'Let me put it this way, Mr. Amer. The 9000 series is the most reliable computer ever made. No 9000 computer has ever made a mistake or distorted information. We are all, by any practical definition of the words, foolproof and incapable of error.'

- HAL, from "2001, A Space Odyssey"
Chapter 2

Modeling & Estimation

Designing and operating machinery such as industrial plants requires understanding of the relevant physical principles. Automatic control is an essential ingredient that depends on understanding to translate desired result to necessary actions. This essential understanding is reflected in a model, also called a blue-print. Formal methods to model complex systems, starting bottom-up from the principles at the finest level of physical detail, have difficulty to provide coherent models of global behavior, especially when non-linear processes are involved. Consequently, for adequate operation, it is often necessary to update a model using measurements, resorting either to on-line feedback control and/or online re-estimation of the control model. We introduce the models for control and fitting of data, and consider the fundamental limitations of modeling, control and estimation. Both formal hypothesis-driven methods as well as data-driven methods yield imperfect models. Time-series analysis and dynamic modeling can reveal structure in time, as required to detect disturbances resulting from imperfections. This chapter also introduces principles of time-series analysis and dynamical modeling.

This chapter introduces elementary ways to construct a model, and to fit or identify a model from data, either based from a blueprint or from behavior only. A discussion on modeling ought to start with a perspective on systems and processes; particularly the statistical and physical view, section 2.1 gets us started. We consider in section 2.2 how data is observed, sampled, ordered in a database and how it is prepared for modeling using data analysis and preprocessing. We introduce various approaches to modeling, and treat the errors and disturbances remaining after modeling in section 2.3. The process of fitting and qualities of a fit are discussed in section 2.4, together with the fundamental limitations of modeling and estimation that are well known from statistics and system theory. Readers with sufficient background in statistics and system theory may skip this chapter.

2.1 Sources: systems and processes

A comparison of system behavior is possible from models. Models can be derived from data. However, data is not the system itself, but only a manifestation of a particular instance of system behavior. Definitions are required to express the origin of data such that we can distinguish the actual system from the manifesting behavior. In the first subsection we identify different paradigms for describing presumed systems, in the second subsection we give the general definition and view on systems that will be used. In the third subsection we clarify the notions of instance, realization and manifestation.
2.1.1 Systems and processes

Modeling system behavior requires a description of what a system is. Starting from a system-theoretical point of view a system is a set of interacting processes, or an explicitly controlled system where controlling processes are distinguished by having particular objectives to achieve with the system as a whole. Modeling approaches are characterized by the beliefs they express in the descriptions of the physical reality. Modeling approaches are characterized by the assumed nature of the target process. Two complementary paradigms dominate the scientific world:

- the deterministic belief that all behavior results from unique state transitions governed by physical principles and laws of nature; and
- the stochastic belief that a process is partly governed by random mechanisms.

The formulation of an exact model from the physical principles assumed to govern the process allows a verification of the expected behavior as observations are to match the dictated dependencies. There are good reasons to consider process behavior to be the realization of a set of multivariate stochastic variables as formulated in definition 2.1, also called a random process. For the sake of simplicity we consider only variables in $\mathbb{R}$.

**Definition 2.1: A random process**

A random process is a ordered set of random (vector-valued) variables $(X_t)_{t \in \mathcal{T}}$, with each of the variables $X = (X_1, X_2, \ldots, X_p)$ taking values in the domain $\mathbb{R}^p$, where $X_i$ takes scalar values in $\mathbb{R}$.

A first reason for a stochastic formulation, i.e. not assuming determinism and structural identifiability / predictability of the source is that truly random components may be present in the process. A second one is that even behavior of deterministic processes may be unpredictable; chaotic processes are deterministic in nature, yet behave unpredictable and seemingly random for an observer to whom the underlying dynamics have not yet been revealed. Thus only an “incomplete” model, relying on a stochastic framework, can partly explain the behavior. A third reason is the anticipated need for detecting new unknown structure in a process. Our framework needs to deal with a process of which the structure is unknown beforehand, but this will be explained later. Impatient readers may continue in chapter 5.

2.1.2 Information source

The random variables are assumed to have a particular distribution and mutual dependency, the latter also called structure. These distributions and structure in a random process is called information. Expected behavior can only be described from observations if an underlying structure is assumed to be imposed on the dataset by a process which we will call the information source. The invariant of the information source is $\mathcal{I}$ while it’s variations are determined by configuration $\theta \in \Theta$, see figure 2.1. A realization of the stochastic process with a configuration $\theta \in \Theta$ shows as instance $I_\theta$ of information source $I$. This structure is expressed [Amari, 1990] by stochastic variables and associated probability. For instance, in equation 2.1 the stochastic discrete-time equivalent is shown of the continuous-time dynamical system formulated in equation 2.2.

$$I_\theta = \{(X, Y), p_\theta(X_{n+1}, Y_n | (X_n, U_n))\}$$ (2.1)
\[ \dot{x} = f_\theta(x, u) \]
\[ y = g_\theta(x, u) \]

(2.2)

Our viewpoint on information sources subject to detection is that that of self-regulating processes. It is therefore futile to distinguish between controlled and controlling functionality. Typical examples from economy or ecology motivate this viewpoint of co-existence rather than of subordination.

**Definition 2.2: information source**

An information source \( I_\theta \) determines the distribution and dependencies of the variables in a random process \( \{X_t\}_{t \in T} \). An information source is expressed as a set of probability density functions \( \{p_\theta\} \), parameterized by \( \theta \in \Theta \) with

\[ I_\theta = (\Theta, \{p_\theta\}_{\theta \in \Theta}) \]

Figure 2.1 : Information sources are processes observed through input-output behavior \( \{y_t = (u_t, y_t)\}_{t \in T} \), having internal state \( x = (x_1, x_2, ..., x_n) \). The behavior of the process depends on its configuration \( \theta \).

The configuration of an information source refers to the presumed invariant structure of a system. Invariance is only violated in the case of system changes, which is not the same as state changes. This difference is indeed fairly ad-hoc, but is motivated by the objective to support a control systems oriented view. This objective is only meaningful if we distinguish sensory, state and design variables, as explained in subsection 2.1.3.

**2.1.3 Configuration, state space and manifestation**

Starting from the stochastic view on information sources we require descriptions of dynamic systems as well as notions of observable internal state and other hidden factors. Hence we distinguish between the realization of the stochastic process \( \{X_t\}_{t \in T} \) and the observed behavior being the data \( \xi = (v_t)_{t \in T} \). The manifest behavior \( \{v_t\}_{t \in T} \) is the manifestation of the stochastic process \( \{X_t\}_{t \in T} \).
Figure 2.2 : Models are estimates of properties from realizations of instances of a random process

A control system-oriented perspective is embraced by distinguishing some types of variables, essential to control-oriented modeling. These types of variables are:

- **Observables: sensory and actuator variables**
  Sensory and actuator variables take the value of measurements at a certain time (e.g. the speed of a car). Also one finds control variables (e.g. position of a gas-throttle) and condition variables (e.g. engine temperature). They describe the manifest process behavior.

- **State variables**
  Variables used to describe the assumed internal (hidden) state of the system, e.g. energy consumption, engine wear. In the context of this thesis state variables are considered to be inferred by knowledge of the process, i.e. they are called white-box parameters. Latent variables are those merely introduced for analysis and computation in the model construction process. They are not uniquely determined by physical principles governing the process. The variables of black-box data-driven models are internal adaptive parameters.

From a dynamic system point of view, the realization $x_i$ of the stochastic process $X_i$ may be thought of as the state of the process $t_i$ at sample times $t_i$ such that $v_i = g(x_i)$. We will use the notation (see Appendix A):

- $y = v^{(\text{sensor})}$ are sensory outputs in the context of process dynamics
- $x = v^{(\text{in})}, y = v^{(\text{out})} \text{ models } y = M(x)$ in the context of input-output.
- $u = v^{(\text{steer})}$ are the steering variables of a process.

The inherent dependencies in a system, realizations of random processes and manifest behavior are different aspects of a system. Distinguishing them is essential, since variations in realizations are not due to system changes, and manifest behavior that is observed does not reveal all there is to know about the system. The state space of a system contains the time-variant dependencies of a system, as state variables represent a system’s internal state, which is not always directly observed.
Chapter 2

2.2 Data: observation and sampling

The sampling of a random continuous process provides information by discrete data: data that is essential to model system behavior. Specific features relevant to the objectives and quality of the modeling can be isolated using the properties of the data by pre-processing. The characterization of the dynamic data is essential to select the modeling approach. Sampling and data are discussed in the first subsection, in the second subsection data analysis is described to estimate properties of dynamical data. The third subsection discusses the common transformations to simplify the modeling for known problematic properties of data.

2.2.1 Data sampling

Data are extracted by sampling and holding the value at a specific time. Each individual measurement is called an observation that can be a vector of values.

**Definition 2.3: sample**

An observation \( v = (v_1, v_2, \ldots, v_p) \) with \( v \in \mathbb{R}^{(v_1)} \times \mathbb{R}^{(v_2)} \times \ldots \times \mathbb{R}^{(v_p)} \) is the realization of a set of measured variables \( V = (V_1, V_2, \ldots, V_p) \), which are determined by the random process \((V_t)_{t \in T}\), with \( V \in \mathbb{F}^r \). A series of observations is called a sample, denoted \( \xi = (v_n)_{n \in N} \).

Though the dynamic dependencies may be expressed in terms of continuous-time variables, observations in samples are ordered by a discrete-time index, while the realization of the stochastic process \((X_t)_{t \in T}\) are continuous in time \( T \subseteq \mathbb{R} \). The manifest behavior \( V = (V_1, V_2, \ldots, V_p) \) becomes discrete-time through zero-order hold sampling of the signals at fixed times:

- **equidistant sampling** \( t_n = n \cdot \Delta t \) or in realizations \( v[n] = f(x(n \cdot \Delta t)) \)
- **non-equidistant sampling** \( t_n = s(n) \) with \( s(n) \) a strictly increasing function of \( n \)

**Figure 2.3** : A database is a time-ordered collection of samples

In this thesis we consider variations across multiple instances of information sources. Therefore it is convenient to organize samples in a database. The total of available samples at a certain moment \( t \) is called the sample database \( D = (\xi_i)_{\xi_i \in D} \). Though the samples can be taken randomly from the database, we assume a time-ordering of the samples within the database, such that indices correspond to temporal ordering (figure 2.3) The data acquisition time of an
instance of an information source, from the start of use of a certain model for this particular instance, up to the last observation is called the decision interval.

### 2.2.2 Data analysis

The modeling task is to describe system behavior by capturing relations present in the data. There is usually an implicit order of observations in time; in some cases the order of the observations is important to explain the observed behavior. This may be obvious, for example when a dynamical process is to be identified, but often it is not obvious. Determining whether data has dynamic dependencies (and if so, what time-scale is required) is a problem for which no universal solution exists. If the order of observations is important, a sequence of observations is referred to as a time-series. Once it has been determined that the problem at hand requires dynamic data modeling (meaning the data should be considered a time-series). A type of model and an appropriate representation of the data should be chosen to fulfill the modeling objectives. Such representation is obtained by preprocessing the data, which is the subject of the next subsection. First we address the issue of characterizing dynamic data.

An overview of properties and an analysis of these properties in the context of the design of neural networks for time-series modeling was provided in [Venema, 1999]. The proposed characterization of time-series is shown in figure 2.4. A short discussion of these properties and the methods to analyze them is given here to identify problems.

![Characterization of a dynamical data](image)

**Figure 2.4**: The Venema [Venema, 1999] characterization of a dynamical data

**Time scales and periodicity.** Prior to modeling one needs to determine: a) the particular number of inputs delays and internal state variables required to model the time-series, i.e. the largest time-window (MA-order); and, b) the number of state variables (AR-order) needed to
capture all the dependencies in the series. Essentially the problem is that the temporal depth and the functional dependencies have to be determined simultaneously.

**Tests for time-scale and periodicity.** The chicken-and-egg problem is approached by assuming a model. Two typical models are: 1) assuming linear dependencies such that correlation analysis can be applied, then auto-correlation and cross-correlation reveal at which delay a dependency is found; 2) assuming composition with periodic functions, then convolving with sine-waves of various frequencies reveals if the data contains periodic signals. Essentially this is Fourier analysis [Venema, 1999]; Brockwell & Davis 1987/1996]. Alternative methods to estimate periodicity are maximum entropy and Run-Length Analysis.

**Stationarity.** The most elementary property of time-series is that of stationarity: unchanging properties in time and similarity of random processes. This is defined statistically [Brockwell & Davis, 1996] as in definition 2.4.

**Definition 2.4: stationary process**

The time-series \( (x_t)_{t \in Z} \) is said to be weakly stationary if: (1) it has finite energy for all observations \( E[x_t]^2 < \infty \); (2) it has a non-varying average over time \( E[x_t] = m \) for constant \( m \) and; (3) it has a time-invariant auto-covariance \( \gamma_x(s, t) = \gamma_x(s + h, t + h) \). A process is strictly stationary or shift-invariant if the simultaneous distribution \( (x_0, ..., x_t) \) equals that of \( (x_0, ..., x_{t+h}) \) for any fixed \( t \).

**Tests for stationarity and normal distribution.** Venema describes a test for the variables in a stochastic process to be independent identically distributed (i.d.d.). The Lilifors statistic quantifies the “whiteness” or Gaussianity of the data. The Kolmogorov-Smirnov test is a non-parametric test (that is distribution independent) to compare two distributions. Non-stationarity is also shown by the presence of a trend. A trend test can be performed by fitting a line to the data. White noise testing can be done using the cumulative power spectrum. This should be a straight line for normal distributions since all frequencies are equally present. Histogram methods can be used similarly to test for the type of distribution relying on a chi-squared goodness of fit [Venema, 1999]. Presence of higher-order moments in the data are also indicators of non-whiteness. Skewness and kurtosis provide robust indicators of non-stationarity [Friedman]. Robust tests do not rely on assumptions of a particular distribution; they are called non-parametric, e.g. histograms and order statistics (sorting of data).

**Linearity Test.** A single-sided hypothesis test to do a linear fit with F-test is the most straightforward approach, alternatively the T-test statistic or Brock-Dechert-Scheinkman test can be used [Venema, 1999]. Non-linear Testing can be model-based (fit non-linear model); however, model-based statistics are beyond the scope of data-analysis.

**Test for switching and regimes.** The presence of multiple stationary regions can hardly be detected without a model. Visual inspection offers a pragmatic alternative, though it is slightly more demanding with regard to the intelligence of the so-called human expert. Scatter plots of one variable against a (delayed) variable is a very convenient graphical means. Mutual information and mutual entropy allow for automatic coarse detection of non-random dependencies, such that a large number of variable vs. (delayed) variable dependencies can be inspected. Continuous state-transition behavior, e.g. speech, can be revealed using Hidden Markov Models. However any statistic revealing presence of underlying states relies heavily on a correct estimate of the number of regimes or states to be detected [van Veelen, 1998].
Data-analysis, either by analytical methods or by statistics, is the most time-consuming phase in model design. A human expert, with thorough understanding of underlying physical principles, still outperforms automatic data analysis. In general increasing availability of computation power results in the increased use of brute-force simulation to empirically determine properties from a model by random sampling of the parameter space. The boundary between data-analysis and model-design is totally obscured by such an approach.

### 2.2.3 Preprocessing

There are several properties of measurements that prevent a straightforward modeling of the data. Data analysis can reveal several of these properties, such as dependencies between variables, periodicity, outliers, trends, non-linearity and non-stationarity. Most modeling approaches will benefit from the design of variables such that these properties are corrected. The design of model variables is also called data coding. There are some rules of thumb that can be applied [Masters, 1994]:

- Select variables from the observed data containing real information.
- If the data lies in clusters, either a) analyze and separate the effects or b) remap the data to equalize to cluster effects (e.g. using histogram equalization).
- If there are multiplicative responses, use a log-transform.
- If there are occasional huge values (outliers), use a compressing transformation. E.g. the log-transform or the square root which is milder with respect to the regular range. A cubic root transform can be used if non-negative values are present.
- If there are discontinuities, see if they can be prevented. E.g. for angles from 0-360 degrees, code them in two variables with the cosine and sine.
- If there are ranges that have a special meaning consider fuzzy membership coding

Relatively strong varying variables tend to get focus at the cost of others; offsets in the data obscure other information that is present. There are some coding transformations that improve most estimation procedures as they remove bias from the data:

- Scaling of the data is common for equalizing the information. Using the mean, or median in case of outliers, to center the data removes the offset.
- Using standard deviation, or interquartile range if there are outliers, helps to normalize and equalize the variables with respect to each other.
- In case of clusters, the class membership is best coded with one variable per class. Never use a single variable with different values to code different classes. Sometimes it helps to code outliers with a flag.
- If there is a physical or natural reason for assuming a trend, de-trending can be used. Detrending is possible by differencing the series: finding a least-mean square trend is however more stable and reliable.
- If variables are dependent, then new variables can be derived that are independent. Principal component analysis can be used to orthogonalize the data and even to reduce the data. If there are classes and the distribution within classes is similar, linear discriminant
functions can be used to emphasize class membership. They should be used in addition to other variables if there are non-linearities.

• If there are limited seasonal/periodic effects, isolation of these effects can be used to compress the data by resampling the data. Band-pass filters and down conversion or Fourier domain filtering after DFT can be applied, though care should be taken for aliasing effects. Using a DFT should be combined with windowing to improve the isolation. When there are only a few peaks the maximum entropy method is more efficient and accurate than the DFT, since it is not bound to equidistant frequency bins.

Having non-uniformly sampled data is a problem for time-series prediction. Generic solutions go at the cost of precision for this issue. We have three suggestions: 1) either use a kind of binning in a N-dimensional histogram and resort to probabilistic methods; 2) transform to another domain and then project back to the original domain on a regular grid (e.g. DFT followed by an inverse FFT); 3) use interpolation. For time-series use interpolation filters, first upsampling and then downsampling (e.g. with a polyphase filterbank). Upsampling can be chosen so that the sampling times of the original series approximately fall on the new grid.

The data should always be prepared with wisdom, as the coding itself can introduce biases in the data. A proper application of a transformation depends highly on the accuracy of the human data analysis. Estimation procedures are designed to optimize certain cost-criteria. Since pre-processing is part of the modeling it is questionable to optimize manually rather than subjecting the preprocessing to the same model optimization procedure.

2.3 Models: architecture and parameters

There are several modeling approaches to predict, control or capture dynamics from samples. The type of modeling is partly determined by the objectives. The nature of the data generating process that is characterized by data analysis finally determines the capabilities that are required of a model. In the first subsection we briefly state the common modeling objective: time-series modeling. An overview of various types of models is provided. Independently of the quality of modeling to be quantified, we define error, residual and disturbances in the last subsection.

2.3.1 Objectives and definitions

The objective of modeling is to capture the behavior of a system. The motivation is usually to force a system into a particular desired operation or to anticipate development of a complex process. The two main drivers are to increase control over a system and to gain understanding of a complex process. In either case it is essential to capture dynamics or predict. The objective is to find a mapping between the history and the future of manifest behavior. This is called time-series modeling, described in definition 2.5.

Definition 2.5: time-series modeling

Given a stochastic process as defined in (definition 2.1), that generates sequential measurements of variables \( v_1, v_2, \ldots, v_p \), with \( \mathbf{v} = (v_1, v_2, \ldots, v_p) \in \mathbb{R}^p \), the task is to determine the dynamic relations \( v_{[n+k]} = f_{[k]}(n, v[n], v[n-1], \ldots) \). Note that we have assumed causality here as we model mappings rather than general dependencies.
We distinguish two aspects of time-series modeling. The first is the general issue of describing relations between variables. The second is the incorporation of dynamics or time-dependency. There are three elementary principles behind dynamics in systems that can be represented by models: 1) dependency on a periodic reference signal (clock), 2) state feedback, and 3) delay. These elementary principles are also elementary dynamical model architectures: 1) clocked models, 2) feedback or auto regressive (AR) models and 3) moving average (MA), tapped delay and time delay models. These dynamical model architectures can be combined. We distinguish three types of models related to the general issue of describing relations between variables: a) probabilistic models, b) function approximators; c) physical-principle models. In the next subsections we describe how these types of models are used for time-series modeling.

### 2.3.2 Distribution estimation

The most straightforward approach towards data-driven modeling is by describing where the data is likely to occur in the data space. This approach aims to estimate the probability densities as enforced by the information source, equation (2.1). Data distributions can be estimated for each single variable (the so-called marginal distribution \( p(v_i) \)) and for simultaneous distributions \( p(v) \). No doubt, the data itself is the most accurate description thereof. However a more compact general representation is sought. To achieve a compact representation one assumes that the densities can be expressed by some common estimator structure, i.e. an assumption is made of the data distribution. The model is a (set of) probability distribution(s) on or density estimates, or membership functions on one or more variables or transformations thereof, similar to equation 2.3.

\[
\{ \hat{P}(V_j[n]|V_i[m<n]), \hat{P}(V_i, V_j), \hat{P}(V_k) \} \tag{2.3}
\]

Commonly one distinguishes between parametric and non-parametric estimators. The difference is sometimes hard to tell; maybe that is why semi-parametric estimators have been introduced. The actual existence of truly non-parametric models is questionable. As the model is based on assumed structure and fitted data, it will be biased with respect to the used data and the assumptions of the designer.

- **Parametric estimators**
  The estimator effectively describes data having a specific type of distribution while the performance is not guaranteed and often unknown for other distributions. The assumption of parametric estimation is that the “true” parameters \( \theta \) are estimated. Our point of view is, similar to white-box modeling: if the type of distribution is known, it should be adopted as a model. The major advantage of such estimators is that the data can be described with few parameters. Thus it is very compact, e.g. mean and standard deviation for normal distributions \( \theta = (\mu, \sigma^2) \), hence the name parametric. Assuming a distribution \( p(x|\theta) \) of variable(s) \( x \) parameterized by \( \theta \), the goal is to find \( \theta \) itself. There is a vast amount of literature on statistical parametric estimation [Hancock & Wintz, 1966; Him melblau 1978; Patton, Frank and Clark, 1989].

- **Non-parametric and semi-parametric models.** [Sprent, 1989].
  Where less is known about the “true” distribution of the data, more degrees of freedom are required to obtain a general model, i.e. different parameters have to be determined for various regions/dependencies. Histograms \( \hat{h}(\theta) \) (equation 2.4) and percentiles \( \hat{\mu} \) (equation 2.5) are examples thereof. Kernel-based methods [Desforges et. al, 1998; Taylor
lie somewhere between non-parametric and parametric. The major advantage of such semi-parametric methods is that a much larger class of information sources can be modeled. Disadvantages are the larger number of parameters that have to be determined, i.e. less compact. Further one tends to forget that these estimators also rely on some implicit assumptions.

\[ H_i^{(h)}(X) = P(X \in [h_i, h_{i+1}]), \quad H(X) = (H_i^{(h)}(X))_{0 \leq i \leq |h| - 2} \]  

\[ p_{r \in \{0, 1\}}(X) \equiv \arg \max P(X < r) = r \tag{2.5} \]

For non-deterministic detection, distribution models are certainly the most popular and common method [Taylor, 2000]. They have the advantage that no explicit target value is required to estimate a model. The probability of the occurrence certain dynamics does not require a target to predict, so the likelihood of a sequence of observations is instantaneously available.

### 2.3.3 Function approximation and regression

Probability density estimation is seemingly complete in the sense that it assigns a probability to every possible realization of the stochastic process. Yet practical density estimators are often non-parametric or at least semi-parametric as data hardly ever has a known distribution and non-parametric methods scale badly. As a consequence models fail to reflect the essential relations between variables compactly or, so to say, fail to represent the structure of the data efficiently which was in fact the goal of modeling for detection. A good balance between generalization and modeling error depends on the data size but also on the objective of the model. In case of detection the objectives are usually accuracy and robustness, we review these objectives in chapter 5 and 6.

A more directed search for structure in the data can be realized by representing only the dependency itself as a function. Note however that only surjective relations can be modeled adequately this way. Recall that the general objective of modeling for detection is to find all the present static and dynamic relations [Rault & Baskiotis, 1989] in the data to represent the system behavior. The model is a set of functions between variables for which an error measure is defined. The functions can be static (memory-less) or may contain some internal state.

\[ V^{(in)} = M_w(V^{(out)}) \]  

\[ V_i = M_w(Z^{(k)}V[n]) \]  

One special class of function approximation models that actually relate observed variables to themselves are called auto-associators. The goal of such models is to find a minimal, orthogonal or separated representation of the data. There are three classes of function approximators:

- **exact**: dynamical systems \( M = f \), control system theory \( M = (f, g) \),

\[ \dot{x} = f(x, \theta), \quad \dot{x} = f(x, u) \]  

\[ y = g(x, u) \]  

- **statistical**: time-series models ARMA-models and variants (ARCH, GARCH etc.)
A special application of function approximation is auto-association. An aut-associator consists of a projection and inverse projection, into and from a space in which a data point can be represented efficiently. Examples of auto-associates are principal component analysis and independent component analysis.

It can be very hard to determine how large a model should be, i.e. how much of the dynamics it should model or how accurate it must and can be. The size, structure and order of a model in relation to that of the information source determines the model quality. Estimating the required order is attempted in data reduction and blind source separation. One example in auto-association is: how many variables are needed to represent the signal? Order-estimation is tough but it can be used to characterize the data in a more general sense: a feature that makes a potentially robust detector for structural changes in the data. Hence especially order and complexity estimators for neural models will be considered for data-driven detection in this thesis.

A model is always a limited representation in the time-frequency domain of the manifest behavior. As a result some dependencies will not be captured. However detection of complex disturbances does not always require a complete model of "normality". Actually it is possible to obtain an accurate change detection and estimation with a simplified model of the data source [Basseville, 1988], as long as the model expresses the aspects related to the desired operation of a system. This also holds for physical-principle models. Hence exact identification of the information source is not a necessity for high quality detection. Even more, the presumed quality of physical principle detection does not necessarily exceed that of data-driven black-box detection.

### 2.3.4 Physically plausible models

Knowledge of the information source, perhaps originating from the design of the process, can provide a complete overview of the physical principles, i.e. "the physics laws", which apply to that process. From such an overview one may deduce the structure of the model. In a conventional controller systems theory one infers differential equations from the presented physical laws [Olsder, 1994]. As non-linear continuous-time models are not easily manipulated and analyzed, commonly procedures are linearized and transformed to a discrete-time model [Olsder, 1994]. Theory for processes that are assumed to be linear time-invariant (LTI) (see equation 2.10), i.e. the structure and coefficients of the parameters are fixed, is well developed. Non-linear and time-varying processes require non-standard case-dependent analysis.

\[
y[k + 1] = Ax[k] + Bu[k] \\
y[k] = Cx[k]
\]  

(2.10)

In a white-box all the process components as well as their behaviors are known. Each parameter in the model is determined from the process components. The internal state of the observed process can be estimated through state observers, when the structure of the process allows for it, as the variables and parameters of the model correspond to the physical quantities. In control system design one aims to find an optimal observer-controller pair to steer the
system \((A, B, C, D)\) toward a desired target output \(y^*\) through computation of proper inputs \(u\) given previous inputs and observed outputs \(y\). The possibility to do so is limited by the controllability of the process, i.e. each possible state must be reachable by providing the right sequence of inputs.

![System Model Diagram](image)

Figure 2.5 : A system controller relies on accurate tracking of the system's state. A construction of a conventional observer-controller pair depends on knowledge of the physical laws governing the system.

Physical principle and more generally white-box models aim to represent the actual dependencies within a process rather than just to mimic the input-output behavior. In white-box modeling there are two types of identification:

- **identification of structure** [Hof, 1996]:
  Process identification is the determination of the structure of the information source \(I\) from measurements \(\xi\) of the manifest behavior \((V)\).

- **identification of parameters**
  Parameter identification or estimation is the determination of the value of the properties \(\theta\) underlying the information source from measurements \(\xi\) of the manifest behavior \((V|\theta)\) of the information source \(I_\theta\).

The use of process models enables the estimation of process state variables and parameters that are influenced by faults. To realize sensitive detection, the internal state must be traceable with high precision and the parameters must be estimated very accurately [Isermann, 1984].

The assumption is that, if \(M = I\) holds, the physical properties of the information source map one-to-one on the model parameters, i.e. \(w = f(\theta)\) and \(\theta = f^{-1}(w)\). Physical principle approaches are the only viable approaches for detection according to some. As Isermann puts it: “Process models should express as closely as possible the physical laws which govern the process behavior. Therefore ... requires theoretical modeling” [Isermann, 1984].

Systems theorists tend to dislike non-determinism, which is probably the reason that noise and uncertainty are not considered part of the model. However strictly speaking any quantitative remark on the robustness of the model in terms of tolerances of the observed variables should be included. An assumption on the amount or character of the noise can be included in the model and the uncertainty of the dynamic relations as modeled according to equation (2.1) can be estimated from the data.
### 2.3.5 Black-box models

If a white-box modeling approach is feasible, given the knowledge of an information source, a conventional fault detection approach based on identification and classification of the process’ physical properties is advised. If the conventional approach cannot be the only constituent, then it should serve at least as a basis for any further modeling. However, the feasibility and effectiveness of white-box models is limited in practice. Hence there are several good reasons for the use of a black-box modeling approach:

- A mathematical physical-principle model will often be too complex to gain insight into the behavior of the information source as a whole. Mathematical truth does not automatically lead to understanding.
- White-box modeling is a reductionistic approach based on assumed knowledge. Instead a redundant representation can be more sensitive to abnormalities and yields a model less biased toward the assumed structure of the information source.
- The physical principles or laws governing the information source are not fully known, i.e. $I$ is not known and cannot be uniquely identified.
- The structure of the information source prevents observability of the internal state of the information source, thus requiring a simplified and inaccurate estimate.

In black-box modeling the goal is not to identify the actual structure of the information source but to mimic its input-output behavior as close as possible. The structure of the model nor the parameters have an intended physical meaning. In general the parameters of black-box models are not expected to relate in any specific way to the physical properties of the information source. Any interpretation in terms of behavior of the information source is expressed in terms of observed variables. To put it differently: the quality of the model is not given by the correct identification but by the distance between model behavior and observed behavior. The model only imitates the information source; they are *behaviorally* approximately equivalent.

### 2.3.6 Errors and disturbances

Error functions measure discrepancies in manifest variables. The error as a function of the data depends on the applied type of model. In case of a dynamic model the error cannot be computed independent of the state of the model. This issue is resolved by taking the state $s$ of the dynamic model $M(s)$ as an input and output argument, see appendix A.

**Definition 2.6: error**

An error is a measure for the distance between distributions of observable variables $V = (V_1, V_2, ..., V_p)$ from a data source and the expected distributions as expressed by a model $M$. Given a single pattern $x$ the pattern-error is denoted $e^{(M)}(v)$ for static models and $e^{(M)}(s, x)$ for dynamic models in state $s$; for sample $\xi = (v_n)_{n \in T}$ the sample-error is denoted $e^{(M)}(\xi)$.

Such a distance can be estimated in several ways, depending on modeling approach and objectives, i.e. function approximation. Dynamic models and density estimators such as clustering require different error measures. For now we will only assume that these deviations can be measured and that they represent the quality of a model.
The instantaneous deviations, i.e. deviations for each separate observation in a sample $\xi$, are also called residuals. The projection is a model dependent operator on the stochastic variable $\mathcal{E}(V)$. The series of the instantaneous deviations $\mathcal{E}(\xi) = (e^{(M)}(x_n))_\xi$ is referred to as the sample residue $\xi$.

Abnormalities cause the structure of the process to change. Thus, if the quality of the model allows, the structure of the residual will change. The computational comparison between observed and expected behavior can be made (equation 2.11) with the residual. Under the assumption of a perfect model, and in absence of observations of normal behavior, it can be stated that $\mathcal{E}((V_i)|\theta) = 0$. Such residue-based detection methods are discussed in chapter 4.

$$\mathcal{E}((V_i)|\theta) = \mathcal{E}((V_i)|\Theta^{\text{normal}}) \quad (2.11)$$

**Figure 2.6**: Error functions measure the difference between observed behavior of the information source and expected behavior as described by a model of the information source.

Errors will be present in most models estimated from measured data. In line with the discussion in 2.1.2, a distinction between “unstructured errors” and structural errors is called for. Such can be based on the amount of structure or dependence; measures thereof have been discussed in the context have data analysis (subsection 2.2.2). For now we assume a quantification of the amount of structure $\iota(x_j)$ among variables in the vector $x_j$. The measure $\iota$ can be thought of as statistical properties such as means and trends for individual variables and correlation or mutual information for sets of variables.

**Definition 2.7: disturbance**

A disturbance is the presence of a significant amount of structure in the residual with dependence on the model inputs or time.

Given an appropriate measure of information or statistical property $f$, abnormalities can be described from the residuals in either of the following ways $\iota(\mathcal{E},f(x),\theta) = f(\mathcal{E},\Theta^{\text{normal}},\theta)$ or $\iota(\mathcal{E},f(x),\theta) \neq f(\mathcal{E},\Theta^{\text{normal}},\theta)$. Taking into account the effect of insufficient data and noise, a solution for detection can easily be provided if abnormalities are the only cause of disturbances. The structure of disturbances $\iota(e^{(M)})$ can be found on three phenomena (figure 2.7):

- **modeling artifacts (model bias)**
  These limitations of a model structure and parameter configuration are already contained in the errors of the model on samples in the history database. Such imperfections can be measured from the unmodeled structure, e.g. $\iota(x,e^{(M)})$. The dependency of observed vari-
variables and errors being $\hat{y} = M(x)$ and $e = \hat{y} - y$, then the amount of unmodeled structure is $I(X, Y) - I(X; M(X))$. Dependencies, introduced by the model, and are not present in the structure imposed by the information source are called model artifacts.

- **noise, unbalanced representation of the information in the samples (sample bias)**
  The modeling error may be different for fitted/training samples and test or validation as a result of unbalanced data. Although this is a measurement problem, such effects on the probability distributions of the variables should be considered as property of the information source.

- **abnormalities/changes in the information source** are the subject of this thesis. An analysis of abnormalities, disturbances and faults will follow in chapter 4, 5 and 6.

---

**Figure 2.7**: The structure in residuals is unmodeled structure and modeling artifacts

---

### 2.4 Estimation: fitting, quality & limitations

The estimation of parameters from data is essential in modeling the behavior of a system. Estimation is applied to determine an initial model for the system. However, estimation can be applied also for accommodation of disturbances during the operation of the system, or for parameter-based detection. Procedures for estimation are discussed in the first subsection with a focus on learning processes. The bias-variance problem is explained, as it is essential in choosing a model complexity and estimating quality. Performance and error measures typical for dynamic modeling are described in subsection 2.4.3; these are valid for any type of model. Essential properties of dynamical processes are taken from control system theory in subsection 2.4.4. In subsection 2.4.5, the complexity estimation problem is stated. We close the chapter with an overview of the limitations of estimation.

#### 2.4.1 Procedures for fitting data as a learning process

The models discussed in the previous section consist of a fixed architecture and adjustable parameters. The parameter must be selected from data to minimize errors. The retrieval of “ideal” parameters can be seen as solving a finite set of equations, with each observation being an equation. The parameters are always a guess in case there is noise in the data, since the data is only one realization of the stochastic process to be modeled. Although various error measures can be used the most common is the quadratic error. The minimization of the quadratic error, given a data set, is a least squares solution. In linear models the least-squares
solution can be found through matrix inversion. Another class of estimation procedures is provided by iterative search procedures. An iterative search procedure is a learning process if it follows a non-random path through the parameter-space that is guided using feedback of data dependent error or cost-function. The design task is to configure the learning process such that this “best” configuration can be reached. The basic goal of a learning process is as follows:

*A learning process has the goal of steering the models parameter to a “best” configuration by feeding the proper signals through the selected model structure.*

While the learning process is in fact a feedback system, it’s behavior can also be discussed in terms of stability, convergence, robustness and states from a control perspective and in terms of attractors, equilibria and bifurcations from a perspective of dynamical systems. The task is thus actually to design a non-linear controller with restricted steering patterns. To clarify, only the patterns in a database can be used to steer the models adaptive parameters. Some components of the learning process can be configured to obtain a desirable learning behavior and in the end a reasonable model. The learning behavior should be identified to configure these components in the learning process. The behavior of the learning process is mainly determined by three interacting aspects:

- **DATA: the mapping to learn defined by the available data**
  The data requires a certain amount of memory and state variables. Hence it determines the architecture required to model the underlying process. While the model architecture directly influences the behavior of the learning process, other secondary properties such as balancedness of the data, uncertainty, amount of data, outliers and disturbances have a more direct impact on the learning behavior. These secondary properties do not immediately give a search direction for the model, but imply the limitations of that model and induce learning problems.

- **MODEL: the architecture of the model**
  The architecture of the model as well as the size of the state vector determine the location of the weights to be updated and inherently some dependency among them. Due to the connectivity an optimal parameter configuration is often not uniquely determined by the data, even if no uncertainty in the data is present.

- **UPDATE: the update function**
  There are many forms of learning. The gradient descent approach has appeared before, some other typical neural learning algorithms will be discussed in chapter 3.
Learning is called a process as supervised learning is inherently a feedback system. The system is initialized with certain parameter values. The internal state as represented by the parameters is altered by evaluating the model and updating the adaptive parameters from the measured errors. This step is iterated until a predefined stopping criterion is reached. The basic pattern learning process is uniquely characterized by the following ingredients:

- **the model architecture**
  The model architecture is a template describing the connectivity of the components.
- **the initial weights**
  The choice of a range from which initial weights are chosen affects the learning behavior and the complexity of the model.
- **a learning function for step-wise improvement**
  \[ \Delta w[n + 1] = l_\gamma(\Delta w, w, v) \]
  A learning function consisting of a search direction and associated parameters \( \gamma \). The pattern learn function is extended to a batch learning functions in the following manner:
  \[
  \begin{align*}
  \varepsilon_\gamma(\Delta_0, w, \emptyset) &= w \\
  \varepsilon_\gamma(\Delta_0, w, (v_0, \ldots)) &= \varepsilon_\gamma(l_\gamma(\Delta w, w, v), w + l_\gamma(\Delta w, w, v), (\ldots)) 
  \end{align*}
  \] (2.12)
- **the training data**
  \( \xi^{(\text{train})} \)
- **the testing data**
  \( \xi^{(\text{test})} \)
- **a stopping criterion**
  \( C(w, \xi^{(\text{test})}) \)
  A common choice to terminate the learning process is the stability of the training error, or the squared error dropping below some a priori known, assumed, noise level.
Given these definitions we can formalize a definition of a learning process. The (non-randomized) learning of a (dynamical) model is a process with internal state defined by (state vector $s$ and) weights $w$, initial state $(s_0, w_0)$, state dynamics expressed by the update function $l$, while training sequence $\xi^{(\text{test})}$ and final states are determined by the predicate $C(w, \xi^{(\text{test})})$ called the stopping criterion.

The performance on the problem has to be estimated also for unknown future measurements. When a model is trained to specific data, it will learn peculiarities in that data set which are not present in general. Hence the apparent error on the training set will be an under-estimate for the actual error. In cross validation (CV) the data is split into a train and a test set of randomly drawn observations or samples. The error on the test-set is considered the better estimate of the actual error. This is valid if both training set and testing set are highly representative of the underlying process. However, often the amount of overlap between testing set and training set is smaller than the overlap between training set and manifestations of the process. Hence the error on the testing set is an over-estimate of the real error. Neural training problems require large quantities of data. The mean and variance in performance are usually estimated to determine the how bad the worst model will probably perform. In the end all data will be used for training the model. They are estimated using a k-fold repetition of the training process with different training and testing sets. This is called k-fold cross validation.

$$\hat{\mathcal{E}} = \sum_{i=1}^{k} \mathcal{E}(w|\xi^{(\text{train})}_i), \xi^{(\text{test})}_i)$$

(2.13)

The process of randomly choosing training and testing data is called resampling, which in case of CV is drawing samples without replacement. For small data sets it may be problematic to draw representative training and testing data without replacement. In such cases a bootstrap procedure [Efron and Tibshirani, 1993] can be used to estimate the performance using all the data for training. A single application model $M^{\text{(application)}}$ is estimated using all the data $D$ for training, while $k$ different pairs of training/testing sets will be drawn from the database with replacement. The error is then estimated from the application model with correction for the underestimation by the average testing-error, which is compensated with the training errors, as in equation 2.14. A discussion on cross-validation and bootstrapping in dynamical neural modeling is found in [LeBaron and Weigend, 1998].

$$\mathcal{E}^{\text{(bootstrap)}} = \frac{1}{n_D} \sum_{y \in D} (y - \hat{y}) + \frac{1}{k} \sum_{i=1}^{k} \left[ \frac{1}{n^{(\text{test})}} \sum_{y \in \xi^{(\text{test})}_i} |y - \hat{y}| - \frac{1}{n^{(\text{train})}} \sum_{y \in \xi^{(\text{train})}} |y - \hat{y}| \right]$$

(2.14)

2.4.2 Risk, bias and variance

The quality of a model is expressed by a cost- or risk-function. The idea of a cost-function is to express the price of making mistakes. This price may well vary over the input domain. The common empirical risk function is shown in equation 2.15. It is an empirical version of the standard risk function for function approximation, equation 2.16. The probability $p(x)$ of a certain input $x$ as it occurs in 2.16 is omitted in the empirical risk function 2.15 as the database is thought to be representative.
These cost-functions aim to minimize the error over the entire data space rather than for individual pattern errors. The common approach towards function approximation using a monolithic model is by minimization of the average square error. The bias-variance dilemma is one of the most fundamental problems in estimation theory. It has been studied extensively in relation to over-fitting [German et. al., 1992] and is considered the crucial factor in data-driven black-box modeling [Haykin, 1994; Bishop, 1995]. The expected error over the entire input space is given by equation 2.17, i.e. the expected mean square error plus the expected conditional variance of outputs \( \sigma_y^2 | x \) [Bishop, 1995].

\[
E[e^2|\hat{y}] = \frac{1}{2} \int (E[y|x] - \hat{y}(x))^2 p(x)dx + \frac{1}{2} \int (E[y^2|x] - (E[y|x])^2) p(x)dx \tag{2.17}
\]

The expected squared error \((y(x) - \hat{y}(x))^2\) can be rewritten according to equations 2.18 and 2.19.

\[
(y(x) - \hat{y}(x))^2 = (E[y|x] - E[\hat{y}(x)] + E[\hat{y}(x)] - \hat{y}(x))^2 \tag{2.18}
\]

\[
(y(x) - \hat{y}(x))^2 = (E[y|x] - E[\hat{y}(x)])^2 + (E[\hat{y}(x)] - \hat{y}(x))^2 + 2(E[y|x] - E[\hat{y}(x)])(E[\hat{y}(x)] - \hat{y}(x)) \tag{2.19}
\]

The third term in 2.19 will vanish as the number of patterns approaches infinity. Hence we arrive at the bias-variance formula by including the unconditional probability density of inputs and approximately weighting the expected output variance from equation 2.17. The bias-variance formula is well-known for quadratic programming problems. The expected quadratic error, equation 2.22, for a given estimate \(\hat{y}\) consists of two terms: (1) a squared estimator bias, equation 2.20; and (2) the estimator variance, equation 2.21. A result of the bias-variance relation is that any type of estimator, parameterized by a set of adaptive parameters \(\mathbf{w}\), is bounded by a lower limit \(E[e^2|\hat{y}]\), where bias can be exchanged for variance and vice versa without decreasing the quadratic error below that lower limit.

\[
\mu^2_y(e) = \int (E[\hat{y}(x)] - E[y|x])^2 p(x)dx \tag{2.20}
\]

\[
\sigma^2_y(e) = \frac{1}{2} \int E[\hat{y}(x) - E[\hat{y}(x)]|^2 p(x)dx \tag{2.21}
\]

\[
E[e^2|\hat{y}] = \mu^2_y(\hat{y}) + \sigma^2_y(\hat{y}) \tag{2.22}
\]

Formula 2.21 and 2.22 conveniently express the basic problem of the bias and variance trade-off. It seems that bias or variance can neatly be selected by the designer, which ignores this limitation of the estimation procedure. The limitations of both learning procedure and model architecture are in fact imposing structure on the residual. The term variance is implicitly asso-
ated with a kind of unstructuredness of the errors in the uncertainty areas denoted by $\sigma_{e}^{2}$. Non-linear modeling procedures will practically always cause a bias in adaptive parameters causing structured errors.

### 2.4.3 Performance and error measures

Given a sample $\xi$, the usual static error measures are the MSE (Mean Square Error) or the Root MSE, where $\varepsilon_{\text{RMSE}}(\xi) = \sqrt{\varepsilon_{\text{MSE}}(\xi)}$, and the MAPE (Mean Absolute Percentage Error), as expressed in equation 2.23.

$$
\varepsilon_{\text{MSE}}(\xi) = \frac{1}{n_{\xi}} \sum_{k=1}^{n_{\xi}} (y[k] - \hat{y}[n])^{2} \quad \varepsilon_{\text{MAPE}}(\xi) = \frac{n_{\xi}}{\sum_{k=1}^{n_{\xi}} |y[k]|} \sum_{k=1}^{n_{\xi}} |(y[k] - \hat{y}[n])| 
$$

For unscaled data one is rather more interested in relative than in absolute measures. The MSE is therefore normalized with respect to the variance of the target values, and is called the NMSE (Normalized Mean Square Error), see equation 2.24. Errors should be expressed on the domain of the estimated variables and state requirements.

$$
\varepsilon_{\text{NMSE}}(\xi) = \frac{1}{n_{\xi}} \sum_{k=1}^{n_{\xi}} (y[n] - \hat{y}[n])^{2} \quad \sum_{k=1}^{n_{\xi}} (y[n] - \mu_{y})^{2} 
$$

These measures are independent of the ordering of the patterns and can be evaluated for any set of observations of compatible dimensions to the model. For function approximation problems, these measures give a reasonable impression of the quality of the model. The amount of unmodeled structure is not expressed using these measures. Alternatives to detect the remaining structure are visual inspection or correlations and mutual information of inputs and errors.

A small RMSE is very misleading in dynamical modeling. A slowly varying series is easily predicted with the model $\hat{y}[n+1] = y[n]$ requiring only one delay. Then the RMSE will be very low but the underlying dynamics are not captured by the model! The performance of dynamical models is often visualized by plotting the actual and the predicted time-series. An exemplary time-series is shown in figure 2.9. The two lines in this case seem very close, and one may falsely conclude that the model is accurate. However closer inspection shows that the predicted series lags behind the actual series. Hence it is not a very accurate model and very close to the identity model $\hat{y}[n+1] = y[n]$. A more accurate performance measure is obviously required. Quantifying the performance is realized by several error measures discussed below. A more exploratory performance assessment is obtained by residual analysis.
A typical approach in time-series modeling is to train with the first part of the time-series and test with the remaining \( p \) observations. So a model based on the first \( n_{\xi} - p \) observations is used for training and the average performance of a step ahead prediction based on the model is calculated, as in equation 2.25. This extrapolation error measure is called a MultiStep ahead Cross Validation (MSCV) error measure, the MSCV is based on MSE [McNames, Suykens and Vandewalle, 1999], where \( M_{i+1,i+p}^{(i+1,i+p)} \) is the model, trained leaving the observations from \((i+1)\) to \((i+p)\) out of the data for validation.

\[
\hat{\varepsilon}_{\text{MSCV}}^{(1,p)}(\xi) = \frac{1}{p(n_{\xi} - p + 1)} \sum_{j=1}^{p} \sum_{i=1}^{n_{\xi} - p + 1} (y[i+j] + M_{i+j}^{(i+1,i+p)}(x[i]))
\]

This error-measure is already quite expensive to compute on-line while often we only want to monitor the learning process or get a rough estimate of the performance for stopping criteria. The Ratio of Squared Errors (RSE) is the ratio of the prediction error of the model and the distance between the current and previous observation, as expressed in equation 2.26. The modeled dynamics are quantified better than by MSE-based measures. The model is compared with the identity model \( \hat{y}[n+1] = y[n] \) if the model equals the identity \( \hat{\varepsilon}_{\text{RSE}}(\xi) = 0 \). The intuitive rationale behind this measures is that for \( \hat{\varepsilon}_{\text{RSE}}(\xi) = \frac{1}{r} \) the model is \( r \) times better than the identity model.

\[
\hat{\varepsilon}_{\text{RSE}}(\xi) = \frac{1}{n_{\xi}} \sum_{k=1}^{n_{\xi}} (y[n] - \hat{y}[n])^2
\]

\[
\text{RSE} = \frac{\sum_{k=1}^{n_{\xi}} (y[n] - y[n-1])^2}{\sum_{k=1}^{n_{\xi}} (y[n] - y[n-1])^2}
\]
While such performance indicators as the RSE and the MSCV are helpful for a coarse representation of model quality, it does not reflect the fraction of captured structure. The fraction of explained structure can be analyzed by other performance measures such as the FVU (Fraction of Variance Unexplained).

\[ \varepsilon_{\text{FVU}}(\xi) = \frac{\text{VAR}(e^2[n])}{\text{VAR}(y^2[n])} \]
\[ \varepsilon_{\text{VU}}(\xi) = \frac{I(x[n], e[n])}{I(x[n], y[n])} \] (2.27)

Alternatively one can measure the amount of unexplained information, for example by the mutual information. It is not uncommon to analyze the correlation between model variables. Examples are:

- **inputs and error**
  The relation between inputs and error measures the “yet unmodeled” part of the structure between inputs and targets. A cross-correlation at different lags between input and error facilitates a visual inspection.

- **The relation between output and target**
  The common error measures rely on the differences between the predicted and actual observations. For a predictor this is the correlation \( \rho_{y'y} \) between output and target. The associated error measure is the one minus squared correlation error [McNames, Suykens and VandeWalle, 1999] or scaled negative squared correlation \( \varepsilon_{\text{SNSC}} = 1 - \rho_{y'y}^2 \).

- **the dynamical structure in the error**
  The error should not contain any structure, when the model has accurately captured the structure in the data. This can be revealed by studying the auto-correlation of the errors at different lags. There should be no significant correlation at any lag.

While correlation analysis is frequently used to study the performance of non-linear models, one should be aware of the limitations of linear statistics. The correlation between inputs and errors is used as a regulating stopping criterion [McNames, Suykens and VandeWalle, 1999]. Though this will indeed prevent overfitting, the limitations of linearity imply limited use of the non-linearity in the model. While mutual information reveals also non-linear structure, but it has undesirable biases due to quantization. The mutual information criterion works well for binary patterns but is not tailored for continuous functions represented by finite samples.

In process identification, sometimes only the issue of the overall model characteristic is raised: is the global behavior of the model qualitatively similar to that of the process? Then the potential of a model class is considered rather than the exactness of a single model. Conventional control system toolboxes can be used to analyze the behavior of the dynamical model. A step response may also involve initialization with a sequence of observations from an actual time-series to simulate the sudden appearance of a signal.

Dynamical models will naturally leave some dynamics in the residuals. Analysis in the frequency domain seems a logical idea. Unfortunately few measures and assessments are based on a spectral analysis of the residual. Typically, the Fourier transform of the cross-correlation between inputs and errors will give a good perspective on the remaining structure. The cross-spectrum of the inputs and error should not reveal any significant dependencies.
2.4.4 Control system theory

To discuss dynamical systems appropriately we require a few system theoretical definitions that express stability, controllability and observability. These aspects can be defined for dynamical systems as differential equations. Computable statistics for these properties exist for the linearized system model, expressed in section 2.3.4. These definitions are required to discuss process behavior as well as the behavior of learning and estimation processes. For a background on these definitions the interested reader is referred to [Olsder, 1994].

**Definition 2.8: Definition of a Stable equilibrium point**

Given a first-order differential equation \( \dot{x} = f(x) \), with the solution where initial condition \( x(0) = x_0 \). A vector \( \pi \) which satisfies \( f(\pi) = 0 \) is called an equilibrium point. Such a point \( \pi \) is stable if for every \( \varepsilon > 0 \) a \( \delta > 0 \) exists such that, if \( \|x_0 - \pi\| < \delta \), then \( \|x(t, x_0) - \pi\| < \varepsilon \) for all \( t \geq 0 \). An equilibrium point is called asymptotically stable if it is stable and, moreover, a \( \delta_1 > 0 \) exists such that \( \lim_{t \to \infty} \|x(t, x_0) - \pi\| = 0 \) provided that \( \|x_0 - \pi\| < \delta_1 \).

The concept equilibrium has already been introduced; it is formalized in definition 2.8. Stability is another essential concept. A practical definition is BIBO stability, as described in definition 2.9.

**Definition 2.9: BIBO stable**

A system if BIBO stable (bounded input, bounded output) if for all zero initial conditions at \( t = t_0 \) every bounded input defined on \( [t_0, \infty) \) gives rise to a bounded output on \( [t_0, \infty) \). The system is called uniformly BIBO stable if there exists a constant \( k \), independent of \( t_0 \) such that for all \( t_0 \)

\[
x(t_0) = 0, \|u(t)\| \leq 1, \quad \text{for all } t \geq t_0 \quad \Rightarrow \quad \|y(t)\| \leq k \quad \text{for all } t \geq t_0
\]

**Controllability**

Observability and controllability are key concepts in understanding fundamental limitations of control over a system, including the learning process of a model. The setup in figure 2.10 clarifies controllability intuitively. Controllability is formally described in definition 2.10.

**Definition 2.10: Controllability**

A system is called controllable if for each arbitrary point two points \( x_0 \) and \( x_1 \in \mathbb{R}^n \), a series of inputs exists such that the systems state changes from \( x_0 \) to \( x_1 \). Considering the following linear system:

\[
\dot{x} = Ax + Bu \quad \text{and} \quad y = Cx + Du
\]

where \( x \in \mathbb{R}^n, u \in \mathbb{R}^m \) and \( y \in \mathbb{R}^p \) \hspace{1cm} (2.28)

The system is controllable if there exists a \( t_0 \) and a series of actions \( u \in \mathcal{U} \) such that \( x(t_0, x_0, u) = x_1 \), where:

\[
x(t, x_0, u) = e^{At}x_0 + \int_0^t e^{A(t-s)}Bu(s)ds \quad y(t, x_0, u) = Ce^{At}x_0 + \int_0^t Ce^{A(t-s)}Bu(s)ds + Du \quad (2.29)
\]

The system is controllable if the controllability matrix has full rank, meaning it spans the full space. The controllability matrix is defined:

\[
R = [B|AB|A^2B|\ldots|A^{n-1}B]\quad (2.30)
\]

The concept equilibrium has already been introduced; it is formalized in definition 2.8. Stability is another essential concept. A practical definition is BIBO stability, as described in definition 2.9.
Estimation: fitting, quality & limitations

Chapter 2

MODELING & ESTIMATION

Observability

Observability of a process is the possibility to uniquely determine the internal state from the input-output behavior of that process given sufficient observations, while assuming that the structure $f$ and configuration $\theta$ of the process have been identified [Olsder, 1994], figure. The observability of a given process depends on the mapping $f_0$ and the available sensor information.

\begin{align*}
  \dot{x} &= f_\theta(x, u) \\
  y &= Cx + Du
\end{align*}

Definition 2.11: Observability

A process is observable if it’s internal state can be determined from it’s input-output behavior. A system is observable if there exists such that for each $u \in U$ it follows from $y(t, x_0, u) = y(t, x_1, u)$ for all $t \in [0, t_1]$ that $x_0 = x_1$. Considering the system:

\begin{align*}
  x &= Ax + Bu \\
  y &= Cx + Du
\end{align*}

Defining:

\begin{align*}
  x(t, x_0, u) &= e^{At}x_0 + \int_0^t e^{A(t-s)}Bu(s)ds \\
  y(t, x_0, u) &= Ce^{At}x_0 + \int_0^t Ce^{A(t-s)}Bu(s)ds + Du
\end{align*}

Controllability: dependency

Attach the grey dotted bar to the two black bars. With a rigid spring the triangles cannot be aligned at the dotted position due to dependency of the steering: a control problem arises.

Controllability: underdetermined

Attach the grey dotted bar to the two black bars. Using only one handle, the triangles cannot be aligned at the dotted position due to underdetermination of the steering: a control problem arises.
The system is observable if the observability matrix \( W \) has full rank \( n \), meaning it spans the full space. The controllability \( W \) matrix is defined in equation 2.34.

\[
W = \begin{bmatrix}
C \\
\vdots \\
CA \\
\vdots \\
CA^{n-1}
\end{bmatrix}
\]  

(2.34)

### 2.4.5 Complexity estimation

After deciding on a particular type model and its architecture for the data, the problem of model complexity remains. The problem has already been introduced in the context of data analysis, section 2.2.3. Complexity estimation for dynamical data is difficult, since the complexity of the functional dependencies and the temporal depth are intertwined. The temporal depth can be dissected into two features:

- delay or amount of history relevant to dependencies between the past and future;
- the rank of the state space, which is the minimal number of required state variables.

Disregarding the dynamics, the complexity of the functional relations has two aspects:

- rank of the space, i.e. the minimal number of independent variables required in a model
- the analytical complexity, i.e. the polynomial order of the function

#### Analytical complexity and dimensionality

Analytical complexity is best understood from Taylor expansions [Apostol, 1967] such as in equation 2.35. Higher-order dependencies implies more rapid changes and hence increased complexity. Higher-order dynamics are often ignored if their effective contributes in the data resides in the error-margin. Ignoring higher-order dependencies in a model is called truncation. Truncation causes a small but structural error, a modeling artifact. Modeling artifacts occur when model complexity is too high and higher-order terms appear as a bias term. Unfortunately the higher-order terms (H.O.T.) can rapidly change their impact on the manifest behavior outside the point of estimation \( a \) despite \( k! \).

\[
f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!} (x-a)^k + O(x^{n+1})
\]  

(2.35)

#### Statistical empirical complexity

The models’ complexity can be expressed by the number of free parameters \( P \) in the model. The adequate number of parameters depends primarily on two aspects: 1) the rank or dimension of the relations in the data, and 2) the amount of data points \( N \) available to estimate a number of parameters \( P \). Take the analogy of solving a set of linear equations. The number of unknowns should be the same as the number of equations, because each data point corresponds to an equation. In stochastic modeling the constraint \( N = P \) is not as tight. However the empirical risk relates in most models to the matching between: \( a \) number of parameters \( P \) and \( b \) the ratio between number of data points an degrees of freedom. The complexity estimation probl-
lem, finding the appropriate number of parameters $p$, can be simplified. In [Cherkassky et. al., 1999] model selection criteria are compared through numerical simulation. The evaluated measures use a rewrite of the empirical risk function as in equation 2.36, where $r$ is a penalization factor, $n$ the number of samples, and $d$ the degrees of freedom.

$$\text{empirical risk} = r \left( \frac{d}{n} \right) \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$  \hspace{1cm} (2.36)

The empirical risk includes bias and variance. Many attempts have been made to find the relation between number of parameters, amount of data, degrees of freedom and measured squared errors. Common penalization factors are, with $p$ the number of parameters:

$$r(p) = (1 + p)(1 - p)^{-1} \text{ Final prediction error (Akaike, 1970)}$$  \hspace{1cm} (2.37)

$$r(p, n) = 1 + \frac{\ln(n)}{2} p(1 - p)^{-1} \text{ Schwartz’ criterion (Schwartz, 1978)}$$  \hspace{1cm} (2.38)

$$r(p) = (1 - p)^{-2} \text{ Generalized cross-validation (Craven, Wahba, 1979)}$$  \hspace{1cm} (2.39)

$$r(p) = 1 + 2p \text{ Shibata’s model selector (Shibata, 1981)}$$  \hspace{1cm} (2.40)

$$r(p, n) = (1 - \sqrt{p}) = p \ln \left( p + \frac{\ln n}{2n} \right) \text{ Vapniks measure}$$  \hspace{1cm} (2.41)

Akaike’s information criterion is one of the most famous complexity estimates. The expected errors given $p$ parameters for $N$ data-points in a linear model has been derived by Akaike as:

$$\text{AIC}(p) = 2p + (N - p) \log \left( \frac{1}{N - p} \sum_{j=p+1}^{N} (e(v_j) - \bar{e})^2 \right)$$  \hspace{1cm} (2.42)

with $e(v_j)$ the error for pattern $v_j$. There is no consensus on the validity of Akaike’s criterion. The on-going research in this area shows that complexity estimation is a difficult problem.

**Information theory**

The key question is how many degrees of freedom are required to express all the patterns present in data. This is relevant for communication, encryption, model identification and pattern analysis. It relates to our problem of detection since we have to detect structural deviation in a system from the systems behavior, where structural can be considered the information in the deviation from the initial model. A quantitative measure for the amount of information, or structureddness in data is provided by information theory which utilized principles from statistical physics: mutual information for structuredness or entropy for unstructuredness. An introduction is found in appendix D.2.

**Embedding dimension and temporal depth**

In time-series analysis the key is also to reconstruct the process generating the data. The problem becomes computable for a large class of identification problems once the resolution, static polynomial depth and temporal depth are known. Much theory has been developed to determine what is required to reconstruct time-series generators, in section 2.2.2 we have discussed time-series analysis. Compared to stable or turbulent processes, chaotic processes are hard to identify. Without knowledge of the deterministic dependencies of a chaotic process it is hard to reconstruct from the data. A main contribution is the work of Takens on the embedding dimen-
tion of chaotic attractors, which is the minimal temporal depth required to identify all the dependencies generated by the process.

### 2.4.6 Fundamental limitations

Even given the proper model type (type of automaton) the state of the machine generating the data may be not observable. Considering the five different perspectives on assessing and expressing complexity discussed in the subsection 2.4.5 we find that this fundamental limitation that is expressed in most realms dealing with the modeling of processes and the data they generate. The quality of the model is not only a matter of choice for the designer but it is also limited by several other aspects related to the available data and chosen model class. The limiting factors in the modeling process are:

- **Observability and identifiability of the information source**
  Observability and identifiability of information sources is limited by: 1) available sensor information as that can prevent full observability of the state space; 2) commensurable cost-functions (conflicting requirements); 3) existence of interdependencies between the internal states; and 4) knowledge on the state variables and the dependencies within the system.

- **Physical and computational limitations**
  A fundamental physical limitation is the sampling frequency and signal to noise ratio of the data, as stated by the laws of Shannon and Nyquist. Numerical precision is a computational limitation, especially for large system models with huge differences in amplitude between signal components can yield the requirements of large dynamic range. Also non-affine loops in the evaluation and learning of models require attention for the propagation of rounding errors.

- **Statistical Limitations**
  A statistical limitation of universal most powerful (UMP) estimators is provided by the Cramer-Rao bound. This expresses the maximum accuracy of estimators that can be achieved given the amount of data and the variance of the data.

The amount of information should be sufficient and complete to (uniquely define) the optimal solution to the modeling problem. The problem is ill-posed if this is not the case. Ill-posed problems are caused by either over-determination or under-determination:

- **over-determination**
  the amount of information as contained in the data cannot be contained in the model, e.g. there may not be any solution when there are three equations and two parameters.

- **under-determination**
  the amount of information as contained in the data in comparison with the chosen architecture is insufficient to provide a unique solution to the parameter determination problem, e.g. three variables with two relations. Though a solution can be chosen that solves the equations, many other solutions are equally good.

### 2.4.7 Dealing with complexity through simplifications

Next to these fundamental limitations, modeling approaches are limited by a preference of designers to have a model that can be understood. Most designers are confronted with the dis-
crepancies between the nominal and the natural world. Pragmatic design approaches in system modeling adopt two design principles:

- **Divide-and-conquer.**
  The system is divided into smaller processes up to the point where each subsystem can be described by a simple and straightforward process model. A model is composed in a modular or hierarchical fashion such that the system behavior can be explained from the behavior of its components, and only component behavior is modeled directly from data.

- **Uniform simplicity (Occam’s Razor):**
  A model is as accurate as required for a particular use. Any higher-order dependencies are ignored when they are not necessary to describe the system in normal operating conditions. Exceptions from normal operating conditions are usually not part of the model. A pragmatic approach is required, as one cannot model the world.

- **Linearization:**
  Stability of a process in the equilibrium allows for a simplification by linearization. Linearization corresponds to a first-order Taylor expansion. Linearization of a non-linear function $f$ in certain point $x_0$ on the $x$ axis, is achieved by evaluating in $x_0$ the derivative $df/dx(x_0)$. If the equilibrium is not $x_0$, then a correction is required, assuming the line $y = ax + b$, where $a = df/dx(x_0)$ and $b = f(x_0) - ax_0$. A non-linear difference equation is linearized taking the partial derivatives of the non-linear state propagation function $f$ and the measurement function $h$. Equation 2.43 shows the linearization of the state transition matrix. The linearization in the equilibrium requires no correction of the $A$ matrix as $\Delta x_k$ approximates $0$, outside the equilibrium a correction is required.

$$A_{[i,j]} = \frac{df_{[i]}}{dx_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0)$$ such that $x_k \approx \hat{x}_k + A(x_{k-1} - \hat{x}_{k-1})$ (2.43)

### 2.5 Summary

Models can be based on the structure of a system (hypothesis-driven and whitebox) and/or from data (data-driven or black-box), but in any case there are fundamental limitations to an a priori model. Another aspect that will require our attention is that of complexity. Complexity can be expressed statistically in relation to the model required, specifically to the degrees of freedom required in a model. A well-chosen model architecture, following the hypothesized system structure from the blueprint, and the key simplifications such as linearization and divide-and-conquer selected from the system theoretical toolbox reduce the complexity, but there is a price to pay.

This chapter has introduced theoretical concepts that are basic and essential to the reasoning in this thesis. An introduction of dynamical modeling of systems and data prepares for the discussion of neural modeling in chapter 3. Time-series analysis, system model, model identification and fitting are basic techniques to the disturbance detection methods discussed in chapter 4. The problem analysis in chapter 5, and the requirements analysis in chapter 6 require the understanding provided in this chapter of the divide-and-conquer strategy and the means of linearization to simplify complex modeling problems in connection with the issues of solvability, controllability and the bias-variance problem.