Delft University of Technology

Delft Center for Systems and Control

Interactive Collaborative Information Systems

Enhanced Situation Awareness

Technical report 06-004

State Estimation under Uncertainty: A Survey

Zs. Lendek, R. Babuška and B. De Schutter

February 2006

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
URL: `http://www.dcsc.tudelft.nl`

State Estimation under Uncertainty: A Survey

Zs. Lendek, R. Babuška and B. De Schutter

February 2006

Contents

List of Abbreviations

PDF	Probability Density Function
BN	Bayesian Network
CF	Certainty Factor
KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
PF	Particle Filter
TS	Takagi-Sugeno
LMI	Linear Matrix Inequality
NN	Neural Network
RBF	Radial Basis Function
MAS	Multiagent System
ICIS	Interactive Collaborative Information Systems

Chapter 1

Introduction

1.1 Motivation

When making decisions is control systems, one often has to rely only on imperfect information is available, or uncertain premises. The information ideally needed is in general impossible or impractical to obtain. In practical situations, one has reason based on the information available.

In the following, we refer as "imperfect information" to any information different or lacking those ideally required, and based on which decisions have to be made.

Before finding possible solutions to reduce imperfection and transform the available data such that it becomes usable for decision making, it is necessary to identify possible causes of imperfection in practical situations.

First of all, no mathematical system model is perfect. A mathematical model usually describes the characteristics and relations of interest, leaving out other features, that may affect the system. This is natural, since a complete model would be too complex to be of any use in practice and even impossible to obtain. So, many aspects of a system remain unmodeled.

Second, the aspects which are modeled are necessarily approximated. In other cases, physical laws or approximations provide only a structure of the model, with the parameters to be determined.

Third, no sensor provides perfect and complete data about a system. Even more, dynamic systems may be corrupted by disturbances, which cannot be controlled, or even measured.

In certain restrictive cases, unless vital information is missing, the imperfection can be disregarded, assuming that the difference between the ideally required and available information has no influence on the decision to be made. However, in other cases, using even a slightly imprecise data can lead to disastrous consequences. In general, one has to take into account the fact that the information available is imperfect.

1.2 Imperfection

In practice, imperfection can appear due to:

- Imperfect models: no real system can be modeled up to all the details

- Incomplete data: the value of some variable is not available (measurable) at certain moments. A severe particularization of this problem is when the values of a variable is not measurable at all.
- Contradicting data: in case of redundant sensors, one may obtain different values for the same variable in the same moment. Causes for this type of imperfection include sensor imprecision or malfunction, use of multiple, different models, transmission errors, etc. We consider the data obtained partially contradicting, if the contradiction appears only casually.
- Corrupted data: the data obtained is corrupted by noise. This type of imperfection is inherently related to all physically measured data.
- Set-valued data: the data has vague properties, but the exact value is unknown.

When handling practical situations, one can observe that the imperfection present can be of different types: decision making based on uncertain premises, missing values or contradicting data. Also, the imperfection may appear due to the subjectiveness of an observer or can be present regardless an observer.

A taxonomy of imperfection is given by Smets (1991), according to whom imperfection can be divided into three main categories:

- *incompleteness* – the obtained data is not complete, e.g., the value of some variable is missing (casually or permanently). In general any model of a physical system is incomplete: only a few of its variables can be measured, the rest have to be determined on other ways.
- *imprecision* – the value of a variable is given, but not with the required precision. An important subcategory is added in this survey here: contradiction, or partial contradiction: two different values may be given/ measured, directly or indirectly, for the same variable at the same moment. While one could argue that in this case the information is erroneous, and hence it should be discarded, when dealing with numerical computations, we reduce the problem to imprecision.
- *uncertainty* – an action has to be taken based on premises for which reliability has not been clearly established.

Imperfection can also categorized as:

- *subjective* – the imperfection is present only related to an observer: a robot cannot measure the luminosity in a certain direction because it is turned to another direction.
- *objective* – the imperfection is present regardless the presence of an observer: the same robot cannot measure luminosity due to the lack of a sensor. However, one could argue, that except for quantum physics, objective imperfection does not exist. We consider objective imperfection one that cannot be reduced without altering the physical structure of the system.

The two categorizations are not excluding one another. Incompleteness and imprecision can be labelled as objective forms of imperfection, since they do not depend on the observer (Smets, 1997).

Uncertainty on the other hand is clearly subjective imperfection, since it is directly related to an observer. For instance, consider a robot that has to move to a certain point in an environment that is (partially) unknown to it, and it recognizes its "goal state" only in the moment it gets there. In this case, the robot's decisions as to in which direction to move are based inherently on uncertain premises. However, for an outside observer, for which the environment is known, there is no uncertainty present.

For the above categories, many subcategories may be defined. The interested reader is referred to (Smets, 1998b, 1999).

In the following chapters, instead of the term imperfection, the term uncertainty is used. Also, we deal only with numerical forms of uncertainty, e.g., uncertainty, for which a numeric quantification may be defined.

Several ways of handling imperfection/uncertainty are briefly overviewed in the following section.

1.3 Handling uncertainty

When dealing with imperfection an ultimate goal would be to obtain true information. However, this is impossible due to imperfections in measurements, missing information or model mismatches. We focus on reducing the imperfection, and finding data which (given the problem circumstances) can be considered reliable for using for decision.

General methods to deal with imperfect information are presented in what follows:

- Incomplete data: estimating the missing values (casually or continuously).
- (Partially) contradicting data: since it usually appears when fusing data, its cause is usually imprecision. If a fault (erroneous sensor) cannot be detected, all data should be considered imprecise, and the imprecision reduced by combination of the available data.
- Corrupted data: if some properties of the corrupting noise are known, the data can be filtered.
- Set-valued data: combining the known properties of the data with other available information.

Several theories for handling uncertainty have been developed for different types of imperfection. The best developed and most often used is probability theory, which quantifies the likelihood of an imperfect (noise or otherwise corrupted) information to be a correct one. Fuzzy logic can deal with vague information, while degrees of belief can be assigned to uncertain premises in order to quantify there (subjective) reliability.

With regard to missing or noise corrupted data, two methods are representative: filtering and state estimation.

We consider filtering as in signal processing: a version of a signal is measured that is corrupted with noise. Given the properties of the noise, the problem is to reconstruct the original signal.

Estimation is considered when only some variables of the system can be measured at a certain moment. Also, measured data do not necessarily include the data of interest, which have to be determined. Assuming that the change in the variables of interest is reflected (possibly with delay), in the measured data, we try to reconstruct the missing information.

Throughout the survey we use filtering and estimation interchangably. If not stated otherwise, we refer to estimation.

This review is organized as follows:

Chapter 2 presents a brief overview of the theories which can be used for reasoning under uncertainty: probability and related theories, fuzzy sets and possibility theory.

Chapter 3 focuses entirely on two families of probabilistic state estimation methods: Kalman filters and particle filters. These methods are described in detail, with the afferent advantages and shortcomings, and possible application areas.

Chapter 4 deals with black- and grey-box models. Several types of fuzzy and neural observers are presented, and also possible combinations of fuzzy and probabilistic filters are described.

Chapter 5 gives a possible application of the presented estimators in a multiagent framework.

Finally, Chapter 6 presents a set of conclusions and outlines possible research directions.

Chapter 2

Theories for reasoning under uncertainty

2.1 Introduction

A real-time control system has to deal with information acquired online. As discussed in Chapter 1, in general, available information is not perfect and the information ideally required is rather impossible to obtain. Models that can quantify and handle different types of uncertainties have to be developed.

Several theories used for handling uncertainty are presented in the following sections: the oldest and possibly still best developed is probability theory. Several other theories are based on or are generalizations of probability theory, such as upper and lower probabilities, certainty factors or Bayesian networks. Fuzzy logic is a theory dealing with uncertainty deriving from the vagueness of the terms. Possibility and necessity measures are also presented. together with possible combinations of fuzzy and probability theory.

2.2 Probability theory

The word "probable" was derived from the Latin probare, meaning to test, to prove. Informally, "probable" is often interchanged with "possible", "likely", "uncertain", "maybe". However, in the following, the term "probability" is used in the mathematical sense, defined below.

Probability theory attempts to quantify the notion of probable. The general idea is divided into two concepts:

- aleatory (objective) probability, which represents the likelihood of future events whose occurrence is governed by some random phenomena. Examples include tossing a coin or spinning a roulette wheel.

- epistemic (subjective) probability, which expresses one's uncertainty about the outcome of some event, in the lack of knowledge or causes. The common example is when one tries to determine how probable is that a suspect committed a crime, based on the evidence presented.

It can be argued whether the second type of probability really exists or that it can be represented by objective probability combined with incompleteness. Choosing one of the representations has implications on how one models the real world.

Probabilistic concepts are formalized so that they can be considered apart from their meaning. The formal terms are manipulated by the rules of mathematics and logic.

2.2.1 Probabilities

In the following, the Kolmogorov formalization of probability is presented. In this formulation, sets are interpreted as events and probability as a measure defined on these sets. In this way, the probability $P(\cdot)$ of an event E, denoted $P(E)$ is defined with respect to a universe or a sample space Ω of all possible elementary events so that $P(\cdot)$ satisfies the Kolmogorov axioms.

Let Ω (sample space/ universe/ frame) denote the set of all events. If A is a subset of the sample space Ω, then A contains events that ca happen. The empty set $\emptyset$ denotes impossible events. It is usually assumed that a probability is a field on the sample space, that is, a set of subsets of the sample space, which is closed under finite units and complementation on the sample space.

The Kolmogorov axioms are formalized in the following way:

Definition 2.1 *Let Σ be an algebra on Ω. Then, $P(\cdot)$ is called a probability function on Σ if $P(\cdot)$ assigns real values to the members of Σ and satisfies the following conditions:*

(i) $\forall A \in \Sigma,\ P(A) \geq 0$

(ii) $P(\Omega) = 1$

(iii) $\forall\ A,\ B \in \Sigma,\ A \cup B \neq \emptyset,\ P(A \cup B) = P(A) + P(B)$

Some useful properties of probabilities are given in what follows (Voorbraak, 1995).

Proposition 2.1 *Let Σ be an algebra on Ω and $P(\cdot)$ a probability function on Σ, and assume that A and B are elements of Σ. Then:*

(i) $P(A) = P(A \cap B) + P\left(A \cap \bar{B}\right)$

(ii) $P\left(\bar{A}\right) = 1 - P(A)$

(iii) $P(\emptyset) = 0$

(iv) $B \subseteq A \Rightarrow P(B) \leq P(A)$

(v) $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Proposition 2.2 *Assume that $A, B_1, \ldots, B_n \in \Sigma$ so that $B_1, \ldots, B_n$ is a partition of Σ. Then:*

$$P(A) = \sum_{i=1}^{n} P(A \cap B_i)$$

Proposition 2.3 *Two sets $A, B \in \Sigma$ are called independent if:*

$$P(A \cap B) = P(A) \cdot P(B)$$

2.2.2 Conditional probability and Bayes rule

Every set A with nonzero probability $P(A)$ can define another probability:

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}$$

on the sample space. This probability is called the *conditional probability* of B given A. Two sets A and B are independent if the probability of B given A is the same as the probability of B.

Some useful properties of the conditional probability are given in the following propositions.

Proposition 2.4 *Assume that* $A, B_1, \ldots, B_n \subseteq \Sigma$ *so that* $B_1, \ldots, B_n$ *is a partition of* Σ *and* $P(B_i) > 0, \forall i = 1, 2, \ldots, n$. *Then:*

$$P(A) = \sum_{i=1}^{n} P(A \cap B_i) \cdot P(B_i)$$

Proposition 2.5 *(Chain rule) Assume that* $A_1, A_2, \ldots, A_n \subseteq \Sigma$ *so that* $P(\cap_{i=1}^{n} A_i) > 0$. *Then:*

$$P(\cap_{i=1}^{n} A_i) = P(A_1) \cdot P(A_2 \mid A_1) \cdot P(A_3 \mid A_1 \cap A_2) \ldots P\left(A_n \mid \cap_{i=1}^{n-1} A_i\right)$$

For the computation of (conditional) probabilities, a simple result provided by Thomas Bayes is useful. This result, known as *Bayes rule* or *Bayes theorem* is given in what follows. Based on the definition of conditional probability we have:

$$P(A \cap B) = P(A \mid B) \cdot P(B) = P(B \mid A) \cdot P(A)$$

If $P(B) \neq 0$, then:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Theorem 2.1 *(Bayes rule) Assume* $A, B \subseteq \Sigma$ *so that* $P(A) > 0$, $P(B) > 0$. *Then*

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Each term in Bayes theorem has a conventional name:

- $P(A)$ is the *prior* or the marginal probability of A
- $P(A \mid B)$ is the *posterior* probability of A, given B
- $P(B \mid A)$ for a specified value of B is the *likelihood* function of A, given B. It can also be written as $\mathrm{L}(A|B)$
- $P(B)$ is the prior probability of B. In computations, it generally acts as a normalizing constant.

A common example of using this rule is medical diagnosis: one is interested in the probability of a disease A, given the symptoms B. Usually it is easier to obtain $P(B \mid A)$ (from experts) and the prior probabilities of the symptoms $P(B)$ and the disease $P(A)$ than directly $P(A \mid B)$.

Proposition 2.6 *Assume A, B, $C \subseteq \Sigma$ so that $P(C) > 0$. Then A and B are conditionally independent given C, if*

$$P(A \cap B \mid C) = P(A \mid C) \cdot P(B \mid C)$$

2.2.3 Probability distributions

A probability distribution is a function that assigns probabilities to events. The choice of a distribution depends on the assumptions made about the events in question. A common way to specify a probability distribution is a probability density function (PDF). In this case, the probability of an event is obtained by integrating the density function. The probability distribution function can also be specified directly, both for continuous and discrete domains.

A distribution is discrete if it is defined on a countable, discrete set; it is continuous, if it has a continuous distribution function. Common distributions used in practice are the Delta distribution, Gaussian or normal, uniform, Poisson and exponential distributions.

Bayes rule is also defined for continuous distributions.

Proposition 2.7 *Let p be a continuous probability density function. Then*

$$p(x \mid y) = \frac{p(y \mid x) \cdot p(x)}{p(y)}$$

where $p(y)$, according to the law of total probability is

$$p(y) = \int_{-\infty}^{\infty} p(y \mid x) \cdot p(x)\, dx$$

In the discrete case, $p(x, y)$ is called the joint distribution of the random variables X and Y where $p(X|y)$ is the posterior distribution of X, given $Y = y$, $p(y \mid X) = \mathrm{L}(X|y)$ is the likelihood function of X, given $Y = y$, and $p(X)$ and $p(Y)$ are marginal distributions of X and Y, respectively.

2.3 Probability-related theories

2.3.1 Generalized probability theory (upper and lower probabilities)

While probability theory uses only one measure, which gives an exact value to the probability of an event, generalized probability theory uses more general measures, which may not satisfy the additivity property from the Kolmogorov axioms.

Definition 2.2 *Let Ω be a sample space. A real-valued function F is called a capacity if it satisfies the following conditions:*

(i) $\forall A \subseteq \Omega,\ F(A) \geq 0$

(ii) $F(\Omega) = 1$

(iii) $\forall A,\, B \subseteq \Omega, A \subseteq B, \Rightarrow F(A) \leq F(B)$

In the literature, capacities are also known as fuzzy or Sugeno measures (Zadeh, 2005). A probability measure over Ω is also a capacity.

Capacity measures are motivated by constraints applied to the probabilities of the events one is interested in. Ideally, the constraints should determine a probability measure, but in general, there is a class of probability measures that satisfies the constraints.

A set of probability functions induce two natural measures, the upper and lower probability.

Definition 2.3 *Assume* Π *is a nonempty subset of the probability functions that can be defined on* Ω, $P(\Omega)$. *Then the upper and lower probabilities are defined as:*

$$\forall A \subseteq \Omega \qquad \Pi_{\text{low}}(A) = \inf\{P(A) : Pr(\cdot) \in \Pi\}$$
$$\forall A \subseteq \Omega \qquad \Pi_{\text{upp}}(A) = \sup\{P(A) : Pr(\cdot) \in \Pi\}$$

Theories derived from upper and lower probabilities include families of probability functions, and inner and outer measures (Smets, 1998b).

2.3.2 Certainty factors

The *certainty factor* model was developed for expert systems, and it is based on handling facts and rules. This model handles uncertainty by assigning a measure of (un)certainty, called a *certainty factor* to every rule and basic fact. The certainty factor of a conclusion is computed by applying some simple formulae to the certainty factors of the rules and facts used.

A certainty factor is usually a real number from the interval $[-1, 1]$. The user of the expert system has to provide the certainty factors of the basic propositions and also of the production rules. The certainty factor of a rule $A \rightarrow B$ (if A then B) is denoted by $\text{CF}[A \rightarrow B]$.

In this case, if A is evidence to support B, then $\text{CF}[A \rightarrow B] > 0$. If A is evidence against B, then $\text{CF}[A \rightarrow B] < 0$. The meaning of $\text{CF}[A \rightarrow B] = 1$ is that B is certainly true, given A (A fully supports B), and $\text{CF}[A \rightarrow B] = -1$ means that B is certainly not true, given A (A invalidates B). As specified before, in this case A is evidence against B.

The propagation rules are the following:

(i) serial combination of evidences

$$\text{CF}[B, A] = \text{CF}[A \rightarrow B] \cdot \max(0,\, \text{CF}[A])$$

(ii) conjunction

$$\text{CF}[A \cap B] = \min(\text{CF}[A],\, \text{CF}[B])$$

(iii) disjunction

$$\text{CF}[A \cup B] = \max(\text{CF}[A],\, \text{CF}[B])$$

(iv) parallel combination

$$\text{CF}[B, A_1,\, A_2, \ldots,\, A_n] = g(\text{CF}[B, A_1,\, A_2, \ldots,\, A_{n-1}],\, \text{CF}[B,\, A_n])$$

with g given as:

$$g(x, y) = \begin{cases} x + y - xy & \text{if } x,\, y > 0 \\ x + y + xy & \text{if } x,\, y < 0 \\ x + y & \text{otherwise} \end{cases}$$

2.3.3 Dempster-Shafer theory

Dempster-Shafer theory was introduced in the late 70's by Glen Shafer, as a "mathematical theory of evidence". The main innovation of the Dempster-Shafer theory is Dempster's rule, which provides means for combining the effects of different bodies of evidence.

In this formalism, the chance (probability, possibility of an event to appear) is represented as a belief function. Probability values are assigned to sets of possibilities rather than events. The Dempster-Shafer theory interprets belief functions as mass functions (Sentz and Ferson., 2003):

Definition 2.4 *A function m over a sample space Ω is a mass function or basic probability assignment, if:*

(i) $m : \mathrm{Pow}(\Omega) \rightarrow [0, 1]$

(ii) $m(\emptyset) = 0$

(iii) $\sum_{A \in \mathrm{Pow}(\Omega)} m(A) = 1$

where $\mathrm{Pow}(\Omega)$ *is the power set of* Ω.

The quantity $m(A)$ is a measure of belief that is assigned to exactly the set A. To account for the measures of beliefs assigned to the subsets of A, the belief functions are introduced.

Definition 2.5 *Let m be a mass function over a sample space Ω. The belief function* $\mathrm{Bel}(\cdot)$ *induced by m is defined by:*

$$\forall A \subseteq \Omega \qquad \mathrm{Bel}(A) = \sum_{B \subseteq A} m(B)$$

In this way, $\mathrm{Bel}(A)$ is the measure of total belief that is certainly assigned to A. There may still be beliefs that can be assigned to A, i.e., beliefs that are not assigned to propositions that falsify A. These beliefs are taken into account by plausibility functions.

Definition 2.6 *Let m be a mass function over a frame Ω. The plausibility function $Pl(\cdot)$ induced by m is defined by:*

$$\forall A \subseteq \Omega \qquad Pl(A) = \sum_{A \cap B \neq \emptyset} m(B)$$

Plausibility may be understood as a measure of belief assigned to some B-s, that are consistent with A. Beliefs/evidence may be combined by using the above measures. For this, first define the focal element of a belief function:

Definition 2.7 *A is a focal element of a belief function* $\mathrm{Bel}(\cdot)$ *over Ω if $m(A) > 0$.*

Now, the following rule can be formulated: if one piece of evidence is represented by a mass function m_1, which assigns $m_1(A)$ to A, and a second piece of evidence is represented by a mass function m_2, which assigns $m_2(B)$ to B, then the combined evidence assigns $m_1(A) \cdot m_2(B)$ to $A \cap B$.

Formally, for focal elements $A_1, A_2, \ldots, A_n$, $B_1, B_2, \ldots, B_m$, with belief functions m_1 and m_2, the combination of beliefs $Bel_1(\cdot)$ and $Bel_2(\cdot)$, is the belief function $Bel_1 \oplus Bel_2$, induced by the mass function $m_1 \oplus m_2$, defined as:

$$m_1 \oplus m_2(A) = \begin{cases} 0 & \text{if } A = \emptyset \\ \dfrac{\sum_{A_i \cap B_j = A} m_1(A_i) \cdot m_2(B_j)}{\sum_{A_i \cap B_j \neq \emptyset} m_1(A_i) \cdot m_2(B_j)} & \text{if } A \neq \emptyset \end{cases}$$

$Bel_1 \oplus Bel_2$ is also called the orthogonal sum of $Bel_1(\cdot)$ and $Bel_2(\cdot)$.

However, Shafer mentions two requirements which have to be satisfied in order to apply Dempster's rule, namely that:

1. when choosing the sample space, one has to consider not only the evidence, but also the possible interaction between the evidences.

2. the belief functions to be combined have to be based on entirely distinct bodies of evidence (Dempster-Shafer independence).

If one refers to probability theory, the second constraint may be reformulated as probabilistic independence of the sources of the two bodies of evidence.

An interpretation of the Dempster-Shafer theory is the *Transferable Belief Model* of Philippe Smets (Smets, 1998a). This model does not assume a relation between belief functions and probability theory. The information provided by an evidence is represented by the mass function and for the transfer of belief, Dempster's rule is used.

2.4 Bayesian networks

A causal network consists of a set of variables and a set of directed links between these variables. This structure is called a directed graph. In a directed graph, there are two basic relations: if there is a link from a node A to a node B, then it is said that B is a children of A or that A is a parent of B.

The variables (nodes of the graph) represent events. In general, a variable can have any number of states, but, at a certain moment, it has to be in exactly one of the states. Connections between the nodes/variables may be: serial (node A has a child B, which in turn has a child C), diverging (a node may have several children) and converging (a node may have several parents). While causal networks are graphical models, they also have a quantitative side: namely the strength attached to links between nodes.

Reasoning in this case is defined as computation of certainty factors of some nodes, given the instantiations of other nodes.

Consider the following situation: let B be a parent of A. Using probability theory, one would say that the strength of the link $B \rightarrow A$ should be defined as $P(A \mid B)$. However, C may also be the parent of A, and B and C may also have a common ancestors. In this case,the separate conditionals $P(A \mid B)$ and $P(A \mid C)$ would not give a correct evaluation of the interaction between the nodes. Just as in the case of Dempster-Shafer theory, the interactions between the nodes have to be considered.

While it is possible for a directed graph to contain cycles, no method for dealing with cyclic relations exist. In what follows, we assume that the graphs do not contain cycles.

Definition 2.8 *(Pearl, 1996) Bayesian networks (*BN*) are directed acyclic graphs in which the nodes represent variables of interest and the links represent causal influences among the*

variables. The strength of the influence is represented by conditional probabilities that are attached to each cluster of parent-child node of the network.

In this way, the chain rule is redefined for Bayesian networks.

Proposition 2.8 *(The chain rule) Let* BN *be a Bayesian network over* $U = A_1, \ldots, A_n$. *The joint probability density distribution* $P(U)$ *is the product of all conditional probabilities specified in* BN

$$P(U) = \Pi_i P(A_i \mid \mathrm{pa}(A_i))$$

where $\mathrm{pa}(A_i)$ *is the parent set of* A_i.

A conditional probability table $P(A \mid B_1, B_2, \ldots, B_n)$, which lists all the possible combinations, is attached to each variable A with parents $B_1, B_2, \ldots, B_n$. These tables are in fact the parameters of the model. The most common task for which Bayesian networks are used is probabilistic inference. Depending on which variables are unknown, the inference can be done in two directions: computing the posterior of the evidence using Bayes rule or computing the probability of an effect, given the prior probabilities. In the second case, the model is called generative. In order to simplify the computations, conditional independence relationships are defined: a node is independent of its ancestors, given its parents. In this way, the chain rule of probability may be used.

2.5 Fuzzy sets

Unlike crisp sets, which assign a value in $\{0, 1\}$ to an event, meaning that it is true or false, respectively, fuzzy sets generalize this concept, assigning a membership function to a set A:

$$\mu_A : \Omega \to [0, 1]$$

In the general case, fuzzy membership functions on a set A may be expressed as

$$\mu_A : \Omega \to L$$

where L is a partially ordered set.

Definition 2.9 *The support of a set* A *are those elements of the universe* Ω, *which have nonzero membership grade, i.e.*

$$\mathrm{supp}\, A = \{\omega \in \Omega | \mu_A(\omega) > 0\}$$

Definition 2.10 *A fuzzy set is normal, if* $\max\{\mu_A(x)\} = 1$ *and* $\exists\, \omega \in \Omega$ *s.t.* $\mu_A(x) = 1$.

Definition 2.11 *An* α*-cut of a fuzzy set is a set, that contains all elements of* Ω *with membership degree greater or equal to* α, *i.e.*

$$A_\alpha = \{\omega \in \Omega | \, \mu_A(\omega) \geq \alpha\}$$

Operations on fuzzy sets

The original theory of fuzzy sets was formulated in the terms of specific operations:

$$\begin{aligned} \mu_{\bar{A}}(\omega) &= 1 - \mu_A(\omega) \\ \mu_{A\cup B}(\omega) &= \max(\mu_A(\omega), \mu_B(\omega)) \\ \mu_{A\cap B}(\omega) &= \min(\mu_A(\omega), \mu_B(\omega)) \end{aligned}$$

However, other operators, called T-norms are often used.

Definition 2.12 *A function $T : [0,1]^2 \to [-1,1]$ is a T-norm, if:*

(i) $T(a,1) = a$

(ii) $a \leq b \Rightarrow T(a,c) \leq T(b,c)$

(iii) $T(a,b) = T(b,a)$

(iv) $T(a,T(b,c)) = T(T(a,b),c)$

The most used T-norms are:

$$\begin{aligned} T_{\min}(a,b) &= \min\{a,b\} \\ T_{\text{Luka}}(a,b) &= \max\{0, a+b-1\} \\ T_{\text{prod}}(a,b) &= ab \end{aligned}$$

Fuzzy sets serve as a compressed description of imprecise, generally contradictory pieces of information. The basic assumption is that the data is vague, due to the set-valued information.

However, this might not be always the case, hence there should be a mechanism to transform crisp (exact) values to fuzzy values and vice-versa. This procedure uses the notion of membership functions and is called fuzzification and defuzzification, respectively. Fuzzification consists of finding the corresponding membership values of a crisp value, while by defuzzification one approximates the crisp value corresponding to some fuzzy values.

2.6 Possibility theory

Possibility theory is an alternative to probability theory for dealing with uncertainty. But while probability theory assignes a single number to an event (the probability of the event to occure), possibility theory uses two numbers: possibility and necessity.

Possibility and necessity can also be seen as set measures. The possibility measure $\Pi()$ defined on a set Ω for an event/set of events A is the degree of possibility that the event A occurs.

If the possibility measure is normalized, then on a space Ω we have:

$$\Pi(\emptyset) = 0 \qquad \Pi(\Omega) = 1$$

Unlike in probability theory, the fundamental axiom of possibility theory is that:

Definition 2.13 *The disjunction of two events A and B is the maximum of their individual possibilities:*

$$\Pi(A \cup B) = \max\{\Pi(A), \Pi(B)\}$$

or, in a general:

$$\Pi(\cup_{i=1}^{n} A_i) = \sup_{i=1,2,\ldots,n} \{\Pi(A_i)\}$$

According to the above definition, in general it will not be true that

$$\Pi\left(\bar{A}\right) = 1 - \Pi(i)$$

In this way, the necessity measure is introduced.

Definition 2.14 *The necessity of an event A is the negation of the possibility of the negation of the event:*

$$N(A) = 1 - \Pi\left(\bar{A}\right)$$

The following properties hold:

Proposition 2.9 *Let A and B be two subsets of the sample space Ω. Then:*

$$\begin{aligned} N(A \cap B) &= \min\{N(A), N(B)\} \\ \Pi(A \cap B) &\leq \min\{\Pi(A), \Pi(B)\} \\ N(A \cup B) &\geq \max\{N(A), N(B)\} \end{aligned}$$

Proposition 2.10 *Let A and B be two subsets of the sample space Ω. Then A and B are disjoint, if*

$$\Pi(A \cap B) = \min\{\Pi(A), \Pi(B)\}$$

2.6.1 Conditional possibility

The role of conditional possibility and necessity is similar to that of conditional probabilities. However, they are defined using fuzzy membership functions.

Definition 2.15 *Let A, B be subsets of Ω, μ_A, $\mu_B : \Omega \to [0,1]$ membership functions. Then the conditional possibility can be written as:*

$$\Pi(A|B) = \sup_{\omega \in \Omega} \min\{\mu_A(\omega), \mu_B(\omega)\} = \sup_{\omega \in \Omega} \mu_{A \cap B}$$

and the conditional necessity as:

$$N(A|B) = \inf_{\omega \in \Omega} \min\{\mu_A(\omega), \mu_B(\omega)\} = \inf_{\omega \in \Omega} \mu_{A \cap B}$$

Some properties of conditional distributions are listed in what follows.

Proposition 2.11 *Let A, B be subsets of Ω. Then:*

(i) $N(A|B) = 1 - \Pi\left(\bar{A}|B\right)$

(ii) $\Pi(A|B) = \Pi(B|A)$

(iii) $N(A|B) = N\left(\bar{B}|\bar{A}\right)$

(iv) *if* ω *is a singleton in* Ω*, then*

$$\Pi(A|\{\omega\}) = N(A|\{\omega\}) = \mu_A(\omega)$$

(v) *if* B *is normal, then*

$$\Pi(A|B) \geq N(A|B)$$

Like probability distributions, it is also possible to define a possibility distribution:

$$\pi : \Omega \to [0,1] \qquad \pi(x) = \Pi(\{x\}), \forall x \in \Omega$$

Due to the definition of possibility, in this case one has:

$$\Pi(A) = \max_{x \in A} \pi(x)$$

Possibility theory is closely related to several other theories:

- The link between *possibility* and *fuzzy sets* is given by Zadeh's possibilistic principle, which states, that the grade of membership of an event is numerically equal to the possibility of the event.
- possibility and *probability*:
  - both can be represented by distributions
  - probability and necessity may be seen as upper and lower probabilities
  - possibility can be seen as plausibility in Dempster-Shafer theory

2.7 Fuzzy sets and probabilities

Two main approaches that combine fuzzy sets and probabilities can be distinguished based on the available literature: defining probabilities on fuzzy sets, developed based on set-valued probabilities and applying probabilistic methods to fuzzy systems.

The second approach is much more restricted, as it does not really combine the two theories, but applies both for the same problem. In this case, fuzzy sets are not used for representing uncertainty, but rather to represent a model.

For instance, in case on state estimation, instead of using a non-linear filter, one can approximate the underlying process by a TS fuzzy system with linear consequents and apply Kalman filters.

2.7.1 Combination of fuzzy and probability theory

A good example for defining probabilities on fuzzy sets is given by Kato et al. (1999), where the normal probability density function is redefined, having fuzzy numbers as mean and variance. Also, already in 1984, Zadeh defined probabilities on discrete fuzzy sets, based on their cardinality (Zadeh, 1984). The same idea is continued by Pan et al. (1996), where Bayesian inference is realized first on interval valued probabilities, and then extended to fuzzy distributions.

Furthermore, several authors defined probabilities on the rules of fuzzy systems, in order to model uncertainty, particularly in classifier systems. In this case, a fuzzy rule is expressed as follows (Meghdadi and Akbarzadeh, 2003):

$$\begin{aligned} \text{if } x_i \text{ is } A_i \text{ then} \quad & y = B_1 \text{ with probability } P_1 \\ & y = B_2 \text{ with probability } P_2 \\ & y = B_m \text{ with probability } P_m \end{aligned}$$

so that $P_1 + P_2 + \cdots + P_m = 1$.

The output is selected based on a roulette wheel mechanism. For multiple input, multiple output models, two possible forms can be defined, which depend on the statistical dependencies of the output: coupled and independent. Inference rules have been developed for both models.

2.7.2 Probabilities defined on fuzzy sets

A common method for defining probabilities on fuzzy sets is presented in what follows (van den Berg et al., 2002).

First, define the probability of a singleton fuzzy event as:

$$P(S_i) = \mu_{S_i} \dot{f}(x_i) = \sum_{x_k \in X} \mu_{S_i}(x_k) f(x_k)$$

where $f(x_k) = P(x_k)$ by definition and x_k is a fuzzy sample.

Extending the above definition to a countable set of discrete fuzzy events A_b defined on the sample space Ω, one obtains a vector of membership values: $\mu_{A_b}(x_k) = m_{A_b,k}$.

For a fuzzy partition, it can be written that:

$$\sum_{A_b} \mu_{A_b}(x_k) = 1, \qquad \forall x_k$$

In this way, several properties from probability theory can be rewritten for fuzzy sets.

- membership function of the intersection:

$$\begin{aligned} \mu_{A_b \cap B_c}(x_k) &= \mu_{A_b}(x_k) \mu_{B_c}(x_k) \\ P(A_b \cap B_c) &= \sum_{x_k \in X} \mu_{A_b}(x_k) \mu_{B_c}(x_k) f(x_k) \end{aligned}$$

- conditional probability

$$P(A_b | B_c) = \frac{\sum_{x_p} \mu_{A_b}(x_p) \mu_{B_c}(x_p)}{\sum_{x_p} \mu_{B_c}(x_p)}$$

where x_p is a finite set of representative samples.

- total law of probability:

$$\sum_{A_b} P\left(A_b|B_c\right) = 1$$

Teixeira and Zaverucha (2003) considered a hidden Markov model with fuzzy events as class and attribute values, so fuzzy probabilities are estimated. In this case, the probability of a fuzzy variable is defined as the expectation of its membership function, which in turn, is estimated from samples.

Inference in a fuzzy Bayesian network is also considered in literature (Baldwin and Di Tomaso, 2003). Since continuous variables have to be discretized, in order to provide some sort of interpolation, fuzzy partitions are used with a modified inference mechanism.

2.8 Summary and conclusions

Imperfection is present in every real system, due to the modeling, missing values or imperfect data. The common method used to compensate the imperfection is using probability theory. The notion "probable", often interchanged with "likely" or "uncertain", gives a framework in which one can quantize the imperfection. In this chapter, the Kolmogorov formalization, conditional probabilities and probability distributions were presented.

An attempt to generalize probability theory resulted in upper and lower probabilities. When introducing constraints, one might not be able to clearly define a probability. However, the constraints define a class of probability measures, for which an upper and a lower measure can be given.

Certainty factors were developed for expert systems, and they still do not have a proper theoretical justification. The uncertainty is handled by assigning a measure to each fact and rule, which quantifies one's belief that the fact is reliable.

Dempster-Shafer theory provides a different framework to combine bodies of evidence, by introducing belief functions. However, when combining evidence, one has to consider the interactions and only entirely distinct evidences can be combined. This requirement in certain cases can be nearly impossible to fulfill.

Bayesian networks are a graphical method to represent variables and interactions between them and provide a mean for reasoning. Their severe shortcoming is that they can represent only a finite number of variables and states for these variables.

Fuzzy sets, unlike probability and related theories, provide a mean to represent vagueness, instead of quantifying the likelihood of an event. These sets generalize the concept of a truth value, assigning a membership value instead to a set of events. The basic assumption is that the data is vague,due to set-valued information, and fuzzy sets can serve as a compressed description of imprecise, generally contradicting pieces of such information. Formulae for dealing with fuzzy sets are given by possibility theory.

However, fuzzy sets are also used to represent models, in which case only the basic assumptions are used. Several authors combined probabilities and fuzzy sets, the attempts ranging from using probability theory related methods on fuzzy models to defining concepts of probability theory on fuzzy sets.

All the above presented theories are suited for dealing with imperfect information. Each of them has its advantages and shortcomings: probability and related theories quantify the certainty of an event, but cannot deal with vague information. Several of these theories are

suitable only for a finite number of possibilities. A general rule for which theory should be applied cannot be given: each application has to be evaluated separately, and suitable means for handling the imperfection present has to be found.

Chapter 3

Estimation using probability theory

3.1 Introduction

As presented in Chapter 1, not all the variables of a system, and not even the variables of interest can be measured. In order to determine the variables of interest, other methods then measuring have to be used.

Consider the following description of a (possibly) time-varying, nonlinear system:

$$x_k = f_k(x_{k-1}, u_{k-1}, \theta_{k-1}, k) \tag{3.1}$$
$$y_k = h_k(x_k, u_k, \mu_k, k) \tag{3.2}$$

where:

- k - current time step
- x - state variables
- u - control variables
- θ, μ - unknown/varying parameters
- y - measurements

It is also assumed that both the states and the measurements may be corrupted by noise.

In the following, assume that the system is always observable (in control theory sense), however, (as in most of the cases), it is only partially measurable. It has to be emphasized, that in this case, (theoretically) any state can be determined (up to a delay), from a finite number of exact input-output pairs. Consider also, that the states and/or the parameters are corrupted with (some) noise and/or part of the model is unknown, but an alternative model (in the worst case a random walk) can be found.

Clearly, the goal is the estimation of the states/parameters of interest.

In the following, two family of methods will be presented for the estimation of states/ parameters: Kalman filters and particle filters. While both families rely on notions from probability theory, there is one fundamental difference between them: while Kalman filters are essentially deterministic, particle filters are stochastic methods.

While both methods assume probabilistic models, several properties are listed in what follows.

Kalman filters:

- assume Gaussian noise (except for sigma-point filters)
- approximate the posterior as Gaussian, compute its mean and covariance
- deterministic

Particle filters:

- can handle any type of noise
- represent the posterior as weighted samples
- stochastic

While presenting the two methods, parameter estimation is not handled separately. It is assumed that the parameter variation may be represented as a stochastic process and the states are simply augmented with the parameters to be estimated.

3.2 Kalman filters

3.2.1 General description

Filtering can be defined as the problem of estimating the states of a system, given its model and a set of observations. These observations are usually noisy measurements of some quantities related to the states. On the other hand, system models are usually inferred based on ideal conditions, which are not accurate. When filtering, model inaccuracy is considered as noise.

In the case of linear systems, corrupted by white Gaussian noise, the Kalman filter is proved to be an optimal filter in the mean least square sense. The Kalman filter is a recursive algorithm, that incorporates all the provided information (model and observations) and processes the available measurement to estimate the current state of the system. For this, it makes use of the system model, the measurement model and the known statistics of the corrupting noise. The filter works in two steps: prediction and update. The *prediction* step uses the system model and the information incorporated so far in order to predict the process' states. This stage is also known as the time update step, as it projects forward in time the current state and error. The *update* stage uses the latest (noisy) measurement to modify (correct) the projected state. This stage is also known as measurement update, since it incorporates the information brought by the new measurement.

While the Kalman filter is optimal only in the case of linear systems corrupted by white Gaussian noise, several extensions to nonlinear systems exist: the extended Kalman filter (based on linearizing the models around the current states), or the family of sigma point Kalman filters (based on approximating the distribution of the states).

In the following, only the recursive algorithms for the Kalman filter, Extended Kalman filter and Unscented Kalman filter will be presented. For the computational origins of the filters, see Kalman (1960), Julier and Uhlmann (1997), and Welch and Bishop (2002).

Note: In the description of the models the control input and other known/controlled parameters will not be separately stated. Thus, instead of $f(x_{k-1}, v_{k-1}, u_{k-1}, \theta_{k-1})$, where u_{k-1} is the control input and θ_{k-1} is a vector of some known/controlled parameters, we use the notation $f(x_{k-1}, v_{k-1})$.

3.2.2 Gaussian random variables

The Kalman filter usually works under the assumption that both the state transition and measurement noises are white and Gaussian. If this assumption is removed (the noises are considered arbitrary), the Kalman filter can be shown to be the best (minimum error variance) filter from the class of linear unbiased filters.

A Gaussian (or normal) distribution is usually assumed because it remains linear under linear transformations and its mean, mode and median have the same value. Also, an arbitrary distribution can be approximated as an (infinite) sum of Gaussian distributions. For any distribution for which only the mean and the variance is known, there exists a normal distribution with the same parameters.

For a normal distribution $\mathcal{N}(m, \sigma^2)$ the following properties hold:

1. Its linear transformation is also normal: if $X \sim \mathcal{N}(m, \sigma^2)$ and $a,\ b \in \mathcal{R}$, then $aX + b \sim \mathcal{N}(a \cdot m + b, (a\sigma)^2)$
2. The sum of independent normal random variables is also normal: $X \sim \mathcal{N}(m_x, \sigma_x^2)$, $Y \sim \mathcal{N}(m_y, \sigma_y^2)$ then $X + Y \sim \mathcal{N}(m_x + m_y, \sigma_x^2 + \sigma_y^2)$
3. The integral over two conditionally dependent Gaussian distribution is also Gaussian: $\int \mathcal{N}(x|Ay + b, \sigma_x^2)\mathcal{N}(y|z, \sigma_y^2)dy = \mathcal{N}(x|Az + b, \sigma_x^2 + A\sigma_y^2 A^T)$
4. The product of two Gaussian distributions has a Gaussian form (though it is not a Gaussian distribution anymore): $\mathcal{N}(m_x, \sigma_x^2)\mathcal{N}(m_y, \sigma_y^2) = z\mathcal{N}(m_z, \sigma_z^2)$

3.2.3 Linear Kalman filter

Consider the following linear model:

$$x_k = Fx_{k-1} + Bu_{k-1} + v_{k-1} \tag{3.3}$$
$$y_k = Hx_k + \eta_k \tag{3.4}$$

with x_0 (initial state) and P_0 (initial covariance) known or previously estimated, and v_{k-1} (the state transition noise) and η_k (the measurement noise) zero mean Gaussian noises of known covariances Q_k and R_k, respectively.

The objective is to recursively estimate (filter) the state x_k based on available measurements.

The next state and its covariance are constructed as follows:

Prediction: at time k, the state x_{k-1}, its covariance P_{k-1} (both estimated in the previous step) and the control input u_{k-1} are available. These are used to predict the next state and its covariance via the system model:

$$x_{k|k-1} = Fx_{k-1} + Bu_{k-1} \tag{3.5}$$
$$P_{k|k-1} = FP_{k-1}F^T + Q_k \tag{3.6}$$

Update: when the measurement y_k becomes available, it is used to compute the so-called Kalman gain and correct the estimates:

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \tag{3.7}$$
$$x_k = x_{k|k-1} + K_k(y_k - Hx_{k|k-1}) \tag{3.8}$$
$$P_k = (I - KH)P_{k|k-1} \tag{3.9}$$

3.2.4 Extended Kalman filter

As mentioned before, the Kalman filter is optimal (in the mean least square sense) only when dealing with linear processes and white Gaussian noise. However, several attempts were made to extend/adapt it for nonlinear systems. One of them, which linearizes the system equations around the current mean and covariance is referred to as the Extended Kalman filter (EKF).

Consider the following nonlinear model:

$$x_k = f(x_{k-1}, v_{k-1}) \tag{3.10}$$
$$y_k = h(x_k, \eta_k) \tag{3.11}$$

with x_0 and P_0 known or estimated, and v_{k-1} and η_k zero mean Gaussian noises of known covariances Q_k and R_k, respectively.

The extended Kalman filter linearizes the model at every step around the current estimate, assuming zero noise. Thus, we define:

$$F_k = \frac{\partial f}{\partial x}(x_{k-1}, 0) \tag{3.12}$$
$$G_k = \frac{\partial f}{\partial v}(x_{k-1}, 0) \tag{3.13}$$
$$H_k = \frac{\partial h}{\partial x}(x_k, 0) \tag{3.14}$$
$$D_k = \frac{\partial h}{\partial \eta}(x_k, 0) \tag{3.15}$$

These matrices have to be recomputed at every step. The basic (linear) Kalman filter is now applied to this linearized model.

The prediction stage becomes:

$$x_{k|k-1} = f(x_{k-1}, 0) \tag{3.16}$$
$$P_{k|k-1} = F_k P_{k-1} F_k^T + G_k Q_{k-1} G_k^T \tag{3.17}$$

The update equations are:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + D_k R_k D_k^T)^{-1} \tag{3.18}$$
$$x_k = x_{k|k-1} + K_k (y_k - h(x_{k|k-1}, 0)) \tag{3.19}$$
$$P_k = (I - K_k H_k) P_{k|k-1} \tag{3.20}$$

Though the EKF is the most used Kalman filter Maybeck (1979), especially in tracking, navigation, localization problems, in the case of an unobservable process, or when the mapping between the measurement y_k and the state is not one-to-one, the filter can diverge.

3.2.5 The Unscented Kalman filter

The fundamental flaw of the EKF is that the distributions of the random variables, after undergoing a nonlinear transformation will no longer be normal. A variation of the Kalman filter, which preserves the normal distribution through the transformation (up to a certain point) was developed by Julier and Uhlmann (1997).

The unscented filter belongs to the family of the sigma point Kalman filters, and it is based on the unscented transformation. The unscented transformation computes the statistics of a

random variable which undergoes a nonlinear transformation. For this, a number of weighted "samples" or "sigma points" are chosen deterministically, so that they completely capture the mean and covariance of a random variable.

Assume that an n_x-dimensional random variable x has to be propagated through the nonlinear function g in order to generate y:

$$y = g(x)$$

Assume also that x has a known mean x_0 and a known covariance P_x. In this case, a number of $2n_x + 1$ sigma points can be generated deterministically, so that they capture the mean and variance. A selection procedure is the following:

$$\mathcal{X}_0 = x_0 \quad w_0 = \kappa/(n_x + \kappa) \tag{3.21}$$

$$\mathcal{X}_i = x_0 + [\sqrt{(n_x + \kappa)P_x}]_i \quad w_i = 1/[2(n_x + \kappa)] \quad i = 1, 2, ...n_x \tag{3.22}$$

$$\mathcal{X}_i = x_0 - [\sqrt{(n_x + \kappa)P_x}]_i \quad w_i = 1/[2(n_x + \kappa)] \quad i = n_x + 1, ...2n_x \tag{3.23}$$

where κ is a scaling parameter, $[\sqrt{(n_x + \kappa)P_x}]_i$ is the i^{th} row of the matrix square root of $(n_x + \kappa)P_x$, and w_i is the weight associated to the i^{th} sample.

The sigma points are now propagated though the nonlinear function g: $\mathcal{Y}_i = g(\mathcal{X}_i)$, $i = 0, 1, \ldots, 2n_x$. The mean and covariance are estimated as:

$$y_0 = \sum_{i=0}^{2n_x} w_i \mathcal{Y}_i \tag{3.24}$$

$$P_y = \sum_{i=0}^{2n_x} w_i (\mathcal{Y}_i - y_0)(\mathcal{Y}_i - y_0)^T \tag{3.25}$$

These estimates are accurate to the second order of the Taylor series expansion of $g(x)$, for any nonlinear function and for any distribution. However, in certain cases the computed covariance matrix can be non-positive semidefinite, in which case the filter collapses.

In order to apply the Kalman filter using the unscented transformation, the state variables are augmented with the state transition and measurement noise, and the state covariance with the state transition and measurement covariance:

$$x_k^a = [x_k^T \; v_k^T \; \eta_k^T]^T$$
$$P^a = \text{diag}([P_k \; Q_k \; R_k])$$

The sigma points are computed based on the augmented state and covariance.

The prediction is extended, since the sigma points have to be computed and propagated through the state transition model to predict the new states. The predicted states are also have to be propagated though the measurement model in order to predict the measurement. The equations are the following:

Compute sigma points:

$$\mathcal{X}_{k-1}^a = [x_{k-1}^a \; x_{k-1}^a \pm \sqrt{(n_a + \kappa)P_{k-1}^a}] \tag{3.26}$$

where a sigma point will have the form:

$$\mathcal{X}_{k-1}^a = [\mathcal{X}_{k-1}^x \; \mathcal{X}_{k-1}^v \; \mathcal{X}_{k-1}^\eta] \tag{3.27}$$

Propagate sigma points through the state transition model:

$$\mathcal{X}_{k|k-1} = f(\mathcal{X}^x_{k-1},\ \mathcal{X}^v_{k-1}) \tag{3.28}$$

Predict next state:

$$x_{k|k-1} = \sum_{i=0}^{2n_a} w_i \mathcal{X}_{i,k|k-1} \tag{3.29}$$

Predict covariance:

$$P_{k|k-1} = \sum_{i=0}^{2n_a} w_i (\mathcal{X}_{k|k-1} - x_{k|k-1})(\mathcal{X}_{k|k-1} - x_{k|k-1})^T \tag{3.30}$$

Propagate transformed sigma points through the measurement model:

$$\mathcal{Y}_{k|k-1} = h(\mathcal{X}_{k|k-1},\ \mathcal{X}^\eta_{k-1}) \tag{3.31}$$

Predict measurement:

$$y_{k|k-1} = \sum_{i=0}^{2n_a} w_i \mathcal{Y}_{i,k|k-1} \tag{3.32}$$

Predict measurement covariance:

$$P_{yy} = \sum_{i=0}^{2n_a} w_i (\mathcal{Y}_{k|k-1} - y_{k|k-1})(\mathcal{Y}_{k|k-1} - y_{k|k-1})^T \tag{3.33}$$

Compute cross-correlation matrix:

$$P_{xy} = \sum_{i=0}^{2n_a} w_i (\mathcal{X}_{k|k-1} - x_{k|k-1})(\mathcal{Y}_{k|k-1} - y_{k|k-1})^T \tag{3.34}$$

The update stage remains the same as the Kalman filter:
Compute Kalman gain:

$$K_k = P_{xy} P_{yy}^{-1} \tag{3.35}$$

Correct predicted state:

$$x_k = x_{k|k-1} + K_k(y_k - y_{k|k-1}) \tag{3.36}$$

Correct the covariance:

$$P_k = P_{k|k-1} + K_k P_{yy} K_k^T \tag{3.37}$$

A generic Kalman filter algorithm is given in Algorithm 3.1.

The presented procedure is a general form of the unscented Kalman filter. For special cases, such as additive state transition and/or measurement noise, the computational complexity may be reduced (Julier and Uhlmann, 1997).

Algorithm 3.1 Unscented Kalman filter

Input: $u,\ y,\ Q,\ R,\ f,\ h$
Output: $x,\ P$
for $k = 1, 2, \ldots$ **do** ▷ each input
Prediction:
$x^a_{k-1} = [x^T_{k-1}\ v^T_{k-1}\ \eta^T_{k-1}]^T$ ▷ augment states
$P^a_{k-1} = \mathrm{diag}[P_{k-1}\ Q_{k-1}\ R_{k-1}]$ ▷ augment covariance
$\mathcal{X}^a_{k-1} = [x^a_{k-1} x^a_{k-1} \pm \sqrt{(n_a + \kappa) P^a_{k-1}}]$ ▷ compute sigma points

$\mathcal{X}_{k|k-1} = f(\mathcal{X}^x_{k-1},\ \mathcal{X}^v_{k-1})$ ▷ propagate sigma points
$x_{k|k-1} = \sum_{i=0}^{2n_a} w_i \mathcal{X}_{i,k|k-1}$ ▷ predicted next state
$P_{k|k-1} = \sum_{i=0}^{2n_a} w_i (\mathcal{X}_{k|k-1} - x_{k|k-1})(\mathcal{X}_{k|k-1} - x_{k|k-1})^T$ ▷ predict covariance

$\mathcal{Y}_{k|k-1} = h(\mathcal{X}_{k|k-1},\ \mathcal{X}^\eta_{k-1})$ ▷ propagate transformed sigma points
$y_{k|k-1} = \sum_{i=0}^{2n_a} w_i \mathcal{Y}_{i,k|k-1}$ ▷ predicted measurement
$P_{yy} = \sum_{i=0}^{2n_a} w_i (\mathcal{Y}_{k|k-1} - y_{k|k-1})(\mathcal{Y}_{k|k-1} - y_{k|k-1})^T$ ▷ predicted measurement covariance

$P_{xy} = \sum_{i=0}^{2n_a} w_i (\mathcal{X}_{k|k-1} - x_{k|k-1})(\mathcal{Y}_{k|k-1} - y_{k|k-1})^T$ ▷ cross-correlation matrix

Update:
$K_k = P_{xy} P^{-1}_{yy}$ ▷ Kalman gain
$x_k = x_{k|k-1} + K_k (y_k - y_{k|k-1})$ ▷ correct the state
$P_k = P_{k|k-1} + K_k P_{yy} K^T_k$ ▷ correct the covariance
end for

3.2.6 Advantages and shortcomings

In the case of a linear process model corrupted by zero-mean Gaussian noise, the Kalman filter is the optimal estimator in the least square sense. This is because of the property of the Gaussian signal of remaining Gaussian after passing through a linear system, and because of the sum of two independent Gaussian random variables is also Gaussian. However, when passing through a nonlinear function, a Gaussian distribution is not likely to remain Gaussian.

While in general the noise covariances are considered constant, in some cases, they might be variable, and have to be tuned. In an actual implementation, the measurement noise covariance is measured before filtering, but there are also cases, when the measurement noise covariance changes while filtering (e.g. due to a sensor change).

The state transition noise covariance is more difficult to determine. Sometimes even a poor model can produce acceptable results, if enough uncertainty is considered, i.e. the state transition noise covariance is great enough. In either case, the covariances may be tuned. This is usually performed off-line, using another Kalman filter.

If the covariances are constant, both the estimation error covariance and the Kalman gain will stabilize quickly and remain constant. If this is the case, they may be computed off-line and stored.

The Extended Kalman filter is based on simple linearization around the current estimate. While for highly nonlinear equations this linearization introduces considerable errors, and it might even diverge, this filter is used in many cases with success.

The EKF may collapse due to two causes:

1. the random variables are no longer normal after the nonlinear transformation and this effect is increased at every step.

2. by linearization, a nonlinear observable process may become unobservable, and the correction performed wrongly (linearization may produce highly unstable filters if local linearity is violated).

These collapses may be prevented by the UKF, which is not restricted to Gaussian noises. However, its best performance is achieved when the random variables are Gaussian. Since it uses the nonlinear equations, it is more accurate than the EKF. Also, it does not need to calculate Jacobians, but a matrix square root. Its superior performance has been reported in many publications, such as (Li et al., 2004; van der Merwe and Wan, 2003; van der Merwe et al., 2001).

While it would seem that the Unscented Kalman filter is a solution for state estimation, there are certain cases, when it cannot be used successfully. The filter may collapse due to robustness issues: the estimated posterior covariance can increase unlimited if the model is incorrect. While theoretically it can handle non-Gaussian noise, this is not always the case in practice.

3.3 Particle filters

3.3.1 General description

While Kalman filters represent the distribution of the random variables as Gaussians, this is not always the case. Moreover, the result of the propagation of a Gaussian through a nonlinear function in most cases is not a Gaussian. Also, there is no general method to compute the resulting distribution analytically. This is why the particle filters represent the distributions by samples, rather than a compact analytical form.

In the Bayesian approach of state estimation, the objective is to construct the posterior probability density function (Probability Density Function (PDF)), based on all available information (models and measurements). In general, a state estimate is required every time a measurement is received. The filter works in two stages: prediction and update. The prediction stage uses the system model to predict the state PDF one step ahead. The PDF obtained after prediction is called the *prior* PDF. The update stage uses the latest measurement to modify (correct) the predicted PDF. This correction is achieved via Bayes rule. The PDF obtained after the update (correction) is called the *posterior* PDF.

3.3.2 Preliminaries

In the following, in the description of the models, the control input and the parameters will not be explicitly noted. Thus, instead of $f(x_{k-1}, v_{k-1}, u_{k-1}, \theta_k)$ where u_{k-1} is the control input and θ_k is a vector of some known/controlled parameters, we use the notation $f(x_{k-1}, v_{k-1})$.

Likelihood function: any function proportional to a conditional probability density function, considered a function of its second argument, with its first argument held fixed.

$$L(B = b|A) = \alpha P(A|B = b)$$

Likelihood functions are evaluated up to a proportionality constant.

Initial PDF: $p(x_0)$, or $p(x_0|y_0)$ (sometimes the notation $p(x_0) = p(x_0|y_0)$ is used) is assumed to be known or previously estimated, meaning that initial samples and weights can be obtained from an (estimated) distribution. Methods for initialization include sampling from:

- $\mathcal{N}(\hat{x}_0, \sigma_0)$, where $\hat{x}_0$ is the estimated initial state and σ_0 is its estimated variance
- $\mathcal{U}(x_{0\,\min}, x_{0\,\max})$, sampling uniformly the state space

Dirac delta function/distribution: is a function $\delta(\cdot)$ with the property

$$\int_{-\infty}^{\infty} f(x)\delta(x)dx = f(0)$$

valid for any continuous function f. As a distribution, the Dirac delta has the following parameters: support $x \in [x_0; x_0]$, PDF $\delta(x - x_0)$, mean and median x_0.

Note: $p(x|x_{k-1})$ means both the prior PDF in the analytical case, and the value of the probability in the implementation of the particle filter

3.3.3 Bayesian filtering

Consider the process described by the following system and measurement models:

$$x_k = f(x_{k-1}, v_{k-1}) \tag{3.38}$$
$$y_k = h(x_k, \eta_k) \tag{3.39}$$

and x_0 is given/estimated, or, equivalently, the following PDFare given (see Rekleitis, 2004):

$$p(x_k|x_{k-1}) \tag{3.40}$$
$$p(y_k|x_k) \tag{3.41}$$

and $p(x_0)$ is given/estimated.

The objective is to recursively estimate the state x_k based on the available measurements φ_k, or, from a Bayesian point of view, to construct the posterior probability density function $p(x_k|\varphi_k)$.

Example: Consider the following system:

$$x_k = f(x_{k-1}) + v_{k-1}$$
$$y_k = h(x_k) + \eta_k$$

where v_k and η_k are zero mean Gaussian noises, with constant covariances Q and R, respectively.

This model is equivalent with:

$$p(x_k|x_{k-1}) = \mathcal{N}(x_k; f(x_{k-1}), Q)$$
$$p(y_k|x_k) = \mathcal{N}(y_k; h(x_k), R)$$

The objective is to recursively estimate $p(x_k|\varphi_k)$. □

The posterior PDF $p(x_k|\varphi_k)$ is constructed in two steps: prediction and update.

Prediction: At step k, the PDF $p(x_{k-1}|y_{k-1})$ is available. The prior PDF, $p(x_k|\varphi_{k-1})$ is constructed using the system model, via the Chapman-Kolmogorov equation (see Arulampalam et al., 2002):

$$p(x_k|\varphi_{k-1}) = \int p(x_k|x_{k-1})\, p(x_{k-1}|\varphi_{k-1}) dx_{k-1} \tag{3.42}$$

Update: As the measurement y_k becomes available, it is used to update the prior PDF via the Bayes rule. The posterior PDF can be computed as:

$$p(x_k|\varphi_k) = \frac{p(y_k|x_k)\, p(x_k|\varphi_{k-1})}{p(y_k|\varphi_{k-1})} \tag{3.43}$$

where the normalizing constant is given by

$$p(y_k|\varphi_{k-1}) = \int p(y_k|x_k)\, p(x_k|\varphi_{k-1}) dx_k \tag{3.44}$$

Example (cont'd): Assume that $p(x_0|y_0) = \mathcal{N}(x_0, \sigma_0)$ is given/estimated.

At step $k = 1$, the PDF $p(x_{1-1}|y_{1-1}) = \mathcal{N}(x_0, \sigma_0)$ is available. The prior is constructed by using (3.42):

$$p(x_1|y_0) = \int p(x_1|x_0)\, p(x_0|y_0) dx_0$$

$$p(x_1|y_0) = \int \mathcal{N}(f(x_1), Q)\, \mathcal{N}(x_0, \sigma_0) dx_0$$

When the measurement y_1 becomes available, the prior is updated via the Bayes rule:

$$p(x_1|y_1) = \frac{p(y_1|x_1)p(x_1|y_0)}{p(y_1|y_0)}$$

with the normalizing constant

$$p(y_1|y_0) = \int p(y_1|x_1)p(x_1|y_0) dx_1$$

This constant in the general case cannot be computed analytically. Special cases and approximations are presented by Doucet et al. (2000).

3.3.4 Representation

Except for a few special cases (such as linear Gaussian state space models), the posterior PDF cannot be analytically evaluated. To overcome this difficulty, particle filters represent the PDF at step k by a number N_S of random samples x_k^i with associated weights w_k^i. The estimates are computed based on these samples and their weights.

The weights are normalized, so that $\sum_{i=1}^{N_S} w_k^i = 1$, for each k. Since the estimated posterior PDF is represented by the set samples, there is no analytical expression for it. However, the approximation is conventionally denoted by:

$$p(x_k|\varphi_k) \approx \sum_{i=1}^{N_S} w_k^i \delta(x_k - x_k^i) \tag{3.45}$$

3.3.5 Importance sampling

Ideally, the samples should be drawn from the posterior distribution. Since this is the PDF to be estimated, in general it is impossible to sample directly from this distribution, but it can be evaluated (up to a normalizing constant). Therefore, the samples are drawn from another distribution, having a support which includes the support of the posterior PDFs. The density function from which the samples are drawn is called *importance (proposal) density function*. The procedure of sampling from the importance density instead of the posterior is called *importance sampling*.

In general, a density function $p(\cdot)$, which cannot be sampled, but can be evaluated, can be represented by the weighted samples of an importance density function $q(\cdot)$, denoted by:

$$p(x) \approx \sum_{i=1}^{N_S} w^i \delta(x - x^i) \tag{3.46}$$

where the weights w^i are calculated by

$$w^i = \frac{p(x^i)}{q(x^i)} \tag{3.47}$$

and normalized.

At step k, $p(x_{k-1}|\varphi_{k-1})$ is available as a set of samples x^i_{k-1} and corresponding weights w^i_{k-1}, $i = 1, ..., N_S$. New samples x^i_k and weights w^i_k, $i = 1, ..., N_S$ have to be generated, which should approximate $p(x_k|\varphi_k)$. To achieve this, the samples x^i_k are drawn from a (chosen) importance density $q(x^i_k|x^i_{k-1}, \varphi_k)$, and the weights are updated as

$$w^{i'}_k = w^i_{k-1} \frac{p(y_k|x^i_k)\, p(x^i_k|x^i_{k-1})}{q(x^i_k|x^i_{k-1}, \varphi_k)} \tag{3.48}$$

The weights are then normalized

$$w^i_k = \frac{w^{i'}_k}{\sum_{j=1}^{N_S} w^{j'}_k} \tag{3.49}$$

The posterior PDF is represented by the set of weighted samples, denoted by:

$$p(x_k|\varphi_k) \approx \sum_{i=1}^{N_S} w^i_k \delta(x_k - x^i_k) \tag{3.50}$$

Remarks:

- Theoretically, there is no restriction on how to select the importance density, except that its support should include the posterior's. Several importance functions (obtained by local linearization, the optimal importance function, the prior, and fixed importance functions) are described by Doucet et al. (2000).
- In the case of Bayesian filtering the filter's convergence is defined as the convergence of the estimated PDF to the real posterior. This clearly depends on the number of particles. For bounds on errors depending on the number of particles, see (Boers, 1999). Statistical methods for verifying convergence are presented by Brooks and Gelman (1998).

3.3.6 Degeneracy

A common problem of the particle filters is the degeneracy of the samples: after a few iterations, all but one particle will have negligible weight. A measure of the degeneracy of the samples is the effective sample size, approximated by:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N_{\text{S}}} (w_k^i)^2} \tag{3.51}$$

Since after normalization, $0 \leq w_k^i \leq 1$ for every particle and at every step k, $0 \leq N_{\text{eff}} \leq N_{\text{S}}$.

A small N_{eff} indicates a severe degeneracy, meaning that many of the particles will have very small weights. These particles should be eliminated, or replaced by particles with greater weights. Thus, a new sample set has to be generated. The procedure, by which a new set of samples is generated from $p(x_k|\varphi_k)$ is called *resampling*. Several resampling algorithms are given by Fearnhead (1998).

One of the resampling procedures is presented in Algorithm 3.2. This algorithm replaces the particles with smaller weights by multiplying the ones with greater weights and then resetting all the weights to $1/N_{\text{S}}$.

Algorithm 3.2 Resampling

Input: x, w
Output: x_{new}, w_{new}
  for $i = 1, 2, \ldots, N_{\text{S}}$ **do**
    Compute cumulative sum of the weights: $w_c^i = \sum_{j=1}^{i} w_k^j$
  end for
  Sample u_1 from $\mathcal{U}[0, 1/N_{\text{S}}]$
  for $i = 1, 2, \ldots, N_{\text{S}}$ **do**
    Find x_k^{+i}, the first sample for which $w_c^i \geq u_i$.
    Replace particle i: $x_{k,\text{new}}^i = x_k^{+i}$, $w_{k,\text{new}}^i = 1/N_{\text{S}}$
    $u_{i+1} = u_i + 1/N_{\text{S}}$
  end for

3.3.7 The generic particle filter algorithm

For the algorithm it is assumed that the initial samples are generated from a normal distribution, with mean x_0 (the (estimated) initial state) and a given variance σ_0. The importance density function is denoted by $q(\cdot)$, and it is assumed that it depends only on the previous state, and not on the measurement, $q(x_k|x_{k-1}, y_k) = q(x_k|x_{k-1})$. A generic particle filter algorithm is given in Algorithm 3.3.

3.3.8 Sequential Importance Resampling (SIR) particle filter

The importance density should be chosen so that it maximizes N_{eff}, which is equivalent to minimizing the variance of the weights. It has been shown that the optimal importance density is (Doucet et al., 2000):

$$q(x_k|x_{k-1}, \varphi_k) = p(x_k|x_{k-1}, \varphi_k) \tag{3.52}$$

Algorithm 3.3 Particle filter

Input: N_S, $p(x_k|x_{k-1})$, $p(y_k|x_k)$, x_0, σ_0, $q(\cdot)$, N_T (resampling threshold)
Initialize:
  for $i = 1, 2, \ldots, N_\mathrm{S}$ **do**
    Draw a new particle: $x_1^i \sim \mathcal{N}(x_0, \sigma_0)$
    Assign weight: $w_1^i = 1/N_\mathrm{S}$
  end for
At every step $k = 2, 3, \ldots$
  for $i = 1, 2, \ldots, N_\mathrm{S}$ **do**
    Predict: Draw particle from proposal distribution: $x_k^i \sim q(x_k^i|x_{k-1}^i)$
    if y_k is available **then**
      Update: Compute weight: $w_k^{i'} = w_{k-1}^i \frac{p(y_k|x_k^i)\, p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i)}$
    end if
  end for
  Normalize weights: $w_k^i = \frac{w_k^{i'}}{\sum_{i=1}^{N_\mathrm{S}} w_k^{i'}}$
  Compute effective sample size: $N_\mathrm{eff} = \frac{1}{\sum_{i=1}^{N_\mathrm{S}} (w_k^i)^2}$
  if $N_\mathrm{eff} < N_\mathrm{T}$ **then**
    Resample using Algorithm 3.2.
  end if

Analytic evaluation of this expression is possible only in the case of models, for which the system dynamics is nonlinear, the measurement model is linear, and the noise is Gaussian (see Doucet et al., 2000).

It is often convenient to choose as importance density the prior:

$$q(x_k|x_{k-1}, \varphi_k) = p(x_k|x_{k-1}) \tag{3.53}$$

In this case sampling from the prior is necessary. Samples can be generated by first generating a process noise sample v_{k-1}^i and then setting $x_k^i = f(x_{k-1}^i, v_{k-1}^i)$.

For this choice of the importance density, the weight update becomes:

$$w_k^i = w_{k-1}^i \cdot p(y_k|x_k^i) \tag{3.54}$$

Another difference between the SIR filter and the generic particle filter is that for the SIR filter resampling is applied at every step. In this way, after the weights are normalized, resampling is applied, and the weights will have the value $1/N_\mathrm{S}$. The weight update equation (3.54) can be written as:

$$w_k^i = p(y_k|x_k^i) \tag{3.55}$$

In the following, it is assumed that the state transition noise distribution is denoted by $\mathcal{D}_\mathrm{N}(\cdot)$, and the initialization is done in the same manner as for a generic particle filter. In these circumstances, the SIR filter algorithm is given in Algorithm 3.4.

Though the SIR filter has the advantage that the weights are easily evaluated and the importance density can be easily sampled, this filter can be inefficient and sensitive to outliers, since the state-space is explored without any knowledge of the observation.

Algorithm 3.4 SIR filter

Input: N_S, $f(x_{k-1}, v_{k-1})$, $p(y_k|x_k)$, x_0, σ_0, $\mathcal{D}_N(\cdot)$, N_T (resampling threshold)
Initialize: same as for the generic particle filter
At every step $k = 2, 3, \ldots$
  for $i = 1, 2, \ldots, N_S$ **do**
    Draw a process noise sample: $v_{k-1}^i \sim \mathcal{D}_N(\cdot)$
    Predict: Compute corresponding particle: $x_k^i = f(x_{k-1}^i, v_{k-1})$
    if y_k is available **then**
      Update: Compute weight: $w_k^{i'} = p(y_k|x_k^i)$
    end if
  end for
  Normalize weights: $w_k^i = \frac{w_k^{i'}}{\sum_{i=1}^{N_S} w_k^{i'}}$
  Resample using Algorithm 3.2.

3.3.9 Advantages and shortcomings

Particle filters have one main advantage over Kalman filters: the noise corrupting the system, whose states have to be estimated does not have to be Gaussian.

However, while for at least the linear Kalman filter, stability is proven, this is not the case for several types of particle filters. Also, obtaining the posterior distribution does not help when an exact estimate of the state is needed. Several problems raise due to the approximation of the posterior with weighted samples:

- even if the samples are drawn from the real posterior, an infinite number of samples are necessary to reconstruct it.
- the mean of the posterior can not be considered a correct estimate of the state, since no guaranties exist that the posterior is unimodal.
- in practice, a large number of samples are necessary for a good estimation. Hence, the algorithm is time-consuming, and not suitable for fast processes. In order to reduce the time needed, the number of samples has to be decreased, which may render the filter unstable. However, increasing the number of particles beyond a certain threshold does not significantly improve the results.
- in order to prevent degeneracy, the importance density has to cover the possible states, which may not be determined beforehand.
- in practice, the filter may collapse due to numerical errors, and re-initialization can be complicated

While in theory particle filters are the best of the above estimators for non-linear systems, it can not be considered a universal solution. In case of small nonlinearities or fast processes may be handled better by a Kalman filter, even if the estimate is not perfect.

3.4 Summary and conclusions

In the probabilistic framework, two classes of filtering methods have been presented: Kalman and particle filters. Both classes consider a probabilistic representation of a dynamic model.

While not considered separately, both the Kalman and particle filters can be used for state and parameter estimation.

The best known variant of the Kalman filter is the linear Kalman filter, which is an optimal estimator in the least square sense. However, it can be used only for linear systems, and when the model is completely known, up to a probabilistic uncertainty. Its most used extension to nonlinear systems, the Extended Kalman filter, is based on the linearization of the system at each moment around the current mean and variance. However, in highly nonlinear systems, when the linearization is valid only in a very restricted space, this filter is likely to diverge.

Another extension of the Kalman filter is the Unscented Kalman filter, which uses the true nonlinear model. This method is based on representing the distribution of the random variables by a few, deterministically chosen points (the so-called sigma points), which are chosen so that they completely capture the mean and variance of the random variables. While this method cannot be rendered divergent due to the model nonlinearity, it may still diverge, if the mean and the variance are not sufficient for representing the distribution (e.g. multimodal distributions). However, when this is not the case, its performance is superior to that of the Extended Kalman filter.

Particle filters represent distributions by random samples and associated weights. Their main advantage is that they can handle an type of nonlinear models and any distributions. However, there is a severe shortcoming: their implementation is time-consuming in such a way, that it can render them unusable in real-time estimations.

There is no general rule which estimator is better for nonlinear systems. In principle, particle filters are the most accurate. However, small nonlinearities may be better handled by Extended Kalman filters, or, assuming the the random variables are unimodal, the Unscented Kalman filter.

Chapter 4

Black and grey box models

4.1 Introduction

In this chapter, some approaches for analysis and design of observers for Takagi-Sugeno fuzzy systems and systems represented by neural networks are presented.

The motivation for representing dynamical systems by black or grey box models is straightforward: in certain cases, no prior information is available of the dynamical system that has to be studied. In this case, one could assume a random walk model, however, this model does not give any useful insight on the dynamics, and no predictions as how the system would evolve can be made. If using grey box modelling, one can incorporate the known dynamics.

Also, in certain cases, the structure of a system can (partially) change. Instead of identifying an unknown, nonlinear system, it is more convenient to assume from start an adaptive structure, and modify it in concordance with the plant to be modelled. Approximating a process by an adaptive fuzzy or neural model make this adaptation possible.

In what follows, fuzzy and neural models and the corresponding observers will be presented. While several types of observers have been developed for Takagi-Sugeno type fuzzy systems, these deal with models having stationary structures. The observers for neural modelled systems are more restricted, considering the general structure of the systems modelled, however, some of these observers can deal with changing nonlinearities.

4.2 Deterministic fuzzy observers

4.2.1 Takagi-Sugeno models and observers

The TS fuzzy model is composed of a if-then rule base that partitions a space into fuzzy regions called antecedents. The consequent of each rule is a simple functional expression. A rule can be described as follows:

If θ_1 is F_1^i and ... and θ_m is F_m^i then $y_i = f_i(x)$

where the vector θ stand for the premise variables and may be a subset of the states.

For a general continuous dynamic system an approximative fuzzy system can be written as in what follows. Consider the dynamical system given as:

$$\begin{aligned} \dot{x} &= f(x, u, \mu) \\ y &= g(x, u, \mu) \end{aligned}$$

where f and g are nonlinear, possibly time-varying functions and μ is a vector of parameters. The corresponding fuzzy system can be written as a set of fuzzy rules of the following form:

If θ_1 is F_1^i and ... and θ_m is F_m^i then

$$\begin{aligned} \dot{x} &= \hat{f}_i(x, u, \mu) \\ y &= \hat{g}_i(x, u, \mu) \end{aligned}$$

or

$$\begin{aligned} \dot{x} &= \sum_{i=1}^{m} w_i(\theta)\hat{f}_i(x, u, \mu) \\ y &= \sum_{i=1}^{m} w_i(\theta)\hat{g}_i(x, u, \mu) \end{aligned}$$

where the consequents ($\hat{f}_i$ and $\hat{g}_i$) are of lower order than the initial nonlinear functions (f and g).

In the general case, $\hat{f}_i(x, u, \mu)$ and $\hat{g}_i(x, u, \mu)$ can be non-linear, time varying functions. However, most of the fuzzy observers were developed for linear or affine consequents.

Consider the fuzzy system described as

$$\begin{aligned} \dot{x} &= \sum_{i=1}^{r} w_i(\theta)(A_i x + B_i u) \\ y &= \sum_{i=1}^{r} w_i(\theta)C_i x \end{aligned}$$

when the consequents is linear, or as

$$\begin{aligned} \dot{x} &= \sum_{i=1}^{r} w_i(\theta)(A_i x + B_i u + a_i) \\ y &= \sum_{i=1}^{r} w_i(\theta)(C_i x + c_i) \end{aligned}$$

with affine consequent. It is assumed, that the system is locally observable, i.e. the subsystems are observable.

4.2.2 Fuzzy Thau-Luenberger observers

A fuzzy Thau-Luenberger observer for this system can be written in the following way (Bergsten, 2001):

$$\begin{aligned} \dot{\hat{x}} &= \sum_{i=1}^{r} w_i(\theta)(A_i \hat{x} + B_i u + L_i(y - \hat{y})) \\ \hat{y} &= \sum_{i=1}^{r} w_i(\theta)C_i \hat{x} \end{aligned}$$

or (for the affine system)

$$\begin{aligned}\dot{\hat{x}} &= \sum_{i=1}^{r} w_i(\theta)(A_i\hat{x} + B_iu + a_i + L_i(y - \hat{y})) \\ \hat{y} &= \sum_{i=1}^{r} w_i(\theta)(C_i\hat{x} + c_i)\end{aligned}$$

Two cases are distinguished:

1. If the scheduling vector does not depend on some of the unmeasurable states, then the system is asymptotically stable, if $\exists P = P^T > 0$, so that

$$\begin{aligned}(A_i - L_iC_i)^TP - P(A_i - L_iC_i) < 0 \\ (G_{i,j} + G_{j,i})^TP - P(G_{i,j} + G_{j,i}) < 0\end{aligned}$$

   where $G_{i,j} = A_i - L_iC_i \ \ \forall i, j$.

2. The second case is when the observer's scheduling vector does depend on states to be estimated. For the affine case, the observer becomes:

$$\begin{aligned}\dot{\hat{x}} &= \sum_{i=1}^{r} w_i(\hat{\theta})(A_i\hat{x} + B_iu + a_i + L_i(y - \hat{y})) \\ \hat{y} &= \sum_{i=1}^{r} w_i(\hat{\theta})(C_i\hat{x} + c_i)\end{aligned}$$

   Clearly, there will be a time-varying difference between the estimated and the true states. In order for the observer to be stable, this difference has to go to zero. The stability conditions for the first case are rewritten as: the system is asymptotically stable, if $\exists P = P^T,\ Q = Q^T,\ \mu > 0$ so that

$$\begin{aligned}P, Q &> 0 \\ (A_i - L_iC)^TP - P(A_i - L_iC) &< 0 \\ \begin{pmatrix} Q - \mu^2 & P \\ P & I \end{pmatrix} &> 0 \\ \|\Delta(\theta, \hat{\theta}, x, u)\| &\leq \mu\|e\|\end{aligned}$$

   i.e., the modelling error also has to be bounded.

   This case is developed only for constant measurement equations, i.e. $C_i = C_j = C\ \forall i, j$.

A common method to design stable fuzzy observers is by reducing the problem to LMIs. Bergsten and Palm (2000) presents a LMI based design algorithm for achieving the best robustness measure. The drawback of this method is that the observer's poles may be badly positioned, but this can be corrected by constraining the LMI regions for the solution.

4.2.3 Fuzzy sliding mode observers

Beside Thau-Luenberger observers, also sliding mode observers are developed (Bergsten et al., 2002, 2001).

For this case, the fuzzy approximation of the system is written as:

$$\begin{aligned} \dot{x} &= \sum_{i=1}^{r} w_i(\theta)(A_i x + B_i u + a_i) + D\zeta \\ y &= Cx \end{aligned}$$

It is assumed that a linear change of coordinates $z = Tx$ exists, so that $\bar{D} = TD = [0\ \bar{D}_2]$, $\bar{C} = CT^{-1} = [0\ I_p]$, $\bar{A}_i = TA_iT^{-1}$, $\bar{B}_i = TB_i$, $\bar{a}_i = Ta_i$, and the uncertainty/modeling error is bounded , i.e. $\exists \eta > 0$ s.t. $\|\zeta\| \leq \eta$.

Then the observer for the transformed system can be written as:

$$\begin{aligned} \dot{\hat{z}} &= \sum_{i=1}^{r} w_i(\hat{\theta})(\bar{A}_i \hat{z} + \bar{B}_i u + \bar{a}_i + \bar{L}_i(y - \hat{y})) + \bar{D}\zeta \\ \hat{y} &= \bar{C}\hat{z} \end{aligned}$$

The stability conditions are similar to those for the fuzzy Thau-Luenberger observer.

The same idea can be used for developing sliding mode observers for linear dominant systems in (Palm and Bergsten, 2000). In this case, the system is essentially linear with modeling uncertainties and further uncertainties, of which upper bound is known.

If the j^{th} subsystem is dominant, then the system can be written as:

$$\dot{x} = A_j x + B_j u + a_i + \Delta_j + f_u m$$

where Δ_j is the known "uncertainty" (the difference between the subsystems).

Using the same transformation as for the sliding mode observer, the transformed system is:

$$\begin{aligned} \dot{z} &= A_{11j} z + A_{12j} y + B_{1j} u + f_{z,j} \\ \dot{y} &= A_{21j} z + A_{22j} y + B_{2j} u + f_{y,j} + f_{z,j} \end{aligned}$$

The observer will have the form:

$$\begin{aligned} \dot{\hat{z}} &= A_{11j} \hat{z} + A_{12j} \hat{y} + B_{1j} u + \hat{f}_{z,j} + A_{12j} C y \\ \dot{\hat{y}} &= A_{21j} \hat{z} + A_{22j} \hat{y} + B_{2j} u + \hat{f}_{y,j} + (A_{22j} - A_{s,j})(y - \hat{y}) - M \end{aligned}$$

where $A_{s,j}$ is a stable design matrix.

The observer for the original system becomes:

$$\begin{aligned} \dot{\hat{x}} &= \sum_{i=1}^{r} w_i(\hat{\theta})(A_i \hat{x} + B_i u + a_i) + G_L(y - \hat{y})) + G_{nl} M \\ \hat{y} &= C\hat{x} \end{aligned}$$

where $G_l = T_j^{-1} \begin{pmatrix} A_{12j} \\ A_{22j} - A_{s,j} \end{pmatrix}$ is the linear gain matrix and $G_{nl} = T_j^{-1}[0\ 1]^T$ is the nonlinear gain matrix.

Similar stability criteria are presented in (Wang et al., 1996; Tanaka and Sano, 1994; Tanaka and Wang, 1997).

Several authors consider the case of the observer and linear state-feedback controller together and develop relaxed stability conditions for the augmented system. The conditions usually lead to (generalized) eigenvalue problems, possible to be solved using LMIs (Taniguchi et al., 1999b; Tanaka et al., 1998a; Taniguchi et al., 1999a). For the case when the weights depend on the estimated states, the observer and the controller cannot be designed separately (Tanaka and Sano, 1994).

Fuzzy observers are usually employed together with a PDC controller. A general framework for PDC controllers is given by Wang et al. (2000), while a systematic procedure for fuzzy model construction, rule reduction and robust compensation is presented by Taniguchi et al. (2001).

Applications include state estimation for translational oscillations (Tanaka et al., 1998c), backing of a mobile robot with multiple trailers (Tanaka et al., 1998b) and visual servoing (Kadmiry and Bergsten, 2004).

4.3 Fuzzy Kalman filters

Several methods called fuzzy Kalman filters can be found in the literature. Commonly, these methods use the linear Kalman filter for estimating the states of a system approximated by a TS fuzzy model with linear consequents, combine the two approaches to enhance the performance of the estimator or estimate parameters that otherwise should be know in advance, on-line.

4.3.1 State estimators

When representing a process by a fuzzy model, a rule itself represents a linear system. In this case, for a single rule, the Kalman filter is an optimal estimator. An overview of Kalman filtering for fuzzy discrete time dynamic systems can be found in (Simon, 2003). However, these methods are applicable only for the noise-free case, i.e. the states and measurements are not corrupted by probabilistic noise beyond the possible model mismatch. Also, in most of the cases, it is assumed that the premise variables are independent of the states to be estimated. The main difference between the methods is the moment of linearization. McGinnity and Irwin (1996) present four types of fuzzy local linear models, which can be combined with Kalman filters.

Consider the following general nonlinear model:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned}$$

The four types of fuzzy local linear models are the following:

(i) approximating the system with a fuzzy model and using a single Kalman filter at each step, with a global covariance and Kalman gain. The fuzzy model is determined at each instant:

$$\begin{aligned} \dot{x}_k &= \hat{f}_{i,k}(x,u) \\ y_k &= \hat{g}_{i,k}(x,u), \qquad i = 1,..n \end{aligned}$$

(ii) developing local models, for which a bank of Kalman filters is used, i.e. for each local linear model a separate Kalman filter is used. Each local linear model consists of a state transition and measurement function. The estimated state is computed as the weighted mean of the estimate of every local model. The fuzzy model is determined in advance and it is used unchanged throughout the estimation:

$$\begin{aligned} \dot{x} &= \hat{f}_i(x,u) \\ y &= \hat{g}_i(x,u), \qquad i = 1,..n \end{aligned}$$

(iii) develop local models separately for the state transition and measurement model, and uses the same Kalman filter. In this case, the fuzzy system becomes:

$$\begin{aligned} \dot{x} &= \hat{f}_j(x,u) \qquad j = 1,..m \\ y &= \hat{g}_i(x,u), \qquad i = 1,..n \end{aligned}$$

(iv) global linearization at every step, in fact an extended Kalman filter

Applications include target tracking (McGinnity and Irwin, 1996, 1997), sensor fusion (Zhang and Wei, 2003; Sasiadek and Wang, 1999), model set adaptation (Ding et al., 2001).

4.3.2 Covariance estimators

Several methods combine fuzzy and probabilistic observers in order to enhance their performance, or in order to tune one of the filters. A common example is when the covariance of the corrupting noise is unknown.

Typically, the measurement noise covariance is derived from the properties of the sensors used, but the state transition noise covariance has to be estimated. First, a particular form is assumed, such as the identity matrix multiplied by a constant. Now, the estimation problem is reduced to one variable. One way to estimate it is to apply another Kalman filter. However, fuzzy rules can also be applied. These are based on the difference between the measurement and its predicted value, since, if the filter works correctly, the residual should be a zero-mean Gaussian. Typical rules (Jetto et al., 1999; Aja-Fernandez et al., 2003) can be described as:

if residual is OK then	Q is unchanged
if residual is very near to zero	Q is reduced
if residual is very far from zero	Q is increased

Another covariance estimation case is when the state transition noise is known/previously estimated, but the sensors are changed during the process. Since different sensors may induce different measurement covariances, the new covariance has to be estimated. This type of tuning can be performed by fuzzy logic with rules based on the difference between the theoretical and computed values of the covariance. A predefined difference is added/substracted (Loebis et al., 2004) to/from an initial covariance.

The methods which aim to enhance the performances of the filter are usually apply both filters, with correction gains computed based on the other filter's residual (Li, 1993; Sasiadek and Wang, 1999; Sasiadek and Khe, 2001; Zhang and Wei, 2003).

Fuzzy Kalman filters can also be used for model selection/mixing, in general in applications such as tracking a maneuvering target. When using particle filters, the maneuver detection

is usually reduced to the estimation of a (possibly discrete) variable (Yu and Cheng, 2005), or to computing the likelihood of different models, which depend on a parameter (Angelova et al., 2001). When using fuzzy Kalman filters, the mixing of the models is realized. In case of multiple models, filter banks may also be used, however, fuzzy rules may be applied to adapt a model set (for better covering of possible maneuvers) (Ding et al., 2001), to adjust parameters of the models, or even to reinitialize the model if the track is lost (Simutis et al., 1992). Also, validity domains of the sensors may be specified by fuzzy sets (Caron et al., 2004).

On the other hand, Kalman filters may be used to train fuzzy filters Simon (2002). Training of a fuzzy system can be understood as optimization or estimation of parameters, and this can be done by a Kalman filter. The author describes a possible application, when the input and output membership functions are symmetric triangles, and an extended Kalman filter is used to estimate their centroid and half-widths.

4.4 Neural Network observers

While several applications employ so-called "neural observers", these are usually reduced to trying to represent the underlying partially known/uncertain system by the network. Examples of application for these neural observers include representing the partial distribution of wind (Oztopal, 2005), training a network to approximate a complex relation (Garcia and Shigidi, 2005), or to estimate the tipover stability margins of a mobile robot (Meghdari et al., 2005). All these applications use in general feed-forward networks, trained with backpropagation to represent the relations or to store results of equations which would otherwise require heavy computations.

However, most of the neural network used together with observers represent only a part of the model, that is unknown, uncertain or simply too complex. Commonly, it is assumed that the model can be written as:

$$\begin{aligned} \dot{x} &= Ax + N_{\mathrm{l}}(x, u, \theta) \\ y &= h(x, u) \end{aligned}$$

where Ax is a known linear part and N_{l} the uncertain/unknown part of the model. The uncertainty is also considered in most of the cases bounded.

he most simple form of the system is the following (Wang et al., 2002).

$$\begin{aligned} \dot{x} &= Ax + b[f(x) + g(x)u + d(t)] \\ y &= C^T x \end{aligned}$$

where f,g are unknown, but bounded functions, the upper bound of the disturbance $d(t)$ is known, and the form of A, B and C is given:

$$\mathrm{A}= \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ . & . & . & . & . \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \mathrm{B}=[0 \quad 0 \quad \dots \quad 1], \mathrm{C}=[1 \quad 0 \quad \dots \quad 0]^T.$$

For this system a state feedback controller can be computed, if the states are completely known, hence, an observer is needed. A common solution is to train a fuzzy model or a neural network to approximate the nonlinear part, then adapt it based on the observer. Several authors use an RBF network representation of a fuzzy system, with feedback and observer

gains predefined and known membership functions. The observer for this system has the following form:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + b[\hat{f}(\hat{x}) + \hat{g}(\hat{x})u - v(t)] + K(y - C^T\hat{x}) \\ \hat{y} &= C^T\hat{x}\end{aligned}$$

where $(A - KC^T)$ is strictly Hurwitz.

While above it is considered that at least the linear part is known, the solution may be extended to the case when both the linear and nonlinear parts are unknown (Schroder et al., 2001). However, in this case, a static nonlinearity is considered. Then the model can be expressed as:

$$\begin{aligned}\dot{x} &= Ax + Bu + N_1(u) \\ y &= C^T x + du\end{aligned}$$

and the nonlinearity:

$$N_1(u) = e_{N_1}\bar{N}_1(u)$$

where $\bar{N}_1$ is a scalar nonlinear function and e_{N_1} a known coupling vector.

If the linear part and e_{N_1} is known, the nonlinearity is visible, and an adaptive RBF network can be used to approximate the nonlinear function. The observer is given as:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + e_{N_1}\hat{N}_1(x, u) \\ \hat{y} &= C^T\hat{x} + du\end{aligned}$$

The adaptation laws are commonly error based. The interested reader is referred to (Baz, 1992; Kim et al., 1997; Elanayar and Yung, 2000). Permanent learning of a time-variant nonlinearity is also possible if the network learns faster that the nonlinearity changes.

If both the linear and nonlinear parts are unknown, two recurrent neural networks can be employed: one for the nonlinear part (as presented above) and one for the linear part.

In the same way as for fuzzy systems, sliding-mode observers can also be developed for the above presented linear dominant models. The observer employed estimates the states, and, based on the estimated state, a neural network model approximates the unknown part of the system.

Applications, when a linear part is known and the nonlinear part is represented by a neural network include control of a servomechanism (Lin and Wai, 2002a), self-tuning control of an induction motor drive (Sheu and Chen, 1999), robust control of an induction motor drive (Lin and Wai, 2002b; Lin et al., 2000), or linear quadratic state feedback (Marino et al., 1999).

All the above presented papers try to incorporate into the model a known part, mostly, as a linear part of the system, which may be previously identified. Other approaches represent the underlying system as a black-box and construct linear observers. Consider a general nonlinear system, represented by the equations:

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= h(x, u)\end{aligned}$$

Assuming that the state equation has a unique solution, the plant can be replaced by two linearly parametrized neural networks, and the problem reduces to weight adaptation, using a linear observer (Ruiz Vargas and Hemerly, 2001).

Other approaches (Chin, 1994; Pillutla and Keyhani, 1999; Pillutla et al., 1999) include systems for which the measured outputs may be uniquely mapped into the unmeasurable states. In this case, a (adaptive) neural network is trained to approximate the (possibly time-varying) relation between the outputs and the states.

However, all the above presented observers have a serious limitation: it is assumed that the model is linearly dominant.

4.5 Summary and conclusions

In this chapter, observers developed for black and grey box models, in particular for fuzzy and neural systems have been presented.

A large class of nonlinear systems can be represented or successfully approximated by TS fuzzy systems. For these systems, several types of observers have been developed, ranging from the simple Thau-Luenberger observer to sliding mode observers. Two cases has to be distinguished for all observers: the membership functions do/do not depend on the states to be estimated. In the first case, when the membership function do not depend on the states to be estimated, it is relatively simple to design an adequate observer. In the second case, however, the analysis has to be based on robust stability and quadratic Lyapunov functions. However, the design boils down to solving LMIs.

The observers based on neural representations of dynamic systems are more restrictive. Almost in all of the cases, the authors consider that in order to successfully observe the process, it has to be linearly dominant. This might be the case if one desires to study these systems from a robust stability point of view. However, it is our conclusion, that it is not necessary for the system to be linearly dominant, only that the model should sufficiently approximate the real process.

A drawback of both representation is that they do not take into account corrupting noise. In most of the cases, it is considered that the noise is incorporated in the model mismatch, which, in this case will necessarily be present. In order to prevent this model mismatch, several authors use fuzzy-Kalman filters: in this way, instead of dealing separately with the difference between the real process and its model, the noise is used to compensate for it.

While it is theoretically possible, none of the above presented observers consider state estimation in case of adaptive models.

Chapter 5

Estimation in multiagent systems

5.1 Agents and Multiagent Systems

An agent is informally any entity that can perceive its environment and act on it. By extension, a multiagent system (MAS) is a collection of agents that interact with each other, perceive and act upon their environment.

A formalization of agents and multiagent societies is given in what follows (Buşoniu et al., 2005).

Definition 5.1 *A* dynamic agent *is a tuple* $\langle S, Y, U, p, h, s_0 \rangle$*, where:*

- S *is the internal state space of the agent.*
- Y *is the observation space of the agent.*
- U *is the action space available to the agent.*
- $p : S \times Y \to S$ *is the agent transition function, describing how the agent evolves as a result of its observations of the environment.*
- $h : S \times Y \times U \to [0, 1]$ *is the decision probability distribution of the agent, describing its behavior.*
- s_0 *is the agent's initial state.*

Definition 5.2 *A* DMAS *is a tuple* $\langle A, X, f, \{\omega_i\}_{i \in A}, x_0 \rangle$*, where:*

- A *is the set of dynamic agents,* $n = |A|$ *being their number.*
- X *is the environment state space.*
- $f : X \times U \times X \to [0, 1]$*, with* $U = \times_{i \in A} U_i$ *the joint action space, is the environment transition probability distribution, describing how the environment evolves as a result of the agents' actions.*
- $\omega_i : X \times Y_i \to [0, 1], i \in A$ *are the observation probability distributions, describing how the state of the environment is translated into agent observations.*
- x_0 *is the initial environment state.*

In this interpretation, the agent is in complete control of its state space, dynamics and behavior. However, the observation distribution is part of the environment, and the observations resulting from the actions of an agents are the "interface" of the agent with its environment. It is also assumed that the observation function is not completely known by an agent, as it observes not only the results of its own actions, but the (partial) state of the environment as a result of the actions of all the agents present in the environment and (possibly) the environment's own dynamics.

In what follows, we give another model for a single agent behavior in an environment.

Definition 5.3 (Single agent behavior in environment) *Let $x_{A,k}$ denote the state of an agent A at the time instant k, $x_{E,k}$ denote the overall state of the environment at the time instant k, and y_k the observations of the agent at the time instant k. Then, it can be written, that:*

$$\begin{aligned} x_{A,k+1} &= f_A(x_{A,k}, u_k, y_k) \\ x_{E,k+1} &= f_E(x_{E,k}, u_k, x_{A,k}) \\ y_{k+1} &= g(x_{E,k}) \\ u_{k+1} &= h(x_{A,k+1}, y_{k+1}) \end{aligned}$$

where

- f_A – *a possibly nonlinear, time and/or parameter varying function, which describes the evolution of the agent's states as a result of its previous state, actions and observations;* agent model
- f_E – *a possibly nonlinear, time and/or parameter varying function, which describes the evolution of the environment as a result of its previous state, the agent's actions and the agent's state;* environment model
- g – *a possibly nonlinear, time varying function, which presents the observations as a function of the environment state;* observation model
- h – *a possibly nonlinear, time varying and/or parameter varying function, which the agent uses to decide its next action based on its current state and the observations received;* decision model

It has to be emphasized that in this framework, the models are not necessarily restricted to functions, but they are rather mappings. In this way, the agent and decision models can change in time (learning agents). However, the agent and decision models have to be known to the agent itself up to an uncertainty. It is also assumed that the observation model is stationary (i.e. the agent cannot observe states for which it has no sensors). In general, the environment model is only partially known to the agent, and it is never able to observe all the states of the environment.

A model of MAS can be obtained by extending the above model for multiple agents. In this way, each agent has its own agent, observation and decision model, only the environment model is common. A formal definition is given in what follows.

Definition 5.4 *Let $x^i_{A,k}$ denote the state of an agent i at the moment k, $x_{E,k}$ denote the overall state of the environment at the moment k, and y^i_k the observations of the agent at the moment k. Then, it can be written, that:*

$$\begin{aligned} x^i_{A,k+1} &= f_{A,i}(x_{A,k}, u_k, y^i_k), \qquad i = 1, \dots n \\ x_{E,k+1} &= f_E(x_{E,k}, u_k, x_k) \\ y^i_{k+1} &= g_i(x_{E,k}), \qquad i = 1, \dots n \\ u^i_{k+1} &= h_i(x_{A,k+1}, y^i_{k+1}), \qquad i = 1, \dots n \end{aligned}$$

where

- $f_{A,i}$ *– the model of the* i^{th} *agent*
- f_E *– environment model*
- g_i *– the observation model of the* i^{th} *agent*
- h_i *– the decision model of the* i^{th} *agent*
- $x_{A,k}$ *– the joint states of all agents*
- u_k *– the joint actions of all agents*

Based on the above definition, an agent i regards all the other agents as a (partially known) part of the environment. In this way, it can decide its action based also on some states of the environment. It is also considered, that an agent can not observe directly an *action* of another agent, but it is not excluded, that based on the observed results of the actions the agent may deduce the action itself.

5.2 Estimation issues in agent systems

In the literature, it is often considered, that an agent (partially) knows the model of the environment and/or the models of the other agents present in the environment. For the second part, two main categories can be distinguished: teammate and opponent models. In our opinion, these two categories can be combined into one, assuming, that an agent is able to maintain different models for each other agent.

As a result of the above assumptions, several problems can be identified. For a single agent, these problems can be limited by the following four cases, assuming that (at least an approximate) model of the environment can be found:

- The agent perfectly knows its own state and the environments. In this case, the whole system is a completely defined, deterministic process.
- An agent knows its own state but not the environment's: in order to perform some states of the environment have to be estimated.
- The agent knows the environment's state, but not its own: again, in order to chose the correct actions, the agent has to estimate its own states.
- The agent does not know its states nor the environment's: the system is a completely stochastic process, the actions are chosen randomly.

While the above presented are the limit cases, the first and last one are seldom considered in practice. Common examples for the second and third problems are: the environment, though in most of the cases is considered static, may be dynamic and also only part of the variables of interest can be measured; an agent may not even know for certain its own location. An agent might also not be certain of the outcome of an action. Environment and agent state estimation is usually performed in applications such as simultaneous localization and/or mapping and robot soccer.

When dealing with multiagent systems, the above problems are amplified: estimation in multiagent systems can range from estimating the state or a parameter through estimating beliefs up to estimating the model parameters of a whole multiagent society based on the observed interactions between agents, (Gilli and Winker, 2003). However, most applications consider only the estimation of states/parameters of other agents or the environment. The four, generally used scenarios, extended from the single agent scenario are (Mukhopadhyay and Jain, 2001):

1. full observability, when each agent knows the states and actions of all other agents, and there is no uncertainty at all
2. each agent knows the actions of all other agents but not their states. There are two subcases: the agent knows/does not know its local state.
3. each agent knows only its local state and action
4. complete ignorance: an agent does not know even its local state, does not try to estimate, and actions are selected randomly

However, the above cases do not take into consideration the knowledge about the environment, which may vary for each case, and for each agent.

Typically, in the case of estimation in MAS, the two cases (agent estimation and environment estimation) are considered separately.

Agent estimation

In this case, again, two types are distinguished: teammate modeling and opponent modeling. However, both aim to learning about the other agents in the environment in order to make "good guesses" about their behavior or future actions and to act accordingly (better cooperation or counteracting). Such an approach is used by Boutilier (1996), who employ Bayesian learning methods for updating models of other agents. Based on these models, agents estimate the current behavior of their teammates in order to better cooperate with them.

From the point of view of agent i, it has to model all the other agents, and in return all the other agents model the agent i. In order to completely model the other agents, modeling their beliefs about agent i is also necessary. In this way, beliefs of other agents have to be estimated. Hu and Wellman (1998); Doshi and Gmytrasiewicz (2005) consider several levels of agent models : 0^{th} level, or non-estimating agents, 1^{st} level, when an agent estimates based only on its observations, 2^{nd} level, when an agent estimates based on observations and other agents' beliefs, which in turn depend on observations, and so on.

A 0^{th} level agent believes that no other agent performs learning or modeling activities, and it does not consider their behaviors as adaptive. A 1^{st} level agent considers all the other agents as 0^{th} level agents. In general, an n^{th} level agent considers all the other agents present in the environment as n-1^{th} level agents.

However, starting from the 3rd level, estimation becomes too complex to be implementable (an agent estimates the beliefs of other agents about what it believes). Doshi and Gmytrasiewicz (2005) investigates the use of 0, 1 and 2nd level modeling agents. The authors report that for some domains, modeling other agents' behavior is not necessary.

While in cooperative systems, the need for estimation may be reduced due to communication, Shaban et al. (2002) identify two types of uncertainty: first, a local one, the uncertainty of the agent in its own decision, and second, a global one, concerning knowledge sharing. The information received from another agent may be in contradiction with the agent's own beliefs.

Environment estimation

In case of modeling the environment, again two main cases can be considered: single agent environment modeling and environment estimation in MAS.

In the first case, an agent has to rely on its own observations. In literature, it is generally considered in simultaneous localization and mapping problems, that the environment is static, and can be represented by a map, which can be learned by an agent.

Also, in most of the cases, a first order MDP or POMDP is considered, such as by Simmons and Koenig (1995), where a robot has to navigate through an office environment. In this case, observations are given as numerical data, such as distances or angles. However, for localization, it might be more helpful to model observations in other domains, such as in (Kröse et al., 2001), where localization is performed based on visual information.

In MAS, it would seem that the problem is reduced, assuming that the agents are able to communicate their observations to their teammates. However, other problems are likely to appear: communication may be restricted; the received information may be corrupted during transmission or in contradiction with the agent's own information; it may not be necessary for all the agents to maintain a map of the environment at all times.

Assuming that one of the agents is building a map of the environment, it is necessary to integrate all the information received: this means filtering out erroneous information, resolving contradiction, fusion of knowledge. Also, in general it may not be possible to represent the system as a first order Markov process. When all the agents act upon the environment in the same time, it might not be possible to discern the results of the actions separately, given the observations.

Sensor fusion

A problem inherently present in multiagent systems is the fusion of the data available for each agent in order to obtain a global overview of the environment.

Assuming that each agent in the system is equipped with its own sensors, each measurement can suffer from the following problems: breakdown of the sensor, limited spatial coverage, imprecision. The multiagent system, from the point of view of the sensors, can also be seen as a sensor network, with (possibly) redundancy. If the system is cooperative, by combining the data obtained by each agent, information, which otherwise cannot be accessed, can become available.

Consider in the following a cooperative system. Then, the information fusion can take place on different levels and by different agents. Depending on the application, if every information is transmitted, each agent can fuse the measured and received information, based on its own beliefs. However, this method would most certainly lead to an overhead of communication, and would be unfeasible in a real-time system. Another approach is hierarchical systems. One

agent can be chosen from a number of agents to fuse the information, while the role of the others would be simply to acquire data. However, this approach also leads to problems: if the agent which combines information breaks down, all the information is lost.

Also, the fusion of the information can take place on different levels. The lowest level can be considered if the measured data is transmitted and fused. In this case, each agent, at each time instant when a measurement becomes available, receives it and can proceed to incorporate it. This approach necessitates information about the reliability of all the sensors. Transmitting and fusing information already processed (filtered, corrected, estimated) constitutes a second level of fusion. However, in this case each agent should asses the reliability of the information transmitted, or maintain degrees of reliability of each other agent.

The highest level of fusion can be considered decision fusion: each agent transmits its own beliefs about what decision should be made, together with the reasoning based on which the decision was taken. Again, in this case, each agent has to maintain beliefs about the reliability of the other agents in order to cope with (possibly) contradicting decisions.

In all the cases, one has to keep in mind the application for which the fusion has to be realized and the goal of the fusion. For instance, if a multiagent system has to map an unknown environment, using sensors with limited spatial coverage, in a first step it (probably) does not make sense to direct all the agents to the same place and try to obtain highly reliable information about the same object.

5.3 Predictive agents

In the literature, attention is given to predicting other agents' movements, their states and (most important) their actions. The knowledge on other agents behavior is called the model of the agent. This approach is often called "opponent modeling", however, it is useful in cooperative settings as well, in particular when the agents have only limited knowledge about each other, or the communication is restricted.

Boutilier (1996) presents the two limit cases of estimation of another agent's policies. The "lower" limit can be considered, when the agent i has no information about the model or policy of agent j, and does not attempt to build an analytical expression of it. In this case, agent i maintains a map of the frequencies with which agent j chooses a particular action in a given state. Assuming, that agent j achieves a stationary state in finite time (i.e. its decisions will no longer change), this map eventually becomes accurate. However, this method can only be used in case of a finite universe of states and actions.

The "upper" limit of agent modeling is full Monte Carlo estimation, which may include also the agents beliefs. Assuming, that a prior distribution of agent j decisions given the states is available, the agent i observes the decisions of agent j, and updates its beliefs. However, this approach is computationally too complex, and, except for very simple scenarios, not implementable in real time, especially considering, that not only one agent has to be modeled.

A more realistic approach is to assume that an agent maintains several models about other agents (Chiao and Xydeas, 2003; Carmel and Markovitch, 1996). When the agent which attempts modeling them, obtains a new observation or interacts interact with them, it also classifies the other agents or increases/ decreases the belief that an agent has a certain model. In this setting, it is also possible to estimate parameters of certain models. This principle is applied by Bui et al. (1999), where agents "learn" in a negotiation context about other agents preferences, based on previous knowledge/ observations.

When learning does not concern other agents' internal states, but only attaining a certain goal, simpler scenarios can be considered. For example, agents can learn to coordinate, knowing the outcome of possible actions (Sen et al., 1994).

Two problems can be defined in the context of predictive agents.

The first question to be solved is in how much detail must an agent model another agent: is it enough to be able to predict the actions of another agents, or it also has to model the internal states? While complex modeling schemes, which include modeling the beliefs about other agents may be near impossible to implement, they are not necessary in all the cases. Scheutz and Schermerhorn (2003) compare different types of predictive agents and argue that a complex predictive agent, while it is computationally more expensive, works just slightly better than a simpler one. However, complexity of modeling depends on the application.

On the other hand, since an agent might not observe the outcome of other agents' action at every step, multistep prediction might be necessary in agent societies. Three types of predictions are compared by Henninger et al. (2004): based only on the model, without considering any kind of noise or uncertainty; neural networks and extended Kalman filter. It seems that when the error tolerance is small, the NN is much superior to the other approaches, but this difference decreases as the tolerance is increased.

Chapter 6

Conclusions

This survey presents approaches for dealing with imperfection in data and in models. In particular filters and observers are discussed for different types of model representations. In this chapter, the content of the survey is discussed and some directions for future research are given.

6.1 Summary

Throughout the survey, theories and methods for reducing numerical uncertainty are presented. Uncertainty, or in a more general case, imperfection is present in every system due to unmodeled or unknown aspects of the process, approximate models, unmeasurable variables of interest or simply noise corruption. An overview of the different types of imperfection is given in Chapter 1.

Chapter 2 discussed several type of theories used for handling imperfection: probability and related theories, fuzzy sets and combination of probabilities and fuzzy logic.

The most well-known of them is probability theory. It assumes that the model of the system to be controlled or for which decisions have to be made is known up to a probabilistic uncertainty (corrupting noise). While not explicitly stated, it is often assumed that the known part of the system is dominant. By using probability theory, all imperfections (including variables) are represented as random numbers, for which some properties, such as mean and variance are known, or can be estimated. Inference in the probabilistic framework is commonly realized using Bayes rule. Probability theory can be used for dealing with imperfection in both continuous and discrete systems, as long as (in the continuous case), the analytical expression of properties can be derived.

The generalization of probability theory led to upper and lower probabilities. While the theory is intriguing, the practical implementations is somehow restricted, as long as one has to make one decision, not study a class of decisions.

Bayesian networks are graphical representations of interactions in a system. However, as it cannot represent an infinity of states, continuous systems are excluded.

Dempster-Shafer theory provides a mean to combine evidences. In this theory, the probability of an event to occur is represented by belief function. Its practical implementation may be somehow tedious, as in order to combine evidence, one has to guarantee the independence of the sources of the evidence to be combined.

Fuzzy sets instead of representing the likelihood of an imperfect data, are used to deal with vagueness. However, in this survey we also consider representing a model using fuzzy logic, as a combination of local models.

Combinations of probability theory and fuzzy logic is also presented. However, the attempts of combining the two theories, while technically solid, have yet little practical importance.

Chapter 3 presents two classes of probabilistic estimators: Kalman and particle filters. Both classes consider a probabilistic representation of a dynamic model.

The best known variant of the Kalman filter is the linear Kalman filter, which is an optimal estimator in the least square sense. However, it can be used only for linear systems. Its extensions to nonlinear systems, are the Extended Kalman filter, based on the linearization of the system, and the Unscented Kalman filter, which uses the true nonlinear model.

Particle filters represent distributions by random samples and associated weights. Their main advantage is that they can handle an type of nonlinear models and any distributions.

Both types of filters can be used for state and/or parameter estimation. While in principle particle filters are more accurate, there is no general rule which performs better for a certain application.

In Chapter 3 only the basic variants of the filters are presented. However, there exist several combination of the two filters, with (for certain applications) better performance.

Chapter 4 presents observers for black and grey-box models.

For TS fuzzy systems several types of observers have been developed, ranging from the simple Thau-Luenberger observer to sliding mode observers. For each observer, two cases were considered: the membership functions do or do not depend on the states to be estimated. In both cases, design of a stable observer is reduced to solving LMIs.

The observers based on neural representations of dynamic systems are more restrictive. For almost all cases of neural models, the authors consider that the process to be modeled has to be linearly dominant.

A drawback of both representation is that they do not take into account corrupting noise. In most of the cases, it is considered that the noise is incorporated in the model mismatch, which, in this case will necessarily be present.

While for both representations it is possible to consider adaptive models, none of the authors considered state estimation and model adaptation together.

Chapter 5 gives a short overview of agent systems and the possible estimation problems in these systems. In most of the cases, it is considered that the an agent knows its environment and it only has to estimate/filter its own state or that its states are known and has to map the environment. The only applications that consider state end environment estimations are the simultaneous localization and mapping problems.

Estimation in multiagent systems is more complex. In a general case, and agent should have no information about the environment, its own state or about other agents present in the environment. However, all the studied applications are concerned with or agent estimation or environment estimation separately.

It is our conclusion, that these problems should be dealt simultaneously, building both models of the agents studied and the environment.

6.2 Future research directions

We conclude this survey by identifying several research questions that have not been answered throughout the literature surveyed.

1. Possible combinations of Kalman and particle filter in order to prevent divergence of the estimator. While there exist several Kalman-particle filters, none of the is guaranteed to converge.

2. Automated tuning of unknown model parts: both for Kalman and particle filters it is assumed that the distribution of the corrupting noise is known. In the case of Kalman filters, in general another filter is used prior to the state estimation to determine the covariance of the corrupting noise. However, there might be cases, in which state and covariance has to be estimated simultaneously.

3. Compensation of model mismatch: the common tendency in case of probabilistic models is to compensate with noise for the difference between the model and the actual plant. However, even if the estimates will be close to the real states, the noise on the estimate cannot be filtered. Also, the model mismatch can represent an unmodelled, but important part of the dynamics.

4. Adapting models: in several situations, the process to be estimated changes on-line. In this case, the model should also be adapted simultaneously. A solution for this would be to use fuzzy or neural models, which can be adapted online while estimating. Also, a method for incorporating new information about the model has to be developed.

5. Estimation in multiagent systems: several estimation issues can be considered in the context of multiagent systems: state, environment or even estimation/prediction of other agents' actions. It can also be considered as an application area for the previous item.

List of Symbols

Common notations

$\cdot$	generic placeholder for
v^{T}	vector transpose
v^*	optimal value
$\widehat{v}$	estimated value
$P(\cdot)$	probability
$P(\cdot \mid \cdot)$	conditional probability
$\Pi(\cdot)$	possibility distribution

Model notations

k	current step in discrete time
x_k	state vector at step k
y_k	measurement vector at step k
u_k	input vector at step k
F_k	state transition matrix at step k
B_k	input matrix at step k
H_k	output matrix at step k
v_k	state transition noise at step k
η_k	measurement noise at step k
f	state transition function
h	measurement function
$p(x_0)$	initial probability density function
$p(x_k\|x_{k-1})$	state transition model; the probability $p(x_k\|x_{k-1}) = P(X_k = x_k\|X_{k-1} = x_{k-1})$
$p(y_k\|x_k)$	measurement model; the probability $p(y_k\|x_k) = P(Y_k = y_k\|X_k = x_k)$

Probabilistic filters

$x_{k\|k-1}$	predicted state at step k, based on the results obtained in the previous step
$P_{k\|k-1}$	predicted state covariance at step k, based on the results obtained in the previous step

P_k	updated state covariance at step k
K_k	Kalman gain at step k
x_k^i	sampled state at step k
w_k^i	the weight of the i^{th} sample at step k
P_{xx}	estimated state covariance
P_{yy}	estimated measurement covariance
P_{xy}	cross-correlation matrix
N_{S}	number of samples of the particle filter
φ_k	vector of all measurements up to step k $\varphi_k = [y_1\ y_2\ ...\ y_k]$
$\mathcal{N}(m, \sigma)$	normal distribution with mean m and variance σ
$\mathcal{U}(m, M)$	uniform distribution on the interval $[m, M]$
$\mathcal{N}(x; m, \sigma)$	the value of the normal probability density function with mean m and variance σ evaluated in x
$p(x_k\|\varphi_{k-1})$	the probability density function of the states at step k, given the measurements up to step $k-1$; the *prior*
$p(x_k\|\varphi_k)$	the probability density function of the states at step k, given all the measurements (including y_k); the *posterior*
$q(\cdot)$	importance density function
δ	Dirac delta function

Fuzzy observers

θ	premise variables
w_i	membership function of the i^{th} rule
A_i	state transition matrix of the i^{th} rule
B_i	input matrix of the i^{th} rule
C_i	output matrix of the i^{th} rule
L_i	observer gain of the i^{th} rule

Bibliography

Aja-Fernandez, S., Alberola-Lopez, C., and Ruiz-Alzola, J. (2003). A fuzzy-controlled Kalman filter applied to stereo-visual tracking schemes. *Signal Processing*, 83:101–120.

Angelova, D. S., Semerdjiev, T. A., Jilkov, V. P., and Semerdjiev, E. A. (2001). Application of a Monte Carlo method for tracking maneuvering target in clutter. *Mathematics and Computers in Simulation*, 55:15–23.

Arulampalam, S., Maskell, S., Gordon, N. J., and Clapp, T. (2002). A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188.

Baldwin, J. and Di Tomaso, E. (2003). Inference and learning in fuzzy Bayesian networks. In *Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on*, volume 1, pages 630–635.

Baz, A. (1992). A neural observer for dynamic systems. *Journal of Sound and Vibration*, 152:227–243.

Bergsten, P. (2001). *Observers and controllers for Takagi-Sugeno fuzzy systems*. PhD thesis, Örebro University.

Bergsten, P. and Palm, R. (2000). Thau-Luenberger observers for TS fuzzy systems. In *The Ninth IEEE International Conference on Fuzzy Systems, 2000. FUZZ IEEE 2000.*, volume 2, pages 671 – 676.

Bergsten, P., Palm, R., and Driankov, D. (2001). Fuzzy observers. In *The 10th IEEE International Conference on Fuzzy Systems, 2001.*, volume 2, pages 700–703.

Bergsten, P., Palm, R., and Driankov, D. (2002). Observers for Takagi-Sugeno fuzzy systems. *IEEE Transactions on Systems, Man and Cybernetics*, 32:114–121.

Boers, Y. (1999). On the number of samples to be drawn in particle filtering. *IEE Colloquium on Target Tracking: Algorithms and Applications*, 5:1–6.

Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. In *Conference on Theoretical Aspects of Rationality and Knowledge*, pages 195–210.

Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455.

Buşoniu, L., Schutter, B. D., and Babuška, R. (2005). Learning and coordination in dynamic multiagent systems. Technical Report 05-019, Delft University of Technology.

Bui, H. H., Venkatesh, S., and Kieronska, D. (1999). Learning other agents' preferences in multi-agent negotiation using the Bayesian classifier. *International Journal of Cooperative Information Systems*, 8(4):275–294.

Carmel, D. and Markovitch, S. (1996). Opponent modeling in multiagent systems. In Weiss, G. and Sen, S., editors, *Adaption And Learning In Multi-Agent Systems*, volume 1042 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.

Caron, F., Duflos, E., Pomorski, D., and Vanheeghe, P. (2004). GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. Information Fusion, In Press, Corrected Proof.

Chiao, S.-Y. and Xydeas, C. S. (2003). Modeling the behaviors of players in competitive environments. In *International Conference on Intelligent Agent Technology (IAT 2003)*, pages 566–569.

Chin, L. (1994). Application of neural networks in target tracking data fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 30:281–287.

Ding, Z., Leung, H., Chan, K., and Zhu, Z. (2001). Model-set adaptation using a fuzzy kalman filter. *Mathematical and Computer Modelling*, 34:799–812.

Doshi, P. and Gmytrasiewicz, P. (2005). Approximating state estimation in multiagent settings using particle filters. In *In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems Conference (AAMAS)*.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208.

Elanayar, S. and Yung, C. (2000). State estimation of continuous-time radial basis function networks. *Automatica*, 36:399–407.

Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. PhD thesis, University of Oxford.

Garcia, L. A. and Shigidi, A. (2005). Using neural networks for parameter estimation in ground water. Journal of Hydrology, In Press, Corrected Proof.

Gilli, M. and Winker, P. (2003). A global optimization heuristic for estimating agent based models. *Computational Statistics and Data Analysis*, 42:299–312 .

Henninger, A., Madhaven, R., , and Schlenoff, C. (2004). Empirical comparison of predictive models for mobile agents. *Robotics and autonomous systems*, 49(1-2):91–103.

Hu, J. and Wellman, M. (1998). Online learning about other agents in a dynamic multiagent system. In Sycara, K. P. and Wooldridge, M., editors, *Proceedings of the Second International Conference on Autonomous Agents (Agents-98)*, pages 239–246. ACM Press.

Jetto, L., Longhi, S., and Vitali, D. (June 1999). Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted kalman filter. *Control Engineering Practice*, 7:763–771.

Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*.

Kadmiry, B. and Bergsten, P. (2004). Robust fuzzy gain scheduled visual-servoing with sampling time uncertainties. In *Proceedings of the 2004 IEEE International Symposium on Intelligent Control, 2004.*, volume 1, pages 239 – 245.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45.

Kato, Y., Izuka, T., Ohtsuki, R., and Yamaguchi, S. (1999). A proposal for a new fuzzy probability distribution function. In *IEEE International Fuzzy Systems Conference Proceedings*, volume 3, pages 1334–1339.

Kim, Y. H., Lewis, F. L., and Abdallah, C. T. (1997). A dynamic recurrent neural-network-based adaptive observer for a class of nonlinear systems. *Automatica*, 33:1539–1543.

Kröse, B., Vlassis, N., Bunschoten, R., and Motomura, Y. (2001). A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391.

Li, P., Zhang, T., and Ma, B. (2004). Unscented Kalman filter for visual curve tracking. *Image and Vision Computing*, 22:157–164.

Li, T.-H. (1993). Estimation of one-dimensional radar tracking via fuzzy-Kalman filter. In *Proceedings of the IECON '93., International Conference on Industrial Electronics, Control, and Instrumentation*, volume 3, pages 2384–2388.

Lin, F.-J. and Wai, R.-J. (2002a). Hybrid computed torque controlled motor?toggle servomechanism using fuzzy neural network uncertainty observer. *Neurocomputing*, 48:403–422.

Lin, F.-J. and Wai, R.-J. (2002b). Robust control using neural network uncertainty observer for linear induction motor servo drive. *IEEE Transactions on Power Electronics*, 17:241–254.

Lin, F.-J., Wai, R.-J., Lin, C.-H., and Liu, D.-C. (2000). Decoupled stator-flux-oriented induction motor drive with fuzzy neural network uncertainty observer. *IEEE Transactions on Industrial Electronics*, 47:356–367.

Loebis, D., Sutton, R., Chudley, J., and Naeem, W. (2004). Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent auv navigation system. *Control Engineering Practice*, 12:1531–1539.

Marino, P., Milano, M., and Vasca, F. (1999). Linear quadratic state feedback and robust neural network estimator for field-oriented-controlled induction motors. *IEEE Transactions on Industrial Electronics*, 46:150–161.

Maybeck, P. S. (1979). *Stochastic models, estimation, and control*, volume 1 of *Mathematics in Science and Engineering*, chapter Why sochastic models, estimation and control, pages 1–16. Academic Press.

McGinnity, S. and Irwin, G. (1996). Fuzzy local linear models for target tracking. In *IEE Colloquium on Target Tracking and Data Fusion*, pages 10/1–10/3.

McGinnity, S. and Irwin, G. (1997). Nonlinear Kalman filtering using fuzzy local linear models. In *Proceedings of the 1997 American Control Conference*, pages 3299–3300.

Meghdadi, A. and Akbarzadeh, T.-R. (2003). Uncertainty modeling through probabilistic fuzzy systems. In *Fourth International Symposium on Uncertainty Modeling and Analysis*, pages 56–61.

Meghdari, A., Naderi, D., and Alam, M. (2005). Neural-network-based observer for real-time tipover estimation. *Mechatronics*, 15:989–1004.

Mukhopadhyay, S. and Jain, B. (2001). Multi-agent Markov decision processes with limited agent communication. In *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, pages 7–12.

Oztopal, A. (2005). Artificial neural network approach to spatial estimation of wind velocity data. Energy Conversion and Management, In Press, Corrected Proof.

Palm, R. and Bergsten, P. (2000). Sliding mode observer for a Takagi-Sugeno fuzzy system. In *The Ninth IEEE International Conference on Fuzzy Systems, 2000. FUZZ IEEE 2000.*, volume 2, pages 665–670.

Pan, Y., Klir, G., and Yuan, B. (1996). Bayesian inference based on fuzzy probabilities. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, volume 3, pages 1693–1699.

Pearl, J. (1996). Decision making under uncertainty. *ACM Computing Surveys*, 28:89–92.

Pillutla, S. and Keyhani, A. (1999). Development and implementation of neural network observers to estimate the state vector of a synchronous generator from on-line operating data. *IEEE Transactions on Energy Conversion*, 14:1081–1087.

Pillutla, S., Keyhani, A., and Kamwa, I. (1999). Neural network observers for on-line tracking of synchronous generator parameters. *IEEE Transactions on Energy Conversion*, 14:23–30.

Rekleitis, I. M. (2004). A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, 3480 University St., Montreal, Québec.

Ruiz Vargas, J. and Hemerly, E. (2001). Adaptive observers for unknown general nonlinear systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 31:683–690.

Sasiadek, J. and Khe, J. (2001). Sensor fusion based on fuzzy Kalman filter. In *Proceedings of the Second International Workshop on Robot Motion and Control*, pages 275–283.

Sasiadek, J. and Wang, Q. (1999). Sensor fusion based on fuzzy Kalman filtering for autonomous robot vehicle. In *1999 IEEE International Conference on Robotics and Automation*, volume 4, pages 2970–2975.

Scheutz, M. and Schermerhorn, P. (2003). Many is more, but not too many: dimensions of cooperation of agents with and without predictive capabilities. In *IEEE/WIC International Conference on Intelligent Agent Technology*, pages 378–384.

Schroder, D., Hintz, C., and Rau, M. (2001). Intelligent modelling, observation and control for nonlinear systems. *IEEE/ASME Transactions on Mechatronics*, 6:122–131.

Sen, S., Sekaran, M., and Hale, J. (1994). Learning to coordinate without sharing information. In *Proc. of AAAI-94*, pages 426–431, Seattle, WA.

Sentz, K. and Ferson., S. (2003). Combination of evidence in dempster-shafer theory. Technical report, Sandia National Laboratories.

Shaban, K., Basir, O., Kamel, M., and Hassanein, K. (2002). Intelligent information fusion approach in cooperative multiagent systems. In *5th Biannual World Automation Congress*, volume 13, pages 429–434.

Sheu, T.-T. and Chen, T.-C. (1999). Self-tuning control of induction motor drive using neural network identifier. *IEEE Transactions on Energy Conversion*, 14:881–886.

Simmons, R. and Koenig, S. (1995). Probabilistic robot navigation in partially observable environments. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1080–1087.

Simon, D. (2002). Training fuzzy systems with the extended kalman filter. *Fuzzy Sets and Systems*, 132:189–199.

Simon, D. (2003). Kalman filtering for fuzzy discrete time dynamic systems. *Appl. Soft Comput.*, 3(3):191–207.

Simutis, R., Havlik, I., and Lubbert, A. (1992). A fuzzy-supported Extended Kalman filter: a new approach to state estimation and prediction exemplified by alcohol formation in beer brewing. *Journal of Biotechnology*, 24:211–234.

Smets, P. (1991). Varieties of ignorance and the need for well-founded theories. *Information Sciences*, 57:135–144.

Smets, P. (1997). *Imperfect information : Imprecision - Uncertainty*, chapter Uncertainty Management in Information Systems: from Needs to Solutions, pages 225–254. Kluwer, Boston.

Smets, P. (1998a). *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 1, chapter Probability, Possibility, Belief: Which and Where ?, pages 1–24. Kluwer, Doordrecht.

Smets, P. (1998b). *Numerical Representation of Uncertainty.*, volume 1, pages 1–24. Kluwer, Dordrecht.

Smets, P. (1999). Theories of uncertainty. Handbook of fuzzy computation.

Tanaka, K., Ikeda, T., and Wang, H. (1998a). Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs. *IEEE Transactions on Fuzzy Systems*, 6:250–265.

Tanaka, K., Kosaki, T., and Wang, H. (1998b). Backing control problem of a mobile robot with multiple trailers: fuzzy modeling and LMI-based design. *IEEE Transactions on Systems, Man and Cybernetics*, 28:329–337.

Tanaka, K. and Sano, M. (1994). On the concepts of regulator and observer of fuzzy control systems. In *Proceedings of the Third IEEE Conference on Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence.*, volume 2, pages 767–772.

Tanaka, K., Taniguchi, T., and Wang, H. (1998c). Model-based fuzzy control of TORA system: fuzzy regulator and fuzzy observer design via LMIs that represent decay rate, disturbance rejection, robustness, optimality. In *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence.*, volume 1, pages 313–318.

Tanaka, K. and Wang, H. (1997). Fuzzy regulators and fuzzy observers: a linear matrix inequality approach. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 2, pages 1315 – 1320.

Taniguchi, T., Tanaka, K., Ohtake, H., and Wang, H. (2001). Model construction, rule reduction, and robust compensation for generalized form of Takagi-Sugeno fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 9:525–538.

Taniguchi, T., Tanaka, K., Yamafuji, K., and Wang, H. (1999a). A new PDC for fuzzy reference models. In *1999 IEEE International Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99.*, volume 2, pages 898–903.

Taniguchi, T., Tanaka, K., Yamafuji, K., and Wang, H. (1999b). Nonlinear model following control via Takagi-Sugeno fuzzy model. In *Proceedings of the 1999 American Control Conference*, volume 3, pages 1837–1841.

Teixeira, M. and Zaverucha, G. (2003). Fuzzy Markov predictor in multi-step electric load forecasting. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, pages 3065–6070.

van den Berg, J., Kaymak, U., and van den Bergh, W.-M. (2002). Fuzzy classification using probability-based rule weighting. In *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, volume 2, pages 991–996.

van der Merwe, R., de Freitas, J. F. G., and Doucet, A. (2001). The unscented particle filter. In *Advances in Neural Information Processing Systems*, Cambridge, MA. MIT Press.

van der Merwe, R. and Wan, E. (2003). Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong.

Voorbraak, F. (1995). Reasoning with uncertainty in AI. In *Reasoning with Uncertainty in Robotics*, pages 52–90.

Wang, C.-H., Lin, T.-C., Lee, T.-T., and Liu, H.-L. (2002). Adaptive hybrid intelligent control for uncertain nonlinear dynamical systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 32:583–597.

Wang, H., Li, J., Niemann, D., and Tanaka, K. (2000). T-S fuzzy model with linear rule consequence and PDC controller: a universal framework for nonlinear control systems. In *The Ninth IEEE International Conference on Fuzzy Systems, 2000. FUZZ IEEE 2000.*, volume 2, pages 549–554.

Wang, H., Tanaka, K., and Griffin, M. (1996). An approach to fuzzy control of nonlinear systems: stability and design issues. *IEEE Transactions on Fuzzy Systems*, 4:14–23.

Welch, G. and Bishop, G. (2002). An introduction to the Kalman filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina, NC, USA.

Yu, Y. and Cheng, Q. (2005). Particle filters for maneuvering target tracking problem. Signal Processing, In Press, Corrected Proof.

Zadeh, L. A. (1984). Fuzzy probabilities. *Inf. Process. Manage.*, 20(3):363–372.

Zadeh, L. A. (2005). Toward a generalized theory of uncertainty (GTU) - an outline. *Information Sciences*, 172(1-2):1–40.

Zhang, S.-T. and Wei, X.-Y. (2003). Fuzzy adaptive Kalman filtering for DR/GPS. In *International Conference onMachine Learning and Cybernetics*, volume 5, pages 2634–2637.