Basics of Engineering Statistics

Clair Moeller

First Edition, 2012

ISBN 978-81-323-4089-8

Published by:
White Word Publications
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter 1

Design of Experiments

In general usage, **design of experiments (DOE)** or **experimental design** is the design of any information-gathering exercises where variation is present, whether under the full control of the experimenter or not. However, in statistics, these terms are usually used for controlled experiments. Other types of study, and their design, are discussed in opinion polls and statistical surveys (which are types of observational study), natural experiments and quasi-experiments (for example, quasi-experimental design).

In the design of experiments, the experimenter is often interested in the effect of some process or intervention (the "treatment") on some objects (the "experimental units"), which may be people, parts of people, groups of people, plants, animals, materials, etc. Design of experiments is thus a discipline that has very broad application across all the natural and social sciences.

History of development

Controlled experimentation on scurvy

In 1747, while serving as surgeon on HM Bark *Salisbury*, James Lind carried out a controlled experiment to develop a cure for scurvy.

Lind selected 12 men from the ship, all suffering from scurvy. Lind limited his subjects to men who "were as similar as I could have them", that is he provided strict entry requirements to reduce extraneous variation. He divided them into six pairs, giving each pair different supplements to their basic diet for two weeks. The treatments were all remedies that had been proposed:

- A quart of cider every day
- Twenty five gutts (drops) of *elixir vitriol* (sulphuric acid) three times a day upon an empty stomach,
- One half-pint of seawater every day
- A mixture of garlic, mustard, and horseradish in a lump the size of a nutmeg
- Two spoonfuls of vinegar three times a day

- Two oranges and one lemon every day.

The men who had been given citrus fruits recovered dramatically within a week. One of them returned to duty after 6 days and the other cared for the rest. The others experienced some improvement, but nothing was comparable to the citrus fruits, which were proved to be substantially superior to the other treatments.

Statistical experiments, following Charles S. Peirce

A theory of statistical inference was developed by Charles S. Peirce in "Illustrations of the Logic of Science" (1877–1878) and "A Theory of Probable Inference" (1883), two publications that emphasized the importance of randomization-based inference in statistics.

Randomized experiments

Charles S. Peirce randomly assigned volunteers to a blinded, repeated-measures design to evaluate their ability to discriminate weights. Peirce's experiment inspired other researchers in psychology and education, which developed a research tradition of randomized experiments in laboratories and specialized textbooks in the 1800s.

Optimal designs for regression models

Charles S. Peirce also contributed the first English-language publication on an optimal design for regression-models in 1876. A pioneering optimal design for polynomial regression was suggested by Gergonne in 1815. In 1918 Kirstine Smith published optimal designs for polynomials of degree six (and less).

Sequences of experiments

The use of a sequence of experiments, where the design of each may depend on the results of previous experiments, including the possible decision to stop experimenting, is within the scope of Sequential analysis, a field that was pioneered by Abraham Wald in the context of sequential tests of statistical hypotheses. Herman Chernoff wrote an overview of optimal sequential designs, while adaptive designs have been surveyed by S. Zacks. One specific type of sequential design is the "two-armed bandit", generalized to the multi-armed bandit, on which early work was done by Herbert Robbins in 1952.

Principles of experimental design, following Ronald A. Fisher

A methodology for designing experiments was proposed by Ronald A. Fisher, in his innovative book *The Design of Experiments* (1935). As an example, he described how to test the hypothesis that a certain lady could distinguish by flavour alone whether the milk or the tea was first placed in the cup. While this sounds like a frivolous application, it allowed him to illustrate the most important ideas of experimental design:

Comparison

In many fields of study it is hard to reproduce measured results exactly. Comparisons between treatments are much more reproducible and are usually preferable. Often one compares against a standard, scientific control, or traditional treatment that acts as baseline.

Randomization

There is an extensive body of mathematical theory that explores the consequences of making the allocation of units to treatments by means of some random mechanism such as tables of random numbers, or the use of randomization devices such as playing cards or dice. Provided the sample size is adequate, the risks associated with random allocation (such as failing to obtain a representative sample in a survey, or having a serious imbalance in a key characteristic between a treatment group and a control group) are calculable and hence can be managed down to an acceptable level. Random does *not* mean haphazard, and great care must be taken that appropriate random methods are used.

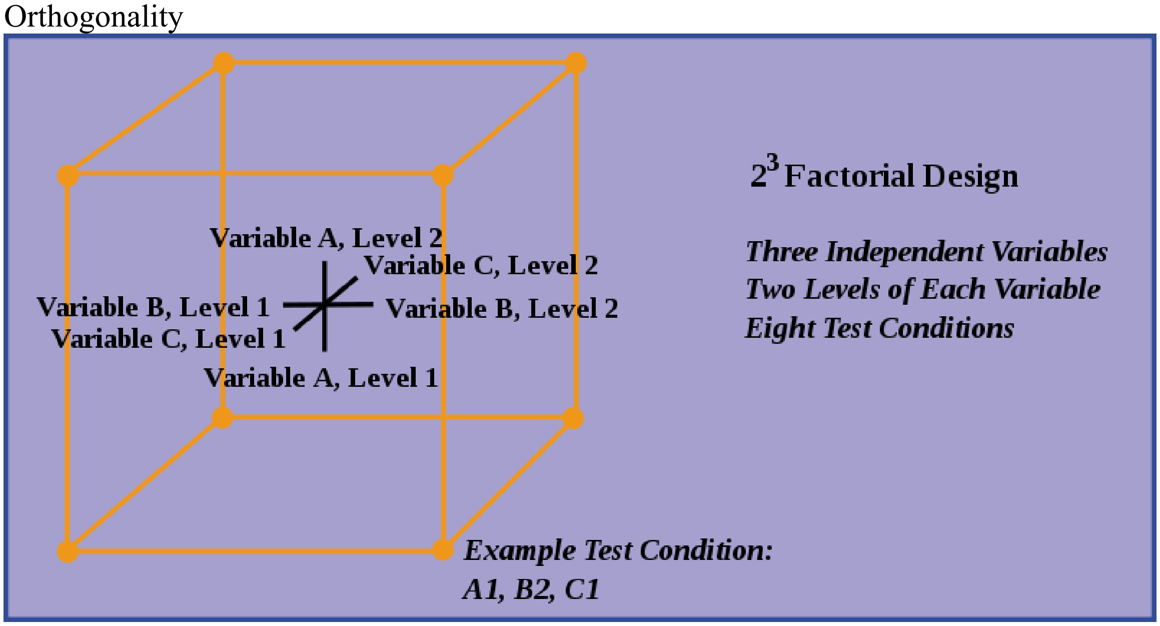
Replication

Measurements are usually subject to variation and uncertainty. Measurements are repeated and full experiments are replicated to help identify the sources of variation and to better estimate the true effects of treatments.

Blocking

Blocking is the arrangement of experimental units into groups (blocks) consisting of units that are similar to one another. Blocking reduces known but irrelevant sources of variation between units and thus allows greater precision in the estimation of the source of variation under study.

Orthogonality



Example of orthogonal factorial design

Orthogonality concerns the forms of comparison (contrasts) that can be legitimately and efficiently carried out. Contrasts can be represented by vectors and sets of orthogonal contrasts are uncorrelated and independently distributed if the data are normal. Because of this independence, each orthogonal treatment provides different information to the others. If there are T treatments and $T - 1$ orthogonal contrasts, all the information that can be captured from the experiment is obtainable from the set of contrasts.

Factorial experiments

Use of factorial experiments instead of the one-factor-at-a-time method. These are efficient at evaluating the effects and possible interactions of several factors (independent variables).

Analysis of the design of experiments was built on the foundation of the analysis of variance, a collection of models in which the observed variance is partitioned into components due to different factors which are estimated and/or tested.

Example

This example is attributed to Harold Hotelling. It conveys some of the flavor of those aspects of the subject that involve combinatorial designs.

The weights of eight objects are to be measured using a pan balance and set of standard weights. Each weighing measures the weight difference between objects placed in the left pan vs. any objects placed in the right pan by adding calibrated weights to the lighter pan until the balance is in equilibrium. Each measurement has a random error. The average error is zero; the standard deviations of the probability distribution of the errors is the same number σ on different weighings; and errors on different weighings are independent. Denote the true weights by

$$\theta_1, \ldots, \theta_8.$$

We consider two different experiments:

1. Weigh each object in one pan, with the other pan empty. Let X_i be the measured weight of the ith object, for $i = 1, ..., 8$.
2. Do the eight weighings according to the following schedule and let Y_i be the measured difference for $i = 1, ..., 8$:

	left pan	right pan
1st weighing:	1 2 3 4 5 6 7 8	(empty)
2nd:	1 2 3 8	4 5 6 7
3rd:	1 4 5 8	2 3 6 7
4th:	1 6 7 8	2 3 4 5
5th:	2 4 6 8	1 3 5 7
6th:	2 5 7 8	1 3 4 6
7th:	3 4 7 8	1 2 5 6
8th:	3 5 6 8	1 2 4 7

Then the estimated value of the weight θ_1 is

$$\widehat{\theta}_1 = \frac{Y_1 + Y_2 + Y_3 + Y_4 - Y_5 - Y_6 - Y_7 - Y_8}{8}.$$

Similar estimates can be found for the weights of the other items. For example

$$\widehat{\theta}_2 = \frac{Y_1 + Y_2 - Y_3 - Y_4 + Y_5 + Y_6 - Y_7 - Y_8}{8}.$$

The question of design of experiments is: which experiment is better?

The variance of the estimate X_1 of θ_1 is σ^2 if we use the first experiment. But if we use the second experiment, the variance of the estimate given above is $\sigma^2/8$. Thus the second experiment gives us 8 times as much precision for the estimate of a single item, and estimates all items simultaneously, with the same precision. What is achieved with 8 weighings in the second experiment would require 64 weighings if items are weighed separately. However, note that the estimates for the items obtained in the second experiment have errors which are correlated with each other.

Many problems of the design of experiments involve combinatorial designs, as in this example.

Statistical control

It is best for a process to be in reasonable statistical control prior to conducting designed experiments. When this is not possible, proper blocking, replication, and randomization allow for the careful conduct of designed experiments.

Experimental designs after Fisher

Some efficient designs for estimating several main effects simultaneously were found by Raj Chandra Bose and K. Kishen in 1940 at the Indian Statistical Institute, but remained little known until the Plackett-Burman designs were published in *Biometrika* in 1946. About the same time, C. R. Rao introduced the concepts of orthogonal arrays as experimental designs. This was a concept which played a central role in the development of Taguchi methods by Genichi Taguchi, which took place during his visit to Indian Statistical Institute in early 1950s. His methods were successfully applied and adopted by Japanese and Indian industries and subsequently were also embraced by US industry albeit with some reservations.

In 1950, Gertrude Mary Cox and William Gemmell Cochran published the book *Experimental Designs* which became the major reference work on the design of experiments for statisticians for years afterwards.

Developments of the theory of linear models have encompassed and surpassed the cases that concerned early writers. Today, the theory rests on advanced topics in linear algebra, algebra and combinatorics.

As with other branches of statistics, experimental design is pursued using both frequentist and Bayesian approaches: In evaluating statistical procedures like experimental designs, frequentist statistics studies the sampling distribution while Bayesian statistics updates a probability distribution on the parameter space.

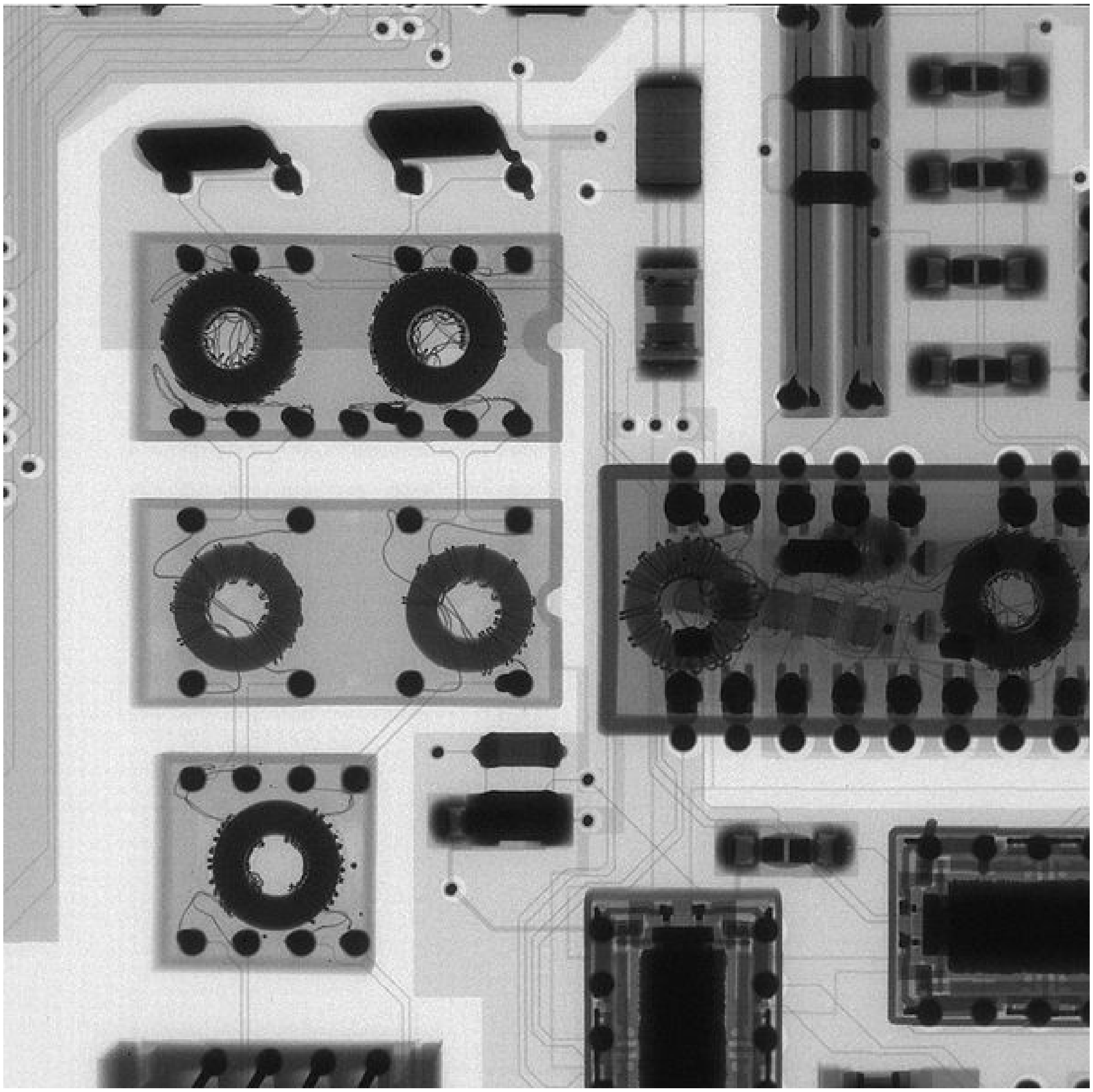
Some important contributors to the field of experimental designs are C. S. Peirce, R. A. Fisher, F. Yates, C. R. Rao, R. C. Bose, J. N. Srivastava, Shrikhande S. S., D. Raghavarao, W. G. Cochran, O. Kempthorne, W. T. Federer, A. S. Hedayat, J. A. Nelder, R. A. Bailey, J. Kiefer, W. J. Studden, F. Pukelsheim, D. R. Cox, H. P. Wynn, A. C. Atkinson, G. E. P. Box and G. Taguchi. The textbooks of D. Montgomery and R. Myers have reached generations of students and practitioners.

Chapter 2

Quality Control and Process Control

Quality control

Maintenance check of electronic equipment on a U.S. Navy aircraft.



X-ray zoom series of a network adapter card.

Quality control is a process by which entities review the quality of all factors involved in production. This approach places an emphasis on three aspects:

1. Elements such as controls, job management, defined and well managed processes, performance and integrity criteria, and identification of records
2. Competence, such as knowledge, skills, experience, and qualifications
3. Soft elements, such as personnel integrity, confidence, organizational culture, motivation, team spirit, and quality relationships.

The quality of the outputs is at risk if any of these three aspects is deficient in any way.

Quality control emphasizes testing of products to uncover defects, and reporting to management who make the decision to allow or deny the release, whereas quality assurance attempts to improve and stabilize production, and associated processes, to avoid, or at least minimize, issues that led to the defects in the first place. For contract work, particularly work awarded by government agencies, quality control issues are among the top reasons for not renewing a contract.

Total quality control

"Total quality control" is a measure used in cases where sales decrease despite implementation of statistical quality control techniques or quality improvements. If the original specification does not reflect the correct quality requirements, quality cannot be inspected or manufactured into the product. For instance, the parameters for a pressure vessel should include not only the material and dimensions, but also operating, environmental, safety, reliability and maintainability requirements.

Quality control in project management

In project management, quality control requires the project manager and the project team to inspect the accomplished work to ensure it's alignment with the project scope. In practice, projects typically have a dedicated quality control team which focuses on this area.

Process control

Process control is a statistics and engineering discipline that deals with architectures, mechanisms and algorithms for maintaining the output of a specific process within a desired range.

For example, heating up the temperature in a room is a process that has the specific, desired outcome to reach and maintain a defined temperature (e.g. 20°C), kept constant over time. Here, the temperature is the **controlled variable**. At the same time, it is the **input variable** since it is measured by a thermometer and used to decide whether to heat or not to heat. The desired temperature (20°C) is the **setpoint**. The state of the heater (e.g. the setting of the valve allowing hot water to flow through it) is called the **manipulated variable** since it is subject to control actions.

A commonly used control device called a programmable logic controller, or a PLC, is used to read a set of digital and analog inputs, apply a set of logic statements, and generate a set of analog and digital outputs. Using the example in the previous paragraph, the room temperature would be an input to the PLC. The logical statements would compare the setpoint to the input temperature and determine whether more or less heating was necessary to keep the temperature constant. A PLC output would then either open or close the hot water valve, an incremental amount, depending on whether more or less hot water was needed. Larger more complex systems can be controlled by a Distributed Control System (DCS) or SCADA system.

In practice, process control systems can be characterized as one or more of the following forms:

- Discrete – Found in many manufacturing, motion and packaging applications. Robotic assembly, such as that found in automotive production, can be characterized as discrete process control. Most discrete manufacturing involves the production of discrete pieces of product, such as metal stamping.

- Batch – Some applications require that specific quantities of raw materials be combined in specific ways for particular durations to produce an intermediate or end result. One example is the production of adhesives and glues, which normally require the mixing of raw materials in a heated vessel for a period of time to form a quantity of end product. Other important examples are the production of food, beverages and medicine. Batch processes are generally used to produce a relatively low to intermediate quantity of product per year (a few pounds to millions of pounds).

- Continuous – Often, a physical system is represented through variables that are smooth and uninterrupted in time. The control of the water temperature in a heating jacket, for example, is an example of continuous process control. Some important continuous processes are the production of fuels, chemicals and plastics. Continuous processes in manufacturing are used to produce very large quantities of product per year (millions to billions of pounds).

Applications having elements of discrete, batch and continuous process control are often called *hybrid* applications.

Statistical Process Control

Statistical Process Control (SPC) is an effective method of monitoring a process through the use of control charts. Much of its power lies in the ability to monitor both process center and its variation about that center. By collecting data from samples at various points within the process, variations in the process that may affect the quality of the end product or service can be detected and corrected, thus reducing waste as well as the likelihood that problems will be passed on to the customer. It has an emphasis on early detection and prevention of problems.

Multivariable Process Control is a type of Statistical Process Control where a matrix of variables (MV and CV) are created and their variations captured by doing a step test. The Dynamics captured in the model curves are used to control the plant

Examples

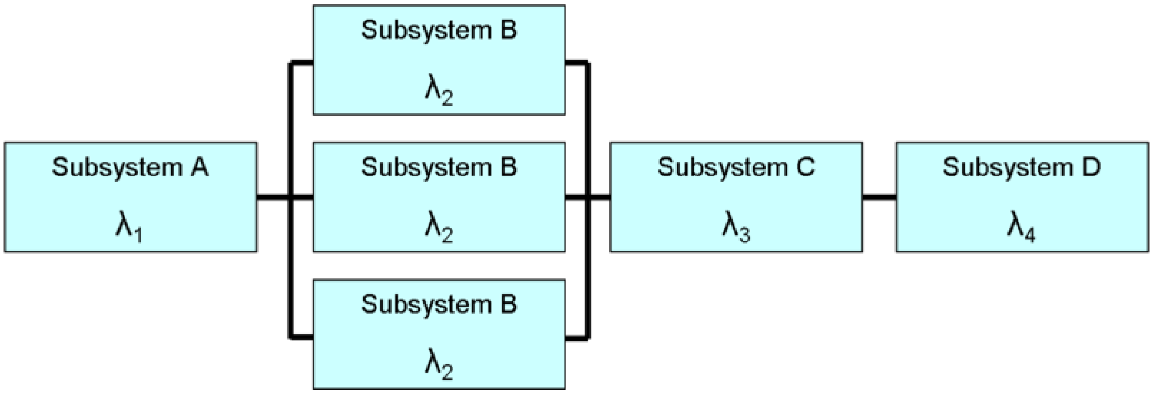
A thermostat is a simple example for a closed control loop: It constantly measures the current temperature and controls the heater's valve setting to increase or decrease the room temperature according to the user-defined setting. A simple method switches the heater either completely on, or completely off, and an overshoot and undershoot of the controlled temperature must be expected. A more expensive method varies the amount of heat provided by the heater depending on the difference between the required temperature (the "setpoint") and the actual temperature. This minimizes over/undershoot.

Chapter 3

Reliability Engineering

Reliability engineering is an engineering field, that deals with the study of reliability: the ability of a system or component to perform its required functions under stated conditions for a specified period of time. It is often reported as a probability.

Overview



A Reliability Block Diagram

Reliability may be defined in several ways:

- The idea that something is fit for a purpose with respect to time;
- The capacity of a device or system to perform as designed;
- The resistance to failure of a device or system;
- The ability of a device or system to perform a required function under stated conditions for a specified period of time;
- The probability that a functional unit will perform its required function for a specified interval under stated conditions.
- The ability of something to "fail well" (fail without catastrophic consequences)

Reliability engineers rely heavily on statistics, probability theory, and reliability theory. Many engineering techniques are used in reliability engineering, such as reliability prediction, Weibull analysis, thermal management, reliability testing and accelerated life testing. Because of the large number of reliability techniques, their expense, and the varying degrees of reliability required for different situations, most projects develop a reliability program plan to specify the reliability tasks that will be performed for that specific system.

The function of reliability engineering is to develop the reliability requirements for the product, establish an adequate reliability program, and perform appropriate analyses and tasks to ensure the product will meet its requirements. These tasks are managed by a reliability engineer, who usually holds an accredited engineering degree and has additional reliability-specific education and training. Reliability engineering is closely associated with maintainability engineering and logistics engineering, e.g. Integrated Logistics Support (ILS). Many problems from other fields, such as security engineering, can also be approached using reliability engineering techniques. Here we, provides an overview of some of the most common reliability engineering tasks.

Many types of engineering employ reliability engineers and use the tools and methodology of reliability engineering. For example:

- System engineers design complex systems having a specified reliability
- Mechanical engineers may have to design a machine or system with a specified reliability
- Automotive engineers have reliability requirements for the automobiles (and components) which they design
- Electronics engineers must design and test their products for reliability requirements.
- In software engineering and systems engineering the **reliability engineering** is the subdiscipline of ensuring that a system (or a device in general) will perform its intended function(s) when operated in a specified manner for a specified length of time. Reliability engineering is performed throughout the entire life cycle of a system, including development, test, production and operation.

Reliability theory

Reliability theory is the foundation of reliability engineering. For engineering purposes, reliability is defined as:

the probability that a device will perform its intended function during a specified period of time under stated conditions.

Mathematically, this may be expressed as,

$$R(t) = Pr\{T > t\} = \int_t^{\infty} f(x)\, dx,$$

where $f(x)$is the failure probability density function and *t* is the length of the period of time (which is assumed to start from time zero).

Reliability engineering is concerned with four key elements of this definition:

- First, reliability is a probability. This means that failure is regarded as a random phenomenon: it is a recurring event, and we do not express any information on individual failures, the causes of failures, or relationships between failures, except that the likelihood for failures to occur varies over time according to the given probability function. Reliability engineering is concerned with meeting the specified probability of success, at a specified statistical confidence level.
- Second, reliability is predicated on "intended function:" Generally, this is taken to mean operation without failure. However, even if no individual part of the system fails, but the system as a whole does not do what was intended, then it is still charged against the system reliability. The system requirements specification is the criterion against which reliability is measured.
- Third, reliability applies to a specified period of time. In practical terms, this means that a system has a specified chance that it will operate without failure before time t. Reliability engineering ensures that components and materials will meet the requirements during the specified time. Units other than time may sometimes be used. The automotive industry might specify reliability in terms of miles, the military might specify reliability of a gun for a certain number of rounds fired. A piece of mechanical equipment may have a reliability rating value in terms of cycles of use.
- Fourth, reliability is restricted to operation under stated conditions. This constraint is necessary because it is impossible to design a system for unlimited conditions. A Mars Rover will have different specified conditions than the family car. The operating environment must be addressed during design and testing. Also, that same rover, may be required to operate in varying conditions requiring additional scrutiny.

Reliability program plan

Many tasks, methods, and tools can be used to achieve reliability. Every system requires a different level of reliability. A commercial airliner must operate under a wide range of conditions. The consequences of failure are grave, but there is a correspondingly higher budget. A pencil sharpener may be more reliable than an airliner, but has a much different set of operational conditions, insignificant consequences of failure, and a much lower budget.

A reliability program plan is used to document exactly what tasks, methods, tools, analyses, and tests are required for a particular system. For complex systems, the reliability program plan is a separate document. For simple systems, it may be combined with the systems engineering management plan or integrated Logistics Support management plan. The reliability program plan is essential for a successful reliability program and is developed early during system development. It specifies not only what the reliability engineer does, but also the tasks performed by others. The reliability program plan is approved by top program management.

Reliability requirements

For any system, one of the first tasks of reliability engineering is to adequately specify the reliability requirements. Reliability requirements address the system itself, test and assessment requirements, and associated tasks and documentation. Reliability requirements are included in the appropriate system/subsystem requirements specifications, test plans, and contract statements.

System reliability parameters

Requirements are specified using reliability parameters. The most common reliability parameter is the mean time between failures (MTBF), which can also be specified as the failure rate or the number of failures during a given period. These parameters are very useful for systems that are operated frequently, such as most vehicles, machinery, and electronic equipment. Reliability increases as the MTBF increases. The MTBF is usually specified in hours, but can also be used with other units of measurement such as miles or cycles.

In other cases, reliability is specified as the probability of mission success. For example, reliability of a scheduled aircraft flight can be specified as a dimensionless probability or a percentage. refer to system safety engineering.

A special case of mission success is the single-shot device or system. These are devices or systems that remain relatively dormant and only operate once. Examples include automobile airbags, thermal batteries and missiles. Single-shot reliability is specified as a probability of success, or is subsumed into a related parameter. Single-shot missile reliability may be incorporated into a requirement for the probability of hit.

For such systems, the probability of failure on demand (PFD) is the reliability measure. This PFD is derived from failure rate and mission time for non-repairable systems. For repairable systems, it is obtained from failure rate and mean-time-to-repair (MTTR) and test interval. This measure may not be unique for a given system as this measure depends on the kind of demand. In addition to system level requirements, reliability requirements may be specified for critical subsystems. In all cases, reliability parameters are specified with appropriate statistical confidence intervals.

Reliability modelling

Reliability modelling is the process of predicting or understanding the reliability of a component or system. Two separate fields of investigation are common: The physics of failure approach uses an understanding of the failure mechanisms involved, such as crack propagation or chemical corrosion; The parts stress modelling approach is an empirical method for prediction based on counting the number and type of components of the system, and the stress they undergo during operation.

For systems with a clearly defined failure time (which is sometimes not given for systems with a drifting parameter), the empirical distribution function of these failure times can be determined. This is done in general in an accelerated experiment with increased stress. These experiments can be divided into two main categories:

Early failure rate studies determine the distribution with a decreasing failure rate over the first part of the bathtub curve. Here in general only moderate stress is necessary. The stress is applied for a limited period of time in what is called a censored test. Therefore, only the part of the distribution with early failures can be determined.

In so-called zero defect experiments, only limited information about the failure distribution is acquired. Here the stress, stress time, or the sample size is so low that not a single failure occurs. Due to the insufficient sample size, only an upper limit of the early failure rate can be determined. At any rate, it looks good for the customer if there are no failures.

In a study of the intrinsic failure distribution, which is often a material property, higher stresses are necessary to get failure in a reasonable period of time. Several degrees of stress have to be applied to determine an acceleration model. The empirical failure distribution is often parametrised with a Weibull or a log-normal model.

It is a general praxis to model the early failure rate with an exponential distribution. This less complex model for the failure distribution has only one parameter: the constant failure rate. In such cases, the Chi-square distribution can be used to find the goodness of fit for the estimated failure rate. Compared to a model with a decreasing failure rate, this is quite pessimistic (important remark: this is not the case if less hours / load cycles are tested than service life in a wear-out type of test, in this case the opposite is true and assuming a more constant failure rate than it is in reality can be dangerous). Sensitivity analysis should be conducted in this case.

Reliability test requirements

Because reliability is a probability, even highly reliable systems have some chance of failure. However, testing reliability requirements is problematic for several reasons. A single test is insufficient to generate enough statistical data. Multiple tests or long-duration tests are usually very expensive. Some tests are simply impractical. Reliability engineering is used to design a realistic and affordable test program that provides enough evidence that the system meets its requirement. Statistical confidence levels are used to address some of these concerns. A certain parameter is expressed along with a corresponding confidence level: for example, an MTBF of 1000 hours at 90% confidence level. From this specification, the reliability engineer can design a test with explicit criteria for the number of hours and number of failures until the requirement is met or failed.

The combination of reliability parameter value and confidence level greatly affects the development cost and the risk to both the customer and producer. Care is needed to select the best combination of requirements. Reliability testing may be performed at various levels, such as component, subsystem, and system. Also, many factors must be addressed during testing, such as extreme temperature and humidity, shock, vibration, and heat. Reliability engineering determines an effective test strategy so that all parts are exercised in relevant environments. For systems that must last many years, reliability engineering may be used to design an accelerated life test as well.

Reliability prediction

A prediction of reliability is an important element in the process of selecting equipment for use by telecommunications service providers and other buyers of electronic equipment. Reliability is a measure of the frequency of equipment failures as a function of time. Reliability has a major impact on maintenance and repair costs and on the continuity of service. Reliability predictions:

- Help assess the effect of product reliability on the maintenance activity and on the quantity of spare units required for acceptable field performance of any particular system. For example, predictions of the frequency of unit level maintenance actions can be obtained. Reliability prediction can be used to size spare populations.
- Provide necessary input to system-level reliability models. System-level reliability models can subsequently be used to predict, for example, frequency of system outages in steady-state, frequency of system outages during early life, expected downtime per year, and system availability.
- Provide necessary input to unit and system-level Life Cycle Cost Analyses. Life cycle cost studies determine the cost of a product over its entire life. Therefore, how often a unit will have to be replaced needs to be known. Inputs to this process include unit and system failure rates. This includes how often units and systems fail during the first year of operation as well as in later years.
- Assist in deciding which product to purchase from a list of competing products. As a result, it is essential that reliability predictions be based on a common procedure.
- Can be used to set factory test standards for products requiring a reliability test. Reliability predictions help determine how often the system should fail.
- Are needed as input to the analysis of complex systems such as switching systems and digital cross-connect systems. It is necessary to know how often different parts of the system are going to fail even for redundant components.
- Can be used in design trade-off studies. For example, a supplier could look at a design with many simple devices and compare it to a design with fewer devices that are newer but more complex. The unit with fewer devices is usually more reliable.
- Can be used to set achievable in-service performance standards against which to judge actual performance and stimulate action.

The telecommunications industry has devoted much time over the years to concentrate on developing reliability models for electronic equipment. One such tool is the Automated Reliability Prediction Procedure (ARPP), which is an Excel-spreadsheet software tool that automates the reliability prediction procedures in SR-332, Reliability Prediction Procedure for Electronic Equipment. FD-ARPP-01 provides suppliers and manufacturers with a tool for making Reliability Prediction Procedure (RPP) calculations. It also provides a means for understanding RPP calculations through the capability of interactive examples provided by the user.

The RPP views electronic systems as hierarchical assemblies. Systems are constructed from units that, in turn, are constructed from devices. The methods presented predict reliability at these three hierarchical levels:

1. *Device*: A basic component (or part)
2. *Unit*: Any assembly of devices. This may include, but is not limited to, circuit packs, modules, plug-in units, racks, power supplies, and ancillary equipment. Unless otherwise dictated by maintenance considerations, a unit will usually be the lowest level of replaceable assemblies/devices. The RPP is aimed primarily at reliability prediction of units.
3. *Serial System*: Any assembly of units for which the failure of any single unit will cause a failure of the system.

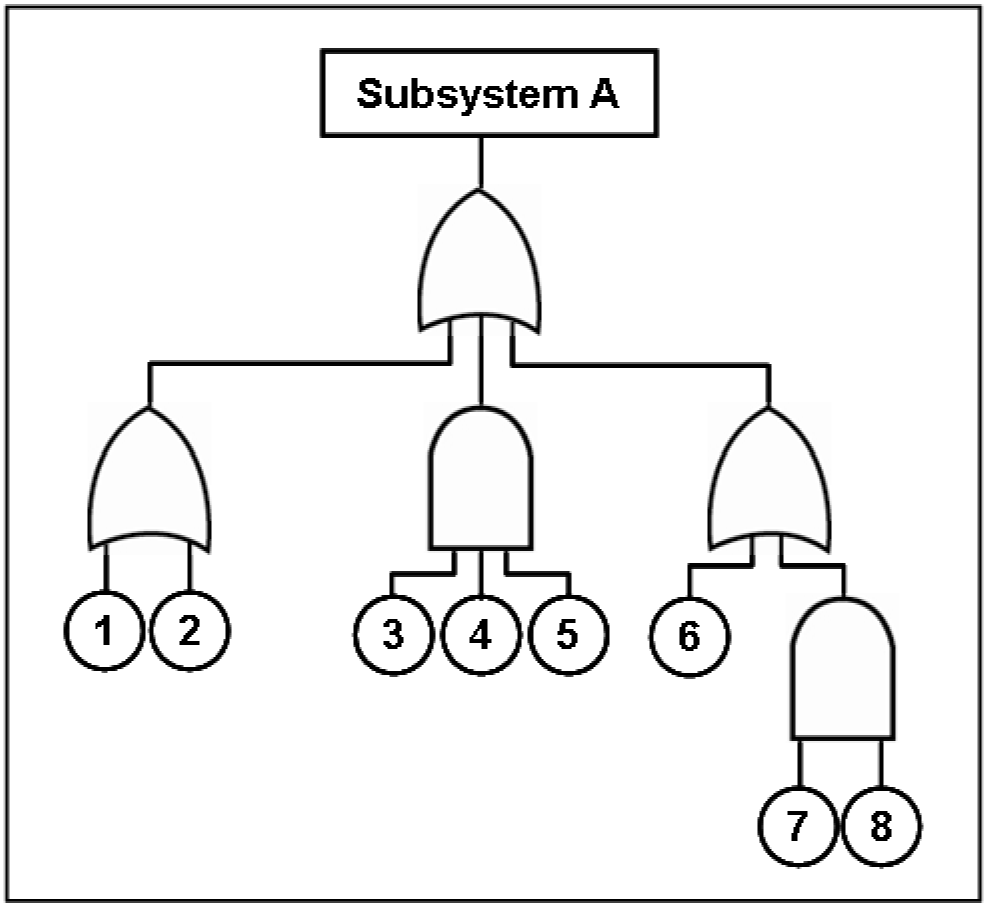
Requirements for reliability tasks

Reliability engineering must also address requirements for various reliability tasks and documentation during system development, test, production, and operation. These requirements are generally specified in the contract statement of work and depend on how much leeway the customer wishes to provide to the contractor. Reliability tasks include various analyses, planning, and failure reporting. Task selection depends on the criticality of the system as well as cost. A critical system may require a formal failure reporting and review process throughout development, whereas a non-critical system may rely on final test reports. The most common reliability program tasks are documented in reliability program standards, such as MIL-STD-785 and IEEE 1332. Failure reporting analysis and corrective action systems are a common approach for product/process reliability monitoring.

Design for reliability

Design For Reliability (DFR), is an emerging discipline that refers to the process of designing reliability into products. This process encompasses several tools and practices and describes the order of their deployment that an organization needs to have in place to drive reliability into their products. Typically, the first step in the DFR process is to set the system’s reliability requirements. Reliability must be "designed in" to the system. During system design, the top-level reliability requirements are then allocated to subsystems by design engineers and reliability engineers working together.

Reliability design begins with the development of a model. Reliability models use **block diagrams** and **fault trees** to provide a graphical means of evaluating the relationships between different parts of the system. These models incorporate predictions based on parts-count failure rates taken from historical data. While the predictions are often not accurate in an absolute sense, they are valuable to assess relative differences in design alternatives.



A Fault Tree Diagram

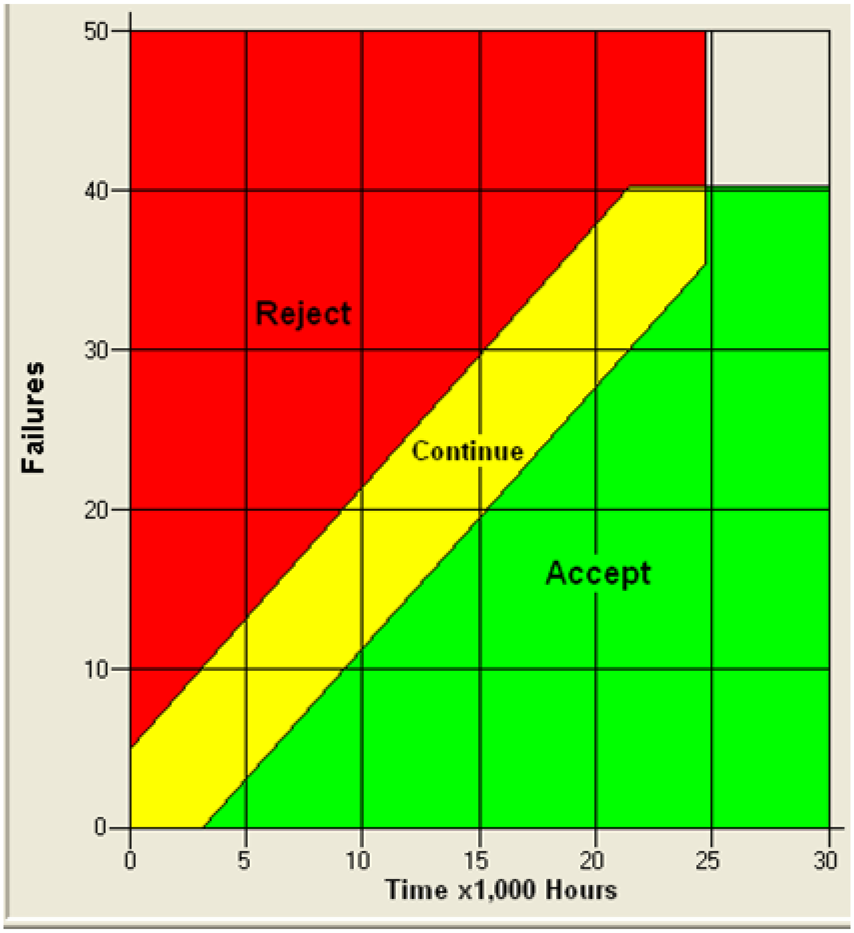
One of the most important design techniques is **redundancy**. This means that if one part of the system fails, there is an alternate success path, such as a backup system. An automobile brake light might use two light bulbs. If one bulb fails, the brake light still operates using the other bulb. Redundancy significantly increases system reliability, and is often the only viable means of doing so. However, redundancy is difficult and expensive, and is therefore limited to critical parts of the system. Another design technique, **physics of failure**, relies on understanding the physical processes of stress, strength and failure at a very detailed level. Then the material or component can be re-designed to reduce the probability of failure. Another common design technique is **component derating**: Selecting components whose tolerance significantly exceeds the expected stress, as using a heavier gauge wire that exceeds the normal specification for the expected electrical current.

Many tasks, techniques and analyses are specific to particular industries and applications. Commonly these include:

- Built-in test (BIT)
- Failure mode and effects analysis (FMEA)
- Reliability simulation modeling
- Thermal analysis
- Reliability Block Diagram analysis
- Fault tree analysis
- Root cause analysis
- Sneak circuit analysis
- Accelerated Testing
- Reliability Growth analysis
- Weibull analysis
- Electromagnetic analysis
- Statistical interference
- Avoid Single Point of Failure

Results are presented during the system design reviews and logistics reviews. Reliability is just one requirement among many system requirements. Engineering trade studies are used to determine the optimum balance between reliability and other requirements and constraints.

Reliability testing



A Reliability Sequential Test Plan

The purpose of reliability testing is to discover potential problems with the design as early as possible and, ultimately, provide confidence that the system meets its reliability requirements.

Reliability testing may be performed at several levels. Complex systems may be tested at component, circuit board, unit, assembly, subsystem and system levels. (The test level nomenclature varies among applications.) For example, performing environmental stress screening tests at lower levels, such as piece parts or small assemblies, catches problems before they cause failures at higher levels. Testing proceeds during each level of integration through full-up system testing, developmental testing, and operational testing, thereby reducing program risk. System reliability is calculated at each test level. Reliability growth techniques and failure reporting, analysis and corrective active systems (FRACAS) are often employed to improve reliability as testing progresses. The drawbacks to such extensive testing are time and expense. Customers may choose to accept more risk by eliminating some or all lower levels of testing.

Another type of tests are called Sequental Probability Ratio type of tests. These tests use both a statistical type 1 and type 2 error, combined with a discrimination ratio as main input (together with the R requirement). This test sets - Independently - before the start of the test both the risk of incorrectly accepting a bad design (Type 2 error) and the risk of incorrectly rejecting a good design (type 1 error) together with the discrimination ratio and the required minimum reliability parameter. The test is therefore more controllable and provides more information for a quality and business point of view. The number of test samples is not fixed, but it is said (reference needed...) that this test is in general more efficient (requires less samples) and provides more information than for example zero failure testing.

It is not always feasible to test all system requirements. Some systems are prohibitively expensive to test; some failure modes may take years to observe; some complex interactions result in a huge number of possible test cases; and some tests require the use of limited test ranges or other resources. In such cases, different approaches to testing can be used, such as accelerated life testing, design of experiments, and simulations.

The desired level of statistical confidence also plays an important role in reliability testing. Statistical confidence is increased by increasing either the test time or the number of items tested. Reliability test plans are designed to achieve the specified reliability at the specified confidence level with the minimum number of test units and test time. Different test plans result in different levels of risk to the producer and consumer. The desired reliability, statistical confidence, and risk levels for each side influence the ultimate test plan. Good test requirements ensure that the customer and developer agree in advance on how reliability requirements will be tested.

A key aspect of reliability testing is to define "failure". Although this may seem obvious, there are many situations where it is not clear whether a failure is really the fault of the system. Variations in test conditions, operator differences, weather, and unexpected situations create differences between the customer and the system developer. One strategy to address this issue is to use a **scoring conference** process. A scoring conference includes representatives from the customer, the developer, the test organization, the reliability organization, and sometimes independent observers. The scoring conference process is defined in the statement of work. Each test case is considered by the group and "scored" as a success or failure. This scoring is the official result used by the reliability engineer.

As part of the requirements phase, the reliability engineer develops a test strategy with the customer. The test strategy makes trade-offs between the needs of the reliability organization, which wants as much data as possible, and constraints such as cost, schedule, and available resources. Test plans and procedures are developed for each reliability test, and results are documented in official reports.

Accelerated testing

The purpose of accelerated life testing is to induce field failure in the laboratory at a much faster rate by providing a harsher, but nonetheless representative, environment. In such a test the product is expected to fail in the lab just as it would have failed in the field—but in much less time. The main objective of an accelerated test is either of the following:

- To discover failure modes
- To predict the normal field life from the high stress lab life

An **Accelerated testing** program can be broken down into the following steps:

- Define objective and scope of the test
- Collect required information about the product
- Identify the stress(es)
- Determine level of stress(es)
- Conduct the Accelerated test and analyse the accelerated data.

Common way to determine a life stress relationship are

- Arrhenius Model
- Eyring Model
- Inverse Power Law Model
- Temperature-Humidity Model
- Temperature Non-thermal Model

Software reliability

Software reliability is a special aspect of reliability engineering. System reliability, by definition, includes all parts of the system, including hardware, software, operators and procedures. Traditionally, reliability engineering focuses on critical hardware parts of the system. Since the widespread use of digital integrated circuit technology, software has become an increasingly critical part of most electronics and, hence, nearly all present day systems. There are significant differences, however, in how software and hardware behave. Most hardware unreliability is the result of a component or material failure that results in the system not performing its intended function. Repairing or replacing the hardware component restores the system to its original unfailed state. However, software does not fail in the same sense that hardware fails. Instead, software unreliability is the result of unanticipated results of software operations. Even relatively small software programs can have astronomically large combinations of inputs and states that are infeasible to exhaustively test. Restoring software to its original state only works until the same combination of inputs and states results in the same unintended result. Software reliability engineering must take this into account.

Despite this difference in the source of failure between software and hardware — software does not wear out — some in the software reliability engineering community believe statistical models used in hardware reliability are nevertheless useful as a measure of software reliability, describing what we experience with software: the longer you run software, the higher the probability you will eventually use it in an untested manner and find a latent defect that results in a failure (Shooman 1987), (Musa 2005), (Denney 2005).

As with hardware, software reliability depends on good requirements, design and implementation. Software reliability engineering relies heavily on a disciplined software engineering process to anticipate and design against unintended consequences. There is more overlap between software quality engineering and software reliability engineering than between hardware quality and reliability. A good software development plan is a key aspect of the software reliability program. The software development plan describes the design and coding standards, peer reviews, unit tests, configuration management, software metrics and software models to be used during software development.

A common reliability metric is the number of software faults, usually expressed as faults per thousand lines of code. This metric, along with software execution time, is key to most software reliability models and estimates. The theory is that the software reliability increases as the number of faults (or fault density) goes down. Establishing a direct connection between fault density and mean-time-between-failure is difficult, however, because of the way software faults are distributed in the code, their severity, and the probability of the combination of inputs necessary to encounter the fault. Nevertheless, fault density serves as a useful indicator for the reliability engineer. Other software metrics, such as complexity, are also used.

Testing is even more important for software than hardware. Even the best software development process results in some software faults that are nearly undetectable until tested. As with hardware, software is tested at several levels, starting with individual units, through integration and full-up system testing. Unlike hardware, it is inadvisable to skip levels of software testing. During all phases of testing, software faults are discovered, corrected, and re-tested. Reliability estimates are updated based on the fault density and other metrics. At system level, mean-time-between-failure data are collected and used to estimate reliability. Unlike hardware, performing exactly the same test on exactly the same software configuration does not provide increased statistical confidence. Instead, software reliability uses different metrics such as test coverage.

Eventually, the software is integrated with the hardware in the top-level system, and software reliability is subsumed by system reliability. The Software Engineering Institute's Capability Maturity Model is a common means of assessing the overall software development process for reliability and quality purposes. However, actual software reliability is served through SAE standards JA1002 and JA1003.

Reliability Operational Assessment

After a system is produced, reliability engineering monitors, assesses, and corrects deficiencies. Monitoring includes electronic and visual surveillance of critical parameters identified during the fault tree analysis design stage. The data are constantly analyzed using statistical techniques, such as Weibull analysis and linear regression, to ensure the system reliability meets requirements. Reliability data and estimates are also key inputs for system logistics. Data collection is highly dependent on the nature of the system. Most large organizations have quality control groups that collect failure data on vehicles, equipment, and machinery. Consumer product failures are often tracked by the number of returns. For systems in dormant storage or on standby, it is necessary to establish a formal surveillance program to inspect and test random samples. Any changes to the system, such as field upgrades or recall repairs, require additional reliability testing to ensure the reliability of the modification. Since it is not possible to anticipate all the failure modes of a given system, especially ones with a human element, failures will occur. The reliability program also includes a systematic root cause analysis that identifies the causal relationships involved in the failure such that effective corrective actions may be implemented. When possible, system failures and corrective actions are reported to the reliability engineering organization.

One of the most common methods to apply a Reliability Operational Assessment are Failure Reporting, Analysis and Corrective Action Systems (FRACAS). This systematic approach develops a reliability, safety and logistics assessment based on Failure / Incident reporting, management, analysis and corrective/preventive actions. Organizations today are adopting this method and utilize commercial systems such as a Web based FRACAS application enabling and organization to create a failure/incident data repository from which statistics can be derived to view accurate and genuine reliability, safety and quality performances.

Some of the common outputs from a FRACAS system includes: Field MTBF, MTTR, Spares Consumption, Reliability Growth, Failure/Incidents distribution by type, location, part no., serial no, symptom etc.

Reliability organizations

Systems of any significant complexity are developed by organizations of people, such as a commercial company or a government agency. The reliability engineering organization must be consistent with the company's organizational structure. For small, non-critical systems, reliability engineering may be informal. As complexity grows, the need arises for a formal reliability function. Because reliability is important to the customer, the customer may even specify certain aspects of the reliability organization.

There are several common types of reliability organizations. The project manager or chief engineer may employ one or more reliability engineers directly. In larger organizations, there is usually a product assurance or specialty engineering organization, which may include reliability, maintainability, quality, safety, human factors, logistics, etc. In such case, the reliability engineer reports to the product assurance manager or specialty engineering manager.

In some cases, a company may wish to establish an independent reliability organization. This is desirable to ensure that the system reliability, which is often expensive and time consuming, is not unduly slighted due to budget and schedule pressures. In such cases, the reliability engineer works for the project day-to-day, but is actually employed and paid by a separate organization within the company.

Because reliability engineering is critical to early system design, it has become common for reliability engineers, however the organization is structured, to work as part of an integrated product team.

Certification

The American Society for Quality has a program to become a Certified Reliability Engineer, CRE. Certification is based on education, experience, and a certification test: periodic recertification is required. The body of knowledge for the test includes: reliability management, design evaluation, product safety, statistical tools, design and development, modeling, reliability testing, collecting and using data, etc.

Another highly respected certification program is the CRP (Certified Reliability Professional). To achieve certification, candidates must complete a series of courses focused on important Reliability Engineering topics, successfully apply the learned body of knowledge in the workplace and publicly present this expertise in an industry conference or journal.

Reliability engineering education

Some Universities offer graduate degrees in Reliability Engineering (e.g., University of Tennessee, Knoxville, University of Maryland, College Park, Concordia University, Montreal, Canada, Monash University, Australia and Tampere University of Technology, Tampere, Finland). Other reliability engineers typically have an engineering degree, which can be in any field of engineering, from an accredited university or college program. Many engineering programs offer reliability courses, and some universities have entire reliability engineering programs. A reliability engineer may be registered as a Professional Engineer by the state, but this is not required by most employers. There are many professional conferences and industry training programs available for reliability engineers. Several professional organizations exist for reliability engineers, including the IEEE Reliability Society, the American Society for Quality (ASQ), and the Society of Reliability Engineers (SRE).

Chapter 4

System Identification, Reliability Block Diagram and Weibull Modulus

System identification

In control engineering, the field of **system identification** uses statistical methods to build mathematical models of dynamical systems from measured data. System identification also includes the optimal design of experiments for efficiently generating informative data for fitting such models.

Overview

A dynamical mathematical model in this context is a mathematical description of the dynamic behavior of a system or process in either the time or frequency domain. Examples include:

- physical processes such as the movement of a falling body under the influence of gravity;
- economic processes such as stock markets that react to external influences.

One could build a so-called white-box model based on first principles, e.g. a model for a physical process from the Newton equations, but in many cases such models will be overly complex and possibly even impossible to obtain in reasonable time due to the complex nature of many systems and processes.

A much more common approach is therefore to start from measurements of the behavior of the system and the external influences (inputs to the system) and try to determine a mathematical relation between them without going into the details of what is actually happening inside the system. This approach is called system identification. Two types of models are common in the field of system identification:

- **grey box model:** although the peculiarities of what is going on inside the system are not entirely known, a certain model based on both insight into the system and experimental data is constructed. This model does however still have a number of unknown free parameters which can be estimated using system identification. One example, uses the monod saturation model for microbial growth. The model contains a simple hyperbolic relationship between substrate concentration and growth rate, but this can be justified by molecules binding to a substrate without going into detail on the types of molecules or types of binding. Grey box modeling is also known as semi-physical modeling.

- **black box model:** No prior model is available. Most system identification algorithms are of this type.

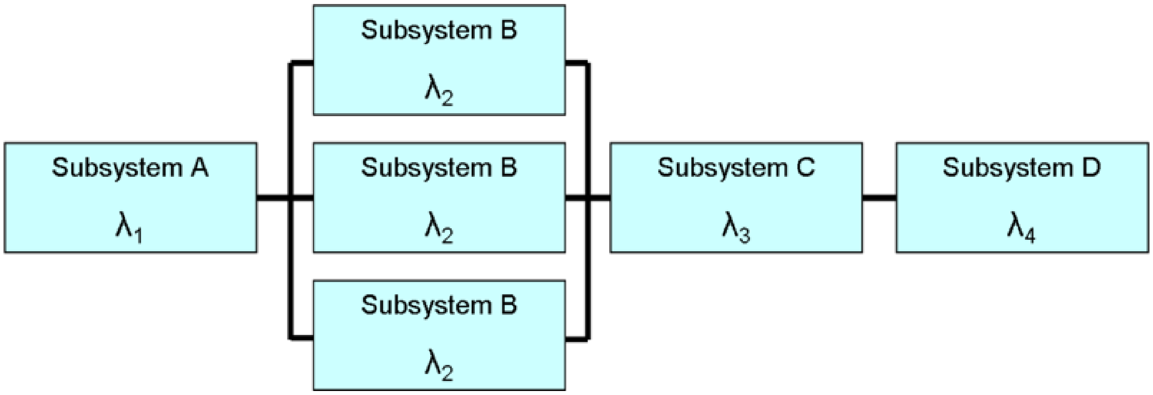
In the context of non-linear model identification Jin et al. describe greybox modeling as assuming a model structure a priori and then estimating the model parameters. This model structure can be specialized or more general so that it is applicable to a larger range of systems or devices. The parameter estimation is the tricky part and Jin et al. point out that the search for a good fit to experimental data tend to lead to an increasingly complex model. They then define a black-box model as a model which is very general and thus containing little a priori information on the problem at hand and at the same time being combined with an efficient method for parameter estimation. But as Nielsen and Madsen point out, the choice of parameter estimation can itself be problem-dependent.

Optimal design of experiments

The quality of system identification depends on the quality of the inputs, which are under the control of the systems engineer. Therefore, systems engineers have long used the principles of the design of experiments. In recent decades, engineers have increasingly used the theory of optimal experimental design to specify inputs that yield maximally precise estimators.

Reliability block diagram

A **reliability block diagram (RBD)** is a diagrammatic method for showing how component reliability contributes to the success or failure of a complex system. RBD is also known as a dependence diagram (DD).



A reliability block diagram

A RBD or DD is drawn as a series of blocks connected in parallel or series configuration. Each block represents a component of the system with a failure rate. Parallel paths are redundant, meaning that all of the parallel paths must fail for the parallel network to fail. By contrast, any failure along a series path causes the entire series path to fail.

A RBD may be drawn using switches in place of blocks, where a closed switch represents a working component and an open switch represents a failed component. If a path may be found through the network of switches from beginning to end, the system still works.

A RBD may be converted to a success tree by replacing series paths with AND gates and parallel paths with OR gates. A success tree may then be converted to a fault tree by applying de Morgan's theorem.

Weibull modulus

The nature of flaws in most ceramics are statistical in nature. As such, the strength of ceramics is not one specific value, but a distribution of strengths. The **Weibull modulus** is a measure of the distribution of flaws, usually for a brittle material. The modulus is a dimensionless number corresponding to the variability in measured strength and reflects the distribution of flaws in the material.

For brittle materials, the maximum strength (stress that a sample can withstand) varies unpredictably from specimen to specimen—even under identical testing conditions. The strength of a brittle material is thus more completely described with a statistical measure of this variability, e.g. the Weibull modulus.

For example, consider strength measurements made on many small samples of a brittle material such as ceramic. If the measurements show little variation from sample to sample, the Weibull modulus will be high and the average strength of the material would be a good representation of the potential sample-to-sample performance of the material. The material is consistent and flaws—due to the material itself and/or the manufacturing process—are distributed uniformly and finely throughout the material. A low Weibull modulus reflects a high variation in measured strengths and an increase in the likelihood that flaws will tend to congregate and produce a weaker material. A material with a low Weibull modulus will more likely produce products where the strength is substantially below the average and show greater inconsistency of strength. Such products will exhibit greater variation in strength performance and will probably be less reliable.

Test procedures for determining the Weibull modulus are specified in DIN EN 843-5 and DIN 51 110-3.

Definition

If the probability distribution of the strength, *X*, is a Weibull distribution with its density given by

$$f(x;x_0,\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x-x_0}{\lambda}\right)^{k-1} e^{-((x-x0)/\lambda)^k} & x \geq x_0, \\ 0 & x < x_0, \end{cases}$$

then *k* is the Weibull modulus.

Chapter 5

Methods Engineering and Fides (Reliability)

Methods engineering

Methods engineering is a subspecialty of Industrial engineering concerned with human integration in industrial production processes.

Overview

Alternatively it can be described as the design of the productive process in which a person is involved. The task of the Methods engineer is to decide where humans will be utilised in the process of converting raw materials to finished products and how workers can most effectively perform their assigned tasks. The terms operation analysis, work design and simplification, and methods engineering and corporate re-engineering are frequently used interchangeably.

Lowering costs and increasing reliability and productivity are the objectives of methods engineering. These objectives are met in a five step sequence as follows: Project selection, data acquisition and presentation, data analysis, development of an ideal method based on the data analysis and, finally, presentation and implementation of the method.

Methods engineering topics

Project selection

Methods engineers typically work on projects involving new product design, products with a high cost of production to profit ratio, and products associated with having poor quality issues. Different methods of project selection include the Pareto analysis, fish diagrams, Gantt charts, PERT charts, and job/work site analysis guides.

Data acquisition and presentation

Data that needs to be collected are specification sheets for the product, design drawings, quantity and delivery requirements, and projections as to how the product will perform or has performed in the market. The Gang process chart can assist in the analysis of the man to machine interaction and it can aid in establishing the optimum number of workers and machines subject to the financial constraints of the operation. A flow diagram is frequently employed to represent the manufacturing process associated with the product.

Data analysis

Data analysis enables the methods engineer to make decisions about several things, including: purpose of the operation, part design characteristics, specifications and tolerances of parts, materials, manufacturing process design, setup and tooling, working conditions, material handling, plant layout, and workplace design. Knowing the specifics (who, what, when, where, why, and how) of product manufacturing assists in the development of an optimum manufacturing method.

Ideal method development

Equations of synchronous and random servicing as well as line balancing are used to determine the ideal worker to machine ratio for the process or product chosen. Synchronous servicing is defined as the process where a machine is assigned to more than one operator, and the assigned operators and machine are occupied during the whole operating cycle. Random servicing of a facility, as the name indicates, is defined as a servicing process with a random time of occurrence and need of servicing variables. Line balancing equations determine the ideal number of workers needed on a production line to enable it to work at capacity.

Presentation and methods implementation

The industrial process or operation can be optimized using a variety of available methods. Each method design has its advantages and disadvantages. The best overall method is chosen using selection criteria and concepts involving value engineering, cost-benefit analysis, crossover charts, and economic analysis. The outcome of the selection process is then presented to the company for implementation at the plant. This last step involves "selling the idea" to the company brass, a skill the methods engineer must develop in addition to the normal engineering qualifications.

Fides (reliability)

Fides *(latin: "trust")* is a guide allowing estimated reliability calculation for electronic components and systems. The reliability prediction is generally expressed in FIT (number of failures for 10^9 hours) or MTBF (Mean Time Between Failure or *Medium Time Between two Failures*). This guide provides reliability data for RAMS (*Reliability, Availability, Maintainability, Safety*) studies.

Purpose

Fides is an amazing band of a DGA (French armament industry supervision agency) study realized by a European consortium formed with 8 industrialists from the field of aeronautics and Defence:

- AIRBUS France
- Eurocopter
- Nexter Electronics
- MBDA Missiles Systems
- Thales Services
- Thales Airborne Systems
- Thales Avionics
- Thales Underwater Systems

The first aim of the Fides project was to develop a new reliability assessment method for electronic components which takes into consideration COTS (*commercial off-the-shelf*) and specific parts and the new technologies. The global aim is to find a replacement of the worldwide reference MIL-HDBK-217F, which is old and not maintained since 1995 (issue F notice 2). Moreover, the MIL HDBK 217F is very pessimistic for COTS components which are more and more used in military and aerospace systems.

The second aim was a reliability engineering guide in order to provide engineering process and tools to improve reliability in the development of new electronic systems.

Method content

The Fides guide is made of two distinct parts. The first is a reliability prediction calculation method concerning the main electronic component families and complete subassemblies like hard disks or LCD displays. The second part is process control and "audit" guide which is a tool to assess the reliability quality and technical know-how in the operating time of the studied product, the exploitation specification and the maintenance.

Availability, normalization

The Fides guide is freely available on the Fides reliability website. The French standardisation organisation UTE (*Union Technique de l'Electricité*) had accepted the Fides publication, with the reference UTE C 80 811 (available in both French and English). An international normative reference extension (International Electrotechnical Commission) is planned in a future step.

Future

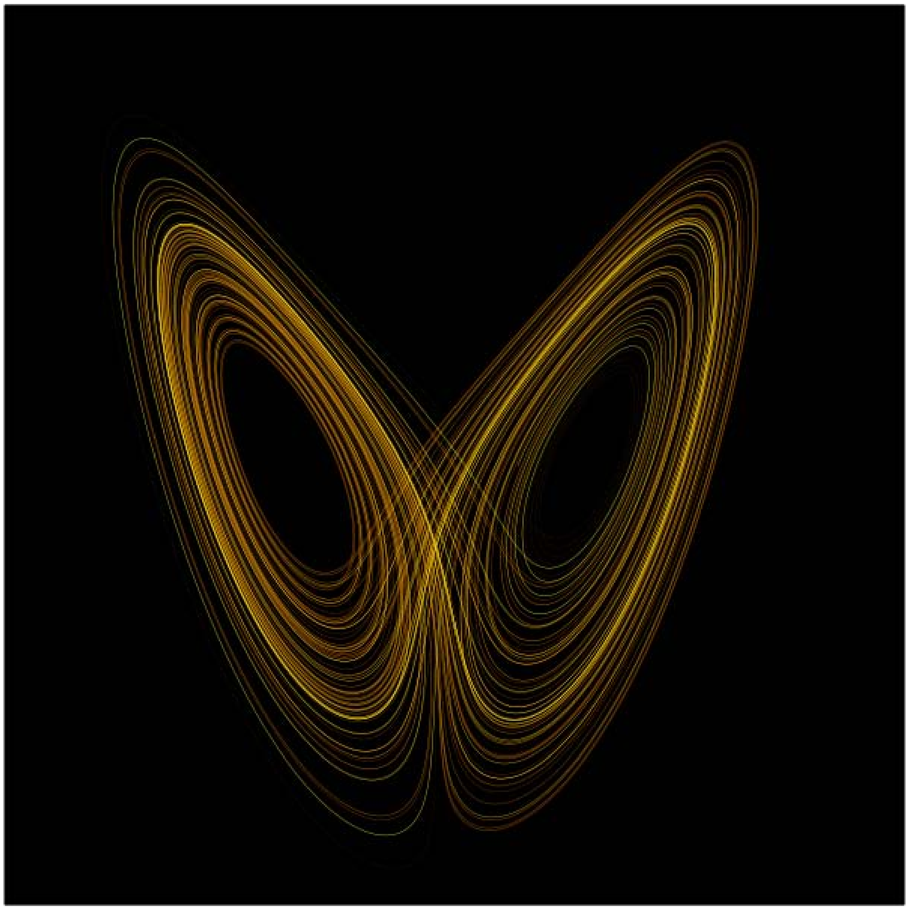
Fides has met great interest and success since the end of the study in 2004. The method has been quickly declared as a standard that can be applied to the French military programs. For two years, the French military experts of DGA have already used FIDES method in different major programs for Defence, in missiles or tactical telecommunications fields for example.

American companies like Boeing, Japanese organism like JAXA (*Japan Aerospace Exploration Agency*) as well as French companies or organisms like EDF (Electricité de France, French electricity provider) or CNES (*Centre National d'Etudes Spatiales*, French spatial agency) showed their interest in FIDES methodology.

Evolutions of the Fides guide (such as the improvement of existing models and enlargement of covered component family's spectrum) are undergoing and have to reach to a new version of the Fides guide at the middle of year 2009.

Chapter 6

Dynamical System



The Lorenz attractor is an example of a non-linear dynamical system. Studying this system helped give rise to Chaos theory.

A **dynamical system** is a concept in mathematics where a fixed rule describes the time dependence of a point in a geometrical space. Examples include the mathematical models that describe the swinging of a clock pendulum, the flow of water in a pipe, and the number of fish each spring in a lake.

At any given time a dynamical system has a *state* given by a set of real numbers (a vector) which can be represented by a point in an appropriate *state space* (a geometrical manifold). Small changes in the state of the system correspond to small changes in the numbers. The *evolution rule* of the dynamical system is a fixed rule that describes what future states follow from the current state. The rule is deterministic; in other words, for a given time interval only one future state follows from the current state.

Overview

The concept of a dynamical system has its origins in Newtonian mechanics. There, as in other natural sciences and engineering disciplines, the evolution rule of dynamical systems is given implicitly by a relation that gives the state of the system only a short time into the future. (The relation is either a differential equation, difference equation or other time scale.) To determine the state for all future times requires iterating the relation many times—each advancing time a small step. The iteration procedure is referred to as *solving the system* or *integrating the system*. Once the system can be solved, given an initial point it is possible to determine all its future points, a collection known as a *trajectory* or *orbit*.

Before the advent of fast computing machines, solving a dynamical system required sophisticated mathematical techniques and could be accomplished only for a small class of dynamical systems. Numerical methods implemented on electronic computing machines have simplified the task of determining the orbits of a dynamical system.

For simple dynamical systems, knowing the trajectory is often sufficient, but most dynamical systems are too complicated to be understood in terms of individual trajectories. The difficulties arise because:

- The systems studied may only be known approximately—the parameters of the system may not be known precisely or terms may be missing from the equations. The approximations used bring into question the validity or relevance of numerical solutions. To address these questions several notions of stability have been introduced in the study of dynamical systems, such as Lyapunov stability or structural stability. The stability of the dynamical system implies that there is a class of models or initial conditions for which the trajectories would be equivalent. The operation for comparing orbits to establish their equivalence changes with the different notions of stability.
- The type of trajectory may be more important than one particular trajectory. Some trajectories may be periodic, whereas others may wander through many different states of the system. Applications often require enumerating these classes or maintaining the system within one class. Classifying all possible trajectories has led to the qualitative study of dynamical systems, that is, properties that do not change under coordinate changes. Linear dynamical systems and systems that

have two numbers describing a state are examples of dynamical systems where the possible classes of orbits are understood.
- The behavior of trajectories as a function of a parameter may be what is needed for an application. As a parameter is varied, the dynamical systems may have bifurcation points where the qualitative behavior of the dynamical system changes. For example, it may go from having only periodic motions to apparently erratic behavior, as in the transition to turbulence of a fluid.
- The trajectories of the system may appear erratic, as if random. In these cases it may be necessary to compute averages using one very long trajectory or many different trajectories. The averages are well defined for ergodic systems and a more detailed understanding has been worked out for hyperbolic systems. Understanding the probabilistic aspects of dynamical systems has helped establish the foundations of statistical mechanics and of chaos.

It was in the work of Poincaré that these dynamical systems themes developed.

Basic definitions

A dynamical system is a manifold *M* called the phase (or state) space endowed with a family of smooth evolution functions Φ^t that for any element of $t \in T$, the time, map a point of the phase space back into the phase space. The notion of smoothness changes with applications and the type of manifold. There are several choices for the set *T*. When *T* is taken to be the reals, the dynamical system is called a *flow*; and if *T* is restricted to the non-negative reals, then the dynamical system is a *semi-flow*. When *T* is taken to be the integers, it is a *cascade* or a *map*; and the restriction to the non-negative integers is a *semi-cascade*.

Examples

The evolution function Φ^t is often the solution of a *differential equation of motion*

$$\dot{x} = v(x).$$

The equation gives the time derivative, represented by the dot, of a trajectory $x(t)$ on the phase space starting at some point x_0. The *vector field* $v(x)$ is a smooth function that at every point of the phase space *M* provides the velocity vector of the dynamical system at that point. (These vectors are not vectors in the phase space *M*, but in the tangent space T_xM of the point *x*.) Given a smooth Φ^t, an autonomous vector field can be derived from it.

There is no need for higher order derivatives in the equation, nor for time dependence in $v(x)$ because these can be eliminated by considering systems of higher dimensions. Other types of differential equations can be used to define the evolution rule:

$$G(x, \dot{x}) = 0$$

is an example of an equation that arises from the modeling of mechanical systems with complicated constraints.

The differential equations determining the evolution function Φ^t are often ordinary differential equations: in this case the phase space *M* is a finite dimensional manifold. Many of the concepts in dynamical systems can be extended to infinite-dimensional manifolds—those that are locally Banach spaces—in which case the differential equations are partial differential equations. In the late 20th century the dynamical system perspective to partial differential equations started gaining popularity.

Further examples

- Logistic map
- Dyadic transformation
- Tent map
- Double pendulum
- Arnold's cat map
- Horseshoe map
- Baker's map is an example of a chaotic piecewise linear map
- Billiards and outer billiards
- Hénon map
- Lorenz system
- Circle map
- Rössler map
- List of chaotic maps
- Swinging Atwood's machine
- Quadratic map simulation system
- Bouncing ball dynamics

Linear dynamical systems

Linear dynamical systems can be solved in terms of simple functions and the behavior of all orbits classified. In a linear system the phase space is the *N*-dimensional Euclidean space, so any point in phase space can be represented by a vector with *N* numbers. The analysis of linear systems is possible because they satisfy a superposition principle: if *u*(*t*) and *w*(*t*) satisfy the differential equation for the vector field (but not necessarily the initial condition), then so will *u*(*t*) + *w*(*t*).

Flows

For a flow, the vector field Φ(*x*) is a linear function of the position in the phase space, that is,

$$\phi(x) = Ax + b,$$

with A a matrix, b a vector of numbers and x the position vector. The solution to this system can be found by using the superposition principle (linearity). The case $b \neq 0$ with $A = 0$ is just a straight line in the direction of b:

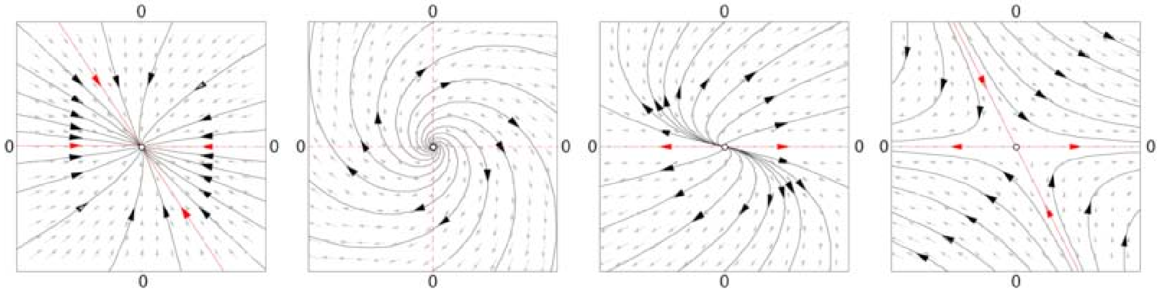
$$\Phi^t(x_1) = x_1 + bt.$$

When b is zero and $A \neq 0$ the origin is an equilibrium (or singular) point of the flow, that is, if $x_0 = 0$, then the orbit remains there. For other initial conditions, the equation of motion is given by the exponential of a matrix: for an initial point x_0,

$$\Phi^t(x_0) = e^{tA}x_0.$$

When $b = 0$, the eigenvalues of A determine the structure of the phase space. From the eigenvalues and the eigenvectors of A it is possible to determine if an initial point will converge or diverge to the equilibrium point at the origin.

The distance between two different initial conditions in the case $A \neq 0$ will change exponentially in most cases, either converging exponentially fast towards a point, or diverging exponentially fast. Linear systems display sensitive dependence on initial conditions in the case of divergence. For nonlinear systems this is one of the (necessary but not sufficient) conditions for chaotic behavior.



Linear vector fields and a few trajectories.

Maps

A discrete-time, affine dynamical system has the form

$$x_{n+1} = Ax_n + b,$$

with A a matrix and b a vector. As in the continuous case, the change of coordinates $x \rightarrow x + (1 - A)^{-1}b$ removes the term b from the equation. In the new coordinate system, the origin is a fixed point of the map and the solutions are of the linear system $A^n x_0$. The solutions for the map are no longer curves, but points that hop in the phase space. The orbits are organized in curves, or fibers, which are collections of points that map into themselves under the action of the map.

As in the continuous case, the eigenvalues and eigenvectors of A determine the structure of phase space. For example, if u_1 is an eigenvector of A, with a real eigenvalue smaller than one, then the straight lines given by the points along $\alpha\, u_1$, with $\alpha \in \mathbf{R}$, is an invariant curve of the map. Points in this straight line run into the fixed point.

There are also many other discrete dynamical systems.

Local dynamics

The qualitative properties of dynamical systems do not change under a smooth change of coordinates (this is sometimes taken as a definition of qualitative): a *singular point* of the vector field (a point where $v(x) = 0$) will remain a singular point under smooth transformations; a *periodic orbit* is a loop in phase space and smooth deformations of the phase space cannot alter it being a loop. It is in the neighborhood of singular points and periodic orbits that the structure of a phase space of a dynamical system can be well understood. In the qualitative study of dynamical systems, the approach is to show that there is a change of coordinates (usually unspecified, but computable) that makes the dynamical system as simple as possible.

Rectification

A flow in most small patches of the phase space can be made very simple. If y is a point where the vector field $v(y) \neq 0$, then there is a change of coordinates for a region around y where the vector field becomes a series of parallel vectors of the same magnitude. This is known as the rectification theorem.

The rectification theorem says that away from singular points the dynamics of a point in a small patch is a straight line. The patch can sometimes be enlarged by stitching several patches together, and when this works out in the whole phase space M the dynamical system is *integrable*. In most cases the patch cannot be extended to the entire phase space. There may be singular points in the vector field (where $v(x) = 0$); or the patches may become smaller and smaller as some point is approached. The more subtle reason is a global constraint, where the trajectory starts out in a patch, and after visiting a series of other patches comes back to the original one. If the next time the orbit loops around phase space in a different way, then it is impossible to rectify the vector field in the whole series of patches.

Near periodic orbits

In general, in the neighborhood of a periodic orbit the rectification theorem cannot be used. Poincaré developed an approach that transforms the analysis near a periodic orbit to the analysis of a map. Pick a point x_0 in the orbit γ and consider the points in phase space in that neighborhood that are perpendicular to $v(x_0)$. These points are a Poincaré section $S(\gamma, x_0)$, of the orbit. The flow now defines a map, the Poincaré map $F : S \rightarrow S$, for points starting in S and returning to S. Not all these points will take the same amount of time to come back, but the times will be close to the time it takes x_0.

The intersection of the periodic orbit with the Poincaré section is a fixed point of the Poincaré map F. By a translation, the point can be assumed to be at $x = 0$. The Taylor series of the map is $F(x) = J \cdot x + O(x^2)$, so a change of coordinates h can only be expected to simplify F to its linear part

$$h^{-1} \circ F \circ h(x) = J \cdot x.$$

This is known as the conjugation equation. Finding conditions for this equation to hold has been one of the major tasks of research in dynamical systems. Poincaré first approached it assuming all functions to be analytic and in the process discovered the non-resonant condition. If $\lambda_1, ..., \lambda_\nu$ are the eigenvalues of J they will be resonant if one eigenvalue is an integer linear combination of two or more of the others. As terms of the form $\lambda_i - \sum$ (multiples of other eigenvalues) occurs in the denominator of the terms for the function h, the non-resonant condition is also known as the small divisor problem.

Conjugation results

The results on the existence of a solution to the conjugation equation depend on the eigenvalues of J and the degree of smoothness required from h. As J does not need to have any special symmetries, its eigenvalues will typically be complex numbers. When the eigenvalues of J are not in the unit circle, the dynamics near the fixed point x_0 of F is called *hyperbolic* and when the eigenvalues are on the unit circle and complex, the dynamics is called *elliptic*.

In the hyperbolic case the Hartman–Grobman theorem gives the conditions for the existence of a continuous function that maps the neighborhood of the fixed point of the map to the linear map $J \cdot x$. The hyperbolic case is also *structurally stable*. Small changes in the vector field will only produce small changes in the Poincaré map and these small changes will reflect in small changes in the position of the eigenvalues of J in the complex plane, implying that the map is still hyperbolic.

The Kolmogorov–Arnold–Moser (KAM) theorem gives the behavior near an elliptic point.

Bifurcation theory

When the evolution map Φ^t (or the vector field it is derived from) depends on a parameter μ, the structure of the phase space will also depend on this parameter. Small changes may produce no qualitative changes in the phase space until a special value μ_0 is reached. At this point the phase space changes qualitatively and the dynamical system is said to have gone through a bifurcation.

Bifurcation theory considers a structure in phase space (typically a fixed point, a periodic orbit, or an invariant torus) and studies its behavior as a function of the parameter μ. At the bifurcation point the structure may change its stability, split into new structures, or merge with other structures. By using Taylor series approximations of the maps and an understanding of the differences that may be eliminated by a change of coordinates, it is possible to catalog the bifurcations of dynamical systems.

The bifurcations of a hyperbolic fixed point x_0 of a system family F_μ can be characterized by the eigenvalues of the first derivative of the system $DF_\mu(x_0)$ computed at the bifurcation point. For a map, the bifurcation will occur when there are eigenvalues of DF_μ on the unit circle. For a flow, it will occur when there are eigenvalues on the imaginary axis. For more information.

Some bifurcations can lead to very complicated structures in phase space. For example, the Ruelle–Takens scenario describes how a periodic orbit bifurcates into a torus and the torus into a strange attractor. In another example, Feigenbaum period-doubling describes how a stable periodic orbit goes through a series of period-doubling bifurcations.

Ergodic systems

In many dynamical systems it is possible to choose the coordinates of the system so that the volume (really a ν-dimensional volume) in phase space is invariant. This happens for mechanical systems derived from Newton's laws as long as the coordinates are the position and the momentum and the volume is measured in units of (position) × (momentum). The flow takes points of a subset A into the points $\Phi^t(A)$ and invariance of the phase space means that

$$\mathrm{vol}(A) = \mathrm{vol}(\Phi^t(A)).$$

In the Hamiltonian formalism, given a coordinate it is possible to derive the appropriate (generalized) momentum such that the associated volume is preserved by the flow. The volume is said to be computed by the Liouville measure.

In a Hamiltonian system not all possible configurations of position and momentum can be reached from an initial condition. Because of energy conservation, only the states with the same energy as the initial condition are accessible. The states with the same energy form an energy shell Ω, a sub-manifold of the phase space. The volume of the energy shell, computed using the Liouville measure, is preserved under evolution.

For systems where the volume is preserved by the flow, Poincaré discovered the recurrence theorem: Assume the phase space has a finite Liouville volume and let F be a phase space volume-preserving map and A a subset of the phase space. Then almost every point of A returns to A infinitely often. The Poincaré recurrence theorem was used by Zermelo to object to Boltzmann's derivation of the increase in entropy in a dynamical system of colliding atoms.

One of the questions raised by Boltzmann's work was the possible equality between time averages and space averages, what he called the ergodic hypothesis. The hypothesis states that the length of time a typical trajectory spends in a region A is vol(A)/vol(Ω).

The ergodic hypothesis turned out not to be the essential property needed for the development of statistical mechanics and a series of other ergodic-like properties were introduced to capture the relevant aspects of physical systems. Koopman approached the study of ergodic systems by the use of functional analysis. An observable *a* is a function that to each point of the phase space associates a number (say instantaneous pressure, or average height). The value of an observable can be computed at another time by using the evolution function φ^{t}. This introduces an operator U^{t}, the transfer operator,

$$(U^t a)(x) = a(\Phi^{-t}(x)).$$

By studying the spectral properties of the linear operator *U* it becomes possible to classify the ergodic properties of Φ^{t}. In using the Koopman approach of considering the action of the flow on an observable function, the finite-dimensional nonlinear problem involving Φ^{t} gets mapped into an infinite-dimensional linear problem involving *U*.

The Liouville measure restricted to the energy surface Ω is the basis for the averages computed in equilibrium statistical mechanics. An average in time along a trajectory is equivalent to an average in space computed with the Boltzmann factor $\exp(-\beta H)$. This idea has been generalized by Sinai, Bowen, and Ruelle (SRB) to a larger class of dynamical systems that includes dissipative systems. SRB measures replace the Boltzmann factor and they are defined on attractors of chaotic systems.

Nonlinear dynamical systems and chaos

Simple nonlinear dynamical systems and even piecewise linear systems can exhibit a completely unpredictable behavior, which might seem to be random. (Remember that we are speaking of completely deterministic systems!). This seemingly unpredictable behavior has been called *chaos*. Hyperbolic systems are precisely defined dynamical systems that exhibit the properties ascribed to chaotic systems. In hyperbolic systems the tangent space perpendicular to a trajectory can be well separated into two parts: one with the points that converge towards the orbit (the *stable manifold*) and another of the points that diverge from the orbit (the *unstable manifold*).

This branch of mathematics deals with the long-term qualitative behavior of dynamical systems. Here, the focus is not on finding precise solutions to the equations defining the dynamical system (which is often hopeless), but rather to answer questions like "Will the system settle down to a steady state in the long term, and if so, what are the possible attractors?" or "Does the long-term behavior of the system depend on its initial condition?"

Note that the chaotic behavior of complicated systems is not the issue. Meteorology has been known for years to involve complicated—even chaotic—behavior. Chaos theory has been so surprising because chaos can be found within almost trivial systems. The logistic map is only a second-degree polynomial; the horseshoe map is piecewise linear.

Geometrical definition

A dynamical system is the tuple $\langle \mathcal{M}, f, \mathcal{T} \rangle$, with $\mathcal{M}$a manifold (locally a Banach space or Euclidean space), $\mathcal{T}$ the domain for time (non-negative reals, the integers, ...) and *f* an evolution rule $t \rightarrow f^t$ (with $t \in \mathcal{T}$) such that f^t is a diffeomorphism of the manifold to itself. So, f is a mapping of the time-domain $\mathcal{T}$ into the space of diffeomorphisms of the manifold to itself. In other terms, *f*(*t*) is a diffeomorphism, for every time *t* in the domain $\mathcal{T}$.

Measure theoretical definition

A dynamical system may be defined formally, as a measure-preserving transformation of a sigma-algebra, the quadruplet (*X*,Σ,μ,τ). Here, *X* is a set, and Σ is a sigma-algebra on *X*, so that the pair (*X*,Σ) is a measurable space. μ is a finite measure on the sigma-algebra, so that the triplet (*X*,Σ,μ) is a probability space. A map $\tau : X \rightarrow X$is said to be Σ-measurable if and only if, for every $\sigma \in \Sigma$, one has $\tau^{-1}\sigma \in \Sigma$. A map τ is said to **preserve the measure** if and only if, for every $\sigma \in \Sigma$, one has $\mu(\tau^{-1}\sigma) = \mu(\sigma)$. Combining the above, a map τ is said to be a **measure-preserving transformation of *X***, if it is a map from *X* to itself, it is Σ-measurable, and is measure-preserving. The quadruple (*X*,Σ,μ,τ), for such a τ, is then defined to be a **dynamical system**.

The map τ embodies the time evolution of the dynamical system. Thus, for discrete dynamical systems the iterates $\tau^n = \tau \circ \tau \circ \ldots \circ \tau$for integer *n* are studied. For continuous dynamical systems, the map τ is understood to be a finite time evolution map and the construction is more complicated.

Examples of dynamical systems

Internal links

- Arnold's cat map
- Baker's map is an example of a chaotic piecewise linear map
- Circle map
- Double pendulum
- Billiards and Outer Billiards
- Henon map
- Horseshoe map
- Irrational rotation
- List of chaotic maps
- Logistic map
- Lorenz system
- Rossler map

Chapter 7

Failure Rate

Failure rate is the frequency with which an engineered system or component fails, expressed for example in failures per hour. It is often denoted by the Greek letter λ (lambda) and is important in reliability engineering.

The failure rate of a system usually depends on time, with the rate varying over the life cycle of the system. For example, an automobile's failure rate in its fifth year of service may be many times greater than its failure rate during its first year of service. One does not expect to replace an exhaust pipe, overhaul the brakes, or have major transmission problems in a new vehicle.

In practice, the mean time between failures (MTBF, 1/λ) is often used instead of the failure rate. The MTBF is an important system parameter in systems where failure rate needs to be managed, in particular for safety systems. The MTBF appears frequently in the engineering design requirements, and governs frequency of required system maintenance and inspections. In special processes called renewal processes, where the time to recover from failure can be neglected and the likelihood of failure remains constant with respect to time, the failure rate is simply the multiplicative inverse of the MTBF (1/λ).

A similar ratio used in the transport industries, especially in railways and trucking is '**mean distance between failures'**, a variation which attempts to correlate actual loaded distances to similar reliability needs and practices.

Failure rates are important factors in the insurance, finance, commerce and regulatory industries and fundamental to the design of safe systems in a wide variety of applications.

Failure rate in the discrete sense

The failure rate can be defined as the following:

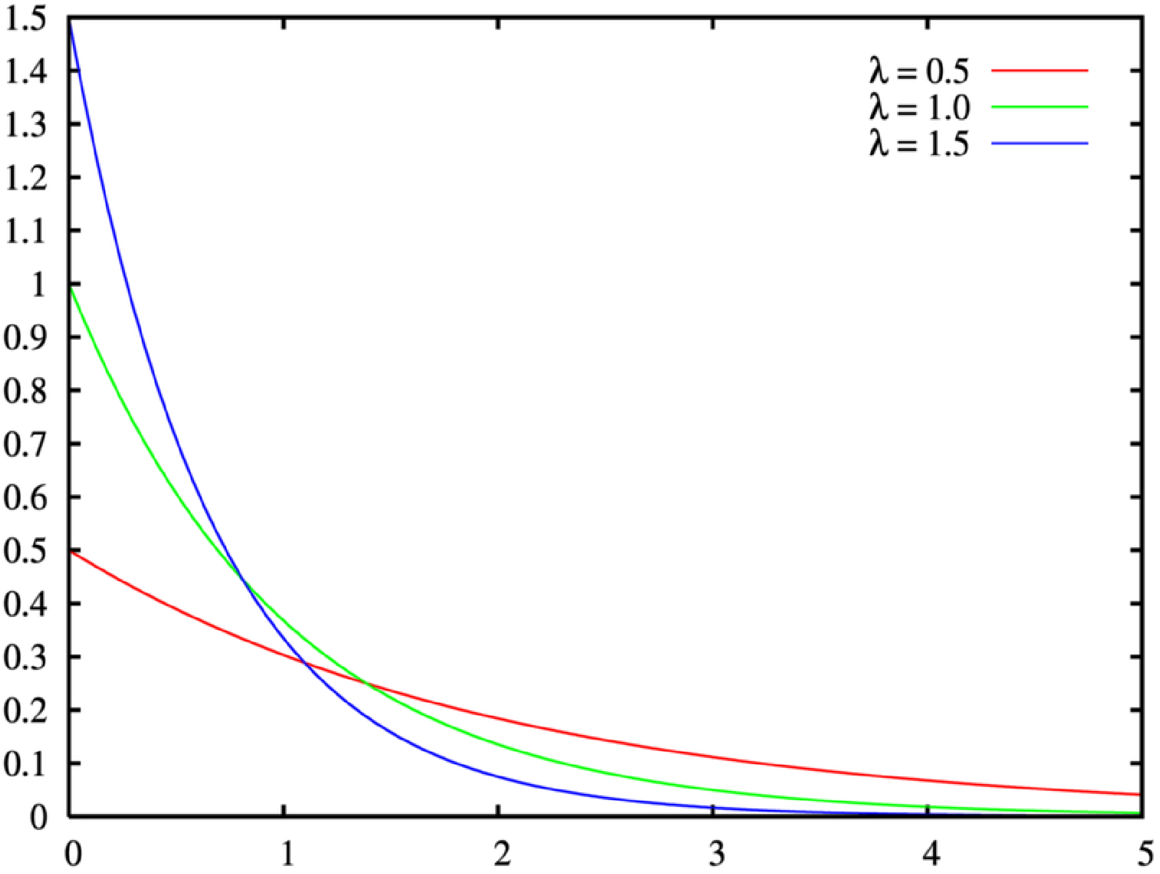
> The total number of failures within an item population, divided by the total time expended by that population, during a particular measurement interval under stated conditions. (MacDiarmid, *et al.*)

Although the failure rate, λ(t), is often thought of as the probability that a failure occurs in a specified interval given no failure before time t, it is not actually a probability because it can exceed 1. It can be defined with the aid of the reliability function or survival function $R(t)$, the probability of no failure before time t, as:

$$\lambda = \frac{R(t_1) - R(t_2)}{(t_2 - t_1) \cdot R(t_1)} = \frac{R(t) - R(t + \triangle t)}{\triangle t \cdot R(t)}$$

over a time interval ($t_2 - t_1$) from t_1 (or t) to t_2 and Δt is defined as ($t_2 - t_1$). Note that this is a conditional probability, hence the $R(t)$ in the denominator.

Failure rate in the continuous sense



Exponential failure density functions

Calculating the failure rate for ever smaller intervals of time, results in the **hazard function** (or **hazard rate**), $h(t)$. This becomes the *instantaneous* failure rate as Δ$t$tends to zero:

$$h(t) = \lim_{\triangle t \to 0} \frac{R(t) - R(t + \triangle t)}{\triangle t \cdot R(t)}.$$

A continuous failure rate depends on the existence of a **failure distribution**, $F(t)$, which is a cumulative distribution function that describes the probability of failure (at least) up to and including time *t*,

$$\Pr(T \le t) = F(t) = 1 - R(t), \quad t \ge 0.$$

where *T* is the failure time. The failure distribution function is the integral of the failure *density* function, *f*(*t*),

$$F(t) = \int_0^t f(x)\,dx.$$

The hazard function can be defined now as

$$h(t) = \frac{f(t)}{R(t)}.$$

Many probability distributions can be used to model the failure distribution. A common model is the **exponential failure distribution**,

$$F(t) = \int_0^t \lambda e^{-\lambda x}\,dx = 1 - e^{-\lambda t},$$

which is based on the exponential density function. The hazard rate function for this is:

$$h(t) = \frac{f(t)}{R(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda.$$

Thus, for an exponential failure distribution, the hazard rate is a constant with respect to time (that is, the distribution is "memoryless"). For other distributions, such as a Weibull distribution or a log-normal distribution, the hazard function may not be constant with respect to time. For some such as the deterministic distribution it is monotonic increasing (analogous to "wearing out"), for others such as the Pareto distribution it is monotonic decreasing (analogous to "burning in"), while for many it is not monotonic.

Failure rate data

Failure rate data can be obtained in several ways. The most common means are:

- Historical data about the device or system under consideration.

  Many organizations maintain internal databases of failure information on the devices or systems that they produce, which can be used to calculate failure rates for those devices or systems. For new devices or systems, the historical data for similar devices or systems can serve as a useful estimate.

- Government and commercial failure rate data.

  Handbooks of failure rate data for various components are available from government and commercial sources. MIL-HDBK-217F, *Reliability Prediction of Electronic Equipment*, is a military standard that provides failure rate data for many military electronic components. Several failure rate data sources are available commercially that focus on commercial components, including some non-electronic components.

- Testing.

  The most accurate source of data is to test samples of the actual devices or systems in order to generate failure data. This is often prohibitively expensive or impractical, so that the previous data sources are often used instead.

Units

Failure rates can be expressed using any measure of time, but **hours** is the most common unit in practice. Other units, such as miles, revolutions, etc., can also be used in place of "time" units.

Failure rates are often expressed in engineering notation as failures per million, or 10^{-6}, especially for individual components, since their failure rates are often very low.

The **Failures In Time** (**FIT**) rate of a device is the number of failures that can be expected in one billion (10^9) device-hours of operation. (E.g. 1000 devices for 1 million hours, or 1 million devices for 1000 hours each, or some other combination.) This term is used particularly by the semiconductor industry.

Additivity

Under certain engineering assumptions (e.g. besides the above assumptions for a constant failure rate, the assumption that the considered system has no relevant redundancies), the failure rate for a complex system is simply the sum of the individual failure rates of its components, as long as the units are consistent, e.g. failures per million hours. This permits testing of individual components or subsystems, whose failure rates are then added to obtain the total system failure rate.

Example

Suppose it is desired to estimate the failure rate of a certain component. A test can be performed to estimate its failure rate. Ten identical components are each tested until they either fail or reach 1000 hours, at which time the test is terminated for that component.

(The level of statistical confidence is not considered in this example.) The results are as follows:

Estimated failure rate is

$$\frac{6\text{ failures}}{7502\text{ hours}} = 0.0007998\frac{\text{failures}}{\text{hour}} = 799.8 \times 10^{-6}\frac{\text{failures}}{\text{hour}},$$

or 799.8 failures for every million hours of operation.

Estimation

The Nelson–Aalen estimator can be used to estimate the cumulative hazard rate function.

Chapter 8

Safety Engineering

Safety engineering is an applied science strongly related to systems engineering and the subset System Safety Engineering. Safety engineering assures that a life-critical system behaves as needed even when pieces fail.

Overview

Ideally, safety-engineers take an early design of a system, analyze it to find what faults can occur, and then propose safety requirements in design specifications up front and changes to existing systems to make the system safer. In an early design stage, often a fail-safe system can be made acceptably safe with a few sensors and some software to read them. Probabilistic fault-tolerant systems can often be made by using more, but smaller and less-expensive pieces of equipment.

Far too often, rather than actually influencing the design, safety engineers are assigned to prove that an existing, completed design is safe. If a safety engineer then discovers significant safety problems late in the design process, correcting them can be very expensive. This type of error has the potential to waste large sums of money.

The exception to this conventional approach is the way some large government agencies approach safety engineering from a more proactive and proven process perspective, known as "system safety". The system safety philosophy is to be applied to complex and critical systems, such as commercial airliners, complex weapon systems, spacecraft, rail and transportation systems, air traffic control system and other complex and safety-critical industrial systems. The proven system safety methods and techniques are to prevent, eliminate and control hazards and risks through designed influences by a collaboration of key engineering disciplines and product teams. Software safety is a fast growing field since modern systems functionality are increasingly being put under control of software. The whole concept of system safety and software safety, as a subset of systems engineering, is to influence safety-critical systems designs by conducting several types of hazard analyses to identify risks and to specify design safety features and procedures to strategically mitigate risk to acceptable levels before the system is certified.

Additionally, failure mitigation can go beyond design recommendations, particularly in the area of maintenance. There is an entire realm of safety and reliability engineering known as Reliability Centered Maintenance (RCM), which is a discipline that is a direct result of analyzing potential failures within a system and determining maintenance actions that can mitigate the risk of failure. This methodology is used extensively on aircraft and involves understanding the failure modes of the serviceable replaceable assemblies in addition to the means to detect or predict an impending failure. Every automobile owner is familiar with this concept when they take in their car to have the oil changed or brakes checked. Even filling up one's car with fuel is a simple example of a failure mode (failure due to fuel exhaustion), a means of detection (fuel gauge), and a maintenance action (filling the car's fuel tank).

For large scale complex systems, hundreds if not thousands of maintenance actions can result from the failure analysis. These maintenance actions are based on conditions (e.g., gauge reading or leaky valve), hard conditions (e.g., a component is known to fail after 100 hrs of operation with 95% certainty), or require inspection to determine the maintenance action (e.g., metal fatigue). The RCM concept then analyzes each individual maintenance item for its risk contribution to safety, mission, operational readiness, or cost to repair if a failure does occur. Then the sum total of all the maintenance actions are bundled into maintenance intervals so that maintenance is not occurring around the clock, but rather, at regular intervals. This bundling process introduces further complexity, as it might stretch some maintenance cycles, thereby increasing risk, but reduce others, thereby potentially reducing risk, with the end result being a comprehensive maintenance schedule, purpose built to reduce operational risk and ensure acceptable levels of operational readiness and availability.

Analysis techniques

Analysis techniques can be split into two categories: qualitative and quantitative methods. The both approaches share the goal of finding causal dependencies between an hazard on system level and failures of individual components. Qualitative approaches focus on the question "What must go wrong, such that a system hazard may occur?", while quantitative methods aim at providing estimations about probabilites, rates and/or severity of consequences.

Traditionally, safety analysis techniques rely solely on skill and expertise of the safety engineer. In the last decade model-based approaches have become prominent. In contrast to traditional methods, model-based techniques try to derive relationships between causes and consequences from some sort of model of the system.

Traditional methods for safety analysis

The two most common fault modeling techniques are called failure mode and effects analysis and fault tree analysis. These techniques are just ways of finding problems and of making plans to cope with failures, as in probabilistic risk assessment. One of the earliest complete studies using this technique on a commercial nuclear plant was the WASH-1400 study, also known as the Reactor Safety Study or the Rasmussen Report.

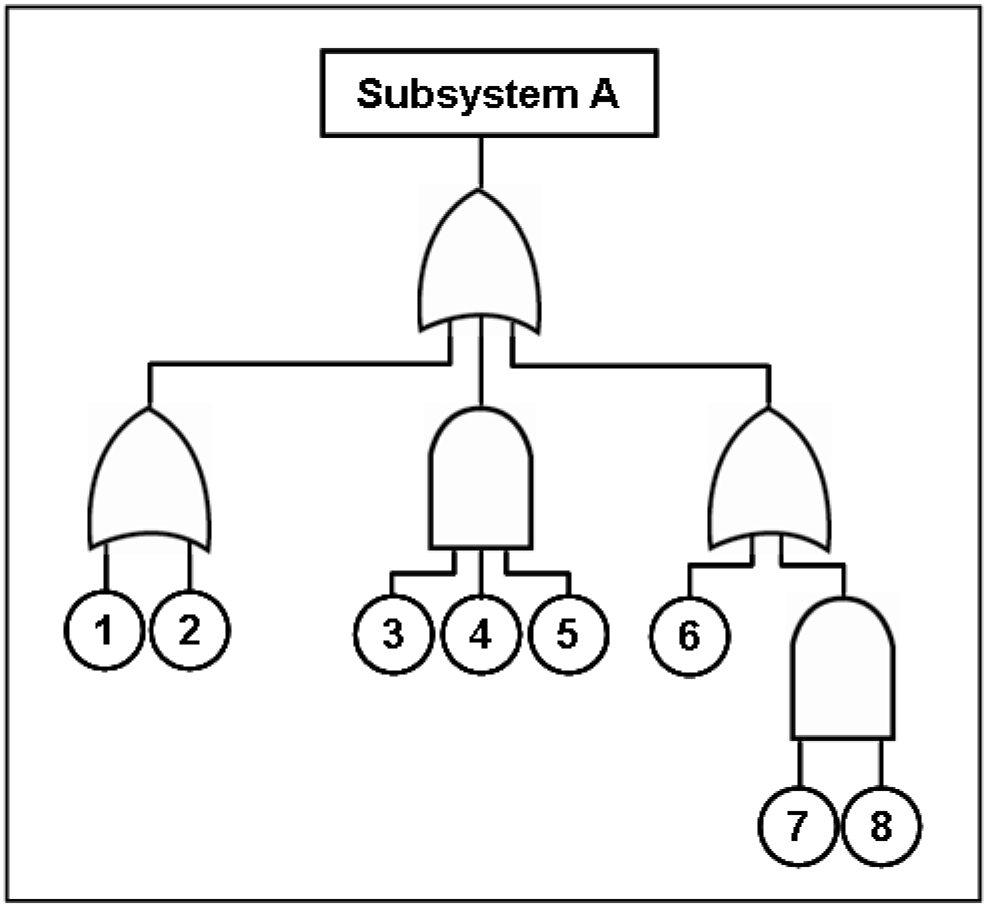
Failure modes and effects analysis

Failure Mode and Effects Analysis (FMEA) is a bottom-up, inductive analytical method which may be performed at either the functional or piece-part level. For functional FMEA, failure modes are identified for each function in a system or equipment item, usually with the help of a functional block diagram. For piece-part FMEA, failure modes are identified for each piece-part component (such as a valve, connector, resistor, or diode). The effects of the failure mode are described, and assigned a probability based on the failure rate and failure mode ratio of the function or component.

Failure modes with identical effects can be combined and summarized in a Failure Mode Effects Summary. When combined with criticality analysis, FMEA is known as Failure Mode, Effects, and Criticality Analysis or FMECA, pronounced "fuh-MEE-kuh".

Fault tree analysis

Fault tree analysis (FTA) is a top-down, deductive analytical method. In FTA, initiating primary events such as component failures, human errors, and external events are traced through Boolean logic gates to an undesired top event such as an aircraft crash or nuclear reactor core melt. The intent is to identify ways to make top events less probable, and verify that safety goals have been achieved.



A fault tree diagram

Fault trees are a logical inverse of success trees, and may be obtained by applying de Morgan's theorem to success trees (which are directly related to reliability block diagrams).

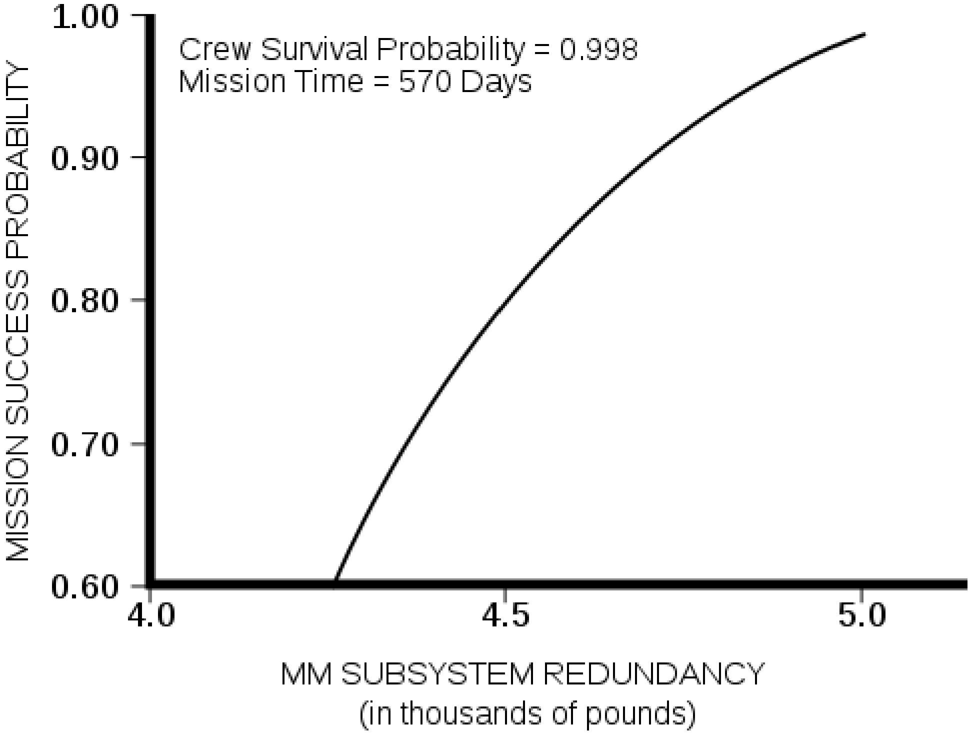
FTA may be qualitative or quantative. When failure and event probabilites are unknown, qualitative fault trees may be analyzed for minimal cut sets. For example, if any minimal cut set contains a single base event, then the top event may be caused by a single failure. Quantitative FTA is used to compute top event probability, and usually requires computer software such as CAFTA from the Electric Power Research Institute or SAPHIRE from the Idaho National Laboratory.

Some industries use both fault trees and event trees. An event tree starts from an undesired initiator (loss of critical supply, component failure etc.) and follows possible further system events through to a series of final consequences. As each new event is considered, a new node on the tree is added with a split of probabilities of taking either branch. The probabilities of a range of "top events" arising from the initial event can then be seen.

Safety certification

Usually a failure in safety-certified systems is acceptable if, on average, less than one life per 10^9 hours of continuous operation is lost to failure. Most Western nuclear reactors, medical equipment, and commercial aircraft are certified to this level. The cost versus loss of lives has been considered appropriate at this level (by FAA for aircraft under Federal Aviation Regulations).

Preventing failure



A NASA graph shows the relationship between the survival of a crew of astronauts and the amount of redundant equipment in their spacecraft (the "MM", Mission Module).

Probabilistic fault tolerance: adding redundancy to equipment and systems

Once a failure mode is identified, it can usually be prevented entirely by adding extra equipment to the system. For example, nuclear reactors contain dangerous radiation, and nuclear reactions can cause so much heat that no substance might contain them. Therefore reactors have emergency core cooling systems to keep the temperature down, shielding to contain the radiation, and engineered barriers (usually several, nested, surmounted by a containment building) to prevent accidental leakage.

Most biological organisms have a certain amount of redundancy: multiple organs, multiple limbs, etc.

For any given failure, a fail-over or redundancy can almost always be designed and incorporated into a system.

When does safety stop, where does reliability begin?

Inherent fail-safe design

When adding equipment is impractical (usually because of expense), then the least expensive form of design is often "inherently fail-safe". The typical approach is to arrange the system so that ordinary single failures cause the mechanism to shut down in a safe way (for nuclear power plants, this is termed a passively safe design, although more than ordinary failures are covered).

One of the most common fail-safe systems is the overflow tube in baths and kitchen sinks. If the valve sticks open, rather than causing an overflow and damage, the tank spills into an overflow.

Another common example is that in an elevator the cable supporting the car keeps spring-loaded brakes open. If the cable breaks, the brakes grab rails, and the elevator cabin does not fall.

Inherent fail-safes are common in medical equipment, traffic and railway signals, communications equipment, and safety equipment.

Containing failure

It is also common practice to plan for the failure of safety systems through containment and isolation methods. The use of isolating valves, also known as the block and bleed manifold, is very common in isolating pumps, tanks, and control valves that may fail or need routine maintenance. In addition, nearly all tanks containing oil or other hazardous chemicals are required to have containment barriers set up around them to contain 100% of the volume of the tank in the event of a catastrophic tank failure. Similarly, in a long pipeline, there are remote-closing valves at regular intervals so that a leak can be isolated. The goal of all containment systems is to provide means of mitigating the consequences of failure.

Chapter 9

Statistical Process Control

Statistical process control (SPC) is the application of statistical methods to the monitoring and control of a process to ensure that it operates at its full potential to produce conforming product. Under SPC, a process behaves predictably to produce as much conforming product as possible with the least possible waste. While SPC has been applied most frequently to controlling manufacturing lines, it applies equally well to any process with a measurable output. Key tools in SPC are control charts, a focus on continuous improvement and designed experiments.

Much of the power of SPC lies in the ability to examine a process and the sources of variation in that process using tools that give weight to objective analysis over subjective opinions and that allow the strength of each source to be determined numerically. Variations in the process that may affect the quality of the end product or service can be detected and corrected, thus reducing waste as well as the likelihood that problems will be passed on to the customer. With its emphasis on early detection and prevention of problems, SPC has a distinct advantage over other quality methods, such as inspection, that apply resources to detecting and correcting problems after they have occurred.

In addition to reducing waste, SPC can lead to a reduction in the time required to produce the product or service from end to end. This is partially due to a diminished likelihood that the final product will have to be reworked, but it may also result from using SPC data to identify bottlenecks, wait times, and other sources of delays within the process. Process cycle time reductions coupled with improvements in yield have made SPC a valuable tool from both a cost reduction and a customer satisfaction standpoint.

History

Statistical process control was pioneered by Walter A. Shewhart in the early 1920s. W. Edwards Deming later applied SPC methods in the United States during World War II, thereby successfully improving quality in the manufacture of munitions and other strategically important products. Deming was also instrumental in introducing SPC methods to Japanese industry after the war had ended.

Shewhart created the basis for the control chart and the concept of a state of statistical control by carefully designed experiments. While Dr. Shewhart drew from pure mathematical statistical theories, he understood that data from physical processes seldom produces a "normal distribution curve" (a Gaussian distribution, also commonly referred to as a "bell curve"). He discovered that observed variation in manufacturing data did not always behave the same way as data in nature (for example, Brownian motion of particles). Dr. Shewhart concluded that while every process displays variation, some processes display controlled variation that is natural to the process (common causes of variation), while others display uncontrolled variation that is not present in the process causal system at all times (special causes of variation).

In 1988, the Software Engineering Institute introduced the notion that SPC can be usefully applied to non-manufacturing processes, such as software engineering processes, in the Capability Maturity Model (CMM). This idea exists today within the Level 4 and Level 5 practices of the Capability Maturity Model Integration (CMMI). This notion that SPC is a useful tool when applied to non-repetitive, knowledge-intensive processes such as engineering processes has encountered much skepticism, and remains controversial today.

General

The following description relates to manufacturing rather than to the service industry, although the principles of SPC can be successfully applied to either. For a description and example of how SPC applies to a service environment, refer to Roberts (2005). SPC has also been successfully applied to detecting changes in organizational behavior with Social Network Change Detection introduced by McCulloh (2007). Selden describes how to use SPC in the fields of sales, marketing, and customer service, using Deming's famous Red Bead Experiment as an easy to follow demonstration.

In mass-manufacturing, the quality of the finished article was traditionally achieved through post-manufacturing inspection of the product; accepting or rejecting each article (or samples from a production lot) based on how well it met its design specifications. In contrast, Statistical Process Control uses statistical tools to observe the performance of the production process in order to predict significant deviations that may later result in rejected product.

Two kinds of variation occur in all manufacturing processes: both these types of process variation cause subsequent variation in the final product. The first is known as natural or common cause variation and consists of the variation inherent in the process as it is designed. Common cause variation may include variations in temperature, properties of raw materials, strength of an electrical current etc. The second kind of variation is known as special cause variation, or assignable-cause variation, and happens less frequently than the first. With sufficient investigation, a specific cause, such as abnormal raw material or incorrect set-up parameters, can be found for special cause variations.

For example, a breakfast cereal packaging line may be designed to fill each cereal box with 500 grams of product, but some boxes will have slightly more than 500 grams, and some will have slightly less, in accordance with a distribution of net weights. If the production process, its inputs, or its environment changes (for example, the machines doing the manufacture begin to wear) this distribution can change. For example, as its cams and pulleys wear out, the cereal filling machine may start putting more cereal into each box than specified. If this change is allowed to continue unchecked, more and more product will be produced that fall outside the tolerances of the manufacturer or consumer, resulting in waste. While in this case, the waste is in the form of "free" product for the consumer, typically waste consists of rework or scrap.

By observing at the right time what happened in the process that led to a change, the quality engineer or any member of the team responsible for the production line can troubleshoot the root cause of the variation that has crept in to the process and correct the problem.

SPC indicates when an action should be taken in a process, but it also indicates when NO action should be taken. An example is a person who would like to maintain a constant body weight and takes weight measurements weekly. A person who does not understand SPC concepts might start dieting every time his or her weight increased, or eat more every time his or her weight decreased. This type of action could be harmful and possibly generate even more variation in body weight. SPC would account for normal weight variation and better indicate when the person is in fact gaining or losing weight.

How to Use SPC

Statistical Process Control may be broadly broken down into three sets of activities: understanding the process, understanding the causes of variation, and elimination of the sources of special cause variation.

In understanding a process, the process is typically mapped out and the process is monitored using control charts. Control charts are used to identify variation that may be due to special causes, and to free the user from concern over variation due to common causes. This is a continuous, ongoing activity. When a process is stable and does not trigger any of the detection rules for a control chart, a process capability analysis may also be performed to predict the ability of the current process to produce conforming (i.e. within specification) product in the future.

When excessive variation is identified by the control chart detection rules, or the process capability is found lacking, additional effort is exerted to determine causes of that variance. The tools used include Ishikawa diagrams, designed experiments and Pareto charts. Designed experiments are critical to this phase of SPC, as they are the only means of objectively quantifying the relative importance of the many potential causes of variation.

Once the causes of variation have been quantified, effort is spent in eliminating those causes that are both statistically and practically significant (i.e. a cause that has only a small but statistically significant effect may not be considered cost-effective to fix; however, a cause that is not statistically significant can never be considered practically significant). Generally, this includes development of standard work, error-proofing and training. Additional process changes may be required to reduce variation or align the process with the desired target, especially if there is a problem with process capability.

For digital SPC charts, so-called SPC rules usually come with some rule specific logic that determines a 'derived value' that is to be used as the basis for some (setting) correction. One example of such a derived value would be (for the common N numbers in a row ranging up or down 'rule'); derived value = last value + average difference between the last N numbers (which would, in effect, be extending the row with the to be expected next value).

Chapter 10

Optimal Design

Optimal designs are a class of experimental designs that are optimal with respect to some statistical criterion.

In the design of experiments for estimating statistical models, **optimal designs** allow parameters to be estimated without bias and with minimum-variance. A non-optimal design requires a greater number of experimental runs to estimate the parameters with the same precision as an optimal design. In practical terms, optimal experiments can reduce the costs of experimentation.

The optimality of a design depends on the statistical model and is assessed with respect to a statistical criterion, which is related to the variance-matrix of the estimator. Specifying an appropriate model and specifying a suitable criterion function both require understanding of statistical theory and practical knowledge with designing experiments.

Optimal designs are also called **optimum designs**.

Advantages of optimal designs

Optimal designs offer three advantages over suboptimal experimental designs:

1. Optimal designs reduce the costs of experimentation by allowing statistical models to be estimated with fewer experimental runs.
2. Optimal designs can accommodate multiple types of factors, such as process, mixture, and discrete factors.
3. Designs can be optimized when the design-space is constrained, for example, when the mathematical process-space contains factor-settings that are practically infeasible (e.g. due to safety concerns).

Minimizing the variance of estimators

Experimental designs are evaluated using statistical criteria.

It is known that the least squares estimator minimizes the variance of mean-unbiased estimators (under the conditions of the Gauss–Markov theorem). In the estimation theory for statistical models with one real parameter, the reciprocal of the variance of an ("efficient") estimator is called the "Fisher information" for that estimator. Because of this reciprocity, ***minimizing* the *variance*** corresponds to ***maximizing* the *information***.

When the statistical model has several parameters, however, the mean of the parameter-estimator is a vector and its variance is a matrix. The inverse matrix of the variance-matrix is called the "information matrix". Because the variance of the estimator of a parameter vector is a matrix, the problem of "minimizing the variance" is complicated. Using statistical theory, statisticians compress the information-matrix using real-valued summary statistics; being real-valued functions, these "information criteria" can be maximized. The traditional optimality-criteria are invariants of the information matrix; algebraically, the traditional optimality-criteria are functionals of the eigenvalues of the information matrix.

- **A**-optimality ("**average**" or **trace**)
    - One criterion is **A-optimality**, which seeks to minimize the trace of the inverse of the information matrix. This criterion results in minimizing the average variance of the estimates of the regression coefficients.

- **C**-optimality

- **D**-optimality (**determinant**)
    - A popular criterion is **D-optimality**, which seeks to minimize $|(X'X)^{-1}|$, or equivalently maximize the determinant of the information matrix X'X of the design. This criterion results in maximizing the differential Shannon information content of the parameter estimates.

- **E**-optimality (**eigenvalue**)
    - Another design is **E-optimality**, which maximizes the minimum eigenvalue of the information matrix. The E-optimality criterion need not be differentiable at every point. Such E-optimal designs can be computed using methods of convex minimization that use *subgradients* rather than gradients at points of non-differentiability. Any non-differentiability need not be a serious problem, however: E-optimality problems are special cases of semidefinite-programming problems which have effective solution-methods, especially bundle methods and interior-point methods.

- **T**-optimality
    - This criterion maximizes the trace of the information matrix.

Other optimality-criteria are concerned with the variance of predictions:

- **G**-optimality

  - A popular criterion is **G-optimality**, which seeks to minimize the maximum entry in the diagonal of the hat matrix $X(X'X)^{-1}X'$. This has the effect of minimizing the maximum variance of the predicted values.

- **I**-optimality (**integrated**)
  - A second criterion on prediction variance is **I-optimality**, which seeks to minimize the average prediction variance *over the design space*.

- **V**-optimality (**variance**)
  - A third criterion on prediction variance is **V-optimality**, which seeks to minimizes the average prediction variance over a set of m specific points.

Contrasts

In many applications, the statistician is most concerned with a "parameter of interest" rather than with "nuisance parameters". More generally, statisticians consider linear combinations of parameters, which are estimated via linear combinations of treatment-means in the design of experiments and in the analysis of variance; such linear combinations are called contrasts. Statisticians can use appropriate optimality-criteria for such parameters of interest and for more generally for contrasts.

Finding optimal designs

Catalogs of optimal designs occur in books and in software libraries.

In addition, major statistical systems like SAS and R have procedures for optimizing a design according to a user's specification. The experimenter must specify a model for the design and an optimality-criterion before the method can compute an optimal design.

Practical considerations

Some advanced topics in optimal design require more statistical theory and practical knowledge in designing experiments.

Model dependence and robustness

Since the optimality criterion of most optimal designs is based on some function of the information matrix, the 'optimality' of a given design is *model dependent*: While an optimal design is best for that model, its performance may deteriorate on other models. On other models, an *optimal* design can be either better or worse than a non-optimal design. Therefore, it is important to benchmark the performance of designs under alternative models.

Choosing an optimality criterion and robustness

The choice of an appropriate optimality criterion requires some thought, and it is useful to benchmark the performance of designs with respect to several optimality criteria. Cornell writes that

since the [traditional optimality] criteria . . . are variance-minimizing criteria, . . . a design that is optimal for a given model using one of the . . . criteria is usually near-optimal for the same model with respect to the other criteria.

Indeed, there are several classes of designs for which all the traditional optimality-criteria agree, according to the theory of "universal optimality" of Kiefer. The experience of practitioners like Cornell and the "universal optimality" theory of Kiefer suggest that robustness with respect to changes in the *optimality-criterion* is much greater than is robustness with respect to changes in the *model*.

Flexible optimality criteria and convex analysis

High-quality statistical software provide a combination of libraries of optimal designs or iterative methods for constructing approximately optimal designs, depending on the model specified and the optimality criterion. Users may use a standard optimality-criterion or may program a custom-made criterion.

All of the traditional optimality-criteria are convex (or concave) functions, and therefore optimal-designs are amenable to the mathematical theory of convex analysis and their computation can use specialized methods of convex minimization. The practitioner need not select *exactly one* traditional, optimality-criterion, but can specify a custom criterion. In particular, the practitioner can specify a convex criterion using the maxima of convex optimality-criteria and nonnegative combinations of optimality criteria (since these operations preserve convex functions). For *convex* optimality criteria, the Kiefer-Wolfowitz equivalence theorem allows the practitioner to verify that a given design is globally optimal. The Kiefer-Wolfowitz equivalence theorem is related with the Legendre-Fenchel conjugacy for convex functions.

If an optimality-criterion lacks convexity, then finding a global optimum and verifying its optimality often are difficult.

Model uncertainty and Bayesian approaches

Model selection

When scientists wish to test several theories, then a statistician can design an experiment that allows optimal tests between specified models. Such "discrimination experiments" are especially important in the biostatistics supporting pharmacokinetics and pharmacodynamics, following the work of Cox and Atkinson.

Bayesian experimental design

When practitioners need to consider multiple models, they can specify a probability-measure on the models and then select any design maximizing the expected value of such an experiment. Such probability-based optimal-designs are called optimal Bayesian designs. Such Bayesian designs are used especially for generalized linear models (where the response follows an exponential-family distribution).

The use of a Bayesian design does not force statisticians to use Bayesian methods to analyze the data, however. Indeed, the "Bayesian" label for probability-based experimental-designs is disliked by some researchers. Alternative terminology for "Bayesian" optimality includes "on-average" optimality or "population" optimality.

Iterative experimentation

Scientific experimentation is an iterative process, and statisticians have developed several approaches to the optimal design of sequential experiments.

Sequential analysis

Sequential analysis was pioneered by Abraham Wald. In 1972, Herman Chernoff wrote an overview of optimal sequential designs, while adaptive designs were surveyed later by S. Zacks. Of course, much work on the optimal design of experiments is related to the theory of optimal decisions, especially the statistical decision theory of Abraham Wald.

Response-surface methodology

Optimal designs for response-surface models are discussed in the textbook by Atkinson, Donev and Tobias, and in the survey of Gaffke and Heiligers and in the mathematical text of Pukelsheim. The blocking of optimal designs is discussed in the textbook of Atkinson, Donev and Tobias and also in the monograph by Goos.

The earliest optimal designs were developed to estimate the parameters of regression models with continuous variables, for example, by J. D. Gergonne in 1815 (Stigler). In English, two early contributions were made by Charles S. Peirce and Kirstine Smith.

Pioneering designs for multivariate response-surfaces were proposed by George E. P. Box. However, Box's designs have few optimality properties. Indeed, the Box-Behnken design requires excessive experimental runs when the number of variables exceeds three. Box's "central-composite" designs require more experimental runs than do the optimal designs of Kôno.

System identification and stochastic approximation

The optimization of sequential experimentation is studied also in stochastic programming and in systems and control. Popular methods include stochastic approximation and other methods of stochastic optimization. Much of this research has been associated with the subdiscipline of system identification. In computational optimal control, D. Judin & A. Nemirovskii and Boris Polyak has described methods that are more efficient than the (Armijo-style) step-size rules introduced by G. E. P. Box in response-surface methodology

Adaptive designs are used in clinical trials, and optimal adaptive designs are surveyed in the *Handbook of Experimental Designs* chapter by Shelemyahu Zacks.

Specifying the number of experimental runs

Using a computer to find a good design

There are several methods of finding an optimal design, given an *a priori* restriction on the number of experimental runs or replications. Some of these methods are discussed by Atkinson, Donev and Tobias and in the paper by Hardin and Sloane. Of course, fixing the number of experimental runs *a priori* would be impractical. Prudent statisticians examine the other optimal designs, whose number of experimental runs differ.

Discretizing probability-measure designs

In the mathematical theory on optimal experiments, an optimal design can be a probability measure that is supported on an infinite set of observation-locations. Such optimal probability-measure designs solve a mathematical problem that neglected to specify the cost of observations and experimental runs. Nonetheless, such optimal probability-measure designs can be discretized to furnish approximately optimal designs.

In some cases, a finite set of observation-locations suffices to support an optimal design. Such a result was proved by Kôno and Kiefer in their works on response-surface designs for quadratic models. The Kôno-Kiefer analysis explains why optimal designs for response-surfaces can have discrete supports, which are very similar as do the less efficient designs that have been traditional in response surface methodology.

History

The prophet of scientific experimentation, Francis Bacon, foresaw that experimental designs should be improved. Researchers who improved experiments were praised in Bacon's utopian novel *New Atlantis*:

Then after divers meetings and consults of our whole number, to consider of the former labors and collections, we have three that take care out of them to direct new experiments, of a higher light, more penetrating into nature than the former. These we call lamps.

In 1815, an article on optimal designs for polynomial regression was published by Joseph Diaz Gergonne, according to Stigler.

Charles S. Peirce proposed an economic theory of scientific experimentation in 1876, which sought to maximize the precision of the estimates. Peirce's optimal allocation immediately improved the accuracy of gravitational experiments and was used for decades by Peirce and his colleagues. In his 1882 published lecture at Johns Hopkins University, Peirce introduced experimental design with these words:

Logic will not undertake to inform you what kind of experiments you ought to make in order best to determine the acceleration of gravity, or the value of the Ohm; but it will tell you how to proceed to form a plan of experimentation.

[....] Unfortunately practice generally precedes theory, and it is the usual fate of mankind to get things done in some boggling way first, and find out afterward how they could have been done much more easily and perfectly.

Like Bacon, Peirce was aware that experimental methods should strive for substantial improvement (even optimality).

Kirstine Smith proposed optimal designs for polynomial models in 1918. (Kirstine Smith had been a student of the Danish statistician Thorvald N. Thiele and was working with Karl Pearson in London.)

Chapter 11

Estimation Theory

Estimation theory is a branch of statistics and signal processing that deals with estimating the values of parameters based on measured/empirical data that has a random component. The parameters describe an underlying physical setting in such a way that the value of the parameters affects the distribution of the measured data. An estimator attempts to approximate the unknown parameters using the measurements.

For example, it is desired to estimate the proportion of a population of voters who will vote for a particular candidate. That proportion is the unobservable parameter; the estimate is based on a small random sample of voters.

Or, for example, in radar the goal is to estimate the range of objects (airplanes, boats, etc.) by analyzing the two-way transit timing of received echoes of transmitted pulses. Since the reflected pulses are unavoidably embedded in electrical noise, their measured values are randomly distributed, so that the transit time must be estimated.

In estimation theory, it is assumed the measured data is random with probability distribution dependent on the parameters of interest. For example, in electrical communication theory, the measurements which contain information regarding the parameters of interest are often associated with a noisy signal. Without randomness, or noise, the problem would be deterministic and estimation would not be needed.

Estimation process

The entire purpose of estimation theory is to arrive at an estimator, and preferably an implementable one that could actually be used. The estimator takes the measured data as input and produces an estimate of the parameters.

It is also preferable to derive an estimator that exhibits optimality. Estimator optimality usually refers to achieving minimum average error over some class of estimators, for example, a minimum variance unbiased estimator. In this case, the class is the set of unbiased estimators, and the average error measure is variance (average squared error between the value of the estimate and the parameter). However, optimal estimators do not always exist.

These are the general steps to arrive at an estimator:

- In order to arrive at a desired estimator, it is first necessary to determine a probability distribution for the measured data, and the distribution's dependence on the unknown parameters of interest. Often, the probability distribution may be derived from physical models that explicitly show how the measured data depends on the parameters to be estimated, and how the data is corrupted by random errors or noise. In other cases, the probability distribution for the measured data is simply "assumed", for example, based on familiarity with the measured data and/or for analytical convenience.
- After deciding upon a probabilistic model, it is helpful to find the limitations placed upon an estimator. This limitation, for example, can be found through the Cramér–Rao bound.
- Next, an estimator needs to be developed or applied if an already known estimator is valid for the model. The estimator needs to be tested against the limitations to determine if it is an optimal estimator (if so, then no other estimator will perform better).
- Finally, experiments or simulations can be run using the estimator to test its performance.

After arriving at an estimator, real data might show that the model used to derive the estimator is incorrect, which may require repeating these steps to find a new estimator. A non-implementable or infeasible estimator may need to be scrapped and the process started anew.

In summary, the estimator estimates the parameters of a physical model based on measured data.

Teori Estimasi

Basics

To build a model, several statistical "ingredients" need to be known. These are needed to ensure the estimator has some mathematical tractability instead of being based on "good feel".

The first is a set of statistical samples taken from a random vector (RV) of size *N*. Put into a vector,

$$\mathbf{x} = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}.$$

Secondly, we have the corresponding *M* parameters

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_M \end{bmatrix},$$

which need to be established with their probability density function (pdf) or probability mass function (pmf)

$$p(\mathbf{x}|\theta).$$

It is also possible for the parameters themselves to have a probability distribution (e.g., Bayesian statistics). It is then necessary to define the Bayesian probability

$$\pi(\theta).$$

After the model is formed, the goal is to estimate the parameters, commonly denoted $\hat{\theta}$, where the "hat" indicates the estimate.

One common estimator is the minimum mean squared error (MMSE) estimator, which utilizes the error between the estimated parameters and the actual value of the parameters

$$\mathbf{e} = \hat{\theta} - \theta$$

as the basis for optimality. This error term is then squared and minimized for the MMSE estimator.

Estimators

Commonly-used estimators and estimation methods, and topics related to them:

- Maximum likelihood estimators
- Bayes estimators
- Method of moments estimators
- Cramér–Rao bound
- Minimum mean squared error (MMSE), also known as Bayes least squared error (BLSE)
- Maximum a posteriori (MAP)
- Minimum variance unbiased estimator (MVUE)
- Best linear unbiased estimator (BLUE)
- Unbiased estimators
- Particle filter
- Markov chain Monte Carlo (MCMC)

- Kalman filter
- Ensemble Kalman filter (EnKF)
- Wiener filter

Examples

Unknown constant in additive white Gaussian noise

Consider a received discrete signal, *x*[*n*], of *N* independent samples that consists of an unknown constant *A* with additive white Gaussian noise (AWGN) *w*[*n*] with known variance σ^2 (*i.e.*, $\mathcal{N}(0,\sigma^2)$). Since the variance is known then the only unknown parameter is *A*.

The model for the signal is then

$$x[n] = A + w[n] \quad n = 0, 1, \dots, N-1$$

Two possible (of many) estimators are:

- $\hat{A}_1 = x[0]$
- $\hat{A}_2 = \frac{1}{N}\sum_{n=0}^{N-1} x[n]$ which is the sample mean

Both of these estimators have a mean of *A*, which can be shown through taking the expected value of each estimator

$$\mathrm{E}\left[\hat{A}_1\right] = \mathrm{E}\left[x[0]\right] = A$$

and

$$\mathrm{E}\left[\hat{A}_2\right] = \mathrm{E}\left[\frac{1}{N}\sum_{n=0}^{N-1} x[n]\right] = \frac{1}{N}\left[\sum_{n=0}^{N-1} \mathrm{E}\left[x[n]\right]\right] = \frac{1}{N}\left[NA\right] = A$$

At this point, these two estimators would appear to perform the same. However, the difference between them becomes apparent when comparing the variances.

$$\mathrm{var}\left(\hat{A}_1\right) = \mathrm{var}\left(x[0]\right) = \sigma^2$$

and

$$\mathrm{var}\left(\hat{A}_2\right) = \mathrm{var}\left(\frac{1}{N}\sum_{n=0}^{N-1} x[n]\right) \overset{independence}{=} \frac{1}{N^2}\left[\sum_{n=0}^{N-1} \mathrm{var}(x[n])\right] = \frac{1}{N^2}\left[N\sigma^2\right] = \frac{\sigma^2}{N}$$

It would seem that the sample mean is a better estimator since, as $N \to \infty$, the variance goes to zero.

Maximum likelihood

Continuing the example using the maximum likelihood estimator, the probability density function (pdf) of the noise for one sample *w*[*n*] is

$$p(w[n]) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} w[n]^2\right)$$

and the probability of *x*[*n*] becomes (*x*[*n*] can be thought of a $\mathcal{N}(A, \sigma^2)$)

$$p(x[n]; A) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x[n] - A)^2\right)$$

By independence, the probability of $\mathbf{x}$becomes

$$p(\mathbf{x}; A) = \prod_{n=0}^{N-1} p(x[n]; A) = \frac{1}{\left(\sigma\sqrt{2\pi}\right)^N} \exp\left(-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n] - A)^2\right)$$

Taking the natural logarithm of the pdf

$$\ln p(\mathbf{x}; A) = -N\ln\left(\sigma\sqrt{2\pi}\right) - \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n] - A)^2$$

and the maximum likelihood estimator is

$$\hat{A} = \arg\max \ln p(\mathbf{x}; A)$$

Taking the first derivative of the log-likelihood function

$$\frac{\partial}{\partial A}\ln p(\mathbf{x}; A) = \frac{1}{\sigma^2}\left[\sum_{n=0}^{N-1}(x[n] - A)\right] = \frac{1}{\sigma^2}\left[\sum_{n=0}^{N-1} x[n] - NA\right]$$

and setting it to zero

$$0 = \frac{1}{\sigma^2}\left[\sum_{n=0}^{N-1} x[n] - NA\right] = \sum_{n=0}^{N-1} x[n] - NA$$

This results in the maximum likelihood estimator

$$\hat{A} = \frac{1}{N}\sum_{n=0}^{N-1} x[n]$$

which is simply the sample mean. From this example, it was found that the sample mean is the maximum likelihood estimator for *N* samples of a fixed, unknown parameter corrupted by AWGN.

Cramér–Rao lower bound

To find the Cramér–Rao lower bound (CRLB) of the sample mean estimator, it is first necessary to find the Fisher information number

$$\mathcal{I}(A) = \mathbf{E}\left(\left[\frac{\partial}{\partial\theta}\ln p(\mathbf{x}; A)\right]^2\right) = -\mathbf{E}\left[\frac{\partial^2}{\partial\theta^2}\ln p(\mathbf{x}; A)\right]$$

and copying from above

$$\frac{\partial}{\partial A}\ln p(\mathbf{x}; A) = \frac{1}{\sigma^2}\left[\sum_{n=0}^{N-1} x[n] - NA\right]$$

Taking the second derivative

$$\frac{\partial^2}{\partial A^2}\ln p(\mathbf{x}; A) = \frac{1}{\sigma^2}(-N) = \frac{-N}{\sigma^2}$$

and finding the negative expected value is trivial since it is now a deterministic constant

$$-\mathbf{E}\left[\frac{\partial^2}{\partial A^2}\ln p(\mathbf{x}; A)\right] = \frac{N}{\sigma^2}$$

Finally, putting the Fisher information into

$$\mathrm{var}\left(\hat{A}\right) \geq \frac{1}{\mathcal{I}}$$

results in

$$\operatorname{var}\left(\hat{A}\right) \geq \frac{\sigma^2}{N}$$

Comparing this to the variance of the sample mean (determined previously) shows that the sample mean is *equal to* the Cramér–Rao lower bound for all values of *N* and *A*. In other words, the sample mean is the (necessarily unique) efficient estimator, and thus also the minimum variance unbiased estimator (MVUE), in addition to being the maximum likelihood estimator.

Maximum of a uniform distribution

One of the simplest non-trivial examples of estimation is the estimation of the maximum of a uniform distribution. It is used as a hands-on classroom exercise and to illustrate basic principles of estimation theory. Further, in the case of estimation based on a single sample, it demonstrates philosophical issues and possible misunderstandings in the use of maximum likelihood estimators and likelihood functions.

Given a discrete uniform distribution $1, 2, \dots, N$ with unknown maximum, the UMVU estimator for the maximum is given by

$$\frac{k+1}{k}m - 1 = m + \frac{m}{k} - 1$$

where *m* is the sample maximum and *k* is the sample size, sampling without replacement. This problem is commonly known as the German tank problem, due to application of maximum estimation to estimates of German tank production during World War II.

The formula may be understood intuitively as:

> "The sample maximum plus the average gap between observations in the sample",

the gap being added to compensate for the negative bias of the sample maximum as an estimator for the population maximum.

This has a variance of

$$\frac{1}{k}\frac{(N-k)(N+1)}{(k+2)} \approx \frac{N^2}{k^2} \text{ for small samples } k \ll N$$

so a standard deviation of approximately *N* / *k*, the (population) average size of a gap between samples; compare $\frac{m}{k}$ above. This can be seen as a very simple case of maximum spacing estimation.

The sample maximum is the maximum likelihood estimator for the population maximum, but, as discussed above, it is biased.

Applications

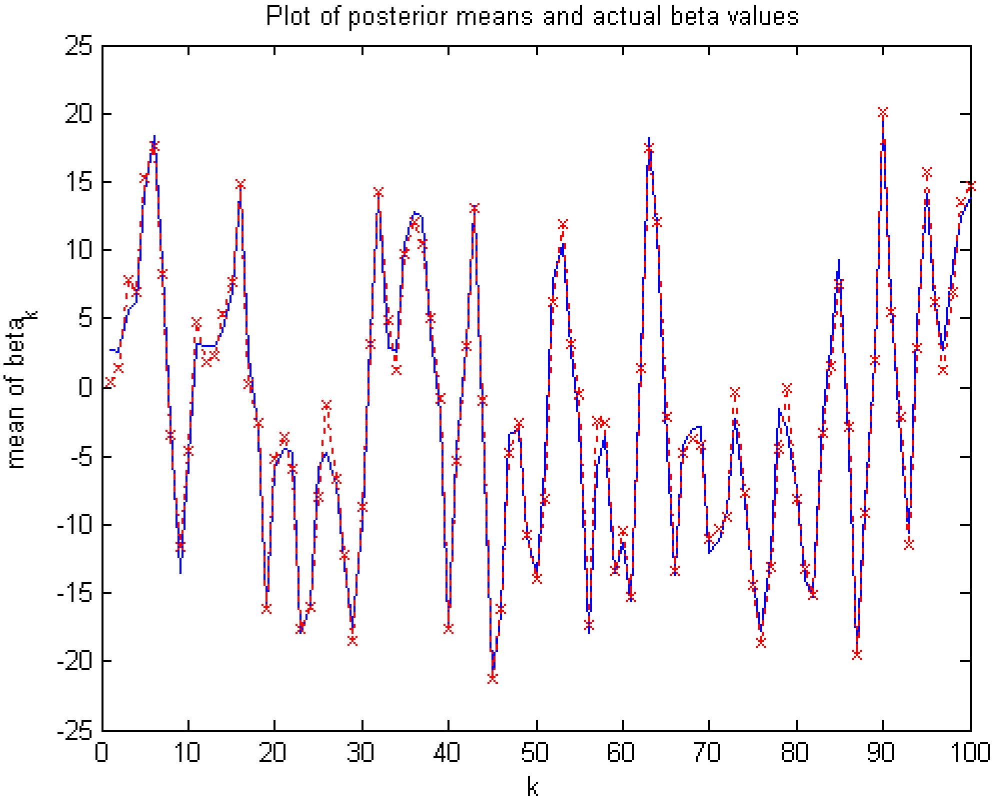
Numerous fields require the use of estimation theory. Some of these fields include (but are by no means limited to):

- Interpretation of scientific experiments
- Signal processing
- Clinical trials
- Opinion polls
- Quality control
- Telecommunications
- Project management
- Software engineering
- Control theory
- Network intrusion detection system
- Orbit determination

Measured data are likely to be subject to noise or uncertainty and it is through statistical probability that optimal solutions are sought to extract as much information from the data as possible.

Chapter 12

Particle Filter



Result of particle filtering (red line) based on observed data generated from the blue line (Much larger image)

In statistics, **particle filters**, also known as **Sequential Monte Carlo methods** (SMC), are sophisticated model estimation techniques based on simulation. Particle filters have important applications in econometrics.

Particle filters are usually used to estimate Bayesian models in which the latent variables are connected in a Markov chain — similar to a hidden Markov model (HMM), but typically where the state space of the latent variables is continuous rather than discrete, and not sufficiently restricted to make exact inference tractable (as, for example, in a linear dynamical system, where the state space of the latent variables is restricted to Gaussian distributions and hence exact inference can be done efficiently using a Kalman filter). In the context of HMMs and related models, "filtering" refers to determining the distribution of a latent variable at a specific time, given all observations up to that time; particle filters are so named because they allow for approximate "filtering" (in the sense just given) using a set of "particles" (differently-weighted samples of the distribution).

Particle filters are the sequential ("on-line") analogue of Markov chain Monte Carlo (MCMC) batch methods and are often similar to importance sampling methods. Well-designed particle filters can often be much faster than MCMC. They are often an alternative to the Extended Kalman filter (EKF) or Unscented Kalman filter (UKF) with the advantage that, with sufficient samples, they approach the Bayesian optimal estimate, so they can be made more accurate than either the EKF or UKF. However, when the simulated sample is not sufficiently large, they might suffer from sample impoverishment. The approaches can also be combined by using a version of the Kalman filter as a proposal distribution for the particle filter.

Goal

The particle filter aims to estimate the sequence of hidden parameters, x_k for $k = 0,1,2,3,\ldots$, based only on the observed data y_k for $k = 0,1,2,3,\ldots$. All Bayesian estimates of x_k follow from the posterior distribution $p(x_k \mid y_0,y_1,\ldots,y_k)$. In contrast, the MCMC or importance sampling approach would model the full posterior $p(x_0,x_1,\ldots,x_k \mid y_0,y_1,\ldots,y_k)$.

Model

Particle methods assume x_k and the observations y_k can be modeled in this form:

- $x_0, x_1, \cdots$ is a first order Markov process such that

  $$x_k|x_{k-1} \sim p_{x_k|x_{k-1}}(x|x_{k-1})$$

and with an initial distribution $p(x_0)$.

- The observations $y_0, y_1, \cdots$ are conditionally independent provided that $x_0, x_1, \cdots$ are known

  In other words, each y_k only depends on x_k
  $$y_k|x_k \sim p_{y|x}(y|x_k)$$

One example form of this scenario is

$$x_k = f(x_{k-1}) + w_k$$
$$y_k = h(x_k) + v_k$$

where both w_k and v_k are mutually independent and identically distributed sequences with known probability density functions and $f(\cdot)$and $h(\cdot)$are known functions. These two equations can be viewed as state space equations and look similar to the state space equations for the Kalman filter. If the functions $f(\cdot)$and $h(\cdot)$are linear, and if both w_k and v_k are Gaussian, the Kalman filter finds the exact Bayesian filtering distribution. If not, Kalman filter based methods are a first-order approximation (EKF) or a second-order approximation (UKF in general, but if probability distribution is Gaussian a third-order approximation is possible). Particle filters are also an approximation, but with enough particles they can be much more accurate.

Monte Carlo approximation

Particle methods, like all sampling-based approaches (e.g., MCMC), generate a set of samples that approximate the filtering distribution $p(x_k|y_0, \ldots, y_k)$. So, with *P* samples, expectations with respect to the filtering distribution are approximated by

$$\int f(x_k)p(x_k|y_0, \ldots, y_k)dx_k \approx \frac{1}{P}\sum_{L=1}^{P} f(x_k^{(L)})$$

and $f(\cdot)$, in the usual way for Monte Carlo, can give all the moments etc. of the distribution up to some degree of approximation.

Sequential Importance Resampling (SIR)

Sequential importance resampling (SIR), the original particle filtering algorithm (Gordon et al. 1993), is a very commonly used particle filtering algorithm, which approximates the filtering distribution $p(x_k|y_0, \ldots, y_k)$by a weighted set of P particles

$$\{(w_k^{(L)}, x_k^{(L)}) \; : \; L \in \{1, \ldots, P\}\}.$$

The *importance weights* $w_k^{(L)}$ are approximations to the relative posterior probabilities (or densities) of the particles such that $\sum_{L=1}^{P} w_k^{(L)} = 1$.

SIR is a sequential (i.e., recursive) version of importance sampling. As in importance sampling, the expectation of a function $f(\cdot)$can be approximated as a weighted average

$$\int f(x_k)p(x_k|y_0,\dots,y_k)dx_k \approx \sum_{L=1}^{P} w^{(L)} f(x_k^{(L)}).$$

For a finite set of particles, the algorithm performance is dependent on the choice of the *proposal distribution*

$$\pi(x_k|x_{0:k-1},y_{0:k}).$$

The *optimal proposal distribution* is given as the *target distribution*

$$\pi(x_k|x_{0:k-1},y_{0:k}) = p(x_k|x_{k-1},y_k).$$

However, the transition prior is often used as importance function, since it is easier to draw particles (or samples) and perform subsequent importance weight calculations:

$$\pi(x_k|x_{0:k-1},y_{0:k}) = p(x_k|x_{k-1}).$$

Sequential Importance Resampling (SIR) filters with transition prior as importance function are commonly known as bootstrap filter and condensation algorithm.

Resampling is used to avoid the problem of degeneracy of the algorithm, that is, avoiding the situation that all but one of the importance weights are close to zero. The performance of the algorithm can be also affected by proper choice of resampling method. The *stratified sampling* proposed by Kitagawa (1996) is optimal in terms of variance.

A single step of sequential importance resampling is as follows:

1) For $L=1,\dots,P$ draw samples from the *proposal distribution*
$x_k^{(L)} \sim \pi(x_k|x_{0:k-1}^{(L)},y_{0:k})$
2) For $L=1,\dots,P$ update the importance weights up to a normalizing constant:
$$\hat{w}_k^{(L)} = w_{k-1}^{(L)} \frac{p(y_k|x_k^{(L)})p(x_k^{(L)}|x_{k-1}^{(L)})}{\pi(x_k^{(L)}|x_{0:k-1}^{(L)},y_{0:k})}.$$
Note that this simplifies to the following :
$$\hat{w}_k^{(L)} = w_{k-1}^{(L)} p(y_k|x_k^{(L)}),$$
when we use : $\pi(x_k^{(L)}|x_{0:k-1}^{(L)},y_{0:k}) = p(x_k^{(L)}|x_{k-1}^{(L)}).$
3) For $L=1,\dots,P$ compute the normalized importance weights:
$$w_k^{(L)} = \frac{\hat{w}_k^{(L)}}{\sum_{J=1}^{P} \hat{w}_k^{(J)}}$$
4) Compute an estimate of the effective number of particles as

$$\hat{N}_{eff} = \frac{1}{\sum_{L=1}^{P}\left(w_k^{(L)}\right)^2}$$

5) If the effective number of particles is less than a given threshold $\hat{N}_{eff} < N_{thr}$, then perform resampling:
a) Draw *P* particles from the current particle set with probabilities proportional to their weights. Replace the current particle set with this new one.
b) For $L = 1, \ldots, P$ set $w_k^{(L)} = 1/P.$

The term *Sampling Importance Resampling* is also sometimes used when referring to SIR filters.

Sequential Importance Sampling (SIS)

- Is the same as Sequential Importance Resampling, but without the resampling stage.

"Direct version" algorithm

The "direct version" algorithm is rather simple (compared to other particle filtering algorithms) and it uses composition and rejection. To generate a single sample *x* at *k* from $p_{x_k|y_{1:k}}(x|y_{1:k})$:

1) Set p=1
2) Uniformly generate L from {1,...,*P*}
3) Generate a test $\hat{x}$from its distribution $p_{x_k|x_{k-1}}(x|x_{k-1|k-1}^{(L)})$
4) Generate the probability of $\hat{y}$using $\hat{x}$from $p_{y|x}(y_k|\hat{x})$where y_k is the measured value
5) Generate another uniform u from [0,m_k]
6) Compare u and $p\left(\hat{y}\right)$
6a) If u is larger then repeat from step 2
6b) If u is smaller then save $\hat{x}$as $x_{k|k}^{(p)}$and increment p
7) If p > P then quit

The goal is to generate P "particles" at *k* using only the particles from *k* − 1. This requires that a Markov equation can be written (and computed) to generate a x_k based only upon x_{k-1}. This algorithm uses composition of the P particles from *k* − 1 to generate a particle at *k* and repeats (steps 2-6) until P particles are generated at *k*.

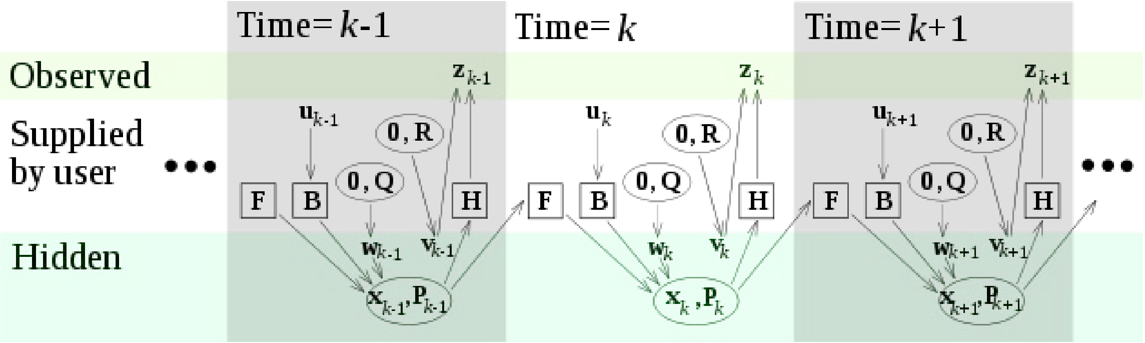
This can be more easily visualized if *x* is viewed as a two-dimensional array. One dimension is *k* and the other dimensions is the particle number. For example, *x*(*k*,*L*) would be the L^th^ particle at *k* and can also be written $x_k^{(L)}$ (as done above in the algorithm). Step 3 generates a *potential* x_k based on a randomly chosen particle ($x_{k-1}^{(L)}$) at time $k - 1$ and rejects or accepts it in step 6. In other words, the x_k values are generated using the previously generated x_{k-1}.

Other Particle Filters

- Auxiliary particle filter
- Gaussian particle filter
- Unscented particle filter
- Monte Carlo particle filter
- Gauss-Hermite particle filter
- Cost Reference particle filter
- Rao-Blackwellized particle filter

Chapter 13

Kalman Filter



Roles of the variables in the Kalman filter. (Larger image here)

In statistics, the **Kalman filter** is a mathematical method named after Rudolf E. Kalman. Its purpose is to use measurements observed over time, containing noise (random variations) and other inaccuracies, and produce values that tend to be closer to the true values of the measurements and their associated calculated values. The Kalman filter has many applications in technology, and is an essential part of space and military technology development. Perhaps the most commonly used type of very simple Kalman filter is the phase-locked loop, which is now ubiquitous in FM radios and most electronic communications equipment. Extensions and generalizations to the method have also been developed.

The Kalman filter produces estimates of the true values of measurements and their associated calculated values by predicting a value, estimating the uncertainty of the predicted value, and computing a weighted average of the predicted value and the measured value. The most weight is given to the value with the least uncertainty. The estimates produced by the method tend to be closer to the true values than the original measurements because the weighted average has a better estimated uncertainty than either of the values that went into the weighted average.

From a theoretical standpoint, the Kalman filter is an algorithm for efficiently doing exact inference in a linear dynamical system, which is a Bayesian model similar to a hidden

Markov model but where the state space of the latent variables is continuous and where all latent and observed variables have a Gaussian distribution (often a multivariate Gaussian distribution).

Naming and historical development

The filter is named after Rudolf E. Kalman, though Thorvald Nicolai Thiele and Peter Swerling developed a similar algorithm earlier. Richard S. Bucy of the University of Southern California contributed to the theory, leading to it often being called the Kalman-Bucy filter. Stanley F. Schmidt is generally credited with developing the first implementation of a Kalman filter. It was during a visit by Kalman to the NASA Ames Research Center that he saw the applicability of his ideas to the problem of trajectory estimation for the Apollo program, leading to its incorporation in the Apollo navigation computer. This Kalman filter was first described and partially developed in technical papers by Swerling (1958), Kalman (1960) and Kalman and Bucy (1961).

Kalman filters have been vital in the implementation of the navigation systems of U.S. Navy nuclear ballistic missile submarines, and in the guidance and navigation systems of cruise missiles such as the U.S. Navy's Tomahawk missile and the U.S. Air Force's Air Launched Cruise Missile. It is also used in the guidance and navigation systems of the NASA Space Shuttle and the attitude control and navigation systems of the International Space Station.

This digital filter is sometimes called the *Stratonovich–Kalman–Bucy filter* because it is a special case of a more general, non-linear filter developed somewhat earlier by the Soviet mathematician Ruslan L. Stratonovich. In fact, some of the special case linear filter's equations appeared in these papers by Stratonovich that were published before summer 1960, when Kalman met with Stratonovich during a conference in Moscow.

Overview of the calculation

The **Kalman filter** uses a system's dynamics model (i.e., physical laws of motion), known control inputs to that system, and measurements (such as from sensors) to form an estimate of the system's varying quantities (its state) that is better than the estimate obtained by using any one measurement alone. As such, it is a common sensor fusion algorithm.

All measurements and calculations based on models are estimates to some degree. Noisy sensor data, approximations in the equations that describe how a system changes, and external factors that are not accounted for introduce some uncertainty about the inferred values for a system's state. The Kalman filter averages a prediction of a system's state with a new measurement using a weighted average. The purpose of the weights is that values with better (i.e., smaller) estimated uncertainty are "trusted" more. The weights are calculated from the covariance, a measure of the estimated uncertainty of the prediction of the system's state. The result of the weighted average is a new state estimate that lies in between the predicted and measured state, and has a better estimated uncertainty than either alone. This process is repeated every time step, with the new estimate and its covariance informing the prediction used in the following iteration. This means that the Kalman filter works recursively and requires only the last "best guess" - not the entire history - of a system's state to calculate a new state.

When performing the actual calculations for the filter (as discussed below), the state estimate and covariances are coded into matrices to handle the multiple dimensions involved in a single set of calculations. This allows for representation of linear relationships between different state variables (such as position, velocity, and acceleration) in any of the transition models or covariances.

Example application

The Kalman filter is used in sensor fusion and data fusion. Typically real time systems produce multiple sequential measurements rather than making a single measurement to obtain the state of the system. These multiple measurements are then combined mathematically to generate the system's state at that time instant.

As an example application, consider the problem of determining the precise location of a truck. The truck can be equipped with a GPS unit that provides an estimate of the position within a few meters. The GPS estimate is likely to be noisy; readings 'jump around' rapidly, though always remaining within a few meters of the real position. The truck's position can also be estimated by integrating its speed and direction over time, determined by keeping track of the amount the accelerator is depressed and how much the steering wheel is turned. This is a technique known as dead reckoning. Typically, dead reckoning will provide a very smooth estimate of the truck's position, but it will drift over time as small errors accumulate. Additionally, the truck is expected to follow the laws of physics, so its position should be expected to change proportionally to its velocity.

In this example, the Kalman filter can be thought of as operating in two distinct phases: predict and update. In the prediction phase, the truck's old position will be modified according to the physical laws of motion (the dynamic or "state transition" model) plus any changes produced by the accelerator pedal and steering wheel. Not only will a new position estimate be calculated, but a new covariance will be calculated as well. Perhaps the covariance is proportional to the speed of the truck because we are more uncertain about the accuracy of the dead reckoning estimate at high speeds but very certain about the position when moving slowly. Next, in the update phase, a measurement of the truck's position is taken from the GPS unit. Along with this measurement comes some amount of uncertainty, and its covariance relative to that of the prediction from the previous phase determines how much the new measurement will affect the updated prediction. Ideally, if the dead reckoning estimates tend to drift away from the real position, the GPS measurement should pull the position estimate back towards the real position but not disturb it to the point of becoming rapidly changing and noisy.

Kalman filter in computer vision

Data fusion using a Kalman filter can assist computers to track objects in videos with low latency. The tracking of objects is a dynamic problem, using data from sensor and camera images that always suffer from noise. This can sometimes be reduced by using higher quality cameras and sensors but can never be eliminated, so it is often desirable to use a noise reduction method.

The iterative predictor-corrector nature of the Kalman filter can be helpful, because at each time instance only one constraint on the state variable need be considered. This process is repeated, considering a different constraint at every time instance. All the measured data are accumulated over time and help in predicting the state.

Video can also be pre-processed, perhaps using a segmentation technique, to reduce the computation and hence latency.

Technical description and context

The Kalman filter is an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements. It is used in a wide range of engineering and econometric applications from radar and computer vision to estimation of structural macroeconomic models, and is an important topic in control theory and control systems engineering. Together with the linear-quadratic regulator (LQR), the Kalman filter solves the linear-quadratic-Gaussian control problem (LQG). The Kalman filter, the linear-quadratic regulator and the linear-quadratic-Gaussian controller are solutions to what probably are the most fundamental problems in control theory.

In most applications, the internal state is much larger (more degrees of freedom) than the few "observable" parameters which are measured. However, by combining a series of measurements, the Kalman filter can estimate the entire internal state.

In control theory, the Kalman filter is most commonly referred to as **linear quadratic estimation** (LQE).

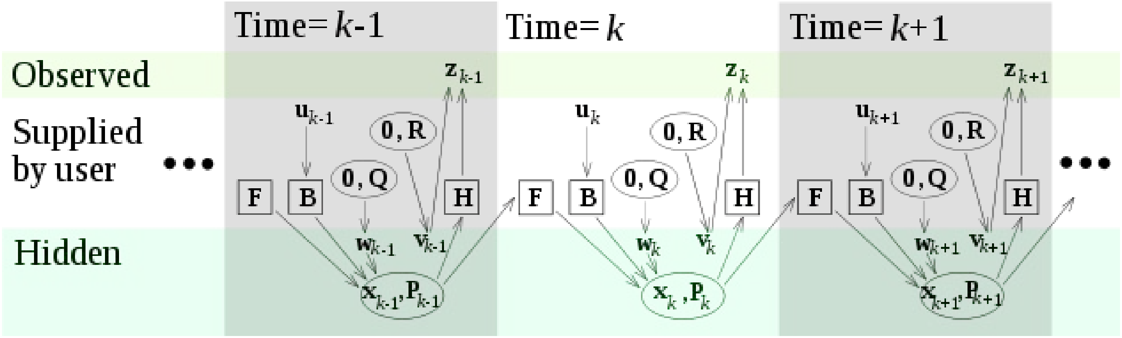
In Dempster-Shafer theory, each state equation or observation is considered a special case of a Linear belief function and the Kalman filter is a special case of combing linear belief functions on a join-tree or Markov tree.

A wide variety of Kalman filters have now been developed, from Kalman's original formulation, now called the "simple" Kalman filter, the Kalman-Bucy filter, Schmidt's "extended" filter, the "information" filter, and a variety of "square-root" filters that were developed by Bierman, Thornton and many others. Perhaps the most commonly used type of very simple Kalman filter is the phase-locked loop, which is now ubiquitous in radios, especially frequency modulation (FM) radios, television sets, satellite communications receivers, outer space communications systems, and nearly any other electronic communications equipment.

Underlying dynamic system model

Kalman filters are based on linear dynamic systems discretized in the time domain. They are modelled on a Markov chain built on linear operators perturbed by Gaussian noise. The state of the system is represented as a vector of real numbers. At each discrete time increment, a linear operator is applied to the state to generate the new state, with some noise mixed in, and optionally some information from the controls on the system if they are known. Then, another linear operator mixed with more noise generates the observed outputs from the true ("hidden") state. The Kalman filter may be regarded as analogous to the hidden Markov model, with the key difference that the hidden state variables take values in a continuous space (as opposed to a discrete state space as in the hidden Markov model). Additionally, the hidden Markov model can represent an arbitrary distribution for the next value of the state variables, in contrast to the Gaussian noise model that is used for the Kalman filter. There is a strong duality between the equations of the Kalman Filter and those of the hidden Markov model. A review of this and other models is given in Roweis and Ghahramani (1999) and Hamilton (1994), Chapter 13.

In order to use the Kalman filter to estimate the internal state of a process given only a sequence of noisy observations, one must model the process in accordance with the framework of the Kalman filter. This means specifying the following matrices: $\mathbf{F}_k$, the state-transition model; $\mathbf{H}_k$, the observation model; $\mathbf{Q}_k$, the covariance of the process noise; $\mathbf{R}_k$, the covariance of the observation noise; and sometimes $\mathbf{B}_k$, the control-input model, for each time-step, k, as described below.



Model underlying the Kalman filter. Squares represent matrices. Ellipses represent multivariate normal distributions (with the mean and covariance matrix enclosed). Unenclosed values are vectors. In the simple case, the various matrices are constant with time, and thus the subscripts are dropped, but the Kalman filter allows any of them to change each time step.

The Kalman filter model assumes the true state at time k is evolved from the state at $(k - 1)$ according to

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

where

- $\mathbf{F}_k$ is the state transition model which is applied to the previous state $\mathbf{x}_{k-1}$;
- $\mathbf{B}_k$ is the control-input model which is applied to the control vector $\mathbf{u}_k$;
- $\mathbf{w}_k$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $\mathbf{Q}_k$.

  $$\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$$

At time k an observation (or measurement) $\mathbf{z}_k$ of the true state $\mathbf{x}_k$ is made according to

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

where $\mathbf{H}_k$ is the observation model which maps the true state space into the observed space and $\mathbf{v}_k$ is the observation noise which is assumed to be zero mean Gaussian white noise with covariance $\mathbf{R}_k$.

$$\mathbf{v}_k \sim N(0, \mathbf{R}_k)$$

The initial state, and the noise vectors at each step $\{\mathbf{x}_0, \mathbf{w}_1, ..., \mathbf{w}_k, \mathbf{v}_1 ... \mathbf{v}_k\}$ are all assumed to be mutually independent.

Many real dynamical systems do not exactly fit this model. In fact, unmodelled dynamics can seriously degrade the filter performance, even when it was supposed to work with unknown stochastic signals as inputs. The reason for this is that the effect of unmodelled dynamics depends on the input, and, therefore, can bring the estimation algorithm to instability (it diverges). On the other hand, independent white noise signals will not make the algorithm diverge. The problem of separating between measurement noise and unmodelled dynamics is a difficult one and is treated in control theory under the framework of robust control.

The Kalman filter

The Kalman filter is a recursive estimator. This means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. In contrast to batch estimation techniques, no history of observations and/or estimates is required. In what follows, the notation $\hat{\mathbf{x}}_{n|m}$represents the estimate of $\mathbf{x}$at time n given observations up to, and including at time m.

The state of the filter is represented by two variables:

- $\hat{\mathbf{x}}_{k|k}$, the *a posteriori* state estimate at time k given observations up to and including at time k;
- $\mathbf{P}_{k|k}$, the *a posteriori* error covariance matrix (a measure of the estimated accuracy of the state estimate).

The Kalman filter can be written as a single equation, however it is most often conceptualized as two distinct phases: "Predict" and "Update". The predict phase uses the state estimate from the previous timestep to produce an estimate of the state at the current timestep. This predicted state estimate is also known as the *a priori* state estimate because, although it is an estimate of the state at the current timestep, it does not include observation information from the current timestep. In the update phase, the current *a priori* prediction is combined with current observation information to refine the state estimate. This improved estimate is termed the *a posteriori* state estimate.

Typically, the two phases alternate, with the prediction advancing the state until the next scheduled observation, and the update incorporating the observation. However, this is not necessary; if an observation is unavailable for some reason, the update may be skipped and multiple prediction steps performed. Likewise, if multiple independent observations are available at the same time, multiple update steps may be performed (typically with different observation matrices $\mathbf{H}_k$).

Predict

Predicted (*a priori*) state estimate	$\hat{\mathbf{x}}_{k\|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1\|k-1} + \mathbf{B}_k\mathbf{u}_k$
Predicted (*a priori*) estimate covariance	$\mathbf{P}_{k\|k-1} = \mathbf{F}_k\mathbf{P}_{k-1\|k-1}\mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$

Update

Innovation or measurement residual	$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k\|k-1}$
Innovation (or residual) covariance	$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k\|k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$
Optimal Kalman gain	$\mathbf{K}_k = \mathbf{P}_{k\|k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{S}_k^{-1}$
Updated (*a posteriori*) state estimate	$\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k$
Updated (*a posteriori*) estimate covariance	$\mathbf{P}_{k\|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k\|k-1}$

The formula for the updated estimate and covariance above is only valid for the optimal Kalman gain. Usage of other gain values require a more complex formula found in the *derivations* section.

Invariants

If the model is accurate, and the values for $\hat{\mathbf{x}}_{0|0}$and $\mathbf{P}_{0|0}$accurately reflect the distribution of the initial state values, then the following invariants are preserved: (all estimates have a mean error of zero)

- $\mathrm{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] = \mathrm{E}[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}] = 0$
- $\mathrm{E}[\tilde{\mathbf{y}}_k] = 0$

where E[ξ] is the expected value of ξ, and covariance matrices accurately reflect the covariance of estimates

- $\mathbf{P}_{k|k} = \textrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})$
- $\mathbf{P}_{k|k-1} = \textrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$
- $\mathbf{S}_{k} = \textrm{cov}(\tilde{\mathbf{y}}_k)$

Estimation of the noise covariances Q*k* and R*k*

Practical implementation of the Kalman Filter is often difficult due to the inability in getting a good estimate of the noise covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$. Extensive research has been done in this field to estimate these covariances from data. One of the more promising approaches to doing this is called the **Autocovariance Least-Squares (ALS)** technique that uses autocovariances of routine operating data to estimate the covariances. The Matlab or GNU Octave code used to calculate the noise covariance matrices using the **ALS** technique is available online under the GNU General Public License license.

Example application, technical

Consider a truck on perfectly frictionless, infinitely long straight rails. Initially the truck is stationary at position 0, but it is buffeted this way and that by random acceleration. We measure the position of the truck every *Δt* seconds, but these measurements are imprecise; we want to maintain a model of where the truck is and what its velocity is. We show here how we derive the model from which we create our Kalman filter.

Since **F**, **H**, **R** and **Q** are constant, their time indices are dropped.

The position and velocity of the truck are described by the linear state space

$$\mathbf{x}_k = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

where $\dot{x}$is the velocity, that is, the derivative of position with respect to time.

We assume that between the (*k* − 1) and *k* timestep the truck undergoes a constant acceleration of a_k that is normally distributed, with mean 0 and standard deviation σ_a. From Newton's laws of motion we conclude that

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k$$

(note that there is no $\mathbf{B}u$term since we have no known control inputs) where

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

and

$$\mathbf{G} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$$

so that

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k$$

where $\mathbf{w}_k \sim N(0, \mathbf{Q})$and

$$\mathbf{Q} = \mathbf{G}\mathbf{G}^{\mathrm{T}}\sigma_a^2 = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}\sigma_a^2.$$

At each time step, a noisy measurement of the true position of the truck is made. Let us suppose the measurement noise v_k is also normally distributed, with mean 0 and standard deviation σ_z.

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$$

where

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

and

$$\mathbf{R} = \mathrm{E}[\mathbf{v}_k \mathbf{v}_k^{\mathrm{T}}] = \begin{bmatrix} \sigma_z^2 \end{bmatrix}$$

We know the initial starting state of the truck with perfect precision, so we initialize

$$\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and to tell the filter that we know the exact position, we give it a zero covariance matrix:

$$\mathbf{P}_{0|0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

If the initial position and velocity are not known perfectly the covariance matrix should be initialized with a suitably large number, say L, on its diagonal.

$$\mathbf{P}_{0|0} = \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix}$$

The filter will then prefer the information from the first measurements over the information already in the model.

Derivations

Deriving the *a posteriori* estimate covariance matrix

Starting with our invariant on the error covariance $\mathbf{P}_{k|k}$ as above

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})$$

substitute in the definition of $\hat{\mathbf{x}}_{k|k}$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k))$$

and substitute $\tilde{\mathbf{y}}_k$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})))$$

and $\mathbf{z}_k$

$$\mathbf{P}_{k|k} = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})))$$

and by collecting the error vectors we get

$$\mathbf{P}_{k|k} = \mathrm{cov}((I - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k\mathbf{v}_k)$$

Since the measurement error $\mathbf{v}_k$ is uncorrelated with the other terms, this becomes

$$\mathbf{P}_{k|k} = \mathrm{cov}((I - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})) + \mathrm{cov}(\mathbf{K}_k\mathbf{v}_k)$$

by the properties of vector covariance this becomes

$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(I - \mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k\mathrm{cov}(\mathbf{v}_k)\mathbf{K}_k^{\mathrm{T}}$$

which, using our invariant on $\mathbf{P}_{k|k\text{-}1}$ and the definition of $\mathbf{R}_k$ becomes

$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(I - \mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}}$$

This formula (sometimes known as the "**Joseph form**" of the covariance update equation) is valid for any value of $\mathbf{K}_k$. It turns out that if $\mathbf{K}_k$ is the optimal Kalman gain, this can be simplified further as shown below.

Kalman gain derivation

The Kalman filter is a minimum mean-square error estimator. The error in the *a posteriori* state estimation is

$$\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}$$

We seek to minimize the expected value of the square of the magnitude of this vector, $\mathrm{E}[|\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}|^2]$. This is equivalent to minimizing the trace of the *a posteriori* estimate covariance matrix $\mathbf{P}_{k|k}$. By expanding out the terms in the equation above and collecting, we get:

$$\begin{aligned}\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}} + \mathbf{K}_k(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k)\mathbf{K}_k^{\mathrm{T}} \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}} + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\mathrm{T}}\end{aligned}$$

The trace is minimized when the matrix derivative is zero:

$$\frac{\partial\; \mathrm{tr}(\mathbf{P}_{k|k})}{\partial\; \mathbf{K}_k} = -2(\mathbf{H}_k\mathbf{P}_{k|k-1})^{\mathrm{T}} + 2\mathbf{K}_k\mathbf{S}_k = 0$$

Solving this for $\mathbf{K}_k$ yields the Kalman gain:

$$\begin{aligned}\mathbf{K}_k\mathbf{S}_k &= (\mathbf{H}_k\mathbf{P}_{k|k-1})^{\mathrm{T}} = \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}} \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{S}_k^{-1}\end{aligned}$$

This gain, which is known as the *optimal Kalman gain*, is the one that yields MMSE estimates when used.

Simplification of the *a posteriori* error covariance formula

The formula used to calculate the *a posteriori* error covariance can be simplified when the Kalman gain equals the optimal value derived above. Multiplying both sides of our Kalman gain formula on the right by $\mathbf{S}_k\mathbf{K}_k^T$, it follows that

$$\mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T$$

Referring back to our expanded formula for the *a posteriori* error covariance,

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T$$

we find the last two terms cancel out, giving

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}.$$

This formula is computationally cheaper and thus nearly always used in practice, but is only correct for the optimal gain. If arithmetic precision is unusually low causing problems with numerical stability, or if a non-optimal Kalman gain is deliberately used, this simplification cannot be applied; the *a posteriori* error covariance formula as derived above must be used.

Sensitivity analysis

The Kalman filtering equations provide an estimate of the state $\hat{\mathbf{x}}_{k|k}$and its error covariance $\mathbf{P}_{k|k}$recursively. The estimate and its quality depend on the system parameters and the noise statistics fed as inputs to the estimator. This section analyzes the effect of uncertainties in the statistical inputs to the filter. In the absence of reliable statistics or the true values of noise covariance matrices $\mathbf{Q}_k$and $\mathbf{R}_k$, the expression

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k$$

no longer provides the actual error covariance. In other words, $\mathbf{P}_{k|k} \neq E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T]$. In most real time applications the covariance matrices that are used in designing the Kalman filter are different from the actual noise covariances matrices. This sensitivity analysis describes the behavior of the estimation error covariance when the noise covariances as well as the system matrices $\mathbf{F}_k$and $\mathbf{H}_k$that are fed as inputs to the filter are incorrect. Thus, the sensitivity analysis describes the robustness (or sensitivity) of the estimator to misspecified statistical and parametric inputs to the estimator.

This discussion is limited to the error sensitivity analysis for the case of statistical uncertainties. Here the actual noise covariances are denoted by $\mathbf{Q}^a_k$and $\mathbf{R}^a_k$respectively, whereas the design values used in the estimator are $\mathbf{Q}_k$and $\mathbf{R}_k$respectively. The actual error covariance is denoted by $\mathbf{P}^a_{k|k}$and $\mathbf{P}_{k|k}$as computed by the Kalman filter is referred to as the Riccati variable. When $\mathbf{Q}_k \equiv \mathbf{Q}^a_k$and $\mathbf{R}_k \equiv \mathbf{Q}^a_k$, this means that $\mathbf{P}_{k|k} = \mathbf{P}^a_{k|k}$. While computing the actual error covariance using $\mathbf{P}^a_{k|k} = E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T]$, substituting for $\hat{\mathbf{x}}_{k|k}$and using the fact that $E[\mathbf{w}_k\mathbf{w}_k^T] = \mathbf{Q}^a_k$and $E[\mathbf{v}_k\mathbf{v}_k^T] = \mathbf{R}^a_k$, results in the following recursive equations for $\mathbf{P}^a_{k|k}$:

$$\mathbf{P}^a_{k|k-1} = \mathbf{F}_k\mathbf{P}^a_{k-1|k-1}\mathbf{F}_k^T + \mathbf{R}^a_k$$
$$\mathbf{P}^a_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^a_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}^a_k\mathbf{K}_k$$

- While computing $\mathbf{P}_{k|k}$, by design the filter implicitly assumes that $E[\mathbf{w}_k\mathbf{w}_k^T] = \mathbf{Q}_k$and $E[\mathbf{v}_k\mathbf{v}_k^T] = \mathbf{R}_k$.
- The recursive expressions for $\mathbf{P}^a_{k|k}$and $\mathbf{P}_{k|k}$are identical except for the presence of $\mathbf{Q}^a_k$and $\mathbf{R}^a_k$in place of the design values $\mathbf{Q}_k$and $\mathbf{R}_k$respectively.

Square root form

One problem with the Kalman filter is its numerical stability; when the process is well known (the process noise covariance $\mathbf{Q}_k$ is small), it is easy for rounding error to render the state covariance matrix **P** invalid: negative diagonal entries or otherwise not positive semi-definite.

Positive definite matrices have the property that they have a triangular matrix square root $\mathbf{P} = \mathbf{S}\cdot\mathbf{S}^T$. This can be computed efficiently using the Cholesky factorization algorithm, but more importantly if the covariance is kept in this form, it can never have a negative diagonal or become asymmetric. An equivalent form, which avoids many of the square root operations required by the matrix square root yet preserves the desirable numerical properties, is the U-D decomposition form, $\mathbf{P} = \mathbf{U}\cdot\mathbf{D}\cdot\mathbf{U}^T$, where **U** is a unit triangular matrix (with unit diagonal), and D is a diagonal matrix.

Between the two, the U-D factorization uses the same amount of storage, and somewhat less computation, and is the most commonly used square root form. (Early literature on the relative efficiency is somewhat misleading, as it assumed that square roots were much more time-consuming than divisions,[:69] while on 21-st century computers they are only slightly more expensive.)

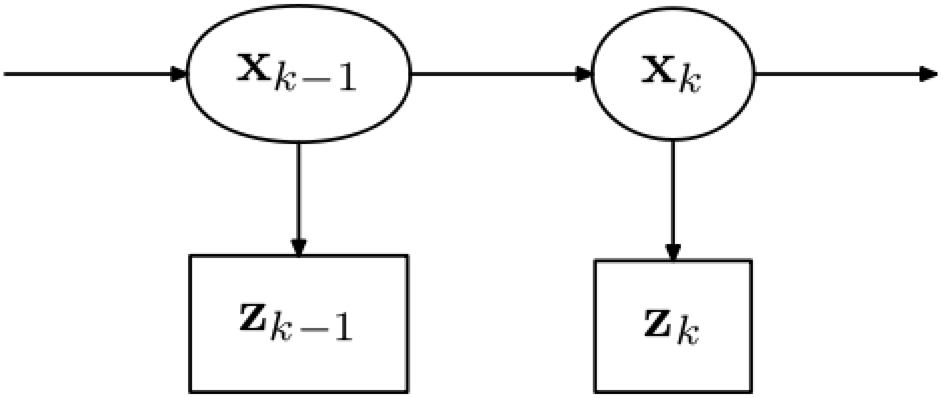
Efficient algorithms for the Kalman prediction and update steps in the square root form were developed by G. J. Bierman and C. L. Thornton.

The $\mathbf{L}\cdot\mathbf{D}\cdot\mathbf{L}^T$ decomposition of the innovation covariance matrix $\mathbf{S}_k$ is the basis for another type of numerically efficient and robust square root filter. The algorithm starts with the LU decomposition as implemented in the Linear Algebra PACKage (LAPACK). These results are further factored into the $\mathbf{L}\cdot\mathbf{D}\cdot\mathbf{L}^T$ structure with methods given by Golub and Van Loan (algorithm 4.1.2) for a symmetric nonsingular matrix. Any singular covariance matrix is pivoted so that the first diagonal partition is nonsingular and well-conditioned. The pivoting algorithm must retain any portion of the innovation covariance matrix directly corresponding to observed state-variables $\mathbf{H}_k\cdot\mathbf{x}_{k|k-1}$ that are associated with auxiliary observations in $\mathbf{y}_k$. The $\mathbf{L}\cdot\mathbf{D}\cdot\mathbf{L}^T$ square-root filter requires orthogonalization of the observation vector. This may be done with the inverse square-root of the covariance matrix for the auxiliary variables using Method 2 in Higham (2002, p. 263).

Relationship to recursive Bayesian estimation

The Kalman filter can be considered to be one of the most simple dynamic Bayesian networks. The Kalman filter calculates estimates of the true values of measurements recursively over time using incoming measurements and a mathematical process model. Similarly, recursive Bayesian estimation calculates estimates of an unknown probability density function (PDF) recursively over time using incoming measurements and a mathematical process model.

In recursive Bayesian estimation, the true state is assumed to be an unobserved Markov process, and the measurements are the observed states of a hidden Markov model (HMM).



Because of the Markov assumption, the true state is conditionally independent of all earlier states given the immediately previous state.

$$p(\mathbf{x}_k|\mathbf{x}_0,\dots,\mathbf{x}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$$

Similarly the measurement at the *k*-th timestep is dependent only upon the current state and is conditionally independent of all other states given the current state.

$$p(\mathbf{z}_k|\mathbf{x}_0,\dots,\mathbf{x}_k) = p(\mathbf{z}_k|\mathbf{x}_k)$$

Using these assumptions the probability distribution over all states of the hidden Markov model can be written simply as:

$$p(\mathbf{x}_0,\dots,\mathbf{x}_k,\mathbf{z}_1,\dots,\mathbf{z}_k) = p(\mathbf{x}_0)\prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}_i)p(\mathbf{x}_i|\mathbf{x}_{i-1})$$

However, when the Kalman filter is used to estimate the state **x**, the probability distribution of interest is that associated with the current states conditioned on the measurements up to the current timestep. This is achieved by marginalizing out the previous states and dividing by the probability of the measurement set.

This leads to the *predict* and *update* steps of the Kalman filter written probabilistically. The probability distribution associated with the predicted state is the sum (integral) of the products of the probability distribution associated with the transition from the (*k* - 1)-th timestep to the *k*-th and the probability distribution associated with the previous state, over all possible x_{k-1}.

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\,d\mathbf{x}_{k-1}$$

The measurement set up to time *t* is

$$\mathbf{Z}_t = \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$$

The probability distribution of the update is proportional to the product of the measurement likelihood and the predicted state.

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})}$$

The denominator

$$p(\mathbf{z}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})d\mathbf{x}_k$$

is a normalization term.

The remaining probability density functions are

$$\begin{aligned}
p(\mathbf{x}_k|\mathbf{x}_{k-1}) &= \mathcal{N}(\mathbf{F}_k\mathbf{x}_{k-1}, \mathbf{Q}_k)\\
p(\mathbf{z}_k|\mathbf{x}_k) &= \mathcal{N}(\mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k)\\
p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}) &= \mathcal{N}(\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1})
\end{aligned}$$

Note that the PDF at the previous timestep is inductively assumed to be the estimated state and covariance. This is justified because, as an optimal estimator, the Kalman filter makes best use of the measurements, therefore the PDF for $\mathbf{x}_k$given the measurements $\mathbf{Z}_k$ is the Kalman filter estimate.

Information filter

In the information filter, or inverse covariance filter, the estimated covariance and estimated state are replaced by the information matrix and information vector respectively. These are defined as:

$$\begin{aligned}
\mathbf{Y}_{k|k} &= \mathbf{P}_{k|k}^{-1}\\
\hat{\mathbf{y}}_{k|k} &= \mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k}
\end{aligned}$$

Similarly the predicted covariance and state have equivalent information forms, defined as:

$$\mathbf{Y}_{k|k-1} = \mathbf{P}_{k|k-1}^{-1}$$
$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{P}_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1}$$

as have the measurement covariance and measurement vector, which are defined as:

$$\mathbf{I}_k = \mathbf{H}_k^{\mathrm{T}}\mathbf{R}_k^{-1}\mathbf{H}_k$$
$$\mathbf{i}_k = \mathbf{H}_k^{\mathrm{T}}\mathbf{R}_k^{-1}\mathbf{z}_k$$

The information update now becomes a trivial sum.

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \mathbf{I}_k$$
$$\hat{\mathbf{y}}_{k|k} = \hat{\mathbf{y}}_{k|k-1} + \mathbf{i}_k$$

The main advantage of the information filter is that *N* measurements can be filtered at each timestep simply by summing their information matrices and vectors.

$$\mathbf{Y}_{k|k} = \mathbf{Y}_{k|k-1} + \sum_{j=1}^{N}\mathbf{I}_{k,j}$$
$$\hat{\mathbf{y}}_{k|k} = \hat{\mathbf{y}}_{k|k-1} + \sum_{j=1}^{N}\mathbf{i}_{k,j}$$

To predict the information filter the information matrix and vector can be converted back to their state space equivalents, or alternatively the information space prediction can be used.

$$\mathbf{M}_k = [\mathbf{F}_k^{-1}]^{\mathrm{T}}\mathbf{Y}_{k-1|k-1}\mathbf{F}_k^{-1}$$
$$\mathbf{C}_k = \mathbf{M}_k[\mathbf{M}_k + \mathbf{Q}_k^{-1}]^{-1}$$
$$\mathbf{L}_k = I - \mathbf{C}_k$$
$$\mathbf{Y}_{k|k-1} = \mathbf{L}_k\mathbf{M}_k\mathbf{L}_k^{\mathrm{T}} + \mathbf{C}_k\mathbf{Q}_k^{-1}\mathbf{C}_k^{\mathrm{T}}$$
$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{L}_k[\mathbf{F}_k^{-1}]^{\mathrm{T}}\hat{\mathbf{y}}_{k-1|k-1}$$

Note that if *F* and *Q* are time invariant these values can be cached. Note also that *F* and *Q* need to be invertible.

Fixed-lag smoother

The optimal fixed-lag smoother provides the optimal estimate of $\hat{\mathbf{x}}_{k-N|k}$for a given fixed-lag *N* using the measurements from $\mathbf{z}_1$to $\mathbf{z}_k$. It can be derived using the previous theory via an augmented state, and the main equation of the filter is the following:

$$\begin{bmatrix} \hat{\mathbf{x}}_{t|t} \\ \hat{\mathbf{x}}_{t-1|t} \\ \vdots \\ \hat{\mathbf{x}}_{t-N+1|t} \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \hat{\mathbf{x}}_{t|t-1} + \begin{bmatrix} 0 & \ldots & 0 \\ I & 0 & \vdots \\ \vdots & \ddots & \vdots \\ 0 & \ldots & I \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{t-1|t-1} \\ \hat{\mathbf{x}}_{t-2|t-1} \\ \vdots \\ \hat{\mathbf{x}}_{t-N|t-1} \end{bmatrix} + \begin{bmatrix} K^{(1)} \\ K^{(2)} \\ \vdots \\ K^{(N)} \end{bmatrix} y_{t|t-1}$$

where:

- $\hat{\mathbf{x}}_{t|t-1}$is estimated via a standard Kalman filter;
- $y_{t|t-1} = z(t) - \hat{\mathbf{x}}_{t|t-1}$is the innovation produced considering the estimate of the standard Kalman filter;
- the various $\hat{\mathbf{x}}_{t-i|t}$with $i = 0, \ldots, N$are new variables, i.e. they do not appear in the standard Kalman filter;
- the gains are computed via the following scheme:

  $$K^{(i)} = P^{(i)} H^T \left[HPH^T + R \right]^{-1}$$

  and

  $$P^{(i)} = P \left[\left[F - KH \right]^T \right]^i$$

  where *P* and *K* are the prediction error covariance and the gains of the standard Kalman filter.

If the estimation error covariance is defined so that

$$P_i := E \left[\left(\mathbf{x}_{t-i} - \hat{\mathbf{x}}_{t-i|t} \right)^* \left(\mathbf{x}_{t-i} - \hat{\mathbf{x}}_{t-i|t} \right) | z_1 \ldots z_t \right],$$

then we have that the improvement on the estimation of $\mathbf{x}_{t-i}$is given by:

$$P - P_i = \sum_{j=0}^{i} \left[P^{(j)} H^T \left[HPH^T + R \right]^{-1} H \left(P^{(i)} \right)^T \right]$$

Fixed-interval smoothers

The optimal fixed-interval smoother provides the optimal estimate of $\hat{\mathbf{x}}_{k|n}$ ($k < n$) using the measurements from a fixed interval $\mathbf{z}_1$ to $\mathbf{z}_n$. This is also called "Kalman Smoothing". There are several smoothing algorithms in common use.

Rauch–Tung–Striebel

The Rauch–Tung–Striebel (RTS) Smoother is an efficient two-pass algorithm for fixed interval smoothing. The main equations of the smoother are the following (assuming $\mathbf{B}_k = \mathbf{0}$):

- forward pass: regular Kalman filter algorithm
- backward pass:

$\hat{\mathbf{x}}_{k|n} = \tilde{F}_k \hat{\mathbf{x}}_{k+1|n} + \tilde{K}_k \hat{\mathbf{x}}_{k+1|k}$, where

-
    - $\tilde{\mathbf{F}}_k = \mathbf{F}_k^{-1}(\mathbf{I} - \mathbf{Q}_k \mathbf{P}_{k+1|k}^{-1})$
    - $\tilde{\mathbf{K}}_k = \mathbf{F}_k^{-1} \mathbf{Q}_k \mathbf{P}_{k+1|k}^{-1}$

Modified Bryson-Frazier Smoother

An alternative to the RTS algorithm is the Modified Bryson-Frazier (MBF) fixed interval smoother developed by Bierman. This also uses a backward pass that processes data saved from the Kalman filter forward pass. The equations for the backward pass involve the recursive computation of data which are used at each observation time to compute the smoothed state and covariance.

The recursive equations are

$$\begin{aligned}
&\tilde{\Lambda}_k = \mathbf{H}_k^T \mathbf{S}_k^{-1} \mathbf{H}_k + \mathbf{C}_k^T \hat{\Lambda}_k \mathbf{C}_k \\
&\hat{\Lambda}_{k-1} = \mathbf{F}_k^T \tilde{\Lambda}_k \mathbf{F}_k \\
&\hat{\Lambda}_n = 0 \\
&\tilde{\lambda}_k = -\mathbf{H}_k^T \mathbf{S}_k^{-1} \mathbf{z}_k + \mathbf{C}_k^T \hat{\lambda}_k \\
&\hat{\lambda}_{k-1} = \mathbf{F}_k^T \tilde{\lambda}_k \\
&\hat{\lambda}_n = 0
\end{aligned}$$

where $\mathbf{S}_k$ is the residual covariance and $\mathbf{C}_k = \mathbf{I} - \mathbf{K}_k \mathbf{H}_k$. The smoothed state and covariance can then be found by substitution in the equations

$$\mathbf{P}_{k|n} = \mathbf{P}_{k|k} - \mathbf{P}_{k|k} \hat{\Lambda}_k \mathbf{P}_{k|k}$$

$$\mathbf{x}_{k|n} = \mathbf{x}_{k|k} - \mathbf{P}_{k|k}\hat{\lambda}_k$$

or

$$\begin{aligned}\mathbf{P}_{k|n} &= \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\tilde{\Lambda}_k\mathbf{P}_{k|k-1}\\ \mathbf{x}_{k|n} &= \mathbf{x}_{k|k-1} - \mathbf{P}_{k|k-1}\tilde{\lambda}_k.\end{aligned}$$

An important advantage of the MBF is that it does not require finding the inverse of the covariance matrix.

Non-linear filters

The basic Kalman filter is limited to a linear assumption. More complex systems, however, can be nonlinear. The non-linearity can be associated either with the process model or with the observation model or with both.

Extended Kalman filter

In the extended Kalman filter (EKF), the state transition and observation models need not be linear functions of the state but may instead be non-linear functions. These functions are of differentiable type.

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k\\ \mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k\end{aligned}$$

The function *f* can be used to compute the predicted state from the previous estimate and similarly the function *h* can be used to compute the predicted measurement from the predicted state. However, *f* and *h* cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed.

At each timestep the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate.

Unscented Kalman filter

When the state transition and observation models – that is, the predict and update functions *f* and *h* (see above) – are highly non-linear, the extended Kalman filter can give particularly poor performance. This is because the covariance is propagated through linearization of the underlying non-linear model. The unscented Kalman filter (UKF) uses a deterministic sampling technique known as the unscented transform to pick a minimal set of sample points (called sigma points) around the mean. These sigma points are then propagated through the non-linear functions, from which the mean and covariance of the estimate are then recovered. The result is a filter which more accurately captures the true mean and covariance. (This can be verified using Monte Carlo sampling or through a Taylor series expansion of the posterior statistics.) In addition, this technique removes the requirement to explicitly calculate Jacobians, which for complex functions can be a difficult task in itself (i.e., requiring complicated derivatives if done analytically or being computationally costly if done numerically).

Predict

As with the EKF, the UKF prediction can be used independently from the UKF update, in combination with a linear (or indeed EKF) update, or vice versa.

The estimated state and covariance are augmented with the mean and covariance of the process noise.

$$\mathbf{x}^a_{k-1|k-1} = [\hat{\mathbf{x}}^T_{k-1|k-1} \quad E[\mathbf{w}^T_k]\ ]^T$$
$$\mathbf{P}^a_{k-1|k-1} = \begin{bmatrix} \mathbf{P}_{k-1|k-1} & 0 \\ 0 & \mathbf{Q}_k \end{bmatrix}$$

A set of 2*L*+1 sigma points is derived from the augmented state and covariance where *L* is the dimension of the augmented state.

$$\chi^0_{k-1|k-1} = \mathbf{x}^a_{k-1|k-1}$$
$$\chi^i_{k-1|k-1} = \mathbf{x}^a_{k-1|k-1} + \left(\sqrt{(L+\lambda)\mathbf{P}^a_{k-1|k-1}}\right)_i \ i = 1..L$$
$$\chi^i_{k-1|k-1} = \mathbf{x}^a_{k-1|k-1} - \left(\sqrt{(L+\lambda)\mathbf{P}^a_{k-1|k-1}}\right)_{i-L} \ i = L+1, \dots 2L$$

where

$$\left(\sqrt{(L+\lambda)\mathbf{P}^a_{k-1|k-1}}\right)_i$$

is the *i*th column of the matrix square root of

$$(L+\lambda)\mathbf{P}^a_{k-1|k-1}$$

using the definition: square root *A* of matrix *B* satisfies

$$B \equiv AA^T.$$

The matrix square root should be calculated using numerically efficient and stable methods such as the Cholesky decomposition.

The sigma points are propagated through the transition function *f*.

$$\chi^i_{k|k-1} = f(\chi^i_{k-1|k-1}) \quad i = 0..2L$$

where $f : R^L \rightarrow R^{|\mathbf{x}|}$. The weighted sigma points are recombined to produce the predicted state and covariance.

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \sum_{i=0}^{2L} W_s^i \chi^i_{k|k-1} \\ \mathbf{P}_{k|k-1} &= \sum_{i=0}^{2L} W_c^i\ [\chi^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}][\chi^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}]^T \end{aligned}$$

where the weights for the state and covariance are given by:

$$\begin{aligned} W_s^0 &= \frac{\lambda}{L+\lambda} \\ W_c^0 &= \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \\ W_s^i = W_c^i &= \frac{1}{2(L+\lambda)} \\ \lambda &= \alpha^2(L+\kappa) - L \end{aligned}$$

α and κ control the spread of the sigma points. β is related to the distribution of *x*. Normal values are α = 10^{-3}, κ = 1 and β = 2. If the true distribution of *x* is Gaussian, β = 2 is optimal.

Update

The predicted state and covariance are augmented as before, except now with the mean and covariance of the measurement noise.

$$\begin{aligned} \mathbf{x}^a_{k|k-1} &= [\hat{\mathbf{x}}^T_{k|k-1} \quad E[\mathbf{v}^T_k]\ ]^T \\ \mathbf{P}^a_{k|k-1} &= \begin{bmatrix} \mathbf{P}_{k|k-1} & 0 \\ 0 & \mathbf{R}_k \end{bmatrix} \end{aligned}$$

As before, a set of 2*L* + 1 sigma points is derived from the augmented state and covariance where *L* is the dimension of the augmented state.

$$\begin{aligned} \chi^0_{k|k-1} &= \mathbf{x}^a_{k|k-1} \\ \chi^i_{k|k-1} &= \mathbf{x}^a_{k|k-1} + \left(\sqrt{(L+\lambda)\mathbf{P}^a_{k|k-1}}\right)_i \quad i = 1..L \\ \chi^i_{k|k-1} &= \mathbf{x}^a_{k|k-1} - \left(\sqrt{(L+\lambda)\mathbf{P}^a_{k|k-1}}\right)_{i-L} \quad i = L+1, \dots 2L \end{aligned}$$

Alternatively if the UKF prediction has been used the sigma points themselves can be augmented along the following lines

$$\chi_{k|k-1} := [\chi_{k|k-1}^T \quad E[\mathbf{v}_k^T]\ ]^T \pm \sqrt{(L+\lambda)\mathbf{R}_k^a}$$

where

$$\mathbf{R}_k^a = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{R}_k \end{bmatrix}$$

The sigma points are projected through the observation function *h*.

$$\gamma_k^i = h(\chi_{k|k-1}^i) \quad i = 0..2L$$

The weighted sigma points are recombined to produce the predicted measurement and predicted measurement covariance.

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{2L} W_s^i \gamma_k^i$$
$$\mathbf{P}_{z_k z_k} = \sum_{i=0}^{2L} W_c^i\ [\gamma_k^i - \hat{\mathbf{z}}_k][\gamma_k^i - \hat{\mathbf{z}}_k]^T$$

The state-measurement cross-covariance matrix,

$$\mathbf{P}_{x_k z_k} = \sum_{i=0}^{2L} W_c^i\ [\chi_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1}][\gamma_k^i - \hat{\mathbf{z}}_k]^T$$

is used to compute the UKF Kalman gain.

$$K_k = \mathbf{P}_{x_k z_k} \mathbf{P}_{z_k z_k}^{-1}$$

As with the Kalman filter, the updated state is the predicted state plus the innovation weighted by the Kalman gain,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

And the updated covariance is the predicted covariance, minus the predicted measurement covariance, weighted by the Kalman gain.

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - K_k \mathbf{P}_{z_k z_k} K_k^T$$

Kalman–Bucy filter

The Kalman–Bucy filter is a continuous time version of the Kalman filter.

It is based on the state space model

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{w}(t)$$
$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t)$$

where the covariances of the noise terms $\mathbf{w}(t)$and $\mathbf{v}(t)$are given by $\mathbf{Q}(t)$and $\mathbf{R}(t)$, respectively.

The filter consists of two differential equations, one for the state estimate and one for the covariance:

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = \mathbf{F}(t)\hat{\mathbf{x}}(t) + \mathbf{K}(t)(\mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t))$$
$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{Q}(t) - \mathbf{K}(t)\mathbf{R}(t)\mathbf{K}^T(t)$$

where the Kalman gain is given by

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^T(t)\mathbf{R}^{-1}(t)$$

Note that in this expression for $\mathbf{K}(t)$the covariance of the observation noise $\mathbf{R}(t)$represents at the same time the covariance of the prediction error (or *innovation*) $\tilde{\mathbf{y}}(t) = \mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t)$; these covariances are equal only in the case of continuous time.

The distinction between the prediction and update steps of discrete-time Kalman filtering does not exist in continuous time.

The second differential equation, for the covariance, is an example of a Riccati equation.

Hybrid Kalman filter

Most physical systems are represented as continuous-time models while discrete-time measurements are frequently taken for state estimation via a digital processor. Therefore, the system model and measurement model are given by

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{w}(t), &\mathbf{w}(t) &\sim N\bigl(\mathbf{0}, \mathbf{Q}(t)\bigr)\\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, &\mathbf{v}_k &\sim N(\mathbf{0}, \mathbf{R}_k)\end{aligned}$$

where $\mathbf{x}_k = \mathbf{x}(t_k)$.

Initialize

$$\hat{\mathbf{x}}_{0|0} = E\big[\mathbf{x}(t_0)\big], \mathbf{P}_{0|0} = Var\big[\mathbf{x}(t_0)\big]$$

Predict

$$\begin{aligned}
&\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}(t)\hat{\mathbf{x}}(t) + \mathbf{B}(t)\mathbf{u}(t), \text{ with } \hat{\mathbf{x}}(t_{k-1}) = \hat{\mathbf{x}}_{k-1|k-1} \\
\Rightarrow &\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}(t_k) \\
&\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t)^T + \mathbf{Q}(t), \text{ with } \mathbf{P}(t_{k-1}) = \mathbf{P}_{k-1|k-1} \\
\Rightarrow &\mathbf{P}_{k|k-1} = \mathbf{P}(t_k)
\end{aligned}$$

The prediction equations are derived from those of continuous-time Kalman filter without update from measurements, i.e., $\mathbf{K}(t) = 0$. The predicted state and covariance are calculated respectively by solving a set of differential equations with the initial value equal to the estimate at the previous step.

Update

$$\begin{aligned}
&\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\big(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k\big)^{-1} \\
&\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \\
&\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}
\end{aligned}$$

The update equations are identical to those of discrete-time Kalman filter.

Applications

- Attitude and Heading Reference Systems
- Autopilot
- Battery state of charge (SoC) estimation
- Brain-computer interface
- Chaotic signals
- Dynamic positioning
- Economics, in particular macroeconomics, time series, and econometrics
- Inertial guidance system
- Radar tracker
- Satellite navigation systems
- Simultaneous localization and mapping
- Speech enhancement
- Weather forecasting
- Navigation Systems
- 3D-Modelling