulty with the statistical method, i.e., data for validation purposes can be consumed quickly. This makes the deterministic approach a critical part of the model building process. One must try to take the deterministic approach as far as possible without optimization, leaving the statistical approach till last, as more of a validation effort than an optimization effort.

Deterministic Models for Prediction

The deterministic approach consists of formulating models from knowledge of how the system operates internally. The modeler tries to determine the rules that take in the influence factors and cause the future responses. Determining these rules - the internal operation of the system - is the key step to incorporating maximum information in a prediction model. This implies conditioning the probability statement on more information.

If a modeler can create a model of how a system translates the observable influence factors into future responses without looking at any history data, then the history data is available for validation of the model. If, on the other hand, one takes all of the data and performs statistical fits, that data is no longer useful for validation purposes.

Loss of data is not the most important reason for using the deterministic approach first. Representing the underlying *cause and effect* relationships internal to the system is most important. As will be seen in the following chapters, the system itself must have certain properties that make it predictable. These properties depend on delays and time constants that are inherent in the system, being the causal properties that relate the observable influence factors to future responses. Unless the modeler can represent these properties, accurate prediction of nonstationary system responses cannot be accomplished in a reliable manner.

This approach is not to be confused with that of finding statistical correlation between some observable time-series and the system response. *Correlation does not imply causality*. Furthermore, if we restrict our correlation tests to *linear* relations, we will likely not uncover the correlations between causal factors and the response of a *nonlinear* system. The problem of finding nonlinear transformations does not lend itself to a naive or "black box" approach as can be used with linear systems. The modeler must resort to an understanding of the mechanics of how the system operates.

General Approach to Building and Validating Prediction Models

The following steps provide a summary of the proposed approach to building and validating models to predict the future responses of nonstationary systems.

1. Build a deterministic model that characterizes the inherent cause and effect properties of the system, i.e., a structural model that characterizes those properties that translate observable influence factors into predictable system responses.

2. Using history data and optimization techniques, find values for any remaining unknown parameters that minimize measures of model error.

3. Using new "hidden" data, validate the model, measuring prediction error.

4. As more data and *cause and effect* knowledge become available, repeat the process.

5. As the process is repeated, try to obtain correlation between changes in model error and prediction error.

6. This correlation can be used to guide additional steps toward improving prediction accuracy. Obviously, if reductions in model error do not correspond to reductions in prediction error, much data can be wasted in this process.

The most difficult step in the above process is building the deterministic model. Representing the structural properties of a system which afford accurate prediction of future responses is the key. These structural properties may take the form of rules or algebra. The algebra may typically take the form of dynamic difference or differential equations. In fact, the state space framework used in physics and engineering is well suited to characterizing models of this type using an algebraic approach. The next few chapters are aimed at providing the background for attacking this problem in an organized way, starting with a description of the state space framework.

In recent years, discrete event simulation has opened the door to a discrete systems theory. An extended version of this approach allows one to write rules that govern judgments and decision processes in an English-like language, with algebraic expressions mixed in. This new paradigm for modeling complex systems has been used extensively and successfully by PSI to predict results of communication systems testing. It has been demonstrated that the extremely complex models needed to obtain acceptable prediction accuracies could be built much more easily using this new approach. In fact models that were heretofore intractable were easily developed and tested. Although the treatment in this book follows a traditional mathematical approach, the theory derived applies directly to the discrete event paradigm.

5. A GENERALIZED MODELING FRAMEWORK

The State Space Framework

A State Space framework, commonly used in engineering and physics, [1], [10], [14], [19], [27], [31], is most convenient for defining the prediction problem, as well as the framework upon which to build the structural models themselves. We start with the basic definitions. The *state* of a system is defined as a set of values that, along with the input driving forces to the system, are sufficient to describe the behavior of the system, reference Figure 5-1. This framework has been shown to encompass the most general modeling problem, see for example Gelb, [14], or Schweppe, [27].

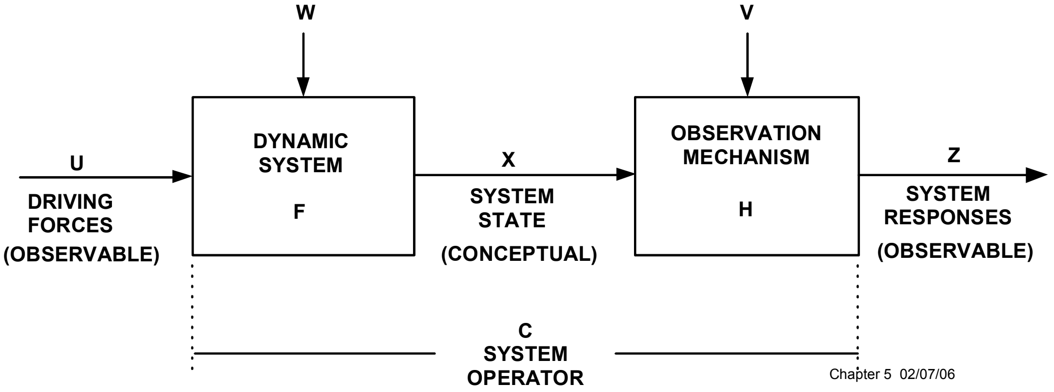


Figure 5-1. The State Space model.

Once a collection of attributes representing the state of a system has been selected, the modeler can describe the system in terms of its causes and effects. To do this, the modeler must describe conceptually the relationships which he perceives to exist in the system. They must then be incorporated into the framework of a state space model. These relationships, which must be described by the modeler, typically represent significant additional information about the structure of a system which can lead to a corresponding improvement in model accuracy. The convenience of using the state space framework comes about by a separation of observation from the conceptual dynamics of a system. It is this separation of concept from observation which affords the modeler a powerful tool for mathematically formulating his conceptual knowledge about the structure of a system.

Additional accuracy can be obtained by modeling the effects of driving forces which are assumed to be observable, causal, and to "lead" the response. This is the normal use of the concept of driving forces, otherwise no additional information could be obtained from them for improving prediction accuracy. In particular, we are concerned with the description of *nonhomogeneous* models which relate system responses to nonstationary driving forces which need not be characterized statistically. We must understand that these relations can be highly nonlinear, and difficult to model. However, incorporation of these effects can also lead to significant improvements in model accuracy.

These two aspects of a model,

- expression of the structural properties of a system
- modeling the effects of nonhomogeneous driving forces

represent *additional information* which is generally *not contained in the response data*, particularly when the system is either nonlinear or nonstationary.

We wish also to allow for development of complex models without arbitrary confinement due to rules of parsimony advocated by a number of authors, e.g., Tukey, [29]. When using methods where the structure of a system is ignored, and a naive approach is pursued for model identification, then unknown coefficients are used merely to *fit* the response data. In this case, the modeler may be concerned about parsimony. This is because additional coefficients add no additional information to condition the probability statement so as to be more accurate. However, if a model is enhanced by the benefit of additional knowledge of the structure of the system, then these model additions will serve to condition the probability statement so as to be more accurate (by definition) and the constraints of parsimony do not apply.

A user of predictions will judge one model to be superior to another if it provides him with consistently more accurate predictions of the future. On this basis, there are many examples in engineering (e.g., modeling of integrated circuit chips) where models have been carefully constructed based on knowledge of a physical structure. The complexity of these models would appear to violate rules of parsimony as advocated by many statistical forecasters. Nevertheless, these models have provided excellent consistency with test results, long after model development.

A Generalized State Space Framework

In order to deal with more complex models, particularly those involving human decision processes, one must move towards a more generalized framework. To do this, we will use Generalized State Space where the state vector is not limited to numeric values. This provides for states that take on words as well as numeric values, e.g., GREEN, YELLOW, RED, etc. In addition, the transformations need not be limited to mathematical operators, but can contain conditional statements, e.g., IF ... THEN ... ELSE ... , as well as statements that move and change data words. Although described originally by Cave in the earliest versions (1982-1983) of the VisiSoft Users Manual, [15], a similar but more restricted concept has subsequently been described by others, see [24] and [25].

The Generalized State Space framework allows the modeler to more accurately represent a physical system, particularly one that is nonlinear and contains human or computer decision algorithms. This framework eliminates cumbersome abstractions that make it difficult for subject area experts to relate to the models. It provides for more direct validation of a model. Examples of this approach are provided in Simulation of Complex Systems, [10], along with comparisons to standard mathematical approaches. We will stay with a mathematical framework in this book to expose the theory. However, there is a one-to-one correspondence between the mathematical principles described here and those of the generalized framework.

General Model Formulation

A general model formulation of a system would include several properties denoted by

$$X(T) = \begin{bmatrix} X_1(T) \\ X_2(T) \\ \cdot \\ \cdot \\ \cdot \\ X_n(T) \end{bmatrix}$$

where X(T) is a vector valued function of time in some n-dimensional space. Some of these properties may be observable, but *none* need be. The important criteria is to select a set of properties which simplifies the modeler's conceptual view of the "mechanics" of the system, e.g., how a market physically operates or moves from observed time point to observed time point. In many cases, these conceptual properties cannot be measured, at least for economic reasons. For example, we can envision a market as being composed of a mass of people who enter the "market place" upon making a decision to buy or sell. Upon striking a deal which satisfies their desire to buy or sell, they leave the market place. We can write the "equations of motion" which describe their rate of entry, their number at any time, and rate of departure based on external influences. Whether we can observe these properties directly is unimportant, as long as we can relate them to things we can observe, such as high price, low price, and volume of trading for the time period of interest. The objective is to predict X(T+1), the *state* of the system at the next time step.

To this end, a dynamic model of the form

(5-1) $$X(T+1) = F[X(T+1), X(T), U(T), T]$$

is proposed. Thus, the next state of the system can depend upon itself X(T+1) (i.e., it is nonlinear), the current state X(T), the stimulus or driving force U(T), and upon time, T, directly. In particular, the driving force vector

$$U(T) = \begin{bmatrix} U_1(T) \\ U_2(T) \\ \cdot \\ \cdot \\ \cdot \\ U_m(T) \end{bmatrix}$$

must be directly observable, and must lead and affect the response. Typically the driving force is unpredictable. Otherwise, it could be incorporated as a response to another driving force with a further lead, or as a known function of time.

We will denote the observable system response by the vector

$$Z(T) = \begin{bmatrix} Z_1(T) \\ Z_2(T) \\ \cdot \\ \cdot \\ \cdot \\ Z_n(T) \end{bmatrix}$$

If this observation vector, Z, can be derived from the state vector, X, at any time T via a relationship of the form

(5-2) $$Z(T) = H[X(T), T],$$

then, given the prediction of X(T+1) from our dynamic model (5-1), we can calculate Z(T+1) from (5-2). In addition to being a general formulation for dynamical systems, experience has shown that this separation of observation from concept allows the modeler to more easily translate his knowledge of system structure into mathematical form.

In future sections we will have cause to view equations (5-1) and (5-2) as a single transformation, C, denoting the relationship between the driving force vector at time T, and the observation vector at time T+1.

(5-3) $$Z(T+1) = C[X(T), U(T)]$$

We will refer to C as the *system operator*, reference Figure 5-1.

Nonlinear Considerations

To clarify the concept of linearity versus nonlinearity in a model, consider the operator

(5-4) $$G[X(T+1), X(T), U(T), T] = 0$$

We say the model is *linear* when G above is linear in its first argument, X(T+1). In such cases, X(T+1) is relatively simple to isolate algebraically, leading to

(5-5) $$X(T+1) = L[X(T), U(T), T]$$

Any other situation with respect to the first argument of G in (5-4) above is said to be a nonlinear model. Most often, when modeling nonlinear systems, no simple isolation of the X(T+1) term is possible.

As an example of a nonlinear relation, consider the scalar state equation

$$(5\text{-}6) \qquad x(T+1) \;=\; a \cdot x(T) \;+\; b \cdot u(T)$$

where b depends on x as shown in Figure 5-2. This relation is typical of market saturation effects. For example, let u represent advertising budget and x product demand. For small values of x, demand increases linearly with u. As x increases sufficiently, b decreases, and increased advertising will cause smaller increases in demand. For large x, advertising has little effect. We note that the relationship between b and x is independent of time, a characteristic of nonlinear relationships. We note also that at time T+1, x(T) has taken on a known numerical value, whereas b(x(T+1)) and x(T+1) must be determined simultaneously. Methods for finding simultaneous solutions to the equations of nonlinear dynamic systems are treated by Gear [13] and Nordsieck [22].

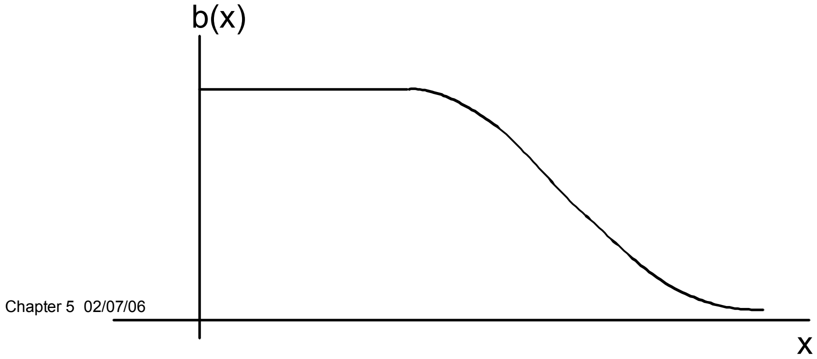


Figure 5-2. Example of a nonlinear relationship.

Nonhomogeneous Considerations

We wish to characterize "forced response" or nonhomogeneous type systems. We start with equation (5-1), and consider the case when the driving forces are zero. Under this condition, we assume the state of the system to be stationary, i.e., in periodic or constant equilibrium.[†] (Markets of interest can be modeled such that price becomes stationary when driving forces are removed.) In the linear case, equation (5-5), we can define the linear operator £ such that, when the driving forces are zero,

$$(5\text{-}7) \qquad \pounds[X] \;=\; X(T+1) - L[X(T)] \;=\; 0.$$

Equations of this form are termed homogeneous (Tikhonov, [28]).

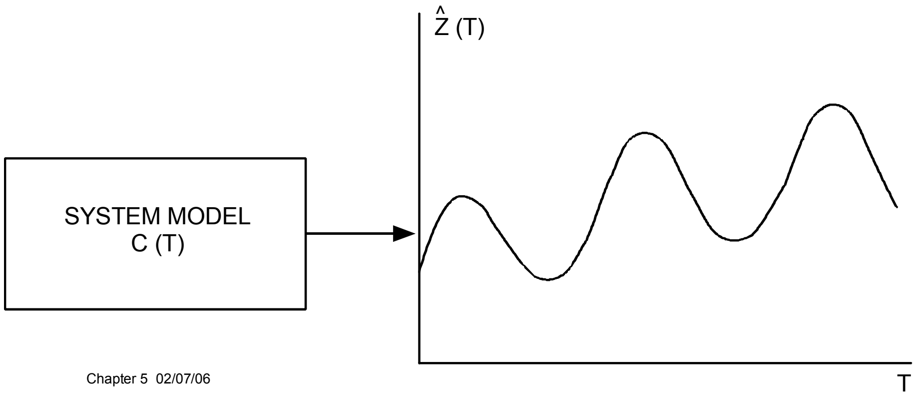
When driving force U(T) is introduced, equation (5-7) typically takes the form

$$(5\text{-}8) \qquad \pounds[X] \;=\; U.$$

[†] We are describing the system deterministically here, not statistically, reference Tikhonov, [28].

We note that, once we have defined the equations, the forced response of a stable[‡] linear system is obtainable by standard techniques, e.g., convolution or Green's function (see, for example Friedman, [12]), whereas numerical methods can be used to obtain responses from nonlinear systems (as in Gear, [13], or Nordsieck, [22]).

Where the forced response is itself stationary or periodic, the model can be recast in a homogeneous form. The new model will be seen as having no driving force; i.e., the periodic component will appear as an internal part of the system rather than a stimulus. This principle can be extended to the more general case where either the driving force or the forced response is a known function of time over the time frame of interest (future included). Refer to Figure 5-3. Using a Fourier series expansion over a bounded time period yields a linear sum of periodic functions, which admit to a homogeneous model.



Chapter 5 02/07/06

Figure 5-3. Homogeneous model of the system response.

In general, we are interested in predicting responses to systems which are ultimately influenced by driving forces which are *not* known functions of time, periodic or otherwise. To this end, we define our model to be nonhomogeneous if it fits the form of equation (5-4) above, where U(T) is a nonzero observable driving force which may be independent of X, but has a causal effect on X. We note that this definition is independent of the ability to isolate either X(T+l) or U in (5-4). In general, U(T) is unknown until observed, and need not be characterized statistically.

At this point we introduce the concept of a nonhomogeneous system, i.e., one which can only be represented accurately by a nonhomogeneous model. This concept is helpful in determining the best form for model equations. Basically, if a system is driven by external forces which appear to be random, and if the system response can be related to these driving forces through delays and time constants, then one can predict the response more accurately by using a nonhomogeneous model which accounts for these driving forces. This important concept is further described in Chapter 8, STOCHASTIC MODELS under Nonstationary Considerations.

[‡] By "stable" we imply that bounded inputs yield bounded outputs, refer to Chapter 6 under The Concept of Boundedness.

Modeling Inertial Subsystems

To demonstrate the significance of incorporating "leading factor" driving forces into a model, we offer an example which is representative of many actual cases. Let U(T) be the driving force (Figure 5-4a), and let Z(T) be the output response of the system, (Figure 5-4b). Both are observable at discrete time points T. Figure 5-4b represents a typical superposition of two exponential response functions as used in engineering. TD1 and TD2 are delay times measured from the input impulse. TD1 is the time before the first exponential starts to rise (positive). TD2 is the time before the second exponential starts to fall (negative). TAU1 and TAU2 are the rise and fall time constants for these two exponentials. These same delay times and exponentials are applied to all succeeding inputs.

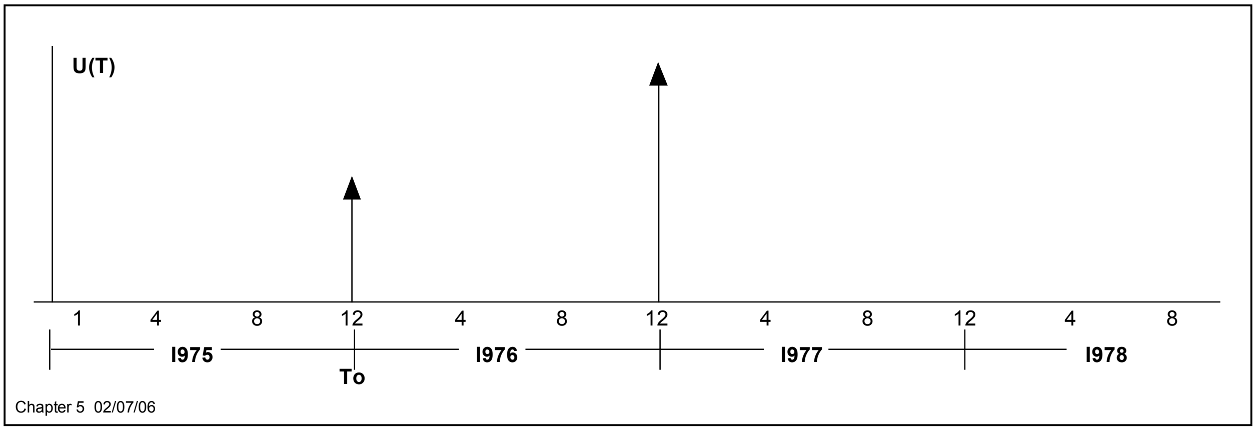


Figure 5-4a. Driving force input.

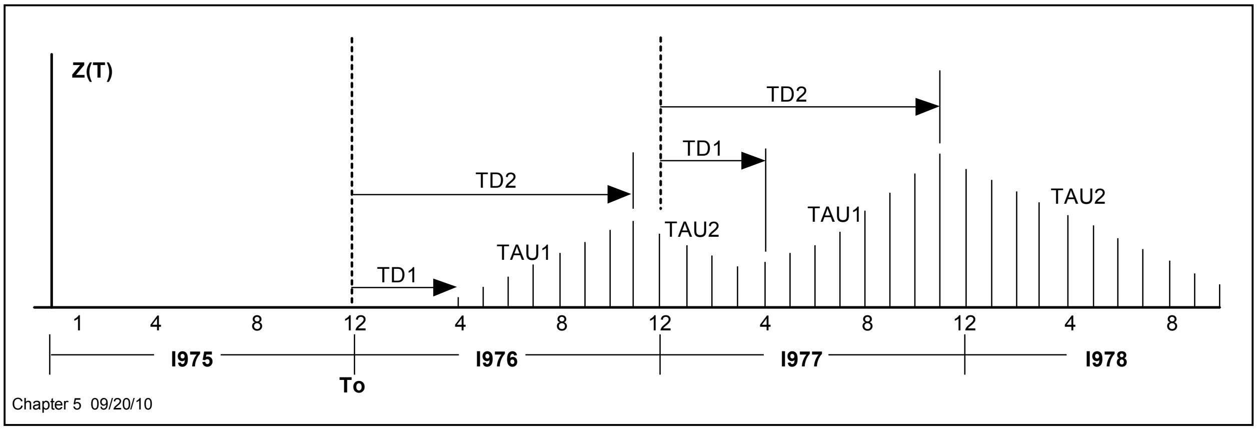


Figure 5-4b. System response.

The impulse at To causes inertial properties within the system to react over time. These reactions are represented by exponential rise and fall times with time constants TAU1 and TAU2. They are then superimposed using linear superposition. Without being able to model these inertial effects and their sufficiently long time constants, one cannot hope to predict the future beyond a single time step.

As the figures indicate, the response is 0 prior to (and at) T_0, when an impulse occurs in U, and remains zero for three more time steps. Simply using the observed output of the system at T_0, T_1, T_2, and T_3 will be of no value in determining the output at T_4. The information, that an impulse has occurred, cannot be derived from the response data alone.

Assuming our model in Figure 5-4 represents the system perfectly, we could predict *with no error* up to four steps into the future. Furthermore, when the input appears to be purely random, so does the response; but this does not preclude us from making perfect predictions of the response up to four steps into the future.

Modeling Distributed Responses To Events

When modeling populations of elements of nature, one must face the fact that all elements or individuals do not produce the same response to an event, and if they do, it is not produced at the same time. Instead, responses are typically characterized by distributions in time and state space. For example, if the electrical power is lost in a given populated area, large numbers of people will start trying to communicate with police, neighbors, radio stations, relatives, etc. Their response to this well defined event produces a distribution of follow-on events that can be quite varied in their actions as well as the time of occurrance.

It is this very behavior that produces inherent predictability in a system. However, we must be able to model these types of responses accurately as they directly affect the accurate prediction of behavior of such a population. We can create such models quite easily using VisiSoft.

We will start with an example of distributed responses to a sequence of events to show how the resulting cumulative response can be modeled quite acurately. This is useful in predicting responses to events that occurred many timesteps in the past. To demonstrate this, we will build a model of housing unit completions as a function of the event of taking out a building permit. We can make a number of simplifying assumptions to start, and then build a more accurate model.

Modeling Housing Completions As A Funciton Of Building Permits - An Example

We would like to predict the number of housing completions in a given geographical area months in advance. These predictions could then be used to predict sales of appliances, phone systems, furniture, carpeting, etc., purchased after a house is complete. Actual completions can be measured by certificates of occupancy issued in a given month. The most significant factor in predicting housing completions is building permits. These are normally taken out many months prior to completion. We start with an analysis of one month's worth of building permits to determine the resulting distribution of completions for those permits. Let's assume that our investigation yielded an average distribution that took on a shape as shown in Figure 5.5. Then we could model the resulting distribution as shown where the number of housing units in the distribution equaled the building permits taken out (or some percentage if all did not result in completions). Figure 5.6 shows the superposition of housing completions due to building permits taken out in months 3 and 10.

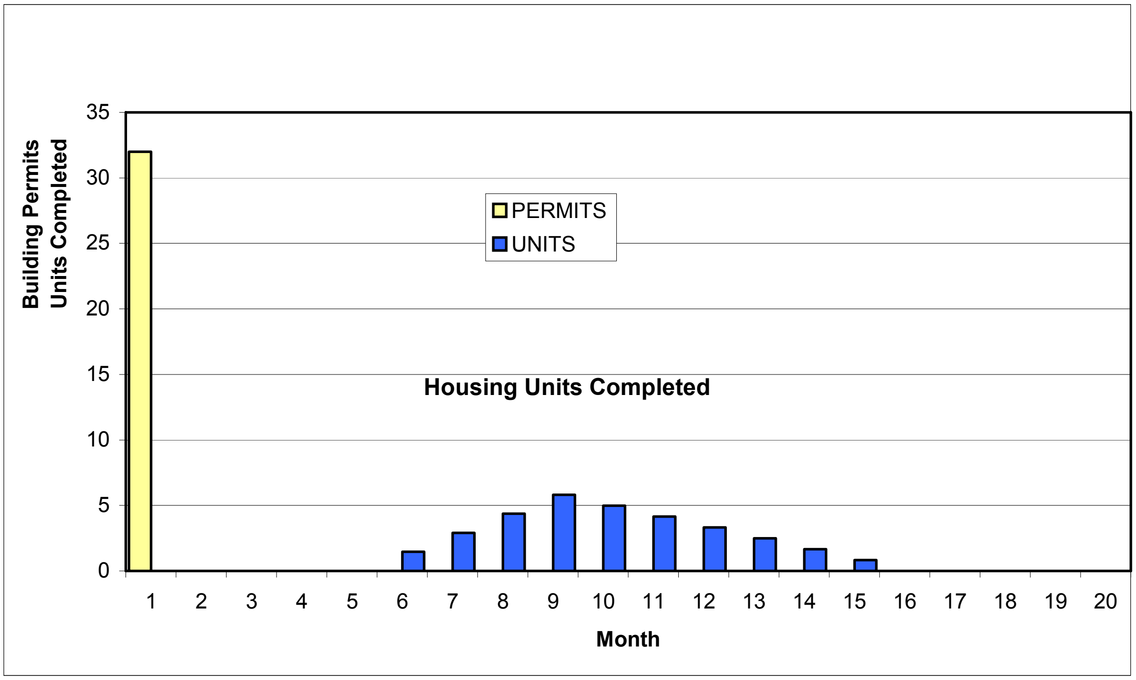


Figure 5.5. Housing units completetled in months 6 through 15 as a result of building permits in month 1.

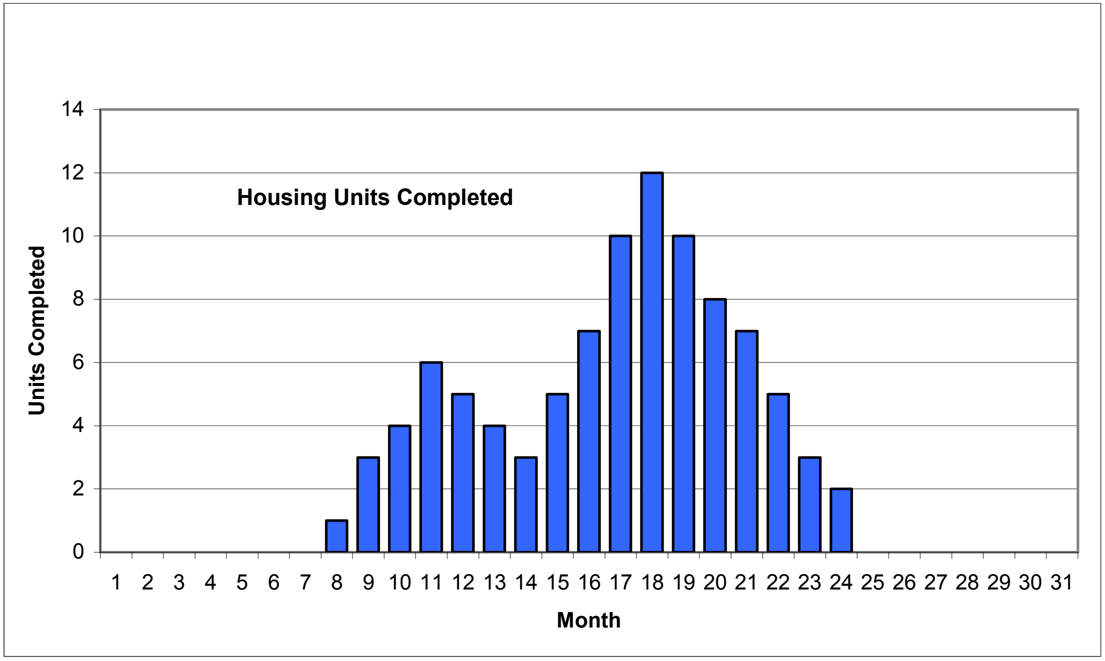


Figure 5.6. Housing units completed in months 8 through 24 as a result of building Permits in taken out in months 2 and 10.

Figure 5.7 shows the predicted housing completions as a function of building permits starting in month 1. Note the time it takes for a build up of the proper memory of housing completions due to building permits in the past. If this were an actual prediction, then the result would not be valid until the results of any permits in months prior to month 1 were washed out. With the distribution shown, this would take a total of 15 months. Based on this model, the data beyond that point would not be affected by anything before month 1.

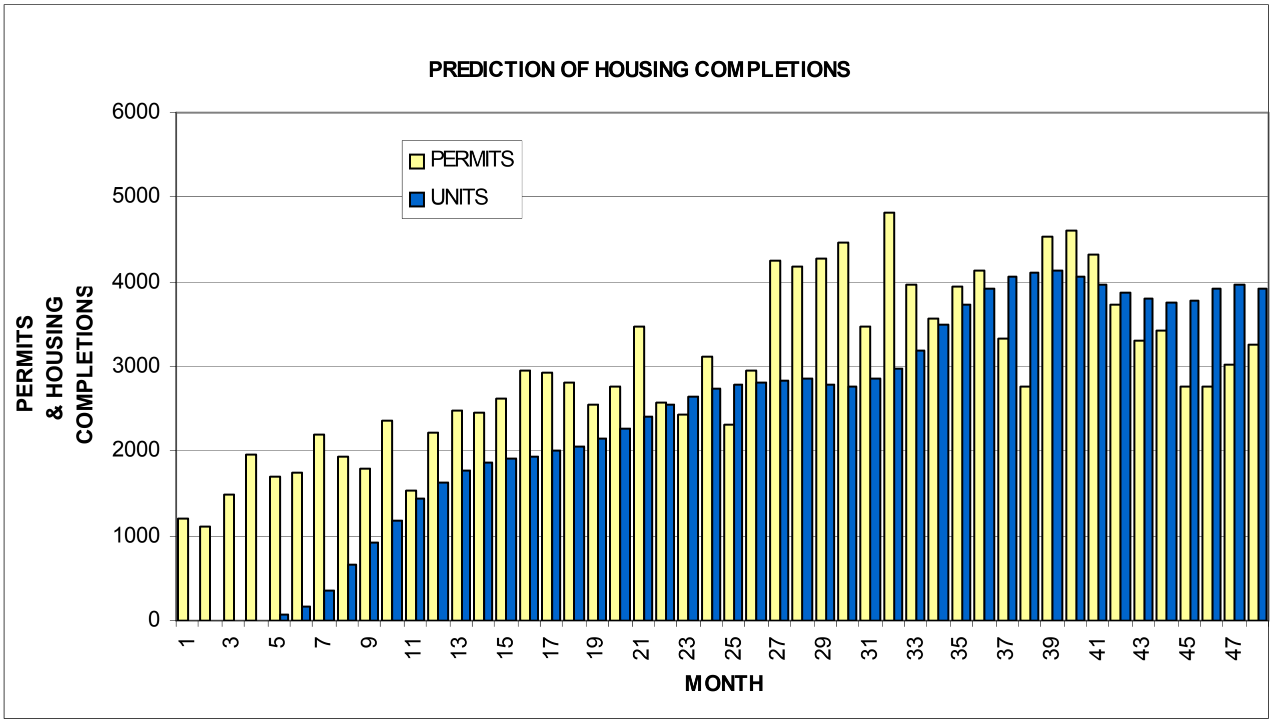


Figure 5.7. Housing units completed a result of building permits starting in month 1.

If one were to plot the correlation between housing completions and building permits as a function of time for this model, it would peak around 10 months. This is because the mean of the distribution falls at about this point. Note also that, if the distribution function accurately represented the actual housing completions, there would be no error in predictions up to five months out because the unknown quantities of building permits taken out after the current time would not affect housing completions during this period. Obviously, there are other factors that affect housing completions, and we would look to incorporate those into our model.

The VisiSoft model described above contains simple submodels that produce the prediction of housing unit completions. This model is based on a single external factor, namely building permits, and the distributed response function described above. It is also based upon time correlation factors that are developed in a weighted manner over prior years and applied to the current year to improve the prediction accuracy. Time correlation is described further below.

We note that this same approach has been used to accurately predict U.S. Money Supply over many years, see for example Figure 3-1.

Linear Versus Nonlinear Systems

It is easy to confuse linearity and nonlinearity. For example, one may look at the response of a system with nonhomogeneous inputs (external driving forces) and conclude that the time-domain waveform appears “chaotic”. In some fields, this is translated as the response of a nonlinear system. To impose a more careful consideration, the word *chaotic* is used to describe the *apparent behavior* of a function of time. It is not a mathematical term. Apparent chaotic behavior is easily generated by linear systems.

Consider the linear addition of nonhomogeneous inputs to a system. As long as we are looking at the waveform over finite time periods (difficult to avoid), the inputs may be sine waves that are linearly superimposed, where at least one of the sine waves has a period longer than that of the observation period. The result may appear “chaotic” to some observers. Examples of this type are common when studying Fourier series or Fourier transforms. What is important is that the inputs are linear stationary functions that may be operated upon using a linear transformation as described in the next chapter.

We also note that it may be difficult to assess nonstationarity. This is a common cause of misunderstanding when building prediction models. Such misunderstandings are typically uncovered only after properly characterizing and validating statements about prediction accuracy. One finds that nonstationarity is often the result of nonlinearity.

When characterized properly, nonlinear system properties are generally invariant with time (in special cases they may also vary with time but these cases do not change our analysis). Thus they can be invoked at any time and their properties will hold. This is further discussed below.

6. CLASSIFYING SYSTEMS AND THEIR MODELS TO PREDICT THEIR RESPONSE

In the prior section, a general model formulation was provided on a mathematically deterministic basis, without concern for additional real world considerations. In fact, when modeling real world systems to predict their future, a number of constraints must be addressed which serve to both help and hinder the modeling process. These constraints are addressed below in terms of "boundedness," "randomness," and "stationarity." The purpose of this section is to further understand the inherent properties of systems and the functions used to represent their dynamical behavior. This is particularly important when distinguishing between prediction and curve-fitting (the underlying approach in many books on forecasting). This section is also aimed at helping the transition from deterministic models to probabilistic models so that we can provide predictions whose probabilities are conditioned on all available information about the system.

Proper characterization of the properties of a system is an essential step in the modeling process. Although it is not necessary that the properties of a model be identical to those of the system to obtain good results, it is important that the properties of a model be distinguished from those of the actual system to gain a proper perspective. This may sound obvious, but lack of this distinction has been the source for much confusion in building models for prediction. If certain properties of a system must be modeled for valid results, then those properties must be reflected in the model.

In the prior section, we addressed systems with both nonlinear and nonhomogeneous properties. The intent of the examples was to expose both the likelihood of occurrence of these properties within systems of interest, as well as the effects they can have on prediction accuracy if not modeled properly. The effects of nonlinearity and nonhomogeneity are more apparent than those to be discussed. This is because they can be described in terms of deterministic models of physical systems for which measurement is relatively easy.

The systems with which we are concerned have additional properties which are much more subtle to recognize and model. In addition, much care must be taken when characterizing these properties to ensure that the definitions being used properly apply to the real world system being modeled. This word of caution may also appear obvious, but there are many definitions of similar concepts which do not apply when building models for prediction.

The following discussion highlights properties of *systems* which can afford insights into corresponding properties of their prediction models. If properly reflected in a model, they can significantly enhance prediction accuracy.

The Concept of Boundedness

The systems of concern here are always bounded in time and measures of their state. These properties of boundedness are described below.

- *Discrete sampled data* - Observations are available only at discrete points in time, i.e., we are concerned with *sampled data systems*. For example, markets may be monitored on a yearly, monthly, daily, or transaction by transaction basis. All are sampled at discrete points in time.

- *Finite time period* - The time period of interest is finite. All market history and future horizons of interest are finite.

- *Finite number of sample points* - The number of observations is finite, i.e., bounded. The number of data points may be extremely large, but is not infinite.

- *Stable* - Bounded inputs yield bounded outputs. Any influences on the system which are bounded can only cause responses which are bounded. Nonlinear effects, such as saturation, prevent market responses from going to infinity.

- *Bounded Measures* - The system is always bounded. Any measures or variables describing the state of the system or its responses are always bounded. Real world systems are always finite.

It is important to realize that we have enumerated properties of a system, its observations, and our time frame of interest, these being distinct entities. We note that all of these measures are bounded, by definition. In particular, we are only interested in systems described by a finite number of sample points which are bounded. This implies that the *spectral properties* of such systems, i.e., their properties in the frequency domain, must also be bounded.

The Concept of Randomness

We will limit our discussions of randomness to that of bounded data sets which could represent the system response, or external factors that affect the system response. By "bounded" we imply the properties of the previous section, i.e.:

- bounded time frame of interest
- bounded number of discrete sample points
- bounded values of the data

The usual definition of randomness implies no correlation with time, i.e., no autocorrelation. The usual test states that Z(T) is random when the expected value of the inner product of the deviates is sufficiently close to zero for all $\tau > 0$. We will use the notation:

(6-1) $$E\{Z(T), Z(T+\tau)\} < \delta \approx 0 \quad \text{for all } \tau > 0.$$

where $$E\{Z(T), Z(T+\tau)\} = \frac{1}{T_T} \cdot \sum_{T=1}^{T_T} [DZ(T) \cdot DZ(T+\tau)] \ ,$$

$$DZ(T) = [Z(T) - \mu_Z],$$

and μ_Z is the expected value of Z over the period of interest:

$$\mu_Z = E\{ Z(T) \} = \frac{1}{T_T} \cdot \sum_{T=1}^{T_T} Z(T) \quad .$$

Since we are dealing with bounded data sets, we will interpret randomness as follows. Z(T) is *not* random if a transformation C can be found such that for some $\tau > 0$,

(6-2) $$E \{C[Z(T)], \ Z(T+\tau)\} \geq \varepsilon_\tau$$

where ε_τ is a sufficiently large value based on judgement. When this is true, Z(T) is predictable to some extent up to τ steps into the future. Otherwise, Z(T) is *apparently random*. The word "apparently" is used to imply that we can never be sure that a data set is random, i.e., how do we know that, if a C cannot be found, one does not exist. This is best explained by way of example. Modeler A uses a standard autocorrelation test and comes up with a value ε_A which is less than ε_τ. Modeler B uses a special "window" to search for autocorrelation and obtains $\varepsilon_B > \varepsilon_A$, but still less than ε_τ. Modeler C uses a special function C which allows for variations in the "period of periodicity" of the data, and comes up with $\varepsilon_C >> \varepsilon_\tau$. (As an example of changing periodicity, the product of two periodic functions with different periods will appear aperiodic over a bounded time frame). We would expect model C to provide reasonably accurate real time predictions relative to models A and B.

The above examples indicate that what one person perceives to be random in time, another may determine as having a high degree of order with time. In other words, there appears to be no single measure of randomness for a bounded data set.

Probably the best example of this phenomenon is encountered in cryptography. Here one creates ciphers using "pseudo" random codes which, when tested by people from whom information is to be hidden, *appears* to be random. Those having the "key" to decipher the code (i.e., they know the transformation C), can retrieve intelligible data which can contain information relating to future values of the data set, including new keys.

We must explore the concept of randomness as it pertains to information about the future. In this context, a data set appears random if past values contain no apparent information about future values. If, however, we can find a transformation, C, that clearly improves prediction accuracy, then the data is not random. As in the cryptography example, the amount of information in a data set may appear negligible. It's the ability to find the transformation, C, that will determine how accurately the future can be predicted for a given τ_p.

We may also want to look at statistical properties of the data set. For example, consider the data set in Figure 6-1a. It appears that Z(T) is increasing in value with T. In other words, if we look at a sequence of subsets of the data (ensembles), the mean value is increasing as a linear function of time. Knowing this, we could determine C by posing a straight line and picking values for slope and cutoff to minimize modeling error. In this case, C[T] becomes a *known function of time*. Future values of Z, e.g., Z(T+τ), can be determined "more accurately" if we look at values of C[T+τ]. We will still encounter error, but this error is significantly less. Thus, the function is *not* "purely" random, i.e., it contains elements which are predictable. In addition, the resulting error, e, shown in Figure 6-1b, appears to be bounded when normalized to the value of the line.

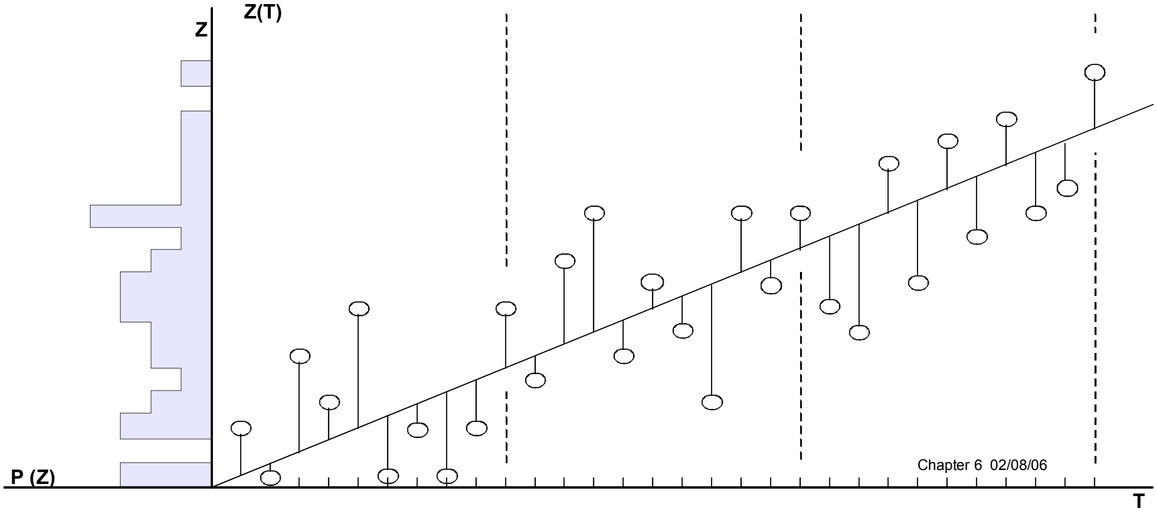


Figure 6-1a. Statistics of a response function.

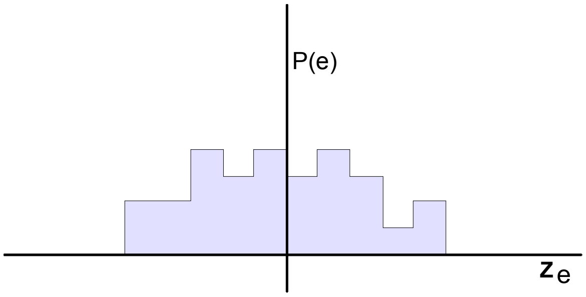


Figure 6-1b. Statistics of the error function.

Measures of Prediction Error

Models of a system can be compared in terms of their accuracy based upon measures of error. This can be accomplished using a measure of the sequence of differences between predictions, $\hat{Z}$, and observed values of the response, Z. A convenient measure uses the sequence of normalized residuals up to T+ τ,

$$\mathrm{RN}[\hat{Z}(T+\tau)] = \frac{\hat{Z}(T+\tau) - Z(T+\tau)}{Z(T+\tau)} ,$$

over the period from the looking back horizon, T_B. This measure is denoted by ε_Z :

$$\varepsilon_Z = E\{\mathrm{RN}[\hat{Z}(T+\tau)]\} = \frac{1}{T_T - T_B} \bullet \sum_{T = T_B}^{T} \left| \frac{\hat{Z}(T + \tau) - Z(T + \tau)}{Z(T + \tau)} \right|$$

To compare model accuracies, one can compute the error statistics for the above measures using data that has not been used to build the models. If the data has been used to build the models, then one is comparing how well the model fits the history data, not how well it predicts the future, see [2].

We will now investigate the residual error sequence (*residuals*) to determine if any information can be found which can help to improve prediction accuracy. We might note that this error may appear statistically stationary with mean μ_e and variance σ_e^2. In fact, the error might only be characterized as bounded, i.e., we cannot determine a particular probability density function to characterize it. In either case, there may still exist another transformation C_e that we can apply to the residuals which "filters" or derives information to help improve our model, and thus prediction accuracy.

Now let's assume that the error sequence itself is purely random, i.e., this sequence of error data contains no additional information to improve prediction accuracy. Can anything else be done? If the *system* is nonhomogeneous, the answer is yes.

Consider the model in Figure 5-1. There may exist observable driving forces:

$$U_i(T), I = 1, 2, ..., m$$

that correlate to future values of Z, i.e., Z(T+τ). Refer to Figure 6-2. Both the driving forces and the response appear to be random. However, knowing the current value of the driving forces, one may be able to predict the response τ steps in the future. Again, the same problem must be solved. One must find the transformation which extracts the information to improve prediction accuracy. In this case, find U(T) and C, such that

(6-3) $$E\,[C[U(T), Z(T)],\ Z(T+\tau)] \geq \varepsilon_\tau .$$

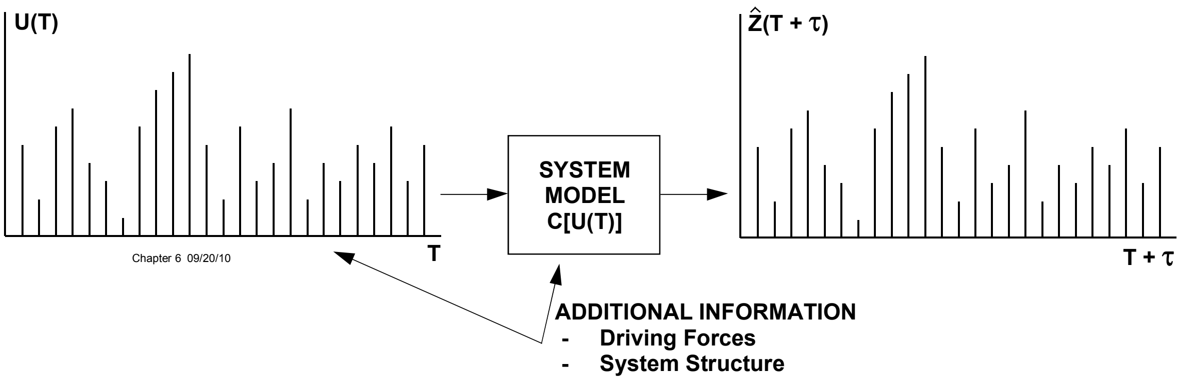


Figure 6-2. Predicting a "random process."

The Concept of Stationarity

A property which appears to cause confusion in the field of forecasting is that of stationarity. The concept of stationarity is well established in physics and engineering, see for example Tikhonov & Samarski [28], page 105. Stationarity applies to mathematical functions, be they deterministic or statistical. A functional definition will be used here which is consistent with the literature, and which hopefully serves to clarify properties of prediction models which are easily misunderstood. Since we are concerned only with functions Z(T) bounded in time and measures of state, we limit the definition accordingly.

> *A Stationary Function* is one which is fixed over a given time period τ, i.e., one can always find a τ such that $Z(T + \tau) = Z(T)$ for any T.

Sine waves are examples of stationary functions that are relatively obvious. Less obvious are functions composed of the product of sine functions e.g., $\sin(\omega t) \cdot \sin(3\omega t) \cdot \sin(6\omega t)$. In this case the period τ is 6 times the basic time period.

In addition, it will be convenient to use quasi stationary functions. These are defined as follows.

> A *Quasi Stationary Function* is one which can be transformed into a stationary function using a homogeneous transformation, i.e., one can find a homogeneous transformation C and time period τ such that $C[Z(T + \tau)] = C[Z(T)]$ for any T.

Quasi stationary functions have the property that, although the original functions depend explicitly on time, a homogeneous transformation may be found which converts them into a stationary function of time. A simple example is the quasi stationary function $Z(T) = T$. The homogeneous transformation is simply the multiplier 1/T, or $1/T \cdot Z(T) = 1$, which is certainly stationary. As a more pertinent example, periodic functions can be transformed from the time domain to the frequency domain using a Fourier series. In fact, any finite time function can be transformed into a stationary function using an orthogonal transformation. More practically, one may represent any finite function to a prescribed measure of accuracy using a sufficient number of terms from an infinite series, e.g., sines, cosines, or exponentials. Since we are concerned only with bounded functions, we note that they are all quasi stationary. Note also that these definitions apply to statistical functions (probability densities) as well as deterministic functions.

We note that any bounded function can be approximated to any degree of accuracy desired by using an orthogonal transformation. This is known as "curve fitting." It is important to understand the underlying assumption, i.e., that one can fit a "known function of time" to the original data set. Applying this technique when building a prediction model implies that the future values of the data set will be an extrapolation of the known function of time selected. This assumption can only be validated by showing correlation between prediction error and modeling error for the function being used. Stated more simply, one must show that future values of the response are expected to take on the same functional form as past values.

The Concept of Orthogonality

At this point we would like to explore the development of and limitations of complex models, since the systems we are trying to model are typically quite complex. Moreover, their responses are typically nonstationary, statistically as well as deterministically. Specifically, we want to avoid arbitrary confinement to simplistic models based on rules of parsimony. This consideration often arises when developing statistical models, see for example Tukey, [29]. Although our principal interest is the development of deterministic models, there is a parallel concern which should be exposed. This concern is based on the concept of linear independence or, more strongly, orthogonality.

If, for example, we are trying to relate the response of a system to three candidate driving forces, we must first be sure that the three are linearly independent. This implies that any one cannot be expressed as a linear combination of the other two. There are simple tests for linear independence; see, for example, Friedman, [12]. If the three driving forces are linearly independent, they must each contain information which is *orthogonal* to that in the others. Next, if the response can be shown to depend on each of these orthogonal streams of information, then each driving force adds new knowledge to help predict the response, and thus improve accuracy. This "orthogonality principle," as it is described by Papoulis, [23], is a cornerstone in the foundation of information theory. Its application to the Kalman filter is described in Chapter 8 under Closed Loop Considerations. In that case, one designs the filter to ensure the predicted response is orthogonal to the noise.

With these concepts in mind, let's now consider a modeler who is prone to use a curve-fitting approach. Having sufficient data that appears functionally nonstationary, he believes he can use a large number of coefficients that are linearly independent to fit it. The problem arises when trying to correlate prediction error to modeling error. If more coefficients are used to reduce modeling error, then the ratio of prediction error to modeling error can get quite large if the functional form of the history data does not closely correlate to future values. The underlying difficulty has nothing to do with the number of coefficients used to fit the history data. The root of the problem lies with the assumption that the functional form of the history data will continue into the future.

To summarize, we may have an additional driving force that is a candidate input to our model to improve prediction accuracy. To accomplish this, we must:

- Ensure that the candidate driving force has sufficient information content that is orthogonal to that in the existing driving forces.
- Find the proper internal model that can "extract" this information from the driving force data set. (This can be a difficult modeling problem.)

In practice, we must be concerned about error in the data. This topic is reviewed in Chapter 8 under Closed Loop Considerations.

Statistical Stationarity

We will now investigate concepts of stationarity as they apply to statistical distributions used to characterize systems, their responses, and their models. Consider the sequence of T data points, where T is a finite number,

$$1,\ 2,\ 4,\ 8,\ 16,\ 32,\ ...,\ x_T$$

x_T is perfectly predictable just knowing T, i.e.,

$$x_T = 2^{(T-1)}$$

Because it is a known function of time, i.e., the value of x_T is known for all time T, statistical methods are not needed to fit it. However, if one uses standard statistical tests for stationarity of the data, this process is nonstationary. Simply stated, the mean and variance vary with T. We also note that the response can be represented with a homogeneous model since it is a known function of time.

We summarize the critical facts. The system portrayed in the above example operates as a known function of time. Therefore it can be modeled as a homogeneous model, i.e., no external driving forces affect the system response. The system response data sequence is statistically nonstationary by standard tests. The system is perfectly predictable. Hopefully, these statements serve to illustrate the importance of differentiating between the properties of systems, their models, and their response data.

Returning to Figure 6-1, assume that we can look at enough ensembles of Z(T), each containing enough sample points, to determine that the "statistics", e.g., the mean and variance, are not stationary. However, we can "fit" a function to the data (in this case a straight line) which results in error statistics which are stationary over the history data, i.e., changes in the error statistics from ensemble to ensemble are insignificant. We must now correlate reduction in estimation error with reduction in prediction error, i.e., we must show autocorrelation on the ensemble by ensemble sequence. To do this, we must show that

$$E[C(Z(T)), Z(T+\tau)] \ > \ \varepsilon_\tau,$$

and therefore is significant.

Using enough coefficients of a higher order function to fit the sample points, we can reduce the model error as much as we want. However, if the system were a linear increasing function, and the error represented that due to measurements, we could not reduce prediction error below that of a straight line. As we reduce modeling error, the ratio of prediction error to modeling error becomes larger. All we are doing is fitting the history data more accurately (a table of history points would be ideal). However, we may be *increasing* prediction error!

From the above examples, we can draw the following important conclusions:

- Any bounded data set may be "modeled" (fitted) by a known function of time using a homogeneous model. However, the error encountered when fitting a known function of time to the system response (history) data may bear no correlation to prediction error, independent of the number of coefficients used in the fit.
- When using optimization to find the "best" coefficients, convergence to a stationary model requires that the error statistics be stationary.
- A system response which appears statistically random and nonstationary may be predictable if the system is nonhomogeneous, the driving forces are observable, and a corresponding nonhomogeneous model can be constructed.

7. TIME CORRELATION

Calendar Correlation Models

Three years of M1 data, Jan 1981 - Jan 1984, Not Seasonally Adjusted (NSA), are shown in Figure 7-1 where the data is moving up and down much more randomly than that which would result from the input driving forces. Clearly one must look for correlation with other sources. Although the data jumps around in what may at first appear to be a random fashion, it quickly becomes clear that the up and down movement is correlated with the calendar. Thus we will look for correlation with the calendar. Figure 7-2 shows the data behind the plot in Figure 7-1.

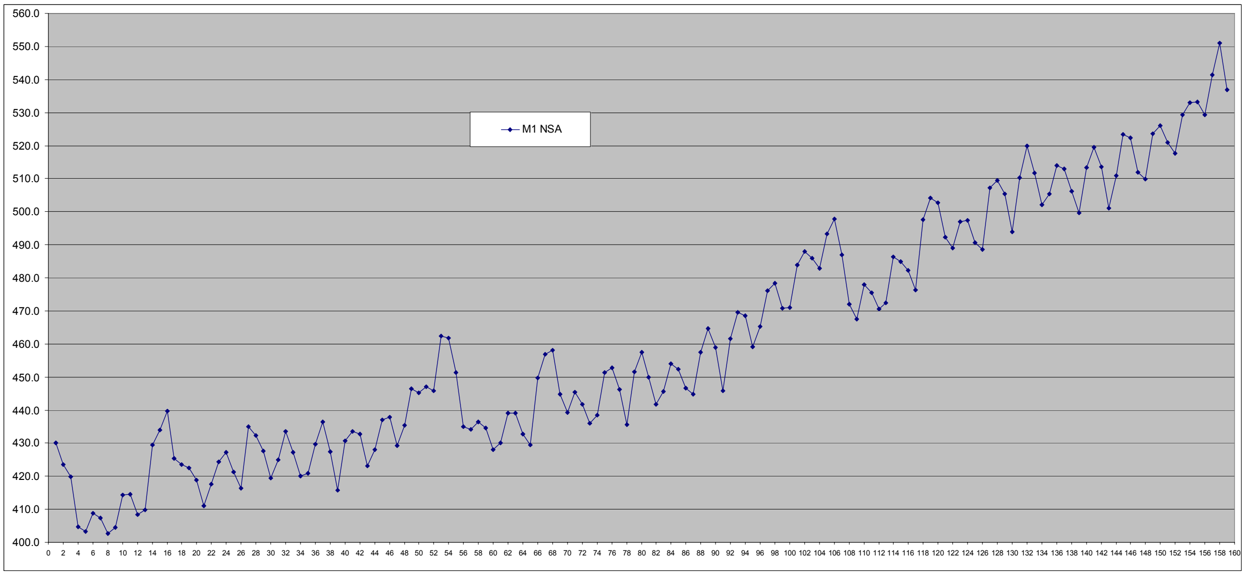


Figure 7-1. Actual curve appears almost random.

The actual data in Figure 7-2 has all of the “bottom” points highlighted in yellow. There are 12 of these in each year, each occurring at the transition between months. Four major peaks are highlighted in blue. These peaks occur in the 1st or 2nd week of the beginning of the year. Three major “double” peaks are highlighted in red. These occur the week before and the week after April 15th, tax time.

From the curves, it is clear that a special type of correlation analysis - based upon the calendar - is required to determine coefficients that could be used to improve the accuracy of predictions so that the width of the 80% envelope is as small as possible.

Although the data is produced once a week, it is correlated on a monthly and annual as well as weekly basis. Thus, it is necessary to do special correlation analyses. These can be done independently, where the time scale with the most correlation can be used to pull out that component and redo the correlation analysis on the residual data using the second component.

To do calendar correlation, one must be able to perform comparisons for a 5 or 7 day week; a 4 or 5 week month; and a 12 month year. One must also determine how to handle transitions when there are holidays, and especially when holidays fall on Friday or Monday, the transition at the end of a week, or transitions at the end of a month or year.

	DATE					M1 - NSA
1	Jan	5	1981			430.1
2	Jan	12	1981			423.6
3	Jan	19	1981	3	4	419.8
4	Jan	26	1981			404.7
5	Feb	2	1981			403.3
6	Feb	9	1981			408.7
7	Feb	16	1981	3	4	407.4
8	Feb	23	1981			402.6
9	Mar	2	1981			404.5
10	Mar	9	1981			414.3
11	Mar	16	1981	3	5	414.6
12	Mar	23	1981			408.3
13	Mar	30	1981			409.8
14	Apr	6	1981			429.4
15	Apr	13	1981	3	4	433.9
16	Apr	20	1981			439.7
17	Apr	27	1981			425.4
18	May	4	1981			423.5
19	May	11	1981	3	4	422.5
20	May	18	1981			418.9
21	May	25	1981			411.0
22	Jun	1	1981			417.6
23	Jun	8	1981	4	5	424.4
24	Jun	15	1981			427.2
25	Jun	22	1981			421.3
26	Jun	29	1981			416.4
27	Jul	6	1981			435.0
28	Jul	13	1981	3	4	432.3
29	Jul	20	1981			427.7
30	Jul	27	1981			419.5
31	Aug	3	1981			425.0
32	Aug	10	1981	4	5	433.5
33	Aug	17	1981			427.2
34	Aug	24	1981			420.0
35	Aug	31	1981			420.8
36	Sep	7	1981			429.6
37	Sep	14	1981	3	4	436.5
38	Sep	21	1981			427.5
39	Sep	28	1981			415.7
40	Oct	5	1981			430.6
41	Oct	12	1981	3	4	433.5
42	Oct	19	1981			432.8
43	Oct	26	1981			423.2
44	Nov	2	1981			428.0
45	Nov	9	1981	3	5	437.1
46	Nov	16	1981			437.8
47	Nov	23	1981			429.2
48	Nov	30	1981			435.4
49	Dec	7	1981	4	4	446.5
50	Dec	14	1981			445.3
51	Dec	21	1981			447.0
52	Dec	28	1981			445.9
53	Jan	4	1982			462.5
54	Jan	11	1982			461.7
55	Jan	18	1982	3	4	451.4
56	Jan	25	1982			435.0
57	Feb	1	1982			434.2
58	Feb	8	1982			436.5
59	Feb	15	1982	3	4	434.5
60	Feb	22	1982			428.0
61	Mar	1	1982			430.1
62	Mar	8	1982			439.0
63	Mar	15	1982	3	5	439.0
64	Mar	22	1982			432.8
65	Mar	29	1982			429.4
66	Apr	5	1982			449.7
67	Apr	12	1982	4	4	456.9
68	Apr	19	1982			458.2
69	Apr	26	1982			444.9
70	May	3	1982			439.3
71	May	10	1982			445.4
72	May	17	1982	3	5	441.8
73	May	24	1982			436.0
74	May	31	1982			438.4
75	Jun	7	1982			451.4
76	Jun	14	1982	3	4	452.8
77	Jun	21	1982			446.3
78	Jun	28	1982			435.5
79	Jul	5	1982			451.6
80	Jul	12	1982	3	4	457.4
81	Jul	19	1982			450.0
82	Jul	26	1982			441.8
83	Aug	2	1982			445.6
84	Aug	9	1982	4	5	454.0
85	Aug	16	1982			452.4
86	Aug	23	1982			446.7
87	Aug	30	1982			444.8
88	Sep	6	1982			457.5
89	Sep	13	1982	3	4	464.6
90	Sep	20	1982			459.0
91	Sep	27	1982			445.9
92	Oct	4	1982			461.5
93	Oct	11	1982	3	4	469.5
94	Oct	18	1982			468.5
95	Oct	25	1982			459.1
96	Nov	1	1982			465.3
97	Nov	8	1982	4	5	476.1
98	Nov	15	1982			478.3
99	Nov	22	1982			470.8
100	Nov	29	1982			471.0
101	Dec	6	1982			483.9
102	Dec	13	1982	3	4	488.0
103	Dec	20	1982			486.0
104	Dec	27	1982			482.9
105	Jan	3	1983			493.2
106	Jan	10	1983	4	5	497.7
107	Jan	17	1983			486.9
108	Jan	24	1983			472.0
109	Jan	31	1983			467.5
110	Feb	7	1983			477.9
111	Feb	14	1983	3	4	475.5
112	Feb	21	1983			470.5
113	Feb	28	1983			472.5
114	Mar	7	1983			486.3
115	Mar	14	1983	3	4	484.9
116	Mar	21	1983			482.3
117	Mar	28	1983			476.3
118	Apr	4	1983			497.5
119	Apr	11	1983	4	4	504.2
120	Apr	18	1983			502.8
121	Apr	25	1983			492.2
122	May	2	1983			489.1
123	May	9	1983			497.0
124	May	16	1983	3	5	497.4
125	May	23	1983			490.6
126	May	30	1983			488.6
127	Jun	6	1983			507.3
128	Jun	13	1983	3	4	509.4
129	Jun	20	1983			505.3
130	Jun	27	1983			494.0
131	Jul	4	1983			510.2
132	Jul	11	1983	3	4	519.9
133	Jul	18	1983			511.7
134	Jul	25	1983			502.1
135	Aug	1	1983			505.4
136	Aug	8	1983	4	5	513.9
137	Aug	15	1983			513.0
138	Aug	22	1983			506.1
139	Aug	29	1983			499.7
140	Sep	5	1983			513.3
141	Sep	12	1983	3	4	519.5
142	Sep	19	1983			513.5
143	Sep	26	1983			501.1
144	Oct	3	1983			510.9
145	Oct	10	1983	4	5	523.4
146	Oct	17	1983			522.3
147	Oct	24	1983			511.9
148	Oct	31	1983			509.8
149	Nov	7	1983			523.5
150	Nov	14	1983	3	4	526.1
151	Nov	21	1983			520.9
152	Nov	28	1983			517.6
153	Dec	5	1983			529.4
154	Dec	12	1983	3	4	533.0
155	Dec	19	1983			533.2
156	Dec	26	1983			529.4
157	Jan	2	1984			541.3
158	Jan	9	1984	4	5	551.0
159	Jan	16	1984			536.8
160	Jan	23	1984			520.5
161	Jan	30	1984			508.8

Figure 7-2. Data used to produce time correlation factors.

Sample Size Differences Within A Sample Period

There are many ways to handle the differences in number of samples within a sample period. To approach this problem, one must determine the best space in which to deal with the data. Looking at the periodicity of the data, there are always 52 weeks in a year, and 12 months in a year. However, the week and month boundaries do not fall in the same place, but vary at the end of these periods. In the above example where the data is on a weekly basis, there are 4 samples in most of the sample periods, but there are 5 samples in four periods in a year. The result will depend upon when one chooses to start and end the sample period.

Taking this a step further, the yellow highlights typically fall in the last week or first week of a month, and this is typically determined by which is closer to the start or end of the month. However, when there are holidays, e.g., Thanksgiving, this may change (see sample 47).

In trying to determine rules to follow to recognize correlation cycles, it is clear that some cycles are 4 weeks and some are 5 weeks. Sometimes this correlates with the number of days in a given month that fall on the published date. However, this is not always true. It appears that there are 4 times per year where there are 5 sample days in a period (≈ 1 month) - and there are always 12 of these periods occurring within the data observed in a year.

There are 13 cases in Figure 7-2 where there are 5 weeks in a period. In 8 of these cases all 5 samples fall in the same period. However, in the other 5, one period starts in the month before and four periods end in the month after.

Thus it appears that one can define 12 periods corresponding to each year, where each period may contain 4 or 5 weeks, depending upon the peak of the cycle. If we define the peak to occur at the end of the cycle, then the peak occurs in the last week of the period, and may correspond to the 4th or 5th week of the period. Thus the database that is used to represent the space and store the data must provide for 4 or 5 week periods.

Requirement For More Data

To perform the correlation analysis, one must consider that the weekly cycles are varying with the year as well as the month. Therefore, one must use multiple years of data to determine the parameters to represent the correlation with sufficient accuracy. Another consideration is the nature of the current markets versus those of thirty years ago.

To consider this, data representing a recent period of 3.5 years is shown in Figures 7-3 and 7-4. We note that the cycles appear to be the same. However, when looking at the data shown in Figures 7-1 and 7-2, we find a reversal of the peaks and valleys at the transition points. What was colored in yellow before is now colored in orange. This is because the minimum points that were in yellow are now maximum points in orange. Somewhere between 1984 and 1998, there was a shift in the time correlation. However, the structure of the cycles appears the same, so that the correlation analysis and resulting data structures remain the same. Only the data changes within the cycles, with the reversal from valleys to peaks at the transition points.

It appears that the transition about the new year causes the same “highest” peak. However, it is not clear that there is as much of a change about the April 15th tax deadline.

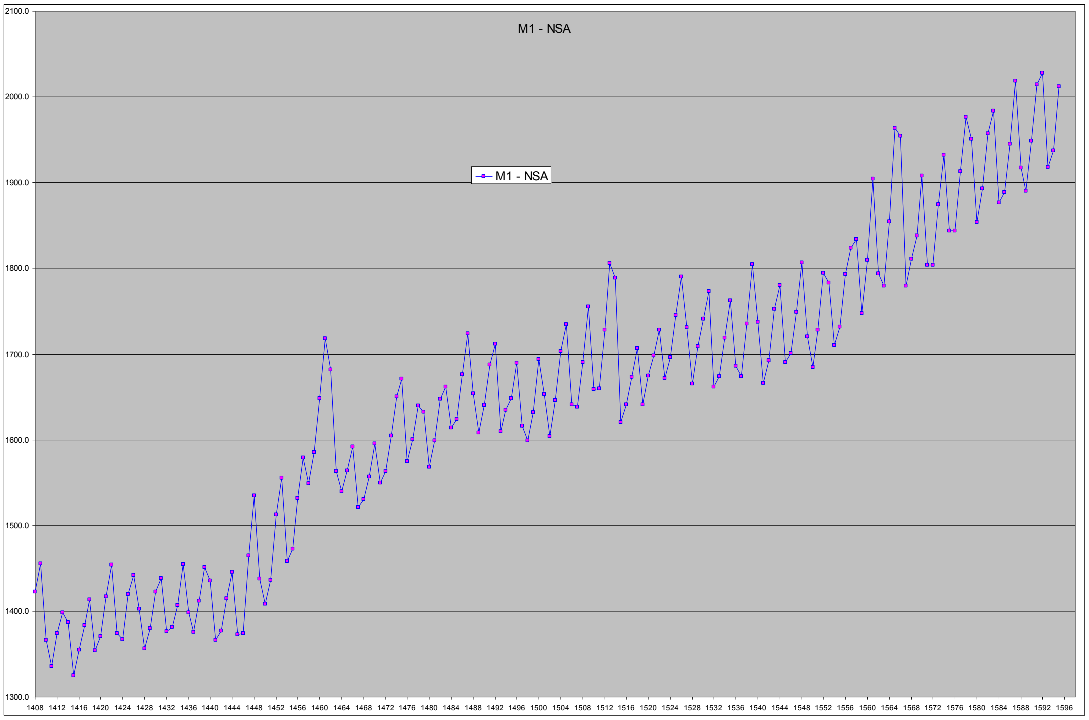


Figure 7-3. Plot of 3.5 years of data from 2008 to 2011.

Defining The Periods

Because the periods change in terms of numbers of weeks per “month” (4 or 5), and because the transitions may occur twice in the same month (e.g., in March), it is best to define these cycles in terms of transitions and periods instead of months, even though there are 12 every year. Thus we will define “annual” transitions, even though both may occur in the same year. Similarly, we can think of “monthly” transitions, even though there may be 2 in a month. A monthly period may contain 4 or 5 weeks.

The critical transition occurs around the change of a month, and may take two directions, a 4 week period or a 5 week period. These cycle times may be different for the same month in different years. Since the coefficients will likely be different, we will store two sets for each month, and keep a flag indicating the number of weeks for each month. An example of how these may be stored within each year is shown below.

```
1  PERIOD(12)
   2  NUMBER_OF_WEEKS        INDEX *** (4 or 5)
   2  WEEK_4_VALUE(4)
   2  WEEK_5_VALUE(4)
```

To determine the transition between months, one must know the number of days in a month (28, 29, 30, or 31). For example, it appears that for months less than 31 days, the transition occurs after the 27th day and before the 3rd day. For 31 day months, it may start after the 28th day, etc. A more recent example is shown in Figure 7-4.

1408	Dec	24	2007			1422.5
1409	Dec	31	2007			1455.8
1410	Jan	7	2008			1366.6
1411	Jan	14	2008	3	4	1335.6
1412	Jan	21	2008			1374.5
1413	Jan	28	2008			1398.6
1414	Feb	4	2008			1386.8
1415	Feb	11	2008	4	4	1325.0
1416	Feb	18	2008			1355.3
1417	Feb	25	2008			1383.4
1418	Mar	3	2008			1413.4
1419	Mar	10	2008			1354.4
1420	Mar	17	2008	3	5	1370.5
1421	Mar	24	2008			1417.1
1422	Mar	31	2008			1454.0
1423	Apr	7	2008			1373.9
1424	Apr	14	2008	3	4	1366.9
1425	Apr	21	2008			1419.9
1426	Apr	28	2008			1442.1
1427	May	5	2008			1403.0
1428	May	12	2008	4	4	1356.3
1429	May	19	2008			1380.2
1430	May	26	2008			1422.4
1431	Jun	2	2008			1438.7
1432	Jun	9	2008			1376.7
1433	Jun	16	2008	3	5	1381.4
1434	Jun	23	2008			1406.9
1435	Jun	30	2008			1455.1
1436	Jul	7	2008			1398.3
1437	Jul	14	2008	3		1376.0
1438	Jul	21	2008		4	1411.9
1439	Jul	28	2008			1451.0
1440	Aug	4	2008			1435.6
1441	Aug	11	2008	4	4	1366.5
1442	Aug	18	2008			1377.2
1443	Aug	25	2008			1414.6
1444	Sep	1	2008			1445.3
1445	Sep	8	2008			1373.0
1446	Sep	15	2008	3	5	1374.4
1447	Sep	22	2008			1464.7
1448	Sep	29	2008			1534.7
1449	Oct	6	2008			1437.8
1450	Oct	13	2008	4	4	1408.5
1451	Oct	20	2008			1436.0
1452	Oct	27	2008			1512.7
1453	Nov	3	2008			1555.4
1454	Nov	10	2008			1458.7
1455	Nov	17	2008	3	4	1472.4
1456	Nov	24	2008			1532.1
1457	Dec	1	2008			1578.8
1458	Dec	8	2008			1548.9
1459	Dec	15	2008	3	5	1585.5
1460	Dec	22	2008			1648.0
1461	Dec	29	2008			1717.9
1462	Jan	5	2009			1681.9
1463	Jan	12	2009	4	4	1563.3
1464	Jan	19	2009			1539.6
1465	Jan	26	2009			1564.1
1466	Feb	2	2009			1592.2
1467	Feb	9	2009			1521.1
1468	Feb	16	2009	3	4	1530.8
1469	Feb	23	2009			1556.8
1470	Mar	2	2009			1595.7
1471	Mar	9	2009			1550.0
1472	Mar	16	2009	4	5	1563.2
1473	Mar	23	2009			1604.6
1474	Mar	30	2009			1650.7
1475	Apr	6	2009			1671.3
1476	Apr	13	2009			1575.0
1477	Apr	20	2009	3	4	1600.5
1478	Apr	27	2009			1639.4
1479	May	4	2009			1632.7
1480	May	11	2009			1568.3
1481	May	18	2009	3	4	1599.1
1482	May	25	2009			1647.2
1483	Jun	1	2009			1661.8
1484	Jun	8	2009			1613.9
1485	Jun	15	2009	3	5	1624.2
1486	Jun	22	2009			1676.2
1487	Jun	29	2009			1723.7
1488	Jul	6	2009			1654.2
1489	Jul	13	2009	4	4	1608.1
1490	Jul	20	2009			1640.2
1491	Jul	27	2009			1687.6
1492	Aug	3	2009			1711.6
1493	Aug	10	2009			1609.6
1494	Aug	17	2009	3	5	1634.4
1495	Aug	24	2009			1648.1
1496	Aug	31	2009			1689.9
1497	Sep	7	2009			1616.1
1498	Sep	14	2009	3	4	1599.3
1499	Sep	21	2009			1632.2
1500	Sep	28	2009			1694.1
1501	Oct	5	2009			1653.5
1502	Oct	12	2009	4	4	1604.0
1503	Oct	19	2009			1646.2
1504	Oct	26	2009			1703.2
1505	Nov	2	2009			1734.7
1506	Nov	9	2009			1640.8
1507	Nov	16	2009	3	5	1638.6
1508	Nov	23	2009			1690.3
1509	Nov	30	2009			1755.1
1510	Dec	7	2009			1659.0
1511	Dec	14	2009	3	4	1660.0
1512	Dec	21	2009			1728.2
1513	Dec	28	2009			1806.2
1514	Jan	4	2010			1788.9
1515	Jan	11	2010	4	4	1620.2
1516	Jan	18	2010			1640.9
1517	Jan	25	2010			1673.0
1518	Feb	1	2010			1707.0
1519	Feb	8	2010			1641.4
1520	Feb	15	2010	3	4	1675.0
1521	Feb	22	2010			1698.1
1522	Mar	1	2010			1728.3
1523	Mar	8	2010			1671.9
1524	Mar	15	2010	3	5	1696.2
1525	Mar	22	2010			1745.2
1526	Mar	29	2010			1790.0
1527	Apr	5	2010			1731.4
1528	Apr	12	2010	4	4	1665.5
1529	Apr	19	2010			1708.7
1530	Apr	26	2010			1740.9
1531	May	3	2010			1772.8
1532	May	10	2010			1661.9
1533	May	17	2010	3	5	1673.8
1534	May	24	2010			1718.7
1535	May	31	2010			1762.2
1536	Jun	7	2010			1686.3
1537	Jun	14	2010	3	4	1673.9
1538	Jun	21	2010			1735.0
1539	Jun	28	2010			1804.4
1540	Jul	5	2010			1737.7
1541	Jul	12	2010	4	4	1666.1
1542	Jul	19	2010			1692.3
1543	Jul	26	2010			1752.5
1544	Aug	2	2010			1780.5
1545	Aug	9	2010			1690.2
1546	Aug	16	2010	3	5	1700.8
1547	Aug	23	2010			1748.7
1548	Aug	30	2010			1806.7
1549	Sep	6	2010			1720.4
1550	Sep	13	2010	3	4	1684.7
1551	Sep	20	2010			1727.9
1552	Sep	27	2010			1794.7
1553	Oct	4	2010			1783.0
1554	Oct	11	2010	4	4	1710.6
1555	Oct	18	2010			1731.7
1556	Oct	25	2010			1792.8
1557	Nov	1	2010			1823.9
1558	Nov	8	2010			1833.7
1559	Nov	15	2010	3	5	1747.5
1560	Nov	22	2010			1809.2
1561	Nov	29	2010			1904.2
1562	Dec	6	2010			1793.6
1563	Dec	13	2010	3	4	1779.9
1564	Dec	20	2010			1854.6
1565	Dec	27	2010			1963.6
1566	Jan	3	2011			1954.6
1567	Jan	10	2011	4	5	1779.6
1568	Jan	17	2011			1810.7
1569	Jan	24	2011			1838.0
1570	Jan	31	2011			1907.7
1571	Feb	7	2011			1803.5
1572	Feb	14	2011	3	4	1804.1
1573	Feb	21	2011			1874.5
1574	Feb	28	2011			1932.2
1575	Mar	7	2011			1843.6
1576	Mar	14	2011	3	4	1844.0
1577	Mar	21	2011			1912.9
1578	Mar	28	2011			1976.3
1579	Apr	4	2011			1950.6
1580	Apr	11	2011	4	4	1853.7
1581	Apr	18	2011			1893.0
1582	Apr	25	2011			1957.5
1583	May	2	2011			1983.5
1584	May	9	2011			1876.6
1585	May	16	2011	3	5	1888.7
1586	May	23	2011			1945.1
1587	May	30	2011			2018.3
1588	Jun	6	2011			1917.2
1589	Jun	13	2011	3	4	1890.2
1590	Jun	20	2011			1948.9
1591	Jun	27	2011			2014.2
1592	Jul	4	2011			2028.0
1593	Jul	11	2011	3	4	1918.1
1594	Jul	18	2011			1937.4
1595	Jul	25	2011			2012.2

Figure 7-4. 3.5 years of data from 2008 to 2011.

Algorithm For Selecting Weights

The following algorithm may be used to select the weights to apply to a given week.

- Start the algorithm immediately following a transition point (for a new "month").
- Based upon the month corresponding to the period, determine the days in that period. This may be 28, 29, 30 or 31.
- Determine the transition week at the end of the period based upon the days in the current month.
- Determine the transition day for the period. It may be a day in the prior month or in the beginning (first 3 days) of the next month.
- Determine the number of weeks (4 or 5) in the current period.

Given the transition week and number of weeks in the period, obtain the weights from the database.

Algorithm For Computing The Weights

To compute the weights for a given week, build a table by year, using enough years to obtain a sufficiently accurate statistic. Note that the periods will change 7 times a year due to the rotation of the days of the week. Thus a minimum of 7 years is needed to get a sufficient statistic.

Within each year, store tables by period number using 12 periods.

Within each period, store data by week, allowing for 4 or 5 weeks, with two elements per week - one each for 4 week periods and one for 5 week periods. Data for each of the weeks within the period must be used to compute the (4 or 5 week) average value for the period.

Compute the normalized deviations for each week in the period relative to the average for the period.

Add the specific deviations (for each week within each transition period) for all years to obtain the sum for the year, and divide by the number of years to get the average deviations (for each week within each transition period).

Weighting Data By Year

To obtain more accurate weights when they may be changing over the years, one may weight the data by year. For example, one may use the following formula to adjust the weights to reflect more current dynamics:

$$\mathtt{WEIGHT} = \frac{(\mathtt{YEAR\_POINTER} + \mathtt{M})}{\mathtt{N}}$$

where M is a boosting factor, e.g., 10 years; and N represents a total number of years, e.g., 15. This provides a linear bias weighting favoring the recent years.

Using The Weights To Improve Prediction Accuracy

To improve the prediction accuracy, one can use the weights to produce the predictions by multiplying the predicted value by the normalized deviations for each week. To do this, one must know the number of weeks in the period to select the correct weights (from the 4 week or 5 week weights). The VisiSoft data structure below is used to compute the weights for a year of data, i.e., 12 periods per year, and up to 5 weeks per period. Observations include WEEK_COUNT which determines actual number of weeks in that period. The actual year, month, and day are stored as well as the average value of the period, the actual value for the week and the deviate.

```
CALENDAR_WEIGHTS
    1  PERIOD_STATE                          STATUS START_OF_PERIOD
                                                    INSIDE_PERIOD
                                                    NO_MORE_PERIODS
    1  WEEK_STATE                            STATUS NEW_PERIOD
                                                    INSIDE_PERIOD
    1  MONEY_MULT                            DREAL
    1  CAL_WEIGHT                            DREAL
    1  PERIOD                                INTEGER
    1  WEEK_COUNT                            INTEGER
    1  PERIODS QUANTITY(12)
       2  WEEKS  QUANTITY(5)
          3  WEEK_4_WEIGHT                   REAL
          3  WEEK_5_WEIGHT                   REAL
```

An example of the resulting weights are shown in the table below.

```
* P          WEEK_4    WEEK_5
* E    WK    WEIGHT    WEIGHT
******************************
  1     1    -.00187    .01949
  1     2    -.01101   -.01676
  1     3    -.00046   -.02254
  1     4     .03010   -.02092
  1     5     .00000    .00818
  2     1    -.00924    .00079
  2     2    -.02135   -.00535
  2     3     .00783   -.00294
  2     4     .06923    .00238
  2     5     .00000    .01138
  3     1    -.01327   -.00497
  3     2    -.01635   -.01934
  3     3     .01052   -.00569
  3     4     .07174    .01821
  3     5     .00000    .05993
              ...
              ...
              ...
 10     1    -.00901   -.00132
 10     2    -.01492   -.02935
 10     3     .00966   -.01886
 10     4     .05299    .02535
 10     5     .00000    .07972
 11     1    -.01348   -.00049
 11     2    -.03274   -.01333
 11     3     .00704   -.00848
 11     4     .11000    .01168
 11     5     .00000    .03532
 12     1    -.01781   -.01040
 12     2    -.02272   -.03723
 12     3     .01969   -.01398
 12     4     .09043    .06234
 12     5     .00000    .09045
```

8. ENVELOPE PREDICTION MODEL

The envelope model will produce predictions of daily limits within which future prices will fall for up to a specified number of days in the future. The sequence of daily limits comprises an envelope over the *prediction horizon,* e.g., 12 days into the future. The prediction statement is that future prices will fall within the envelope with a specified probability. This is measured by the percentage of times that *unseen* actual values fall within the predicted envelope over some *looking-back horizon.* This model will also produce confidence level estimates for the probability statement based upon recent history, i.e., history that goes back a given period of time, e.g., fifty sets of envelopes. This is an *historic time frame* that must be selected based upon the changing nature of a specific system.

ENVELOPES

We have used 80% envelopes in the past to determine the probability of being inside. We have also used 95% as the confidence level in the envelopes, implying that there is a 5% margin of error on the envelopes. These numbers are used to change the width of the envelopes to match the prediction accuracy adaptively. They are computed based upon a Looking Back Horizon (LBH) where 95% confidence implies not being outside the envelopes more than 1 time in 20. In the past, an LBH of 50 time steps was used to obtain the confidence level.

The envelope model may adaptively open up or close down the envelope to maintain a statistical confidence limit for the prediction probabilities. This model will determine the width of the deviates at each time point in the future to insure that the future prices fall within the envelopes 80% of the time. It will be adjusted to obtain a 95% confidence level in the 80% envelopes.

A prediction is described by the envelopes, within which the predicted value must fall a specified percentage of the time (e.g., 80%) with a specified confidence level (e.g., 95%) over a prescribed looking-back horizon (e.g., a 50 data point ensemble), for a prescribed historic time frame (e.g., 250 data points).

The test for meeting the confidence level criteria requires multiple ensembles of contiguous data points looking back into the past. For example, using three years of weekly data (our historic time frame) would produce 3 x 52 = 156 points. With an ensemble length of 50, there are 106 ensembles of contiguous points that overlap by all but one point. Then at least 95% of those ensembles must have at least 80% of the actual points fall within the predicted envelope.

Updating Envelope Estimates

Figure 8-1 illustrates envelopes (e.g., 80%) used to characterize predictions. The current set of residuals is used to update envelope estimates. One must determine the number of times that the actuals, Z(T), went outside the envelope, for each prediction step, for each time step in the looking back horizon. Then one must use the number and size of the violations to expand or contract the envelope. To be conservative, the expansion must be accelerated faster than the contraction.

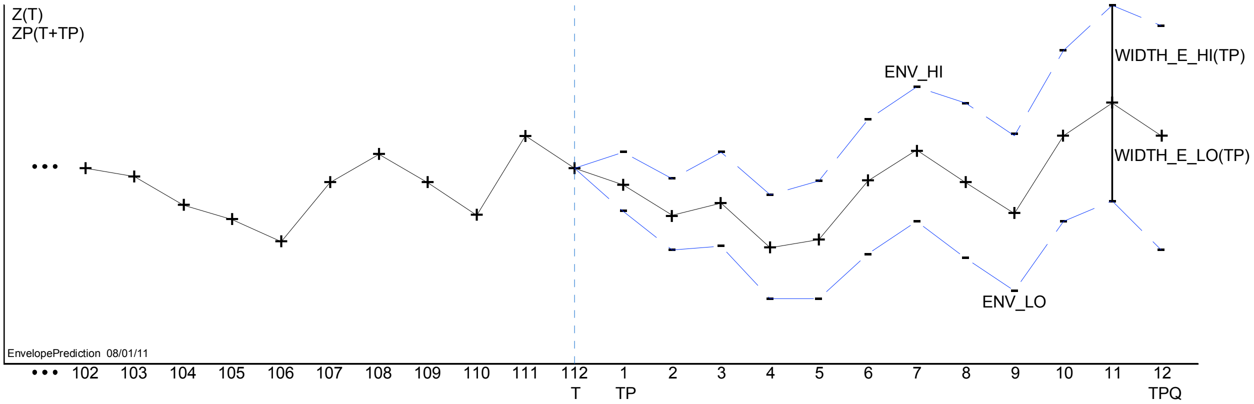


Figure 8-1. Observation Data, Prediction Matrix & Residuals Matrix.

Initial Envelope

We are concerned with producing initial values for the low and high envelopes for each prediction step:

```
ENV_LO(TP) & ENV_HI(TP)   for TP = 1, 2, ..., TPQ
```

One approach to determine values for an initial envelope is to compute the distance from the mean of a distribution to determine the probability of being inside the boundaries as illustrated in Figure 8-1. For example, if the distribution is normal, one may use 1.5 standard deviations from the mean (≈ 82%) as shown in Figure 8-2 to achieve an 82% probability envelope.

Thus, assuming that the predicted values ZP at TP time steps in the future represent the mean of a normal distribution, one may multiply the mean deviations at each time step by the number of standard deviations required to produce an initial envelope with the desired probability of being within the envelope. This initial envelope may then be used to compute actual statistics for remaining inside the envelope over the looking back horizon as actual data is infused. This approach will provide for updated predicted values for the envelope widths at each future time step.

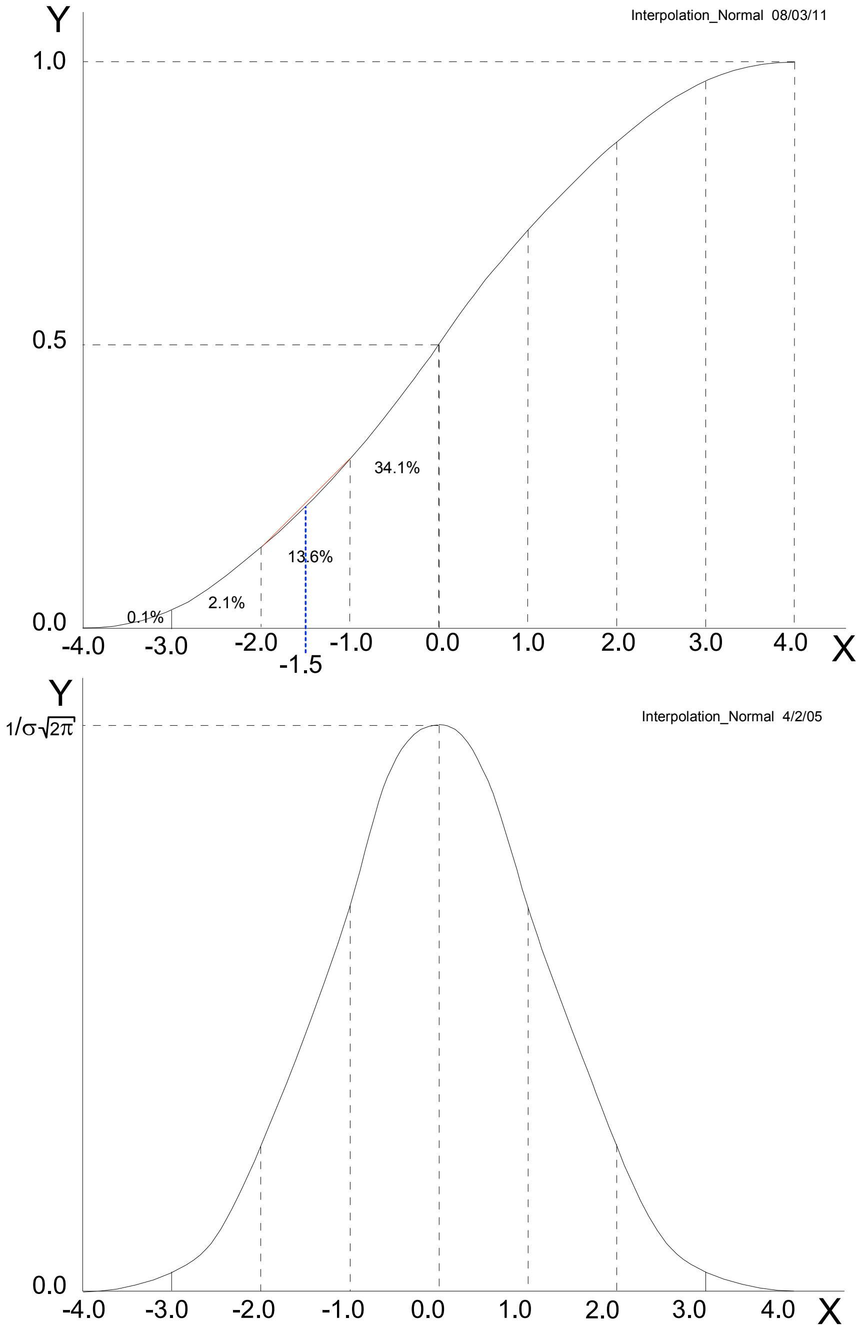


Figure 8-2. Using the standard deviation to determine probability limits.

Updated Envelopes

Given that the initial envelope widths are used to collect statistics over the first look-back horizon, one can proceed to use the incoming observation data to determine the actual statistics of being outside the envelope. If the number of times one is outside the envelope over the look-back horizon exceeds 20%, then one must increase the width of the envelopes to maintain the 80% probability. If it is sufficiently less than 20%, then one may reduce the width of the envelopes. What is clear is that one wants the width of the 80% envelope to be as small as possible.

Minimizing The Envelopes

When using two driving forces, one may determine that the data is moving up and down beyond the envelopes derived from these forces alone. What may be needed is another driving force that operates in conjunction with the others to predict the future states. Alternatively, one may find that, after characterizing the first two driving forces, a high correlation now exists with the calendar.

Chapter 13 describes how one can optimize the parameters in each submodel using VisiSoft, producing more accurate results for each driving force. As a final step this may be done for a calendar correlation model. An example of the result of such a model is illustrated in Figure 8-3. As illustrated by this model, the envelopes moved up and down with the data.

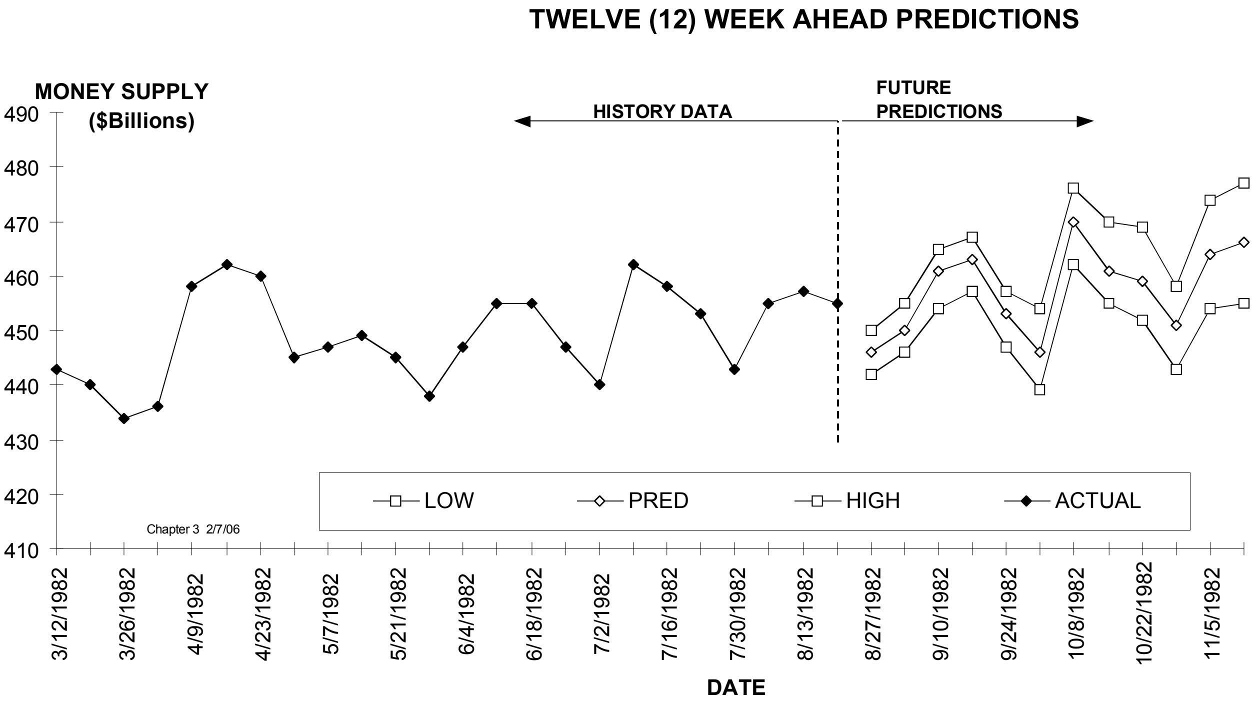


Figure 8-3. Using the standard deviation to determine probability limits.

Dealing With Volatility

Market volatility must be tracked to determine the confidence in the predictions. As markets become more volatile, there is a greater chance of large swings, and the possibility of higher profit, but inherently more risk of loss.

The envelopes must be widened when the 80% rule is violated more than 5% of the time. The 5% accounts for the confidence margin, but may be too optimistic. In any event, the envelopes should be widened faster than they are narrowed, and this must be assessed statistically to ensure that the defined percentages are maintained. The widening occurs because the prediction model is not sufficiently accurate, implying that additional information is not being accounted for (it may not be available and must be treated as random variations). For example, when people start running scared, they may be considered as overreacting by the marketeers, but in fact they are protecting their assets. When such variations occur, the control system should signal to stop trading (get out of the market) because of the unpredictable swings that may be too large to cover.

Envelope Controls

Decisions to Open/Close the envelopes may be based upon the points closest to the envelope boundary. This implies ordering the points based upon nearness to the boundary. For example, those within X% of the boundary could be weighted based upon nearness and counted on a weighted basis to determine how much to open or close the envelope. One can also consider the open/close process to occur on a prescribed incremental basis (e.g., prescribed notches).

Predicting Future Time Steps

When performing future predictions, the model is used to produce predictions at each new major time step, T, i.e., where outcomes are recorded for future values. For multi-step prediction, the clock is advanced multiple time steps into the future. Then it must be set back to the next actual time step to update adaptive model parameters. This is followed by another set of prediction steps.

When the prediction steps are produced, databases that store the prediction information, e.g., the envelope values, must be updated. These are stored for tracking the width of the envelopes to make adaptive decisions on the future width required to maintain specified accuracy measures. These computations require tracking prediction steps as well as the normal time steps.

Finally, one must maintain a “Looking Back” horizon over which we compute: (1) the probability of being inside the envelope; and (2) the confidence level in the probability statement. As described in Chapter 3, choice of this horizon depends upon the trade-off of having enough samples versus ensuring that the model represents the current time frame, which is of special concern when using adaptive parameter optimization.

9. THE CONSTRAINED OPTIMAL CONTROL PROBLEM

To solve the real problems of controlling complex systems that are subject to dynamic parameter variations, one must solve the constrained optimal control problem. This chapter describes the elements required to address the problem of finding the best solution while meeting hard (inequality) constraints. We will start by looking at the time-invariant (single time point) problem.

Objectives And Constraints

As described in the linear or nonlinear programming literature, optimization problems are defined in terms of objectives to be optimized and constraints that must be met. Except for special (rare) cases, there is only one optimal solution. So objectives are generally grouped and weighted into a single *objective function*. In practice, the constraints are more important than the objective function in that they must be satisfied to provide a *feasible solution*. This translates to the selection of a control sequence, or set of actions that, when applied as inputs to a system, yield a response that satisfies the operational constraints for that system. Examples of such constraints are: limits on risk of failure or catastrophe, limits on time, limits on personnel, limits on fuel, limits on platform availability, flight path restrictions, communications restrictions, etc. These are referred to as *hard constraints*, in that a violation of any such constraint renders the selected control sequence unacceptable (i.e., the solution infeasible).

Constraints may be mapped onto the n-dimensional space of parameters that affect them in terms of a vector of variables, V. Constraints are then posed in terms of the variables in this space such that a constraint function H is positive when the constraint is satisfied and negative when it is violated. The boundary of a constraint is represented by a surface defined at H(V) = 0. Figure 9-1 below illustrates such a mapping for four constraints viewed in a 2D space.

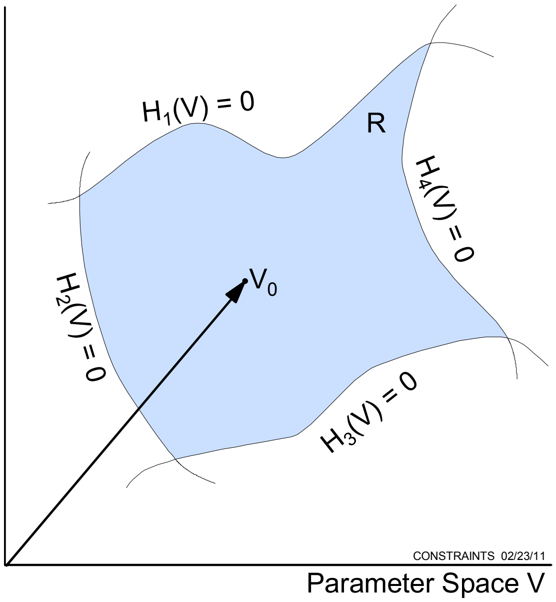


Figure 9-1. Hard constraints defined by surfaces Hi(V) = 0.

By visualizing the surfaces Hi(V) = 0 (the curves bounding the region R in Figure 7-1), one can interpret this geometrically. A solution, Vo, is defined as *feasible* if it satisfies all the constraints. If all points inside the region R ensure that all of the values of the Hi are greater than zero, then R is defined as a feasible region, bounded by the constraint surfaces Hi(V) = 0. For example, a constraint on the probability of being engaged by a missile will depend upon where one travels in x, y, z space. A constraint on fuel will depend upon altitude and distance traveled, a function of the way points of a flight path in (Lat, Lon, Altitude) or (x, y, z). It may also depend upon the position of refueling tankers - a different set of parameters. The dimension of the constraint parameter space can be quite large. In the case of nonlinear systems, there may be more than one feasible region.

Note that this illustration may be for an instant of time, implying that Figure 7-1 is a snapshot from a trajectory in time, showing only the spatial parameters. We will limit the current discussion to be independent of time and deal with trajectories in time in a later section.

The feasible region in Figure 7-1 appears large. This is done to illustrate the definitions. In practical problems, the vector V will depend upon many parameters that affect at least one of the constraints. If the parameter vector, Vo, moves outside the region R, at least one of the constraints is violated. Typically, only a subset of the parameter vector will be optimized.

In addition, the constraint surfaces can be very nonlinear functions of the parameters. In a large parameter space, it may take a significant effort to develop the complete set. There have been studies of such constraint surfaces for various problems, and it is known that they can take on exotic shapes. This can make the feasible region very small in areas of the space. Depending upon how the problem is posed, it is not unusual for the feasible region to occur in multiple disconnected sets, or to be non-existent.

Accounting For Parameter Variations

In most real problems of interest, the actual values of parameters will vary. For example, a refueling tanker may have been tasked to follow a given flight path, but circumstances may cause it to vary off the prescribed path. Target positions will be known to within some range of error. Radar coverage may not be known precisely. This implies that we only know the value of many parameters to within a distribution. We may not know the shape of the distribution, and may only have some knowledge of its bounds - in terms of percentile limit values. This is the classic worst-case design problem. We will not delve into the details here, but will outline it and provide references for detail.

Figure 9-2 illustrates a simple case of what happens when parameter variations are taken into account. If Vo is the selected (nominal) solution, and we apply all of the possible variations, T, on each parameter, a region $\mathbf{r_o}$ will be created as shown. Thus, $\mathbf{r_o}$ is the region of all possible values of parameter vector V, determined by applying all possible variations on these parameters within a specified set of limits on their distributions. This implies that all of the points in $\mathbf{r_o}$ must remain in R, else a constraint may be violated and the solution no longer feasible. This is a complex problem to solve directly, and one typically resorts to Monte Carlo analyses to determine the probability of violating a constraint. An alternative approach is to perform a constraint transformation as defined by Cave, [5], using T in the space of all possible parameter variations. This is illustrated in Figure 9-3.

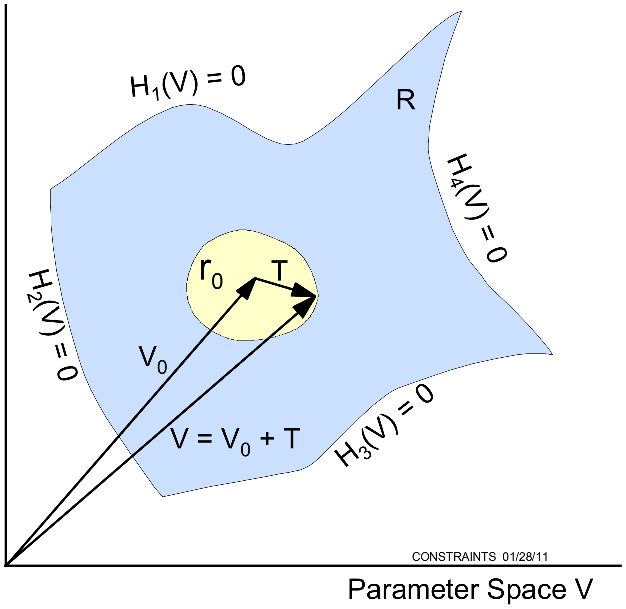


Figure 9-2. Possible variations of the solution due to parameter variations.

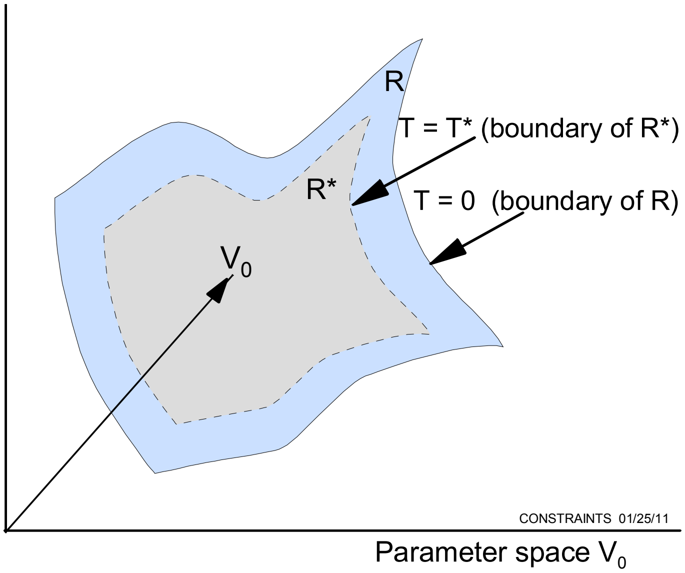


Figure 9-3. Transforming the constraint boundaries using optimization.

The solution illustrated in Figure 7-3 has been verified using Computer-Aided Design (CAD) techniques, see [6], and [7]. A simplified derivation is as follows. For each point, Vo*, on a selected constraint boundary, H, we can determine the value of T = T* that causes H(Vo* + T*) to be most negative. These values form the inner region R* bounded by the surface H(Vo* + T*). Applying this to all of the constraints transforms the original feasible region into a smaller region, R*, such that, if a solution, Vo, falls within the transformed region, it will meet the prescribed constraints under worst case conditions. This is described in further detail in [5].

A major benefit of this approach is that it supports direct optimization and therefore synthesis of a solution that meets worst case constraints. One avoids the iterative approach of finding feasible solutions and then running Monte Carlo analysis to determine if constraints are violated. It is a proven technique that has been used extensively to solve difficult worst case design problems.

Worst Case Design / Optimization

Before considering optimal solutions, one must further investigate the worst case problem, i.e., searching for the feasible region after applying the worst case transformation. By "worst case", we are implying combinations of variations that can realistically occur simultaneously. To properly account for these variations, one must estimate the probabilities of such variations occurring. This implies characterizing the probability distributions of the variations to the extent that the bounds may be sufficiently estimated. Figure 9-4 provides an illustration of a probability distribution where the light green shaded area represents the probability of being within the bounds [Vo-T_D, Vo+T_U]. Alternatively, one may determine the probability of being outside these bounds, or above the Vo+T_U boundary. Given that one can estimate these boundaries and probabilities, they may be used to determine the probability of failure of a system or a mission.

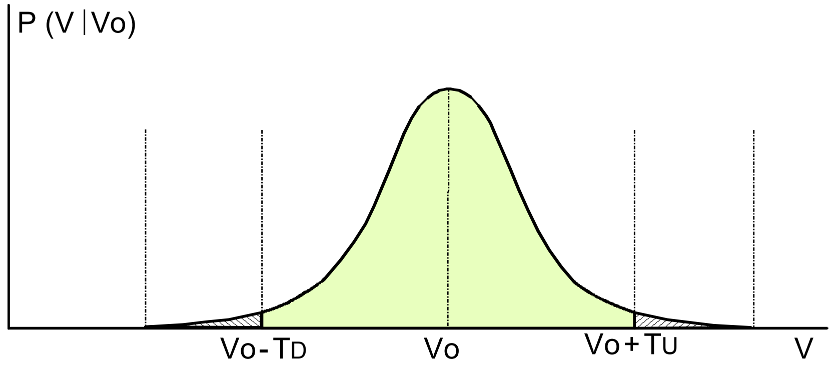


Figure 9-4. Defining the constraint boundaries for optimization.

Estimating such bounds and probabilities may appear quite difficult. However, any form of estimate generally leads to much more information regarding the likelihood of failure. For example, one may consider that crossing a certain boundary yields a high probability of failure, and is considered a GO - NO GO or 1 - 0 situation. In this case, the boundary is the constraint. Knowing the shape of the distribution is not important, but one must know the bounds.

Using Human Judgment

A practical example of constraint boundaries is the requirement to stay out of a restricted area, where crossing any of the boundaries into such an area violates the constraint, rendering the solution a failure. In this case, human judgment may be the best way to determine where to define the bounds to ensure a safe estimate. Another example is the requirement to stay within certain boundaries, e.g., a playbox, where going outside violates the constraint. Such boundaries may be defined geometrically as shown in Figure 9-5 below, where the yellow boundary represents the playbox and the red boundaries represent restricted "no-fly" zones.

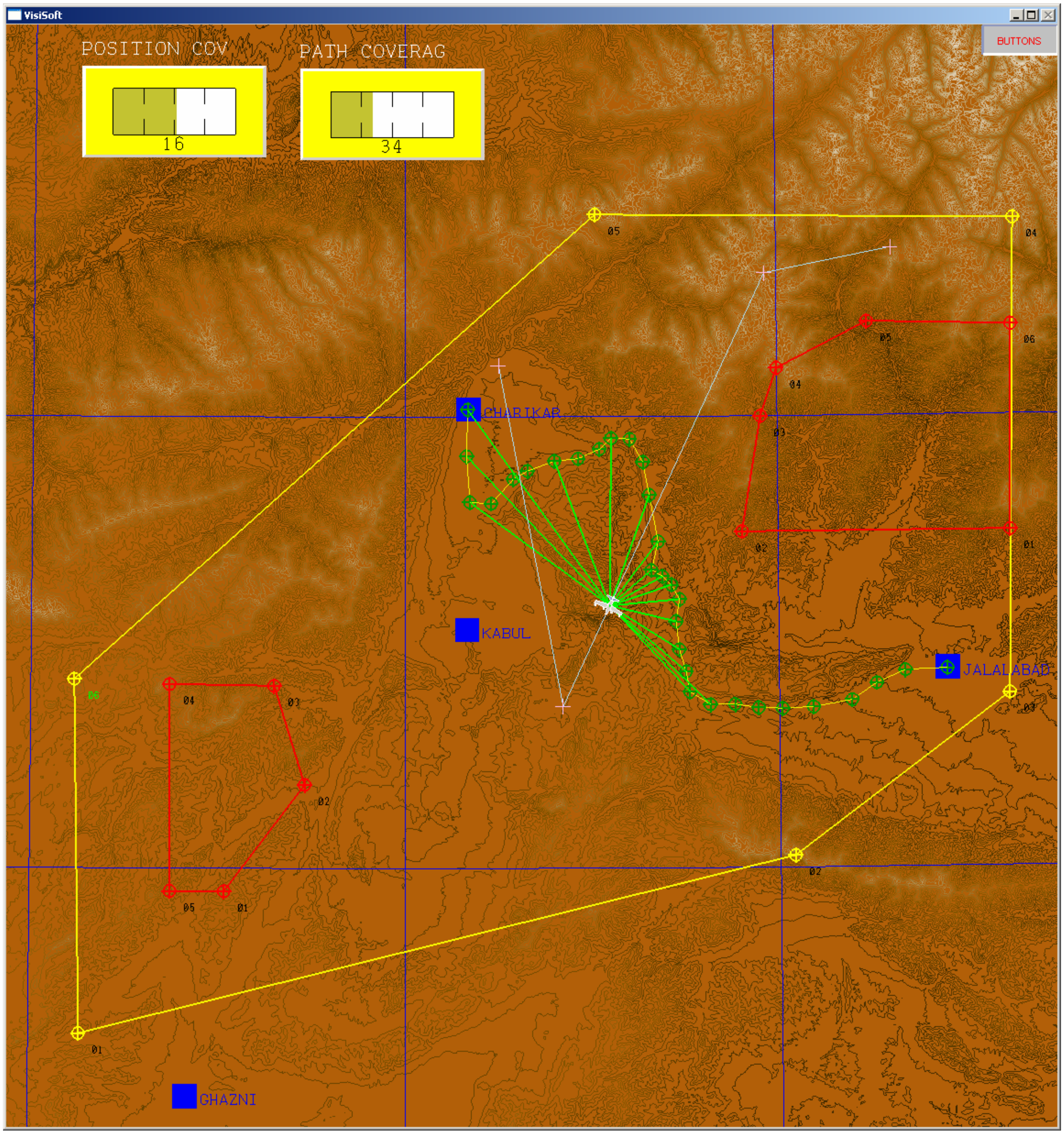


Figure 9-5. Defining constraint boundaries for optimization.

Another example is the constraint on fuel. When optimizing flight paths in air and space, one must be concerned about the constraints on fuel. Because of the variations on use of fuel, the flight time to run out of fuel may be characterized by an error distribution. Again, the problem of where to draw the line for a given airplane on a given set of missions is best estimated with the help of human judgment. In the case of military air flights, one may be able to refuel to achieve an acceptable solution. We note that the difference between the solution using refuel and one with no refuel is typically highly nonlinear (again GO - NO GO), falling into a category known as the bang-bang control problem.

The overall problem illustrated in Figure 7-5 is that of optimizing way-points for a flight path so as to maximize radio connectivity with ground convoys. Again, the use of human judgment is likely to be best when determining many of the constraints. However, there are cases such as those described below that require a mathematical approach to varying parameters that may cause the constraints to be violated.

One may question the mix of human judgment and mathematics. This was discussed in Chapter 1 and subsequent chapters, with the conclusion that any approach that adds a sufficient amount of additional information will help to improve prediction accuracy. By sufficient amount we imply that the level of information is sufficiently greater than the level of noise in the data being considered.

Determining Worst Case Constraint Boundaries

As defined in the section above on parameter variations, constraint boundaries may be defined at H(V) = 0 , where the vector of constraints, H, is defined in (7-1) below.

(9-1)
$$H = \begin{bmatrix} h_1(V) \\ . \\ . \\ . \\ hm(V) \end{bmatrix}$$

One may define the constraint boundaries such that all of the constraints are met when H(V) ≥ 0 . We note that some or all of the elements of the solution vector may affect the value of a given constraint function. Given a nominal solution vector, Vo, one must consider the parameter variations about nominal that determine the V vector. These variations are defined by the T vector in (9-2) below (often referred to as the tolerance vector in engineering systems).

(9-2)
$$T = \begin{bmatrix} t_1 \\ . \\ . \\ . \\ tn \end{bmatrix}$$

It is important to understand that each tolerance component, ti, is associated with the corresponding nominal component, voi, as shown in (9-3) below.

$$V = \begin{bmatrix} v_1 \\ . \\ . \\ . \\ v_n \end{bmatrix} = \begin{bmatrix} vo_1 + t_1 \\ . \\ . \\ . \\ vo_n + t_n \end{bmatrix} \tag{9-3}$$

As stated above, a critical part of the problem is identifying the limits, Td and Tu, on the tolerance variation components. This is illustrated in (9-4).

$$\begin{bmatrix} td_1 \\ . \\ . \\ . \\ td_n \end{bmatrix} \leq \begin{bmatrix} t_1 \\ . \\ . \\ . \\ t_n \end{bmatrix} \leq \begin{bmatrix} tu_1 \\ . \\ . \\ . \\ tu_n \end{bmatrix} \tag{9-4}$$

Given that Td ≤ T ≤ Tu, one must find the vector T* that minimizes each constraint function. This produces the worst-case transformation on each of the constraint boundaries as shown in Figure 9-3. Then, given that a nominal vector Vo can be found that meets the transformed constraints, one searches for the nominal vector that maximizes the objective function.

Finding Stable Feasible Solutions

In practice, it is not unusual for the transformed feasible region to be null, i.e., the feasible region has disappeared (there are no feasible solutions). This implies that the problem as posed cannot be solved without violating one or more constraints under worst case conditions.

In this case, one must go back and rethink the problem, and this usually means relaxing one or more constraints or changing the overall design architecture to get a solution. One can then analyze different solutions and determine which constraints are hard to meet. Alternatively, with good optimization techniques, this information may be produced as a by-product of the feasible search process.

Before moving on to the optimal control problem in the time domain, there are two other problems that must be addressed when attempting to solve highly nonlinear constrained optimization problems. First, it is not unusual to find multiple disconnected feasible regions. This means that the optimization algorithms must be able to seek out multiple regions for better solutions.

Second, solutions may be unstable. In the case of a feasible solution, this occurs when the solution is not a worst case solution but is very close to a constraint boundary. This implies that a very small change will render the solution infeasible, i.e., one or more constraints is easily violated. At this point, judgment must be used. Either these conditions must be anticipated and accounted for in advance, or a decision must be made on the spot. Again, with good optimization techniques, this information may be a by-product of the feasible search process.

Another form of instability occurs with optimal solutions when the objective function has very narrow peaks. Again, with very small changes in the solution, large changes can occur in the value of the function being optimized. In a very nonlinear problem, this can be significant. All of these difficulties may be accounted for and alerted using good optimization techniques. Alternatively, one may have to resort to Monte Carlo analysis.

The Worst Case Optimal Control Problem

As indicated above, a control sequence implies a trajectory in time, e.g., a flight path. In the constrained optimal control problem, if there is a feasible region, the control sequence (and resulting trajectory) is bounded by a constraint manifold in time and space. This is illustrated in Figure 9-6. Instead of finding a solution to stationary problems as described above, one must find a sequence of steps that weaves a trajectory through this manifold without going out of bounds. Clearly, this is a much more difficult problem to pose and solve. This problem - finding a sequence of controls that weaves the critical system parameters through a set of constraint boundaries without a violation - is the *worst-case constrained* optimal control problem.

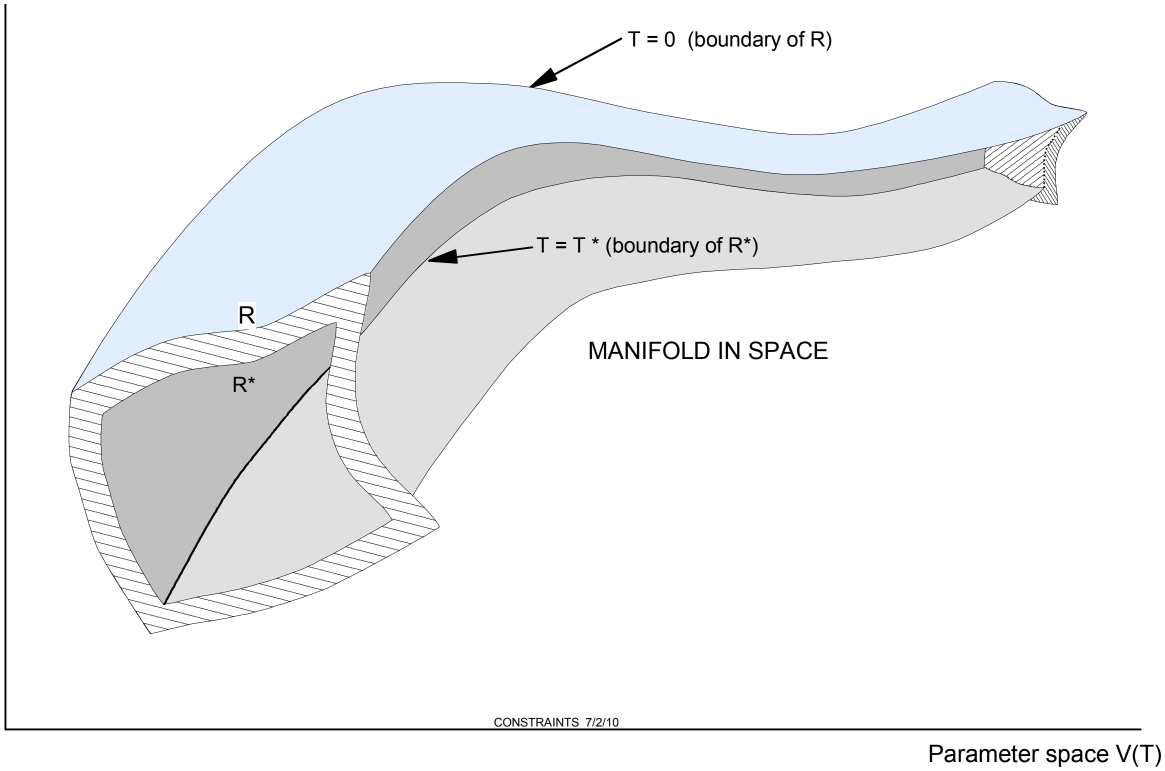


Figure 9-6. Transforming the constraint boundaries using optimization.

Figure 9-6 is a simplified illustration of a problem in 3 dimensions parameterized in time. Most often, the parameters to be controlled are represented by a much larger state vector. Over the course of a scenario, such real world manifolds of interest may take on very complex shapes, being multi-valued as well as nonlinear in space. As a function of time, they are likely to be highly nonstationary as a result of nonhomogeneous inputs.

This all implies that they cannot be addressed in a typical analytical sense mathematically. Typically, the dynamics of such a system must be simulated based upon event driven scenarios. To solve the constrained optimal control problem for a complex system, one may have to run large numbers of simulations, first to find a feasible solution, and then to find the optimal solution.

10. THE STOCHASTIC CONTROL PROBLEM

Nonstationary Considerations

If we want to predict complex system responses, we must be able to model systems driven by forces which are neither stationary nor statistically characterized, i.e., they are bounded but unknown. When driving forces contain random components as well as known functions of time, the model may be reformulated so that *only its random portion* acts as the driving force. As stated above, any known functions of time contained in the driving forces can be treated as a homogeneous part of the system model. It is also possible for stationary random forces to be similarly recast as model error. See for example Papoulis, [23]. Hence, models can be structured so that only nonstationary components remain as driving forces.

As experienced by the author, nonstationary driving forces have a significant influence on the state of most real world systems. Homogeneous models of these types of nonstationary systems are likely to be subject to significant prediction error. Building nonhomogeneous models represents a difficult part of the problem. As an aid to solving this problem, the state space framework provides a conceptual partitioning of the driving forces of a nonhomogeneous model. The modeler can decide whether the known functions of time or statistically stationary components of the state of a system are best represented by driving forces, or by a homogeneous model. It is this conceptual framework which makes the state space framework an attractive tool for modeling nonstationary systems.

As defined in Chapter 5, the general form of equations for a linear dynamic system may be represented in state space as follows,

(10-1) $$X(T+1) = f[X(T), U(T), T],$$

where X is the vector representing the state of the system at any time T, where T and T+1 are just pointers to successive time steps that may vary in size. The state transition operator, f, takes the system to subsequent states, and U is the external driving force vector at time T. The observation equation is represented by,

(10-2) $$Z(T) = h[X(T), T],$$

where Z is the observation vector and h is the transformation from state X to observation Z at any time T. In the case of a nonlinear system, the general form of the state transition equation is given by

(10-3) $$X(T+1) = f[X(T+1), X(T), U(T), T].$$

The general *linearized* form of a nonlinear dynamic system may be formulated as follows,

(10-4) $$X(T+1) = F(T+1, T)\bullet X(T) + B(T)\bullet U(T),$$

where F(T) contains nonlinear coefficients in X(T+1). The nonlinear solution may be obtained using iterative methods, e.g., describing functions. The measurement model is given for any time T as follows,

(10-5) $$Z(T) = H(T)\bullet X(T).$$

Stochastic Considerations

Most systems of interest encounter disturbances which cannot be controlled or modeled deterministically. Observations of driving forces and system responses can also be corrupted. Such phenomena cause uncertainties, leading to the incorporation of error terms in the model which are characterized stochastically. This leads to the stochastic representation of our model as

(10-6) $X(T+1) = f[X(T+1), X(T), U(T), T, W(T)]$

(10-7) $Z(T) \quad = h[X(T), T, V(T)]$

where W(T) represents uncertainty in the dynamic model, and V(T) represents uncertainty in the observation mechanism. Reference Kalman, [20].

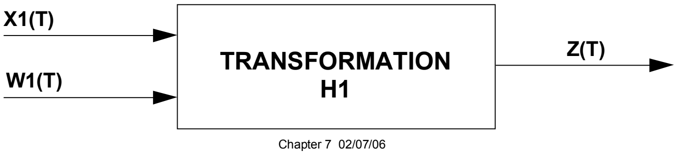
To start we assume that a deterministic model has been built such that the statistics of the uncertainty elements, W and V, can be characterized as normally distributed random processes with zero means, and covariance matrices given by

(10-8) $Q(T) = E[W \bullet W^{T}]$

(10-9) $R(T) = E[V \bullet V^{T}]$.

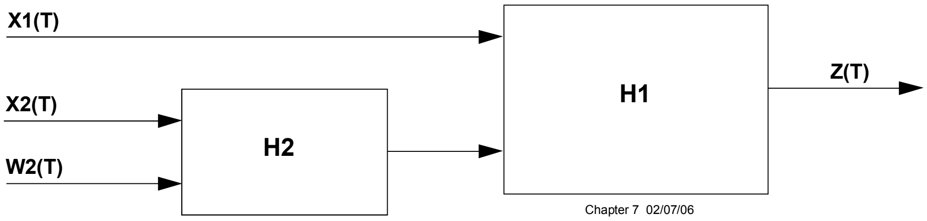
Determining values or bounds for these covariance matrices is part of the model identification process. Depending upon the design process as described in the previous chapter, one may want to reduce this requirement to distributions that are unknown but bounded. This requires a tailored process for characterization of the stochastic processes that are used to project the prediction error.

We will view stochastic processes as generally being composed of known functions of time and functions which appear to vary randomly. Referring to Figure 10-1, Z(T) depends upon $X_1(T)$, a known function of time, and $W_1(T)$, which appears to vary randomly with time, through transformation H1. It may be possible to further investigate W_1 (T) to reduce the random component, as indicated in Figure 10-2. Thus, the error in Z(T) is reduced by adding information X_2 and H_2 relative to H_1.



Chapter 7 02/07/06

Figure 10-1. Transformation H1 with W_1 as random.



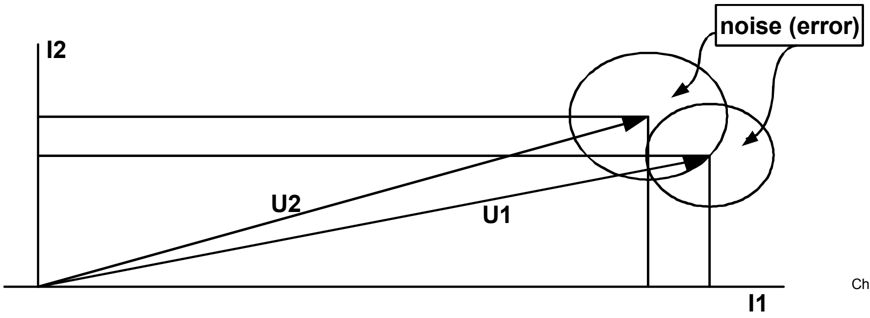
Chapter 7 02/07/06

Figure 10-2. Transformation H2 with W1 broken into known and random components.

Referring back to Figure 5-1, what we actually observe may not agree with our conceptual model of the observation mechanism, and how it relates the state of the system to the response. At this point, one could say that the actual observation mechanism contains error, and terminate further modeling efforts. However, as in engineering, we will assume that the modeler's task is to model the actual mechanism completely and accurately, particularly with regard to any biased errors it contains. However, regardless of the accuracy of the actual observation mechanism, our observation model may be in error, and this must be accounted for. Thus, if V(T) is nonzero, then we have no direct inverse for obtaining the actual value of X(T), and therefore can only estimate the current state of the system using all observations up to and including the current values.

Nonhomogeneous Models - Noise Considerations

When considering the addition of a candidate driving force to our model, we must be aware that the observation data for that driving force typically contains noise (error) as well as information. Figure 10-3 illustrates driving force U_1, contributing to *orthogonal* information components I_1 and I_2. If U_2 is independent of U_1, then it will contribute additional information on I_1 and I_2. However, adding U_2 will also bring in the additional noise (error) associated with the U_2 data set.



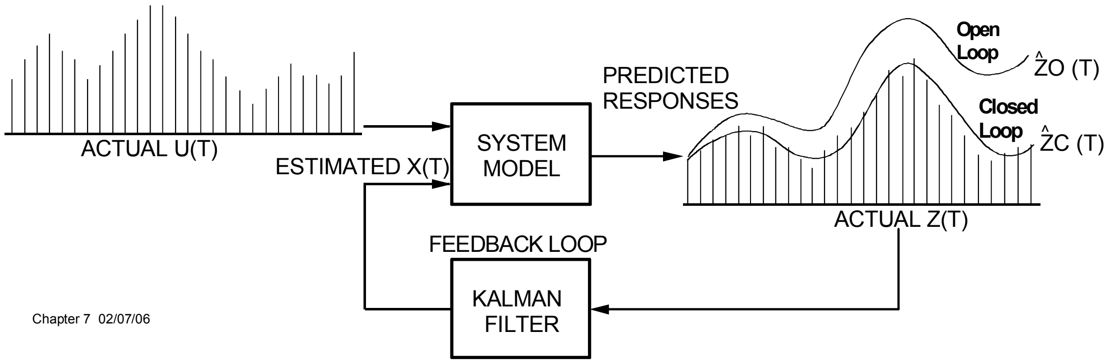
Chapter 7 02/07/06

Figure 10-3. Independence of information content relative to size of noise.

If U_2 is close to U_1 relative to independence of the information content, then the modeler must be concerned about the amount of noise (error) being introduced relative to the amount of information. Theoretically, this is a difficult problem to address, and is beyond the scope of this treatment. From a practical standpoint, which is our interest, the relative amount of information content in any candidate driving force can be tested by incorporating it into the model and determining if prediction accuracy improves. We must be aware that modeling error is likely to improve if additional coefficients are added, and a good nonlinear optimization algorithm is used to identify these coefficients. However, prediction error is the measure to be used to determine the benefit of adding a candidate driving force. Finally, the model modifications required to extract the additional information from the data set are critical to improvements in prediction accuracy, and this kind of modeling requires a thorough understanding of the underlying system mechanisms being represented.

Closed Loop Considerations

The previous illustration shown in Figure 5-1 is known as an open-loop process, i.e., one where information comparing the actual observed output and predicted response is not fed back to the model. A closed loop process is shown in Figure 10-4. Here, we can compare the predicted response $\hat{Z}$ to the observed response Z, and use this information to improve our estimate of the current state of the system. In this figure, a Kalman filter is depicted as the estimator of the current state.



Chapter 7 02/07/06

Figure 10-4. Using observation feedback to improve the estimate of the current state.

Let's now look back at our prediction of the current state of the system at time T. This was based on our dynamic model, conditioned on all information available up to and including time T-1, before the measurement of Z(T). We will refer to this as the current state prediction. The notation commonly used is

$$X^-(T) = X(T|T-1) \tag{10-10}$$

After Z(T) is observed, the current state can be re-estimated, conditioned on all information through time T, i.e,

$$X^+(T) = X(T|T) \tag{10-11}$$

Various mechanisms, e.g., the Kalman filter, [20], may be used to provide optimal updated estimates, $X^+(T)$, when added to the model as indicated in Figure 10-4.

We have purposely used X(T) to represent the current state and X(T-1) to represent the prior state, as opposed to X(T+1) being the predicted state and X(T) the current state. This is done to emphasize the purpose of the filter. Estimators or filters provide an optimal means for estimating the *current* state of the system based on all information available at the current time T. By definition, *a filter is not a predictor*. Predictions, at future time (T + τ), are accomplished by the dynamic model.

The filtering property does provide a facility to test models for consistency of short term predictions over large data sets. It can also be used as a tool to "filter" out phenomena which may obscure that which the modeler is trying to identify. In general, given the model

$$\hat{Z}(T+\tau) = C[U(T), Z(T)], \tag{10-12}$$

the filter can be used as an aid in determining the best U and C.

Suppose, for example, we wish to optimize a model that provides 10 day predictions for a commodity market. Assume we plan to use 1000 days of history to identify model parameters. If driven open loop, the model $\hat{Z}_O(T)$ may drift far from the actual response, Z, after a few hundred days, reference Figure 10-4. The short term (10 day) prediction error is difficult to detect due to the much larger long term error. By adding the filter, the long term error can be virtually eliminated (filtered out) through the current state tracking process which the filter obtains from the observation data, yielding $\hat{Z}_C(T)$.

To accomplish this, the Kalman filter is derived to maximize orthogonality between the model error vector and response vector by minimizing the expected value of their inner product

$$(10\text{-}13) \qquad \Phi \; = \; E\,[\{C[U(T), Z(T)] - Z(T+\tau)\},\; Z(T+\tau)]$$

Refer to Papoulis, [23], for a further discussion of the orthogonality properties of the filter.

Using the filter this way, one can find C and U more quickly and with less error. This helps to improve overall accuracy of the prediction model. In addition, if it can be determined that the error covariance matrices are nonstationary, then the filter can be designed to "adapt" to changes in the error statistics, see Gelb, [14], and be incorporated as part of the prediction model itself.

When using filters as indicated above, one must be concerned about the ability to separate the time constants of the various processes and the associated frequencies of change in the system being modeled. If these frequencies are too close, it can be difficult to separate them with the filter. Furthermore, when building adaptive filters, the separation of frequencies will be changing as the error properties are changing. These considerations impose practical limitations on the use of powerful tools from engineering, where one may have the fortune that systems can be *designed* to have widely separated time constants, ensuring that the filters will work well. However, this is not always the case. Below we present the basic form of the Kalman filter.

Computational Aspects Of The Kalman Filter

A diagram of the state vectors associated with the Kalman filter are shown in Figure 10-5.

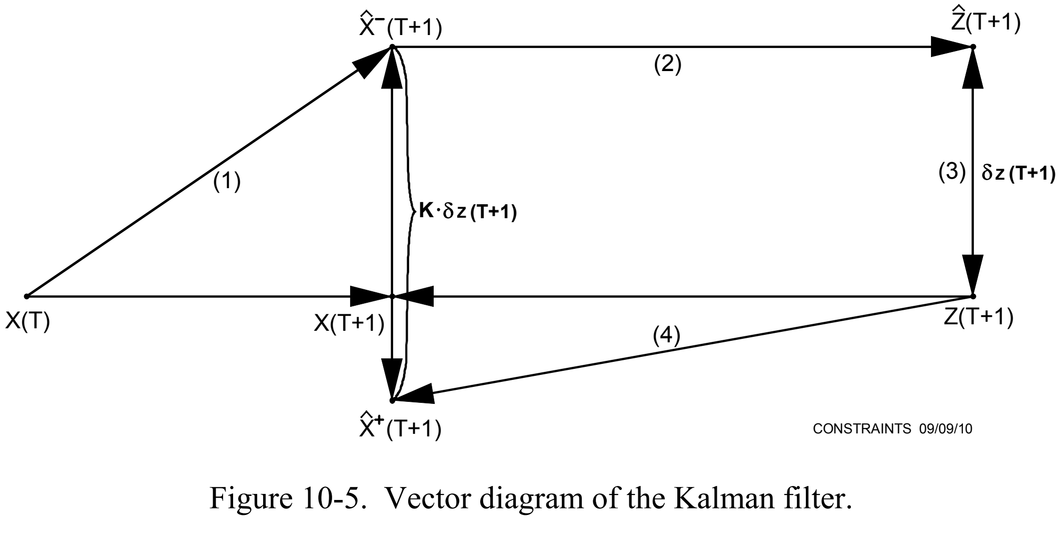


Figure 10-5. Vector diagram of the Kalman filter.

The equations associated with the basic Kalman filter will not be derived here since they are derived in a multitude of readily available sources. In this author's view, one of the clearest derivations was done by Papoulis, [23], using the orthogonality principle as the basis for maximizing the amount of additional information being used.

In the diagram in Figure 10-5 above, one starts at the state X(T) and makes a *single step prediction* using the model described in the following equation,

(10-14) $\hat{X}^{-}(T+1) = \hat{X}(T+1|T) = F(T) \cdot X(T|T)$.

In this equation, the minus sign after X is used to denote the a priori prediction, where the state at T+1 is predicted at time T. This implies that the probability statement is conditioned on information only up to time T. This can be extended to T+τ, as indicated in Chapters 5 and 6, but to minimize the complexity of the equations presented here we will stay with single step prediction.

Once time advances, and the observations are available, the *current state estimate* may be updated using the Kalman operator, K, on the observed residuals δz,

(10-15) $\hat{X}^{+}(T+1) = \hat{X}(T+1|T+1) = \hat{X}^{-}(T+1) + K(T+1) \cdot \delta_z(T+1)$,

where the observed residuals are given by:

(10-16) $\delta_z(T+1) = \hat{Z}(T+1) - Z(T+1)$.

The Kalman operator is given as follows,

(10-17) $K(T+1) = P_x^{-}(T+1) \cdot H^{T}(T+1) \cdot [P_z(T+1)]^{-1}$

where Pz represents propagation of the observation error covariance,

(10-18) $Pz(T+1) = E[\delta_z(T+1) \cdot \delta_z^{T}(T+1)]$,

and Px represents propagation of the state error covariance,

(10-19) $Px(T+1) = E[\delta_x(T+1) \cdot \delta_x^{T}(T+1)]$.

Then

(10-20) $P_z(T+1) = H(T+1) \cdot P_x^{-}(T+1) \cdot H^{T}(T+1) + R(T+1)$, where

(10-21) $P_x^{-}(T+1) = F(T) \cdot P_x^{+}(T) \cdot F^{T}(T) + Q(T)$, and

(10-22) $P_x^{+}(T+1) = [I - K(T+1) \cdot H(T+1)] \cdot P_x^{-}(T+1)$

is the state error covariance matrix predicted for the next time step.

The above equations represent the basic Kalman filter (estimator). There are many approaches that may be applied to improve the accuracy of a filter for different systems. Examples are adaptive filters, nonlinear or extended filters, etc. In addition, innovative techniques may be applied using more state variables to estimate changing parameters in models of the system and the environment. Multi-step prediction becomes a complex bookkeeping problem, requiring tracking and matching all the residuals at each time step into the future so that observations may be applied if and when they are available. Finally, a good understanding of the mechanics of the system and its environment is most important when trying to improve accuracy.

11. DEFINING THE PREDICTION PROBLEM

With the framework provided in the previous chapters, we can now proceed to define the prediction problem.

System Uncertainty

Based upon the above, we define an *uncertain process* as follows. Recall that a process, Z(T), is said to *appear random* when no transformation C can be found for which

$$(11\text{-}1) \qquad E\,[C[Z(T)], Z(T+\tau)] \;\geq\; \varepsilon_\tau, \quad \text{for any } \tau > 0.$$

For nonhomogeneous systems, we say that Z(T) is an *uncertain process* relative to driving force U(T) when no transformation C can be found such that, for any $\tau > 0$,

$$E\,[C[U(T), Z(T)],\; Z(T+\tau)] \;\geq\; \varepsilon_\tau$$

System Predictability

We say Z(T) is a *predictable process* of order τ when a driving force U and transformation C can be found such that for $\tau > 0$,

$$(11\text{-}2) \qquad E\,[C[U(T), Z(T)],\; Z(T+\tau)] \;\geq\; \varepsilon_\tau$$

We note that a process which appears random by standard statistical tests can be predictable since, based on our example of Figure 6-2, Z(T) can be a delayed function of a purely random process U(T). This represents a generalization of the *Markoff Process,* being conditioned on (nonhomogeneous) driving forces, observed τ states back.

Referring again to Figure 5-4b, we see that the process shown is predictable of order four, and that no error is incurred. Should we attempt predictions five steps into the future with this model we incur an error, since an impulse at the next (future) time step will effect the response five steps from T_0. This is a *prediction* error due to the inherent order of predictability of the system. This must be distinguished from the model or observation error (described in Chapter 7 under Stochastic Considerations) which is generally treated in control theory literature. We are assuming, of course, that the driving force has unpredictable components. When we construct state equations containing error terms, we must incorporate an *additional error term* beyond those reflecting uncertainty in the model and in the data.

Modeling or Estimation Error

The following measure is offered to optimize the choice of U and corresponding transformation C. We want to find C(T) and U(T) such that

$$\text{(11-3)} \qquad \Phi(C, U) \;=\; D\,[C[U(T), Z(T)],\; Z(T+\tau)]$$

is minimized, where D is some measure of distance (e.g., mean absolute deviation) between the predicted response based on the model,

$$\text{(11-4)} \qquad \hat{Z}(T+\tau) \;=\; C[U(T), Z(T)] \;=\; \hat{Z}(T+\tau|T)$$

and the actual response Z(T+τ). For example, C and U can be selected to minimize the mean absolute error function

$$\text{(11-5)} \qquad \hat{e}^{-}\,(C, U, Z) \;=\; \hat{e}\;[\hat{Z}\;(T+\tau|T),\; Z(T+\tau)]$$

$$= \;\; \mathbf{E}\left[\;\left|\frac{C[U(T),\;Z(T)]\;-\;Z(T+\tau)}{Z(T+\tau)}\right|\;\right]$$

A similar measure would be to minimize the mean square error. We note that the selection of U and C depend, in general, on τ. In practice, one can select the value of τ most critical to the application. Or, some functional combination of $\hat{e}^{-}$ at various values of τ can be used.

However, once we use (11-5) as a performance measure in an optimization process, then *information at* T+τ *has been incorporated into the model*. Therefore,

$$\text{(11-6)} \qquad Z(T+\tau) \;=\; C[U(T), Z(T), Z(T+\tau)] \;=\; Z(T+\tau|T+\tau),$$

is not a true prediction - *it is an estimation* - and any future error measure will be of the form $\hat{e}^{+}$(C, U, Z).

Correlating Prediction Error to Modeling or Estimation Error

The measure ê used for modeling error can also be used for prediction error. What is important is that the data sets are different. All data up to the current time T can be used to optimize C and U so as to minimize $\hat{e}^{+}$(C, U, Z), providing an optimal estimate. Future data beyond the current time must be used to measure prediction error. If reductions in modeling error do not correlate to reductions in prediction error, then the modeler has no consistent method for improving model accuracy in a way that reduces prediction error.

To summarize, if the same error function, e.g., ê in (11-5) above, is used to measure both modeling error and prediction error, the difference in the measures is essentially the use of previously available data versus the use of unseen "future" data.

12. MODELING APPROACHES TO SUPPORT PREDICTION

Stationary Versus Non-Stationary Systems

One of the most important concepts to be understood when building prediction models is the difference between stationary and non-stationary systems. As described in Chapters 5 and 8, stationary systems may be represented by a curve-fit. This assumes repetitive behavior, something not found in nonlinear nonhomogeneous systems. This implies the need for models that transform nonstationary observable driving forces, using delays and time constants that are inherent in systems with inertia, into multi-step predictions. It also implies the need to model the internal nonlinear elements of a system. This is particularly true when modeling decision systems that depend on turning points based on parameter values. This is virtually intractable using naive mathematical approaches.

Practical problems require model designs based upon detailed knowledge of the physical properties of a system. This requires capturing knowledge about how a system works internally, requiring human judgment and detailed expertise. A significant ingredient of this type of modeling effort is understanding the nonlinear elements of the system, and representing these elements using rule-based decision models.

Much of the theoretical discussions presented above have represented general models of dynamic systems using systems of equations. Typically, there are better ways to obtain accurate transformations of driving forces, U, into the next system internal state, X, and then into prediction of observables, Z. For example, one can build intelligent models using rule-based algorithms that are difficult to achieve with pure mathematics. In general, a dynamic system can be represented by a large model, composed of many submodels, all working together to produce the desired transformations. This approach does not need a constant discrete time-base, but can advance in time based upon discrete events, see for example [10], and [13]. An illustration of an interconnected set of models is shown in Figure 12-1. Each model can run independently within a time specified frame, sharing data with the others, including feedback paths where appropriate.

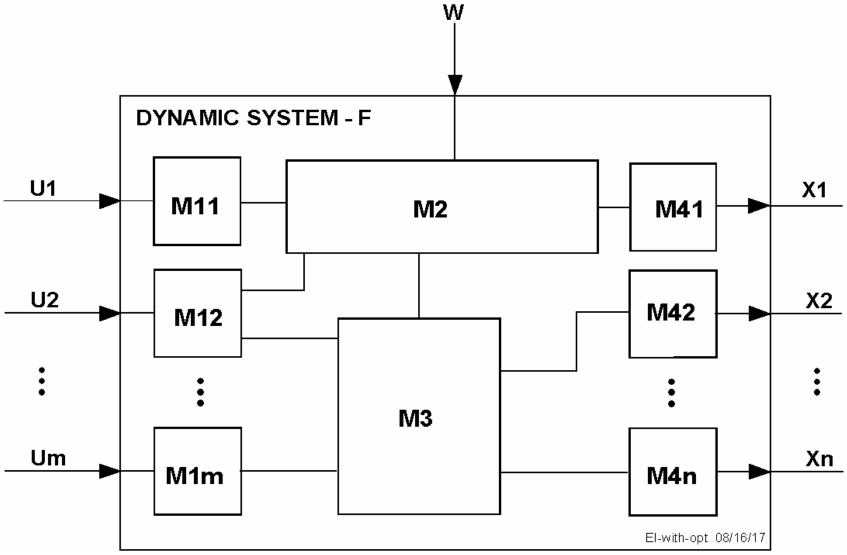


Figure 12-1. Illustration of a dynamic model.

Artificial Intelligence (AI) Models

It is important to consider and compare the use of Artificial Intelligence (AI) approaches, especially since the definition of AI has changed in recent years. Figure 12-2 illustrates a typical representation of the interconnection of synapses in the brain. Very simply, synapses contain memory elements that are interconnected in a manner that allows conduction of pulses to cause transmission reactions. As inputs impinge on animal senses, they are processed by the brain and transmitted to other body parts to produce desired physical actions. These organs and processes have evolved over millions of years within animals that have survived. This process may be loosely compared to the logical processing in a digital computer.

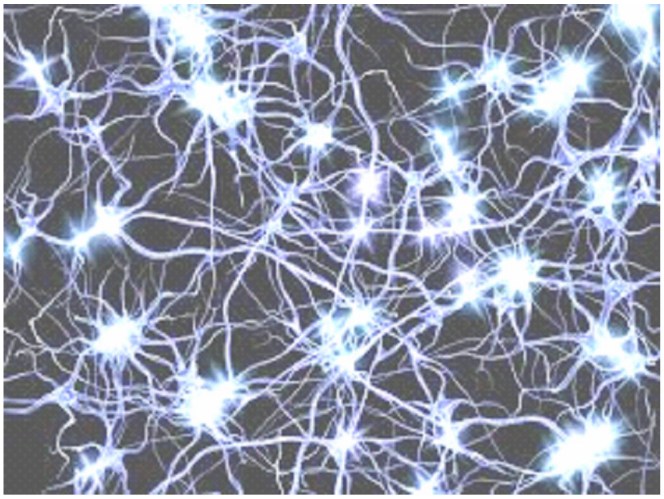


Figure 12-2. Typical representation of the interconnection of synapses in the brain.

In the early days, AI was defined as computer decision approaches that followed that of the brain. This involved the development of Neural Nets, using a combination of computer hardware and software approaches where the decision processes followed those similar to brain synapses. Many of the applications were described as falling into the general category of pattern recognition. Figure 12-3 illustrates a greatly simplified example of a computerized AI approach using neural nets to operate on driving forces, U, to produce a desired outcome, Z. If patterns in the input, U, are recognized over some observation period, then signals are produced indicating what patterns occurred.

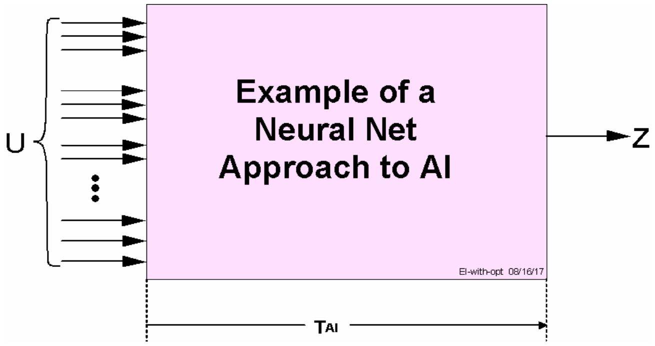


Figure 12-3. Simplified illustration of an Neural Net approach to AI.

Artificial Intelligence (AI) - Training Versus Programming Periods

As indicated above, neural nets can be "trained" to recognize patterns. This has been shown to be a way to identify objects, moving in 3-D, by their shapes. It can work even when object images are fuzzy. However, similar to human learning processes, the training period is typically very long compared to most computer computational algorithms. This is different from the typical use of computers to support computation and decision processes.

Computers have been used to augment human decision processes since their beginnings when they were used to solve difficult computational problems. In the very early days, before digital computers, people using mechanical calculators were known as "computers". To solve large problems, e.g., solving ballistic equations, rooms were filled with people using calculators. Businesses were soon built around electro-mechanical calculators that were programmed by wiring boards. These programmers were effectively doing logical design, accounting for delays and race conditions of electronic signals that invoked the decision processes. The first all-electronic computer (The ENIAC) was very fast compared to the electro-mechanical devices.

Before the ENIAC was completed, its designers, John Mauchly and J. Presper Eckert, improved the design by bringing instructions into the same memory as data. Concurrently, John Von Neumann, from the Institute for Advanced Studies (IAS) at Princeton University, developed the set of instructions for the first stored-program digital computer - the MANIAC - at Princeton, NJ. This computer was extremely fast compared to the ENIAC, allowing the solution of problems that heretofore could not be solved. Today's computers can solve problems in seconds that humans could not solve in years. Depending on the approach used to produce the programs, systems can be developed in relatively short periods of time compared to alternatives.

When comparing typical computer approaches to AI, except for special applications, one faces the time to build and test a program versus the learning period of older AI approaches. For example, cracking codes is a pattern recognition problem, and AI approaches have been applied for years. However, more recent encryption techniques render these approaches hard to apply, with the effort required to break codes being limited economically to extremely deep (typically government) pockets using huge parallel processing facilities.

The cases of interest in AI are recognition of nonstationary patterns in large data sets, U, that occur in advance of patterns in Z. These could be predictive, but require a nonlinear nonhomogeneous model. AI type approaches have been applied to the stock market where daily data has been recorded for publicly traded stocks for over 70 years. An enormous database of history exists for training neural nets, and one would expect this to be a powerful approach. However, its success appears limited to estimation as opposed to prediction, e.g., simply determining if a specific measure of buy/sell orders is above or below a threshold. This has worked extremely well in the past, but as more organizations use this technique, the time constants to making decisions have gone down to seconds, making speed of solution a major factor. We note again that this is an estimation approach.

A prediction approach would require predicting changes in market value of particular stocks or commodities over a number of days into the future, where prediction accuracy would be the determining factor in success. This is a significant application for prediction. As a result, as clearly specified at The 2016 Economist Conference in Chicago, the term AI was redefined to mean "Prediction," implying the development of algorithmic approaches to solving problems.

Expert Intelligence (EI) Models

There are two basic types of data used in prediction models. In one type, the input driving forces and resulting responses are generally large. This is the data typically used for "learning" in an AI model using the old definition. It is also used to identify optimal values of parameters in prediction models, as well as to test prediction accuracy. The other type is based upon interconnected models derived from knowledge of the internal workings of a system. For example, the knowledge required to build models of financial markets is generally provided by subject area experts who have spent many years learning the operational mechanics of a particular market. This knowledge is used to develop complex mathematical models that effectively condition the probability statements that quantify predictions. A simplified depiction of a set of interconnected models is illustrated in Figure 12-4. The difference between this EI approach and an AI approach is that the models typically contain complex algorithms derived by their designers based on reasonably detailed knowledge of the decision processes, actions and reactions that occur within that application.

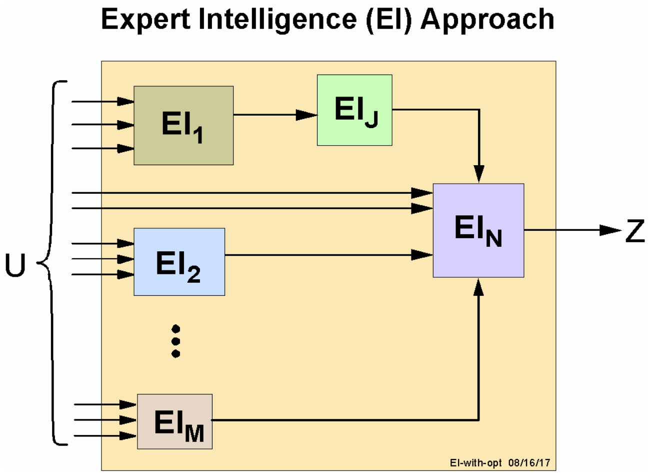


Figure 12-4. Models derived from knowledge of subject area experts.

Experience has shown that, using small amounts of data with no repeating patterns, one can build models that provide sufficiently accurate predictions, see [8] and [9]. These are developed using expert knowledge of the mechanics of how driving forces affect the system. Small amounts of data are sufficient to characterize a small number of model parameters, just as is done in the field of physics. In these cases, the systems can be highly nonlinear. Using nonlinear models, one can provide much more accurate predictions.

As described in prior chapters and roughly illustrated in Figure 12-1, nonhomogeneous nonlinear models can be built and interconnected to predict the behavior of complex systems. PSI has used the discrete event simulation environment in VisiSoft, based upon a "generalized" state space framework to build such models. It provides for vector spaces containing discrete states that can be described by words as well as numbers. Transformations can be generic rules, not restricted to mathematical operators. This permits a rule oriented format, e.g.,

IF *this* …., THEN *do that* …., ELSE *do something else* …. , a format for decision rules.

This format supports direct translation of subject area expert knowledge into model process rules. Since this language is translated directly to machine code, models are built and maintained directly in this environment. Subject area experts can read and write the rules written in this language without knowledge of computer programming. As indicated in earlier chapters, the use of subject area experts is considered a major factor in building accurate prediction models.

EI Models With Optimization

Another approach to modeling system behavior uses optimization to identify parameters in an EI model. This is illustrated in Figure 12-5. The optimization process can be applied off-line, or adaptively in real time. This is the approach found to be most useful when modeling decision processes for which expert knowledge can be introduced.

Using VisiSoft, expert human knowledge is incorporated into the EI models as shown in Figure 12-5. Unknown parameters are used to account for lack of knowledge. However, these parameters are selected judiciously. Their placement in the model is determined based upon where information is lacking. Often, one has reasonable knowledge of ranges on these parameter values. Any piece of additional information that can be used in a model to cut down on the size of the unknown space, leads to a faster - as well as more accurate - solution.

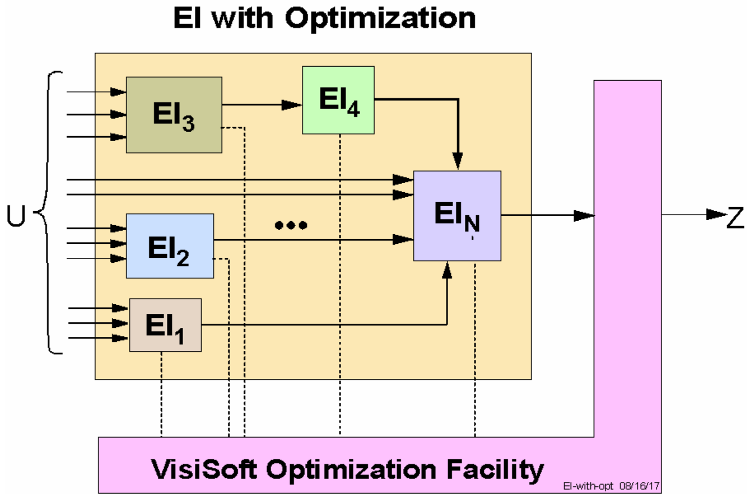


Figure 12-5. Illustration of an optimized EI approach.

First hand experience on many projects clearly demonstrates that experts are aided significantly when observing the models as they behave in a simulated environment. This generally leads to significant improvements in representation of the decision process.

The VisiSoft optimization system referenced here is built into the simulation environment. It uses adaptive algorithms that automatically formulate ensembles of data to generate the distributions used to update the search process. This system has been used to solve a wide variety of highly nonlinear problems, such as finding the best location for antennas in a very mountainous environment under threat jamming, or finding optimal flight paths for ELINT or SIGINT collections, accounting for threat air defense systems.

Parametric And Sensitivity Analysis To Support The Modeling Process

Another means of identifying EI model parameters is by running simulations to support parametric analysis. For example, one can generate distributions of responses by running a sufficient number of simulations while varying parameters to determine if model results fall inside sensible ranges.

EI Models With Adaptive Estimators

PSI has built EI models with embedded adaptive estimators. These adaptive estimators are used to improve the estimate of the state vector, or specified subsets of the state vector, as observations become available. When an observation comes in, an improved estimate of the current state vector is determined, and new predictions are produced.

Accurate estimates can be achieved using different forms of Kalman or similar filters, including nonlinear and adaptive filters. When adaptive filters are used, parameters in the filter are estimated along with the state vector. Adaptive filter parameters are typically estimated on a longer term basis, so that the time-constants of variations between the state estimates and the filter parameter estimates are sufficiently separated. This is illustrated in Figure 12-6 which shows an optimized adaptive EI approach.

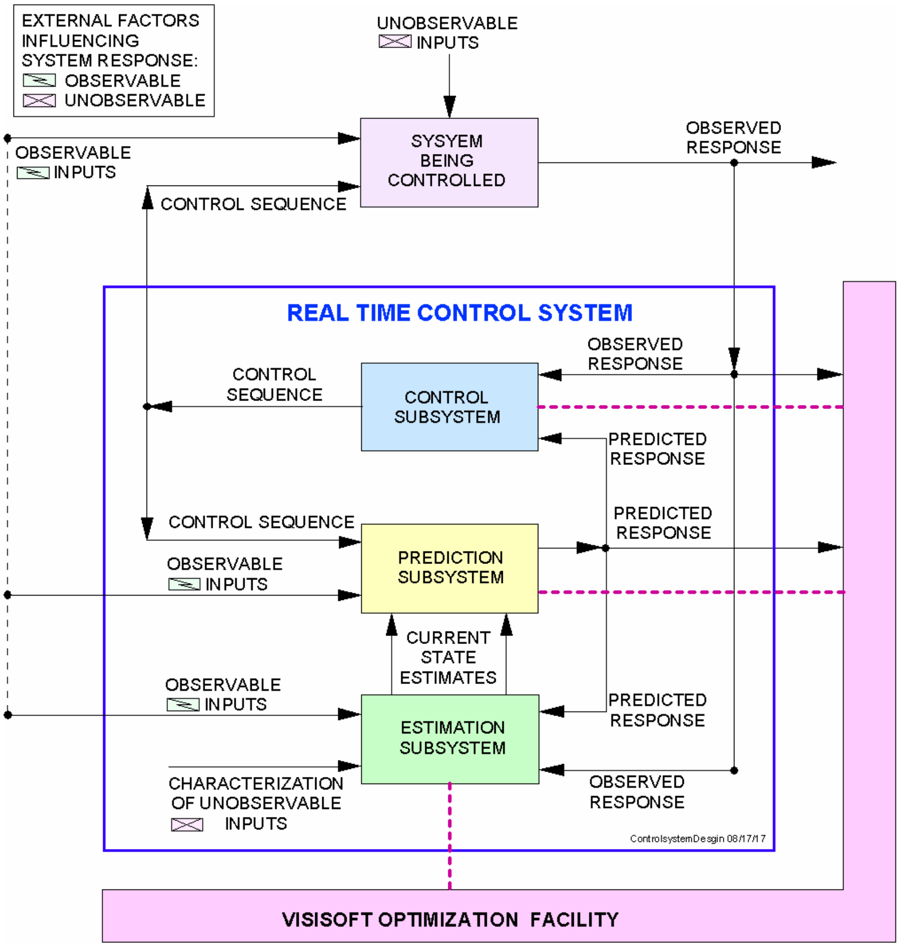


Figure 12-6. Illustration of an optimized - adaptive EI approach.

EI Models With Interactive Graphical Visualization

The use of graphical interfaces are also considered a major factor in understanding what is occurring dynamically as models generate results. Visualization of dynamic behavior is one of the best ways to understand system and model behavior, and to use that knowledge to improve the models. Figure 12-7 illustrates multiple subject area experts observing system and model behavior as the dynamics unfold. Equally important is their ability to make changes interactively to improve predictions.

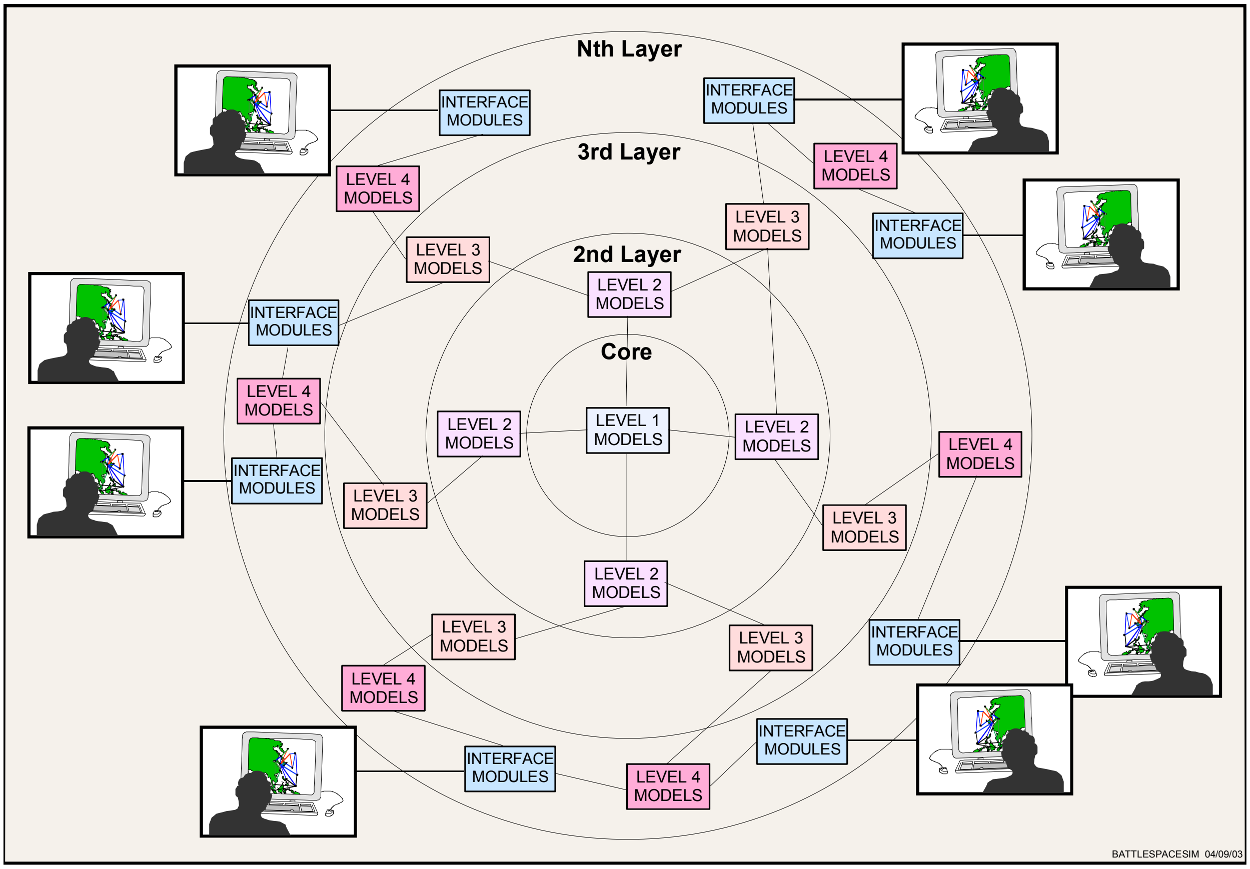


Figure 12-7. Subject area experts interacting with prediction models.

13. PREDICTION AND CONTROL - A MILITARY EXAMPLE

Problem Overview

Military operations require highly accurate termination of a weapon's path at the target to avoid collateral damage as well as complete a successful mission. This requires an accurate control system to guide the weapon to the target. In the relatively simple example used here, the target is at a fixed location. However, the solution approach described can be expanded to the case where the target is moving. This problem is typically initiated by a moving platform, e.g., an Unmanned Aerial Vehicle (UAV), performing a hand-off of control to the onboard weapon control system, whereby the initial state of the weapon is passed by the initiating platform. Current approaches to solving this problem depend upon accurate estimates of the weapon state (position, velocity, etc.) at hand-off. This is because the on-board weapon control system may not be turned on until launch, and accurate estimates of the position of the weapon itself may take time without an accurate initial estimate from the launch platform. This problem is typically solved by having the control systems on both platforms obtain position data using GPS receivers to update estimates of measurements from an Inertial Measurement Unit (IMU).

Having developed GPS coverage mapping tools using accurate models of the GPS constellation, including receiver connectivity in very rough terrain (e.g., Afghanistan), PSI has shown that satellite coverage may fall below accuracy requirements at certain times of day, even without jamming, see [32]. In situations where coverage is degraded, it is difficult to accurately guide weapons to a target when the control systems on both platforms depend upon accurate GPS signals. Various approaches to mitigating this problem have been proposed and developed over the past decade. These include use of distributed relative navigation systems that depend upon improved versions of sophisticated radio networks. However, when dealing with small munitions, restrictions on size, weight, and power of on board equipment limit approaches to design of the onboard control system. The best solutions use more computational power and memory, see [33].

A pertinent example scenario may take place in the mountainous region of Northeast Afghanistan where connectivity between platforms is difficult for communications, see Figure 13-1. This environment places stress on connectivity with nearby platforms as well as satellites. In such an environment, the launch platform may have difficulty determining its own position with sufficient accuracy before launching a weapon.

Depending upon the host platform and weapon, power to the weapon may be limited until a certain point relative to the launch process. For example, full power may not be available until 30 sec after launch. Assuming compatibility of radios used on the munition with those in range, there may be a synchronization/identification delay before reliable reception of position messages can occur. Depending upon the distance to the target and the speed of the munition, these delays may be a significant factor affecting the ability to receive enough observations, refer to Table 13-1.

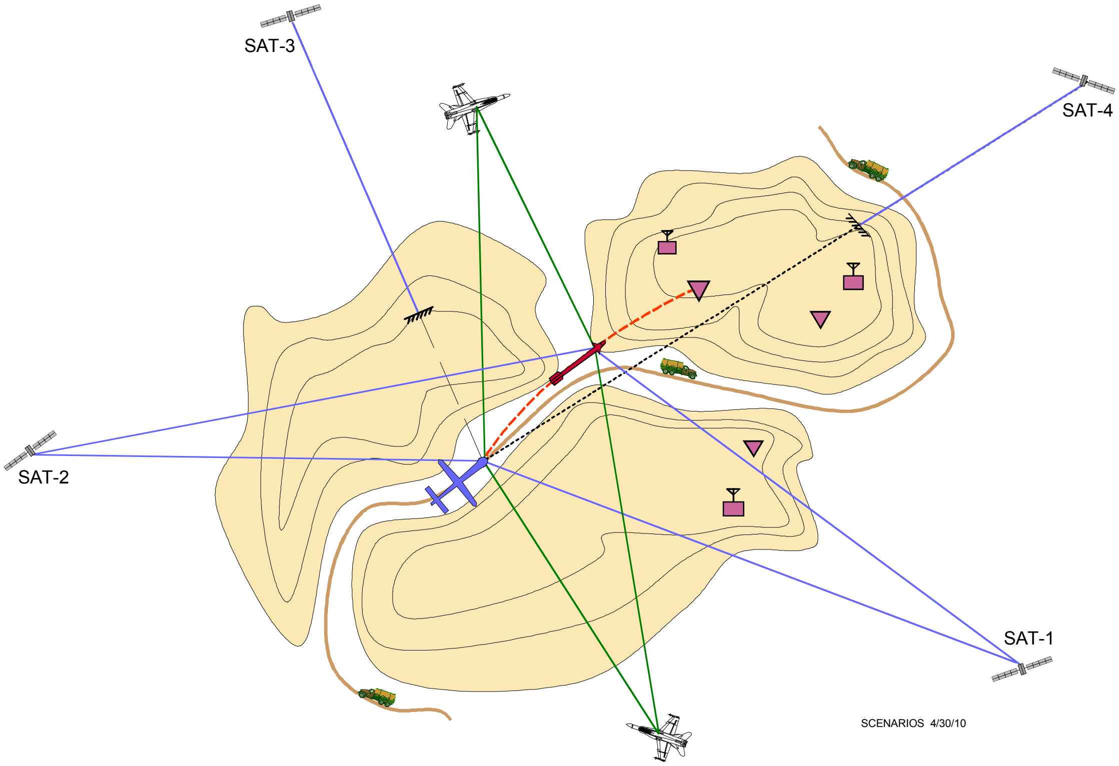


Figure 13-1. Example of a scenario of interest.

Table 13-1. Time of flight as a function of distance and speed.

DISTANCE TRAVELED (Km)	TIME (Seconds)			
	SPEED			
	25(m/sec)	50(m/sec)	100(m/sec)	200(m/sec)
5	200	100	50	25
10	400	200	100	50
25	1000	500	250	125
50	2000	1000	500	250

Error Analysis 09/13/09

Figure 13-2 is a simplified example illustrating a weapon launched at T0 from a platform that may or may not have sufficient GPS coverage to obtain an accurate position of itself. If it does, and gets that information to the weapon, wind forces may work to drive the weapon trajectory off course. Then sufficiently accurate observable inputs are still needed to guide the weapon to the target. Many systems depend upon GPS inputs to provide an accurate update to an IMU that is feeding the control system maintaining the weapon on a desired trajectory. However, if sufficient numbers of timely messages are not received from the GPS constellation, accuracy may be lost and the weapon may be thrown off course by wind or other atmospheric effects. If this occurs multiple times along the path, getting it back on track may not be feasible.

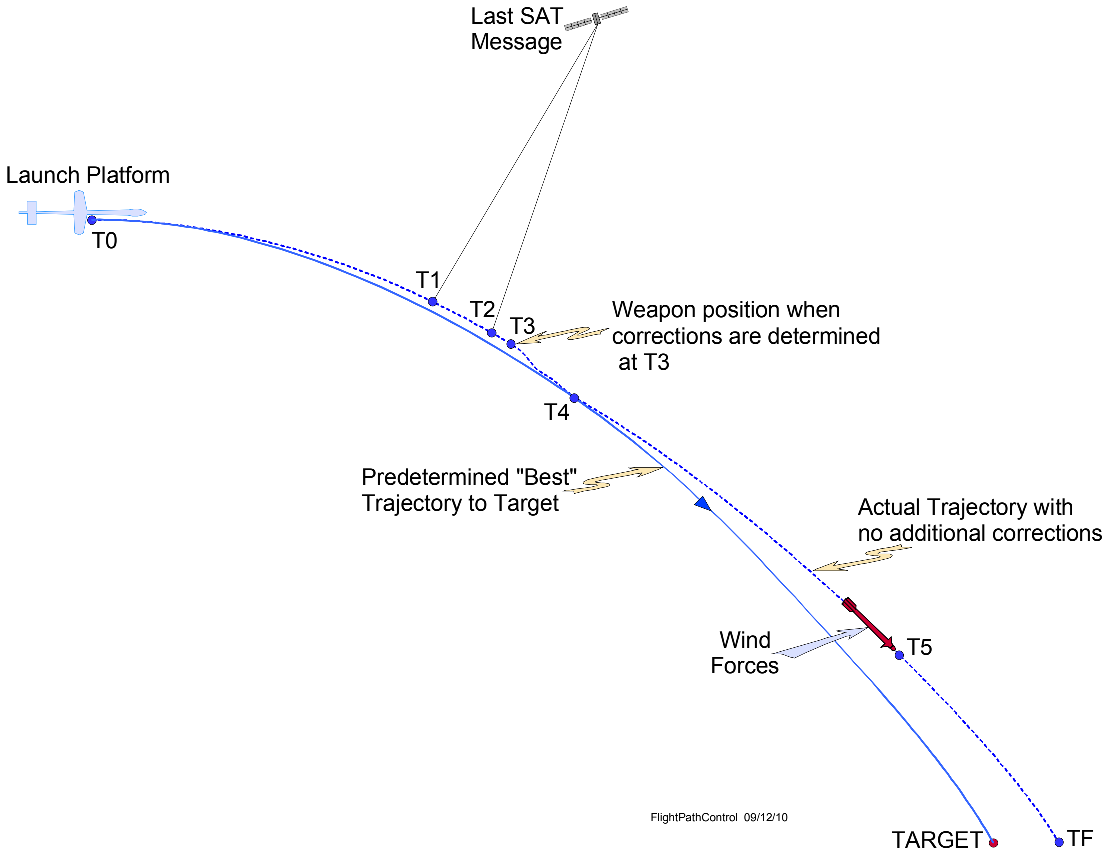

Last SAT Message
Launch Platform
T0
T1
T2
T3
Weapon position when corrections are determined at T3
T4
Predetermined "Best" Trajectory to Target
Actual Trajectory with no additional corrections
Wind Forces
T5
FlightPathControl 09/12/10
TARGET
TF

Figure 13-2. Simplified illustration of weapon trajectories.

Assuming the weapon has an IMU on board, then depending upon the initial conditions of that system, it may take time to update it with accurate position information. If it gets accurate GPS fixes up to a point in the flight, then the time constants of accuracy decay may be critical depending upon the remaining flight time and additional observable measurements. As indicated above, a solution is to get position information from sources other than GPS as it moves along the trajectory. This implies sufficient accuracy and time to receive this information, and this depends upon the on board radios being used to receive the messages. For example, in more advanced radio systems, multiple relayed messages may be sent at the same time to ensure reception, but that causes the potential of mutual interference.

The position, orientation, antenna gain and polarization of transmitting platforms, along with the terrain, atmospheric conditions, etc., between platforms, and the position, orientation, antenna gain and polarization, and noise environment at the receiver, and the signal processing gain are some of the factors that determine the communications connectivity between platforms. Connectivity between platforms determines the number of platforms that can be used to provide position updates to the weapon. The waveform used to transmit position messages, and the number of messages that must be received from different platforms will determine the time to obtain a sufficiently accurate position update, see [34] and [35].

Characterizing the speed and accuracy of solutions to this obviously complex problem is a key part of the design process. Fair comparison of different approaches is critical to making decisions on technology investments. Sufficiently detailed live testing is hard to control and becomes very expensive when analyzing variations to ensure decisions are based upon accurate information. In this type of situation, detailed simulations have been used successfully to account for all of the sources of error and variations. These have provided the best approach to support both design and comparison of solutions. Models may be validated using limited test data. Trade-offs on the design of measurement and communications equipment, (IMUs, radio receivers, sensors, etc.) to achieve accuracy within limited size, weight, and cost bounds are highly complex. The critical part of the simulation problem is time and cost to build and run the simulations.

Approach To Maximizing Accuracy Of The Control System

The technical approach to designing a sufficiently accurate control system of this nature requires software approaches that mitigate the problems defined above while fitting into on-board computers. The intent is to provide software solutions to take maximum advantage of available hardware to gain the required accuracy as well as meet the constraints on size, weight, and cost.

When looking at existing solutions to this problem, they typically take the form depicted in Figure 1-2, but without an explicit prediction system. This is because control theory is generally based upon the exclusive use of estimation. Prediction models are not used, a serious flaw when trying to achieve an accurate guidance system. The purpose of this book is to use a prediction approach to maximize the accuracy of control systems.

As described above, the guidance problem is to maximize the probability of hitting the target. If the probability statement itself is inaccurate, then the probability of hitting the target is decreased. To maximize the accuracy of the probability statement, one must maximize the information used to condition the probability statement. The difference between estimation and prediction is simply the ability to add more information to the conditioning of the probability statement. As stated in prior chapters, *additional information* implies that which is above any additional noise and is orthogonal to what is already there. This is accomplished using prediction models that account for future driving forces, delays, and time constants that cannot be accounted for using estimation.

Control System Architectures

As described in the first chapter, control system architectures may be represented as in Figure 1-2, where the control sequence is input to the system being controlled. Such systems generally rely upon observable data and knowledge of flight mechanics to obtain accuracy. As observations become available, control sequence outputs are updated. Prediction accuracy is represented by a probability statement conditioned on all information up to current time T. This information is usually obtained from observable data and knowledge of the system mechanics.

However, additional information may be obtained from knowledge of the unobservable as well as observable factors affecting the future (T > 0). We propose to include a prediction subsystem to predict these factors to gain accuracy. We will use wind as an example.

The framework in Figure 1-2 shows an overall control system composed of generalized sets of components for control, estimation and prediction. Note that control subsystems may in turn contain their own estimation and prediction subsystems. For example, an inertial navigation component, e.g., an IMU, may itself contain a control subsystem. The purpose of this approach is to support optimized architectures of control subsystems which may not share observable inputs and state vector components with other control subsystems. This approach supports design optimization of both hardware and software components on a relatively independent basis, where error state subvectors may be separated for improved estimation as well as architecture.

A critical part of the prediction subsystem is the model of the navigation unit's response to different atmospheric conditions, e.g., wind. This model must account for dynamic changes in forces hitting the munition as it continues along its path. This requires the ability to differentiate between changes in state due to normal movement in an unchanging environment versus those due to changes in external forces. Other sources of atmospheric measures may also be available.

As illustrated in Figure 13-2 above, factors, e.g., changing wind forces, directly affect the path of the weapon. Given a navigation system component, e.g., an IMU, the atmospheric environment will affect both the state estimates from that navigation unit as well as the actual weapon path. We propose to develop models of the environment that use changes in state estimates from the navigation unit to predict atmospheric behavior down the flight path, see Figure 13-4. This will produce more accurate predictions of where the munition is headed in the future. The improved accuracy comes from the additional information contained in models of future atmospheric effects, see [32].

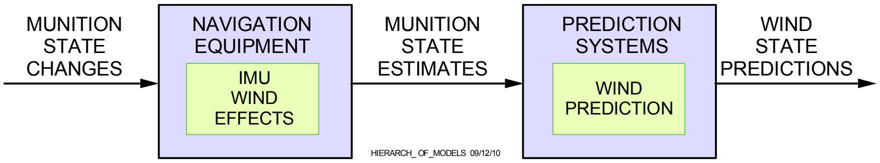


Figure 13-4. Illustration of a prediction model to characterize the effect of wind on navigation.

The wind model may be designed with coefficients that are added to the state vector for adaptive estimation. This implies that data collected while in flight may be used to improve the prediction of winds further down the path.

One may also produce control sequences that account for the predicted future changes. Instead of making corrections to get back to a trajectory that is predetermined or computed on the fly, one can use the predictions to produce control sequences that are optimized for multiple steps into the future, see Figure 13-5. For example, if predictions of future wind forces add accuracy, then controls that are produced to account for the wind changes will improve the actual trajectory at future states. This will reduce the difference between the actual and desired state and the corresponding control changes required to follow a more accurate trajectory.

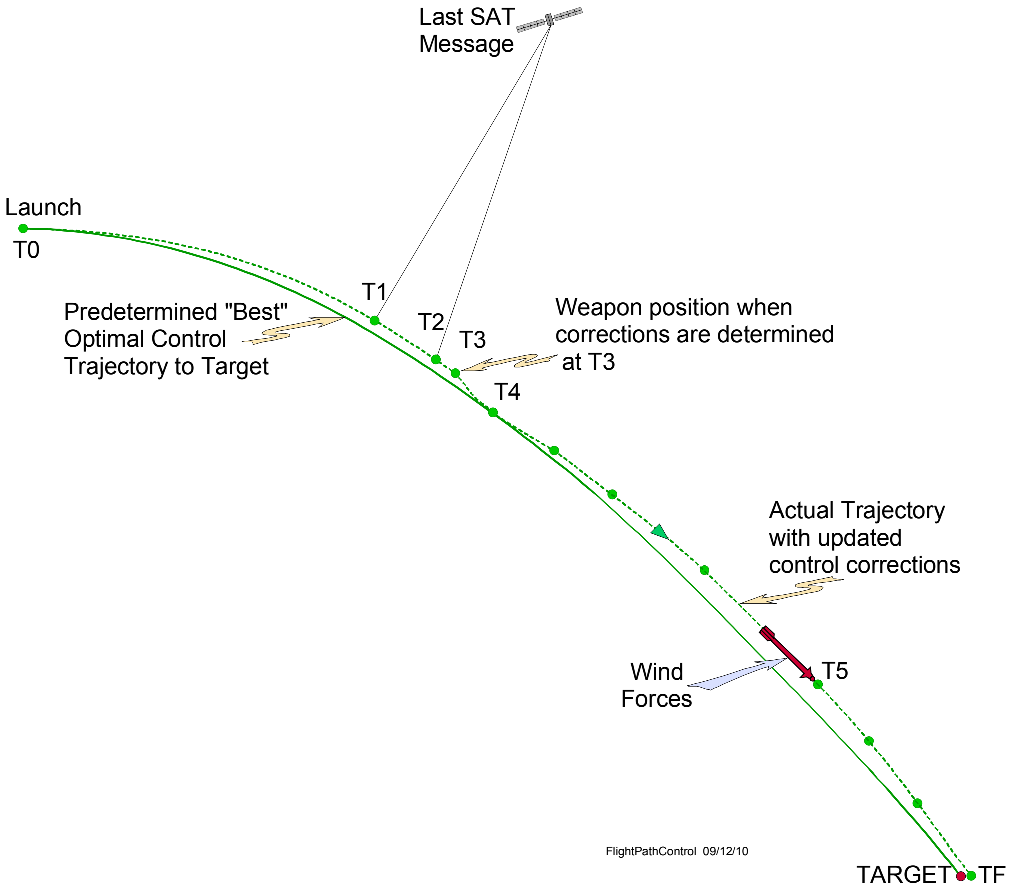


Figure 13-5. Obtaining more accurate weapon trajectories.

If we can translate the differences in position, velocity, acceleration and attitude states produced by a navigation unit (e.g., an IMU) as the munition goes through changing atmospheric conditions, then we can incorporate these characteristics into a model to predict the future state of atmospheric forces based upon the changing IMU state estimates. This depends upon a sufficiently accurate model of the IMU responses to munition platform state changes.

To do this, we must first derive a model of the IMU's response to physical changes in state. If such a model does not already exist, then this effort will require actual IMU test data. Given a model of the IMU responses, we can create a simulation using wind scenarios to characterize the state estimates coming from a navigation unit based upon simulated munition state changes. From this we can produce an IMU Wind Effects model.

Given estimates of the changes in wind, we can derive the characteristics of a wind prediction model. Creating and testing these models implies an architecture illustrated in Figure 13-6. We note that the wind effects model may be incorporated into a final wind prediction subsystem with its own estimation subsystem. This is an architectural design issue that is best resolved from a model simplification standpoint.

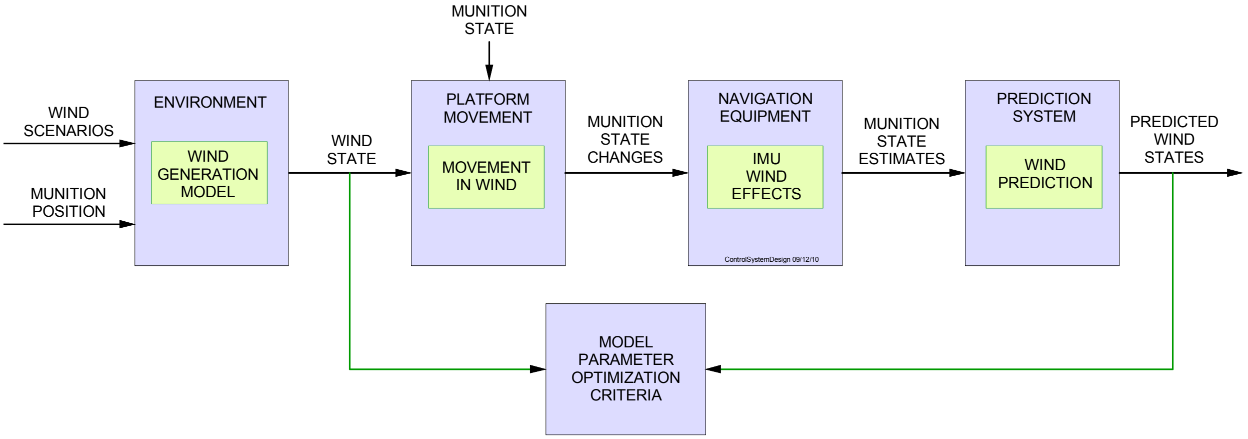


Figure 13-6. Identification of the prediction model.

The above approach is not limited to an IMU but may incorporate other navigation units. Likewise, it is not limited to wind, but can incorporate other atmospheric effects. Separate prediction models may be built, each with their own state estimators to provide adaptive updates to model coefficients. The intent of this approach is to expand the use of software and computer memory to maximize accuracy of the trajectory to meet requirements using a minimum equipment suite best suited to the weapon.

Another benefit of using multi-step prediction is shown in Figure 13-7. A byproduct of multi-step prediction is the ability to create an envelope model characterizing future error, e.g., in the position state. This is done by propagating the covariance of the position vector a selected number of time steps into the future, and finally to the terminal point on the trajectory, as illustrated in Figure 7. This information is used to update coefficients in the envelope model.

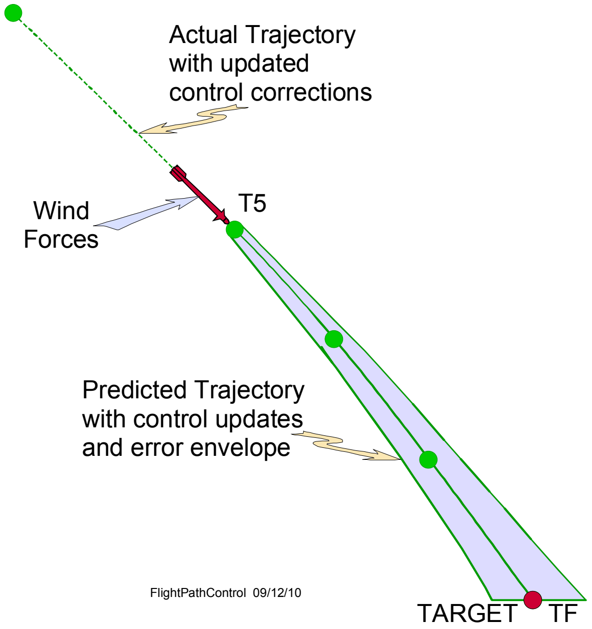

Actual Trajectory with updated control corrections
Wind Forces
T5
Predicted Trajectory with control updates and error envelope
FlightPathControl 09/12/10
TARGET
TF

Figure 13-7. Illustration of a prediction envelope near the terminal point on the trajectory.

The percent error that the envelope represents can be predetermined. This information may be used to terminate the weapon when the predicted error exceeds a predefined limit. The envelope model may be developed using a nonlinear optimization system in a simulation. The next section provides background on how this may be accomplished.

Computer-Aided Design (CAD) / Optimization

One can optimize control systems by embedding them in a VisiSoft simulation. This helps to provide a fair comparison of different approaches. The simulation must contain models of all factors affecting the flight of the weapon to produce an accurate assessment of its ability to hit the target. This implies that models have been validated using live test data, and that the simulation environment contains a subsystem that allows designers to optimize parameters in the control, estimation and prediction subsystems of the overall control system shown in Figure 1-2 above.

The VisiSoft CAD architecture shown in Figure 13-8 has been designed so that subsystems may be optimized separately as well as jointly. The optimization facility must support the design of nonlinear dynamic systems, nonlinear optimization functions, and realistic worst case nonlinear design constraints. Typically, one wants to compare different designs to determine which one meets the constraints while minimizing an error or cost function. Cost in this case may imply adding another sensor or radio to increase the probability of having a sufficient number of accurate observable inputs at points along the trajectory.

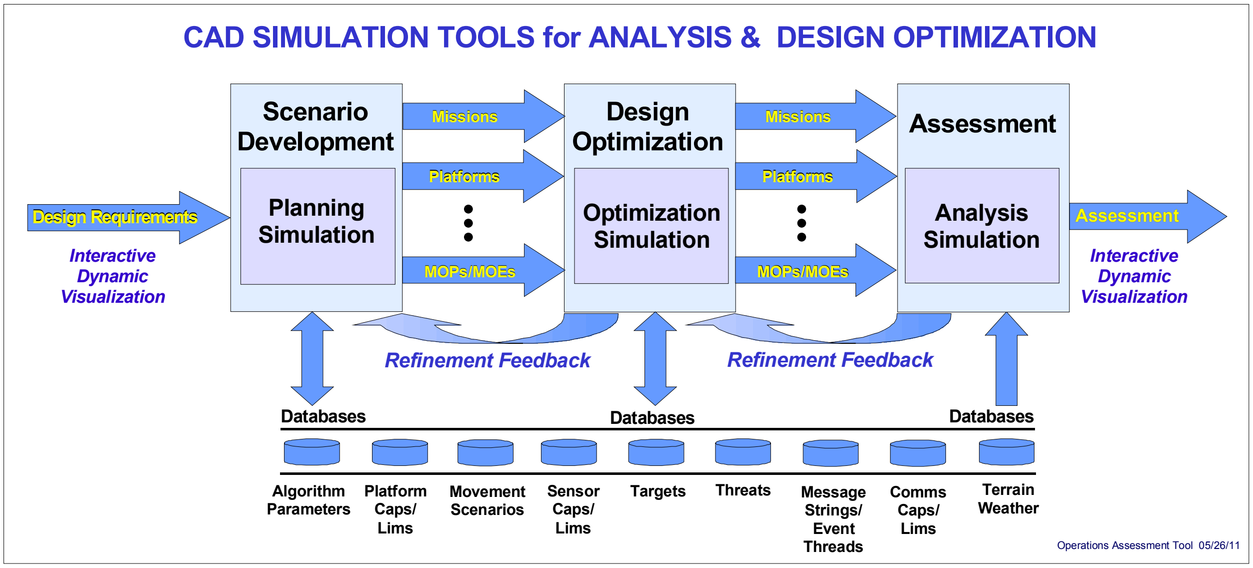


Figure 13-8. VisiSoft Simulation capability for analysis & optimization.

Figure 13-8 shows three simulations, one for scenario development (the Planning Simulation), one for design optimization (the Optimization Simulation) and one for assessment (the Analysis Simulation). The planning simulation is used to support interactive creation and modification of scenarios. The optimization simulation is used to determine optimal design parameters, e.g., those used for optimal sensor placement. The analysis simulation is used to perform general analysis, producing various MOEs and MOPs using large dynamic scenarios, and to perform parametric, sensitivity, and Monte Carlo analyses. Some of the features of this simulation capability are described below relative to functions to be performed when supporting an analysis effort.

Controlling A Large Number Of Large Complex Models

A large number of models and submodels are required to simulate complex scenarios of the control system *environment* as well as the *control system* itself. Many of the models are complex by their nature. Use of the VisiSoft CAD environment is required for combining, changing and controlling these models quickly for rapid prototyping and testing. Modeling along physical lines using makes the resulting architecture of models independent so it can support changes. Many of these models already exist in libraries from prior projects.

When dealing with this type of problem, one must examine all contributing component factors as sources of error. These components are selected based upon trade-offs such as cost, size, availability within time frame of interest, etc., as well as worst case scenarios to be met. Typical components include (not limited to):

- Sensors on launch platform
- Receivers on launch platform (to obtain fixes from other systems)
- Sensors on weapon (IMU, Altimeter, Air Speed, Wind Estimation, Video Sensor, ...)
- Receivers on weapon (to obtain fixes from other systems, e.g., GPS, other radios, ...)

One must also estimate error budgets for all contributing components so that specific error constraints for each unit may be addressed by design. Typical state errors may include (not limited to):

- Position, velocity, attitude and rotation of weapon
- Weapon clock bias, drift
- GPS clock bias & drift
- Electro-Magnetic wave propagation delays
- Accelerometer input misalignment, bias, scale factor error, drift
- Gyro input misalignment, bias, scale factor error, drift

Using Realistic Stress Scenarios

To produce measures of accuracy used for comparison or to assess combining of technologies, scenarios must contain all factors that are sources of error. Scenarios must produce measures of mission effectiveness and provide realistic stress conditions that test and compare different solution approaches. To speed the process, multiple vignettes may be run in parallel as shown in Figure 13-9. In addition, parametric analysis is typically required to assess the variations that occur and to drive out realistic worst case conditions. Monte Carlo analysis may be performed to determine the variance of distributions of outcomes.

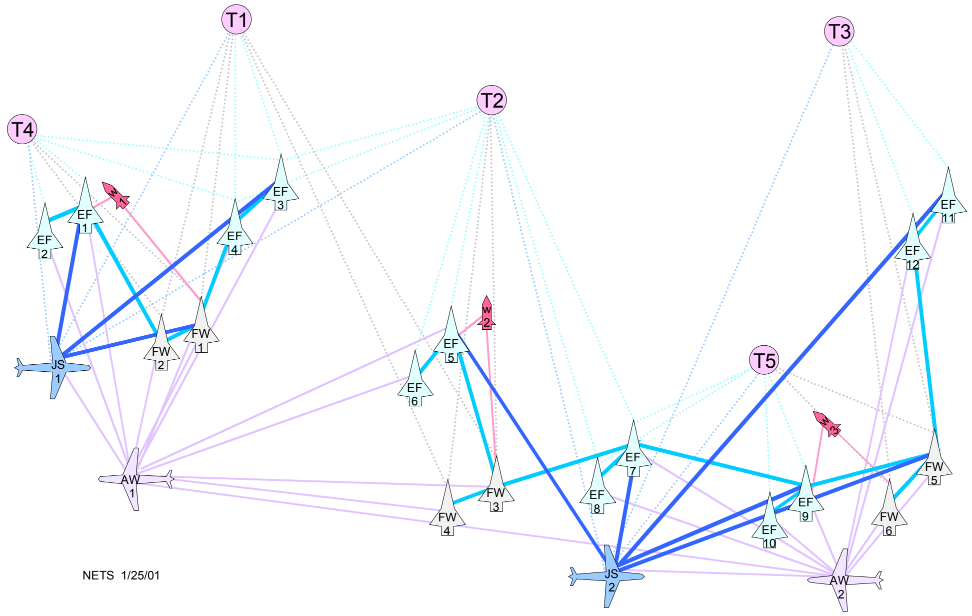


Figure 13-9. Example of a scenario of interest.

Measuring Mission Effectiveness

When making design decisions that require long term investments, they must be backed by a reasonable probability of success. Such probabilities can only be determined using detailed simulations that account for all of the factors that may result in system failure. Determining these probabilities is best accomplished by embedding the design in realistic stress scenarios to determine the effectiveness of missions depending as a function of different designs.

Mission effectiveness can only be determined by playing all of the events and messages that mark completion of each step along the chain, and tracking the cumulative error and time to complete those steps.

Messages are typically triggered by events (e.g., target sightings or mission management decisions), and incoming messages on one platform may trigger outgoing messages to others. Messages may trigger events as well as other messages, so one must model all of the events and traffic that affects mission outcomes. Measures of timing and synchronization are natural by-products of this approach.

In an RF environment, power transmitted in a frequency band is received as noise by all parties attempting to receive from a different transmitter in that band. The noise level created at a receiver depends upon many factors (power of the transmitter, distance to the receiver, antenna gains, polarization, propagation path loss due to terrain and foliage, etc.). Therefore, background traffic and noise from other sources in the bands of interest must be represented with sufficient accuracy to determine the probability of receiving a message.

Figure 13-10 illustrates one of the problems currently faced by sensor, communication and navigation systems in Northeast Afghanistan. Green lines show Radio Frequency (RF) connectivity or Line-Of-Sight (LOS) and red lines show lack of RF connectivity or LOS - often due to terrain masking. Even in this scenario which is small compared to a large theater, connectivity may be difficult. This is an example of a flight viewed interactively from the CAD system described above - while the simulation is running.

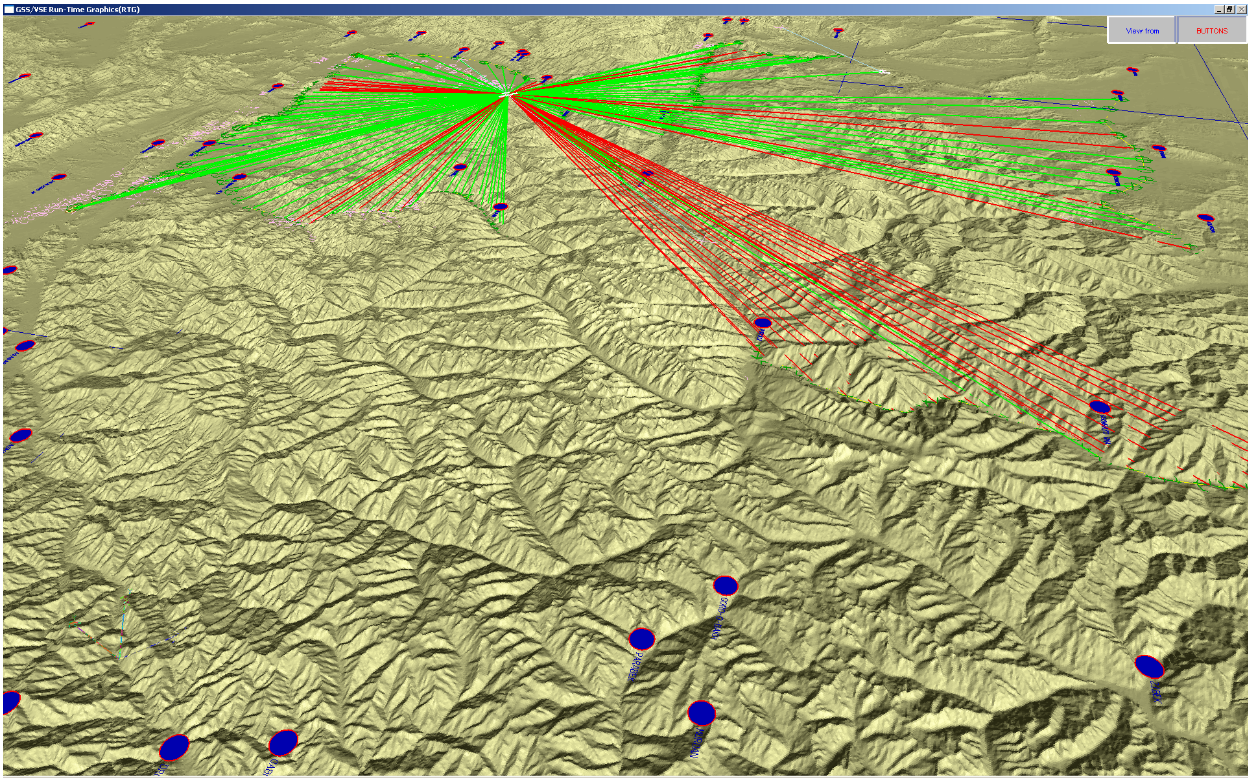


Figure 13-10. Example of RF connectivity or LOS.

14. PREDICTION AND CONTROL - A FINANCIAL EXAMPLE

Figure 14-1 illustrates the changing values of the Euro against the U.S. dollar. Clearly the values can change significantly in a short period of time. The purpose of the Position Control System is to control the movement of positions (in or out of a currency) that a trader can take given a limited number of options (e.g., a maximum of 10 currencies) so as to maximize the probability of improvement of the position (the Optimal Control problem). The case of interest here is when an action that can be taken to be in or out of a position. We will use the example of the decision to Buy or Sell a given currency. In general, a position can be in or out of multiple currencies at the same time. We may want to start by restricting our rules of engagement to be in multiple currencies only when the probability of future improvement of these currencies is close to equal.

The initial scheme described here will cause actions to be taken when the highest probability of improvement requires a move to another currency. In the case where there is more than one option at the highest level, then a position may be taken on multiple currencies.

We note that the currencies defined here are relative, in the sense that each position must be defined relative to a reference currency. For example, consider that all currency values are expressed in terms of the U.S. dollar (i.e., their cost in U.S. dollars). Thus, if all currencies are in the sell position (going down), one must be holding the reference currency, i.e., the U.S. dollar. We note that all currencies, including the U.S. dollar, may be going down relative to a position in some other market, e.g., gold. In that case, one must consider expanding the options to include a new reference position. One may view the reference position as being "out of the market" or "on the sidelines." In this view, if the currency one normally holds is the U.S. dollar, then one would satisfy these conditions by using it as the reference position.

The control system of interest here will make recommendations on buying and selling currencies valued in terms of the U.S. dollar. The decision process within the control system will be dependent upon predictions of the values of the currencies of interest over a future time horizon, e.g., 12 days. The accuracy of these predictions must be high enough to produce decisions that have an acceptable probability of a sufficient return on investment ROI over the course of a year.

Investment Considerations

The ROI must be a sufficient percentage of that investment to warrant running the system on a daily basis. The return must include all expenses, e.g., the cost of obtaining data and the cost of trading. The amount of money put on a position will affect the dollar return, with higher amounts producing higher returns, but having the risk of greater loss. For example, one may take positions on margin, but these must be counter-balanced by a low probability of loss. The purpose of the control system is to determine the best position to take based upon model predictions and the many factors that affect the investment outcome. Although this can all be automated, the actual decision to keep or change a position will likely be the responsibility of the person using the control system.

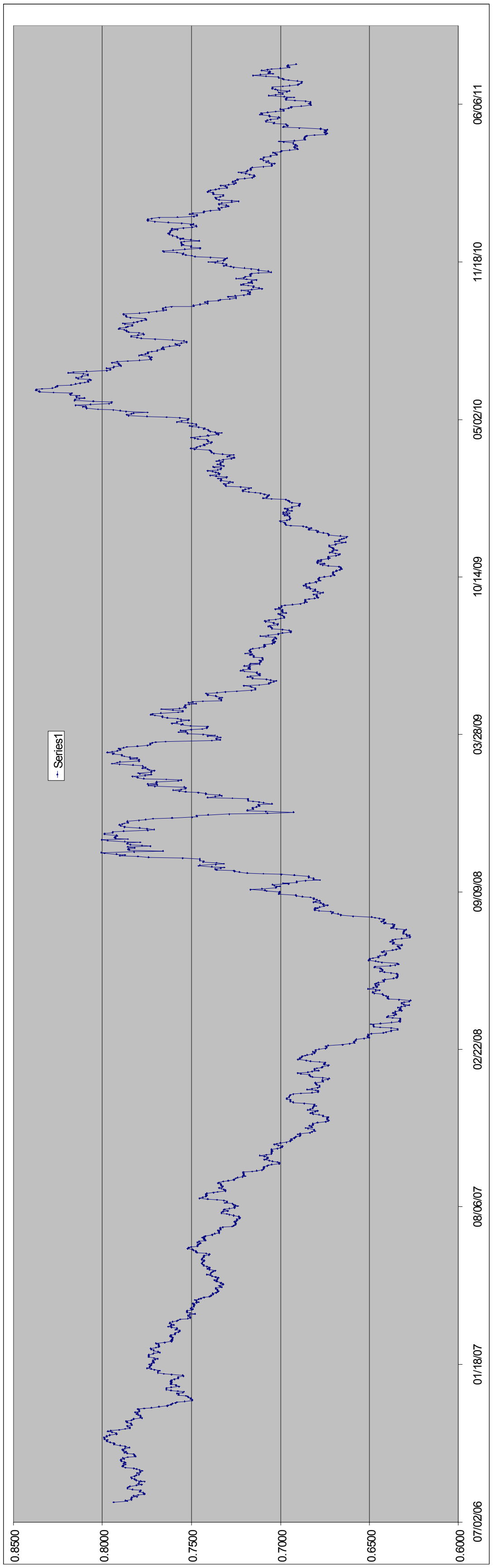


Figure 14-1. Euro versus U.S. Dollar

Technical Considerations

Figure 14-2 illustrates the movement of the predicted price of a currency whose value is increasing relative to the U.S. dollar. Points on the curve represent a predicted price at a particular time step into the future. If the maximum time horizon for prediction is 12 steps (TPQ = 12 days), then this could be a chart of predictions for any of the future days (1-12). But this chart does not depict information on accuracy or volatility. If the predictions were accurate, then one would want to buy when the currency is predicted to go up against the dollar, and sell when it is predicted to go down. In addition, one may want to buy that currency which is predicted to go up the fastest relative to other rising currencies. Buying at the lowest point and selling at the highest point as shown in Figure 14-2 is an ideal scheme, one not realistically achieved in a fair market.

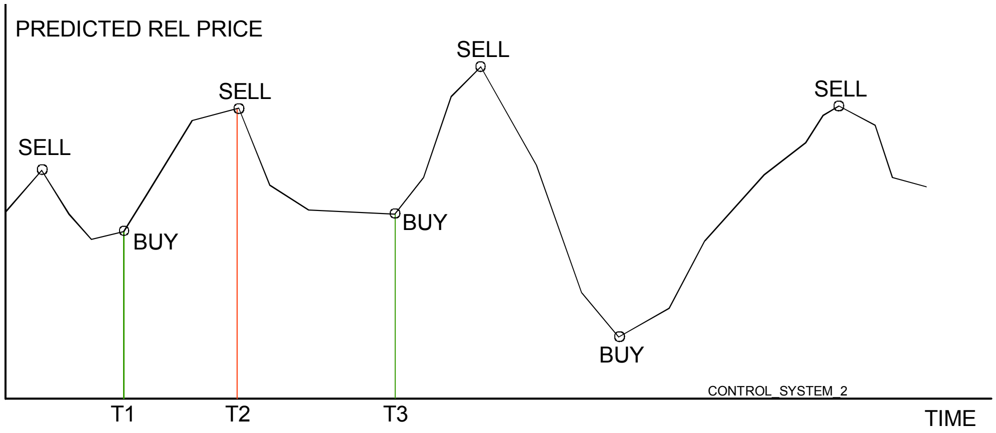


Figure 14-2. Relative price predictions.

The inherent problem one faces is the accuracy of prediction, implying that there are variations in the potential outcomes that are not accounted for. These variations must be estimated and accounted for when making decisions on which currency to buy. If the predicted rate of increase of value is sufficiently high, then one can make buy decisions, but one must interpret what is *sufficient*. To do this requires that the variations are characterized in terms of probabilities and these probabilities must be characterized in terms of confidence levels. The estimated variations must also be applied to each prediction step into the future and by their nature, they will increase with each time step into the future. Multiple time step predictions can be characterized in terms of envelopes.

Figure 14-3 illustrates an envelope prediction where the upper and lower limits of the envelope must be defined in terms of the probability of being inside the envelope. In this example, the envelope represents predictions at future time steps beyond the current time step, from TS+1 to TS+TPQ. This implies that one has the latest information on the value of the currency at time TS. Given this information, the control system must compare it to the current position to determine if it should issue a buy recommendation.

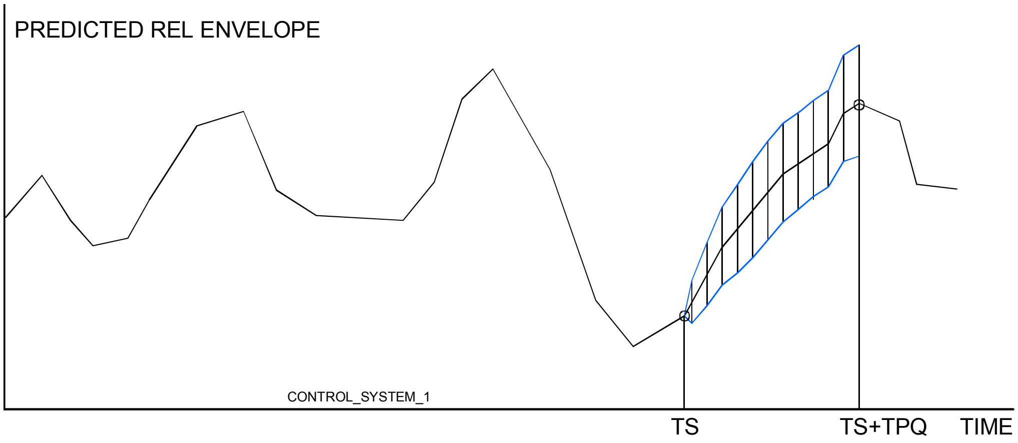


Figure 14-3. Relative price envelope for TPQ = 12 days.

To grow an investment at the fastest rate, one must be able to move to positions that satisfy the following constraints:

(1) Make a sufficient increase in value to cover the expenses of trading (the move);

(2) Match if not exceed the maximum increase in value relative to other positions.

The critical element in both of these constraints is to obtain a *sufficient increase in value within a specified time frame* of the currency in which a position is taken. The time frame is important because it determines the amount of improvement to be gained as well as the rate at which the investment is improving. For example, if a position can be taken that grows at a higher rate than all others, but levels off after a very short period of time, then one may not make enough return to cover the cost of trading. Looking at Figure 14-3, this implies that the probability of a return on a new position taken at TS based upon the value derived from the envelope at the turning point (TS+TPQ) must exceed the movement cost.

Looking at this from another perspective, one must compare the increased value obtained by making a change in position - after expenses - to that of staying in the current position which may not be growing as fast but will incur no additional trading expense.

Alternatively, the trading expense may be negligible, especially if one is trading very large amounts. In this case, one is likely looking to move to the currency that is predicted to be the fastest growing from the current point of observation out to some point in the future where another currency is predicted to grow faster. At that time, one would move to the other currency.

In addition, the amount of investment placed on a buy position affects the risk of Ruin (losing the total amount of money reserved for investments). Knowing the limited amount of total investment money, one must determine the fraction to be placed on a position given the probability distribution defining what may be lost on that position. This is reflected in Figure 14-4 which represents the distribution of the probability envelope at a given T+TP time in the future, where the probability of incurring a loss may be estimated from the distribution.

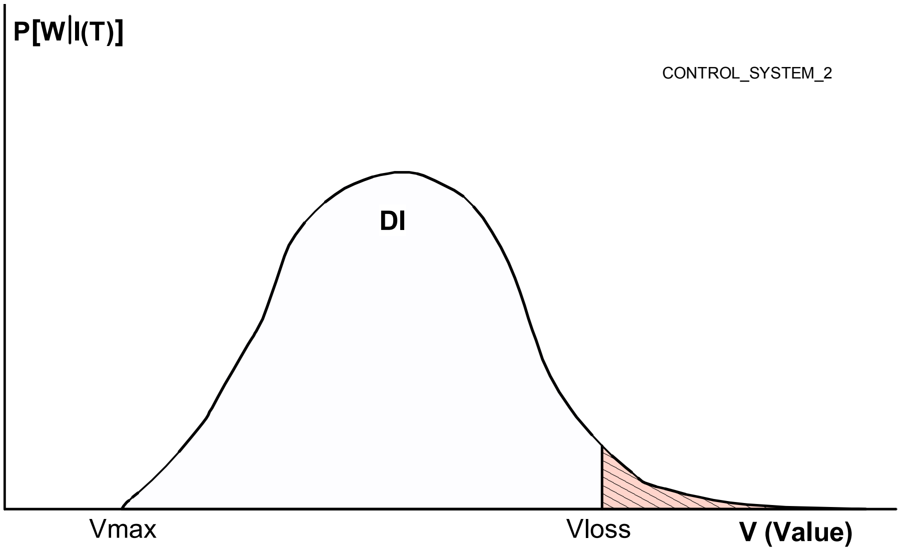


Figure 14-4. Distribution of potential value outcomes (V).

It is important to note that, in general, the loss value, Vloss, does not coincide with the limits defining the prediction envelope which may be selected to imply an 80% probability of being within the envelope, with the further implication of a 95% confidence in the limits (yielding a 0.76 overall probability). For example, if the line on the envelope leading to the time point of interest is flat, then this would coincide with the probability of Vloss being 0.76 for envelopes whose accuracy is defined as stated above. If that line has a negative slope, then the probability of loss will be greater. Similarly, if it has a positive slope, then the probability of loss will be less.

One must still be concerned about the amount of loss that may be incurred. This will be affected by the volatility of the potential outcome. We will attribute the volatility to information not contained in - or available to - the model. This may be treated as a random component whose bounds - as well as distribution - are unknown due to changes in the environment.

This may be accounted for to some extent by expanding the envelope when actual values fall outside. The amount of expansion can be a function of the distance from the point to the envelope, and can cause the envelope to expand rapidly to help ensure that the 80% level characterizes the volatility. Likewise, when there are no points outside the envelope for some predetermined period of time, the envelope width may be contracted to the 80% level, typically at a smaller rate than that used to expand it. These expansions and contractions provide some measure of volatility. However, it must be noted that they occur after the fact (unexpected movement) without prior information.

Volatility can be modeled to some extent, but requires human judgment regarding changes in the market environment not accounted for in the models. Some of these effects may be incorporated after the fact, improving the ability of the model to deal with similar movements in the future. The more information known in advance of such changes that can be placed in the model, the more precautions that may be taken by the position control system.

Buy / Sell Price Spread Model

Based upon the envelope predictions, and the spread between the buy price and sell price (cost of trading), a decision must be made as to whether it pays to change a position or stay with the current position. The envelope must be predicted to change sufficiently in a positive direction relative to the U.S. dollar to ensure that a trade will be profitable. Thus, predicted changes in the envelope must be larger than the cost of trading (it can also be changing with the volatility of the currency). The relationship between the envelope and the trading spread must be determined so that sufficiently accurate predictions of the future spreads can be produced by this model. The accuracy of the prediction system coupled with the accuracy of factors accounted for in the position control system will determine the profit of trading opportunities. As accuracy is improved, more profitable trading decisions can be made for the same relative risk.

Trading Policies

There are two ways to profit from a currency. Trades can be made by buying a currency and holding it until it rises sufficiently against the dollar, or by taking a short position until the currency falls sufficiently against the dollar. If ten currencies are traded against the U.S. dollar, then there are twenty possible positions that can be taken against the U.S. dollar. Positions can be moved directly as predictions change and different currencies switch relative to being the most desirable one in which to take a position.

It may be desirable to take a position in more than one currency at a time, provided a sufficient profit is predicted for each currency. Volume of trades can be weighted in accordance with the probabilities that each currency has for being profitable.

One can also hedge a trading position by buying an option to limit the exposure to a large loss. This involves evaluating the tradeoffs between probable profits, likelihood of a large loss, and cost of the option.

15 ACCURATELY PREDICTING U. S. CORONAVIRUS - DAILY

UNDERSTANDING THE PROBLEM

Unlike many of the other problems addressed here, the CORONAVIRUS (COVID-19) is a medical outbreak that came upon many countries on earth unexpectedly. Its rapid expansion caused heavy demands for facilities (hospitals, beds), equipment (ventilators, masks, etc.) and people. This required layered support from many different organizations across the globe. To suppress the spread, political organizations laid down restrictions on individuals within their purview. These restrictions included elimination of travel, group meetings, and confinement to their homes. This was met with major problems since it could not be applied to many people who had to support the medical environment and supply chains for food and other essential day-to-day living requirements such as toilet paper and cleaning supplies.

Having imposed heavy restrictions on the population in general, except those in the special medical and supply chain needs, politicians such as mayors, governors, the U.S. president and his special CORONAVIRUS team were soon pressed to relieve restrictions where they were not clearly needed. Because of the apparent differences in effects in different geographical areas, this reduction of restrictions soon became a careful political issue. In areas that had problems that appeared to be contracting, how does one know when to reduce or remove the restrictions without causing the problem to expand again.

This quickly led to models producing curves describing the rising and falling trend of the future spread. But these models were basically curves indicating the rise and fall of the spread. As data became available to check out these models (curves), it became apparent that they were way off the actual test data. Although one can now argue that many more people had it but did not know it, the original objective was to predict the need for hospital facilities, equipment and people to support the huge number of people that would need help.

As people became stressed by confinement to their homes, the small number of cases in many areas and leveling of cases in others caused significant calls for reduction in the restrictions. This was followed by a plan at the federal level for states to relax constraints on a planned basis. The U.S. President stated that the state governors would make the decisions on the relaxation of the constraints. Mayors have looked to people at the county level to help in such a decision. The possibility that a large number of people could have the virus with little or no symptoms, and could pass it on has become a major concern.

The bottom line: How does one predict where the virus is headed. Or, more importantly, if decisions are made to relax the restrictions, how will those decisions affect future outcomes. As described below, this is not a simple problem. There are many factors, including the delay between contraction and recognition, assuming the effects are recognizable. To produce accurate predictions requires a complex model that represents all of the contributing factors. The approach to this model is described below. It starts with an understanding of how these factors show up in the existing data. It quickly becomes clear that just as the restrictions must be levied based on outcomes in specific areas, the model must represent these specific areas.

OVERVIEW

Based on the data for seven sample dates in Monmouth and Ocean Counties in New Jersey, one can see the huge differences in the percentage of cases based on the differences in numbers and types of populations in the municipalities. (See data on the next two pages). This data is available and necessary on a daily basis. These daily differences reflect the well-defined factors, e.g., behavior - on the spreading of the CORONAVIRUS. These differences become smoothed when one combines this data into total county data - or even worse, state level data. Even though large differences exist between counties and states, the relative weights on causes of the differences do not stand out – and cannot be identified - except at the municipal level.

This leads to the obvious conclusion that the causal factors must be weighted at the municipal level to produce accurate predictions of future spreading or decline. In addition, when performing predictions at the municipal level, one must understand the amount of work to be done before one can determine if that work is best done at the state or county level. The relative quantity of work is obvious from the chart in Figure 1 below.

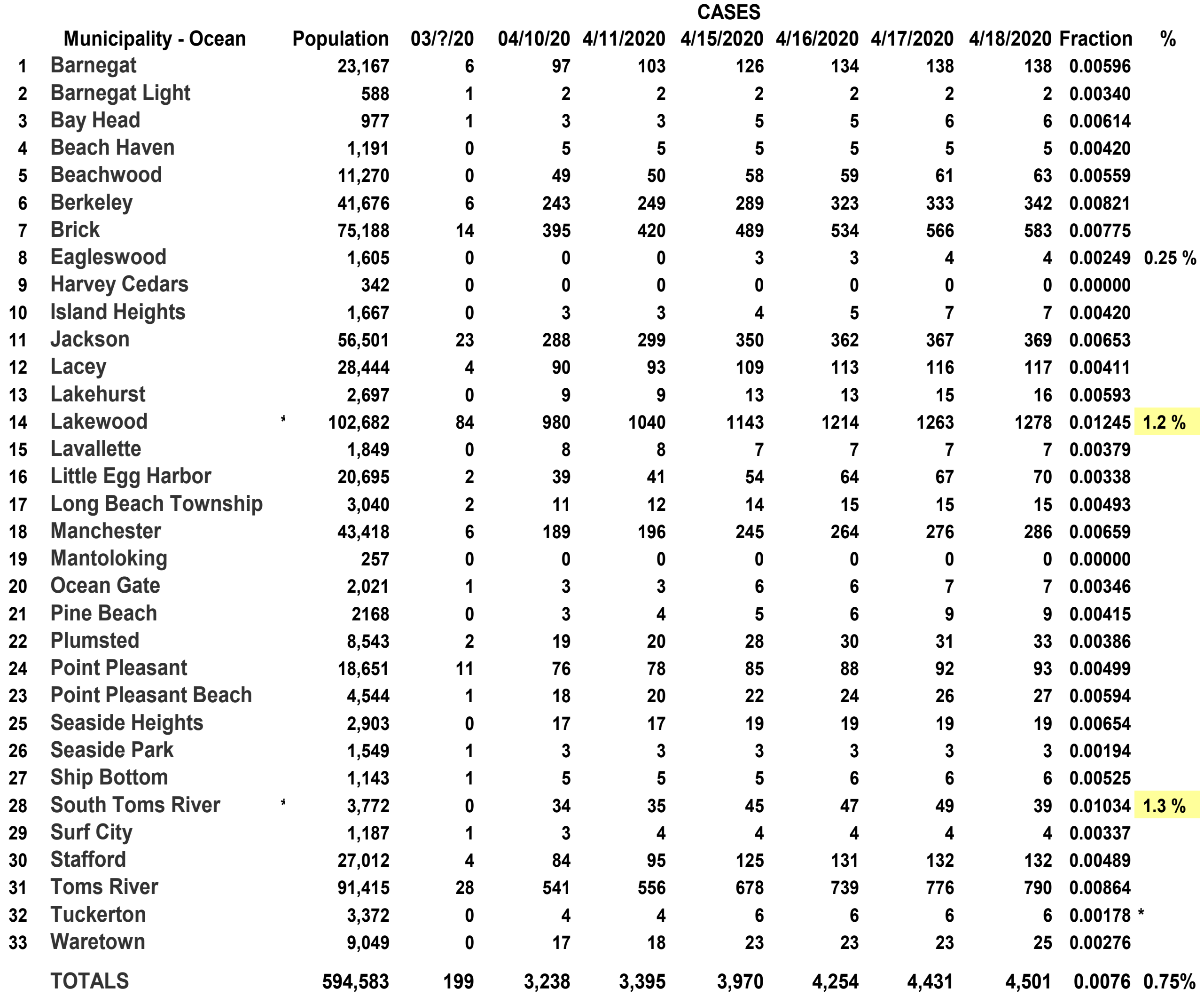
DATA ON THE UNITED STATES	QUANTITY
NUMBER OF STATES	50
TOTAL NUMBER OF COUNTIES	3141 + D.C = 3142
TOTAL NUMBER OF MUNICIPALITIES	19,522
NUMBER OF COUNTIES IN NJ	21
NUMBER OF MUNICIPALITIES IN NJ	565
NUMBER OF MUNICIPALITIES IN MONMOUTH COUNTY	54
NUMBER OF MUNICIPALITIES IN OCEAN COUNTY	33
NUMBER OF MUNICIPALITIES IN BURLINGTON COUNTY	40

Figure 1. An extremely rough look at possible situations.

If one had to do all of the municipalities at the state level of New Jersey, one would require processing the data for 565 municipalities every day. This becomes apparent when looking at a sampling of the data in Figures 2 and 3 below. Doing the daily predictions may require reassessing weights on the factors described in the sections below**.** As described in this chapter, producing predictions for all of the municipalities at the county level requires a reasonable effort. One must consider potential changes in weights applied to each of the factors that affect the changes in number of cases within a municipality. An example is people changing their behavior. This requires a sufficient understanding of the model, and how to perform steps to optimizing the coefficients on these factors to produce accurate predictions. This effort can be done by a person with some mathematical background at the county level. Accurate county data is determined simply by adding the municipal data. States can then combine the accurate county level data.

						CASES					
	Municipality - Monmouth	Population	03/?/20	04/10/20	04/11/20	04/15/20	04/16/20	04/17/20	04/18/20	Fraction	%
1	Aberdeen	18,372	10	104	105	115	121	123	127	0.00691	
2	Allenhurst	489	0	2	2	2	2	2	2	0.00409	
3	Allentown *	1,811	0	2	2	3	3	3	4	0.00221	0.22%
4	Asbury Park	15,511	6	62	68	79	79	87	91	0.00587	
5	Atlantic Highlands	4,316	2	14	13	14	16	17	17	0.00394	
6	Avon By The Sea	1,780	0	10	10	8	9	9	9	0.00506	
7	Belmar	5,587	2	6	8	8	8	8	10	0.00179	
8	Bradley Beach	4,174	2	15	14	17	17	19	19	0.00455	
9	Brielle	4,691	4	19	20	20	22	22	22	0.00469	
10	Colts Neck	10,018	8	46	47	49	50	50	51	0.00509	
11	Deal *	723	0	21	22	22	23	23	23	0.03181	3.2%
12	Eatontown	12,242	13	100	103	124	126	132	134	0.01095	
13	Englishtown	1,925	5	12	13	15	16	16	16	0.00831	
14	Fair Haven	5,820	10	15	15	17	17	17	18	0.00309	
15	Farmingdale	1,321	1	9	9	9	10	10	9	0.00681	
16	Freehold *	11,767	1	93	106	127	135	140	144	0.01224	1.2%
17	Freehold Township	35,429	28	288	304	344	357	365	388	0.01095	
18	Hazlet	20,082	17	138	139	159	161	164	168	0.00837	
19	Highlands	4,769	0	12	12	18	19	19	19	0.00398	
20	Holmdel	16,648	8	112	115	139	147	147	147	0.00883	
21	Howell	52,076	14	316	332	366	386	395	397	0.00762	
22	Interlaken	821	0	11	11	1	1	1	1	0.00122	
23	Keansburg	9,719	2	60	62	64	70	72	77	0.00792	
24	Keyport	7,053	1	39	39	50	53	53	55	0.00780	
25	Lake Como	1,694	1	10	11	12	12	13	11	0.00649	
26	Little Silver	5,813	12	24	24	25	26	26	24	0.00413	
27	Loch Arbour	183	0	1	1	1	1	1	1	0.00546	
28	Long Branch	30,406	7	184	196	229	236	248	251	0.00825	
29	Manalapan	40,096	27	274	282	312	320	320	321	0.00801	
30	Manasquan	5,846	3	20	20	22	23	25	23	0.00393	
31	Marlboro	40,466	29	267	271	294	303	309	310	0.00766	
32	Matawan *	8,736	3	85	90	106	112	114	115	0.01316	1.3%
33	Middletown	65,952	22	284	296	330	336	348	362	0.00549	
34	Millstone	10,522	3	41	42	47	51	53	52	0.00494	
35	Monmouth Beach	3,288	2	14	14	14	14	16	16	0.00487	
36	Neptune	27,728	12	188	197	213	223	227	230	0.00829	
37	Neptune City	4,645	1	20	21	22	23	25	24	0.00517	
38	Ocean Grove	3,342	1	5	5					0.00000	
39	Ocean Township	27,006	7	153	157	170	178	181	183	0.00678	
40	Oceanport	5,751	4	40	40	39	39	39	40	0.00696	
41	Red Bank	12,048	2	64	72	83	85	87	94	0.00780	
42	Roosevelt	854	0	3	2	2	2	2	2	0.00234	
43	Rumson	6,776	5	23	23	24	24	24	25	0.00369	
44	Sea Bright	1,364	0	8	7	7	7	7	7	0.00513	
45	Sea Girt	1,771	3	9	9	9	9	9	9	0.00508	0.50%
46	Shrewsbury Boro	4,085	1	21	21	24	24	25	24	0.00588	
47	Shrewsbury Township	1,117	2	4	5	6	7	7	7	0.00627	
48	Spring Lake	2,925	0	6	6	6	6	8	8	0.00274	0.27%
49	Spring Lake Heights	4,555	0	13	14	15	15	15	15	0.00329	
50	Tinton Falls	17,563	6	61	62	76	82	89	95	0.00541	
51	Union Beach	5,562	0	18	18	28	29	30	30	0.00539	
52	Upper Freehold:	6,975	5	26	26	26	28	29	29	0.00416	
53	Wall	26,020	4	110	118	141	151	160	159	0.00611	
54	West Long Branch:	7,909	4	40	40	46	48	49	50	0.00632	
	TOTALS	628,142	300	3,522	3,661	4,099	4,262	4,380	4,465	0.0071	0.68%

Figure 2. An extremely rough look at possible situations.



							CASES					
	Municipality - Ocean		Population	03/?/20	04/10/20	4/11/2020	4/15/2020	4/16/2020	4/17/2020	4/18/2020	Fraction	%
1	Barnegat		23,167	6	97	103	126	134	138	138	0.00596	
2	Barnegat Light		588	1	2	2	2	2	2	2	0.00340	
3	Bay Head		977	1	3	3	5	5	6	6	0.00614	
4	Beach Haven		1,191	0	5	5	5	5	5	5	0.00420	
5	Beachwood		11,270	0	49	50	58	59	61	63	0.00559	
6	Berkeley		41,676	6	243	249	289	323	333	342	0.00821	
7	Brick		75,188	14	395	420	489	534	566	583	0.00775	
8	Eagleswood		1,605	0	0	0	3	3	4	4	0.00249	0.25 %
9	Harvey Cedars		342	0	0	0	0	0	0	0	0.00000	
10	Island Heights		1,667	0	3	3	4	5	7	7	0.00420	
11	Jackson		56,501	23	288	299	350	362	367	369	0.00653	
12	Lacey		28,444	4	90	93	109	113	116	117	0.00411	
13	Lakehurst		2,697	0	9	9	13	13	15	16	0.00593	
14	Lakewood	*	102,682	84	980	1040	1143	1214	1263	1278	0.01245	1.2 %
15	Lavallette		1,849	0	8	8	7	7	7	7	0.00379	
16	Little Egg Harbor		20,695	2	39	41	54	64	67	70	0.00338	
17	Long Beach Township		3,040	2	11	12	14	15	15	15	0.00493	
18	Manchester		43,418	6	189	196	245	264	276	286	0.00659	
19	Mantoloking		257	0	0	0	0	0	0	0	0.00000	
20	Ocean Gate		2,021	1	3	3	6	6	7	7	0.00346	
21	Pine Beach		2168	0	3	4	5	6	9	9	0.00415	
22	Plumsted		8,543	2	19	20	28	30	31	33	0.00386	
24	Point Pleasant		18,651	11	76	78	85	88	92	93	0.00499	
23	Point Pleasant Beach		4,544	1	18	20	22	24	26	27	0.00594	
25	Seaside Heights		2,903	0	17	17	19	19	19	19	0.00654	
26	Seaside Park		1,549	1	3	3	3	3	3	3	0.00194	
27	Ship Bottom		1,143	1	5	5	5	6	6	6	0.00525	
28	South Toms River	*	3,772	0	34	35	45	47	49	39	0.01034	1.3 %
29	Surf City		1,187	1	3	4	4	4	4	4	0.00337	
30	Stafford		27,012	4	84	95	125	131	132	132	0.00489	
31	Toms River		91,415	28	541	556	678	739	776	790	0.00864	
32	Tuckerton		3,372	0	4	4	6	6	6	6	0.00178	*
33	Waretown		9,049	0	17	18	23	23	23	25	0.00276	
	TOTALS		594,583	199	3,238	3,395	3,970	4,254	4,431	4,501	0.0076	0.75%

Figure 3. An extremely rough look at possible situations.

Looking At Actual Data

Figure 4 below is a plot of three municipalities (Aberdeen, Freehold and Neptune) in Monmouth County, NJ containing 54 daily data points for each starting at 03/30/20 and ending at 05/21/20. This implies the ability to test the 5 to 10 day prediction accuracy for each of 20 day start times. Note that the values for cases are summations to date implying actual testing requires taking the difference. We also note that the County of Monmouth has indicated that inaccuracies were detected after assignments were made during the periods 04/19 to 04/22 and 05/03 to 05/04. These are easily adjusted to be representative of the correct values.

Upon scanning the ratio of cases / population in Monmouth County municipalities in Figure 2, one notes a factor of 10 difference in some of these ratios at different times independent of populations. The latest minimum ratio is 0.387% while the maximum is 3.5%. Most of the differences in ratios appear to be caused by behavioral differences. It is also clear that many of the municipalities are still rising rapidly, while some have leveled off.

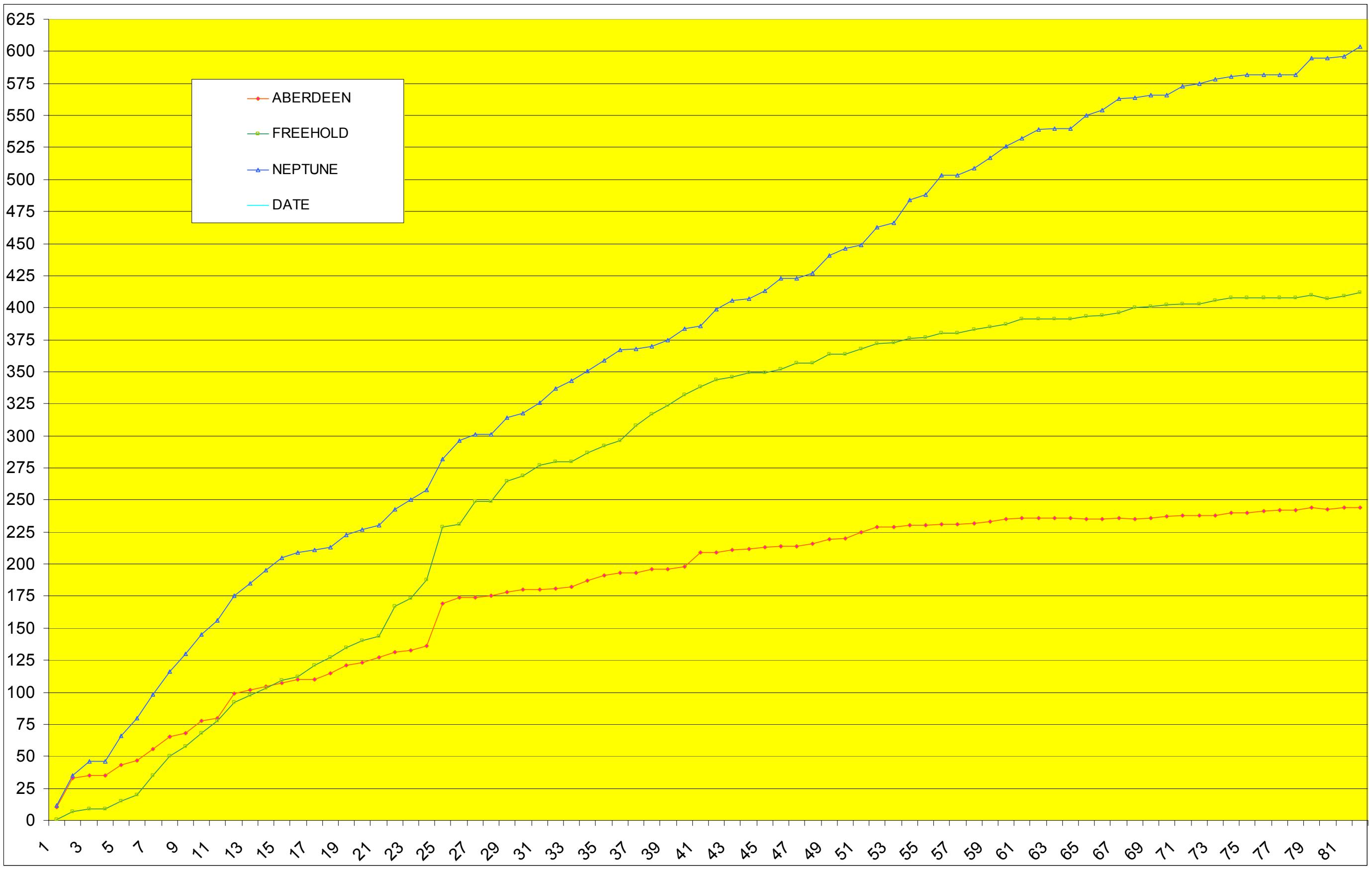


Figure 4. Plots of daily tested cases for three municipalities in Monmouth county, NJ.

PSI has obtained sufficient data to test and prove the accuracy of its municipal model, and expects to be able to offer a copy of the PSI model to all counties in the U.S. to run it daily for each of their municipalities. This will require training to use the optimization facilities described in Chapter 9 to track and optimize the changing factors affecting each municipality.

EXPANDING TECHNOLOGY

One must start by understanding basic principles of science – principles that many take for granted. Organizations depending on some technologies - e.g., those building electronic chips, phones, satellites, and medical solutions - must improve their technology to remain competitive and help society. To do this, scientists must *seek the truth* based on underlying physics, chemistry, biology and mathematics. As stated by Lord Kelvin and quoted by Anselmo and Ledgard, [36], “When you can measure what you are speaking about, … you know something about it; but when you cannot measure it, … your knowledge is of a meager and unsatisfactory kind …” This implies developing solid measures of both the positive effects and shortcomings of a new approach.

In some technologies, it is difficult to get people to define - let alone take - the required measures. In others, measures are taken on a regular basis. In the latter case, use of the technologies are likely to expand rapidly. This implies ease of understanding of measures and tests that demonstrate results. As shown below, this becomes obvious with prediction models.

IMPORTANCE OF FACTORS AFFECTING THE CORONAVIRUS SPREAD

Major factors affecting accurate predictions of the virus spread are described below. Upon reading these descriptions, one can observe the importance of using measures at the municipal level. This is because huge differences in outcomes are the obvious result of differences in the properties and behavior between adjacent municipalities. These differences are essential to determine weights on the corresponding model factors that produce accurate predictions.

Contractions And Deaths At The Municipal Level (By County By State)

This data is recorded on a daily basis and contains the number of people who have contracted the virus and who have died in a given municipality in a given county and state.

The importance of comparing daily municipal level data is shown by Lakewood - a municipality in Ocean County, NJ which had 84 cases of the virus in a population of 102,682. On that day, Manchester - also in Ocean County - had 6 cases of the virus with a population of 43,418. Manchester is located just below Lakewood, in a very similar environment.

Similarly, in Monmouth County, NJ, the Freehold municipality had 1 case out of a population of 11,767, while on the same day Little Silver municipality had 12 cases with a population of 5,813. Clearly there is at least one significant factor affecting the above differences. This shows the need to use municipal level data to weight the importance of factors that cause the differences.

Accuracy of municipal level predictions directly affects that of a county and state. Using these differences, one can weight the importance of causal factors affecting the probability of spread of the disease much more accurately, and therefore the ability to improve actions to be taken to avoid contraction, as well as improve accuracy of prediction.

Weather For The Municipality

It’s been determined that sun affects the spread of corona particles that decreases in the summer. Wind and rain may also affect the spreading of particles. These factors must be tested to understand their importance as it affects prediction accuracy at the municipal/county level.

Collaborative Functions At The Municipal Or County Level

Collaborative functions attracting 10 or more people are conducive to spreading the virus, especially if numbers are large and tightly grouped. Collaborative functions that are held at the municipal or county level must be noted. It is especially important to develop factors of impact based on past events (e.g., the problem with Lakewood) and their effects. Religious, sports, and school gatherings (even shopping) have been cited as major contributing causes. These factors must be quantified to improve measures to avoid contraction and improve prediction accuracy.

Living And Working Quarters At The Municipal Level

One must characterize the housing environment and the percent population it represents, e.g., percent of people in small condominiums or row houses versus single houses on large land parcels. One must also consider equivalent conditions in which people work. This can be used to determine the average distance and forced interaction between people and the limited surroundings they share. If daily travel conditions can be estimated based on a municipality, this may provide an additional factor.

Age and Gender At The Municipal Level

Age and gender are significant factors that affect the probability of contraction and death. It is apparent that younger people tend to get the virus to a smaller level that leads them to ignore the minor symptoms. They do not report having the virus. Random testing has shown that the number of people not reporting is a major percentage of populations that have a significant level of people under 30. This leads to the major difference in Figures 5 and 6 that show much higher actual numbers versus early test numbers. Men also have a higher percentage of cases than women. Averages for age groups and gender in a municipality improve prediction accuracy.

Personal Habits At The Municipal Level

This includes keeping a distance from others when near people who could be carriers or have been close to carriers. These include people one knows and relatives - who may not know they have it. As indicated above, people can contract it but not be aware of it. It takes about two to three weeks for the virus to take effect. Even after this period, some people are immune and do not notice the effects. It also includes where people go if they leave their living quarters, e.g., shopping. It also includes habits, e.g., wearing masks or scarves, cleanliness, e.g., washing their hands, etc. Finally, personal habits determine whether people attend collaborative functions, a major factor in some municipalities. All of these factors can change as people learn about the results of certain habits over time. The effects of personal habits must be monitored to affect changes in the model as a function of time.

The Effects Of Time Constants

There are also time constants associated with factors. Most important is the 2 to 3 week period between contraction and observation. Such delays must be accounted for in the model.

A ROUGH LOOK AT POSSIBLE SITUATIONS

Figures 5 and 6 below are not based on actual data. They are only drawn to provide what is indicative of the problem with many models. The blue curve represents data based on testing (only 9 points are marked). It is not intended to reflect actual data. The red curve - ACTUAL data - represents an example of the actual cases - to differentiate it from TEST data.

The critical observation is that the number of people representing the actual data has been well above the test data, and certainly at the beginning of testing. Only after people demonstrated significant differences in levels of effects (some did not know they had it) did anyone recognize the substantial difference. As the number of people affected rose rapidly, the level of concern started to rise exponentially, and so did the amount of testing.

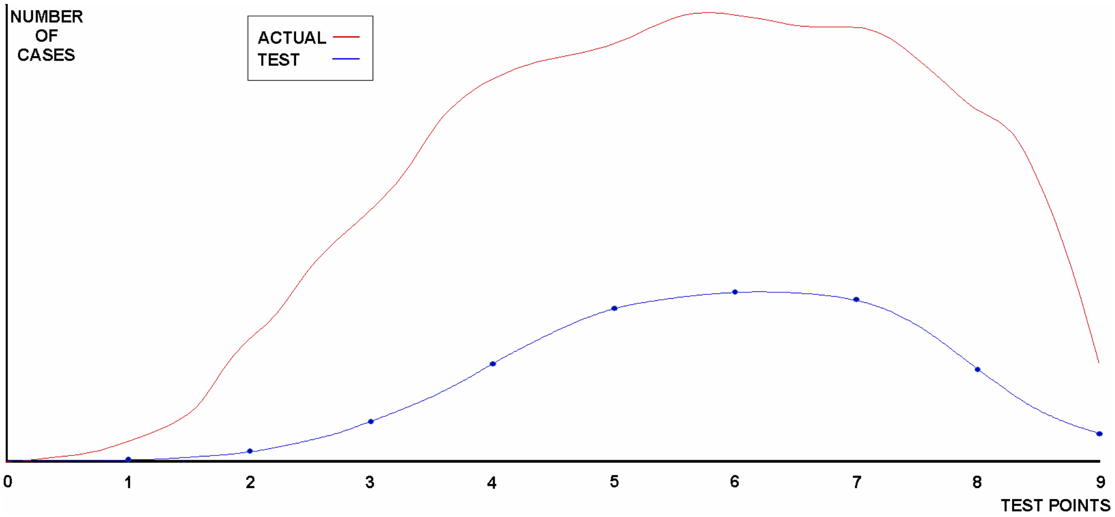


Figure 5. An extremely rough look at possible situations.

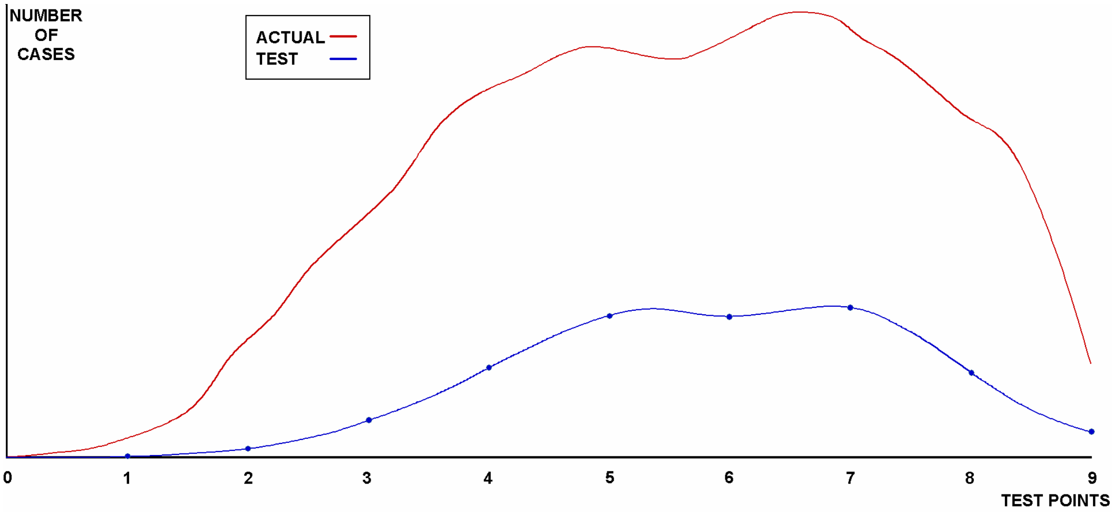


Figure 6. Another rough look at possible situations.

Even though tests are biased, limited to those having real symptoms, it is clear that the actual number of cases is well above those reported by tests. Without sufficient random testing, it is hard to estimate the actual number. But, limited test facilities have prohibited early random testing. Furthermore, people that had only minor effects were back to normal in relatively short time frames. Although these people do not test positive, they are identified by their immunity.

Since the Second World War, global travel has increased dramatically, and travel causes a virus to spread from country to country. Depending on the size of a country having an early introduction of the virus, and the nature of the introductions within that country, the test curve may take a dip as illustrated in Figure 6. Clearly travel between countries, and then between states, counties and municipalities within them represents important factors in determining the spread of the virus. One must be aware of such situations when determining when and where travel affects the spread.

As indicated in Figure 6, even though there is the start of a decline, this can be turned around depending upon many factors including testing and corresponding restrictions. This is also affected by people who have it and don't realize it, but still produce particles that spread it.

Most important is the complex nature of the spread and the corresponding approach to testing. A person with mild symptoms may not know they have it and pass it on to others who may or may not become highly affected. If those tested only represent a small part of the population (those reporting it), particularly with respect to those who did not know they had it, then the tests will not represent the actual cases and potential for expansion.

In areas with many younger people, testing is limited because younger people have limited effects and don't realize they have it. These young people can spread it, causing further outbreaks. When testing is limited to small samples, e.g., those with apparent symptoms, one can draw incorrect conclusions about important factors.

Additional factors are necessary to characterize the state of a county or country. These are the levels and time constants associated with spreading among different types and ages of people. Those living in tight quarters (major cities) will be most affected. When trying to determine the state of a given community, county or country, all the above considerations must be taken into account. This presents the need to quantify the factors affecting the spread, including their relative time constants based on age and gender, and their surrounding living conditions.

Achieving Prediction Accuracy

We note that, based on existing municipal data, the above factors can vary considerably from municipality to municipality. To use these factors in a model, one must express them in terms of mathematical functions that represent the dynamics of the physical systems. Then, by optimizing coefficient multipliers on these functions to match the actual data - at the municipal level - one can predict the future values of a county much more accurately. This translates to a much more accurate prediction of the outcomes of a state.

A mock example of the number of cases of COVID-19 in a single municipality is illustrated in Figure 7 for January through April. Figure 8 illustrates PSI's prediction approach using an 80% envelope encasing predicted results up to 19 days in advance. This implies that the actual outcomes will be inside the predicted envelope 80% of the time.

Changes To The Factors

Although there are at least five factors now, this may change with an improved understanding of the causal factors. The existing factors may also change based on the physics of the municipality, and be updated by rerunning the optimization system over the most recent data set for that municipality. These updates should not be hard to implement at each county. This requires training on the use of the VisiSoft optimization system.

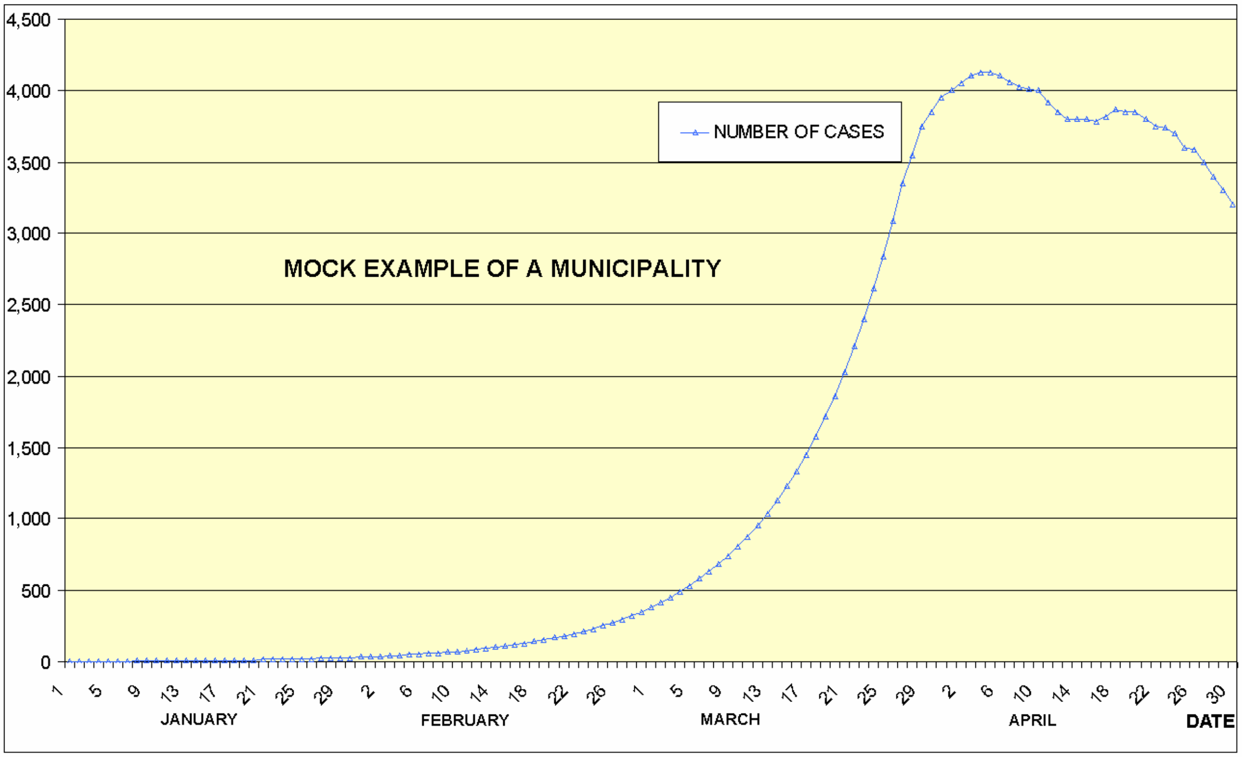


Figure 7. A Mock example of a particular municipality in some county in some state.

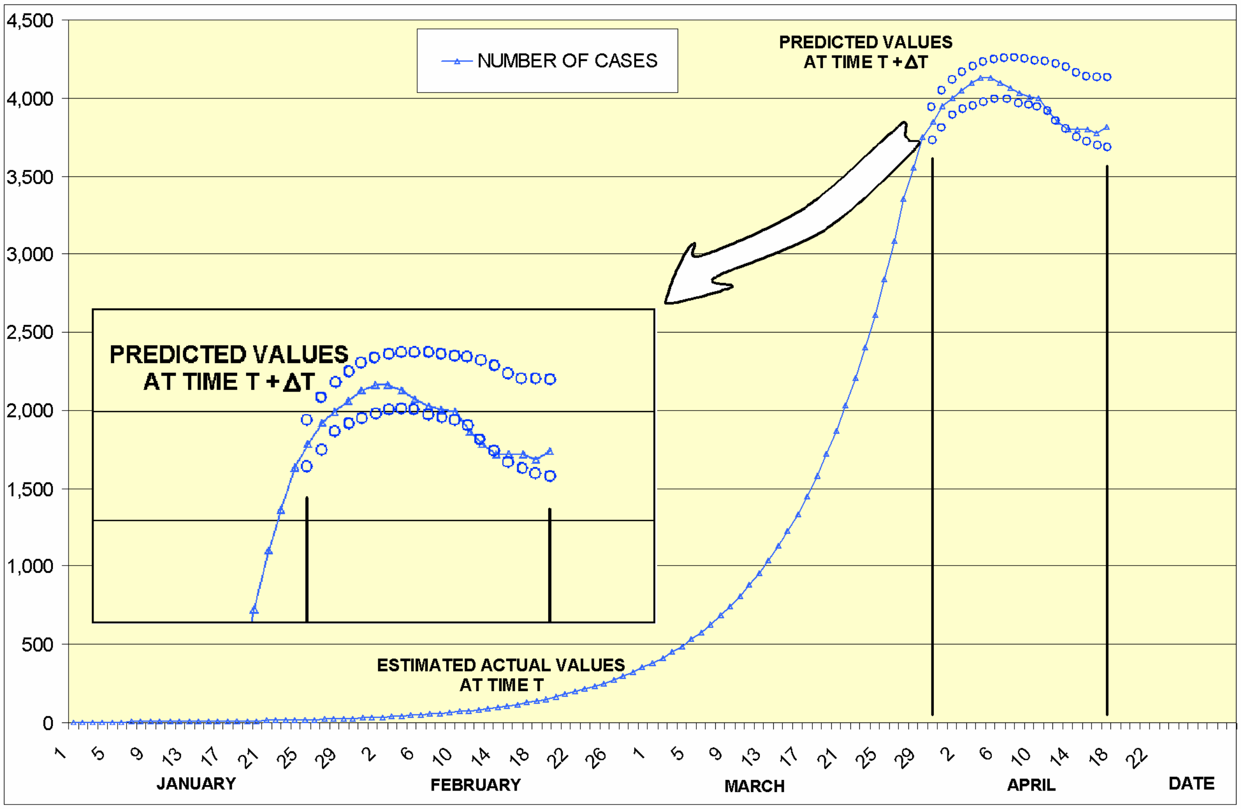


Figure 8. A Mock example of PSI's prediction model using an envelope of predicted outcomes.

Plotting The Results

Figures 7 and 8 represent plots of data and resulting predictions. Figure 7 is a mock representation of a possible output plot of cases by day over the months of January through April. Figure 8 is a mock representation of a plot that includes the envelope predictions for a 19 day period in April. The theory behind the predictions is explained in the section below.

MATHEMATICAL MODELS

The subset of equations below represents an approach to estimating the actual number of people with the CORONAVIRUS by determining an approximation to the difference between the test data and the actual values. The estimation function must be determined using a random sampling of tests of the entire population. This sampling may be small compared to the test on those believed to have the virus. These random sample tests may be done on a few selected municipalities to provide data for optimizing the parameters. It may be possible to do (almost) complete testing of very small communities to verify the approximations. The New Jersey counties/municipalities with whom we are familiar produce this data on a daily basis.

Mathematical Model Entities

Having produced an initial set of functions, they can be used to estimate the functional difference between municipalities. These can be used to determine the need for an approach to additional random testing to characterize remaining municipalities. The following definitions provide an overview of the approach. Note: Time(56) = Day 56. End of time ==> end of day

$\text{Time}(T + 1) = \text{Time}(T + \Delta T)$ where $\Delta T = 1$ day

$N_P(T)$ = Total population of the area of interest at time T

$N_A(T)$ = Number of actual cases at time T - This is being predicted but is not tested

$N_D(T)$ = Number of cases of death during time T - Available data

$N_R(T)$ = Number of total recoveries during time T - Must be estimated

$N_T(T)$ = Number of cases tested positive during time T (excludes deaths and recoveries)

$N_\Delta(T) = N_A(T) - N_T(T)$ = Difference between total cases and tested cases at time T

$N_C(T)$ = Number of people unaffected (prone to infection) to date at time T
$= N_P(T) - N_A(T) - \Sigma N_D(T) - \Sigma N_R(T)$

Note that, if the above data is not available daily, then the equations below must be modified to suit the availability.

Estimation Of Unmeasured Values (To Date)

Unmeasured values needed to do computations must be determined through estimation. This can be done by optimizing coefficient multipliers on known or estimated quantities as described below. These coefficients are generally considered to be constant, but may vary with time as well as municipality, and may have to be optimized for different time periods. Ideally, as more measurements are taken using random samples, estimates of these values will be known.

COEFFICIENTS   *** Multipliers on the following attributes

N_A - Number of ACTUAL_CASES = Coeff * TEST_CASES

```
     1  C_ACT_CASES                   REAL INITIAL_VALUE 5
```

N_R - Number of RECOVERIES = Coeff * ACTUAL_CASES

```
     1  C_RECOVERIES                  REAL INITIAL_VALUE 0.2
```

N_T - Number of TESTED_CASES = Coeff * ACTUAL_CASES

```
     1  C_TESTED_CASES                REAL INITIAL_VALUE 0.2
```

ΔN_A - Number of ACTUAL_INCREASES = Coeff * UNAFFECTED (population – affected)

```
     1  C_ACT_INCREASE                REAL INITIAL_VALUE 0.2
```

N_D - Number of DEATHS = Coeff * ACTUAL_CASES

```
     1  C_DEATHS                      REAL INITIAL_VALUE 0.01
```

Model Equations

To predict what will happen on a daily basis, one must estimate how the changes occur. To start, one must predict the actual cases at time T+1:

$$N_A(T+1) = N_A(T) + \Delta N_A(T) = \text{Total cases at time } T+1$$

By definition: $\Delta N_A(T)$ = New cases during time interval [T, T+1]. These cases are not known to the persons at the time of infection, but will appear to a portion of them (≈ 20% - depending on the municipality) after 2 to 14 days. At that time, those that recognize they have it will likely be tested. To predict the actual cases for the next day:

$$N_A(T+1) \approx N_A(T) + F_\Delta(T)$$

Where $F_\Delta(T)$ is computed using functions affecting the daily change,

$$F_\Delta(T) = N_C(T) * [C1*N_A(T) + C2*WH(T) + C3*GA(T) + C4*LQ(T) + C5*CF(T) + C6*PH(T)]$$

Where $N_C(T)$, is those unaffected to date, and

WH(T) is a function of Weather (sunlight, temperature, wind, moisture);

GA(T) is a function of Gender and Age (Must split each gender into 3 age groups);

LQ(T) is a function of Living Environment (May be split;

CF(T) is a function of attendance at Collaborative Functions (Must identify functions);

PH(T) is a function of Personal Habits.

Then,

$$N_T(T+1) = C_A*N_A(T+1)$$

where $N_T(T)$ is the number of existing cases during time T that were tested positive, and $F_\Delta(T)$ is a function of factors used to approximate $N_\Delta(T)$ based on $N_C(T)$, those unaffected to date.

Note that the number of test cases is used since that is a measured number and should add accuracy by reducing the number being estimated. However, this should represent a measure that is independent of the number of tests performed on the population. Instead, there is a ratio of number of tests to the population, or that part of the population that is not immune if these can be tested again.

Some of these factors may be a function of the day of the week. Some must be expanded into multiple factors and coefficients. For example, weather is a function of temperature, wind and moisture. Gender and average age must be split.

Elements within the functions and the coefficients (Cs) can be determined using the VisiSoft Optimization system for each municipality. One starts by developing generic values for the specific functions based on multiple municipalities, i.e., by county. Given those for the county, one can optimize the individual C coefficients for each municipality.

Achieving Accurate Predictions

Prediction is presented in terms of an envelope described by the circles from day 1 to 19 in April in Figure 8. Because of the increasing probability of error, these envelopes expand with future time. Envelopes are defined in terms of the probability of being inside. PSI typically uses an 80% envelope, implying that *the probability of being inside an envelope is 80%.*

The probability statement must be backed by a Confidence Level. PSI uses a 95% Confidence Level, i.e., *the probability of being inside the envelopes 80% of the time is 95%.* The Confidence Level must be derived from many prior predictions over time. When producing daily prediction envelopes, checking them over the course of a year is generally more than sufficient. And there is good reason for using a shorter period to ensure that more recent probabilities are still good. Checking them over a 100 day period provides 100 separate confidence tests, a reasonable number. To achieve a 95% confidence in the envelopes, one must meet the 80% criteria 95 days out of the last 100 for each prediction step.

Modeling Distributed Responses To Events

When modeling populations of elements of nature, one must face the fact that all elements or individuals do not produce the same response to an event, and if they do, it is not produced at the same time. Instead, responses are typically characterized by distributions in time and state space. For example, When people are first infested with the COVID-19, it typically takes from 1 to 2 weeks for them to become aware of it. However, if tested during that period, it will show positive. Once aware that they have it and are tested, their response to this well defined event produces a distribution of follow-on events that can be quite varied in their actions as well as their times of occurrance.

It is this delayed behavior that produces inherent “unpredictability” in a system. However, using the approach described below, one can model these types of distributed responses directly as they affect the behavior of a population, obtaining substantially improved accuracy. Such models are created relatively easily using VisiSoft.

We start with a general example of distributed responses to a sequence of events to show how the resulting delayed cumulative response can be modeled quite easily with reasonable accuracy. This is useful in predicting responses to events that occurred many time steps in the past. To demonstrate this, we illustrate a simple model of CORONA-19 as a function of the initial event of infections.

Modeling Inertial Subsystems

To demonstrate the significance of incorporating "leading factor" driving forces into a model, we offer an example which is representative of many actual cases. Let U(T) be the driving force (Figure 9), and let Z(T) be the output response of the system, (Figure 10). In this example, both are observable at discrete time points T. Figure 10 represents an example of two exponential response functions as used in engineering. TD1 and TD2 are delay times measured from the input impulses U1 and U2. TD1 is the time before the first exponential starts to rise (positive). TD2 is the time before the second exponential starts to rise. TAU1 and TAU2 are the rise and fall time constants for these exponentials. These same delay times and exponentials are applied to all succeeding inputs. The size of the impulses determines the output sizes.

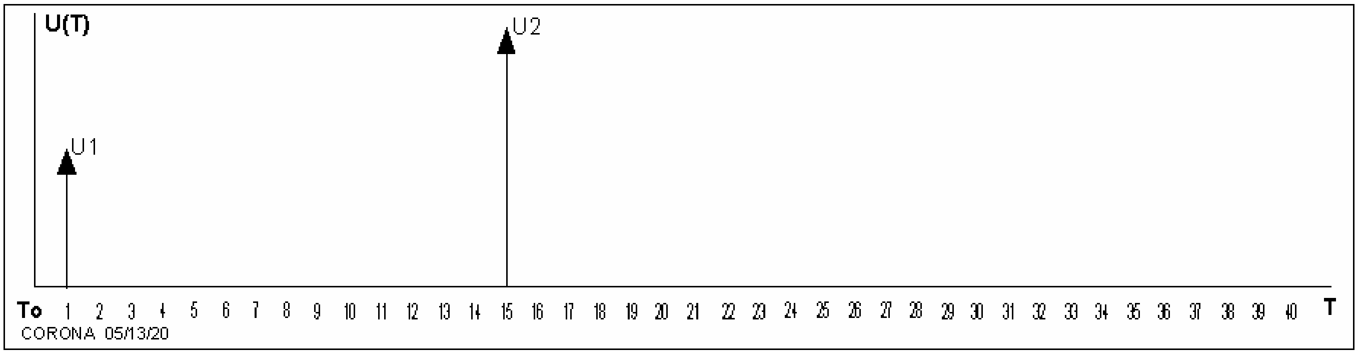


Figure 9. Driving force input.

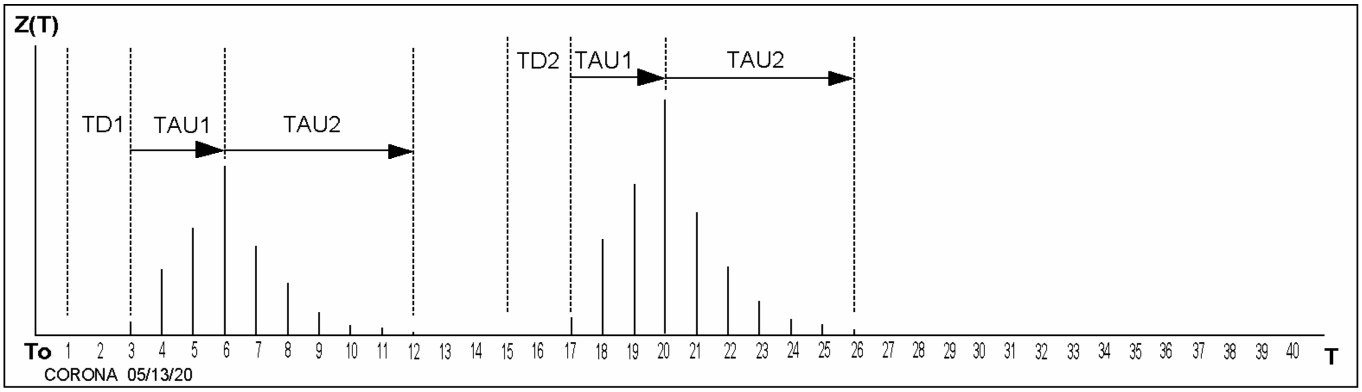


Figure 10. System response.

Impulses U at times T cause inertial properties within the system to react over time. These reactions are represented by exponential rise and fall times with time constants TAU1 and TAU2. They are then superimposed using linear superposition. Without being able to model these inertial effects, their sufficiently long time constants and their superposition, one cannot hope to predict the future beyond a single time step.

As the figures indicate, when an impulse occurs at T = 1, a response does not occur until the 3rd time step. The information, that an impulse has occurred, can be derived from the response data up to T = 12. A similar response occurs based on the input at T = 15. All of these responses may be superimposed.

Assuming the model in Figure 10 represents the system with sufficient accuracy, we could predict with little error up to 11 time steps into the future. Furthermore, when the input appears to be purely random, so does the response; but this does not preclude us from making perfect predictions of the response up to 11 time steps in the future.

Testing has shown that the delay between actual contraction of the virus and recognizing it may take from 2 to 14 days. Thus we can use 2 days for TD1 and then 14 days for TAU1 + TAU2, a total of 16 days. We have cut TAU2 because of the very small numbers at the end of the time constant. This particular distribution provides predictions for 11 days out. We note the difficulty in sample testing to determine these parameters. Instead, they must be estimated using a detailed model as described here.

Note also that some fraction of the population (typically younger people) may never know they have it if not tested. This latter group represents a significant portion of all those who have it, and must be included in the predictions because they may be causing a major part of the spread. Only a portion of those are tested and become part of the data on number of cases reported. This implies that the predictions of all those who have it have no direct measure unless a sufficient number of random samples are taken that tests for positive cases to estimate the percentage of total cases.

The incubation period for COVID-19 is thought to extend from 2 to 14 days, with a median time of 4 to 5 days from exposure to symptoms onset. One study reports that 97.5% of persons with COVID-19 who develop symptoms will do so within 11.5 days of SARS-CoV-2 infection. Early estimates predict that the overall COVID-19 Recovery Rate is between 97% and 99.75%; and also that Deaths = .063 * positive tests and Recoveries = 97 to 99.75% of tested positives within 3 to 6 weeks.

TRANSLATING TEST CASES INTO ACTUAL CASES AND BACK

The test cases defined by the CORONAVIRUS medical community indicate that a test case determined positive on day 12 (or 26) as shown in Figure 11 has a probability distribution of having been infected between days 1 and 10 (or 15 and 24) days back as shown in Figure 12. Because the number of published “TESTED CASES” are accumulated up to the day published, one must take the difference between tested cases on two subsequent days to determine the number of cases tested positive on the later day.

Relation Between Actual Cases And Test Cases

To determine the ACTUAL CASES for a given day - of which only 20% (1/5th) of these are sufficiently severe to cause testing - one must start with the tested (positive) infections. One must then multiply the tested cases by a factor of 5 to get the total actual cases causing the resulting tests. One must then spread that amount back over time using the distribution shown in Figure 12 below which provides the probability distribution of contraction on those days in the distribution. Having spread the number of actuals according to this distribution, the sum of the distributed actual cases will be 5 time greater than the tested cases used for the distribution. Each time a new day of test cases is spread, it adds to the back distribution starting a day later.

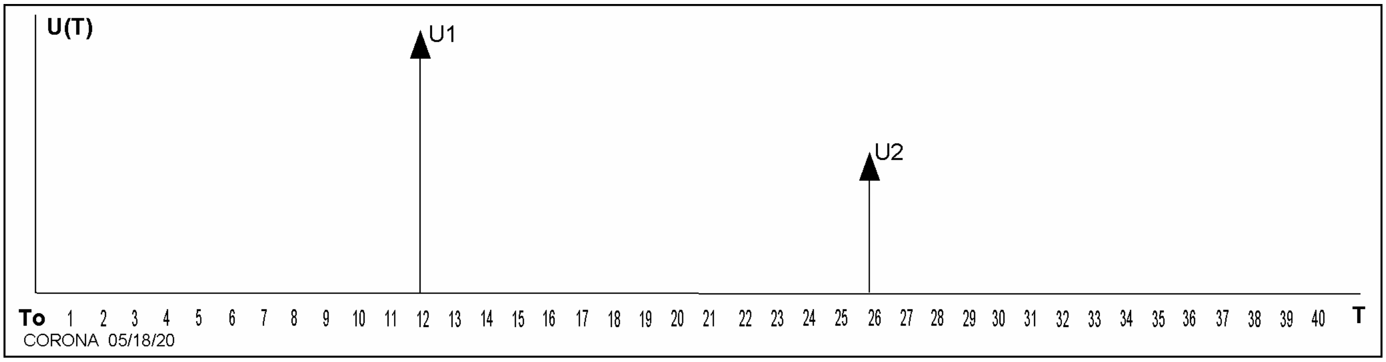


Figure 11. Test cases at T = 12 and T = 26.

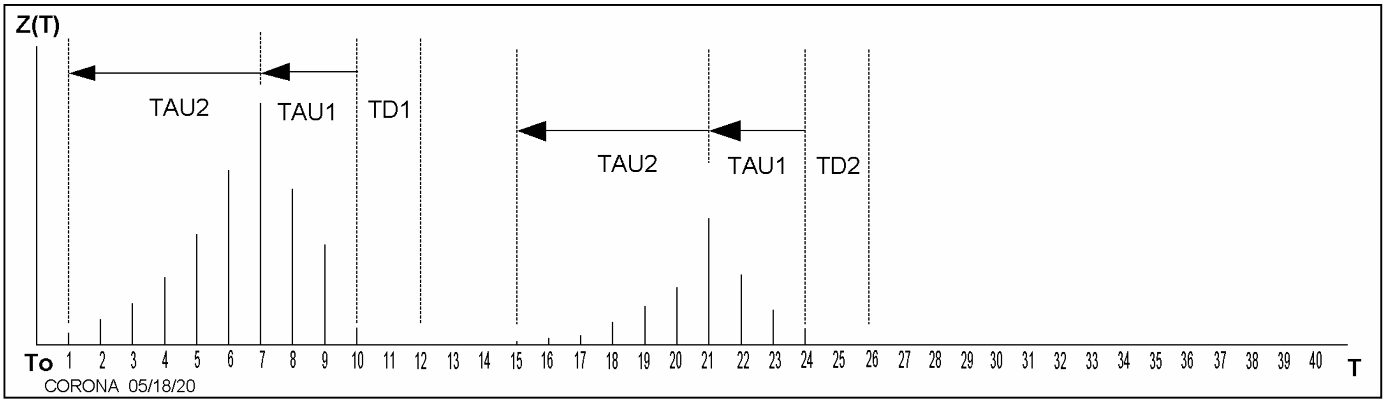


Figure 12. Back distribution of potential infections of test cases at T = 12 and T = 26.

Translating Back To Test Cases From Day 1 Of Actual Cases

Starting on day 1 of the actual case distribution illustrated in Figure 12, one must translate those actual cases into test cases. This implies that one must reduce the actuals by a factor of 5 and then distribute them into the future as illustrated in Figure 9 above. If there are less than 5 actual cases on a given day, they should be accumulated separately while moving on to the next day. When these accumulated cases add to 5, then a single case must be distributed ahead to the test day assuming that the contraction day was in the middle of the contributing days.

Actual Case Recovery Periods

People with mild cases (80% of actuals) recover in about 2 weeks. They are the majority (by a factor of 4) and generally have no tests. Therefore, they must be subtracted from the actuals within the two week period after contracting the virus. This will reduce the daily actuals of cases incurred 14 days back. People with more severe or critical cases (20% actuals) recover within 3 to 6 weeks (tested). These people must also be subtracted from more recent actuals that are still infected and can spread it. All recoveries are immune from future contractions.

The insertion of actual cases based on the back distribution can provide a reasonable estimate of the total cases that existed on each day back. However it does not provide any indication of when those people were infected. To determine when the infections were incurred, one must go back to the first day of stored infections - the starting point of the distribution - and assume that this is the first day for people being infected on that day. Those added after that, i.e., any increases that follow, will be the first day for those providing the increase, etc. Thus, then 80% of these people can be removed after two weeks.

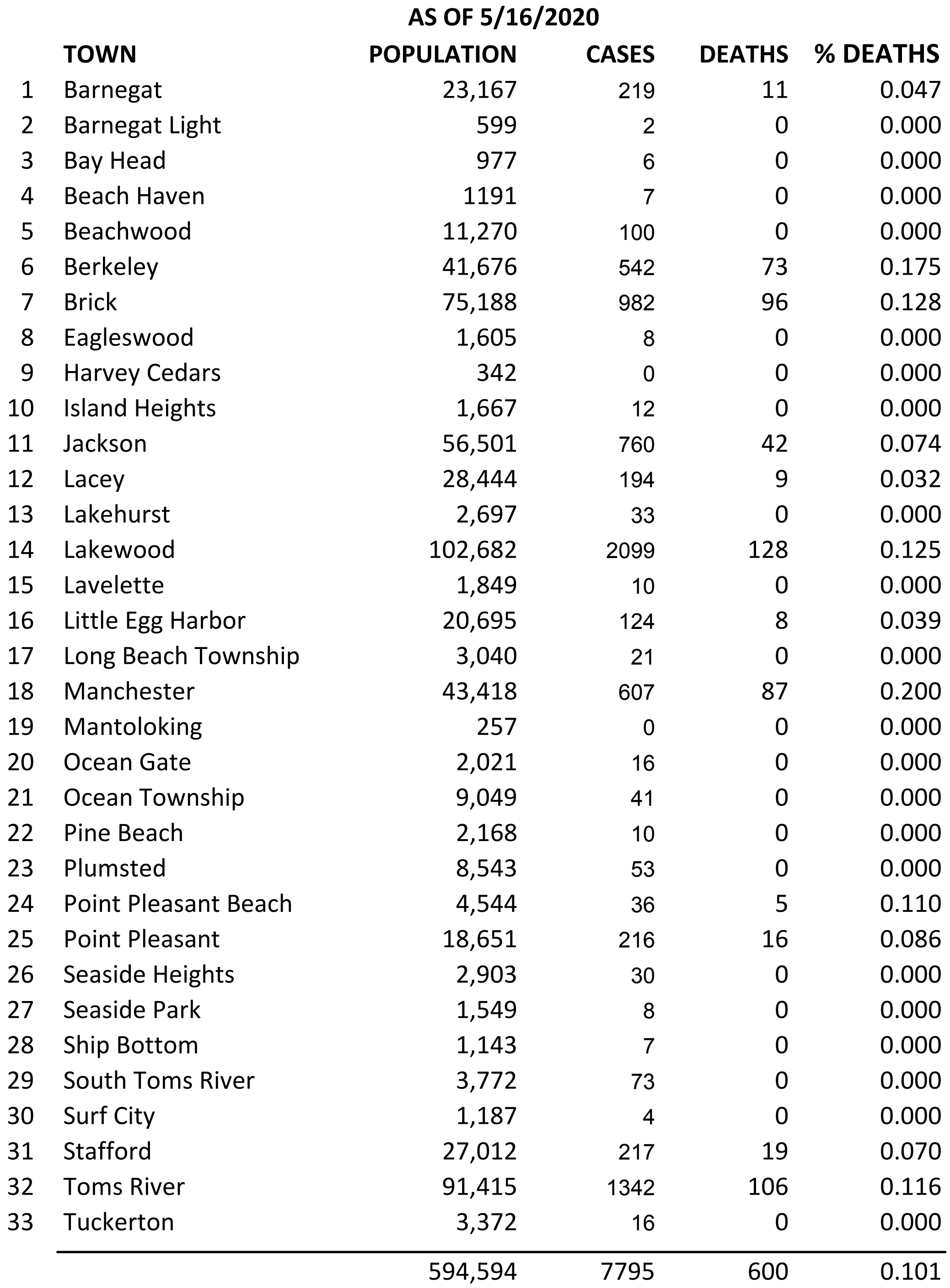
Deaths

As another example of the need to examine data at the municipal level, consider the numbers in Figure 13 showing COVID-19 deaths in Ocean County, NJ. Upon dividing the total deaths in the county (600) by the population one gets very close to 0.1%. Only 12 municipalities out of 34 have deaths (22 out of the 34 have no deaths) yielding an average of these percentages at 0.101% of the cases (due to all the 0's). Because the percentage of deaths is so small for individual municipalities, it will be excluded from the change in actuals for a municipality.

CONTROLLING DESIRED BEHAVIORAL OUTCOMES

How does one come up with the best advice to people when changing government positions on behavior and other factors, especially when trying to loosen up the behavioral constraints. This is a real-time control problem, hardly different from guiding missiles. The first element is having accurate data on the current state of the system. This requires accurate estimates of the factors affecting the outcome as well as their use to directly determine the outcomes. For example, one must have an estimate of the state of behavior of people in a given municipality. How much change can be made without throwing the forward movement off the desired path? Depending upon the change, the virus may start back up. Predicting these outcomes can only be done at the municipal level.

Using the prediction system described here, outcomes based on the changes that political leaders want to make can be optimized to maximize the probability that the future direction follows the desired course. As described in examples in Chapter 13, this is hardly different from controlling a guided missile in a windy environment. We must emphasize that, based on much prior experience, accurate control of complex systems depends heavily on accurate predictions of where they are headed given different choices of controls. PSI has obtained sufficient data to test and prove the accuracy of its municipal model, and expects to be able to offer a copy of the PSI model to all counties in the U.S. to run it daily for each of their municipalities. This will require training to use the optimization facilities described in Chapter 9 to track and optimize the changing factors affecting each municipality.



AS OF 5/16/2020

	TOWN	POPULATION	CASES	DEATHS	% DEATHS
1	Barnegat	23,167	219	11	0.047
2	Barnegat Light	599	2	0	0.000
3	Bay Head	977	6	0	0.000
4	Beach Haven	1191	7	0	0.000
5	Beachwood	11,270	100	0	0.000
6	Berkeley	41,676	542	73	0.175
7	Brick	75,188	982	96	0.128
8	Eagleswood	1,605	8	0	0.000
9	Harvey Cedars	342	0	0	0.000
10	Island Heights	1,667	12	0	0.000
11	Jackson	56,501	760	42	0.074
12	Lacey	28,444	194	9	0.032
13	Lakehurst	2,697	33	0	0.000
14	Lakewood	102,682	2099	128	0.125
15	Lavelette	1,849	10	0	0.000
16	Little Egg Harbor	20,695	124	8	0.039
17	Long Beach Township	3,040	21	0	0.000
18	Manchester	43,418	607	87	0.200
19	Mantoloking	257	0	0	0.000
20	Ocean Gate	2,021	16	0	0.000
21	Ocean Township	9,049	41	0	0.000
22	Pine Beach	2,168	10	0	0.000
23	Plumsted	8,543	53	0	0.000
24	Point Pleasant Beach	4,544	36	5	0.110
25	Point Pleasant	18,651	216	16	0.086
26	Seaside Heights	2,903	30	0	0.000
27	Seaside Park	1,549	8	0	0.000
28	Ship Bottom	1,143	7	0	0.000
29	South Toms River	3,772	73	0	0.000
30	Surf City	1,187	4	0	0.000
31	Stafford	27,012	217	19	0.070
32	Toms River	91,415	1342	106	0.116
33	Tuckerton	3,372	16	0	0.000
		594,594	7795	600	0.101

Figure 13. Deaths Relative to Population in Ocean County, NJ

A BRIEF REVIEW OF IMPORTANT MODEL ENTITIES

The following provides a summary of definitions used in the model. Note the difference between cumulative model entities and those for a given day.

Tested Cases

TESTED_CASES(T) are a cumulative measure of those considered *severe* versus mild. This generally implies that people who anticipated having the virus came to test facilities to be tested, and were tested positive. CASES_TESTED(T) represents those for a given day, i.e., the daily difference.

Mild Cases

Mild cases are generally unreported because they are typically not recognized or simply ignored by the affected person and not tested. MILD_CASES (T) are statistically measured to be 4 times that of TESTED_CASES(T) which is cumulative. CASES_MILD (T) is not cumulative.

Total Recoveries

TOTAL_RECOVERIES(T) are the cumulative total of those who have had it and recovered. Mild cases are statistically measured to recover within two weeks. Severe (tested) cases are statistically characterized and their removal is distributed over 3 to 6 weeks.

Deaths

TOTAL_DEATHS(T) are cumulative deaths up to T. The percentage of TOTAL_DEATHS(T) is on the order of 10% of the TESTED_CASES(T) (and on the order of 0.10% of the population). Because deaths are small, they are not tracked directly. However, they are included in the change in actuals for a municipality as described below.

Actual Cases

ACTUAL_CASES(T) equals the sum:

TESTED_CASES(T) + MILD_CASES(T) - TOTAL_RECOVERIES(T),

a cumulative measure for a given day. Because ACTUAL_CASES(T) are 5 times larger than TESTED_CASES(T), and deaths are replaced by increases in population, deaths are not subtracted. Even though TOTAL_RECOVERIES(T) are small for a give day, they are cumulative and become a significant factor in ACTUAL_CASES(T). By ignoring deaths, they are left in the ACTUAL_CASES(T) and therefore removed from the future actuals along with the TOTAL_RECOVERIES(T),

16. PREDICTION - A BRIEF SUMMARY

We wish to note the complexity involved in defining and solving the general prediction problem. One is typically trying to predict a vector of observable responses out to some maximum number of time steps (horizon) into the future for which predictions are required. Typically, complex systems are neither linear, homogeneous, nor stationary, so that an understanding of the "mechanics" of the system is necessary to approach such a problem. In practice, one must comprehend these mechanics in order to postulate candidate driving force vectors, and then model these mechanics to produce the transformations that relate future values of the response to the driving forces. To accomplish this, one must define the complex spaces illustrated in Figure 1-2. Software implementation requires the ability to represent the complex hierarchies required to define these spaces. It is also necessary to define meaningful distance measures to maximize prediction accuracy (minimize prediction error).

To summarize the results described in the previous chapters, the following tabular comparison is offered.

History Data	**Future Data**
$Z(1), \ldots, Z(T)$	$Z(T+1), \ldots, Z(T+\tau)$
Modeling (Estimation) Error	**Prediction Error**
$\hat{e}^{+}(C, U, Z)$	$\hat{e}^{-}(C, U, Z)$
$\hat{e}^{+}[\hat{Z}(T\|T), Z(T)]$	$\hat{e}^{-}[\hat{Z}(T+\tau\|T), Z(T+\tau)]$

Referring to the comparisons above, modeling (estimation) error can be measured using a model conditioned on all data up to and including the final measurement time, T. In the case of prediction error, the dynamic model, which is part of the error function, can only be conditioned on information up to the current time, T, which is τ steps back from the final measurement.

When optimizing model parameters to reduce prediction error, a correlation must exist between $\hat{e}^{-}$ and $\hat{e}^{+}$, to ensure that reducing modeling error implies reducing prediction error. Else, the modeler has no criteria for improving a model. It is clear that determination of this correlation can involve substantial amounts of hidden data in order to ensure that the correlation test uses true prediction error, i.e., *it is based on data the modeler has not yet seen*.

If one simply uses a naive function to fit the history data, it is doubtful that the properties of the system will be discovered, no matter how powerful the mathematical techniques used to identify or optimize the curve fitting parameters. However, if a modeler builds a structural model based on an understanding of the mechanics of the system, he need only use the data to validate his model and measure prediction accuracy. Furthermore, the likelihood of correlation, between modeling error and prediction error, will be much higher.

17. REFERENCES

1. Athans, M. and Falb, P.L., *Optimal Control*, McGraw-Hill, New York, 1966.

2. Athans, M. and Kendrick, D., "Control Theory and Economics: A Survey, Forecast and Speculation", IEEE Trans, Automatic Control, pp. 518-524, Oct. 1974.

3. Black, Fisher and Myron Sholes, *The Pricing of Options and Corporate Liabilities*, Journal of Political Economy, Vol.81, No. 3, 1973.

4. Box, G. and Jenkins, G., *Time Series Analysis*, Holden Day, San Francisco, CA, 1976

5. Cave, W. C., "An Automated Design formulation for Integrated Circuits," Automated Integrated Circuit Design, G. Herskowitz, Ed., McGraw Hill, NJ, 1969.

6. Cave, W. C. , "Computer Optimization of Microwave Integrated Circuits," IEEE Microwave Integrated Circuits Seminar Proceedings, Monmouth College, NJ, June 1970.

7. Cave, W.C., "The Constrained Optimal Design System," Proceedings IEEE WESCON, San Francisco, CA, 1971.

8. Cave, W.C. and Rosenkranz, E., "A Stochastic State Space Model for Prediction of Product Demand," AFIPS-Conference Proceedings, Volume 48, AFIPS Press, pp. 67-72, 1979.

9. Cave, W.C. and Guilfoyle, R.H., "A General Approach to Supply/Demand Market Prediction Using the State Space Framework". First International Forecasting Symposium, Quebec, CD, 1982.

10. Cave, W.C., *Simulation of Complex Systems*, Prediction Systems, Inc., Spring Lake, NJ, February 2006.

11. Fisher, R.A., "On the Mathematical Foundations of Theoretical Statistics", Phil. Trans, Royal Soc., London, 222, 1922, 309.

12. Friedman, B., *Principles and Techniques of Applied Mathematics*, John Wiley & Sons, New York, pp. 169, 1956.

13. Gear, C.W., "The Automatic Integration of Stiff Ordinary Differential Equations", Proc. 1968 IFIPS Congr., pp. A81-A85.

14. Gelb, A., Editor, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

15. *The General Simulation System (GSS) - User's Reference Manual*, Version 10.5, Visual Software International, Spring Lake, NJ, 2005.

16. Harrison, P.J. and Stevens, C.F., "A Bayesian approach to short-term forecasting", Oper. Res. Quart. Vol. 22, pp. 341-362, 1971.

17. Jazwinski, A., *Stochastic Process and Filtering Theory*, Academic Press, New York, NY 1970.

18. Jenkins, G. and Watts, D., *Spectral Analysis and Applications*, Holden Day, San Francisco, CA, 1969.

19. Kalman, R., "A New Approach to Linear Filtering and Prediction Problems", Trans. ASME, Series D: Journal Basic Engineering 82, pp. 35-45, 1960.

20. Kalman, R., Keynote Address, Third International Symposium on Forecasting Program, Philadelphia, PA., pp. 150, 1983.

21. Mehra, R., "Kalman Filters and Their Applications to Forecasting", TIMS Studies in the Management Sciences 12, North-Holland, 1979.

22. Nordsieck, A., "On Numerical Integration of Ordinary Differential Equations", Math. Comput. Vol. 16, pp. 22-49, 1962.

23. Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, NY pp. 554, 1965.

24. Ramadge, P.J. and W.M. Wonham, "Supervisory Control of a class of discrete-event processes," SIAM J. Control Optimization, vol 25, no.1, pp 206-230, Jan. 1987.

25. Ramadge, P.J. and W.M. Wonham, "The Control Of Discrete-Event Systems," Proc. IEEE, vol 77, no.1, pp 206-230, Jan. 1989.

26. Rosenkranz, E., "Robust Estimation of Time Varying Parameters", Third International Symposium on Forecasting Program, Philadelphia, PA., pp. 150, 1983.

27. Schweppe, F., *Uncertain Dynamic Systems*, Prentice Hall, Englewood, NJ, 1973.

28. Tikhonov, A.N. and Samarski, A.A., *Equations of Mathematical Physics*, the MacMillan Co., NY, pp. 97, 1963.

29. Tukey, J.W., "Discussion emphasizing the connection between analysis of variance and spectrum analysis", Technometrics, 3, 191, 1961.

30. Weinberg, G., *An Introduction to General Systems Thinking*, John Wiley & Sons, NY 1975.

31. Zadeh, L.A. and Desoer, C.A., Linear System Theory: The State Space Approach, McGraw-Hill, NY 1963.

32. A Day Without Space (DWoS) – Initial Analysis, Final Report, GCIC, Langley AFB, Contract FA8750-07-D-0027, Prediction Systems, Inc, Spring Lake, NJ, Apr 2010

33. Placement of Sensing and Communications Platforms for Enhanced C4ISR Operations - Phase II Final Report, SPAWAR, San Diego, Contract N00039-08-C-0037, Prediction Systems, Inc, Spring Lake, NJ, Dec 2009

34. Weapons Data Link (WDL) - Link 16 Analysis – Final Report, AFRL/WPAFB, Contract F33615-00-C-1666, Prediction Systems, Inc, Spring Lake, NJ, Mar 2006

35. Network Enabled Weapons (NEW) - CNR Analysis – Final Report, USAF 685 ARSS, Eglin AFB, Contract FA8750-07-D-0027, Prediction Systems, Inc, Spring Lake, NJ, Jun 2010.

36. Donald Anselmo and Henry Ledgard, "Measuring Productivity in The Software Industry," *Communications of the ACM*, vol. 46. no.11, Nov. 2003.

PREDICTION SYSTEMS, INC.
PREDICTION & CONTROL SYSTEMS ENGINEERS

309 Morris Avenue
Spring Lake, NJ 07762

☎ (732)449-6800 🗎 (732) 449-0897
📫 PSI@predictsys.com 🕸 www. predictsys.com