TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Realzeit-Computersysteme

Reliability and Safety Improving Methods for the Evaluation of Driving Environment Information

Multiple Fault and Target Tracking for Radar and Lidar Based Sensors

Clemente de Castro Bonfim

Vollständiger Abdruck der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität München zur Erlangung des akademischen Grades eines
Doktor – Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Klaus Diepold
Prüfer der Dissertation:
1. Univ.-Prof. Dr.-Ing Georg Färber
2. Univ.-Prof. Dr.-Ing Fernando Puente León

Die Dissertation wurde am 29.09.2008 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 29.03.2009 angenommen.

Contents

Acknowledgements

I am indebted to many for providing me incentive and support during the time I was working at my doctoral thesis. First of all, I wish to thank Mr. Dr.-Ing. Karl Naab for giving me the opportunity to work under his supervision as well as to perform the case study of this thesis at the BMW Group. Especially I am grateful for the scientific freedom he gave me, enabling me to follow my ideas.

My gratitude also goes to Univ.-Professor Dr.-Ing. Georg Färber for giving me the opportunity to work under his supervision in the chair in Real Time and Computer System at Technical University Munich.

Special thanks go to Mr. Dr.-Ing. Klaus Gresser, Mr. Dr.-Ing. Nico Kämpchen and to the whole Sensor Technology group in the Research and Development department at the BMW Group for the many fruitful discussions and excellent team work.

I would like to thank the colleagues and friends Jörg Schneider and Bastian Best, who supported me generously with their advices and collegiality. During the years working on our thesis we shared the same office, our ideas and the ups and downs of a doctoral work.

My gratitude goes especially to my family in Brazil for their support and encouragement during my whole academic life as well as for their understanding of the long-distance relationship during the time I was working on my doctoral thesis.

To my wife Sonja I have not enough words to express my gratitude. She supported and encouraged me not only during the time I was working on my thesis, but also during the whole time I have been living in Germany. Her love and friendship gave me strength to keep on course even during the hardest moments.

Kurzfassung

Motivation

Die Erfassung von Vorgängen in der Fahrumgebung ist eine wesentliche Voraussetzung für die Realisierbarkeit zukünftiger Fahrassistenzsysteme (FAS). Solche Vorgänge können anhand verschiedener Sensoren ermittelt werden. Dabei müssen die Zuverlässigkeit dieser Information und deren Verarbeitungsmethoden, abhängig von den FAS- Anforderungen, gewährleistet sein. In diesem Zusammenhang entsteht eine wichtige Fragestellung: Wie kann die Information von diversitären und teilweise redundanten Sensoren genutzt werden, um eine maximale Leistungsfähigkeit zu erreichen? Die Sensoren müssen so kombiniert werden, dass sie kooperieren, sich ergänzen und konkurrieren können. Durch Sensordaten-Fusion und Multiple Target Tracking wird die Information unterschiedlicher Sensoren vereinigt, die Robustheit und Redundanz erhöht und "neue" Information basierend auf Modellannahmen und Hypothesen abgeleitet.

Ein weiterer wichtiger Aspekt ist, die Zuverlässigkeit der Fahrumgebungsinformation trotz Sensorfehlfunktionen zu garantieren. Sensorfehler müssen schnellstmöglich erkannt und isoliert werden. Fehlerdetektions- und Fehlerisolierungsmethoden (engl. Fault Detection and Identification-FDI) bieten verschiedene wissensbasierte Verfahren, die auf analytischen oder heuristischen Informationen beruhen. Sensordaten- Fusion und FDI- Methoden ermöglichen u.a. die Evaluierung der Fahrumgebungsinformation. Beide Ansätze können sich ergänzen und gegenseitig überwachen, da sie teilweise gemeinsame Ziele und Strategien aufweisen und sich effizient miteinander integrieren lassen. Durch die Kombination und Weiterentwicklung beider Methoden kann eine höhere Zuverlässigkeit von Fahrerassistenzsystemen erreicht werden.

Ansatz

Das Ziel dieser Arbeit ist es, eine neue Methode zu präsentieren, die die Eigenschaften von Sensordaten- Fusion, Multiple Target Tracking und FDI- Strategien vorteilhaft nutzt. Dies soll nicht nur die Erhöhung der Zuverlässigkeit der erfassten Fahrumgebungsinformation, sondern auch die weitere Verbesserung spezifischer FAS (ACC, Bremsassistent, usw.) ermöglichen. Der vorgeschlagene Ansatz basiert auf der Analyse der erfassten Sensordaten. Aus diesen Messdaten werden relevante Merkmale extrahiert, die zu deren Klassifizierung dienen. Merkmale auf der Messwertebene werden in drei Kategorien aufgeteilt: Merkmale aus

Messdaten, Merkmale aus der Vergangenheitsbetrachtungen erkannter Objekte, Merkmale aus der Vergangenheitsbetrachtungen von Sensoreigenschaften. Anhand der extrahierten Merkmale und deren Kombination werden Messdaten unterschiedlichen Hypothesen zugeordnet. Diese Zuordnung wird unterstützt durch mathematische Modelle, die bestimmte Erwartungen an das Verhalten der Daten beschreiben.

Abhängig davon, ob die Merkmale auf das Modell zutreffen oder vom ihm abweichen, wird die Information klassifiziert. Die Abweichung, auch Residuum genannt, dient als Indiz (Symptom) für die Bestätigung von Hypothesen. Messdaten können vier Hypothesenzustände annehmen: bestehende erkannte Objekte, neue erkannte Objekte, spezifische Messfehler, unbestimmte Messfehler. Der Begriff Messfehler wird als "vorgetäuschte" Messgröße interpretiert, die z.B. durch das Messprinzip bedingt ("Geisterobjekte") und die von unterschiedlichem Ausmaß (tolerierbar bis gravierend) sein können.

Ein zeitlich zusammenhängendes Set von Eigenschaften eines Objektes (z.B. Position, Geschwindigkeit) wird als Track bezeichnet. Durch die Kombination von Messdatenhypothesen unterschiedlicher Messdaten mit den extrahierten Track- Merkmalen kann die nächste Hypothesenebene (Information ist einem Track zuordenbar; Information ist keinem Track zuordenbar) bestimmt werden. Analog zur Messdatenklassifizierung werden Symptome für Track-Hypothesen anhand mathematischer Modelle berechnet. Auf den beschriebenen Ebenen werden Hypothesen in jedem Zyklus klassifiziert, die den dynamischen Teil des vorgeschlagenen Ansatzes bilden. Im quasistationären Teil werden die Sensorfehlerhypothesen bestimmt.

Als Sensorfehler wird ein partieller oder kompletter Ausfall z.B. Sensorhardware wie Dejustage, Teil-Blindheit usw. bezeichnet. Auch in dieser Ebene werden die Symptome mit Hilfe mathematischer Modelle bestimmt und die Ergebnisse von den vorherigen Ebenen mit den extrahierten Sensorfehlermerkmalen kombiniert. Der Hauptunterschied liegt darin, dass die Klassifizierung nicht in jedem Arbeitszyklus durchgeführt wird, sondern in vorgegebenen Zeitfenstern, die abhängig vom untersuchten Sensorfehler sind. Sensorfehler werden in drei Hypothesenklassen unterteilt: spezifische Sensorfehler, unbestimmte Sensorfehler, kein Sensorfehler. Die erste Klasse beschreibt modellierte Sensorfehler, bei denen die extrahierten Merkmale zutreffen. Der zweiten Hypothesenkategorie werden Sensorfehler zugeordnet, die nicht modelliert werden konnten (z.B. aus Komplexitätsgründen) und deren Merkmale weder zur ersten noch zu dritten Hypothesenklassen zugeordnet werden konnten. Als fehlerfrei werden Sensoren klassifiziert, wenn die extrahierten Merkmale die erwartete Sensorfunktionalität beschreiben. Eine weitere Eigenschaft des Ansatzes ist die Bestimmung von Hypothesen durch "weiche Entscheidungskriterien". Dies ähnelt der menschlichen Vorgehensweise unter ungewissenen Umständen. Diese Klassifizierung von Messdaten, Tracks und Sensoren kann für unterschiedliche Fahrassistenzfunktionen bereitgestellt werden und bildet einen wesentlichen Beitrag zur Implementierung sicherheitsrelevanter FAS.

Abstract

Current trends in automotive industry focus on efforts to increase safety, efficiency, convenience and comfort of driving making use of intelligent Advanced Driver Assistance Systems (ADAS). An essential requirement for the implementation of such systems consists of acquiring and identifying relevant events in the driving environment. This kind of events can be acquired by sensor devices.

In order to assure required quality and availability levels for several ADAS and thus a safe driver support, a sufficient reliability level of the acquired sensor information and of its processing mechanisms and methods have to be achieved. In this sense an important aspect consists of combining the information of dissimilar sensors in order to extract a maximum of their potentialities. Sensor devices have to be combined in such a form that they can cooperate, complement and supervise each other. Sensor data fusion and Multiple Target Tracking (MTT) methods can connect different sources of information, increase robustness and redundancy and derivate "new" information according to model assumptions and hypotheses.

A further relevant aspect consists of assuring a required reliability of driving environment information even in the presence of sensor anomalies. Sensor failures and faults have to be quickly identified and if possible isolated. Fault detection and identification methods (FDI) offer different knowledge-based procedures that are based on analytical and heuristic information.

By means of the combination and further development of sensor data fusion and FDI strategies, driving environment information can be evaluated regarding its relevancy as well as correctness. In doing so a higher information reliability can be guaranteed for the implementation of ADAS. Both methods can complement and supervise each other due to they exhibit similar strategies, which will be explored along this thesis.

The main goal of this work consists of developing and validating a concept for improving quality, integrity of the acquired driving environment information. A prerequisite is the detection and to some extent the identification of sensor faults and failures that are relevant for the reliable implementation of ADAS. For these purposes several sensor data fusion and multiple target tracking methods along with fault detection and identification techniques will be investigated, adapted and further developed.

The focus of this work consists of founding a solution on how to perform the tracking of multiple targets acquired by dissimilar and to some extent redundant sensors while sensor faults and failures can be detected and identified.

Thus a synergy of MTT and FDI mechanisms will be determined. Relevant is also the level of integration of both strategies and how they can cooperate in order to highlight their advantages as well as attenuate their weaknesses.

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
ANN	Artificial Neural Networks
ARMA	Auto Regressive-Moving Average
ASIL	Automotive Safety Integrity Levels
BN	Bayesian Networks
CoG	Center of Gravity
DIDO	Data In Data Out
DIFO	Data In Feature Out
DoD	Department of Defense
DOS	Dedicated Observer Scheme
DT	Decision Trees
EKF	Extended Kalman Filter
ET	Existent Tracks
ETA	Event Tree Analysis
FDI	Fault Detection and Identification
FFDD	Fusion-based Fault Detection and Diagnosis
FIDO	Feature In Decision Out
FIFO	Feature In Feature Out
FMCW	Frequency Modulation Continuous Wave
FMEA	Failure Mode and Effect Analysis
FMECA	Failure Mode, Effect and Critically Analysis
FRPN	Failure Risk Priority Number
FTA	Fault Tree Analysis
IEC	International Electrotechnical Commission
JDL	Joint Directors of Laboratories
KF	Kalman Filter
LDW	Lane Departure Warning
Lidar	Light Detection and Ranging
MFTT	Multiple Fault and Target Tracking

MTT	Multiple Target Tracking
MTTF	Mean Time To Failure
MTTR	Mean Time To Repair
NT	New Tracks
PCA	Principle Component Analysis
Radar	Radio Detection and Ranging
RAM	Reliability, Availability and Maintainability
ROC	Receiver Operating Characteristic
SF	Sensor Failure
SIL	Safety Integrity Levels
URTF	Uncertainty Reductive Fusion Technique
USFt	Unspecific Sensor Faults

1 Introduction

1.1 Motivation

Intelligent electronic systems have been recently introduced to vehicles in order to support the driver. In the last years in the automotive industry efforts have been performed to increase convenience, comfort and safety of driving. One of the main goals consists of relieving the driver from stress situations, from some workload and also to complement his perception of the environment. Besides the attempt to assure a tradeoff among convenience, comfort and safety has been performed.

While passive vehicle safety (e.g. airbags, safety belts, etc.) has been made widely available, active assistance systems became for some years one of the main point of interest in the automotive field. Then experts claim that the current number of traffic accidents might be reduced by employing this technology. Such techniques either might predict a critical situation by warning the driver or they may start automatic procedures in order to reduce accident severities. Thus the optimal combination of convenience, comfort and safety is the primary aim of advanced driver assistance systems (ADAS). Figure 1.1 illustrates proposed ADAS in the last years by a blurred delimitation between comfort and safety. ADAS categorization as a comfort or safety function depends essentially on the quality of the acquired information in the driving environment and on properties and definitions of these assistance functions. Some relevant ADAS illustrated in figure 1.1 can be summarized as follows (see e.g. Naab [2004] and Kopischke [2000]):

- Adaptive Cruise Control (ACC): system is able to track those vehicles travelling ahead of it. It follows the vehicle in front, automatically slowing down if the vehicle in front slows down and vice versa. Current ACCs only work in free-flowing traffic conditions or on highways.
- ACC new generation: is also known as Stop and Go Cruise Control. It enables similar functionalities as ACC does, but it is extended to lower speed ranges to full stop.

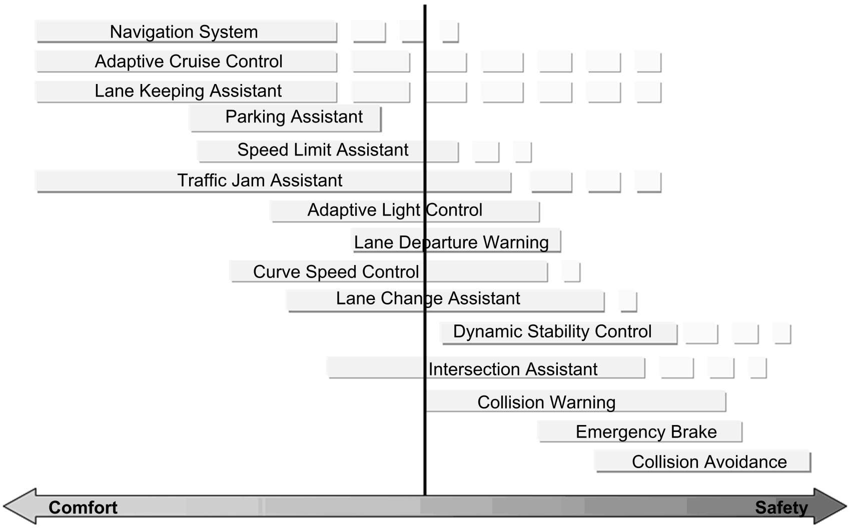


Figure 1.1: Qualitative classification of ADAS regarding comfort and safety standards.

- Park Assistant: supports the driver by parking the vehicle. The parking gap can be measured and under some circumstances the vehicle is able to park automatically.
- Lane Departure Warning (LDW): warns the driver if the vehicle cross traffic lanes unintentionally.
- Blind Spot Detection: it monitors the blind spot on both sides of the vehicle. If a moving obstacle such as another overtaking vehicle is present in the blind spot the driver is warned.
- Collision Avoidance: it helps the driver to avoid a collision by predicting the probability of dangerous situations. According to the functionï¿½s characteristics, an automatic intervention is possible.
- Emergency Brake Assistant: corresponds to the brake action with a maximal deceleration if a collision with a preceding vehicle can not be avoided by any kind of driving maneuver.

Thus an essential part for the implementation of such intelligent, safety related and to some extent autonomous systems is the acquisition of the information about events in the driving environment. By means of different sensor technologies the acquisition of the most relevant events for ADAS is made possible. An example of sensor configuration for the implementation of future ADAS is depicted in figure 1.2.

In order to assure the quality and availability of these assistance functions as well as the safe driver support, a sufficient reliability of the acquired sensor information and of its processing mechanisms and methods have to be achieved. In this context an important question is how to combine the information of dissimilar sensors in order to extract the maximum of their potentialities. These sensors have to be combined in such a form that they can cooperate, complement and supervise each other. Sensor data fusion methods can connect different

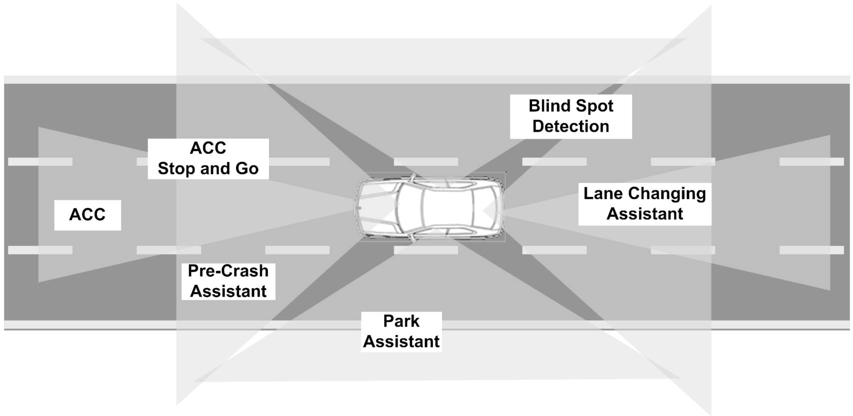
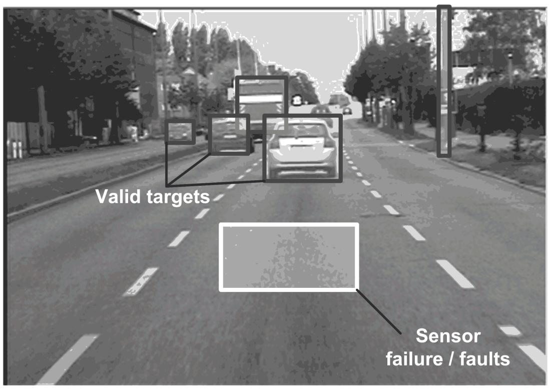


Figure 1.2: Scheme illustrating the field of view of different sensor technologies performing the detection of events in the driving environment in order to support the implementation of safety related and to some extent autonomous ADAS.

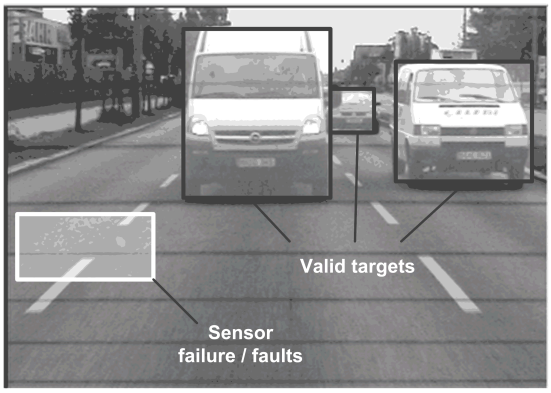
sources of information, increase robustness and redundancy and derivate "new" information according to model assumptions and hypotheses.

A further relevant aspect is how to assure the reliability of the driving environment information in the presence of sensor anomalies. Sensor failures and faults have to be quickly identified and if possible isolated. Fault detection and identification methods (FDI) offer different knowledge-based procedures that are based on analytical and heuristic information. Figure 1.3 illustrates snapshots of the real driving environment. In figures 1.3(a) and 1.3(b) frontal and rear snapshots from the point of view of the own vehicle illustrating typical valid as well as inexistent targets are depicted. Inexistent targets may be generated by sensor failures and faults. They will be explored in details in section 4.

By means of the combination and further development of sensor data fusion and FDI strategies, driving environment information can be evaluated regarding its relevancy as well as correctness. In doing so higher information reliability can be guaranteed for the implementation of ADAS as well as the uncertainty by determining the boundary between safety and comfort can be reduced. Both methods can complement and supervise each other because they exhibit similar strategies, which will be explored along this thesis and specially in chapter 2.



(a) Frontal driving environment snapshot illustrating some examples of valid and inexistent targets. The latter are generated due to sensor anomalies.



(b) Rear driving environment snapshot illustrating some examples of valid and inexistent targets. The latter are generated due to sensor anomalies.

Figure 1.3: Snapshots of the real driving environment represented from own's vehicle point of view in a typical urban area.

1.2 Goals

The main goal of this thesis consists of developing and validating a concept for improving quality, integrity of the acquired driving environment information. Prerequisite for it is the detection and to some extent the identification of sensor faults and failures that are relevant for the reliable implementation of ADAS. For these purposes several sensor data fusion and multiple target tracking methods along with fault detection and identification techniques will be investigated, adapted and further developed.

The focus of this work consists of founding a solution on how to perform the tracking of multiple targets acquired by dissimilar and to some extent redundant sensors while sensor faults and failures can be detected and identified. Thus a synergy of multiple target tracking (MTT) and fault detection and identification (FDI) mechanisms will be determined. Relevant is also the level of integration of both strategies and how they can cooperate in order to highlight their advantages as well as attenuate their weaknesses.

Synergy and level of integration of both mechanisms are achieved by investigating how they deal with acquired sensor data. Basically MTT consists of modeling the expected behavior of objects acquired in the driving environment like pedestrians, vehicles and traffic signs. On the other hand FDI takes into account the modeling of the correspondent unexpected or faulty object behavior like vehicles assuming physically improbable trajectories (e.g. vehicle direction changes in a very short period of time). That is why the terms "model" and "antimodel" are applied to MTT and FDI respectively. Deviations from expected and unexpected behavior are assigned to a grey zone, where none of the behaviors can be surely asserted.

This uncertainty may be justified by insufficient information or knowledge about the object to be modeled. Figure 1.4 illustrates schematically the synergy and level of integration of MTT and FDI strategies determined in this thesis.

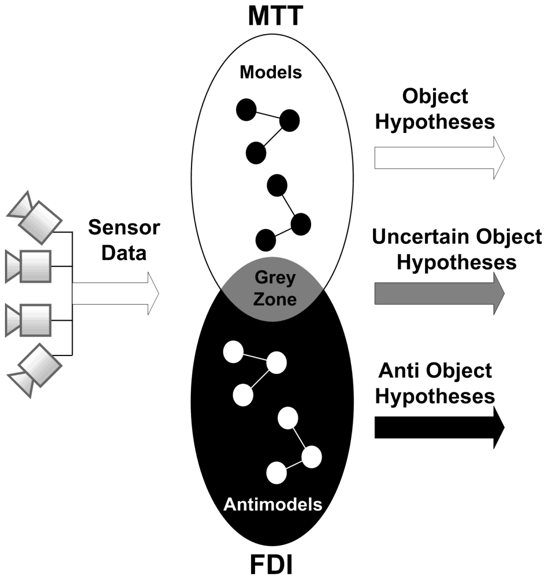


Figure 1.4: Scheme illustrating the synergy and level of integration of MTT and FDI strategies.

For the sake of completeness every model should have at least one correspondent antimodel. But due to insufficient information or knowledge about determined objects or "anti objects" could be available, only one of both types of models might be implemented. This kind of uncertainty is reflect in the grey zone as well. The outcome of the proposed approach consists of three types of different hypotheses shared as follows:

1. Object Hypotheses: describe the assumptions related to real objects in the driving environment like pedestrian, vehicles, traffic signs, etc. They are related to relevant objects that should be taken in to account by the operation of several ADAS (e.g. ACC, emergency brake assistant, etc.).

2. Uncertain Object Hypotheses: describe the uncertainty related to real objects in the driving environment. They could either describe a false alarm or a valid object.

3. Anti Object Hypotheses: are related to sensor fault or failures by performing the acquisition of driving environment information like ghost targets, sensor misalignment and blindness. In most of the cases ADAS should ignore this type of objects.

As a proof of concept the proposed approach will be online validated with aid of a vehicle prototype equipped with a configuration of distance based sensor units (radar and lidar).

1.3 Contents and Structure

Chapter 2 introduces the fundamental theory and state of the art that are applied throughout this thesis. Thereby several sensor data fusion and MTT strategies will be explored. It discusses the nature, odds and drawbacks by fusing sensor data. Furthermore different MTT configurations, algorithms and architectures will be covered. Although MTT may be performed with no fusion of sensor data, it will be considered here as a specific method of sensor data fusion. Thus the impact and relevancy of these methods for increasing information reliability will be investigated. Section 2.1.4 is devoted to the related theory and strategies for supervision and fault management methods. It summarizes the most relevant techniques involving the detection and identification of systems faults and failures. As a central part for the diagnosis of systems anomalies classification methods will be explored. Hence a qualitative comparison among three relevant methods namely decision trees, artificial and probabilistic networks is performed.

In chapter 3 the proposed concept for improving quality, integrity and reliability of the driving environment information will be derived. The proposed approach for performing sensor data fusion, multiple target tracking and the detection and identification of sensor failures and faults will be discussed in details. It covers the methods of extracting symptoms from measurements, tracks and sensor units themselves in order to obtain evidences of anomalies in the acquired data. The extraction of symptoms is based on a model based analysis varying from a white to black box approach. Thus the influence of each of these extracted symptoms or features will be determined by means of dependence models in a probabilistic network architecture following the Bayes's theorem principles.

The case study of this thesis will be covered in chapter 4 where the assembly of the vehicle prototype will be presented. Furthermore the configuration of the applied sensor units will be outlined. Section 4.2 is devoted to the measurement principles of the distance based sensor (radar and lidar) employed in the vehicle prototype. There the most relevant characteristics of these sensor units along with their weaknesses and possible anomalies will be discussed. The last topic of this chapter consists of the investigated sensor failures relevant for ADAS.

The related experiments and obtained results are discussed in chapter 5. It shows the efficiency of the proposed approach for performing the evaluation of actual driving environment information and the sensors themselves regarding their quality and reliability. Finally chapter 6 summarizes the proposed concept and discuss possible further steps within the scope of this thesis.

2 Fundamental Theory and State of the Art

This chapter deals with the theoretical basics and the state of the art of methods and algorithms directly related to this work. It discusses sensor data fusion as a connection mechanism between sensor units and applications, which supports the extraction and combination of most relevant features of events in the driving environment. Thereby several strategies, their meaning, relevancy, odds and improving potentialities will be explored. Additionally it investigates the ability to integrate sensor data fusion with different techniques in order to increase the reliability of information acquired by multiple and dissimilar sensor units. Specially MTT and FDI techniques will be explored. For it the state of the art of algorithms and architectures will be investigated. Main reason to achieve higher quality levels for the acquired driving environment events is to enable the further development of safety critical ADAS. This may be performed by testing the system against possible faults and failures. As central part for the diagnosis of systems anomalies classification methods will be explored. Hence a qualitative comparison among three relevant methods namely decision trees, artificial and probabilistic networks is performed.

2.1 Sensor Data Fusion

Sensor data fusion has been associated to a variety of techniques that use data derived from different information sources (sensors). The scope of its applications varies from real time fusion for navigation to the off-line fusion of human or technical strategic intelligence data [Rothman and Denton, 1991]. Therefore terms like:

- data fusion,
- information fusion,

- multisensor fusion and
- multisensor integration

have been correlated and cause misinterpretation. In the technical literature several attempts have been made to define fusion terms and techniques. Some authors propose the term data fusion to be used as a general one. They define this term as

"a formal framework that comprises methods and tools for the association of data coming from different sensory sources. It tries to win information of high quality, where the specific definition of high quality varies also from one application to another" [Wald, 1998].

Although the data fusion concept is easy to understand, its meaning may strongly vary. In some fusion models, data fusion is used to denote fusion of raw data. In some classic books on fusion the extended definition "multisensor fusion" is proposed [Dasarathy, 1997]. This term is defined for instance in DoD [1992] as

"the technology concerned with the combination of how to combine data from multiple (and possible diverse) sensors in order to make inferences about a physical event, activity, or situation".

In order to avoid misinterpretation the term data fusion will be assumed in this work as an overall denomination for fusion procedures as in Wald [1998]. Additionally the term multisensor fusion will be considered as in DoD [1992]. Another meaningful definition to be considered in this thesis is the one proposed by Naab [2004]:

"Sensor Data Fusion connects dissimilar and partially redundant sensor data, so that it results in a consistent representation of the environment".

In his approach he suggests the separation of sensor information acquisition and processing from the actual functions. Although some sensor configurations are associated exclusively to specific functions these procedures should be avoided. In doing so the modularity and re-usability of some system components will be increased. This means that sensor hardware, data processing and information management should be implemented independently. Figure 2.1 illustrates a sketch of the sensor data fusion dealing as an instance of integration of sensors data and as an interface to the applications.

By means of dissimilar and partially redundant sensors the events in the environment can be acquired. The sensor data fusion block is in charge of combining and processing the incoming environment information. The results are made available for different application software. According to the application purposes their outcomes will have a direct influence over the system hardware (actuators). Otherwise they will play a merely informative role (displays).

In the following sections the aim, odds, drawbacks and procedures for sensor data fusion will be explored. Additionally their influence and support to improve the reliability of the driving environment information will be discussed. This means how and what kind of features can be extracted by sensor data fusion strategies.

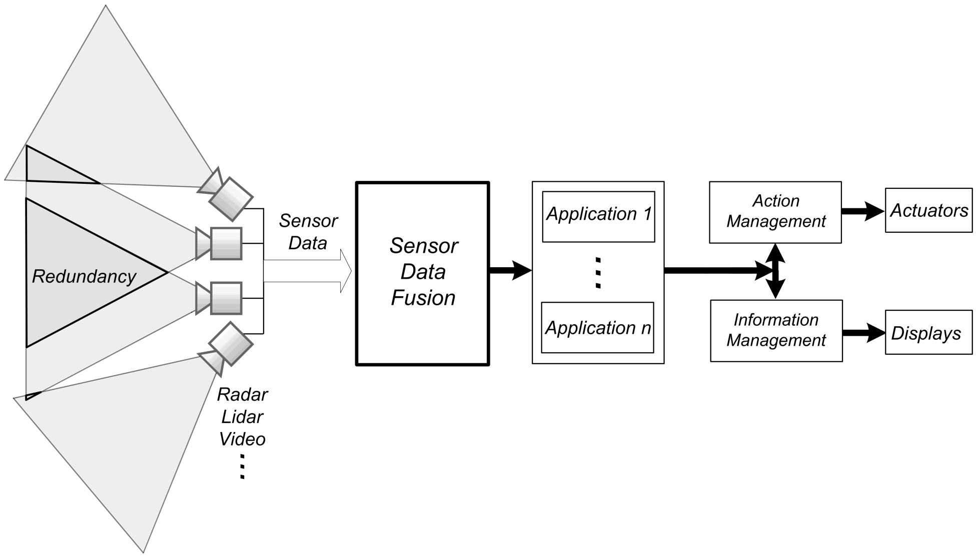


Figure 2.1: Sensor data fusion dealing as an integration platform between sensor hardware and application software.

2.1.1 Odds and Potentialities

The term sensor data fusion has been vastly discussed, but sometimes its real sense is distorted and even misinterpreted. But again, there is much more said than done in this field. Maybe this is due to the fact how most of us are viewing sensor data fusion. By concentrating on fusion as a mean of operating on data, we make the error of thinking in data fusion systems rather than in systems with data fusion capability [Kokar and Kim, 1993]. That is why one of the most important aspects is to consider the necessity of sensor data fusion systems. In the following this point will be briefly covered. First improving potentialities of single sensor systems will be enlightened focusing on the necessity of multiple sensor strategies. After that the benefits of systems using multisensor fusion will be outlined. Finally their improving potentialities will be discussed.

Potentialities of Single Sensor Systems

Improving potentialities of single sensor systems are rather a matter of application's complexity and requirements than weaknesses of sensor units themselves. Several applications have been implemented successfully relying only on the information of single sensor systems. However once applications have to meet demands of challenging constraints, single sensor systems reach their limitations, which can be categorized as follows:

- **Sensor drop outs**: applications based on single sensors or isolated sensor configurations are obviously more susceptible to sensor failures and faults. Single drop outs might affect the functionality of the entirely system.

- **Limited spatial coverage**: in some architectures, the design of a configuration with sensor scanners is impracticable. On account of this the system will have one or more sensors with a limited coverage area. This means that single sensors have finite field of vision depending on its power emission (Radar and Lidar sensors).

- **Limited temporal coverage**: in general sensors have fixed sample time, which is required for its own set-up, information preprocessing and transmission. If some breakdown takes place sensor units may require more time to process the measured information, which may imply the disregard of relevant measurements. The interlace of several sensors with different sample times should reduce these risks.

- **Limited information diversity**: although single sensor systems may acquire environment events reliably, in most of the cases they are only able to detect a specific range of object properties (position, velocity, dimensions, etc.). The type of information of single sensor systems is restricted and may affect application's performance.

Advantages of Multisensor Data Fusion Systems:

- **Robustness and reliability**: through dissimilar sensors acting out on a completing, competitive or even in a redundant form (see fig. 2.2) the final information robustness and reliability may increase. For instance this may compensate the sensor weaknesses one another. Sensor units less susceptible to weather conditions or dark environments may be applied along with units employed to detect more specific object properties (eg. dimensions, countours, etc.), which are dependent on the opposite environment conditions.

- **Extended spatial and temporal coverage**: using a consistent and well organized sensor framework the information obtained may improve spatial and temporal coverage. As mentioned before several coverage areas may be used to complement each other in order to improve sensors field of view. Aside from this the temporal coverage can also be refined by means of several sensors with different sample times.

- **Reduced ambiguity**: an intelligent environment modeling and trustable object hypotheses may help the system to reduce number of dubious information. In a multisensor system several measurements may represent the same object. By means of multisensor data fusion the measurements can be correctly associated to hypotheses representing objects providing then a quality improvement of the acquired information.

- **Increased information diversity**: information fusion of several sensors with different measuring principles may contribute to extract specific properties of acquired objects (e.g. consistency, shape, etc.). These properties may be applied to an unambiguous description of objects and thus reducing error sources by avoiding misinterpretation of actions in the driving environment.

Potentialities of Sensor Data Fusion Systems

Although the fusion of sensor data is extremely practical, most of the time straightforward and generally implies a gain of information content and quality, it may also lead to a couple of undesired effects. That is why it presents some improving potentialities aspects. In such a way fusion can induce to a dubious and even erroneous modification of the acquired information. This statement may become worse depending on the levels of fusion. Normally single sensor units contain small fusion devices already, so an expected further fusion stage would intensify this issue. The main cause for such alteration can be roughly shared as follows:

- **Information uncertainty** : the observability of reliable environment data is the ultimate prerequisite for obtaining information gain while performing sensor data fusion. These measurements are usually corrupted by noise or describe irrelevant or inexistent events in the environment. Cause for these effects can be associated to environment conditions (e.g. weather, extern interferences, etc.) or to sensor faults and failures. Once quality and integrity of the data can not be assured the whole fusion process may be endangered. This can induce to information loss or deterioration by fusing incoming data between faulty and correct sensors.

- **Modeling uncertainty**: is associated with an incorrect or incomplete description of the environment due to insufficient plant observability or knowledge. These aspects can induce to erroneous assumptions and thus affect the quality of fused information.

In order to reduce the impact of information and modeling uncertainties while fusing sensor data a storage pool configuration is generally implemented. A data pool stores the information of several fusion levels and is able to share it with different applications. In doing so applications themselves can chose what kind information is necessary in order to attend their execution requirements. This information can assume several states varying from almost unprocessed until strongly processed data. Details about different fusion levels and architecture will be explored in the remain of this section (see figure 2.4).

Considering sensor data fusion architectures as part of a distributed system, in which execution is shared again in tasks, special measures have to be taken in order to guarantee that the system hold the real time constraints. Some alternatives are the implementation of a distributed sensor data fusion system or even make use of time-triggered techniques.

Cost factors play also a very important role by planning fusion systems. First the costs for different sensor units may increase the costs of a whole project. Then the necessity of more power in order to supply the measurement units is also relevant. For instance the energy consumption within a vehicle is also a critical factor.

Above all, limitations of sensor data fusion are rather a question of planing and strategies than a matter of efficiency of the sensor data fusion method itself. Increasing the number of sensors may lead thus to a performance gain or loss depending on the fusion concept.

2.1.2 Fusion Techniques

In order to reinforce the correlation between fusion and reliability of the acquired information, techniques for fusing sensor data will be investigated. In doing so the categorization of different types of data fusion will be distinguished by outlining some procedures. Then some alternatives for fusion of sensor data will be depicted.

How to fuse Sensor Data

Due to the environment information acquired by sensor units being under some circumstances incomplete or uncertain, fusing redundant sensor data may help the system to accomplish such weaknesses. Redundant means the detection of the same object by means of multiple sensor units. This redundant information should also be combined with time varying information from each single sensor. In general three types of sensor data fusion are distinguished [Durrant-Whyte, 1988]:

1. Competitive: means the fusion of uncertain sensor data obtained from several sources. An example can be represented by a camera and a range sensor pointed at the same object measuring the same parameters. Thus the distance to the object can be obtained more accurately by means of sensor data fusion. It aims at reducing the effect of uncertain and erroneous measurements. If, for example, a sensor is uncertain related to the angle to certain obstacle it can give a rough estimate of this attribute. By means of competitive sensor data fusion this estimates may be refined by other estimates of the same parameter. Figure 2.2 shows S1 (e.g Radar) and S2 (e.g. Camera) in a competitive configuration where both sensors observe redundantly properties of the same object in the environment.

2. Complementary: means the fusion of several disparate sensors that are only able to give partial information of the environment. An example can be represented by several range sensors or camera sensors pointed in different directions at distinguished obstacles. This type of fusion tries to resolve the incompleteness of sensor data. In figure 2.2 S2 and S3 represents a complementary fusion were each sensor observes different parts of the environment.

3. Cooperative: means the fusion of distinguished sensors of which one sensor device rely on the observations of another one to make its own observations. It makes use of the information obtained from multiple sensors to describe the same obstacle. Usually sensors with distinct measurement principles are applied and thus different attributes may be acquired. In doing so not only information completeness may be increased but also sensor weaknesses may be compensate one another. A simple example is the use of one range radar to define the area of search of a camera system. Figure 2.2 depicts S4 and S5 in a cooperative configuration. Both sensors observe the same object, but the measurements are used to form an emerging view on object C.

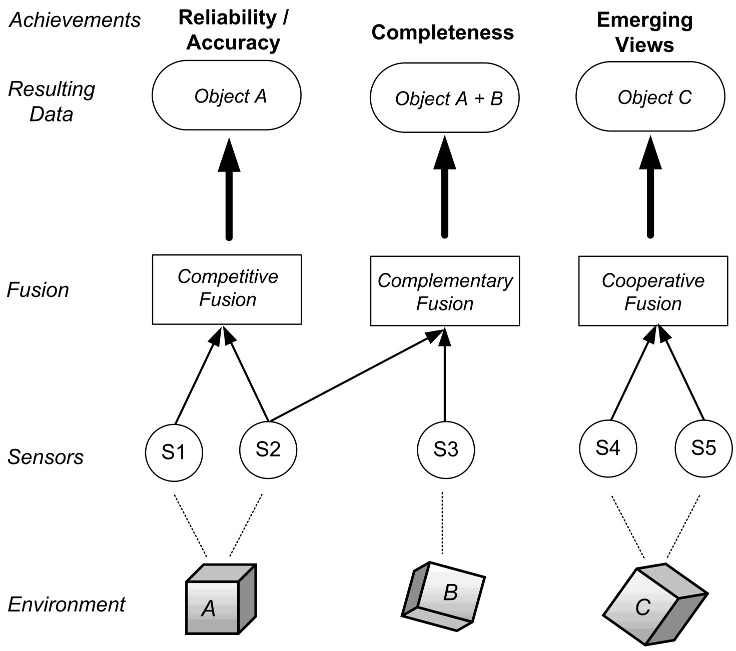


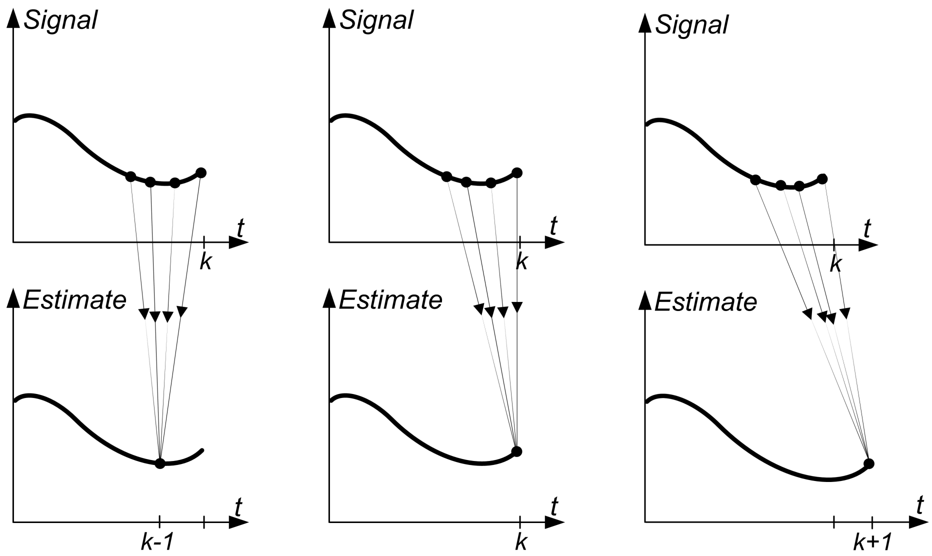
Figure 2.2: Competitive, complementary and cooperative fusion.

In most of the cases these three fusion techniques are applied in a combined form due to not being mutually exclusive. An example of such a hybrid system is the combination of video with radar and lidar sensors. In the common coverage area from all sensors, competitive fusion could take place. Long range distance based units have a narrow beam and consequently a narrow coverage area. Through complementary fusion video based units might support the information obtained from distance based ones.

Smoothing, Filtering and Prediction

Sensors provide information that can be corrupted by noise or even, in some circumstances for a period of time, provide no useful information. Therefore some procedures have to be adopted in order to minimize the noise level, to smooth out the measured signal and also to fulfill the gaps without useful measurements. That is why estimation approaches are usually applied. Estimation can be essentially classified into three different problems, namely Filtering, Smoothing and Prediction. These terms are defined based upon the time that the value output is determined for, relative to the observed data that it has access to. Figure 2.3 illustrates the estimation processes.

For instance an observation vector $\underline{y}_k$ is given, where k corresponds to the elapsed time. The goal is the estimation of the process state vector $\underline{x}_{k+m}$. Depending on the time $k+m$, the following three cases can be distinguished:



(a) Smoothing. (b) Filtering. (c) Prediction.

Figure 2.3: Estimation Processes.

1. Smoothing ($m < 0$): past values can be estimated after a series of measurements have been performed. For each instant of interest several measurements from previous, actual and following instants are used in order to estimate the value of a past process variable (fig. 2.3(a)).

2. Filtering ($m = 0$): current values have to be estimated by using an current measurement and information gained from previous measurements (fig. 2.3(b)).

3. Prediction ($m > 0$): future values are estimated based on a history of previous measurements. Prediction tasks require an adequate system model in order to produce a meaningful estimation (fig. 2.3(c)).

In general all three estimation cases contribute for an uncertainty reduction regarding acquired data. This may be possibly achieved by checking the measured information against estimated hypotheses in form of a standard deviation analysis. Assuming for example a Gaussian distribution, estimated values may be interpreted as an expectancy reference for the correspondent evaluated measurements.

Levels of Fusion

Fusion can be performed in different levels. It varies from the fusion of unprocessed data (early fusion) to a processed data level (late fusion) or even to a decision layer fusion (very late fusion). Figure 2.4 illustrates the different levels of fusion along with the respective input/output modes.

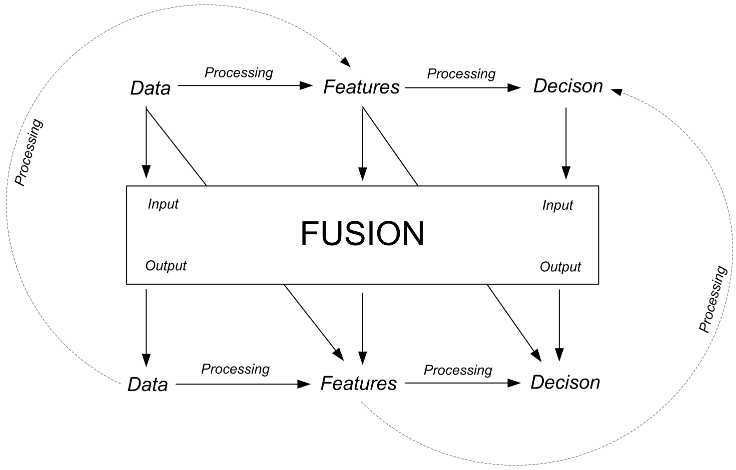


Figure 2.4: Different levels of fusion (based on Bak [2000]).

According to figure 2.4 fusion levels may be classified as follows (see e.g. [Dickmanns, 2005; Bak, 2000]):

Early Fusion: corresponds to the method that integrates data without previous processing. It comprises the following operation modes:

- **Data In $\longrightarrow$ Data Out (DIDO)**: DIDO modes are usually based on the information fusion on the signal processing domain. One example can be represented by raw measurements of different sensors that are fused competitively in order to certify the presence of objects in the environment. Raw data means in the context information without previous processing. It must be considered however that its definition is vague because it depends on the point of view of analysis. This means sensor signals are considered as raw data for the system, but as high level ones for sensors.
- **Data In $\longrightarrow$ Feature Out (DIFO)**: fusion strategies based on DIFO modes combine the information of several sensors to extract features. These features support the system to identify phenomena under observation.

Late Fusion: corresponds to the method that first extract features of the acquired data and afterward integrates it in order to obtain higher level features or final decisions. In this sense it is shared in the following modes:

- **Feature In $\longrightarrow$ Feature Out (FIFO)**: this mode combines different extracted features in order to obtain new ones. For instance shape features extracted by means of video based sensors may be combined with the ones measured by distance based sensors. It provides more specific features for analyzed objects.

- **Feature In$\longrightarrow$ Decision Out (FIDO)**: describes the method in which extracted features from the acquired data are fused providing a basis for different system applications to come to a decision. Based on the combination of different features application software like several ADAS are able to perform their tasks.

Very Late Fusion: in this stage the abstraction level is very high. Fusion is performed here with further processed data. It takes place over a decision based layer. Therefore this fusion paradigm comprises a **Decision In $\longrightarrow$ Decision Out (DIDO)** mode. DIDO mode is the last operation in the hierarchy. Here decisions are processed in previous levels and combined. For example results of different application software are combined in order to achieve more complete decision procedures.

Another important aspect that can be observed (see fig. 2.4) is the utilization of already fused data as an input for other remaining levels. This means information fusion can take place in several stages, fusion can be performed "anywhere" [Naab, 2004]. Furthermore all the presented fusion levels show odds and drawbacks. While early fusion presents positive aspects like a low information lost[1] at fusion time and properties of a data *translator*[2], it requires the use of specific models for fusion data. It provides a weak reusability of the algorithms. On the other hand late and very late fusion may suppress this reusability weakness, but data may be lost due to previous processing layers.

As mentioned before the use of a data pool configuration may minimize information loss by multiple processing stages. It stores the complete information of several fusion levels and allows the access to it by different applications at different times. An overview about data pool configuration will be covered in details in section 2.1.3 while exploring architectures for multisensor systems.

After all the choice for a specific fusion strategy is rather a matter of application software constraints than a deficiency of the fusion paradigm itself. Additionally a combination of the presented strategies is also possible and thus extracting the advantages of all methods.

2.1.3 Multiple Target Tracking

The term target tracking is associated with the processing of measurements obtained from a target in order to determine an estimate of its current or future state and thus building a target's trajectory. As mentioned previously a target state may consist for example of kinematic components (position, velocity, acceleration, etc.), dimensions (length, width, height) and consistency ones (color, structure, etc.). An important aspect is that target measurements are usually corrupted by noise, what hinders the determination of its state. Relevant measurements for multiple target tracking are usually the preprocessed ones obtained from sensor subsystems.

[1] The term *low information lost* refers to data that is fused without a strong previous processing.

[2] Fusion of information is only possible if data of different sensors have similar properties. A *translator* may transform different information types to a common basis.

It implies that measurements may already be fused before they are able to be tracked (see 2.1). For these reasons a track is a state trajectory estimated from a set of measurements that have been associated with the same target [Bar-Shalon and Fortmann, 1988].

One of the main challenges of multiple target tracking is how to deal with the data association process for uncertain measurements. Origins for these uncertainties may be, among others, random false alarms in the detection processes, disturbances due to material reflexion properties, bad weather conditions and interfering targets. Data association challenges may be shared in three groups:

1. Measurement-to-measurement: it describes the track initiation task. When one target is detected for the first time, states of the associated measurement themselves describe this new track.
2. Measurement-to-track: it describes the track maintenance or updating step. Existing tracks are confirmed by the associated measurements.
3. Track-to-track: it describes track fusion. When more tracks are generated in order to describe the same target (for example due to information incompleteness), the remaining ones have to be removed.

Figure 2.5 illustrates schematically the target tracking task along with the information fusion of two hypothetical sensors. It depicts target measurements and their uncertainties acquired in different time slices (t). These measurements represent for example a group of attributes of real targets in the driving environment. Furthermore the association process mentioned before for both track initiation and maintenance is depicted as well. In a first step measurements are associated to each other according to their properties similarity and thus initiating new tracks. In further steps measurement are associated to tracks according to their similarities as well. A certain association tolerance is determined due to both measurement and track uncertainty. The trajectory of several objects being tracked is an additional source of information for identifying events in the driving environment.

Strategies for data association in multiple target tracking process may be shared in two main categories. The first one represents measurements associated to tracks which fall within the uncertainty area or validation gate generated by existing tracks. This is the case when the obtained measurement does not contain sufficient information to describe a target. The second one is represented by tracks which are generated for every obtained measurement from a target. This approach is chosen when measurements contain enough attributes to clearly describe real targets [Bar-Shalon and Fortmann, 1988].

In this context two fundamental approaches for data association may be considered. The standard state estimation considers the most likely associations as being unique alternative to a track description. It implies that misassociations will be ignored or even considered as correct ones. On the other hand a probabilistic approach deals with hypothesis testing. It will consider by means of the analysis of different events the most suitable associations. Although the second approach seems to be the most complete one its practicability depends strongly on

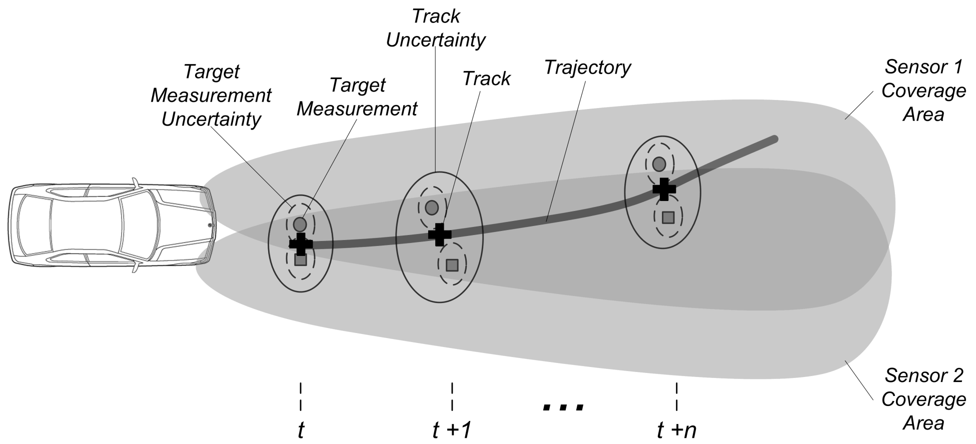


Figure 2.5: Schematic representation of the target tracking mechanism in combination with the information fusion of two hypothetical sensors. Real targets in the driving environment are described by means of sensor measurements along with their uncertainties. Association and estimation tasks of real target states and trajectories corrupted by noise (uncertainty) describe the target tracking mechanism.

the type of the extracted measurement features and on the available resources (computation efforts) as well.

These Tracks and their trajectories determined with aid of mathematical models deal as a reference for evaluating both sensor information and sensor hardware units. They describe relevant object hypotheses in the driving environment and thus supplying plausibility criteria for testing sensor's integrity. Essential premise for it is the use of dissimilar and partially redundant sensors, which is adopted in the proposed approach.

Although multiple target tracking may be performed without sensor data fusion in a one sensor system, it may be considered as a specific method for sensor data fusion as well. Following sections will cover firstly different architectures for the fusion of sensor data and tracking of multiple targets. Afterward relevant algorithms for these purposes will be explored.

Architecture for Multisensor Systems

An architecture, framework or process model for a generic sensor data fusion and multiple target tracking systems has to be able to handle with different types of information. This data may be obtained from several sensors with totally dissimilar working principles and types of targets. Examples of considered sensors are distance based (Radar, Lidar, etc.) and vision based (cameras) ones. Another important property of these architectures is their scalability. Such systems have to be able to easily deal with their own reduction or expansion. This may happen due to changes in the system configuration or by a breakdown (degradation). Frameworks that try to achieve these constraints are vastly listed in technical literature. In the following some of the most relevant ones for this work will be briefly outlined.

JDL Model: was proposed by the US Joint Directors of Laboratories (JDL) group in 1985 and later updated in Waltz and Llinas [1990]. It is considered to be one of the precursors of the fusion architectures and consists of five levels of abstraction as illustrated in figure 2.6.

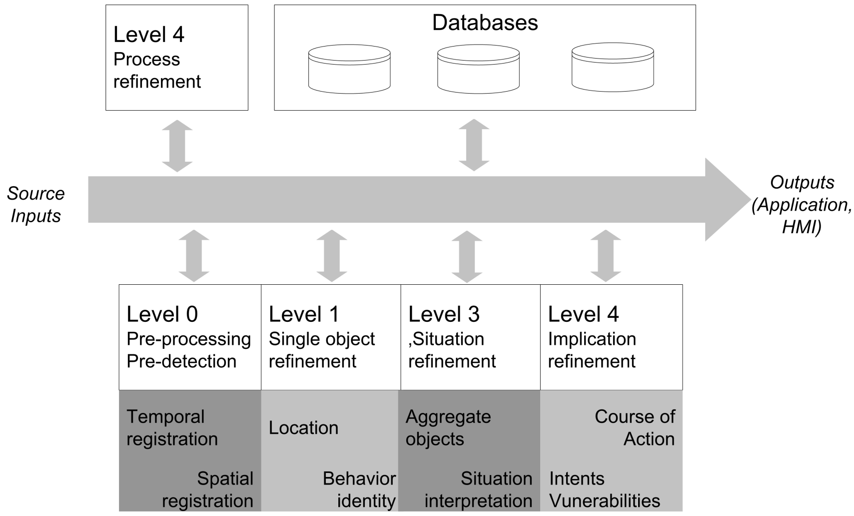


Figure 2.6: JDL Process model consisting of five levels of abstraction for performing sensor data fusion.

Level 0 is related to the sensor hardware environment. It deals with the extraction of information and is closely related to hardware signal processing. For these reasons it is generally performed in the sensor units itself. Level 1 considers an "object quantification". The common purpose of this level is to find a unique representation of all objects in the environment. Therefore real objects within the observation area are described by so called tracks, which are built by data association and state estimation techniques.

Process levels 3 and 4 treat different environment situations and decision consequences. Within these two parts the relationship between different objects is investigated with applied hypotheses. The process refinement executed in level 4 tries to optimize the process itself. This may be obtained by an adaptive data acquisition and processing. Finally the interaction with an information databases should result in the gain of information content and quality, which should be more robust and more reliable than the data obtained from each single sensor. As mentioned before some alterations have to be performed on this current architecture depending on the application in order to get a feasible data sensor fusion.

System Architecture by Luo and Kay: consists of a schematic arrangement, where subsets of sensors are connected to local fusion units while their information are preprocessed [Luo and Kay, 1992] (figure 2.7).

According to figure 2.7 local units transmit their data to superior global fusion units, which fuses the received data. For each sensor set a corresponding model is implemented. By means of sensor and error models the quality from each measurement data is determined. The sensor

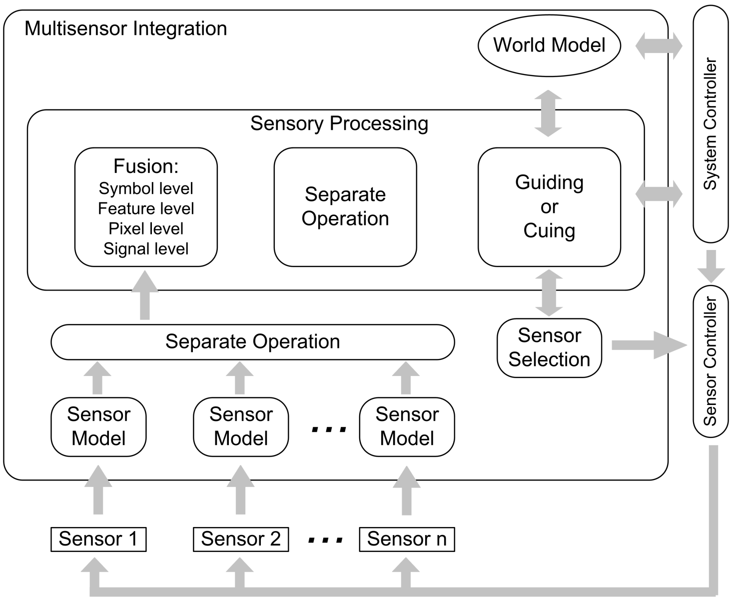


Figure 2.7: System Architecture proposed by Luo and Kay.

registration block adjusts the sensor measurements according to their spatial and temporal features before they are fused. In addition the world model block contains the description of the system states and holds the databases for a priori knowledge of the environment. In the sensor selection block the most adequate sensors for a specific application are interactively chosen. This architecture foresees integration from independent sensors.

System Architecture by Naab: aims the design of an architecture, which is able to support dynamically current and future requirements of driver assistance systems [Naab, 2004]. Thereby an integration platform has to be easily expandable and extremely flexible for new alterations. By means of a strong modularity and structuring not only an easy extension of the driving environment coverage is possible, but also certain scalability in terms of the combination with new driver assistance systems can be achieved. The flexibility of the proposed processing model was proved by the extension and further development of its architecture by de Castro Bonfim [2004].

Another important aspect covered is a system design that enables hardware independence as good as needed. Application independence within the possible limitations is required as well. Application specific parts must not affect one another and they must also be arranged as close as possible to the corresponding general applications. Several applications may use datasets from the same process level or they require it for consistence reasons. Therefore, such datasets have to be considered and managed like common resources.

Naab [2004] defines also an axiom for data handling, where the superior process levels have to have as much access as possible to the inferior ones, but not including the sensor units themselves. The effect of higher process levels over the preceding ones have to be interpreted like an attention control. This means that information obtained in the prior levels should support and also limit the area of work of the higher levels. That is why appropriated access mechanisms have to be designed.

The proposed system architecture is depicted on the figure 2.8. Its concept is based on the assumption that the obtained result can be considered like a "virtual sensor". This means that a sensor at the input of the system could contain the same architecture inside. What recurs to the point that data fusion can take place in several stages, fusion can happen "anywhere". Thereby it originates the concept of high and low level information. Sensor output contents are considered as high level from its own point of view. On the other hand sensor output is interpreted as low level information if considering the point of view of a data fusion system.

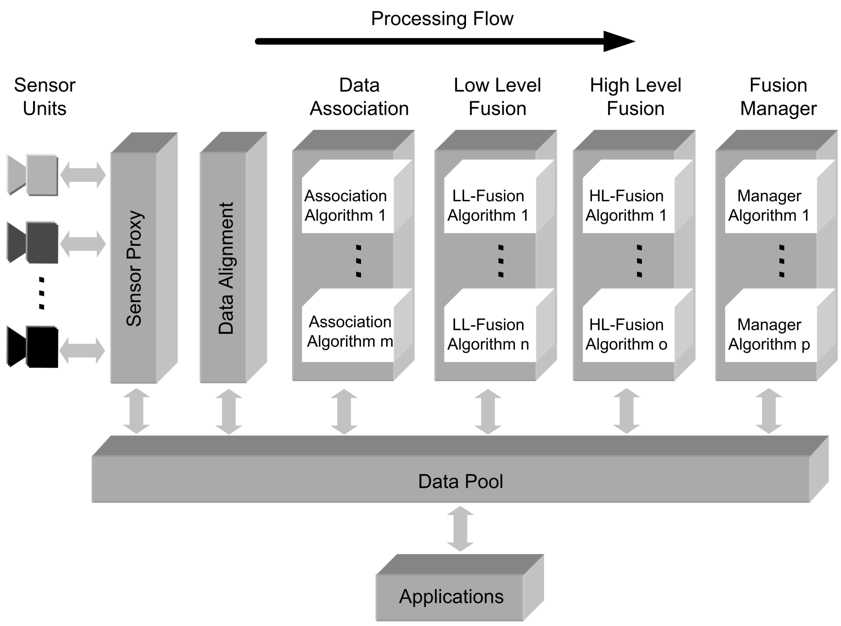


Figure 2.8: Sensor data fusion and multiple target tracking framework by [Naab, 2004].

Sensor Proxy block operates like an interface between the sensors and system. It collects and prepares the obtained information and transmits it to the next process level. Some sensors support feedback adjustment, which is executed in this stage as well.

In the ***Data Alignment*** process unit the obtained information is adjusted according to a global coordinate system. Usually sensors have their own coordinate system and measurements are represented in that form. In addition sensor uncertainties are also assigned. At this point hardware independence should be achieved.

Track definition and characterization take place in ***Data Association*** process unit. In this context they can represent vehicles, traffic lanes, road signs and so on. If they represent similar properties, they are described with the same set of attributes. Their association can be done by means of models that describe their properties and behaviors. These identical and distinguished object types can be processed simultaneously in the way of multiple target tracking. This unit performs track initiation and maintenance representing measurement-to-measurement and measurement-to-track association respectively.

In the first fusion stage or ***Low Level Fusion*** block measurement attributes are combined. For instance by calculating tracks center of gravity approach. In this unit the sensor view of the objects are not drastically changed. The data is not filtered and thus is still independent of the application.

The ***High Level Fusion*** block deals with the filtering of the measured attributes based on filter models. Attributes that cannot be measured directly are estimated using again appropriate prediction models. Required inputs, which cannot be obtained with the available sensor units, could be generated artificially by means of hypotheses as well. The same is also valid for attributes that cannot be predicted. Sometimes different applications make use of identical filter, but just with distinguished parameters. For it filter sets can be implemented, which can be application specific or for common use.

The ***Fusion Manager*** unit acts as a supervisor for the target tracking systems. Several algorithms are applied to perform track-to-track association or fusion. This reflects the last processing stage in the target tracking architecture.

In order to reduce the impact of information lost due to intensive data processing or even modeling uncertainties while fusing sensor data a ***Data Pool*** configuration is implemented. In doing so applications can access specific information, which is necessary to attend their execution requirements. Data from all processing stages is made available.

Alternatives: some alternatives to the architectures mentioned before, which can be relevant to ADAS are the 4D Approach from Dickmanns [1997], the Bayesian Network from Kawasaki and Kiencke [2004] and feature level fusion using a Multi-layer Perceptron Neural Network proposed by Bedworth [1999].

The approach from Dickmanns [1997] is divided in three spatial coordinates and time. It foresees among others the utilization of: dynamic motion models considering time delays and control outputs; 3D models for visual measurements; predicted error feedback in 4D state coordinates.

A Bayesian Network, which is also called "belief net" or "causal network", is a visualized technique for statistical dependencies between variables and also works as a probabilistic estimation machine [Kawasaki and Kiencke, 2004]. Its architecture consists of many object-detection algorithms that are called recognition IPs (Intellectual Properties). They output the target object properties such as lateral center position or width. The outputs of the recognition IPs are fused in the Total Decision IP block, where a Bayesian network is implemented. Bayesian networks working in sensor fusion and target architecture form is covered in details in Kawasaki and Kiencke [2004]

In the architecture proposed by Bedworth [1999], sensor networks are formed and afterward trained separately using the conjugate gradient optimization algorithm. Fusion occurs at the feature level by combining these networks. Bedworth [1999] shows how a mixed error criterion, which incorporates both local performance and fused performance leads to a selection of sources that is both relevant (in a local sense) and complementary (in a global sense).

Algorithms

As mentioned earlier target tracking is a state estimation task. According to the origin of the considered systems (linear or nonlinear) the type of estimation techniques may vary. Linear static systems make use of basic estimation algorithms like maximum likelihood, maximum a posteriori, least squares, minimum mean square error and so on. Linear dynamic ones apply among others Kalman and $\alpha - \beta$ filters. On the other hand, nonlinear systems are related with alternatives like the extended Kalman (EKF) or Particle filters. The later is applied not only for nonlinear, but also for non-Gaussian tracking tasks. In this work dynamic linear and nonlinear systems with Gaussian properties will be emphasized. Therefore inferences about the environment have to be performed at least by means of two types of models: a model describing the state evolution with time (system model) and one related with the noisy measurements to the state (measurement model) [Arulampalam et al., 2002]. Based on these aspects Kalman and $\alpha - \beta$ filters will be covered in details in the following sections.

Figure 2.9 situates the explored estimation algorithms for sensor fusion and multiple target tracking using the architecture proposed by Naab [2004] as an example. These algorithms can be applied in a isolated or in a combined form as well as allowing a multiple model approach.

Kalman Filter: is the state of the art of the model based dynamic filtering due to dealing with nonlinear dynamics and nonlinear models [Naab, 2004]. It handles also well with asynchronous drop outs of measured inputs and with state models with dynamic variable dimensions. The main advantage is the consistent modeling of the environment by means of an a priori or even dynamically model description of the error variances. Under some circumstances the quality rate of the models and filtered information is automatically provided by the filter. The integration of additional kinematic models can be easily performed in order to deduce other states like absolute and relative object speed, yaw rate from objects and so on.

Basically Kalman filters are an extension of least square estimates to time-varying quantities. Due to discrete Kalman filters being the basis for more complex alternatives it will be discussed in more details. Other variations like the Extended or the Unscented Kalman Filter are explored in details in Bar-Shalon [1990] and in Wan and Merwe [2000] respectively.

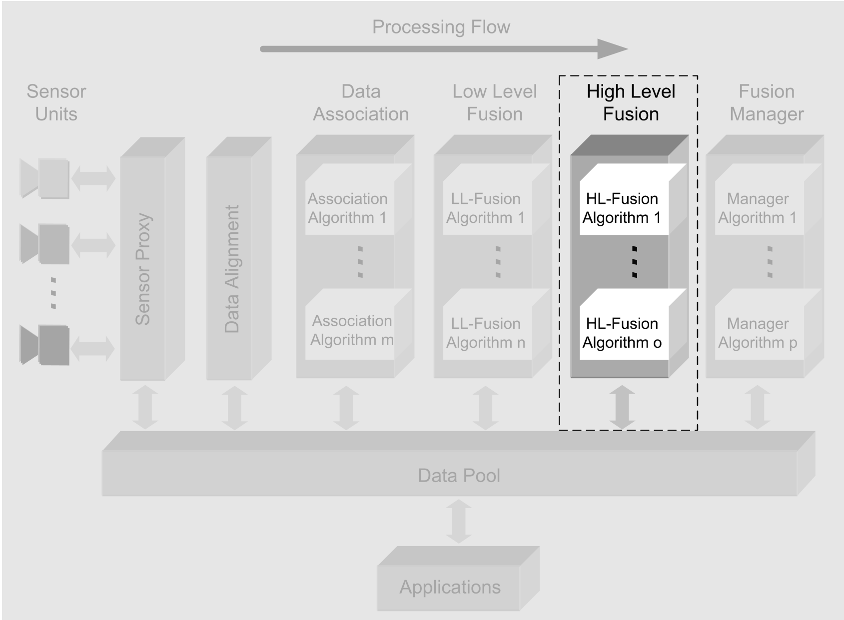


Figure 2.9: Schematic illustration for situating the estimation algorithms for sensor fusion and multiple target tracking.

The estimation problem in the discrete Kalman Filter is the attempt to estimate the state $x \in \mathfrak{R}$ of a discrete-time controlled process that is described by a the linear stochastic difference equation 2.1:

$$\underline{x}_k = \overline{A}\underline{x}_{k-1} + \overline{B}\underline{u}_k + w_{k-1} \tag{2.1}$$

supported by a measurement $y \in \mathfrak{R}$ that is described by equation 2.2:

$$\underline{y}_k = \overline{C}\underline{x}_k + v_k \tag{2.2}$$

The vector $\underline{y}_k$ represents the fusion results to a determined discrete time k. In doing so measurements are fused to the state variables. The measurement noise v_k and the process noise w_k are represented by both random variables. These variables are assumed to be independent from each other, white and with normal probability of distributions.

The square matrix $\overline{A}$ ($\overline{A} \in \mathfrak{R}$), characterized as the system or transition matrix, represents the mathematical modeling of the environment (system model). Consequently it describes the transition of the model states from the instant of time $k-1$ to the current instant k. Matrix $\overline{B}$ ($\overline{B} \in \mathfrak{R}$) is related to the control input $\underline{u}$ ($\underline{u} \in \mathfrak{R}$) to the state vector $\underline{x}$ while the measurement matrix $\overline{C}$ ($\overline{C} \in \mathfrak{R}$) represents the relationship between the measured variable and the estimated states (measurement model).

Chapter 2

The Kalman filter consists of an optimization problem. It aims at the minimization of differences between the estimated states and the measured ones making use of the error variance of measurements, estimations and processes. Its process works in a feedback control, which can smooth, filter and predict the process states (see 2.1.2). The filter predicts the process states at some time and then obtains feedback through measurements that are generally noisy. These two stages are classified as time update and measurement update respectively. They are represented by a group of equations, which govern the process. The equations for the measurement update step are described as follows by equations 2.3, 2.4 and 2.5:

$$\overline{K}_k = \overline{P_k^-}\,\overline{C}^T(\overline{C}\overline{P_k^-}\,\overline{C}^T + \overline{R})^{-1} \tag{2.3}$$

$$\widehat{\underline{x}}_k = \widehat{\underline{x}}_k^- + \overline{K}_k(y_k - \overline{C}\widehat{\underline{x}}_k^-) \tag{2.4}$$

$$\overline{P}_k = (\overline{I} - \overline{K}_k\overline{C})\overline{P_k^-} \tag{2.5}$$

At first the Kalman gain $\overline{K}_k$ is calculated. It weights the residual (difference between measured and predicted values or $y_k - \overline{C}\widehat{\underline{x}}_k^-$) according to the measurement error covariance matrix $\overline{R}$ and the a priori state estimate error covariance $\overline{P_k^-}$. Observing equation 2.3 one realizes that the more $\overline{R}$ approaches to zero ($\lim_{\overline{R}_k \to 0} \overline{K}_k$) the stronger the gain $\overline{K}_k$ weights the residual. On the other hand the more $\overline{P_k^-}$ approaches to zero ($\lim_{P_k^- \to 0} K_k$) the weaker is the influence of $\overline{K}_k$ over the residual. The a posteriori estimates $\widehat{\underline{x}}_k$ are calculated by the weighted residual and by the a priori estimates $\widehat{\underline{x}}_k^-$ predicted at time update step. Finally the a-posteriori estimate error covariance is projected as well.

The equations for the time update step are described by equation 2.6 and 2.7:

$$\widehat{\underline{x}}_k^- = \overline{A}\widehat{\underline{x}}_{k-1} + \overline{B}u_k \tag{2.6}$$

$$\overline{P_k^-} = \overline{A}\overline{P}_{k-1}\overline{A}^T + \overline{Q} \tag{2.7}$$

The previous a posteriori process state estimates $\widehat{\underline{x}}_{k-1}$ are used to project or predict the new a priori process states estimates $\widehat{\underline{x}}_k^-$. The a-posteriori estimate error covariance matrix $\overline{P_k^-}$, which consists of the expectancy of the difference between the previous measured value and the estimated value, is projected ahead by equation 2.7. $\overline{P_k^-}$ describes the track uncertainty, which is represented by ellipsoid or hyperellipsoid axes (depending on the number of hidden states) and is used as membership area in the data association block in the architecture proposed by Naab [2004]. On the other hand matrix $\overline{Q}$ represents the process noise covariance and can be employed to compensate assumptions assumed by the modeling.

The processing cycle begin again after each time and measurement update pair. This recursive feature of the Kalman filter makes it implementation easy for some applications, because it only requires the previous a posteriori value to perform the process states prediction and not a sequence of prior values.

Beside prediction, filtering and smoothing of model parameters, a very important feature of the Kalman Filter consists of its ability to fuse measurement attributes. The attributes obtained from several sensor measurements can be combined by the relation $\underline{y}_k = \overline{C}\underline{x}_k$. Where $\underline{y}_k$ represents the several measurement attributes, $\overline{C}$ the measurement matrix and $\underline{x}_k$ the state vector of the model parameters.

$\alpha - \beta$ **Filter:** is a very simple derived version of a specific Kalman filter. It is characterized by a time-invariant system model the so-called steady-state filters. Their a priori calculated parameters are not adjusted according to motion situations neither because of extern nor intern information. Anyway they should be able to predict, filter, smooth the obtained information of moving targets [Naab, 2004]. An advantage of such filters is that the elements of the gain matrix $\overline{K}$ do not have to be calculated for every duty cycle. Instead of it, the elements are calculated off-line (before the start of operation) and are valid for every identified object obtained from the prior process level of the data sensor fusion. The calculation of the element of the gain matrix considers however both the dynamic and static properties of the targets.

Both Kalman and $\alpha-\beta$ filters may imply a data error reduction or even elimination depending on the accuracy of the applied models. While Kalman filters contain fusion properties by processing simultaneously multiple measurements, $\alpha - \beta$ filters do not present fusion characteristics. The model based estimation is applied only for one measurement.

The equations that govern the process are quite similar to them used by the Kalman filter. Then the equation for the a-posteriori estimate in this case is achieved as follows (equations 2.8 and 2.9):

$$\widehat{x}_k = \widehat{x}_k^- + \alpha(y_k - C\widehat{x}_k^-) \tag{2.8}$$

$$\widehat{\dot{x}}_k = \widehat{x}_k^- + \beta(y_k - C\widehat{x}_k^-) \tag{2.9}$$

As mentioned before the elements of the gain matrix are not updated every cycle. For instance the gain matrix for a process with two states $\widehat{x}$ and $\widehat{\dot{x}}$ is described by equation 2.10.

$$K = \begin{bmatrix} \alpha \\ \frac{\beta}{t} \end{bmatrix} \tag{2.10}$$

where α and β correspond the coefficients of $\alpha - \beta$ filter with a cycle time t.

Due to this filter consists of a time-invariant system model, it can result in important simplification for practical applications. For example its markable feature is the low demand of calculation power as well as the low demand of memory space. The calculation of these two coefficients is obtained by equations 2.11 and 2.12:

$$\beta = \frac{1}{4}(\lambda^2 + 4\lambda - \lambda\sqrt{\lambda^2 + 8\lambda}) \quad and \tag{2.11}$$

$$\alpha = \frac{1}{8}(\lambda^2 + 8\lambda - (\lambda + 4)\sqrt{\lambda^2 + 8\lambda}), \tag{2.12}$$

where λ is called the target maneuvering index or target tracking index and obtained by equation 2.13:

$$\lambda \doteq \frac{\sigma_v t^2}{\sigma_w} \tag{2.13}$$

The variance of the (scalar) measurement noise is denoted as $\sigma_w^2 \doteq R$, where R represents the measurement error covariance. $\sigma_v^2 \doteq Q$ denotes the variance of the (scalar) process noise.

Alternatives: for algorithms involving the fusion of sensor data and tracking of multiple target are numerous. Besides the algorithms discussed here, others alternatives like Bayesian Networks [Kawasaki and Kiencke, 2004; Duda et al., 2001; Jensen, 2001], the Extended Kalman Filter (EKF) [Maybeck, 1979], approximate grid-base methods [Arulampalam et al., 2002], particle filters [Doucet et al., 2001], fuzzy systems [Pacini and Kosko, 1992] and [Loebis et al., 2004], neural networks [Yu et al., 2007] and [Lendaris et al., 1994] among others are applied. Due to the high diversity of algorithms of sensor data fusion and target tracking available and their indirect relevancy in this work, they will not be covered in detail. An overview of such algorithms can be achieved in the references mentioned.

2.1.4 Information Reliability

This section intends to give an overview of different strategies in which sensor data fusion supports an improvement of information reliability. As mentioned earlier sensor data fusion itself is a straightforward method that enables the acquirement of reliable information. Relying on different data fusion alternatives (competitive, complementary or cooperative) a desired quality level may be already achieved (see section 2.1.2). However an important aspect is how to evaluate it. A measure for the fused information has to be determined in order to supply further processing levels with reliable data.

In technical literature the fusion of sensor data has been also applied to diagnostic purposes. Wright and Kirkland [1995] propose the use of sensor data fusion not only to implement instantaneous diagnosis procedures, but also to learn upon trends and patterns in order to facilitate it in the future. In doing so a higher reliability of fused information may be achieved as well. Their approach tries to retrieve previous unknown knowledge implicit within the acquired data (figure 2.10). Usually the extraction of such information is nontrivial and potentially useful for diagnosis. The system semantic is presented as follows:

"given a set of facts (data) F*, a language* L*, and some measure of certainty* C*, one may define a pattern as a statement* S *in* L *that describes relationships among a subset* F_s *with a certainty* c*. According to the user imposed interest measure, a pattern* S *that is interesting and has a sufficiently high degree of certainty is called knowledge."*

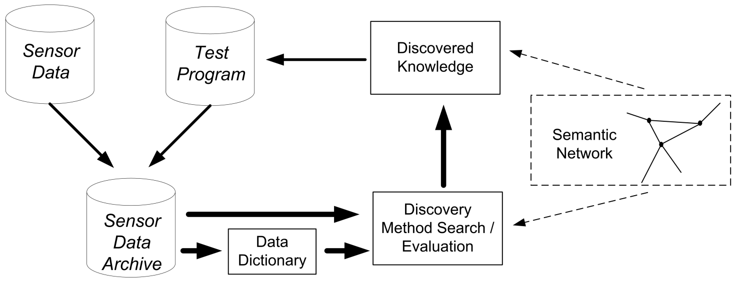


Figure 2.10: Sensor Fusion and Knowledge Discovery architecture according to Wright and Kirkland [1995].

According to figure 2.10 the system core consists of different methods that compute and evaluate patterns from fused sensor data to become knowledge to support test programs. The systems input consists of measured data stored in a central database. This database is filled by the acquired and fused information of different supervision sensors and a priori knowledge in form of test programs (or previous knowledge). The system outcome consists of the discovered knowledge that may be processed by different application software or returned to the domain knowledge (systems cyclic influence). Although the approach of Wright and Kirkland [1995] is suitable for several diagnosis applications, it shows some weaknesses in its raw format with respect to automotive applications. For example the discovery process is computationally expensive and the amount of stored data may grow drastically according to the failures to be diagnosed.

Another valuable example on how sensor fusion supports the acquisition of reliable information is the approach from Park and Lee [1993a] (figure 2.11). In their approach they propose the Fusion-based Fault Detection and Diagnosis (FFDD) system where the sensory information is modeled using Fuzzy numbers, $F-$numbers, and the Uncertainty Reductive Fusion Technique (URTF) proposed by [Park and Lee, 1993b].

According to figure 2.11 the fault detection process is shared in three stages in order to minimize fault propagation. This configuration may also support the fault diagnosis process by providing detailed information about sensor data and the state of system. The outcomes of N redundant sensors are fused by means of URTF. Once the approach deals with competitive sensor information (see 2.1.2) the system is calibrated to provide fusible results. Any infusible data, in any state, would provide an evidence of sensor failure specially by the sensor that supplies this infusible information.

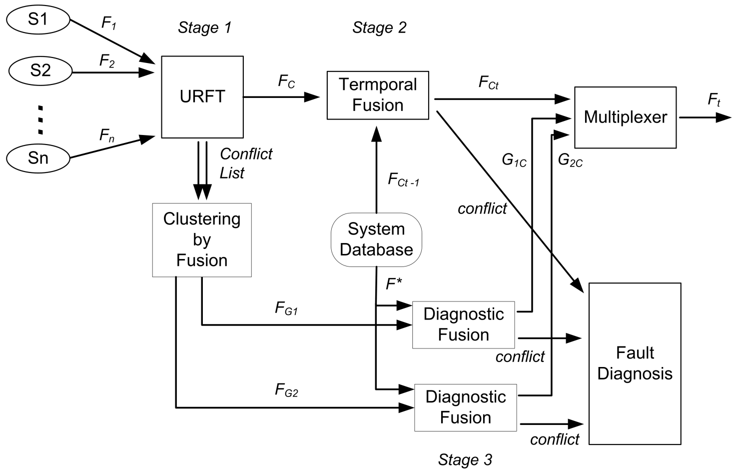


Figure 2.11: Fault detection process according to [Park and Lee, 1993a].

Stage 2 consists of a temporal fusion process. It checks time consistency against the prior information stored in the system database. This means, an extra validation is performed by means of target tracking strategies (see 2.1.3). Fusion is achieved by combining the consensus result of multisensor fusion process, F_C, and the output of the FFDD for one prior time instance, F_{Ct-1}.

Stage 3 is responsible for the Diagnostic fusion process. In this stage the results of each of the fusible groups, $\tilde{F}_{Gj}$ (for $j = 1, 2, \cdots, r$) are fused with the nominal state, $\tilde{F}^*$, which is provided by the system database. Due to URTF is used for clustering three sources of fusible outputs are possible: a) temporal fusion; b) diagnostic fusion; c) none of them. There will be at most one fusible output from the diagnostic fusion process among the "r" fusible groups. The final system output F_t is obtained by means of a multiplexer, which choose the output value out of the temporal and diagnostic fusion processes. Finally the diagnose from sensor faults is obtained by the combination of different conflict lists of infusible data in the Fault Diagnosis block by means of $F-$numbers.

Another relevant method to mention is the work from Basir and Youan [2007]. They investigate the use of Dempster-Shafer evidence theory as a tool for modeling and fusing multisensor data and performing diagnosis. A detailed description of their method is given in Basir and Youan [2007].

Although several approaches were proposed in technical literature, the necessity of methods that easily handle sensor malfunctions and improve the reliability of information in a modular and scalable form is still present.

2.2 Supervision and Fault Management

Due to the necessity of improving safety, reliability and availability of technical processes, different methods and strategies have been extensively explored. Supervision and Fault Management involve several techniques which enable monitoring of automated systems and automatic protection in cause of faults. Therefore this section intends to give an overview about these techniques which are basis for the development of this work. First of all a survey of basic tasks and definitions of supervision an fault management processes will be explored. It also involves the aspects of systems reliability, availability and maintainability (RAM) as well as safety, dependability and integrity. Then, fault detection methods will be covered. Diverse strategies from signal models plausibility till the detection with state observers and state estimation will be covered. Additionally fault diagnosis methods will be investigated.

2.2.1 Tasks and Definitions

Classical supervision and fault management methods are mainly concerned with monitoring and automatic protecting technical processes. Monitoring means the verification of measured variables considering tolerance thresholds. In case of these signals exceeding the established limits, alarms have to be interpreted by an operator in order to let him take appropriate counteractions. Automatic protections usually drive the process to fail safe states, which are normally described by an emergency shut down [Isermann, 2005].

Although classical methods are straightforward and work well if the evaluated technical process works in steady-state, it becomes more complex if the process operating point changes rapidly [Patton et al., 2000]. In some situations a graceful degradation of the system is more desirable than an emergency shut down. In this sense and among other weaknesses a more general scheme for supervision and fault management is required (figure 2.12).

According to the scheme described in figure 2.12 the implementation of advanced methods of supervision, fault detection and diagnosis is made possible. This approach is necessary in order to fulfill the following requirements [Isermann, 2005]:

- **early fault detection:** has to be achieved even for small faults with abrupt or incipient behavior. This should outperform the weak efficiency of simple threshold based supervision methods. Thereby effective counteractions like a graceful system degradation may be performed.
- **overall fault detection:** is a requirement to be achieved specially in processes closed loops. What is performed in contrast to some classical supervision methods, which allow particularly the detection of faults in output signals.
- **overall fault diagnosis:** is a premise, which involves considerably development efforts. The diagnosis of faults in the processes or part of it and other mechanisms like sensors and actuator is essential for the feasibility of safety critical systems.

- **process supervision in multiple states:** have to be assured in complex process plants. Due to operating points changing in such systems rapidly, supervision approaches have to be able to adapt themselves in the same frequency.

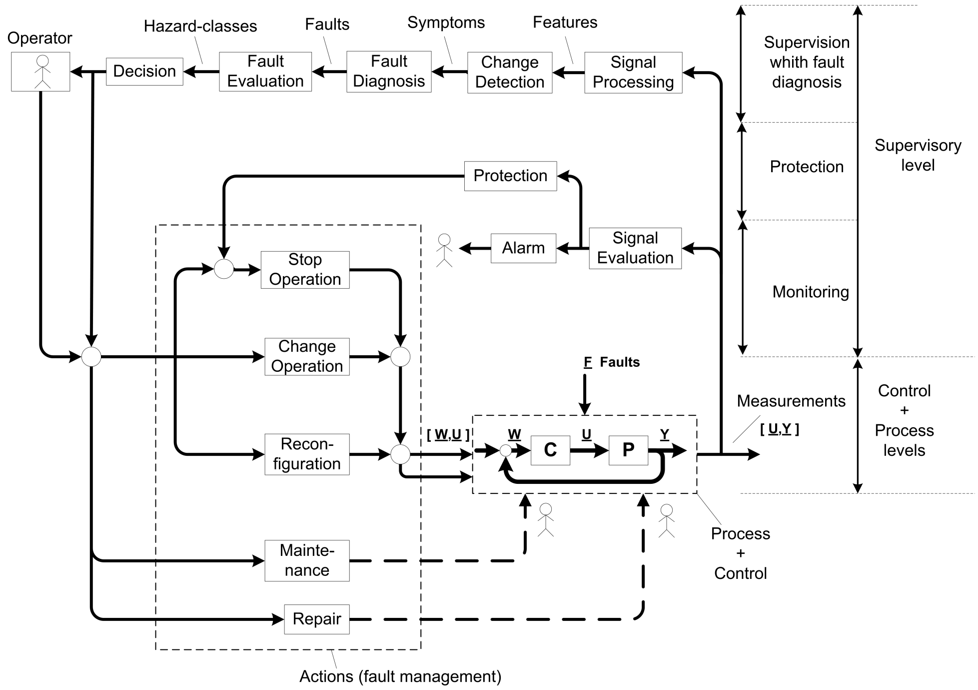


Figure 2.12: General scheme of different supervision methods with fault management (supervisory loop).

This general scheme extend the classical methods by interpreting the system deviations not only in the outputs, but also by considering input signals and intern process states. It is shared into the supervision with fault diagnosis and actions also considered as fault managements.

Supervision with fault diagnosis consists of several approaches which involve signal processing till the operator decision level. Signal processing is related with the extraction of features that are relevant for fault detection and diagnosis. Its major task is to obtain the observable and relevant features out of the acquired data, which consists of multiple redundant and, in some cases, meaningless information. Feature extraction may be achieved by means of different methods like state estimation, identification and parameter estimation or parity relations [Gertler, 1991]. The next step is the processing of these extracted features in order to extract "symptoms" or indicators for faults. Again the achievement of these symptoms may be performed by plausibility criteria by human observations (heuristic) and by modeling expected or unexpected behavior of data or actions in the environment (analytical). Symptoms are characterized by the deviation of the expected or the assertion of the unexpected behavior of the analyzed data. Therefore they are also considered as residuals.

Once symptoms are extracted from data, fault diagnosis may be performed. These analytical and heuristic fault indicators may be evaluated in order to determine the kind, size and location of faults. This task may be performed by means of classification or reasoning methods [Gertler, 1991]. After faults are identified, a classification regarding the hazards that this faults could cause to the whole system is required. According to the level of the hazards the degree of danger counter actions will be taken by the operator. This information will supply the the automatic protection mechanisms.

Another important aspect in the general scheme illustrated in figure 2.12 is the supervision of actions and fault management. Depending on the hazards level several actions may be performed:

- **safe or reliable operation:** corresponds to the avoidance of a further fault expansion by changing the operation state. Sometimes it must be driven to a system shutdown.
- **reconfiguration:** drives the system to a state in which it is activated by alternative sensors, actuators or by redundant (standby) components in order to keep the process in operation or under control.
- **inspection:** allows a detailed diagnosis by operators if the systems is not able to be driven to a safe state or reconfigured.
- **maintenance:** corresponds to all actions which have as as objective to retain an item in or restore it to a state in which it can perform the required function [EFNMS, 1998].
- **repair:** is the process to remove a fault instantaneously or at next possibility by, for instance, exchanging damaged parts.

Although supervision and fault management methods have been extensively explored and discussed in technical literature, the terminology is still ambiguous. One of the main causes for the lack of clearness is due to these methods have been applied for different areas. Fundamental terminology in this context will be presented in the remain of this section. These definition are extracted from Isermann and Balle [1997], Leonhardt and Ayoubi [1997] and Isermann [2005]. The presented terminology will be the basis adopted in this work.

*"**Fault** is an impermissible deviation of at least one characteristic property (feature) of the system from the acceptable, usual standard condition."*

*"**Failure** is a permanent interruption of a system's ability to perform a required function under specified operating conditions."*

*"**Malfunction** is an intermittent irregularity in the fulfillment of a system's desired function."*

Other important terms to be considered here in order increase the quality of technical process are reliability, availability and maintainability, the so called RAM measurements [Duma and Krieg, 2005].

*"**Reliability** is the probability of an item to perform a required function under stated conditions for a specified period of time."*

The amount of reliability of a specific system can be obtained by the reliability function (equation 2.14):

$$R(t) = \frac{n(t)}{N} = \frac{\mathit{failure\ free\ elements}}{\mathit{number\ of\ all\ elements\ at\ the\ begin\ of\ operation}} \tag{2.14}$$

which may be also expressed by an exponential failure law (equation 2.15):

$$R(t) = e^{-\lambda t} \tag{2.15}$$

where λ describes the failure rate (equation 2.16):

$$\lambda = \frac{1}{\mathit{number\ of\ all\ elements}} \frac{\mathit{number\ of\ failures}}{\mathit{time\ interval}} \tag{2.16}$$

Figure 2.13 illustrates a typical failure rate based on empirical evidences in dependence of a component lifetime.

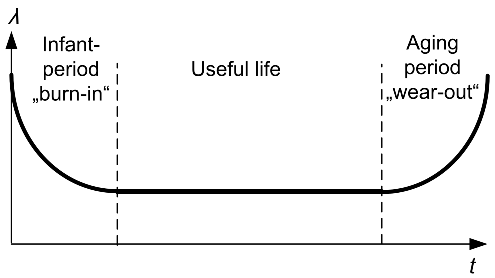


Figure 2.13: Bathtub curve (based on empirical evidences) for describing typical failure rates $\lambda(t)$ for components.

Another important measure for system reliability is the Mean Time To Failure factor (MTTF). It describes an estimate of the average operating time until a designs's or component's failure take place. This can be only considered by constant failure rates. Equations 2.17 and 2.18 describe ways of determination for the MTTF.

$$\mathrm{MTTF} = \frac{n(t)}{\frac{dn_f(t)}{dt}} = \frac{\mathit{number\ of\ operating\ elements}}{\frac{\mathit{number\ of\ failures}}{\mathit{time\ interval}}} \tag{2.17}$$

$$\mathrm{MTTF} = \frac{1}{\lambda} \tag{2.18}$$

where the MTTF is determined as the reciprocal of a constant failure rate. In order to determine the whole system reliability the disposition of the evaluated components has to be considered. The reliability of serial connected components (m), which fail statistically independent from one another, is determined by the multiplication of the reliability of each component (equation 2.19).

$$R_{tot}(t) = \prod_{i=1}^{m} R_i(t) \tag{2.19}$$

On the other hand parallel connected components, which are statistically independent as well, are calculated by the product of each component unreliability $Q_i(t) = 1 - R_i(t)$ (equation 2.20).

$$Q_{tot}(t) = \prod_{i=1}^{m} Q_i(t) = \prod_{i=1}^{m} (1 - R_i(t)) \tag{2.20}$$

*"**Availability** is the measure of the degree to which an item is in a operable state and can be committed at the start of a task when it is called for at an unknown (random) point in time. It is a function of how often failures occur and corrective maintenance is required, how often preventative maintenance is performed, how quickly indicated failures can be isolated and repaired, how quickly preventive maintenance tasks can be performed, and how long logistics support delays contribute to down time."*

By means of equation 2.21 the availability function is described:

$$Availability = \frac{time\ in\ operation}{total\ time} = \frac{\mathrm{MTTF}}{\mathrm{MTTF} + \mathrm{MTTR}} \tag{2.21}$$

where the Mean Time To Repair (MTTR) describes through equation 2.22 the average time over N failures needed to return a failed device or system to service.

$$\mathrm{MTTR} = E\{T_R\} = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} T_{Ri} \tag{2.22}$$

*"**Maintainability** is the ability of an item to be retained in, or restored to, a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance and repair."*

There are several ways to achieve the maintainability of a system even before its activation. They consist of designing the system accurately, using standardized tools and components, achieving an appropriate modularity. A manner to determine the average maintainability of a system is by means of the MTTR factor described in equation 2.22.

Assuring reliability, availability and maintainability of a system implies a direct impact on its safety. As the term safety is also applied for different areas its definition and from its related topics is still ambiguous. Safety will be interpreted in this work like in [IECTR61508-0, 2005]:

*"**Safety** is freedom from unacceptable risk of physical injury or of damage to health of people, either directly, or indirectly as a result of damage to property or to the environment."*

Another important safety requirement for safety-related systems is the level of integrity, which is defined as follows [IEC61508, 1997]:

*"**Safety Integrity** is the probability of a safety-related system satisfactorily performing the required safety functions under all the stated conditions within a stated period of time."*

Common methods for achieving system safety are listed in technical literature like: fault avoidance, removal, tolerance, detection and diagnosis and additionally automatic supervision and protection. Fault avoidance and fault removal are measures that may be achieved at the system design phase. Useful techniques for their determination are among others: hazard and risk analysis, failure mode and effect analysis (FMEA) and fault tree analysis (FTA). Next a brief overview about these design techniques will be given. Online techniques like fault detection (section 2.2.2) and diagnosis (section 2.2.3) and will be covered later on.

Hazard and risk analysis are basic procedures to determine the safety integrity of a system. This kind of analysis provide the first insight in the technical process behavior that could lead to situations of potential danger causing accidents, injury, death, environmental or material damage. According to the level of hazards, depending on its frequency and consequence, risks can be determined. There are several qualitative and quantitative methods to determine the risk factor in a system. One example of a quantitative risk classification is showed in table 2.1.

Frequency	**Consequence**			
	castastrophic	critical	marginal	negligible
frequent	I	I	I	II
probable	I	I	II	III
occasional	I	II	III	III
remote	II	III	III	IV
improbable	III	III	IV	IV
incredible	IV	IV	IV	IV

Table 2.1: Risk Classification of Hazards or Accidents [IEC61508, 1997].

A detailed description of different methods to assign the frequency and consequence of hazards and risks can be found in [IEC61508, 1997] and in its revisions through the following years by the International Electrotechnical Commission (IEC). The interpretation of risk classes are described as follows in table 2.2:

The assertion of the frequency and consequence of hazards or accidents is performed in different areas like automotive, aircraft and nuclear. Thus specific standards were developed. The standard IEC 61508 ([IECTR61508-0, 2005]) defines four levels of safety performance for a safety function called *Safety Integrity Levels* (SIL). SIL1 describes the lowest level and SIL4 the highest one. These levels describe a measure of likelihood for the system safety and comprise,

Risk Class	Interpretation
Class I	intolerable risk
Class II	undesirable risk, and tolerable only if risk reduction is impractical or if the costs are grately disproportional to the improvement gained
Class III	tolerable risk if the cost of the reduction would exceed the improvement gained
Class IV	negligible risk

Table 2.2: Interpretation of risk classes [IEC61508, 1997].

for example, the failure probability per hour, operating hours per failure and operating years per failure. As mentioned earlier safety requirements vary according to the evaluated area. Therefore specific areas adopt SIL standard respecting their demands.

In order to support the hazard and risk analysis FMEA is usually performed. FMEA is a analysis method that evaluate components, their functions, failure modes and the cause of system failures. It performs the following tasks:

- it evaluates effects and sequences of events caused by each identified failure mode.
- it determines the significance or criticality of each failure mode as well as the correct performance of the system and the impact on the availability and safety of the related process.
- it classifies the identified failure modes according to their ability to be among others, detected, diagnosed, testable.
- it estimates the measure of the significance and probability of failures.

FMEA allows the identification of weaknesses on the system in both early and late development stages. Its analysis structure results in tree-like format, where failure modes with regard to components and the respective effects on the system. Figure 2.14 shows a basic structure for FMEA procedures.

Basically FMEA procedures can be shared in four steps [Eschermann, 2004], which are described as follows:

1. Break down the system into components.
2. Identify the functional structure of the system and how the components contribute to functions.
3. Define failure modes of each component:
    - new components: refer to similar already used components
    - commonly used components: base on experience and measurements

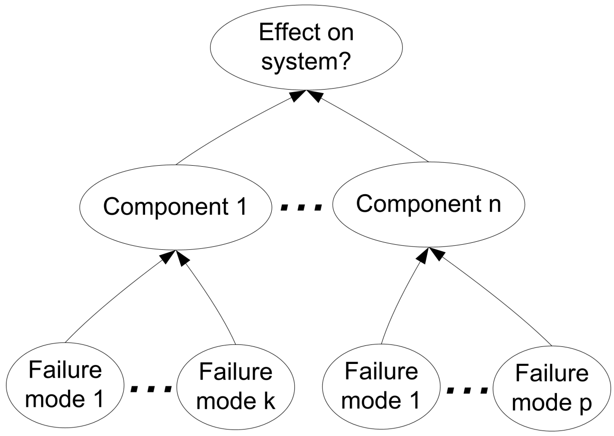


Figure 2.14: An example of a Tree-like structure for FMEA procedures.

- complex components: break down in subcomponents and derive failure mode of component by FMEA on known subcomponents
- other: use common sense, deduce possible failures from functions and physical parameters typical of the component operation

4. Perform analysis for each failure mode of each component and record results in a table worksheet. One possible form to describe the results of a FMEA of a whole system is outlined in table 2.3.

Component	Failure mode	Failure causes	Failure effect on unit	Failure effect on system	Counteraction
...	⋮	⋮	⋮	⋮	⋮

Table 2.3: An example of a FMEA result's table [Isermann, 2005].

Once the formalized failure mode and effect analysis for the whole system is performed, the most critical components can be systematic evaluated. One established technique to execute such a task is the so called Fault Tree Analysis (FTA). FTA is a top-down approach that starts with a potential undesirable event at the top of its structure and determines all the paths in which it can take place. It consists of a graphical representation of the logical structure showing the relationship between a top event an all its probable causes. By means of this technique countermeasures can be developed in order to reduce the probability of the undesired event. Figure 2.15 illustrates an example of a generic FTA.

Based on figure 2.15 the probability of failure from subsystems and the whole system can be calculated. This is performed according to the occurrence probability of faults in the specific components combined to one another. The type of connection specifies if one component fault is enough to generate a subsystem or a system fault (OR gate) or a group of components have to fail simultaneously (AND gate).

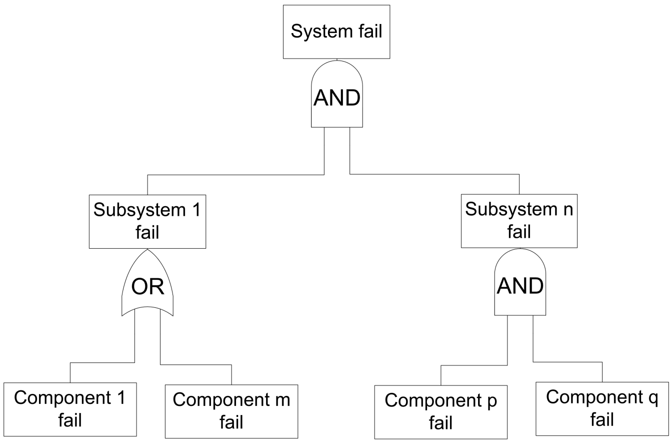


Figure 2.15: An example of a generic FTA.

Several methods has been supporting the achievement of system safety in its design phase. Although the ones covered in this section are the most common applied techniques, there are a lot of alternatives to and extension of these methods. For instance instead of achieving a qualitative statement with FMEA the Failure Modes, Effects and Critically Analysis (FMECA) enables a quantitative statement. Thereby, a risk of operation can be determined by calculating a Failure Risk Priority Number (FRPN) [Sharma et al., 2005]. Among several combination and extensions, different risk classification of hazards and accidents may be adopt (see table 2.1). A detailed explanation of the methods briefly covered in this section and variations and extensions of them for system safety can be found in IEC61508 [1997], Sharma et al. [2005], etc.

2.2.2 Fault Detection

In the previous section strategies for avoiding and removing faults in the design phase were presented. Another important task is how to detect it during system online operation. This challenge is what the following section is about. The scope from fault detection methods involves several techniques. It enfolds among others the detection of faults by means of signal models, process identification, state observers and state estimation methods. Depending on system requirements simple or complex fault detection with limit checking is satisfactory.

First of all this section will consider the basic aspects for fault detection. Focusing on the expected and abnormal behavior of systems and components techniques for modeling processes will be briefly covered. Afterward an overview of some relevant fault detection methods for this work and alternatives will be given. Finally a comparison among these relevant strategies exploring their assets and drawbacks will be discussed.

Basic Aspects

The most basic assumption by detecting faults is that the operation of a system and its components should not drastically drift from the expected behavior. According to system requirements a simple allowed range definition for signal and variables may be sufficient. As soon as such faults are detected the most common countermeasure is to drive the system to a shut down state. Although the system reliability will be increased, its availability will be drastically affected. In order to handle with these aspects, more elaborated techniques have been applied. Such techniques consist of a combination of knowledge-based with limit checking strategies. Key for this "intelligent" systems is the use of redundancy (figure 2.16).

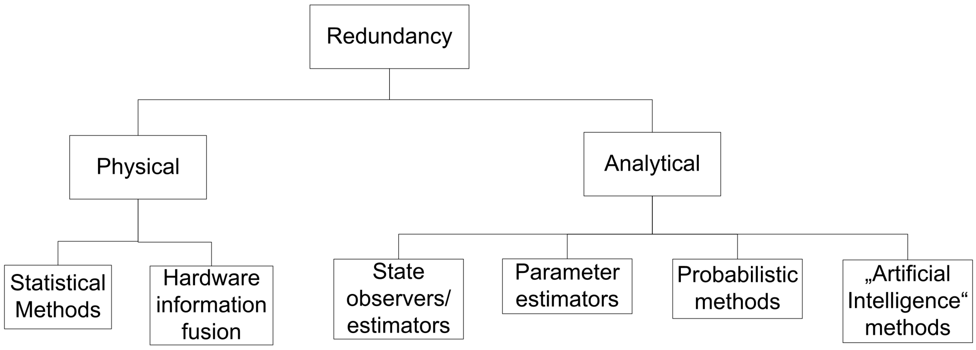


Figure 2.16: Redundancy scheme.

Redundancy is defined as the availability of more than one instrument or resource in a unit in order to perform a determined operation [EDINIEC50-191-15, 1994]. Redundant instruments or resources provide the system with reference criteria for plausibility or even for recovery procedures. This can be performed by means of residuals. In fact residuals consist of the comparison of the information calculated by redundant systems. Deviation between the results of two or more redundant devices will determine the amount of the residual and consequently the intensity of a fault.

A general structure for redundancy and how it may be achieved is illustrated in figure 2.16. Physical redundancy consists of the use of multiple and independent instruments (hardware) with dissimilar infrastructure in different places. Residual generated by means of physical redundancy are usually obtained by the difference of measured or preprocessed signals, easy to be implemented and the level of reliability regarding fault detections is considerably high. On the other hand costs involved in this approach are relatively high in order to achieve a reliable statement about fault occurrences. Common methods used for generating "physical residuals" are among others statistical approaches or the fusion of the hardware information [Betta and Pietrosanto, 2000].

Another way to generate residuals through redundancy methods is the analytical approach. It allows an explicit derivation of the maximum possible number of linearly independent model-based systems for consistency check [Leuschen et al., 2005]. Analytical redundancy exploits the system observability to allow the creation of a set of test residuals [Chow and Willsky, 1984]. These residuals enable the detection of any deviation from the static or dynamic behaviors of the model in real time. They are always defined as differences between measured and expected variables. Their complexity and reliability depend on modeling techniques and on comparison strategies adopted [Betta and Pietrosanto, 2000]. In figure 2.16 some examples of methods for implementing analytical redundancy are depicted. Common practice approaches to the analytical generation of residuals are described as follows:

- **value checks:** is performed by comparing acquired data with its expectations, which are calculated by a system model.
- **signal comparison:** consists of matching measured output signals with signals estimated by a nominal model of the system in case of the real system and its nominal model are supplied with the same input.
- **parameters comparison:** is performed by comparing parameters of the nominal system model with the ones of a continuously estimated actual model.

Although analytical redundancy is a straightforward method, the detection of faults in real time is mostly performed by means of a combination with physical redundancy. Additionally to both methods, the use of heuristic information to the residual generation is applied as well. This consists of qualitative data obtained by human expertise in the form of special noises, smells, vibration, etc, which may be used to determine among others threshold values. Furthermore statistical data achieved from experience with similar processes can be added [Isermann, 1997].

As mentioned before redundancy is the key for the generation of residuals, the so-called fault symptoms. Requirement for the implementation of redundant systems is the utilization of models that describe all types of physical, analytical and heuristic redundancy. Figure 2.17 gives an overview about modeling strategies, which may be implemented for the generation of residuals as well as for the estimation of process states and parameters.

According to the information available about the environment to be described, two strategies may be applied: theoretical and experimental modeling. Theoretical approach is based on fact that physical laws are known. In some circumstances model structure and parameters are known as well. On the other hand, an experimental approach focuses on the measurable input signals and assumptions of a model structure. These input signals are used as a source of information to build new and to validate assumed model structures. Model parameters may be determined in the same way. Outcome of these modeling approaches can be shared in four main classes:

1. White-box: is characterized by its "transparency". In this sense the relationships between physical laws and parameters in this type of models are completely traceable. All necessary information to build a model is available. White-box models are often described by linear and nonlinear differential equations. Although this type of models is easy to understand and implement, it is sometimes only able to describe the environment partially.
2. Light-grey-box: is considered as a "diffuse transparency", where most of the necessary

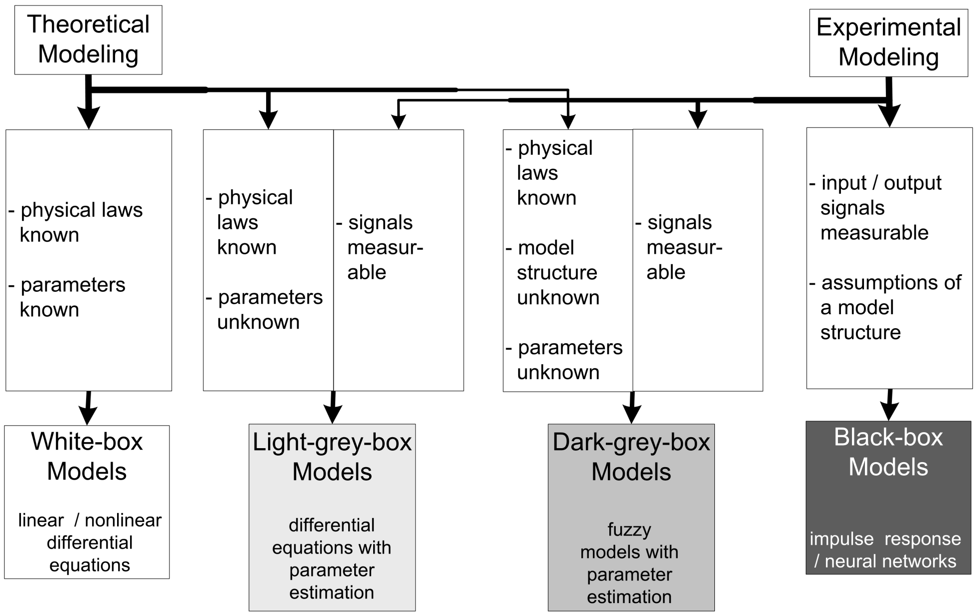


Figure 2.17: Modeling Strategies.

information is available, but parameters and physical laws relationships are still traceable. For instance physical laws are well-known, but model parameters are extracted from the input signals. Light-grey-Box models are usually described by differential equations with parameter estimation techniques. Here theoretical and experimental modeling are combined.

3. Dark-grey-box: is related with the models where most of the information is unavailable. Basically only the physical laws which describe the evaluated environment are familiar. The entirely model structure and parameters have to be deduced from input signals. Dark-grey-boxes may be described by fuzzy models with parameter estimation. Here both theoretical and experimental modeling techniques are combined as well.

4. Black box: is characterized by the lack of a priori information about the evaluated environment. Physical laws are barely known and assumptions for model structures have to be done based basically on input signals. Furthermore the relationship between model parameters and physical laws are no longer traceable. Although this type of models is complicated to comprehend and the amount of input signals to discover relationships and model structures are huge, it may drive to a more complete description of the evaluated environment. It can, in some cases, describe strong nonlinear events that are hardly modeled with mathematical equations. It may also approximate the system to the human form of figuring out the actions in the ambient. Black-box models may be achieved by impulse response analysis or neural networks.

These four model classes can be applied to different layers on the system. For example, it can vary from low level signal models to the high decision layers. The following section will cover some technique for achieving fault detection by means of this different model classes.

Techniques

Over the years several techniques for detecting faults have been investigated and developed. Figure 2.18 gives an overview about it. According to the system requirements regarding its reliability and availability and the type of on-hand information, different procedures have been applied. In this section some relevant techniques for the development of this work will be discussed. Other valuable techniques will be either commented or referenced to literary sources where they are described in detail.

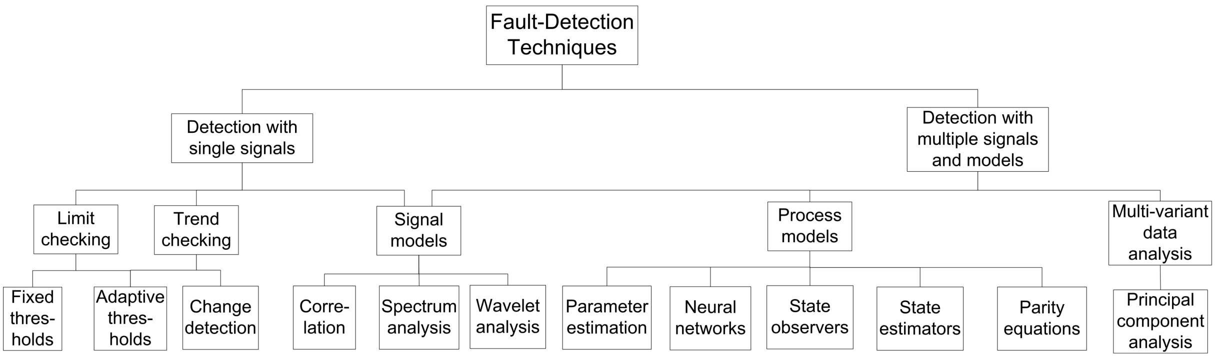


Figure 2.18: Fault Detection Techniques.

Fault detection techniques are basically shared into the categories of detection with single signals and the one with multiple signal and models. The decision for the implementation depends strongly on the type of information available. In case of fault detection mechanisms being implemented in a low level layer, single signals or multiple signals approach is applied. On the other hand if physical laws, model structures and parameters are well-known, different type of models are used (see figure 2.17).

Detection with single signals consists of limit or trend checking and signal models. The first two methods are the most basic approaches. They involve the investigation of signals by means of fixed or adaptive thresholds and by monitoring behavior changes. The determination of fixed thresholds is usually based on the a priori knowledge of the operator or by means of statistical analysis. In case of the component operating point changing rapidly, adaptive decision thresholds may be applied. Different procedures for calculating threshold values like the detection of jumps in the mean, fuzzy strategies and likelihood ratio test are treated in Hiifling and Isermann [1996] and Basseville [1988].

The model based method is enfolded in the approach with single signals as well. Considering the cyclic behavior of many processes and their resulting periodic signals along with their external random perturbances, signal models may be designed. These models act as a form of analytical redundancy in order to generate fault symptoms. In technical literature several methods to build signal models has been applied. Examples are bandpass filtering ([Potocnik et al., 2005]), short-time Fourier analysis ([Satish, 1998]), ARMA[3] parameter estimation ([Rougee et al., 1987]), etc. Figure 2.19 illustrates the principles of fault detection with signal models. It shows the signal flow of a common system along with the scheme for fault detection with signal models. It also depicts that all system components including detection mechanisms themselves are susceptible to faults. The impact of process and signal model faults may be reduced if they can be modeled in the form of uncertainties. Thus the remaining faults (actuators, sensors and non-modeled process faults) can be determined by comparing the features extracted from a modeled signal with reference values. These references also called normal (accepted) behavior can be obtained among others by operator's a priori knowledge and reference (redundant) sensors. Depending on the intensity of the discrepancies between extracted features and reference values, these fault symptoms can give evidences to locate and isolate the faults. The same principle applies to the analysis of multiple signals.

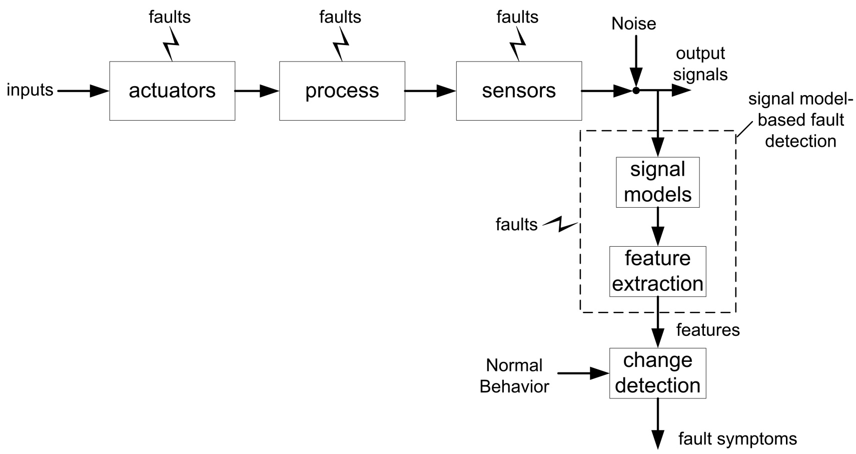


Figure 2.19: Fault Detection with Signal Models.

Strategies for model-based fault detection of processes have been also explored extensively in technical literature. The main goal is to detect unacceptable behaviors in sensor, actuators and processes by means of correlations among the acquired measurements. These correlations are described by applying modeling strategies varying from white-box to black-box models (see figure 2.17). The general scheme for model-based fault detection proposed by [Isermann, 1997] is depicted in figure 2.20.

Analog to the detection of faults with signal models, symptoms will be obtained by comparing features extracted from variables determined by models. Features are considered as residuals r (estimates deviation), model parameter estimates Θ or state estimates $\hat{x}$. Thus the key for detecting faults by process models is the choice of modeling and feature extraction. Modeling

[3] Auto Regressive-Moving Average (for details see Krishnaiah and Rao [1988])

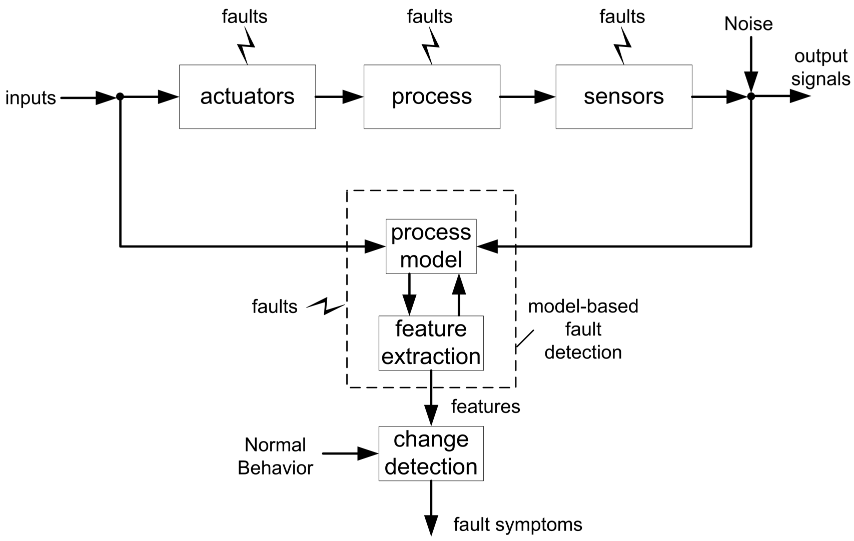
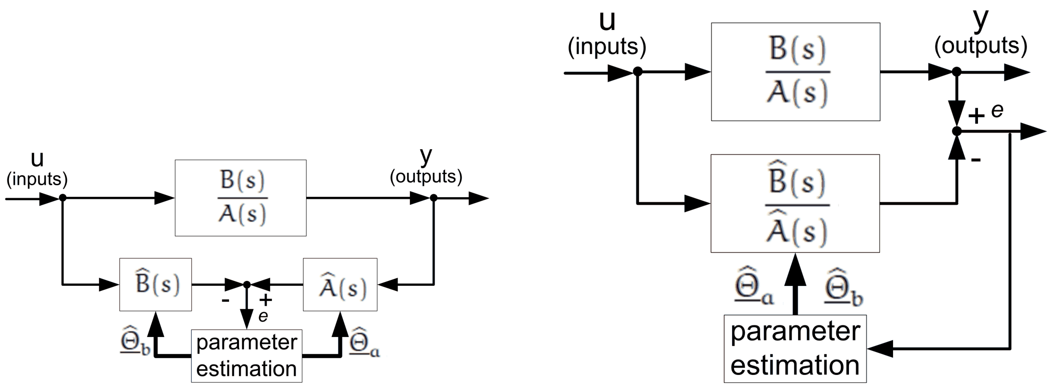


Figure 2.20: Fault Detection with Process Models.

depends on the type of the information available and the feature extraction may be performed by means of parameter and state estimators, state observers, neural networks and parity equations.

Parameter estimators belong to light-grey-box model classes and it is applied if process parameters are completely or partially unknown. They can be achieved by means of equation errors or output error methods [Isermann, 1992]. Figure 2.21(a) shows the scheme for the parameter estimation based on equation errors.



(a) Equation error.

(b) Output error.

Figure 2.21: Model structures for parameter estimation [Isermann, 1992].

Basis for the parameter estimation by means of equation errors is the model description through difference equation (see equation 2.1). It may be rearranged as follows:

$$y(t) = \underline{\psi}^T(t)\underline{\Theta} \quad (2.23)$$

$$\underline{\Theta}^T = [a_1 \cdots a_n \;\; b_0 \cdots b_m] \quad (2.24)$$

$$\underline{\psi}^T(t) = [-y^1(t) \cdots - y^n(t) \;\; u(t) \cdots u^m(t)] \quad (2.25)$$

Thus the definition of equation error (2.26) for parameter estimation is performed by rearranging equation 2.23. It is based on the difference between input and output signals by means of the estimated parameters (see figure 2.21(a)).

$$e(t) = y(t) - \underline{\psi}^T(t)\underline{\Theta} \quad (2.26)$$

A key point for estimating the modeled parameter is minimizing equation 2.26 such as in error free case $e(t) = 0$. There are several methods like least square estimate, α-β and Kalman filters that can be applied to calculate the desired parameters, the error intensity as well as the minimization of $e(t)$.

Output error methods will compare the signal outputs obtained from the processes with the ones calculated by means of the process model and estimation of parameters (figure 2.21(b)). It enables the extraction of residuals for fault detection. The main drawback of this approach is no possible direct calculation of the parameters Θ due to the nonlinearities of calculated residual e (error) in these parameters. Parameters have to be determined by extensive computational efforts in calculating numerical optimization algorithms.

Fault detection with state estimation and observers is performed by extracting the residuals between the hidden process states and the modeled ones. By analyzing linear processes they can be described in state-space form as follows (see also equations 2.1 and 2.2):

$$\underline{x}_k = \overline{A}\underline{x}_{k-1} + \overline{B}\underline{u}_k + w_{k-1} \quad (2.27)$$

$$\underline{y}_k = \overline{C}\underline{x}_k + v_k \quad (2.28)$$

Prerequisite for detecting faults with state estimation and observers is that both model structure and all process parameters $\overline{A}$, $\overline{B}$ and $\overline{C}$ are known (White-Box properties). Hidden state variables can be reconstructed based on the measured inputs and outputs (see also equation 2.4):

$$\hat{\underline{x}}_k = \hat{\underline{x}}_k^- + \overline{K}_k(y_k - \overline{C}\hat{\underline{x}}_k^-) \quad (2.29)$$

$$\underline{e}_k = y_k - \overline{C}\hat{\underline{x}}_k^- \quad (2.30)$$

where $\underline{e}_k$ means the residual (error) between the hidden process states and the modeled ones at time k and $\overline{K}_k$ describes the weighting residual matrix. Based on the previous relationships faults can be detected basically by means of three main approaches [Isermann, 1997]:

1. Dedicated observers for multi-output processes:
    - **Observer excited by one output:** one single input supplies one single observer, while the remaining outputs are reconstructed and measured and compared with the measured outputs [Clark, 1978b].
    - **Kalman filter excited by all outputs:** if a fault occurs the residual changes its property of zero mean white noise with known covariance. By means of hypotheses tests these kind of faults can be detected [Mehra and Peschon, 1971] and [Willsky, 1976].
    - **Bank of observers excited by all outputs:** faults are also detected by means of hypotheses tests if observers are designed for a determined fault signal [Willsky, 1976].
    - **Bank of observers excited by single outputs:** multiple sensor faults can be detected if multiple observers are applied to single inputs (Dedicated Observer Scheme - DOS) [Clark, 1978a].
    - **Bank of observers excited by all outputs excepted one:** one sensor output is supervised while the remaining ones drive each observer [Frank, 1987].
2. Fault detection filters for multi-output processes: are implemented by designing the weighting matrix $\overline{K}$ in a way that it becomes sensitive to a set of specific faults ([Beard, 1971] and [Jones, 1973]).
3. Output observers: are an alternative if the determination of the hidden states is not of primary interest. Analog to the parameter estimation by means of output errors, output observers performs the residual calculation through linear transformations [Patton et al., 2000] and [Tsui, 1993].

Another alternative to detect faults is the approach with parity equations. As performed with state estimation and observers a model of the process will be built. The determination of process hidden states is not of primary interest, but its outputs. These will be compared with the sensor outputs in order to determine the residuals and so extracting potentially fault symptoms. Different variants for their implementation are covered in Omana and Taylor [2007], Isermann [1997] and Isermann [2006]. Although parity equations are a straightforward method, it requires that process parameters are known, its behavior time-invariant and not susceptible to strong disturbances.

Principal Component Analysis (PCA) offers an interesting alternative for fault detection when the obtained measurements are highly correlated but only a small number of events (faults) drive the system to an unexpected behavior. PCA is a method of identifying patterns in data, and so highlighting their similarities and discrepancies [Dunia et al., 1997]. Usually PCA models are linear, static and based on processes normal operations. Several alternatives have been explored in technical literature like in [Smith, 2002], [Isermann, 2005], [Liang and Wang, 2003], etc.

2.2.3 Fault Diagnosis

Once fault symptoms are detected, the next step consists of determining fault properties like type, intensity, location and detection time. These are the tasks of fault diagnosis methods. The main challenge by diagnosing faults is how the knowledge obtained through fault detection techniques can be described. In technical literature several techniques for fault diagnosis have been explored. A survey of these techniques were proposed by Isermann [2005] and is illustrated in figure 2.22.

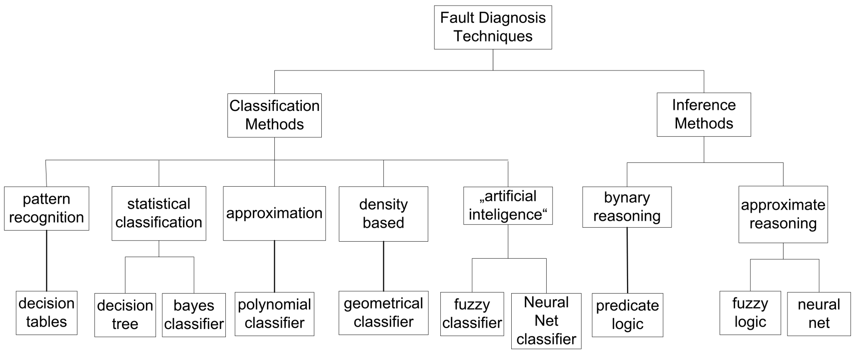


Figure 2.22: Survey of Fault Diagnosis Techniques.

According to the type of technical processes being evaluated and the fault symptoms available, fault diagnosis can be performed by means of classification or inference methods. Classification methods are applied if the relations between symptoms and faults are not completely clear or complex structured. These relationships may be determined by a priori knowledge, training data or a combination of both. Figure 2.23 illustrates the diagnosis of faults by means of classification methods.

Figure 2.23 makes an allusion to the relationship between the vector of symptoms $\underline{S}$ and the vector of faults $\underline{F}$. Although a two dimensional symptom combination is represented, a multidimensional association of symptoms and multiple occurrence of faults is the common case. Basically the goal of a classifier is to assign features (symptoms) to a hypothesis category (fault). The level of difficulty in a classification problem depends on the separability (specific determination) of hypotheses by means of the extracted features. The main classes of classification methods for diagnosing faults is depicted in figure 2.22. A more detailed explanation of relevant classification method will be explored in sections 2.3.1, 2.3.2 and 2.3.3.

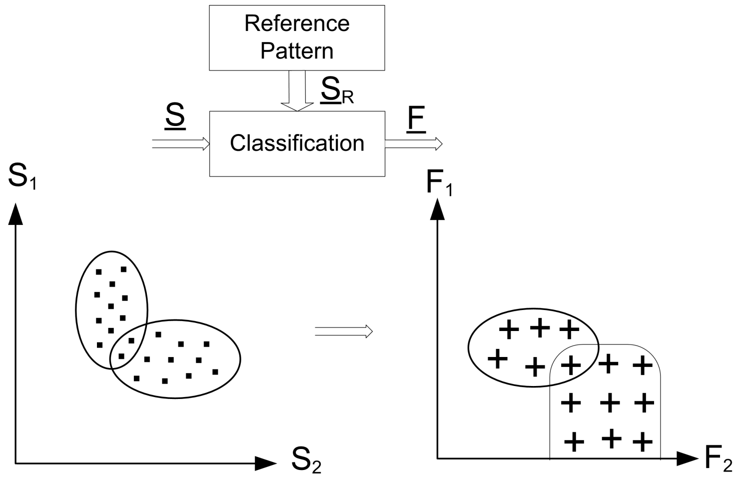
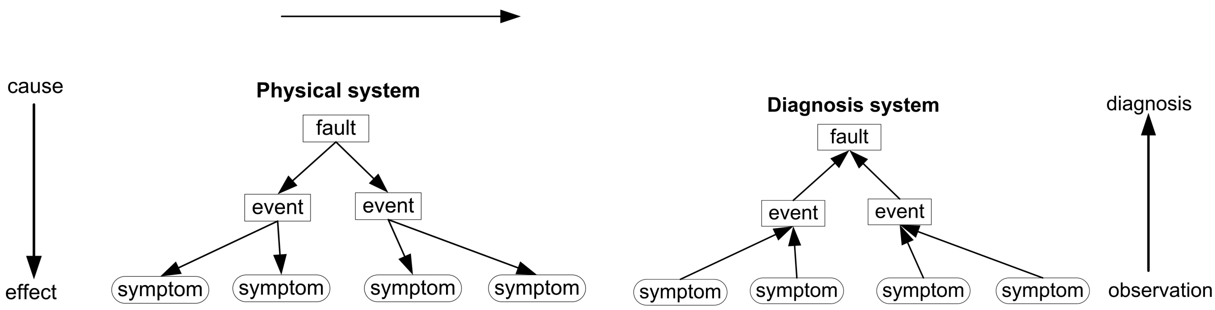


Figure 2.23: Fault Diagnosis by means of classification methods.

Inference methods are applied if the relationships between symptoms and faults are known more precisely. This prior knowledge can be represented in causal relations. Figure 2.24 shows the fault-symptom relationship assumed by inference methods.



(a) physical systems: from faults to symptoms. (b) diagnosis systems: from symptoms to faults.

Figure 2.24: Fault-symptom relationship.

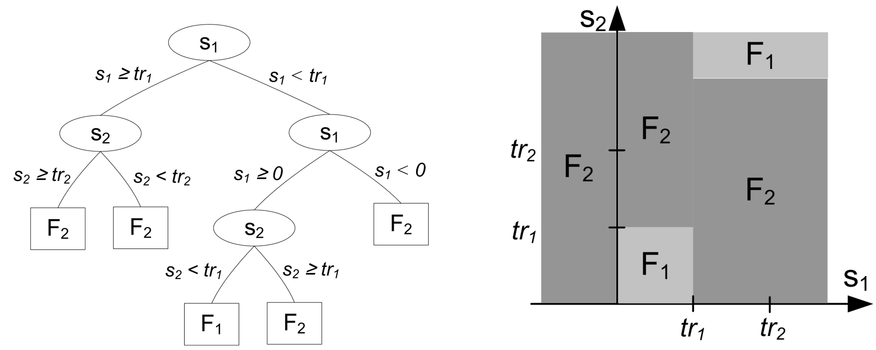
Once faults are assumed to be known as well as their relationships to symptoms, the correspondence between them can be determined by a causal system analysis (figure 2.24(a)). This description can be obtained by means of a Fault Tree Analysis (FTA - see figure 2.15). An alternative consists of proceeding from the symptoms to faults, obtained by means of Event Tree Analysis (ETA) building the diagnostic forward-chaining causalities (figure 2.24(b)). After determining the fault-symptom relationship, diagnosis can be performed by means of different rule like the IF - THEN ones. The most straightforward methods for diagnosing with inference methods are related to approximate reasoning strategies covered in details in Isermann and Ulieru [1993] and Ayoubi [1996].

2.3 Classification Methods

This section intends to discuss classification methods that are relevant for the implementation of this work. As mentioned in section 2.2.3, classification plays a very important role by performing fault diagnosis. That is why some examples of these techniques will be discussed.

2.3.1 Decision Trees

Decision trees are applied to select the best course of action in situations where the evaluated task is faced with uncertainty. They are represented in a graphic attempt in order to compare competing alternatives and evaluate them by combining uncertainties, efforts and pay off into specific numerical values. Figure 2.25 illustrate a decision tree for classifying two hypothetical faults F1 and F2 from two symptoms $s1$ and $s2$.



(a) decision Tree (b) Symptom space.

Figure 2.25: Decision tree for two hypothetical faults.

In the decision tree approach information (circles) are split until they reach a single class of data represented (squares). According to figure 2.25(a) data (symptoms) are evaluated from the top until they reach a leaf (faults). This means that symptoms are associated to the leaf's class membership with the aid of a symptom space segmentation (figure 2.25(b)). The main challenge in implementing decision trees for fault detection and diagnosis is dealing with the kind of symptoms, a multidimensional symptom space as well as defining the threshold levels among different classes. Typical approaches for performing it are based on expertise knowledge or learn based techniques using training data.

The goal with learn based techniques is to implement decisions which results in subsets with maximum "purity". Only the most significant information will be selected in order to perform the decision in favor of a determined class. Instead of calculating the "purity" of a decision subset its impurity is determined based on a statistical definition of an entropy index. Usually an adapted alternative function called Gini-index is applied for faster calculations results. The decision tree is built by choosing the decisions that minimize the entropy or the Gini-index of the sets.

Although the classification with decision trees is extremely straightforward, the decision structure can be easily followed and the computational efforts may be minimized according to the tree dimensions, they present some decisive weaknesses. In case of dynamic decision trees, where several growing and pruning approaches are applied, the algorithms complexity will increase drastically. Major challenges are the modeling of nonlinear decision processes and in same cases the considerably amount of training data for determining the symptom spaces and threshold levels.

2.3.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) have been extensively explored in pattern recognition and data classification through a learning or training process. Their ability to learn and generalize complex behaviors of training data has been used in connection with fault detection and diagnosis strategies [Silva et al., 1998]. It has been possible particularly due to the need of a little a priori knowledge about the process structure.

Core pieces from ANNs are their representations by the mathematical description of neurons. Analogue to the topology in the biological field, neurons interconnection in ANNs also enables the description of systems' input and output signals. This property can support the identification of unknown system parameters. On the other hand, in case of these signals being associated to determined groups or categories, ANNs will be applied to perform classification tasks. An illustration of a general neuron model is depicted in figure 2.26.

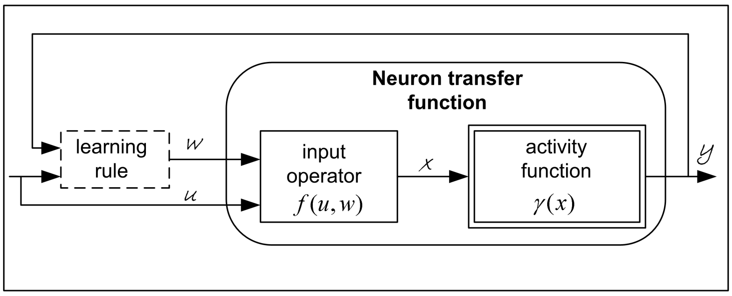


Figure 2.26: General neuron Model.

This mathematically mapped structure of a neuron consist of three main parts: learning rules, input operators and the final activity function. Learning rules consist of adapting the network in such a way that weight vectors $\underline{w}$ will be changed by an amount proportional to the difference between the desired output and the actual one. Thus this difference is minimized. Desired outputs are usually obtained by means of reference data while network adaptation is mostly performed by the minimization of the quadratic loss function:

$$
\begin{aligned}
J(w) &= \frac{1}{2}\sum_{n=1}^{N-1} e^2(n) \\
e(n) &= y(n) - \widehat{y}(n)
\end{aligned}
\tag{2.31}
$$

where $e(n)$ represents the model error and $y(n)$ and $\widehat{y}(n)$ describe the measured output signal and network output respectively. The goal is to find a minimum for the quadratic loss function:

$$
\frac{dJ(w)}{dw} = 0 \tag{2.32}
$$

Because of the nonlinearities between measured and calculated outputs, some numerical optimization methods are applied for solving equation 2.32. A direct solution is hard to be achieved. A detailed overview of different techniques for learning rules is given in [Mehrotra et al., 1996].

The input operator (figure 2.26) is concerned with the similarity measure between the input vector $\underline{u}$ and the weight vector $\underline{w}$ resulting the neuron activation vector $\underline{x}$. Forms for the determination of the activation vector are among others the scalar product (equation 2.33), Euclidian Measure (equation 2.34), feedback model and Sigma-Pi units [Mehrotra et al., 1996].

$$
\underline{x} = \underline{w}^T \underline{u} = \sum_{i=1}^{\#inputs} w_i u_i \tag{2.33}
$$

$$
\underline{x} = \|\underline{u} - \underline{w}\|^2 = \sum_{i=1}^{\#inputs} (u_i - w_i)^2 \tag{2.34}
$$

The last part of a mathematical neuron (figure 2.26) concerns with its activity function. It weights how powerful should be the output signal of a neuron in case of an output available. Examples of activity functions are among others hyperbolic tangent (equation 2.35), Sigmoidal (equation 2.36), binary and and Gauss functions.

$$
y = 1 - \frac{2}{1 + e^{2(x-c)}} \tag{2.35}
$$

$$
y = \frac{1}{1 + e^{-(x-c)}} \tag{2.36}
$$

Another important aspect is the configuration of ANNs. Here several neurons like those illustrated in figure 2.26 will be put together in order to perform different tasks like classification and system identification. Figure 2.27 gives an overview about a generic network architecture. It describes different alternatives on how neuron models (circles) may build a network. Mathematical neurons are disposed in different layers like an input, an output and user-defined hidden layers. Their topology is characterized by different types of connections described in figure 2.27: feedforward, feedback, recurrent and cross-layer links. Concerning the data output type, they can assume a binary, discrete or continuous behavior. Usually binary and discrete signals are applied to classification issues, while continuous ones are related to system identification tasks [Isermann, 2005].

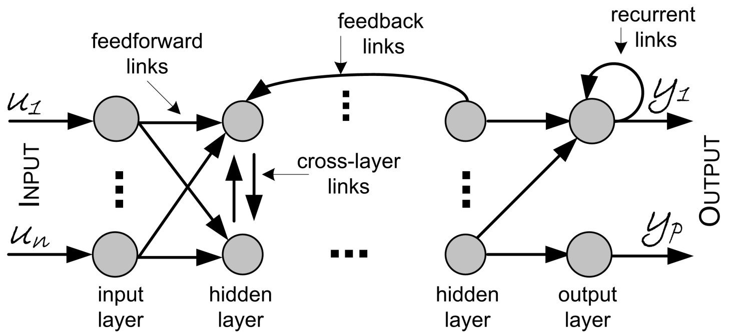


Figure 2.27: Generic ANN Structure.

Unlike other methods for performing diagnosis that rely on a specific function (e.g. Gaussian), ANNs approaches are designed to match any arbitrary function by minimizing an error function $e(n)$. This property makes the diagnosis of faults that are based on strong nonlinear relationships between symptoms and faults feasible. Examples of ANNs topologies for performing fault detection and diagnosis are the Multi-layer Perceptron and Radial-basis function approaches [Bishop, 1995]. Symptoms will be applied to the network inputs ($u_1 \cdots u_n$), which will be weighted by the neurons and activate the faults associated with the network outputs ($y_1 \cdots y_p$). These approaches are covered in details by [Sorsa and Koivo, 1991] and [Schramm, 1991].

Though ANNs are a very powerful method to perform classification and pattern recognitions tasks, their black-box characteristics and extensive required training phases make them difficult to comprehend and to upgrade.

2.3.3 Probabilistic Networks

Probabilistic networks (or probabilistic graphical models) represent variables in a problem and the probabilistic relationships among them. They are treated as a high-level language for structuring these mathematical relations, which may be applied to build complex models for different type of system components. It describes the dependencies between variables by ignoring numerical or functional details. They have been applied to several inference tasks like diagnosis and general classification issues [Buntine, 1996].

The most popular kind of probabilistic networks are the ones based on the Bayes' theorem (Bayesian networks). Bayesian networks are represented by directed acyclic graphs (DAGs) in which nodes represent variables, the arcs describe the existence of direct causal influences between linked variables. The strengths of these influences are expressed by forward conditional probabilities (see figure 2.28). A description of the basic principles of Bayesian networks is explored in details in [Kawasaki and Kiencke, 2004], [Duda et al., 2001], [Jensen, 2001] and [Pearl, 1988].

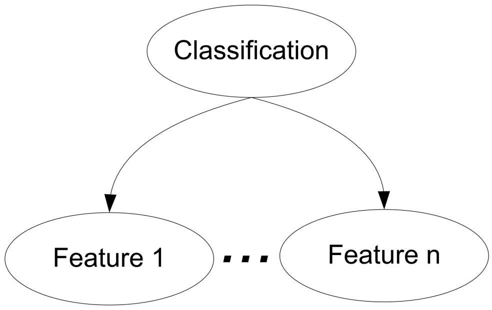


Figure 2.28: A bayesian network model for hypothesis classification represented by a Directed Acyclic Graphic (DAG).

An important aspect for general classification tasks and fault diagnosis are the fusion and propagation of new evidences[4] and beliefs trough Bayesian networks. Each node (variable) can be assigned with a certainty measure according to the axioms of the probability theory [Pearl, 1988]. Figure 2.29 shows the fusion and propagation scheme for evidences in a fragment of a causal tree representing an extract of a Bayesian network.

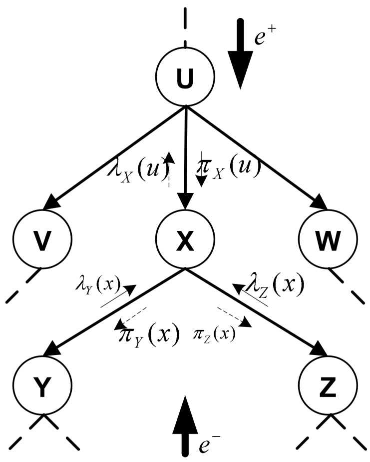


Figure 2.29: Fragments of a causal tree, showing incoming (solid arrows) and outgoing (broken arrows) messages at node X [Pearl, 1988].

As mentioned before each node represents a multi-valued variable, which comprises several mutual exclusive hypotheses (e.g. fault states). The probabilities that these hypotheses may

[4] Evidence refers to the probability assertion of hypotheses due to extern factors (e.g. prior knowledge) which are propagated trough the network. Evidenced nodes are also denominated instantiated.

assume is denoted by the overall belief (BEL) of a specific node determined by means of all evidence so far received. Thus,

$$\mathrm{BEL}(X) \triangleq P(x|e) \tag{2.37}$$

where e represents the fusion of all evidences contained in the tree rooted (e^+) and in the rest of the network (e^-) at a specific node (e.g. X). Assuming the determination of the belief induced on node X by means of the fusion of the overall incoming evidence $e = e_X^- \cup e_X^-$, the $\mathrm{BEL}(X)$ can be determined with the aid of the Bayes'Rule as:

$$\mathrm{BEL}(X) = P(X|e_X^-, e_X^+) = \alpha P(e_X^-|X)P(X|e_X^+) \tag{2.38}$$

where α means the normalizing constant to fulfill the basic axiom of the probability theory: $\sum_x \mathrm{BEL}(x) = 1$. The terms $P(e_X^-|X)$ and $P(X|e_X^+)$ represent respectively the likelihood model and prior probability term in analogy to the Bayes'Rule.

According to equation 2.38 the probability distribution of every node in a network (e.g. figure 2.29) may be calculated if the corresponding propagation vectors can be determined.

$$\underline{\lambda}(X) = P(e_X^-|X) \tag{2.39}$$

$$\underline{\pi}(X) = P(X|e_X^+) \tag{2.40}$$

While vector $\underline{\pi}(X)$ represents the causal or predictive evidence support of the parent nodes of X, vector $\underline{\lambda}(X)$ describes the diagnostic or retrospective evidence support of its children. In case of X being the root node, vector $\underline{\pi}(X)$ will be denoted by a prior evidence and interpreted as background knowledge [Pearl, 1988]. Thus, the belief of node X can be obtained by means of the fusion this two types of support as follows:

$$\mathrm{BEL}(X) = \alpha\underline{\lambda}(X)\underline{\pi}(X). \tag{2.41}$$

According to figure 2.29 the calculation of complete children evidence for node X, if X itself is not instantiated, can be determined by:

$$\begin{aligned}\underline{\lambda}(X) &= P(e_X^-|X) \\ &= P(e_Y^-, e_Z^-|X) \\ &= P(e_Y^-|X)P(e_Z^-|X) \\ &= \underline{\lambda}_Y(X)\underline{\lambda}_Z(X)\end{aligned} \tag{2.42}$$

where $\underline{\lambda}_Y(X)$ and $\underline{\lambda}_Z(X)$ describes the diagnostic support and are calculated by:

$$\begin{aligned} \underline{\lambda}_Y(X) &= M_{Y|X}\underline{\lambda}(Y) \\ \underline{\lambda}_Z(X) &= M_{Z|X}\underline{\lambda}(Z). \end{aligned} \tag{2.43}$$

Here the terms $\underline{\lambda}(Y)$ and $\underline{\lambda}(Z)$ describe the evidence in the instantiated Y and Z nodes respectively. These evidences represents the probability of assertion to the hypotheses referred to the nodes and can be determined by extern factors like prior knowledge or other inference methods. For instance Y and Z may correspond to different fault symptoms caused by fault states associated to X. Evidences $\underline{\lambda}(Y)$ and $\underline{\lambda}(Z)$ could describe the probability of occurrence of the specific symptoms. The influence of these evidences on the hypothesis states are assigned by means of the conditional probability matrix (dependency model) M:

	$\lambda(Y)_{s1}$	$\cdots$	$\lambda(Y)_{s1}$
$\lambda_Y(X)_{s1}$	$p(Y\|X)_{s1,1}$	$\cdots$	$p(Y\|X)_{s1,n}$
$\vdots$	$\vdots$	$\cdots$	$\vdots$
$\lambda_Y(X)_{sm}$	$p(Y\|X)_{sm,1}$	$\cdots$	$p(Y\|X)_{sm,n}$

Table 2.4: An example of a dependence model between the nodes X and Y assuming discrete hypothesis states.

Thus the calculation of the total belief of node X is complete if the evidence available from its parent (Node U) is evaluated as well:

$$\underline{\pi}(X) = \underline{\pi}_u(X) = M_{X|U}\underline{\pi}(U). \tag{2.44}$$

Analogue to the incoming diagnostic support, the incoming causal support in equation 2.44 is achieved by means of the evidence $\underline{\pi}(U)$ in the instantiated U node and of the corresponding dependence model $M_{X|U}$. An adapted model of the example described in table 2.4 is used to represent the influence of the parent node evidences on the hypotheses states of node X.

Once the belief in node X is updated, the belief of both descendants and ascendants of it may be updated as well. The outgoing message $\underline{\lambda}_X(U)$ of X to its parent U is determined similarly to the incoming children messages themselves. On the other hand the outgoing messages (causal support) to X's children have to be extracted of the own child information in order to avoid evidence double-counting. Equation 2.45 formalizes the diagnostic support transmission from nodes X → Y.

$$\underline{\pi}_Y(X) = \frac{BEL(X)}{\underline{\lambda}_Y(X)} \tag{2.45}$$

A summary of the calculations in a network node (e.g. X), called processor, is illustrated in figure 2.30. In this structure each node receives the causal support (π) from its parent and the causal support (λ) from its children. According to the design requirements each node can calculate its own π (to be delivered to its children) and the own λ (to be delivered to its parent). Each processor node comprises his own dependence model M, what improves the scalability of the whole network.

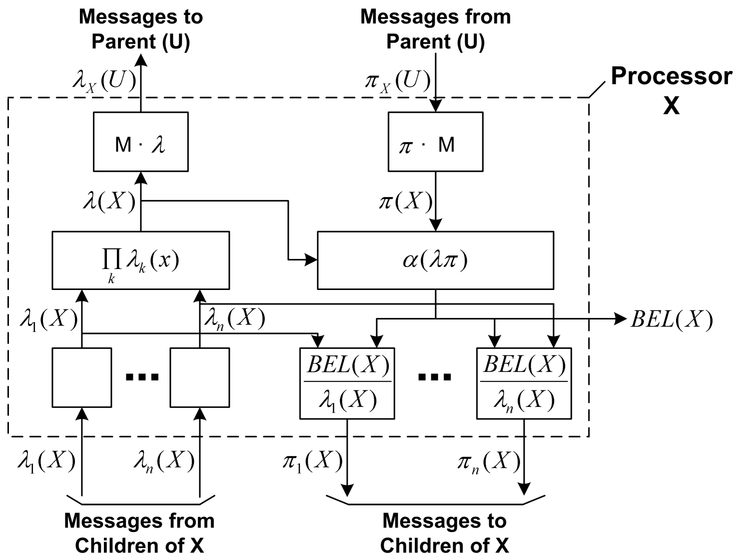


Figure 2.30: The internal structure of a single processor performing belief updating for node X [Pearl, 1988].

Depending on the dimensions of the Bayesian network (number of nodes) the computational efforts for updating the belief of every single node may be very high. That is why network message propagation is usually restricted to a minimum, where only the necessary information to update the belief of relevant nodes is propagated. Relevant nodes are determined by the network designer and are usually placed on the top of the tree. This means that messages are mostly transmitted from the bottom (leaf nodes) to the top (root nodes).

Bayesian network provides a transparent instrument for the general classification of hypotheses which is comprehensible. It enables the integration of different techniques for the extraction of features for classification and the determination of dependence between nodes and deals well with uncertainty. These techniques vary from White-box till Black-box modeling methods. Both network structure and dependence models may be built by means of background (prior) knowledge and learning strategies [Pearl, 1988].

2.3.4 Comparison

The choice of one out of several classification methods is rather due to a matter of application complexity and requirements than weaknesses of the methods themselves. According to the type of information available related to patterns to be recognized or the hypotheses to be classified the excellences of one or other method may be preferred. Tables 2.5 and 2.6 give a quantitative comparison of the properties of the classification methods covered earlier: Decision Tree, Neural and Probabilistic networks [Zhang and Bivens, 2007], [Zheng et al., 1999] and [Chen et al., 2007].

Properties	**Decision Tree**	**Neural Network**	**Probabilistic Network**
Transparency / Interpretability	high (white-box)	low (black-box)	high (white-box)
Classification type	mostly hard decision criteria	hard/soft decision criteria	hard/soft decision criteria
Network architecture definition	mainly training data	training data	prior knowledge / training data
Modularity and Scalability	low	low	high
Integrability with different techniques	low	medium	high
Accuracy	depending on training data	depending extremely on training data	depending on prior knowledge / training data
Training speed	high	high	medium

Table 2.5: Qualitative comparison of properties of three different classification methods: Decision Trees, Neural and Probabilistic Networks (Part I).

Properties	Decision Tree	Neural Network	Probabilistic Network
Model evaluation speed	high	very high	high
Ability to knowledge incorporation	low	very high	high
Handling with nonlinear relations	low	very high	medium
Computational efforts	medium	low	low

Table 2.6: Qualitative comparison of properties of three different classification methods: Decision Trees, Neural and Probabilistic Networks (Part II).

According to Zhang and Bivens [2007] if model performance is concerned, Bayesian networks (BN) offers better accuracy, are more suited for environments that change rapidly, while Neural networks (NNs) can achieve faster model evaluation time and support management routines that require intensive response time predictions. Decision Trees (DT) are also very accurate, but the response time depends on tree dimensions. BNs and DTs can be more easily comprehended by human, while NNs provide more flexible response time representation. Another important advantage of BNs is a multi-direct evaluation support, which may be decisive for several applications depending on a forward and backward analysis. For instance in case of fault diagnosis BNs allow both classification from symptoms to faults and reciprocally.

2.4 Summary

This chapter gave an overview about the fundamental theory and state of the art applied to this thesis. Relevancy, excellence and weaknesses of sensor data fusion methods were discussed. Afterward different principles of fusion techniques were explored in details.

As a specific method for sensor data fusion, the Multiple Target Tracking (MTT) approach was covered. It processes measurements of different targets to determine estimates of target's current states. In section 2.1.3 the different ways or stages for implementing MTT along with its architectures and algorithms were covered. Several proposed architectures for the fusion of sensor data and tracking of multiple target like [Waltz and Llinas, 1990], [Bedworth, 1992], [Luo and Kay, 1992] and [Naab, 2004] were presented.

Naab [2004] proposed a flexible and universal approach to share the actual fusion of sensor data into two stages. It consists of the low and high level fusion parts. The first part is concerned with the combination of measurement attributes without modifying it strongly and so providing an alternative to further and even application dependent fusion algorithms. The later consists of the filtering of the the attributes obtained by the low level fusion based on filter models. The most relevant algorithms for sensor data fusion and target tracking were covered in this section as well.

Finally in section 2.1.4 an overview of different strategies in which sensor data fusion supports an improvement of information reliability were presented. Sensor data fusion itself provides the first steps to increase the reliability of the acquired information. However sensor and system malfunctions are not detected. Therefore information integrity can not be assured.

In section 2.2 supervision and fault Management techniques which enable monitoring of automated systems and automatic protection in cause of faults were discussed. A survey of basic tasks and definitions of supervision a fault management processes were explored. Then, different strategies for the detection of faults varying from signal models till the detection with state observers and state estimation ([Willsky, 1976], [Clark, 1978b], [Clark, 1978a] and [Frank, 1987]) were outlined. Finally the basic principles of fault diagnosis were explored. Techniques for fault diagnosis are basically shared in classification and inference methods and support the determination of fault properties like type, intensity, location and detection time [Isermann, 2005].

This chapter ends with an overview of three examples of classification methods: Decision Tree (DT), Artificial Neural Network (ANN) and Bayesian Network (BN). Classification methods play a very important role for fault diagnosis and therefore they were covered in more detail. The choice of these three out of several methods is due to they ability to classify data rapidly, be reliable and straightforward. A qualitative comparison of this three methods were covered in section 2.3.4. By comparing these techniques it was ascertained that BNs can be easily comprehended by human, deal well with data uncertainty and deliver good results regarding classification performance.

3

Multiple Fault and Target Tracking

This chapter describes the proposed concept for improving quality, integrity and reliability of the driving environment information called Multiple Fault and Target Tracking (MFTT). It further develops sensor data fusion algorithms in order to combine and integrate target tracking and fault detection strategies into a consistent methodological framework. By means of this proposed concept a novell probabilistic model based approach is developed not only for validating acquired information, but also for detecting and identifying component (sensor) faults and failures.

Basis for the theory to be introduced here were discussed in chapter 2. Based on those aspects the new approach to evaluate the acquired data will be presented. First the design concept will be explored. Then a framework architecture that allow efficient implementation of the proposed concept will be introduced and discussed in details.

3.1 Approach

As mentioned previously measuring and interpreting events in the driving environment are essential parts for the feasibility of future and safety-related ADAS. In this sense the acquisition of these events can be performed by means of several sensor systems, which may be faulty or not working properly. That is why the reliability of sensor data and its processing methods have to be assured. Sensor faults and failures should not evocate additional critical situations.

Another important aspect consists of the interpretation of the environment information. Interpreting these events requires mostly the evaluation of redundant and dissimilar sensor information regarding their common properties in order to describe same objects or actions in the environment.

For these reasons a method that deals with:

- the fusion of sensor data (section 2.1),

- the tracking of multiple targets (section 2.1.3) in the driving environment
- the evaluation of sensor information with regard to its integrity, reliability and correctness and
- the supervision and diagnosis of sensor hardware (section 2.2)

is required. MFTT is a technique that extends and combines reasonably the functionality of sensor data fusion (section 2.1), multiple target tracking (section 2.1.3) and fault detection and identification (FDI - section 2.2) methods. It favorably exploits structures and algorithms of these classical approaches.

The first step of this proposed technique consists of interpreting driving environment information regarding its integrity and reliability. Figure 3.1 represents the information flow from the sensor measurements through the three processing layers:

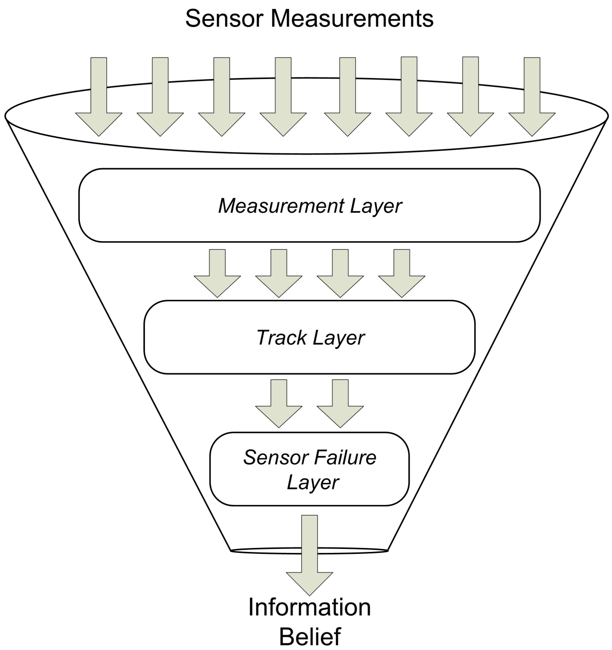


Figure 3.1: Scheme for classifying driving environment information.

1. measurement layer: extracts and processes all relevant and observable features acquired from events in the environment. The belief or level of confidence of measurements regarding specific hypotheses are determined and made available for the following layers. These hypotheses are related with the relevancy, integrity and reliability of the acquired measurements from the point of view of different ADAS.

2. track layer: extracts and processes all relevant features from calculated tracks[1] in order to determine their relevancy. Measurement features in form of believes determined in the measurement layer will be combined with the correspondent tracks' own features and thus contribute for track relevancy calculations as well. The results obtained here will be also made available to the following layer

3. sensor failure layer: extracts and processes all relevant features from the available sensors regarding their malfunctions. Again all previously obtained information in the preceding layers is taken into account along with the own sensor features.

The scheme in figure 3.1 is depicted in form of a filter in order to denote the information reduction not in a sense of lost but in a sense of extraction and processing of the most relevant features for the environment classification. At the output of the system an information evaluation regarding measurements, tracks and sensors is made available. This information represents the believes in a specific hypothesis that such a measurement, track and sensor may assume. Examples of these hypotheses and further details of their processing in the introduced concept will be covered in the following sections.

Essentially the proposed MFTT concept consists of tracking relevant and faulty events in the driving environment. Mathematical models explore properties like redundant field of views of dissimilar sensors as well as the behavior of both relevant and irrelevant targets. These models enable not only validating acquired information, but also the detection and identification of eventual faulty components. The categorization of the acquired events consists of a classification task performed in the three layers discussed previously.

3.2 Design Concept

Design phase starts considering the available system information and the scheme depicted in figure 3.1. The concept for classifying the driving environment information regarding its reliability is illustrated in figure 3.2. Its fundamental premise consists of comparing measured events with assumed expectations. These expectations are settled in mathematical models. First aspect to be considered is what a single or a group of sensor measurements can describe in the environment. According to the sensor fusion and multiple target tracking theory covered in section 2.1, measurements hypotheses may be considered as follows:

- **existent tracks:** it describes the temporal interrelation of object attributes that have been traced by the system during a determined period of time. Examples of attributes are the object position, velocity, acceleration, dimensions and so on. Some of the different types of attributes that compose a object track may be observable (measurable) or

[1] The term Track is related with the state trajectory from events or object hypotheses estimated from a set of measurements that have been associated to the same target [Bar-Shalon and Fortmann, 1988] (see section 2.1.3).

not. One part may be obtained through sensors observations and the other one can be obtained through different sensor fusion and target tracking strategies (section 2.1).

- **new tracks:** it describes the temporal interrelation of object attributes that had not been seen before. Probably they were covered by another valid object track or they were outside the sensors coverage area. Measurements associated to this hypotheses initiates these new tracks. As well as existent tracks, new tracks can describe diverse attributes from different objects like vehicles, pedestrian and other relevant targets in the driving environment.

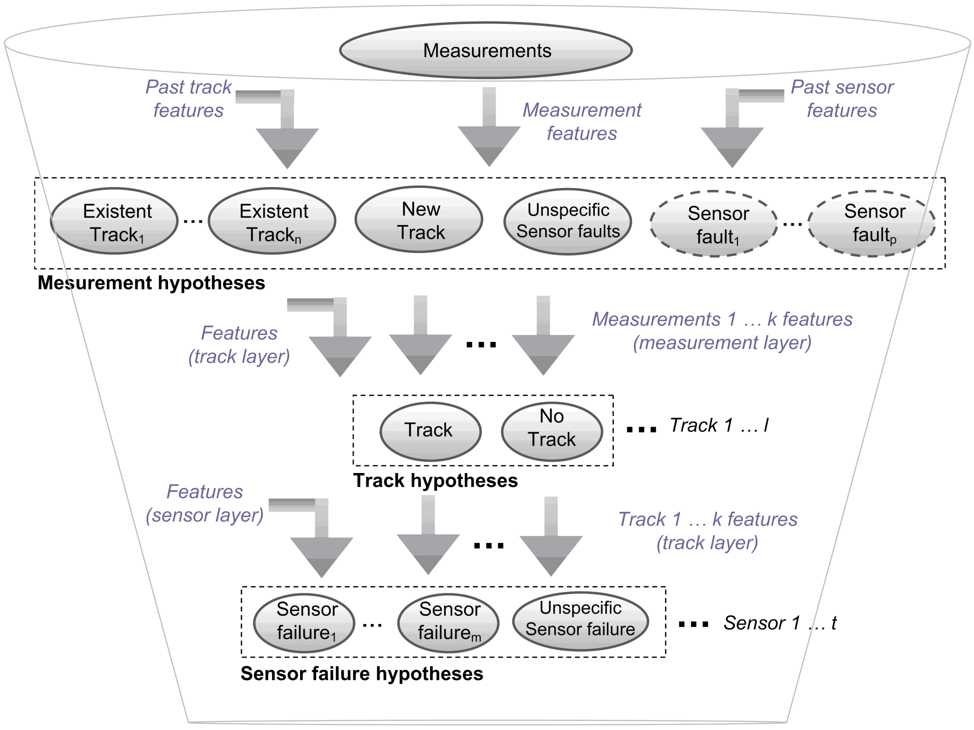


Figure 3.2: Concept for classifying the driving environment information.

- **unspecific sensor faults:** it describes sensor faults when the expected actions in the environment are modeled. This means that a vehicle cannot simply appear in the middle of a highway or between two other vehicles. Thus, this measurement assumptions can be verified against such premises. Basis for these type of hypotheses will be described in section 5.1.2 in the topic sensor faults. As mentioned before, most of these sensor faults are difficult, expensive and not essential to model. Therefore it is more relevant to detect these anomalies than trying to discover their origin.

- **specific sensor faults:** it describes sensor faults when a specific not expected action in the environment is modeled. Even though the modeling of such faults is not straightforward and in some cases expensive (computational efforts), their identification is sometimes necessary. An example of a specific sensor fault is the modeling of ground clutters. Ground clutters occur if sensor signal emissions reflect at the ground. Cause for these erroneous measurements is the abrupt alterations of the pitch or roll angles due to braking maneuver or a steep descent or ascent. The steps for identifying, processing and classifying ground reflexions will be covered in details in the section 5.1.1.

The classification of these hypotheses is performed by means of extracted features obtained from sensor measurements. These features are combined with the ones extracted from past tracks and from the former sensor hypotheses. These measurements hypotheses have to be mutually exclusive. They cannot be all true. After calculating the hypotheses for every sensor measurement available, the information processed in this layer is made available for the next stage, the track layer.

In the track layer features which are extracted from the measurement hypotheses will be used in order to achieve track classification. According to the belief of these hypotheses, measurements will be associated to the corresponding tracks. This means that the driving environment information being cyclically updated by the sensors the association between measurements and tracks is performed dynamically. In every cycle existent or new tracks may receive updates that combined with different extracted track features contribute for the calculation of the track belief. In this sense the belief of a track is shared here in two hypotheses states:

- **tracks:** are related to several measurement attributes that correspond to relevant objects for different ADAS like vehicles, pedestrians, etc. The assertion of this hypothesis is directly supported by the states "existent tracks" and "new tracks" in the measurement layer.
- **no tracks:** are related to several measurement attributes that correspond irrelevant or erroneous objects for different ADAS like "ghost" objects generated by clutters. The assertion of this hypothesis is directly supported by the states "unspecific sensor faults" and "specific sensor faults" in the measurement layer.

Although the hypotheses in this layer are described only by means of these two states: tracks or no tracks, tracks may assume more specifically states according to the observed measurement attributes. Instead of classifying a object just as relevant or irrelevant, one may be able to classify it for instance as car, trucks, motorcycles, pedestrian, etc. depending on the features available.

Similar to the procedures described before, the classification in the sensor failure layer will be performed by means of the hypotheses calculated in the preceding layers combined with the extraction of the own sensor features. The states for this layer are shared as follows:

- **specific sensor failures:** it describes sensor failures if these are modeled and the obtained features correspond to it. In this case failures can be detected and diagnosed. According to the correspondence between data and model, failure intensity can be achieved as well.

Achieving the classification of specific sensor failures will support the system to assume effective countermeasures in order to minimize or even solve the problem. Example of attempts in this case could be fixing or ignoring the measurements originated from a faulty sensor.

- **no sensor failures:** it describes the failure free case if the extracted features correspond to the expected sensor functionality modeled.
- **unspecific sensor failures:** it describes sensor failures if the expected sensor functionality is modeled and the obtained features drift to an unexpected and unmodeled behavior. Due to missing features and attributes failures can not be diagnosed, but at least detected. This type of approach may be very useful for systems, where the specific modeling of sensor failures is not straightforward or the execution time of these models exceeds their time constraints.

In section 5.1.1 examples of specific sensor failures will be covered in details. Other than the hypotheses from the measurement and track layers, the hypotheses here are not mutually exclusive. One sensor can be affected by multiple failures simultaneously (e.g. misalignment and partial blindness).

At the end of the processing chain the hypotheses believes for the analyzed measurements, tracks and sensors are made available. Different ADAS can access the corresponding information that fulfill their implementation requirements. The main goal of achieving a high reliability of the driving environment information and its evaluation can be reached depending on the availability and quality of the computed features. Figure 3.3 summarizes the processing chain in the three explored layers. In the following sections the methods for the extraction of features for all layers and and their correlation with the corresponding hypotheses will be covered in details.

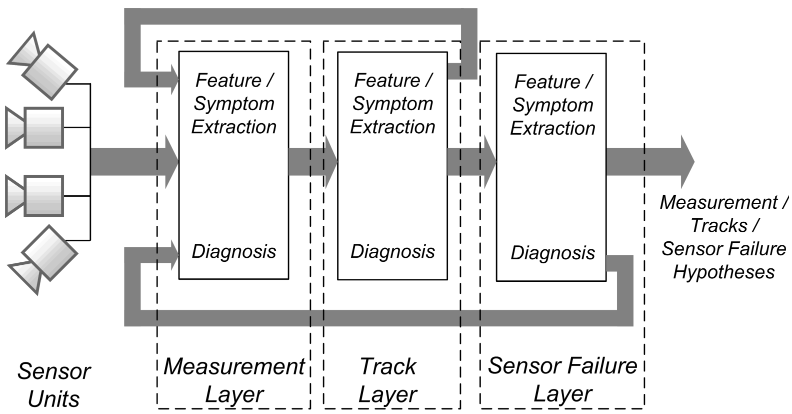


Figure 3.3: Schematic illustration of the data flow in the three processing layers.

According to the concept covered in the preceding sections, a method to perform the classification of the driving environment information regarding its reliability was designed and implemented. The new method consists of a probabilistic track oriented fault detection and classification. This means that relevant and erroneous (e.g. due to sensor failures or faults) are tracked simultaneously within a probabilistic approach (Fig. 3.4).

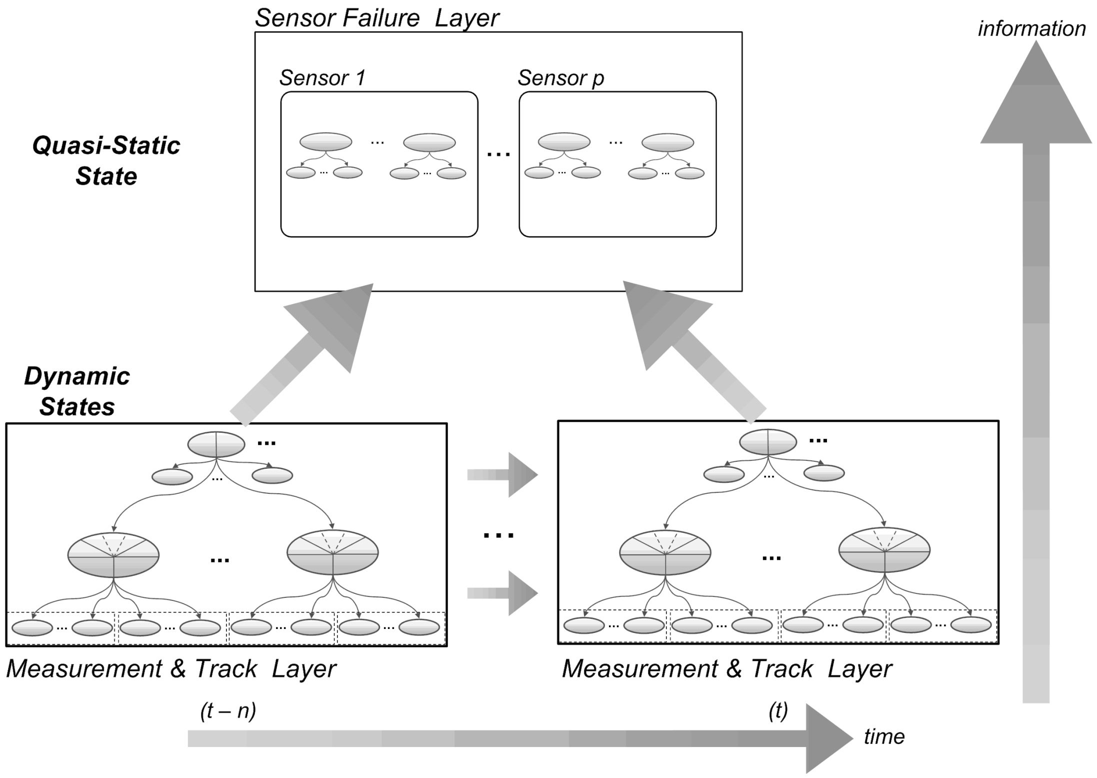


Figure 3.4: Probabilistic network for the classification of the driving environment information regarding its reliability.

The proposed method is a derivation, adaptation and further development of the well-known probabilistic or belief networks (section 2.3.3). The architecture depicted in the figure 3.4 reflects the concept illustrated in the figure 3.2 regarding its feasibility and adaptation in a probabilistic concept. This means that the classification of measurement tracks and sensors hypotheses is achieved by means of a soft decision (probabilistic). The proposed method makes an allusion to the human decision principle, if this decision has to be made under uncertain circumstances.

In order to fulfill the requirements of the proposed concept, the original theory of probabilistic networks has to be adapted and further developed. The new implemented design builds a modified architecture of dynamic probabilistic networks. This architecture consist of two main states (Fig. 3.4): dynamic and quasi-static states.

3.2.1 Dynamic Track Oriented Fault Detection and Classification

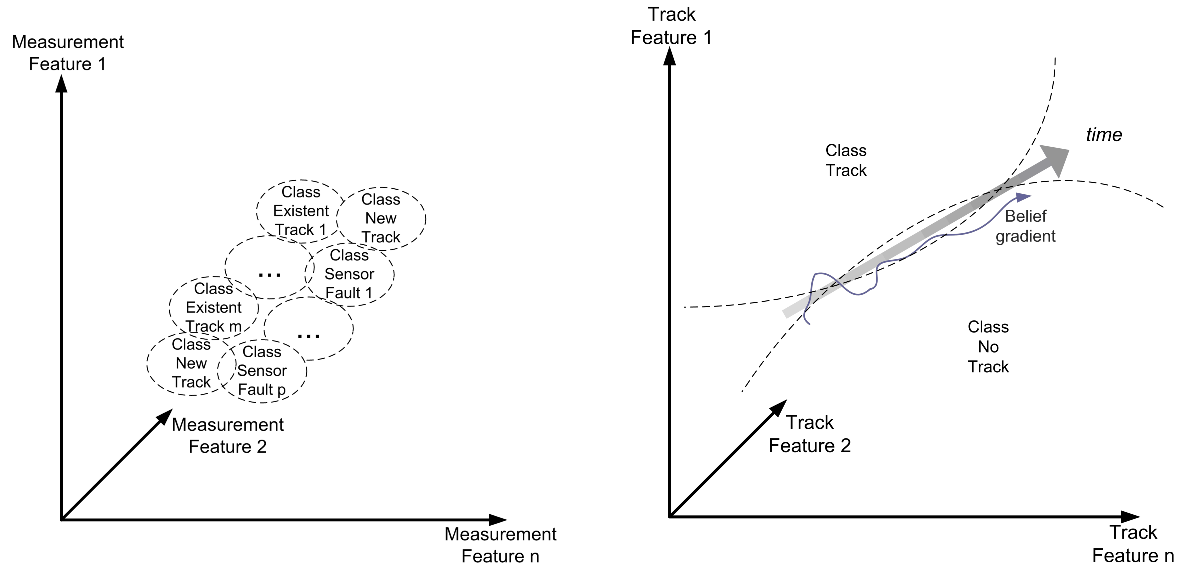
Dynamic states are the ones, which the calculated hypotheses states can and must be determined in the current duty cycle. This is a prerequisite for most assistance functions. In order to enable the reliable implementation of such functions, it is necessary to give at least an a priori evaluation of the acquired information. In the dynamic states the required information to assert these hypotheses are usually available for a prior classification until a stable one is possible. These states are more susceptible to changes. This means that the calculated hypotheses may vary stronger from one duty cycle to another than in a quasi-static state configuration. For theses reasons the calculations from the measurement and track layer are made available in every duty cycle.

As described in section 2.1 new sensor measurements may be acquired in every duty cycle. But only the current ones will be considered for calculation. Past sensor data is not stored. That is why a dynamic classification according to the hypotheses described in 3.2 takes place. On the other hand track hypotheses are considered as long as measurements are associated to it or during a specific period of time without measurement members (section 2.1). This means that the connections between measurement and track layers must be renewed in every cycle. Beyond it the influences between the calculated past hypotheses for tracks are taken into account for their current hypothesis state determination. A tradeoff between feature extracted from the current measurements and the ones extracted from the past tracks is determined.

Figure 3.5 illustrates the class hierarchy for classifying the information in the dynamic state level. According to it the class hierarchy can be interpreted as the data classification in the depicted classes.

An important fact to consider is that the more meaningful the features are, the higher is the separation level among the classes. Figure 3.5(a) shows the class hierarchy for the measurement layer. It consists of a multidimensional classification task. By means of the combination of different features the measurement membership may be determined. The goal is to extract several meaningful features that enables the unambiguous association from the information to the classes by means of a decision threshold assertion. As mentioned before the measurement information is transitory and is available during a duty cycle. Therefore this classification is treated as an instantaneous one. The history information is not used directly from the past measurements, but rather from past sensor and track information.

On the other hand the class hierarchy for the track layer (Fig. 3.5(b)) assumes a more time dependent classification. A belief gradient per track is determined according to the elapsed time. This gradient behaves as a high frequent signal around the track generation time. As soon as tracks reach a stable "age", this belief signal is attenuated adopting a low pass filter behavior. Influences of the past track believes combined with features extracted from the measurements induce to a time dependent classification. Similar to the classes hierarchy in the measurement layer, the class separability depends on the extraction of meaningful features. This results in a multidimensional classification task as well.



(a) Class hierarchy for the measurement layer. (b) Class hierarchy for the track layer.

Figure 3.5: Class hierarchy in the dynamic state level.

Measurement Layer

Figure 3.6 depicts the architecture of the measurement layer within the whole dynamic probabilistic network (Fig. 3.4). It represents only the configuration for one measurement (other measurements can be similarly interpreted). For the sake of simplicity the measurement hypotheses in figure 3.6 are abbreviated as follows:

- $ET_1 \cdots ET_l$: stand for existent tracks.
- NT: stands for new tracks.
- $USFt$: stands for unspecific sensor faults.

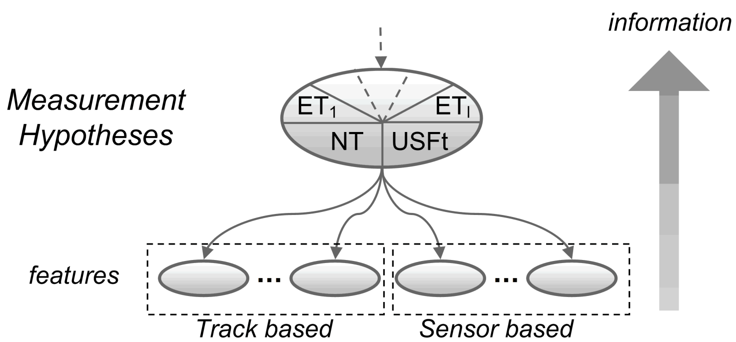


Figure 3.6: Probabilistic network for one measurement in the measurement layer architecture.

This network fragment consists of two main stages: feature extraction and final hypothesis. At this level the extraction of features for the measurement layer are based on the past information of:

- **tracks:** attributes and believes of formerly calculated tracks provide evidences for the assertion of hypotheses in the measurement layer.
- **sensors:** the level of confidence in a specific sensor also provide evidences for the assertion of hypotheses in the measurement layer. These confidence levels are obtained by means of the evaluation of sensor failures described in section 3.2.2.

Figure 3.7 illustrates the feature extraction of sensor data for the measurement layer. According to it feature extraction is achieved by firstly combining sensor data, past tracks and past sensor information. Depending on the type of information available, different kinds of models can be built for evaluating sensor data. These models may vary from White-Boxes to Black-Boxes (see figure 2.17) and support feature extraction (symptom generation) via plausibility criteria checks.

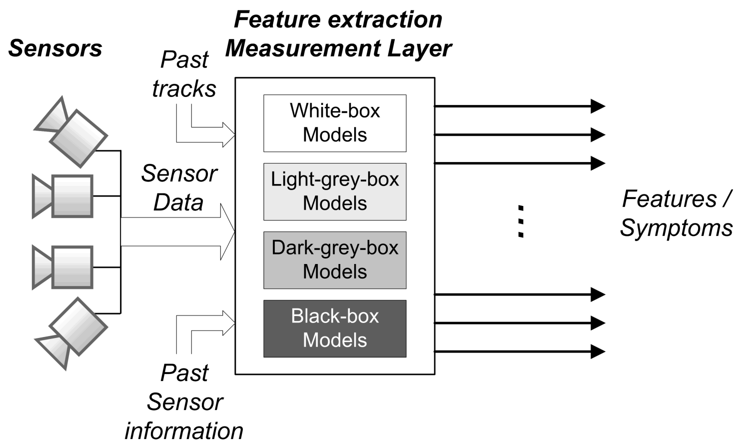


Figure 3.7: Scheme illustrating the feature extraction of sensor data for the measurement layer.

Both track and sensor based features extractions assume the probabilistic principle as well. This means, the feature characteristics are not binary determined (e.g. available/not available), but in a "soft" decision sense by means of membership functions (e.g. cold, warm, hot, very hot, etc.). These functions try to approximate the human decision way under uncertain circumstances. Equation 3.1 gives an example of a membership function, which are applied to the membership determination of different features. In case of features that may assume more than two membership states, the proposed function can be extrapolated (Fig. 3.8). The presented equation consists of an exponential function where: $f(x)$ is the membership state or evidence of a determined feature; the parameter a describes the measure of the analyzed feature in which the membership stays at a maximum or a minimum; α describe the boundaries between two feature state where the membership value assumes 50% of a state.

$$f(x)_i = 1 - \frac{1}{1 + e^{a - \frac{xa}{\alpha_i}}} \quad \forall \quad \alpha_{i-1} \leqslant x \leqslant \alpha_i(1 + \frac{1}{2}) \tag{3.1}$$

Analogue to the class hierarchy in the dynamic state level (Fig. 3.5), the determination of feature states will be performed. The main difference here is that feature states are a one dimensional pre-classification task. This is performed by means of the feature measure itself with the parameters in equation 3.1. One relevant aspect of the presented concept is the way in which the constant a and the parameters α are determined. The concept enables the implementation of two strategies:

1. prior knowledge: it is adopted if enough prior knowledge about the states of a specific parameter is available. The main advantage of this approach is that the time for an extensive training phase can be saved.
2. training data: it is adopted if none or incomplete prior knowledge about certain features are available. According to section 2.3 the weaknesses of this approach are among others a extensive training phase and the dangerous of training based on suspicious data. Usually this approach is used to complement the one before. The main idea is that through prior knowledge the constraints for the training phase may be determined.

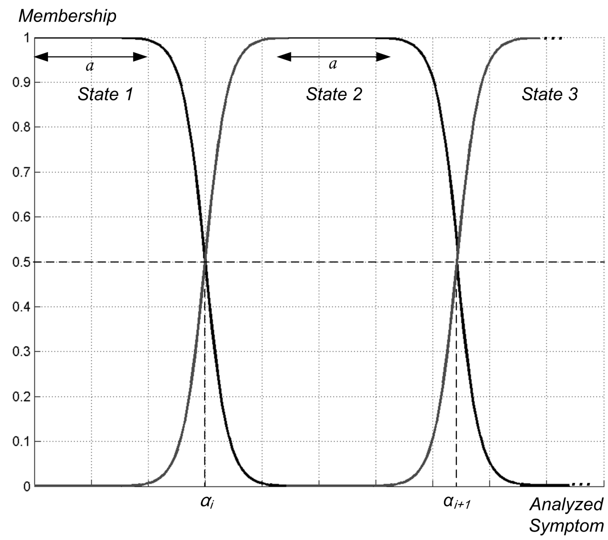


Figure 3.8: Membership functions for the analyzed features.

After calculating the membership state for the features their relevancy is checked according to their extraction principle. For instance track or sensor based features will be proved against their past belief information. This means that if the extraction of a certain feature is based on the information of a determined track or sensor their past belief must be taken into account. This is initially performed by means of weighting models. Their basic structure is illustrated in table 3.1.

evidence ↘ reference	$Ref_{state\ 1}$	$Ref_{state\ 2}$	$\cdots$	$Ref_{state\ n}$
$f(x)_{state\ 1}$	w_{11}	w_{12}	$\cdots$	w_{1n}
$f(x)_{state\ 2}$	w_{21}	w_{22}	$\cdots$	w_{2n}
$\vdots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$
$f(x)_{state\ n}$	w_{n1}	w_{n2}	$\cdots$	w_{nn}

Table 3.1: Weighting Model

The weighting models describe the relationship or influence (w_{ij}) of a determined reference (Ref) state on the specific evidence ($f(x)$) state:

$$w_{ii} = p(Ref_{state\ i}|x_{state\ i}) \tag{3.2}$$

If reference state (Ref_j) assumes a probability of 80% the influence value (w_{ij}) of 30% will affect the evidence value ($f(x)_i$) calculated with equation 3.1. Equation 3.3 describes the general principle of influence calculation.

$$\underline{f(x)}_{norm} = \frac{1}{\sum\limits_{i=1}^{max\ \ states} f(x)_i} \overline{WM}\ \underline{Ref} \tag{3.3}$$

The terms in equation 3.3 are interpreted as follows:

- $\underline{f(x)}_{norm}$: is the normalized vector of feature states after the influence calculation.
- $\frac{1}{\sum\limits_{i=1}^{max\ \ states} f(x)_i}$: is the normalization factor.
- $\overline{WM}$: it describes the weighting matrix model illustrated in table 3.1. Similar to the parameters for the membership determination, the influence values may be determined by means of a priori knowledge or training data.
- $\underline{Ref}$: it describes the vector of reference states.

Once measurement features are extracted and weighted, they may be treated as symptoms for specific hypothesis states. As mentioned before both expected and not expected behavior of the acquired information can be modeled. Thus the deviation between the modeled behavior and the acquired information (residual) is treated as a symptom for the classification of hypotheses. The integration of several model approaches is assured here. For instance several White-Box and Black-Box models may be applied (see section 2.2.2).

The next and final step for the assertion of measurement hypotheses is the combination of all extracted features with their correspondent influence calculation. Based on the probabilistic network approach (2.3.3) the classification of measurement state hypotheses are performed. For each extracted feature a dependence model is built. As mentioned before these model may be determined by means of a priori knowledge or training data. Table 3.2 illustrates an example of a dependence model.

hypotheses ↘ evidences	$ev_{state\ 1}$	$ev_{state\ 2}$	$\cdots$	$ev_{state\ n}$
Existent $Track_1$	d_{11}	d_{12}	$\cdots$	d_{1n}
$\vdots$	$\ddots$	$\ddots$	$\ddots$	$\ddots$
Existent $Track_n$	d_{n1}	d_{n2}	$\cdots$	d_{nn}
New Track	d_{n1}	d_{n2}	$\cdots$	d_{nn}
Sensor Fault	d_{n1}	d_{n2}	$\cdots$	d_{nn}

Table 3.2: Dependence Model

By means of the combination of all extracted features through the correspondent dependence models the hypotheses assertion is assured according to equation 3.4.

$$\underline{bel}_k(t)_{meas} = \eta\ \underline{\lambda}_k(t)_{meas}\ \underline{\pi}_k(t)_{meas} \tag{3.4}$$

According to equation 3.4 the term $\underline{bel}_k(t)_{meas}$ corresponds the vector of believes by means of probability values of the possible hypothesis states for one measurement k in a time stamp t. The term η describes the normalization factor that limits the probability values between 0 and 1. Its calculation is performed through equation 3.5.

$$\eta = \frac{1}{\sum_{i=1}^{max\ \ states} bel_k(t)_{meas(i)}} \tag{3.5}$$

Adopted from the probabilistic network approach (2.3.3) the term $\underline{\lambda}_k(t)_{meas}$ corresponds the so-called diagnostic support from one measurement in a time stamp. It corresponds the combination from the information calculated in the feature level (children nodes) and sent to the measurement level (parent node). The diagnostic support is calculated through the product of all children nodes (equation 3.6). According to the Bayes's theorem, the product of the information between two children nodes is only possible if their features are independent. Pearl [1988] shows that this assumption may be assumed in many dependent cases without influencing the correctness of the results. For the case in which one or more features are not available in the current duty cycle, this missing data does not influence the whole calculation (see 2.3.3).

$$\underline{\lambda}_k(t)_{meas} = \prod_{i=1}^{\#features} \underline{\lambda}_{k,i}(t)_{feat} \tag{3.6}$$

In this sense the diagnostic support from the extracted features ($\underline{\lambda}_{k,i}(t)_{feat}$) is determined as follows (3.7):

$$\underline{\lambda}_k(t)_{feat} = \overline{DM}\ \underline{f(x)}_{norm} \tag{3.7}$$

where $\overline{DM}$ corresponds to the dependence model matrix illustrated in table 3.2 and $\underline{f(x)}_{norm}$ (equation 3.3) denotes the normalized vector of feature states after the influence calculation (evidence states). This product describes how much a feature affects the classification of a specific hypotheses set. The last term for the hypotheses assertion on equation 3.4 is the so-called causal support $\underline{\pi}_k(t)_{meas}$. According to section 2.3.3 the causal support corresponds to the incoming information from the track layer (parent nodes). It also describes the a priori knowledge about a specific set of hypotheses. In case of no information coming from the parent nodes the uncertainty about the set of hypotheses will be equally shared among them. Thus the assertion of measurement hypotheses can be illustrated by means of equation 3.8.

$$\underbrace{\begin{bmatrix} \text{Existent Track}_1 \\ \text{Existent Track}_2 \\ \vdots \\ \text{Existent Track}_n \\ \text{New Track} \\ \text{Sensor Fault} \end{bmatrix}}_{\underline{bel}_k(t)_{meas}} = \eta \underbrace{\begin{bmatrix} \text{Existent Track}_1 \\ \text{Existent Track}_2 \\ \vdots \\ \text{Existent Track}_n \\ \text{New Track} \\ \text{Sensor Fault} \end{bmatrix}}_{\underline{\lambda}_k(t)_{meas}} \odot \underbrace{\begin{bmatrix} \text{Existent Track}_1 \\ \text{Existent Track}_2 \\ \vdots \\ \text{Existent Track}_n \\ \text{New Track} \\ \text{Sensor Fault} \end{bmatrix}}_{\underline{\pi}_k(t)_{meas}} \tag{3.8}$$

Track Layer

Measurement believes are interpreted as children nodes for the determination of track hypotheses. They supply one of the main source of information to classify the track states. The association between one measurement node and a track one depends on the probability asserted to the different states of the hypotheses in the measurement layer. Because of only current measurements are considered for calculation, probabilistic links between tracks and layers will be rebuilt in every duty cycle. This characterizes the dynamic states in the proposed concept. Figure 3.9 shows the probabilistic network for the assertion of the track hypotheses.

Analogue to the classification of measurement hypotheses the information flow assumes a bottom-up characteristic. Several measurement hypotheses may be associated to a track in order to determine its current state. Assuming that a track may not correspond unconditionally single or a whole object (car, truck, pedestrian, etc.), one measurement may contribute for the hypotheses assertion of several tracks. One or more tracks may concur for the measurement membership. This may happen because of the lack of track attributes to build a model for a complete object description. In this sense a measurement can contribute proportionally to one track and to another. Additionally, different features that may be extracted from the

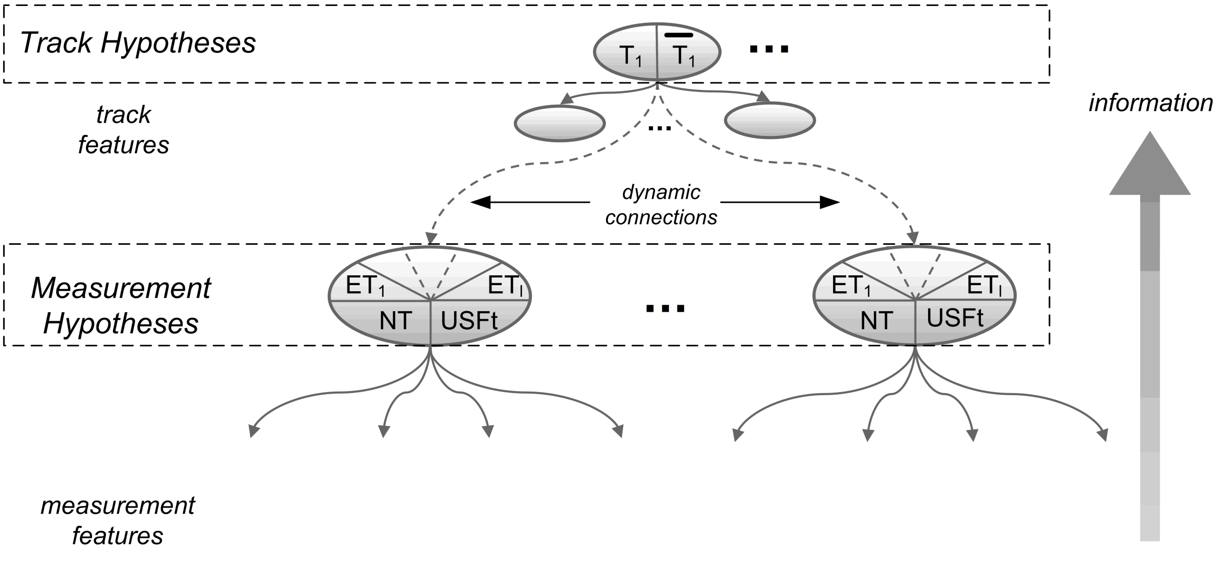


Figure 3.9: Probabilistic Network for the determination of track believes.

correspondent track contributes to its hypotheses assertion. Thus, the incoming children (track features and measurement hypotheses) information are processed in order to classify the current track states:

- T_x: describes the probability that the current track (x) represents one or part of a relevant object [2].
- $\overline{T_x}$: describes the probability that the current track (x) represents an artifact.

The first step for the classification of track hypotheses consists of its feature extraction. It corresponds the acquisition of the most important and observable features which supports the determination of track states. As described in the measurement layer, the whole system is based on a "soft" decision concept. Thats why the feature extraction here is performed by means of "soft" membership functions as well. Figure 3.10 illustrates the feature extraction of track data for the track layer.

Analogue to the feature extraction in the measurement layer, features are extracted in this layer by means of different type of models depending on the track information available via plausibility check. In this sense equation 3.1 can be also applied for membership determination. In order to determine the amount of influence of a certain feature in the classification of tracks, dependence models are used. Table 3.3 illustrates an example of a dependence model for track features.

After determining the influence of different track features, the contribution of the associated measurement hypotheses has to be calculated. This step is also performed by means of dependence models. According to whole concept philosophy, the weighting factors (w_{xx}) are

[2] The term relevant object (car, truck, pedestrian, etc.) depends on the ADAS requirements.

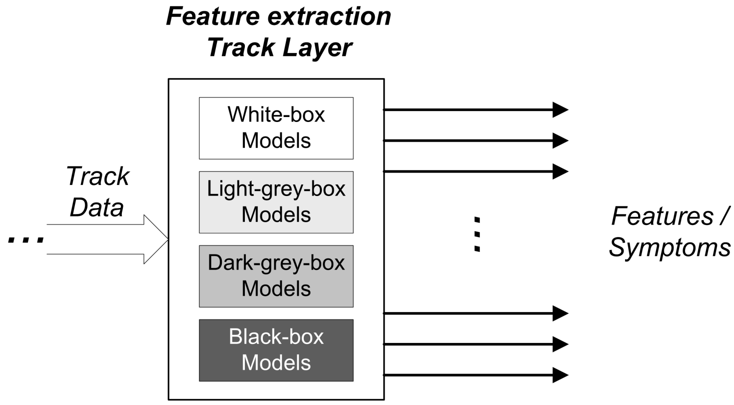


Figure 3.10: Scheme illustrating the feature extraction of track data for the track layer.

hypotheses ↘ evidences	$ev_{state\ 1}$	$ev_{state\ 2}$	$\cdots$	$ev_{state\ n}$
Track	w_{11}	w_{12}	$\cdots$	w_{1n}
No Track	w_{21}	w_{22}	$\cdots$	w_{2n}

Table 3.3: Dependence Model for Track Features.

here also determined through a priori knowledge or training data. Table 3.4 illustrates an example of a dependence model for measurement hypotheses. The acronyms for the evidences in table 3.4 corresponds the the hypothesis states for the measurements (see figure 3.6).

hypotheses ↘ evidences	ET_1	$\cdots$	ET_n	NT	RSf
Track	w_{e1}	$\cdots$	w_{en}	w_{nt}	w_{sf}
No Track	w_{e1}	$\cdots$	w_{en}	w_{nt}	w_{sf}

Table 3.4: Dependence Model for Measurement Hypotheses.

By means of the combination of all extracted features and all associated measurement hypotheses with their correspondent dependence models, the track hypotheses assertion is assured (eq. 3.9).

$$\underline{bel}_k(t)_{track} = \delta\ \underline{\lambda}_k(t)_{track}\ \underline{\pi}_k(t)_{track} \tag{3.9}$$

The term $\underline{bel}_k(t)_{track}$ in equation 3.9 describes the vector of believes by means of probability values of the possible hypothesis states for one track k in a time stamp t. Analogue to equation 3.5 the term δ corresponds to the normalization factor. It limits the track probability range between the values 0 and 1. The diagnostic support vector $\underline{\lambda}k(t)_{track}$ is composed of the combination of the incoming information of all "children" nodes (see section 2.3.3). Equation 3.10 illustrates the calculation of the track diagnostic support.

$$\underline{\lambda}_k(t)_{track} = \prod_{i=1}^{\#children} \underline{\lambda}_{k,i}(t)_{children} \tag{3.10}$$

In this sense the diagnostic support transmitted by the children nodes $\underline{\lambda}_{k,i}(t)_{children}$ is determined as follows (3.11):

$$\underline{\lambda}_k(t)_{child} = \overline{DM}_{feat|meas}\ \underline{ev}_{feat|meas} \tag{3.11}$$

where the term $\overline{DM}_{feat|meas}$ corresponds either the dependence model for track features (see table 3.3) or the one for measurement hypotheses (see table 3.4). Finally $\underline{ev}_{feat|meas}$ describes either the evidence of track features or the measurement evidences (hypotheses). It may be obtained for instance by means of equation 3.1.

Analogue to equation 3.4 the term $\underline{\pi}_k(t)_{track}$ on equation 3.9 corresponds to the causal support. It describes the incoming information from the parent nodes. In case of no parent nodes this information is replaced by the a priori knowledge about the evaluated track. Thus the assertion of track hypotheses can be illustrated by means of equation 3.12

$$\underbrace{\begin{bmatrix} T_x \text{ (Track)} \\ \overline{T_x} \text{ (No Track)} \end{bmatrix}}_{\underline{bel}_k(t)_{track}} = \delta \underbrace{\begin{bmatrix} T_x \text{ (Track)} \\ \overline{T_x} \text{ (No Track)} \end{bmatrix}}_{\underline{\lambda}_k(t)_{track}} \odot \underbrace{\begin{bmatrix} T_x \text{ (Track)} \\ \overline{T_x} \text{ (No Track)} \end{bmatrix}}_{\underline{\pi}_k(t)_{track}} \tag{3.12}$$

Another important aspect is the relevancy of the past track believes for existent tracks in the calculation of the current ones. As mentioned before and depicted in figure 3.5(b) the belief gradient assumes a low pass filter behavior dependent on the track life time. This means that the calculated belief by means of the associated measurements is timely weighted against the past track belief. As long as the evaluated track is "young", the measurements will influence strongly in its belief determination. As soon as this track reaches a specific "age", track history will be strongly considered. A specific threshold value for specific track ages depends directly on the implemented ADAS. Figure 3.11 depicts the influence of past tracks believes on the calculation of current ones. Here the influence of the past time slice $(t-1)$ is weighted against the current acquired measurements at time t.

According to figure 3.11, the final belief for a track considering its past information can be calculated by means of equation 3.13.

$$\underline{bel}_k(t)_{track,filter} = \underline{bel}_k(t-1)_{track} + G\ (\underline{bel}_k(t)_{track} - \underline{bel}_k(t-1)_{track}) \tag{3.13}$$

The term $\underline{bel}_k(t)_{track,filter}$ corresponds to the filtered vector of asserted hypotheses for the k^{th} track in the time t. The main advantage of this type of filter is that the influence of previous time slices $(t-2\ \cdots\ t-n)$ are also taken into account without being stored. It

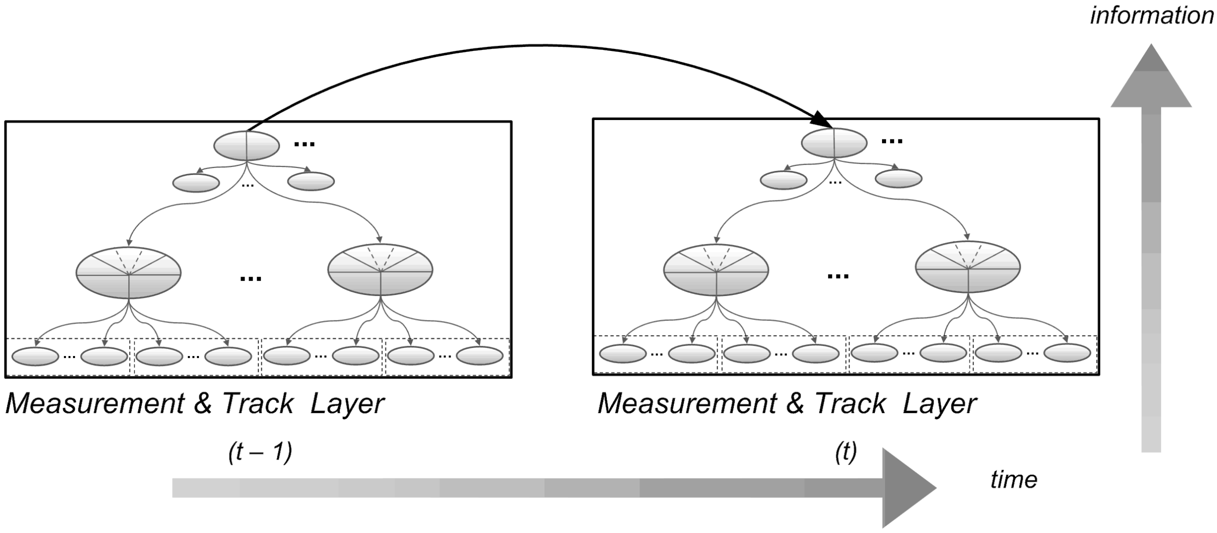


Figure 3.11: Influence of past track believes in the calculation of the current ones.

enables a reduction in the computation time and storage space. This is possible due to the current track enfolds the characteristics of the previous ones through this calculation (eq. 3.13). The term $\underline{bel}^k(t-1)_{track}$ describes the hypotheses course since the track generation until the immediately past time slice. The current assertion of track hypotheses is described by the vector $\underline{bel}_k(t)_{track}$. As mentioned before it takes all associated measurements and track features into account. Finally the variable G corresponds to the weighting factor between the current and past time slices. It determines the proportion of current and past information will affect the final calculation. There are several forms to determine the behavior of the weighting factor (G). An example already mentioned is weight past and present according to the track "maturity". Thus the factor G assumes low pass filter characteristics and can be calculated as follows (eq. 3.14):

$$G = \frac{\Delta t}{\tau} \tag{3.14}$$

where Δt corresponds the sample time or duty cycle. The factor τ describes the time (track life cycles) in which the past track believes will have a stronger influence than the current ones in the final results. According to equation 3.14 the range for the τ time has to be within the scope of: $\tau > 0$. Figure 3.12 shows an example of determination of the factor τ. The value max_τ can be again determined by means of training data of prior knowledge.

The whole process for the classification of tracks tries to reproduce the human decision method under uncertain circumstances. As long as a track is "young" its believe may strongly vary according to the quality of the associated measurements and of its features. As soon as a track reaches the expected "maturity" its quality will not vary so strong as at in its generation time. The process reflects the desired low pass filter characteristic.

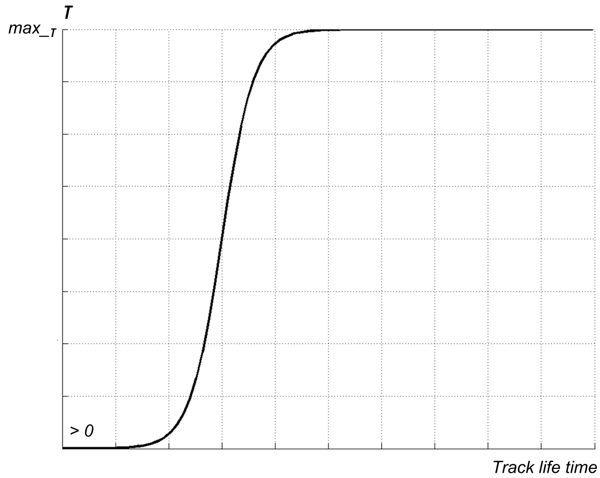


Figure 3.12: An example for the determination of the $delay$ time.

3.2.2 Quasi-Static Track Oriented Failure Detection and Classification

Quasi-static approach concerns with the detection and classification of sensor failures, which usually can not be performed within a single duty cycle. Features are extracted cyclically indeed, but the final hypothesis classification depends on a steady observation and evaluation of the whole environment. According to the intensity of these anomalies, failures may be detected and diagnosed within a short period of time. For most ADAS, these type of sensor failures are usually perceived by means of gradually system degradation. They are detected by means of longer periods of observations compared with the dynamic states (section 3.2.1). For these reasons the sensor failure layer is implemented at this stage. Figure 3.13 illustrates the class hierarchy for classifying the information in the quasi-static state level.

Analogue to the classification performed in the dynamic state, the class hierarchy illustrated in figure 3.13 describe the hypothesis states that a sensor may assume. It also consists of a multidimensional classification task depending on the number of features available. The goal here is also to extract several meaningful sensor failure symptoms that assure an unambiguous classification. Obviously the more meaningful these symptoms are, the higher is the separation level among the classes. In contrast to the classification performed in the dynamic state, the hypotheses here are not complete mutually exclusive. Several failures may occur simultaneously and one symptom may also be associated to different failures. Figure 3.13 shows the class hierarchy for a hypothetical sensor 1 that may assume m different specific and one unspecific failure states. The only case of exclusivity (either one or another class) is characterized by classes overlapping with "No Sensor Failure". A sensor can not be simultaneously in a failure free and faulty state. Although figure 3.13 illustrates the class hierarchy for only one sensor, it can be generalized for multiple sensor systems.

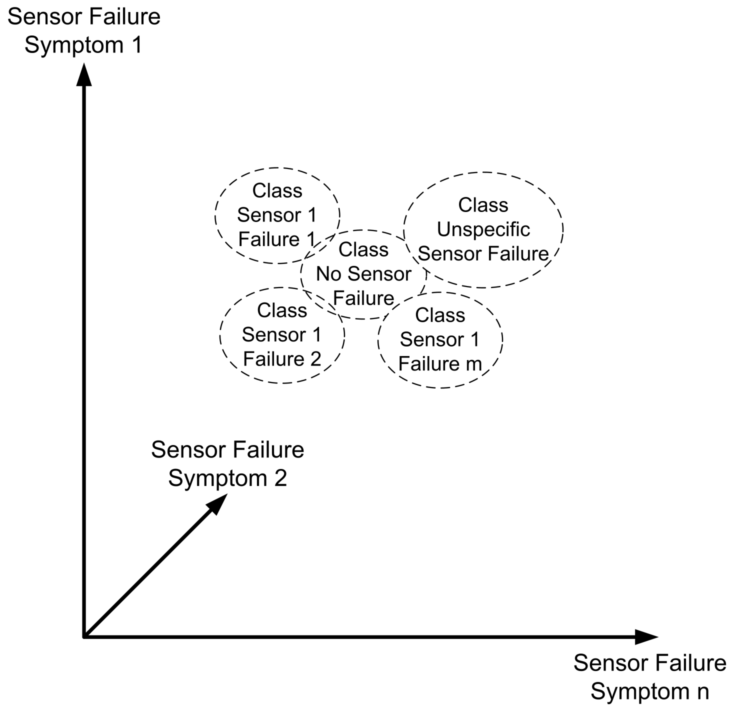


Figure 3.13: Class hierarchy for sensor failure layer.

Sensor Failure Layer

Sensor failure layer is characterized by its processing mechanism of previous data. The information obtained in the previous layers provide a basis for symptoms extraction that corresponds to sensor failures. As mentioned earlier track and measurement hypotheses will be cyclically propagated to this quasi-static state, where sensors may be classified regarding quality, reliability and integrity. Figure 3.14 shows a fragment of the probabilistic network for classifying sensor failures. For the sake of simplicity sensor failure hypotheses are abbreviated as follows:

- $SF_1 \cdots SF_m$: stand for specific sensor failures.
- $\overline{SF}_1 \cdots \overline{SF}_m$: stand for the failure free case of each specific failure.
- USF: stands for unspecific sensor failures.
- $\overline{RSF}$: stands for the failure free case of remaining sensor failure.

Another relevant aspect of this layer is related to the information propagation mechanism between dynamic and quasi-static layers. The connection between them is not performed by means of a probabilistic network approach. This means that the whole system consist basically of the mentioned two types of network.

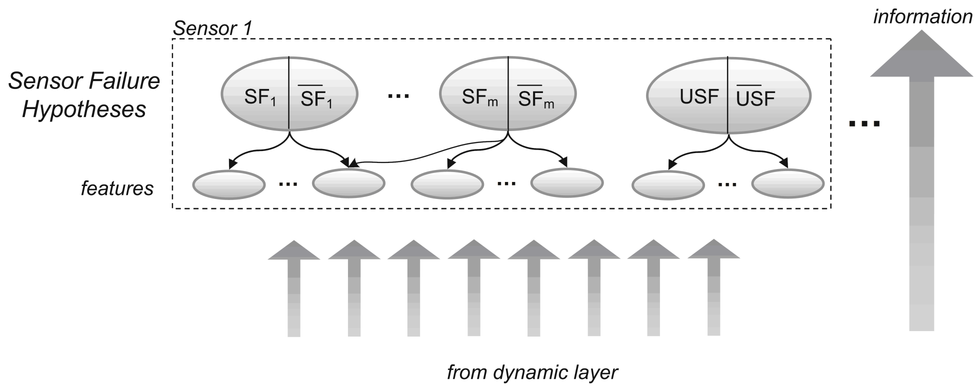


Figure 3.14: Probabilistic network for classifying sensor failures.

Similar to the assertion of hypotheses in the previous layers, the first step here consists of extracting different features. These features corresponds to failure symptoms described earlier. As decisions concerning sensor failures have to be made considering the acquisition environment as a whole, symptoms are periodically extracted indeed, but their contribution for hypothesis assertion are considered in form of occurrence events. Figure 3.15 illustrates the scheme for extracting and processing features in the sensor failure layer.

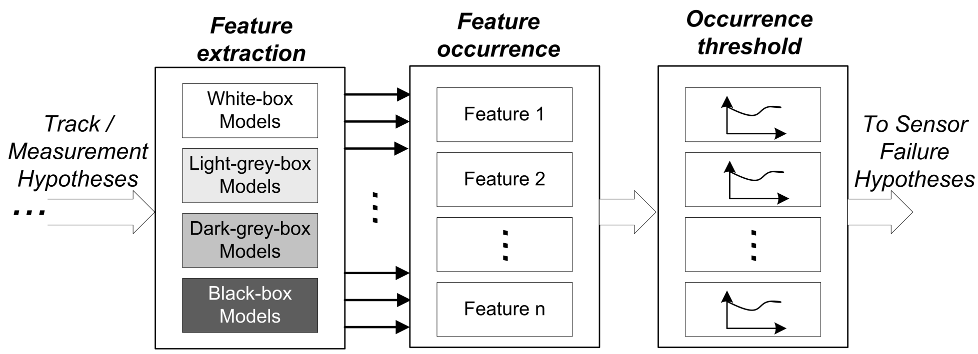


Figure 3.15: Scheme illustrating the feature extraction for the sensor failure layer.

According to figure 3.15 the cyclic information obtained from measurement and track hypotheses is the source for symptoms extraction in this layer. However they are not directly associated to sensor failure hypotheses, but firstly processed by a feature (symptom) occurrence mechanism. In order to achieve a more reliable statement about sensor failures and to save computation efforts, symptoms occurrence are first proved against threshold levels. Again threshold determination may be performed by prior knowledge, training data or a combination of both. Once feature occurrences reaches the corresponding threshold levels, hypothesis assertion can take place. As the assignment between symptoms and failures is also performed following probabilistic premises, symptoms are first evaluated according to membership functions. Equation 3.15 describes an example of a membership function. It recovers to equation 3.1 on section 3.2.1, which is an adaptation of the well-known sigmoidal function:

$$f(x)_i = 1 - \frac{1}{1 + e^{a - \frac{xa}{\alpha_i}}} \quad \forall \quad \alpha_{i-1} \leqslant x \leqslant \alpha_i(1 + \frac{1}{2}) \tag{3.15}$$

Figure 3.16 illustrates the extrapolation of equation 3.15 in order to perform symptom evaluation in a soft (fuzzy) way. According to the symptom intensity it may assume one of the determined states or even something in between. In doing so the influence of these extracted features may be taken into account in a similar way as humans deals with decision under uncertain circumstances.

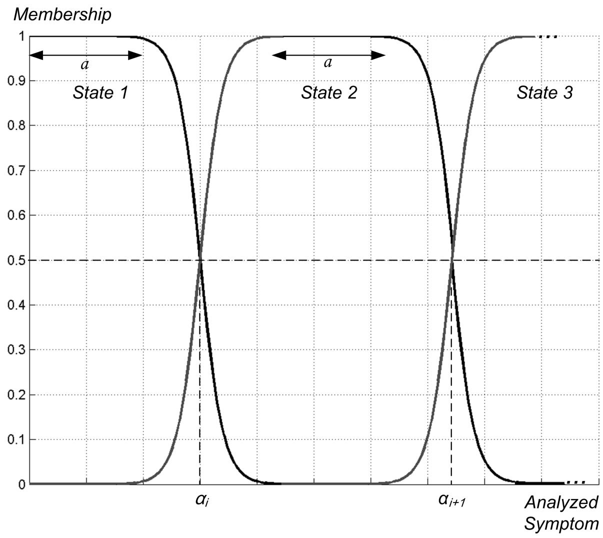


Figure 3.16: A example of a membership function for different symptom states based on the sigmoidal function.

Analogue to the symptom membership determination in the previous layers, parameters a and α for the corresponding symptoms in equation 3.15 may be achieved by means of prior knowledge, training data or a combination of both. As soon as the membership of analyzed symptoms is achieved the next processing step may be performed. In order to determine the amount of influence of a certain symptom in the classification of sensor failures dependence models are used. Table 3.5 illustrates an example of a dependence model for sensor failure symptoms.

hypotheses ↘ symptoms	$s_{state\ 1}$	$s_{state\ 2}$	$\cdots$	$s_{state\ n}$
Sensor Failure 1	w_{11}	w_{12}	$\cdots$	w_{1n}
No Sensor Failure 1	w_{21}	w_{22}	$\cdots$	w_{2n}

Table 3.5: An Example of a dependence model between symptoms and one sensor failure.

According to table 3.5 the weighting factors w_{xx} determine the amount of influence of a symptom state for classifying sensor failures hypothesis. w_{xx} may assume values between 0 and 1 describing respectively no or total dependency. As symptom states are mutually exclusive the sum of the row values in table 3.5 can not exceed 1 (100%).

Once symptoms influence are calculated, sensor failure hypotheses may be asserted. This is performed by the combination of all available symptoms related to a specific failure (equation 3.16).

$$\underline{bel}_k(t)_{SFx} = \delta \, \underline{\lambda}_k(t)_{SFx} \, \underline{\pi}_k(t)_{SFx}, \tag{3.16}$$

where term $\underline{bel}_k(t)_{SFx}$ describes the vector of believes by means of probability values of possible hypothesis states of a sensor failure x for one sensor k in a time stamp t. Based on the Bayesian theorem and analogue to the previous layers (equation 3.5) δ represents the normalization factor. Diagnostic supports are outlined here by the vector $\underline{\lambda}_k(t)_{SFx}$. It describes the weighted combination of all symptoms obtained from the correspondent children "nodes" and is calculated as:

$$\underline{\lambda}_k(t)_{SFx} = \prod_{i=1}^{\#children} \underline{\lambda}_{k,i}(t)_{children} \tag{3.17}$$

In this context the quasi-static influence of children nodes is determined as follows:

$$\underline{\lambda}_k(t)_{child} = \overline{DM}_{\,failure \mid symptoms} \, \underline{ev}_{\,symptoms} \tag{3.18}$$

The term $\overline{DM}_{\,failure \mid symptoms}$ describes in equation 3.18 the dependence model between sensor failure and symptoms (see table 3.5). Finally $\underline{ev}_{\,symptoms}$ corresponds to the symptom evidences extracted from track and measurements data and evaluated against membership functions characterized by figure 3.16. In order to determine the amount of belief in a certain sensor failures hypothesis the last term to be calculated consists of the causal support $\underline{\pi}_k(t)_{SFx}$. As mentioned earlier it corresponds to the incoming information from parent nodes. In case of the absence of parents the causal support is treated as the prior knowledge about this specific node. Therefore the assertion of sensor failure hypotheses can be achieved as follows:

$$\underbrace{\begin{bmatrix} \frac{SF_x}{SF_x} \end{bmatrix}}_{\underline{bel}_k(t)_{SFx}} = \delta \underbrace{\begin{bmatrix} \frac{SF_x}{SF_x} \end{bmatrix}}_{\underline{\lambda}_k(t)_{SFx}} \odot \underbrace{\begin{bmatrix} \frac{SF_x}{SF_x} \end{bmatrix}}_{\underline{\pi}_k(t)_{SFx}} \tag{3.19}$$

3.3 Architecture Concept

This section gives an overview of the framework architecture for the proposed approach. It describes how methods for tracking multiple targets, detecting and identifying malfunctions introduced in the previous sections can be implemented. The proposed architecture explores the reciprocity of sensor data fusion, target tracking, fault detection and identification methods. Figure 3.17 illustrates the overall scheme for the proposed framework and how it acts as an integration platform between sensor hardware and several ADAS.

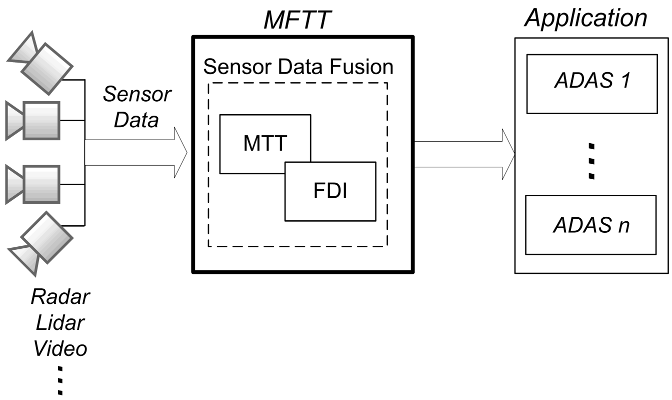


Figure 3.17: Overall scheme for the Multiple Fault and Target Tracking approach (MFTT) acting as a reliable integration platform between sensor hardware and several ADAS.

Sensor data fusion strategies offer one of the most suitable forms of dealing with data of different and dissimilar sensors by processing events in the environment (see section 2.1). It facilitates among others the implementation of different methods like MTT (see section 2.1.3) and FDI (see section 2.2) and supplies an abstraction layer between sensor hardware and different applications.

Both MTT and FDI methods are primary tasks that may be already implemented in the sensor units. However their implementation according to sensor data fusion principles enables mutual sensor supervision and thus reducing the weaknesses of single sensor systems (see section 2.1.1). Additionally an independence level between software and hardware is achieved as well. Sensor information can be delivered in a standardized format to several ADAS after their evaluation. Following sections explore the advantages of sensor data fusion, MTT and FDI for accurately acquiring and describing the driving environment information by proposing a framework approach. This framework enables the implementation of these three techniques under the theory proposed in 3.2.

3.3.1 Framework

As mentioned in the previous section sensor data fusion offers a convenient platform for implementing MTT and FDI methods. That is why the purpose of this section is to present a framework, which explores advantages and further develops both methods. This results in a practical, scalable and reusable architecture that permits the implementation the Multiple Fault and Target Tracking approach (MFTT).

This architecture design was achieved by means of an extensive evaluation of MTT and FDI methods. Table 3.6 illustrates a parallel of both techniques according to some relevant aspects. Through this comparison relevant criteria are evaluated, which allow the determination of shared properties between these two techniques. In doing so the application field of MTT and FDI can be determined for supporting this work. This means the point where these two methods can mutually complement and supervise themselves and thus allow a reliable acquisition and processing of the driving environment information.

By comparing the criteria in table 3.6 the compatibility of these methods can be perceived. They aim to increase the processed information's quality by means of complementary approaches. Thus the main goal of this proposed architecture is to explore and further develop these similar and to some extent overlapping characteristics in order to increase information quality as well as evaluating it along with the sensor hardware.

Criteria	**MTT**	**FDI**
Goal	description of object trajectories by means of measurements (if available) and predicted states	detection of faults for reconfiguration, maintenance or repair of processes
Hardware Usage	sensors cooperate with and complement each other	sensors compete against and supervise each other
Model Usage	uses the residual information to determine an optimum between measurements and predicted values	uses the residual information for the extraction of failure symptoms
Environment Information Usage	determination of object trajectories	explicit fault isolation
Information Quality	implicit used in the determination of the object trajectory	explicit use for fault detection and isolation

Table 3.6: Parallel between MTT and FDI exploring the points where they can mutually complement and supervise each other and thus allow a reliable acquisition and processing of the driving information environment.

Thus based on the premises showed in table 3.6 the proposed architecture explores and further develops fundamental principles elaborated in Naab [2004] and de Castro Bonfim [2004]. Figure 3.18 shows a schematic illustration of the MFTT approach along with a data pool concept. The main advantage of this data pool is to allow several applications to have access to the data processed on every MFTT unit. According to ADAS requirements suitable information may be chosen. This type of information can be obtained directly from sensor units, from simple processed data or by means of model based approaches. In doing so eventual information lost due to several processing steps is attenuated.

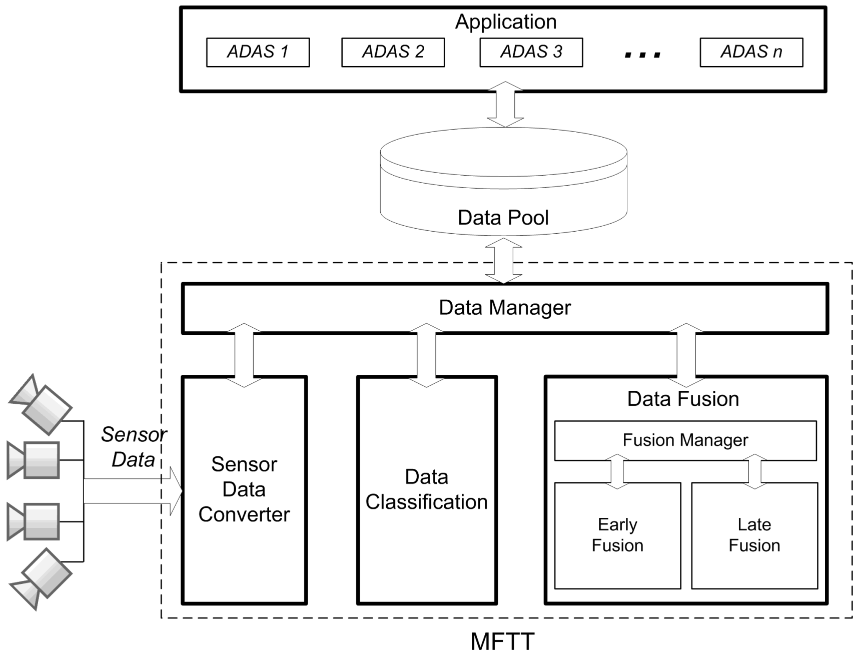


Figure 3.18: Framework architecture for performing the information fusion, tracking of multiple targets, data evaluation and sensor hardware diagnosis.

The processing flow starts with the acquired sensor information being prepared in the ***Sensor Data Converter*** unit. These results are simultaneously stored in the data pool and made available to the ***Data Classification*** unit by the ***Data Manager***. References for this classification are extracted by combining new acquired data with the fused information processed in the previous processing cycles. The final processing step is performed in the ***Data Fusion*** unit. Environment information that was interpreted and classified is fused according to a model based approach based on the methods covered previously in section 2.1. At the end of a processing cycle all interpreted, classified and fused information are made available for several ADAS.

In the remaining of this chapter the processing units illustrated in figure 3.18 along with their peculiarities will be covered in details. The procedures for the implementation of the MFTT theory introduced in section 3.1 and 3.3 will be explored.

3.3.2 Sensor Data Converter Unit

Sensor Data Converter unit consists of mechanisms that deals as a gateway between hardware and data abstraction layer. This means that sensor configurations may vary, but classification and fusion algorithms remain unchanged as long as the information contents are unaltered. Main task of this processing unit involves the data manipulation of dissimilar sensors in such a form that it can be processed meaningfully. Different sensor hardware may work under distinguished time constraints and coordinate systems. Therefore this sensor information must be transformed to a unified description system and thus enabling its processing.

Another relevant task consists of interpreting and forwarding requirements of different ADAS or of the remaining processing units to the sensor hardware. Depending on the field of interest of several applications sensor units can be online configured or even part of their information may be evaluated. A typical example consists of combining radar and video based sensors. Radars may indicate the field of interest for image processing and thus reducing computational efforts.

In case of identifying sensor faults or failures unambiguously, complete or part of the incoming information of the respective sensor unit may be ignored. This would increase systems reliability due to a faulty sensor not being taken into account or being repaired for the fusion and target tracking processes. A typical example can be illustrated by the detection of a misaligned sensor. Once this failure could be clearly identified the acquired data may be ignored or even aligned if the shift angle is known.

3.3.3 Data Classification Unit

Although sensor hardware units already perform a certain level of data classification themselves, most of them are not designed to implement enhanced (high level) classification tasks. Reasons for that are the limitations due to measurement principles or computational efforts. For instance the assertion of exclusive radar measurements to cars or small trucks is hard to be correctly achieved. Additionally the detection of malfunction with a single sensor is difficult as well. Therefore the combination of dissimilar sensors is a straightforward method to perform the tasks mentioned before.

In this unit measurements will be assigned to several hypotheses. The specificity of these assumptions depend obviously on the level of observability supplied from the applied sensor units. This means that this specificity is directly related to type of attributes obtained from sensor measurements. The main goal here is thus to cyclically describe sensor data according to the information available. In doing so multiple targets can be tracked as well as sensor malfunctions can be detected. Basis for it is the consistence check or redundancy described in section 2.2.2. Figure 3.19 illustrates schematically the data classification unit. It represents the three main classification layers namely measurements, track and sensor failures within the correspondent dynamic and quasi-static states.

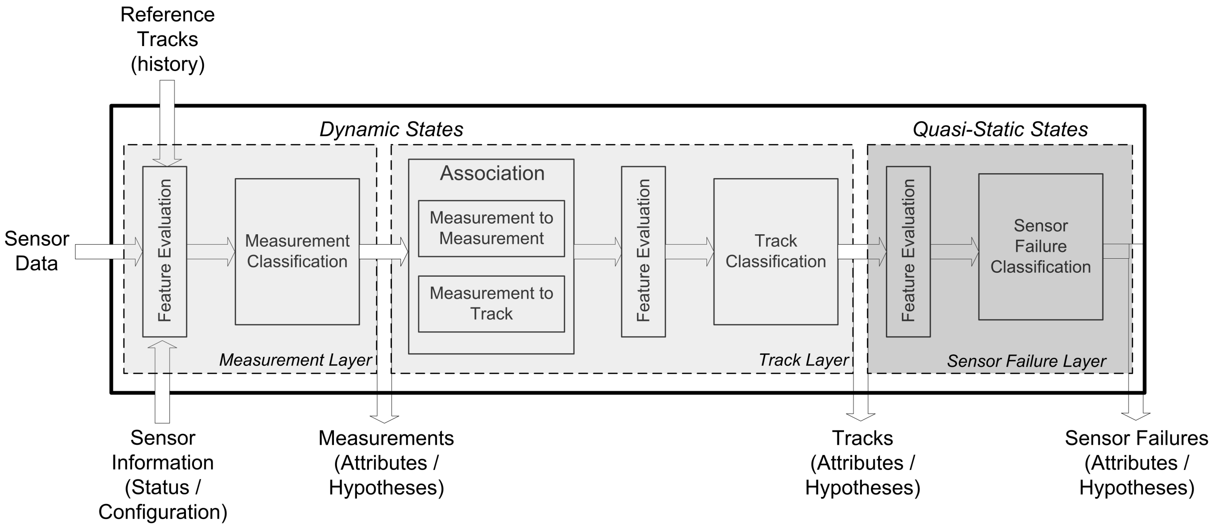


Figure 3.19: Schematic illustration of the data classification processing unit representing the dynamic and quasi-static classification of sensor data. The three layers for detecting and identifying faults and failures are represented as well.

First step for performing the classification of environment information consists of the extraction of symptoms or features from current acquired sensor data. Hereby several plausibility models are built with aid of reference tracks determined in previous cycles along with current information about sensor hardware status and configuration. Examples of plausibility models will be discussed in details on section 4. According to the symptoms or features extracted sensor data will be evaluated regarding their integrity, quality and reliability. This is performed by means of the previously described probabilistic approach. As mentioned earlier sensor data may assume following mutual exclusive states:

- existent tracks
- new tracks
- specific sensor faults
- unspecific sensor faults

After this first classification both measurement attributes and hypotheses are made available to several applications by means of the data manager in the common data pool (see figure 3.18). Concurrently this processed information deals as input for the next processing step namely the track layer.

In the track layer two types of association take place according to the multiple target tracking approach discussed in section 2.1.3. In case of non existent track hypothesis dealing as reference for the measurement association, measurements will be associated one another. This measurement-to-measurement association describes the track initiation task. Associated measurement attributes themselves describe this new track. The second type of association consists of the measurement-to-track one. In this case existent reference tracks will be maintained in an update step and associated measurements will contribute to track trajectory determination.

Previous sensor data classification in the measurement layer plays a crucial role in the association task. According to the probability associated to a measurement hypothesis state the correspondent measurement will contribute to the asserted track level of belief. Assuming a measurement m_1 with following state probability distribution $p(m_1) = \{0.7, 0.2, 0.1, 0.0, 0.0\}$ for the respective state vector $\underline{s}$:

1. s1: existent track 1
2. s2: existent track 2
3. s3: new track
4. s4: specific sensor fault
5. s5: unspecific sensor fault

the measurement m_1 will contribute with 70%, 20% and 10% certainty for the level of belief of respectively existent track 1, existent track 2 and new track. The membership calculation of every single measurement along with the feature extraction of past reference tracks characterize the feature evaluation unit within the track layer. According to the symptoms or features evaluated tracks will be classified regarding their integrity, quality and reliability as well. This is performed by means of the previously described probabilistic approach. As discussed previously tracks may assume following mutual exclusive states:

- track
- no track

Another relevant aspect for the assertion of track hypotheses is the extraction of track symptoms and the calculation of their influences. Examples of track symptoms may be considered as plausibility criteria in order to validate tracks. The specifity of a track state (cars, trucks, cyclist, guard rail, etc) depends strongly on the nature of the sensor data. Similar to the results obtained in the measurement layer track attributes and hypotheses are made available to further processing steps by means of a data manager in the common data pool. The evaluation of measurement and tracks characterizes the dynamic state classification due to the association between measurement and tracks being performed in every single cycle.

By classifying measurements and tracks cyclically evidences for the quasi-static layer (sensor failures) can be attained. As mentioned before in section 3.2.2 the information obtained from the previous layers is processed by a symptom occurrence mechanism. This is necessary because sensor failures have to consider the occurrences in the environment as a whole and not only based on isolated situations. For each sensor incorporated to the system the following classes will characterize this layer:

- specific sensor failures
- absence specific sensor failures
- unspecific sensor failures
- absence of unspecific sensor failures

Based on the probabilistic approach for the assertion of sensor failure hypotheses described in section 3.2.2 sensor classification is performed. Also in this layer symptoms are extracted cyclically. However only after thresholds levels for symptom occurrences are exceeded their influences on the hypothesis states are calculated. As mentioned earlier the determination of the specific threshold level for each symptom has to be determined by means of prior knowledge, learning data or a combination of both.

3.3.4 Data Fusion Unit

As discussed in section 2.1 Sensor Data fusion supports a meaningful combination of incoming data of several dissimilar and partially redundant sensors. Therefore the main goal of a Data Fusion unit in the MFTT context is to provide a mechanism to explore, adapt and enhance this data combination. In doing so the tracking of multiple targets in the driving environment as well as the generation of evidences for the detection and identification of faults can be implemented.

Early Fusion

Early or low level fusion unit represents in the MFTT context the processing stage that summarizes the information obtained from the association algorithms. Its main goal consists of fusing sensor information by avoiding strong data manipulation. It is a mechanism to formalize the previous data classification by exploring the redundancy as well as the most relevant features of several sources of information. In doing so superfluous information may be reduced and further applications may perform their own specific fusion algorithms according to their needs.

This stage builds a kind of memoryless fusion processing. This means that results obtained at this processing level do not take the observation history of objects in the driving environment into account. Thus only current measurement attributes are considered. It describes the building process of low level tracks. These low level tracks are considered by the remaining processing stages like Late Fusion or several ADAS as a virtual sensor. Information will be summarized into an essential description format building a sort of segmentation mechanism. In doing so computational efforts in the following processing level may be reduced.

Example of this early fusion mechanism may be described by the generation of low level tracks by means of a Center of Gravity (CoG) approach. CoG consists of building low level tracks by calculating an absolute mean value based on the associated measurement attributes. Figure 3.20 illustrates the early fusion principle according to the CoG approach. Radar measurement attributes like position and velocity are merged and generate a new virtual measurement namely a correspondent low level track.

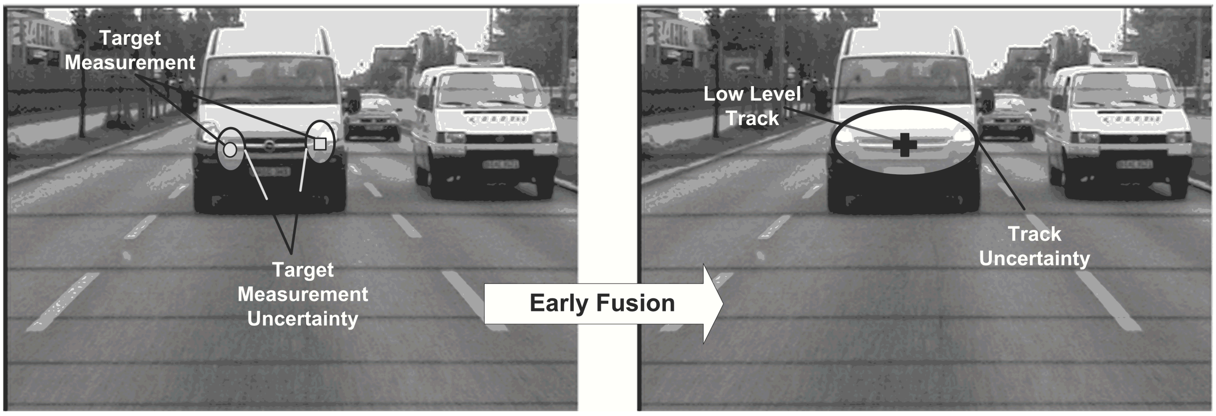


Figure 3.20: Schematic illustration of an example of the early fusion mechanism applying the Center of Gravity (CoG) approach. The mean of radar measurement attributes like position and velocity are calculated and generate a new virtual measurement namely a correspondent low level track.

According to the type of information provided by different several units like object edges or consistency the approaches for early fusion may increase in complexity. Several strategies and algorithms may be applied here in order to reduce the entropy of the acquired sensor data.

Late Fusion

Once the essential driving environment information could be processed and extracted by means of previous processing units, the model based approach for tracking multiple targets may be accomplished. This preprocessed data is considered as the input of several virtual sensors and as basis for performing the late fusion in the MFTT context.

Late Fusion unit represents the meaningful combination of this virtual sensor information taking the observation history of objects in the driving environment into account. In this sense a model based approach for tracking these multiple targets as well as for generating references for the detection and identification of faults is achieved.

In this unit white-box models are applied to perform the estimation of modeled object state variables like position, velocity, acceleration, shape and so on. These estimation results are denominated high level tracks, which deal as reference for the data classification and posteriorly for the early fusion units. According to the estimation process described in section 2.1.2 high level tracks are filtered and predicted in this processing layer. Filtered high level tracks deal as reference for current actions and reactions of several ADAS as well as evidences for the detection and identification of sensor faults. On the other hand predicted high level tracks serve as basis for measurement to track association performed in the data classification unit.

Algorithms applied for the estimation of high level tracks may be based on Kalman, $\alpha - \beta$, Bayesian and particle filter approaches. An overview about this type of filters along with the the references for a detailed description were given in section 2.1.3.

Fusion Manager

Fusion manager is concerned with pruning track hypotheses as well as performing track-to-track associations. As mentioned before Tracks are employed to describe the motion of the targets dynamically. Criteria to exclude a track can be chose according to the following assumptions in an isolated or combined form:

- **oldest survey:** to each track is a life time counter associated. Therefore it can be assumed that the track with greater "experience" should inherit the measurements of the younger track in the next cycle.
- **strongest survey:** a track that contains fewer target measurements associated can be excluded. This tactic enables the chose of more robust track.
- **track fusion survey:** the attributes of both tracks are fused using again a model based approach.

3.4 Summary

This chapter described the proposed concept for improving quality, integrity and reliability of the driving environment information by Multiple Fault and Target Tracking. It enables the fusion of sensor data, tracking of multiple targets, the track oriented evaluation of sensor information and the supervision and diagnosis of sensor hardware.

Premise for the developed concept is the classification of the acquired information of sensor hardware acquired in the driving environment. For this purpose probabilistic networks were adapted and further developed. In doing so a new network configuration for handling with sensor measurements, track and sensor data was developed. It consists of two main states: dynamic and quasi static. The dynamic state network implements not only a cyclically assertion of hypotheses, but also a dynamic connection between sensor measurements and tracks. This describes and fits the dynamic relationship between measurements and tracks. On the other hand the evaluation of sensor hardware is performed by a quasi-static probabilistic network implementation. Similar the detection of sensor hardware failures the occurrences on the driving environment have to considered as a whole, the assertion of hypotheses here has to be performed in longer period of times.

According to the premises of the proposed concept a framework architecture for its implementation was developed as well. It explores the advantages and to some extent overlapping characteristics of both multiple target tracking and fault detection and identification methods. By comparing, adapting and further developing algorithms of both strategies a robust approach for improving the quality of information acquired in the driving environment was accomplished. One important goal is to allow today's considered comfort driver assistance functions to make a further step toward safety related and to some extent autonomous applications.

4

Case Study

In this chapter the particularities of the development and validation platform for the presented approach are discussed. It gives an overview about the conditions and constraints of the environment used to proof the proposed concept covered in section 3. The configuration of the applied sensor network implemented in the experimental vehicle will be described. Furthermore the measurement principles of the applied sensors will be summarily outlined.

4.1 Experimental Vehicle

The test vehicle used to perform the experimental test runs in the driving environment consists of a vehicle prototype equipped with a decentralized sensor network and a central processing unit for the execution of the implemented algorithms. Figure 4.1 gives a schematic overview about the complete coverage area of the applied sensor network.

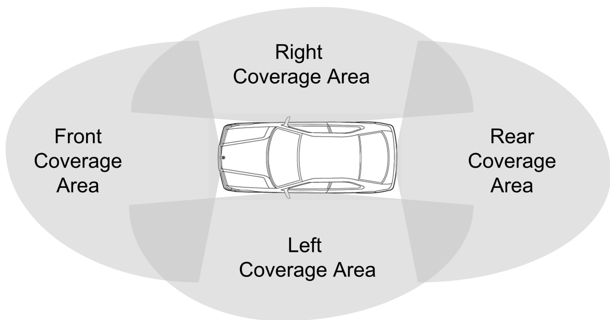


Figure 4.1: Schematic overview of the achieved coverage area with the applied sensor network.

This sensor network distribution was proposed in Rasshoffer and Gresser [2004] in order to attend the requirements of the future generation ADAS, which will have to some extent increased requirements regarding safety and reliability. It consists of a lidar (LIght Detection

And Ranging) sensor network composed of six units and of an additional radar (RAdio Detection And Raging) sensor network composed of three units. Furthermore a video based network consisting of two camera units is available as well. However camera systems were not applied for feature extraction, but only for visualization and validation purposes.

The applied lidar sensor network consists of a long and mid range configuration assembly for the front and rear field, while the lateral fields are covered by two specific lidar scanner sensors, which mechanically rotates in order to scan the driving environment. The beam geometries of lidar's front and rear configuration are illustrated in figure 4.2.

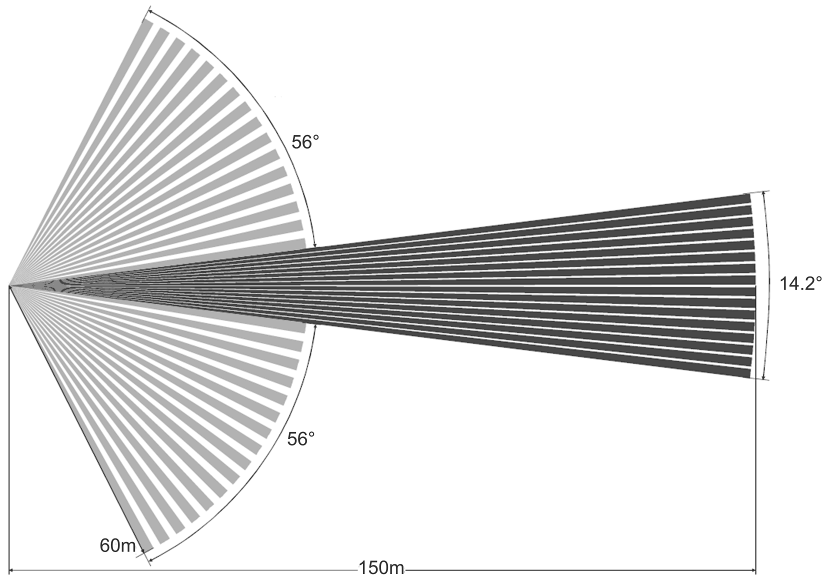


Figure 4.2: Beam geometry of lidar's front and rear configuration.

The front and rear sensor geometry field are shared in a mid and a long range part. Mid range amounts 60 meters and is shared again in two small sectors with an aperture angle of 56 degrees. Both sectors are characterized by 16 channels each with a aperture angle of 2 degrees. The gap between each channels amounts 1.6 degrees. The long range area covers a maximum distance of 150 meters with an aperture angle of 14.2 degrees. It also consists of 16 beams with an aperture angle of 0.7 degrees and a gap of 0.2 degrees between them. Specific properties of the applied lidar sensor units will be discussed in details later on in section 4.2.2.

Both lidar scanner units are characterized by the beam geometry illustrated in figure 4.3. The range for the applied sensor unit reaches a maximum of 20 meters with an aperture angle of 160 degrees. The field of view is shared in 16 beams with aperture angle of 5 degrees each. The gap between the channel amounts 5.4 degrees.

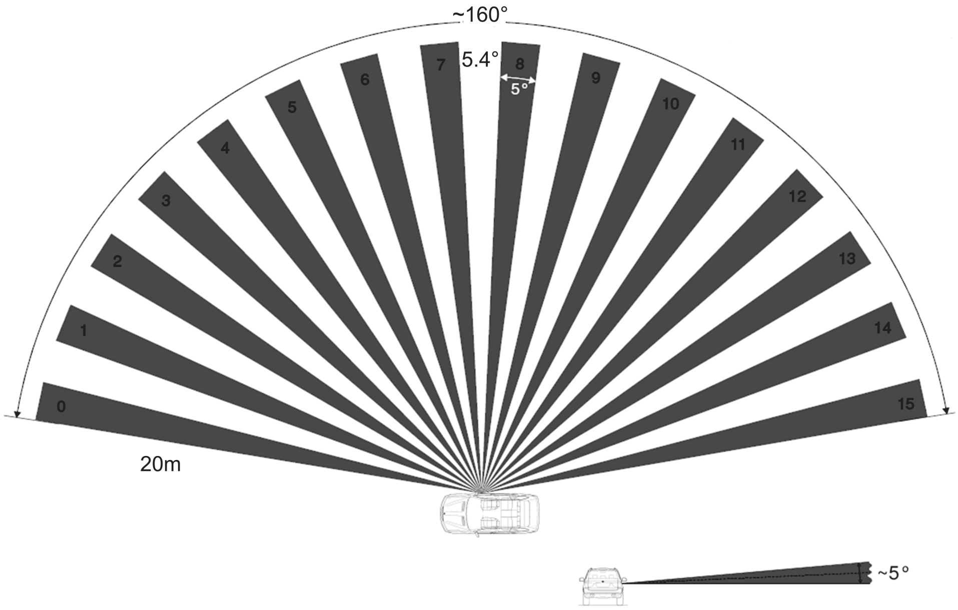
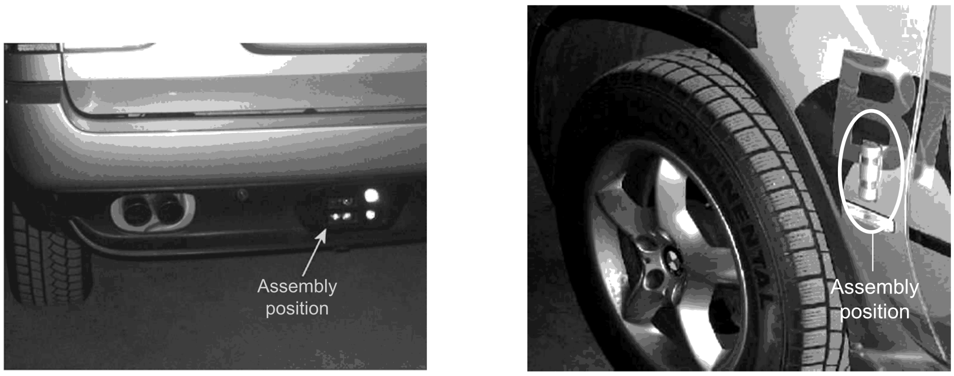


Figure 4.3: Emission geometry of lidar's lateral configuration.

The assembly position for rear and front lidar sensors is the front and rear bumpers respectively, while the lateral sensor are assembled in the vicinity of the rear view mirror. Figure 4.4 shows the actual assembly position of the applied lidar sensors.



(a) Assembly position of rear lidar sensor units

(b) Assembly position of lateral lidar sensor units

Figure 4.4: Assembly position of lidar sensor units in the experimental vehicle.

radar sensor units applied to the actual vehicle prototype are based on 77 GHz and 24 GHz technologies. They consist of one long range radar operating at 77 GHZ with a maximum range of 200 meters and an aperture angle of approximately 20 degrees. Two short range radar units operating at 24 GHz cover a maximum range of 50 meters with an aperture angle of 45 degrees. These three radar sensor units are also positioned in the front of the vehicle prototype.

They provide an extra redundancy source to the lidar sensor units with dissimilar measurement principles. Technical details of the applied radar units will be discussed later on in section 4.2.1. The beam geometry of radar's front configuration is illustrated in figure 4.5.

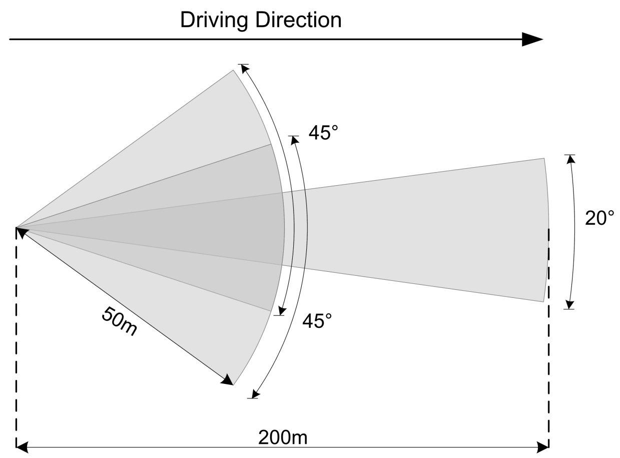


Figure 4.5: Beam geometry of radar's front configuration.

A complete overview of the assembly position of all sensor units is represented in table 4.1.

Position	x_0	y_0	z_0	Azimuth Angle
lidar Front	0.1m	0.0m	-0.4m	0°
lidar Front Left	0.1m	0.0m	-0.3m	35.1°
lidar Front Right	0.1m	0.0m	-0.3m	-35.1°
lidar Rear	-4.7m	0.0m	-0.3m	180.1°
lidar Rear Left	-4.7m	0.0m	-0.3m	144.9°
lidar Rear Right	-4.7m	0.0m	-0.3m	-144.9°
lidar Left	-1.3	0.92	0.025	90°
lidar Right	-1.3	-0.92	0.025	-90°
radar (77GHz)	0.1	-0.28	0.3	0°
radar (24GHz) Left	0.1	0.5	-0.3	0°
radar (24GHz) Right	0.1	-0.5	-0.3	0°

Table 4.1: A complete overview of the assembly position of all applied sensor units.

Thus the origin of the common coordinate system for the sensor network is located at $x = -0.1m$, $y = 0m$ and $z = -0.6$ relative to the ground and to outermost front part of the vehicle (see figure 4.6).

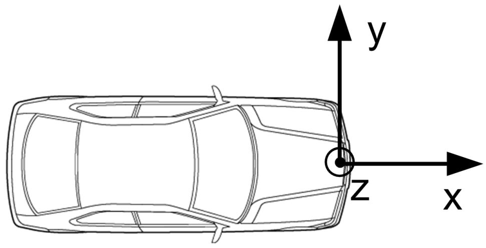


Figure 4.6: Common reference coordinate system for the vehicle prototype relative to the ground and the outermost front part of the vehicle.

4.2 Measurement Principles

4.2.1 Radar

The acronym radar stands for RAdio Detection And Ranging. It has been employed in the automotive field for the determination of the position and velocity of reflective objects in the driving environment. As mentioned previously the most established techniques for long and short range radars in the automotive industry are based on 77 GHz and 24 GHz technologies respectively. Driver assistance functions, which do not require a broad field of view of the environment (e.g. ACC), for instance within highways, employ long range radar. On the other hand assistance functions for urban applications require additional short range ones. Due to their wide aperture angle they enable an enhanced field of view in dense traffic situations. The advantage of radar sensors is that they are not strongly susceptible to the weather conditions, like rain, fog or snow.

Essentially radar function is based on the principle of transmitting and receiving electromagnetic waves in the respective frequencies. The determination of the target parameters like distance, relative velocity, position angles are achieved through the combination of several physical effects, which will be briefly outlined. A detailed description of radar measurement principles can be found in Barton and Leonov [1998].

Object reflectivity is the basic and necessary effect for the determination of target measurements. Electromagnetic waves are transmitted by means of antennas in the frequency range mentioned before. These waves are reflected when they achieve conductive materials and transmitted again as echoes. The angles of incidence are determined using this echoes as well. Most of the time they detect the edge of the objects and jump all the time over the complete coverage area. Obviously reflection effect induces also to a time difference between receiving and transmitting of the signal. This difference is called wave runtime and enables the calculation of the distance to the detected target.

By means of wave runtime the distance d between the emitter and targets can be determined. It can be peformed by emitting a short radar impulse, which the duration amounts $\Delta\tau$. After a determined interval of time τ, the observation time of the receiver is set to the interval of time of the transmitted impulse ($\Delta\tau$). Then the distance d between target and sensor unit for a direct reflection way can be determined by:

$$d = \tau\frac{c}{2} \tag{4.1}$$

where c means the speed of light. In this case the complete energy of the impulse falls into the centred measurement cell of the emitter. In the case that the measured target is placed between $[(-\Delta\tau+\tau)\frac{c}{2}, (+\Delta\tau+\tau)\frac{c}{2}]$, the energy would be shared among the nearest cells. Making use of these properties, the so-called distance profile can be created. It can be built by repeating the process outlined before, but increasing the observation time of the receiver until the time step τ_{step}. After scanning the environment, the actual target distance can be determined by means of the biggest signal amplitude registered in the distance profile (figure 4.7). This method is also called simple pulse radar.

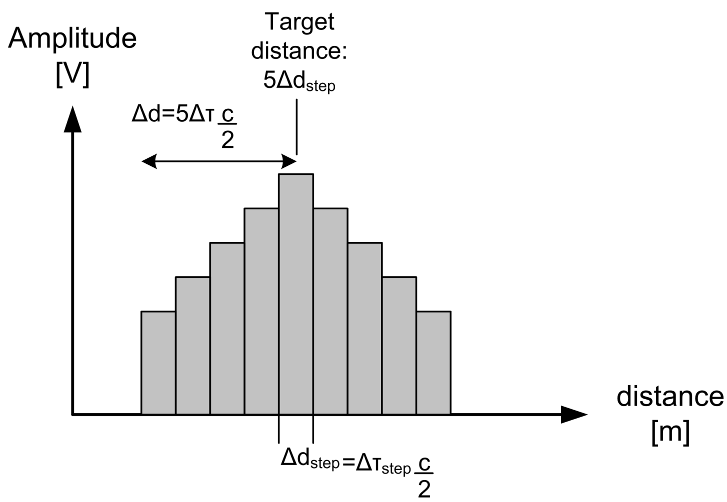


Figure 4.7: Example of a distance profile of a radar sensor.

The main weakness of this method consists on the absence of information about the frequency shift of the received signals and consequently the relative velocity cannot be determined. Furthermore the resolution's quality and the received energy is limited, because of the short receiver's observation time. However a short impulse in time implies a broad spectral expansion in frequency. This means, the resolution's quality depends on the signal bandwidth [Barton and Leonov, 1998]. For these reasons a direct wave runtime measurement is expensive and hard to be achieved. The difficulties and weaknesses from this method can be reduced by employing the Doppler effect on frequency shifts and suppressed by several frequency modulation procedures.

By means of the Doppler effect frequency shifts can be achieved and consequently the relative velocity can be determined. This effect occurs if the transmitter and receiver are moving relatively from each other. In the case that both are mounted in the same device and also moving against a conductive target, the frequency f_D of the transmitted signal is displaced.

This happens due to alterations of the relative velocity of the reflecting signal and can be calculated by:

$$f_D = -2f_C \frac{v_{rel}}{c} \tag{4.2}$$

where f_C represents the carrier frequency. However by this method the receiving signals have to be tracked during a considerable period of time in order to obtain enough measurement points for a spectral analysis. Optimized methods can be achieved by combining the Doppler effect with frequency modulation procedures.

Frequency Modulation Continuous Wave method (FMCW) pursuits the determination of the target parameter and also reduces the issues mentioned before. Instead of transmitting single impulses, the transmitter sends continuous signals for a couple of milliseconds, so that enough signal energy, during a sufficient interval of time, can be detected. Thereby the target's relative velocity and distance can be determined.

The time-lag between sent and received signal is no longer compared. Instead the signal frequencies are compared by means of the modulation of the frequency transmitted. This modulation allows the expansion of the bandwidth in order to increase the resolution's quality. The signal can be modulated using ramp or triangle functions as illustrated in figure 4.8. Thereby the frequency transmitted is described by the flank lead coefficient $m = \frac{df}{dt}$. During the interval between transmission and reception (runtime: $\tau = \frac{2d}{f}$), the transmitted signal is shifted by an amount of $\Delta f = \tau m$. This frequency deviation is then filtered by a mixer and subsequently by a low-pass filter. The spectral peak of Δf corresponds then the distance to the measured target:

$$d = \Delta f \frac{c}{2m} \tag{4.3}$$

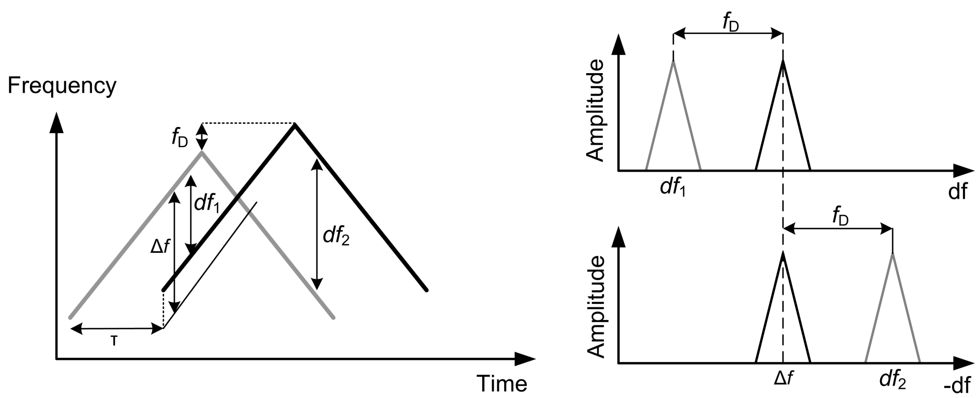


Figure 4.8: Schematic illustration of the FMCW detection method with $v_{rel} < 0$, one target approximating to the sensor.

Chapter 4

Thus the frequency deviation can be determined because of the Doppler shift. If a measured target approximates to the transmitter, the frequency of the reflecting signal increases. On the other hand if a target is getting away from the transmitter this frequency decreases. So the frequency deviation is determined by:

$$df = f + \Delta f - (f + f_D) = \Delta f - f_D \tag{4.4}$$

where f describes the frequency of the original (transmitted) signal while Δf the frequency deviation due to the modulation and f_D the frequency shift due to the Doppler effect. Therefore the frequency shift depends on the target's relative velocity and distance:

$$\frac{v_{rel}}{d} = df \tag{4.5}$$

According to equation 4.5 no definite solution for the determination of the relative velocity and distance. That is why these values are determined by means of a linear combination of each other. This combination is obtained making use of the so-called "plain of distance versus relative velocity" (DV) [Barton and Leonov, 1998]. It results in a straight line with negative flank lead represented over this plain. By applying another FMCW modulation with a negative ramp function the outcome is another straight line, but with positive flank lead. The intersection of both lines determines the distance and relative velocity values (figure 4.9).

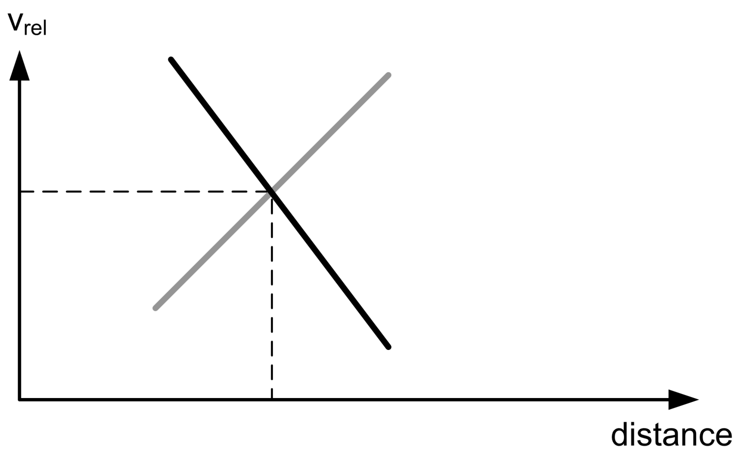
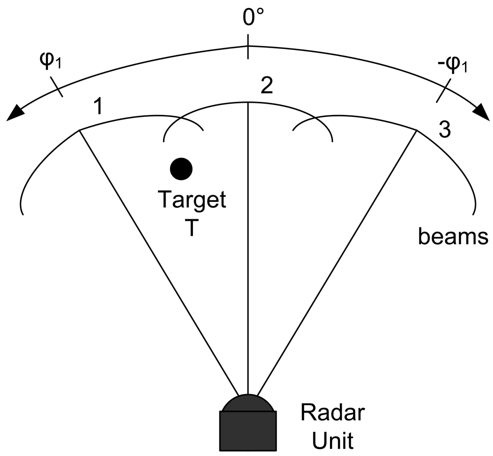


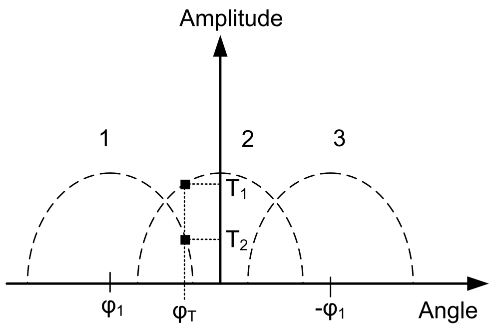
Figure 4.9: Schematic illustration of an example of a distance versus relative velocity diagram.

In order to detect multiple targets several ramp functions with different frequency modulations are employed [Barton and Leonov, 1998]. However the explicit determination of position and relative velocity have to be performed with at least three ramps.

The determination of the angle of a acquired target in the driving environment consists of emitting multiple radar beams (at least three), which overlap each other. This radar beam will be then acquired by one receiving antenna. An alternative for it is based on the emission of one beam, but its intensity is measured making use of multiple antennas. In both cases the more beams or antennas employed, the greater is the measurement resolution. Thus with aid of an antenna diagram the received signal amplitudes can be illustrated and consequently the target angle can be calculated. Figure 4.10 illustrates an example using three radar beams for the target angle determination. This is possible, because each antenna coverage area is shared in two possible angular sectors. For instance if an object can be observed by antennas 1 and 2, the measured angle should fall within their field of vision and should assume an average value according to the amplitude of the reflected signals. Then the intersection point of the amplitudes T1 and T2 with their respective beams determines the target angle. Figure 4.10 depicts the angle calculation of a hypothetical object.



(a) Example illustrating a radar sensor unit with three emission beams and a hypothetical target being acquired between the first and second radar beam.



(b) Example illustrating the target angle determination by means of an antenna diagram.

Figure 4.10: Hypothetical intern angle calculation of an object measured in the driving environment by a radar sensor unit.

Radar Peculiarities and Typical Faults

Radar units applied for validating the proposed concept discussed in section 3 show some peculiarities and induce some typical faults that will be briefly covered on this section.

The detection of targets in the driving environment is directly dependent on the signal to noise ratio of the correspondent signal received. Receiving power P_R depends essentially on the distance d between radar unit and object and on the radar cross section (RCS) ζ of this target [Barton and Leonov, 1998]. It can be represented in a simplified form as follows:

$$P_R = \frac{\zeta}{d^4} \tag{4.6}$$

The RCS value depends directly on target's geometry and surface structure and additionally on the angle of incidence and reflexion of the radar beam. Table 4.2 shows the typical RCS values obtained from the most relevant target in the driving environment for ADAS [Barton and Leonov, 1998].

Target Objects	RCS
Pedestrian	$\approx$ 0 dB m^2
Motorcycle	$\approx$ 10 dB m^2
Cars	$\approx$ 20 - 30 dB m^2
Trucks	$\approx$ 30 - 40 dB m^2

Table 4.2: Typical RCS values obtained from the most relevant target in the driving environment for ADAS [Barton and Leonov, 1998].

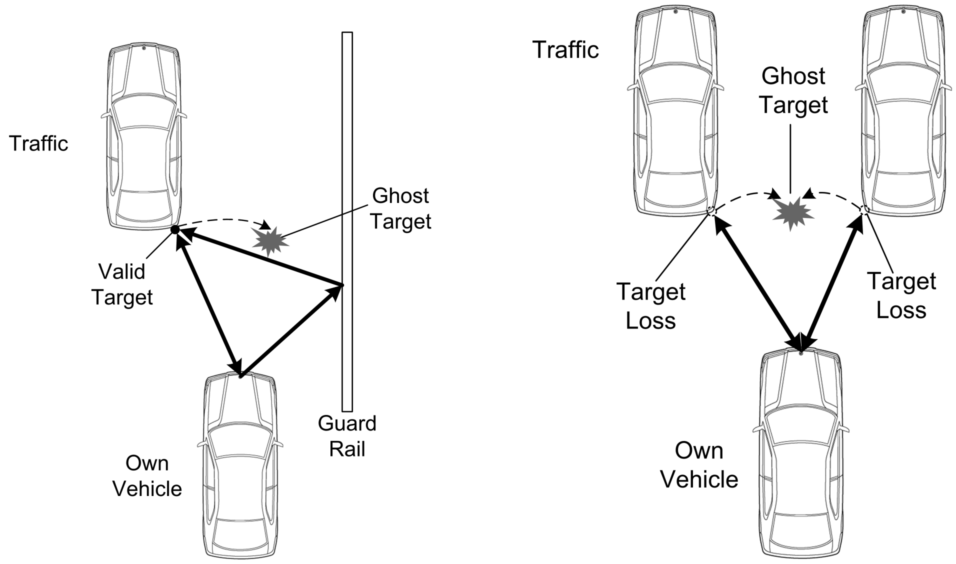
Although the weather influence on radar technology is not so strong if comparing with lidar systems, strong rain and snow along with the distance between radar unit and target may attenuate the receiving signal [Barton and Leonov, 1998].

Typical sensor faults for radar systems consist essentially of the loss from valid target objects and of the generation of inexistent ones (ghost targets). These kind of faults are associated to the measurement principles and their causes can be categorized in the three main topics as follows [Barton and Leonov, 1998]:

1. target loss or generation of ghost targets due to signal interference (multipath fading).
2. target loss and inexistent target generation because of the angle determination method with the multiple beam approach.
3. inexistent target generation due to the multiple ramp approach with the FMCW method.

By means of the interference generated by the multiple paths that a radar signal may travel in the driving environment (topic 1), valid object targets may be lost or even inexistent targets may be generated. A constructive signal interference in the radar unit antenna may generate inexistent targets while a destructive interference may extinguish valid measurements from the driving environment. Figure schematically illustrates the generation of inexistent targets due to multipath fading. It illustrates a hypothetical constructive interference when radar antenna receives a direct signal reflection from the target vehicle and simultaneously from the guard rail.

The simultaneous loss of valid targets along with the generation of inexistent measurements due to the poor angle resolution of static radar units (topic 2) is one of the most crucial weakness of this kind of radar systems. Typically this type of fault occurs when two objects are travelling in the driving environment side by side with similar velocities. By determining the angle position of these two target objects with the multiple beam approach (figure 4.11) their amplitude will interfere mutually, so that their signal will be extinguished and an inexistent target will be generated between them. A detailed description of this effect is covered in Sauer [2001].



(a) Scheme illustrating the generation of inexistent targets due to multipath fading. It shows a hypothetical constructive interference when radar antenna receives a direct signal reflection from the target vehicle and simultaneously from the guard rail.

(b) Scheme illustrating the simultaneous generation of inexistent targets and the loss of valid ones due to poor angle resolution of static radar units. It shows two vehicles travelling side by side while a ghost target between them is generated and their correspondent targets are extinguished.

Figure 4.11: Typical radar fault for static units (no scanner properties).

This discussed information about the peculiarities and typical sensor faults deals as a source of information for validating sensor measurements and posterior tracks and sensor failures. As mentioned before from the point of view of several ADAS than detection of these faults is rather important then their identification. Therefore an exact model describing these undesired effects can be substitute by models covering plausibility criteria for detecting these faults (see section 5.1). Furthermore an exact model of these undesired effects is sometimes not practicable due to most of the necessary information to extract the correspondent features not being available.

4.2.2 Lidar

The acronym lidar stands for LIght Detection And Ranging. It has been recently employed in the automotive field for determining the position of target objects in the driving environment. The applied technology consists essentially of a laser device with emission wavelength typically at 900 nm. Its main advantage compared radar systems consists of a higher measurement resolution due to shorter wavelength properties. On the one hand shorter wavelengths in the radio spectrum make possible a more precise detection of small objects, but on the other hand the necessary emission power for detecting very small objects is hard to be implemented to match automotive requirements. Therefore the emission wavelength at 900 nm offers a suitable tradeoff between power emission (eye safety) and the automotive requirements by detecting objects like pedestrian and vehicles in the drive environment.

The detection of objects by means of lidar systems is performed by determining the time of flight (TOF) from a transmitted infrared pulse between sensor unit and target object. As soon as a transmitted lidar beam hits a target object, this pulse is reflected in different directions. As only a small part of the transmitted signal is received again by the sensor unit, the object detection depends strongly on the signal to noise ratio. Thus the distance d between lidar sensor unit and target object can be achieved analogue to the pulse radar approach showed in figure 4.7 and characterized as follows:

$$d = \tau \frac{c}{2} \tag{4.7}$$

where d represents the radial distance between sensor and target while τ describes the measured TOF and c represents the speed of light. The angle position of target objects may be obtained by applying lidar systems with multiple beams (see figure 4.2) or by a scanning configuration.

Lidar Peculiarities and Typical Faults

A relevant characteristic of the lidar sensors applied to this work consists of the form of its emission beam. Their beam geometry is used as a source for investigation for specific sensor faults (e.g. ground clutters) discussed in details in section 5.1.1. Figure 4.12 shows a snapshot performed with a infrared camera of the long range lidar sensor assembled in the vehicle prototype. It illustrates an actual lidar emission pulse with an aperture angle of 14.2 degrees and 16 channel beams. By means of this practical investigation the assumption of approximating a lidar emission pulse as elliptical cone could be confirmed. This is one of the prerequisite assumptions for build the ground clutter model discussed in detail in section 5.1.1.

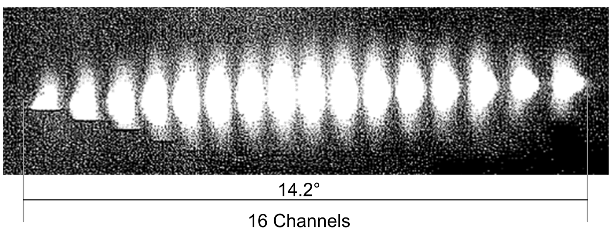


Figure 4.12: A snapshot performed with a infrared camera of the long range lidar sensor assembled in the vehicle prototype. It illustrates an actual lidar emission pulse with an aperture angle of 14.2 degrees and 16 channel beams.

Another matter of special interest is the unfavorable weather performance of lidar sensors. The applied sensors operate in the near infrared region with a wavelength of 900 nm, which is sensitive to atmospheric particles due to precipitations (e.g. rain and snow) and fog. This particles may attenuate light and can be interpreted by lidar sensor units as valid target objects. An extensive investigation of unfavorable weather influence on lidar sensor is described by Senega [2006]. He investigated the behavior of lidar sensor under several rain, snow, fog and dirt conditions in the real environment. Figure 4.13 illustrates a space distribution of irrelevant targets from ADAS point of view by strong snow precipitation.

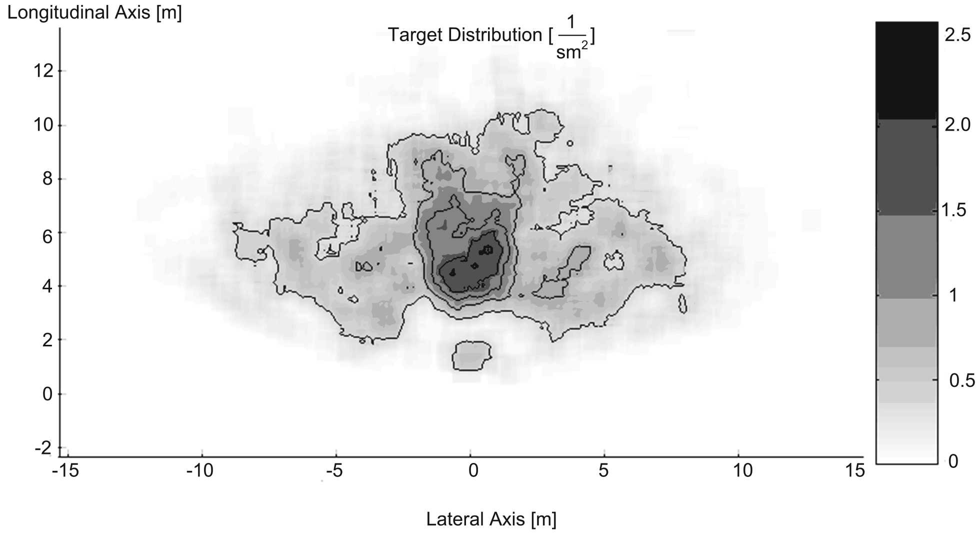


Figure 4.13: Space distribution of irrelevant targets from ADAS point of view by strong snow precipitation [Senega, 2006].

As expected figure 4.13 shows that the obtained measurements are highly concentrated in front of the sensor. Especially for this strong snow precipitation the main concentration of measurements is found around 5 meters in the longitudinal axis direct in front of the vehicle. Senega [2006] confirmed similar target distributions under strong rain and fog conditions. Thus these effects may be used as evidences for the evaluation of sensor measurements and tracks. A further matter of interest related to unfavorable weather conditions is the occurrence of a sort of spray cloud generated by preceding vehicles on wet road surfaces. A schematic illustration of this effect is illustrated in figure 4.14.

Analogue to radar faults the discussed information about the peculiarities and typical lidar sensor faults represent a source of information for validating sensor measurements and posterior tracks and sensor failures. Also the detection of lidar sensor fault play a more important role for most ADAS the their identification. That is why models covering plausibility criteria for detecting theses faults will be implemented as well (see section 5.1).

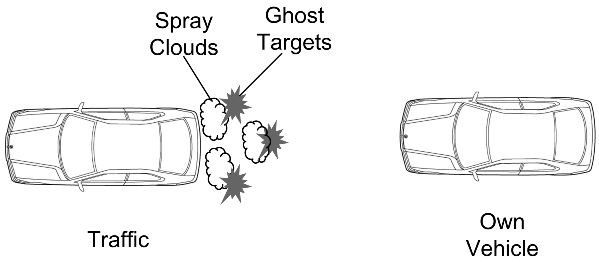


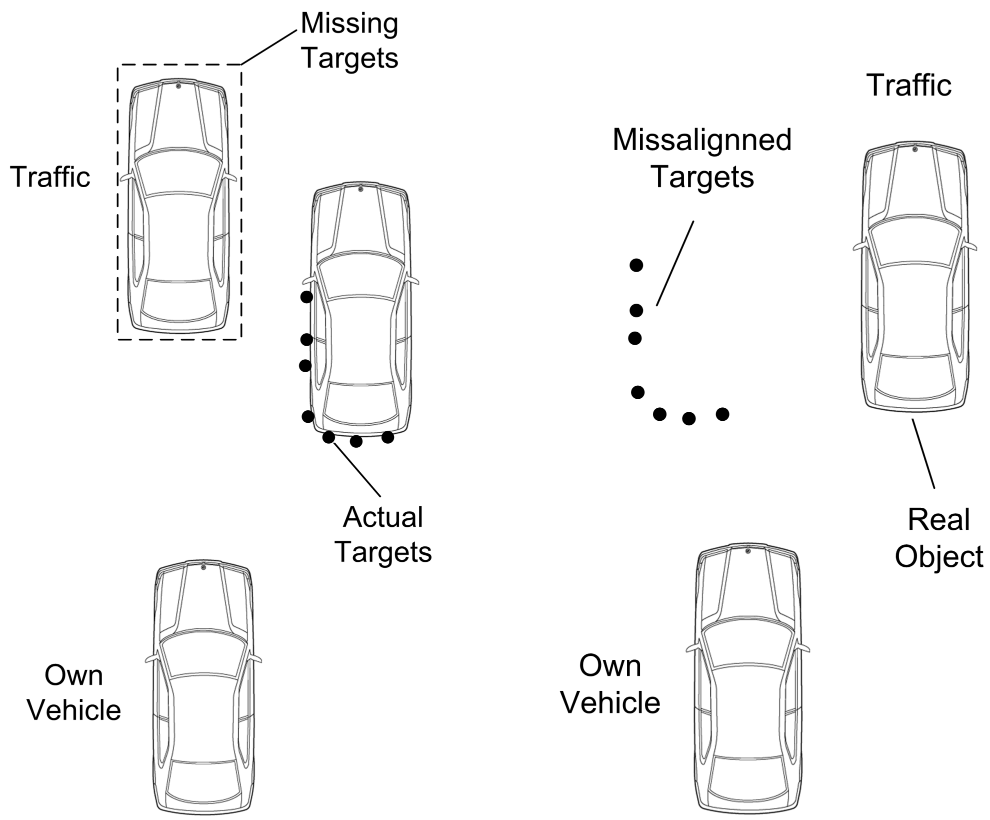
Figure 4.14: Scheme illustrating the generation of inexistent targets due to the occurrence of a spray cloud generated by preceding vehicles by wet driving surface.

4.3 Sensor Failures

This section covers possible specific failures related to the sensors applied for the validation of this work. As mentioned previously sensor failures correspond to a permanent interruption of a system's ability to perform a required function under specified operating conditions. Typical failures that may influence the performance of distance based sensor like radar and lidar are sensor blindness and misalignment.

Sensor complete or partial blindness is categorized by the absence of expected target object measurements in the driving environment. While complete blindness might be easily and already detected by the sensor unit, partial blindness may represent a critical failure state for ADAS. Partial sensor blindness is categorized by the absence of sensor measurements only in determined sections of the correspondent sensor coverage area. Figure 4.15(a) shows a scheme representing sensor partial blindness in the driving environment.

Sensor misalignment represents the situation generated by alteration in the assembly position of a sensor relative to its pitch and yaw angles. Due to pitch angle misalignment being more straightforward to be recognized, sensor yaw angle misalignment will be investigated in detail in order to validate the proposed concept. Figure 4.15(b) illustrates a schematic representation of a sensor yaw angle misalignment.



(a) Scheme illustrating partial blindness failure. It shows a typical traffic situation in which sensor measurement of a valid object are missing.

(b) Scheme illustrating sensor yaw misalignment. It shows a typical driving situation in which the vehicle's position is misinterpreted due to misadjustment of sensor yaw angle assembly position.

Figure 4.15: Typical radar failures for static units (no scanner properties).

As mentioned previously not only the detection, but also the identification of sensor failures are prerequisites for the implementation of safety relevant and to some extent autonomous ADAS. That is why explicit models describing these kind of failures are built in order to extract an evaluate symptoms that may lead back to these anomalies. The extraction of symptoms and the specific detection and identification of sensor misalignment and blindness will be discussed in detail in section 5.3.

5

Experiments and Results

The aim of this chapter is to give an overview on the experiments performed for the proof of concept explored in chapter 3. For this purpose a test vehicle equipped with several radar, lidar and video based sensor units were adopted. A brief survey of these sensor technologies along with their peculiarities were covered in chapter 4.

In the following sections some examples for exploring the potentialities of this proposed approach for evaluating driving environment information will be discussed. As the application field of this concept can be applied to several scenarios other than driver assistance systems and this is beyond the scope of this thesis, only some possible examples will be explored here. In doing so important aspects for evaluation the acquired information regarding quality and integrity as well as sensor hardware regarding their reliability will be discussed.

5.1 Sensor Fault Evaluation

Considering the definition assumed in section 2.2.1 that:

"*a fault is an impermissible deviation of at least one characteristic property (feature) of the system from the acceptable, usual standard condition*",

the premises for evaluating sensor faults can be determined. Sensor faults are anomalies direct related to sensor measurement principles and can be influenced by several constraints like weather or measuring constellations, which can not be considered as a permanent sensor error. Due to these kind of faults may occur promptly their detection have to be performed dynamically. They are associated to the dynamic detection and classification of faults in the measurement layer (section 3.2.1).

According to some types of typical radar and lidar faults covered in section 4.2 the first step for performing their evaluation is the analysis of cause-consequence effects. Essentially sensor faults are characterized by the generation of "ghost" measurements (targets). The occurrence of "ghost" targets can lead several ADAS to erroneous counteractions and thus downgrading drivers safety instead of enhancing it. For instance the detection of an inexistent object in front of the own vehicle could hypothetically activate the emergency brake function causing accident involving other vehicles. That is why the detection of sensor faults is, in most of the cases, rather important than the determination of their cause (fault identification). However the identification of specific faults is under some circumstances necessary due to several ADAS deal differently with it. Figure 5.1 shows a cause-consequence scheme for the evaluation of "ghost" targets.

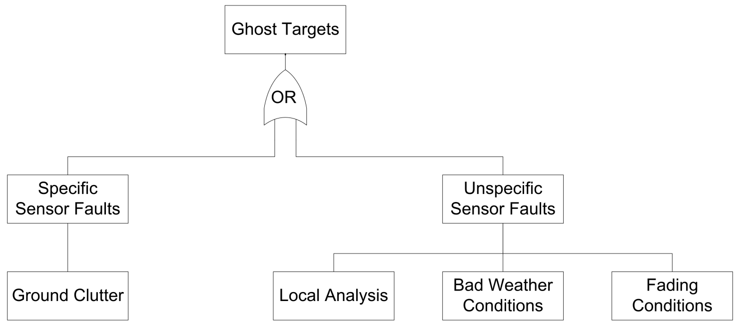


Figure 5.1: Scheme for the evaluation of "ghost" targets.

Depending on the availability of the extracted symptoms and on the relevancy of the evaluated faults the cause for "ghost" targets is shared in specific and unspecific sensor faults. Figure 5.1 illustrates from its bottom to top an example of causes (sensor faults) to the consequence ("ghost" target).

An example of specific sensor faults is characterized by ground clutters due to their relevancy for specific ADAS as well as the possibility to extract symptoms that supply evidences to explicitly classify it. Ground clutters are measurements generated essentially by the pitch and roll vehicle motion and will be explored in details in section 5.1.1. On the other hand unspecific sensor faults are the most probable cause of inexistent targets, but their explicit classification is either hard to be achieved due to the lack of symptoms or their relevancy are very low comparing with computational efforts. That is why a local analysis of acquired data is performed in order to determine the occurrence of unspecific sensor faults. Examples for typical sensor fault which are difficult to be determined by means of distance based sensors are due to bad weather conditions and multipath signal fading.

Once measurements could be classified as "ghost" targets their further classification can be performed. Measurements that are not associated to inexistent or irrelevant targets may be assigned to tracks. This is the complementary classification of "ghost" targets. As mentioned earlier a track is a state trajectory estimated from a set of measurements that have been associated with the same target and works as reference for decisions made by ADAS. Figure 5.2 shows the overall scheme for the evaluation of track hypotheses. Prerequisites for asserting measurement to a specific track is that they are not generated by sensor faults and fall within the membership area of this track. The membership area is built by track attributes similarities, which will be discussed in section 5.1.3.

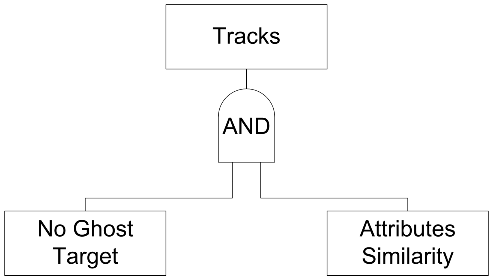


Figure 5.2: Scheme for the evaluation of tracks hypotheses.

Based on the exemplified evidences used to determine sensor fault and respectively the origin of a acquired measurement (“ghost” targets or tracks) the architecture for the probabilistic network is accomplished (figure 5.3).

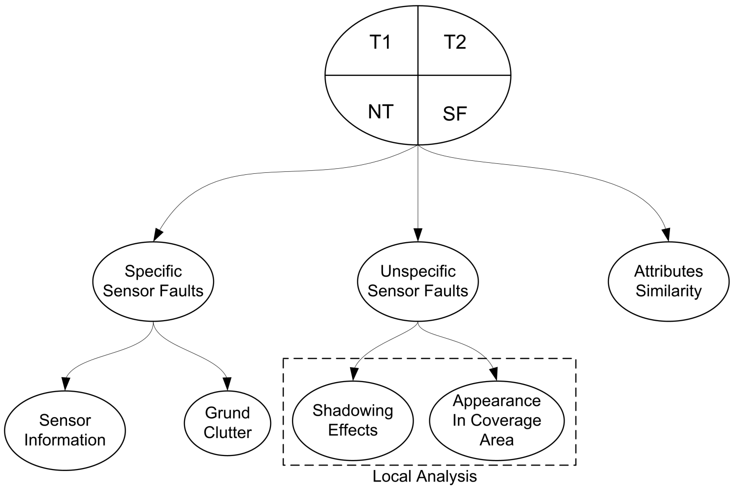


Figure 5.3: Probabilistic network architecture for evaluating sensor faults and the origin of acquired measurements.

Figure 5.3 describe the hierarchy for classifying sensor measurements according to their integrity and reliability. As described in section 3.3 every single measurement is tested against the hypotheses represented in figure 5.3 by nodes. At the top of this probabilistic tree the final hypotheses states, which a measurement may assume are described as follows:

- T1-T2: corresponds to the maximum number of existent track hypotheses, which one measurement can be asserted to. Depending on the system characteristics like sensor type and configuration the number of hypotheses may be increased. For the hardware configuration used in this work (chapter 4) however two hypotheses offered the best tradeoff between computational efforts and classification reliability. Actually the number of possible tracks associated to a measurement depends directly on how precise a track hypothesis corresponds to an object in the driving environment. Tracks consisting of several and meaningful attributes do not need to share measurements with the others. This desired track precision depends also on the type of acquired sensor information.

- NT: represents a new track hypothesis generated specifically for this measurement. It usually occurs when a new object in the driving environment emerges in the coverage area of one or more sensors. This means the point in time when objects are "seen" for the first time by the sensors.
- SF: describes the global sensor fault state without making a reference to either specific or unspecific faults. If a determined measurement assumes this hypothesis states it should not influence ADAS in its performance. This means that an application should not react to it.

All these states are mutual exclusive and assume probabilistic values. This implies that a measurement hypothesis must be shared among these states and the total states sum has to be 1.

This measurement classification is supported by evidences obtained from specific, unspecific faults and the attributes similarities. The specific fault node is described only by the assumption of availability or absence. Example of symptoms for the determination of specific faults are the occurrence of ground clutters and the information obtained directly from sensor hardware units. Typical sensor information are related to misalignment, total and partial blindness.

Unspecific sensor fault classification is mostly performed by a local analysis of measurements and is characterized by availability or absence assumptions as well. Example of applied symptoms are related to shadowing effects generated by reliable object hypotheses in the driving environment as well as by the appearance position of measurements within the sensor coverage area. The extraction of these symptoms along with the determination of their influence in the whole system will be explored in details in section 5.1.2.

Final aspects for classifying sensor measurements are represented by the attributes deviation node. It consists of the membership evaluation (section 5.1.3). In this node residuals between measurements and the mathematical modeling of object hypotheses are evaluated. Instead of only checking the availability or absence of residuals their magnitude is considered. According to the correspondent residuals intensity their contribution for asserting measurement hypotheses is appointed. Residual generation is supported here by a Kalman filter and are used for filtering tracks trajectory as well. Obviously filtering and generating residual may be also performed by means of several filter types like α-β and particle filters. However Kalman filters are adopted here due to their straightforward properties for fusion and filtering sensor data. A detailed evaluation among different filters is not within the focus of this work and therefore not performed.

Classifying measurements according to the probabilistic network architecture illustrated in figure 5.3 enables both the determination of sensor faults as well as the further evaluation of tracks. In the following sections examples for the assertion of specific and unspecific faults will be covered in details.

5.1.1 Specific Sensor Faults

As previously mentioned the detection of specific sensor faults is extremely necessary for determining the quality of data acquired in the driving environment. However their identification usually implies high computational efforts. On the other hand if this phenomenon can be modeled, the extractable symptoms are meaningful and their identification is relevant, its classification can be performed. A valuable example of a specific sensor fault to be detected and identified with aid of distance based sensor is ground clutters. In the following section an example of a ground clutter filter and how it can support the measurement classification in the context of this work will be covered. Main source for developing this filter is the approach covered in Issa [2007] within the scope of a diploma thesis.

Ground Clutters

Ground clutters consist in an ADAS context of reflexion points, which are generated by the pitch and roll motion of a vehicle due to both vehicle dynamics or by driving over a road with an abrupt inclination change. Figure illustrates 5.4 ground clutters acquired with lidar sensors due to an abrupt change in the road inclination.

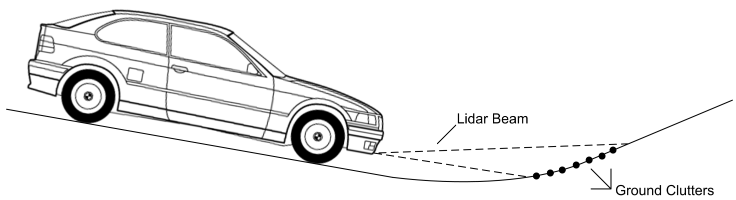


Figure 5.4: Scheme illustrating ground clutters acquired with lidar sensors due to an abrupt change in the road inclination.

According to the measurement principles of the applied lidar sensors discussed in section 4.2.2 evidences for the assertion of ground clutter hypotheses were investigated. By means of several simulations and real tests a mathematical model for describing ground clutter could be built. Main features for building this model are elliptical characteristics of the lidar beam as well as the assumption of roads being modeled as a plane geometric surface. Based on these evidences and on the premises of vehicle pitch and roll motion for forming ground clutters conic sections were investigated as basis for this model implementation.

Conic sections consist of curves obtained by the intersection of a plane with one or two napes of a cone. Considering a plane intersecting the axis of the cone perpendicularly, the resulting curve is a circle. Assuming no perpendicular intersection with a single nape, the resulting curve is either an ellipse or a parabola. While a non perpendicular intersection with two napes describes a hyperbola. Figure 5.5 illustrates the generated curves due to conic sections [Hilbert and Cohn-Vossen, 1999].

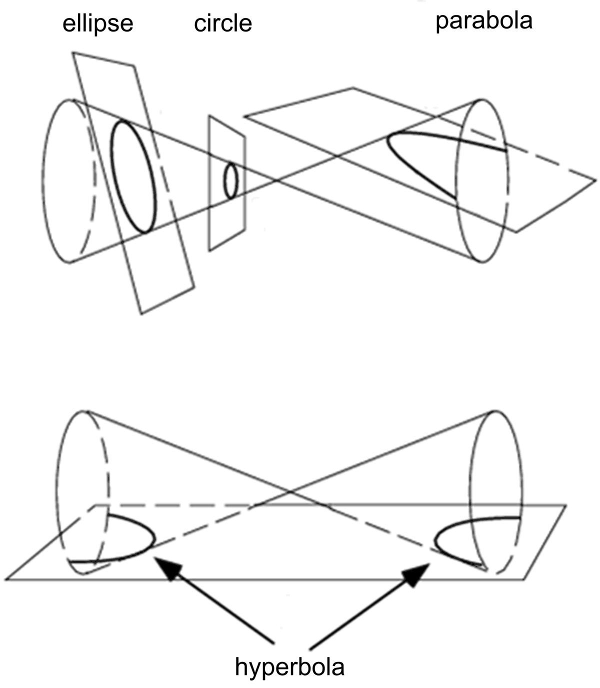


Figure 5.5: Scheme illustrating conic section for the basic analysis of ground clutters [Hilbert and Cohn-Vossen, 1999].

Due to lidar beams can be approximated to elliptical cones (see section 4.2.2) and according to the allowed degrees of freedom of a vehicle, the curve under consideration for representing ground clutter may be approximated to a parabola. This means that a parabola assumption is the most suitable one for describing a lidar beam intersecting the ground. The first step for proving this assumption as well as for extracting further evidences of it is a reliable simulation of this phenomenon. Therefore a mathematical model that describes a lidar beam according to the specification covered in section 4.2.2 and its intersection with the road surface were built and validated.

Figure 5.6 illustrates the simulated lidar beam with a lateral and vertical aperture angle $\alpha = 14^\circ$ and $\beta = 2^\circ$ while the longitudinal range amounts 150 meters. These aperture angles correspond thus a elliptical cone with a vertical and lateral axis $a \cong 3$ meters and $b \cong 18$ meters respectively by a maximum longitudinal range of 150 meters.

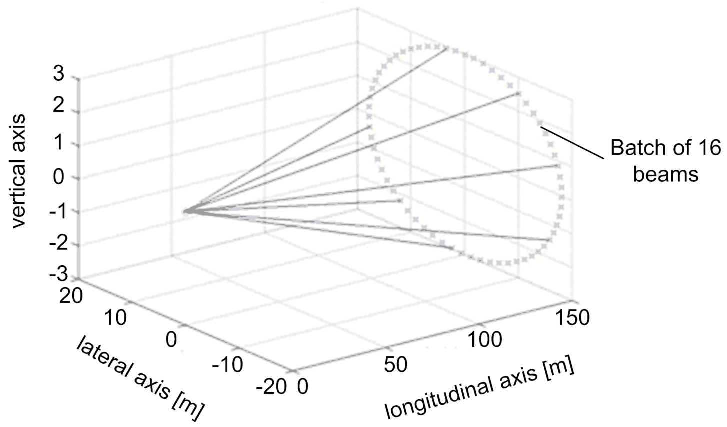


Figure 5.6: Simulated lidar beam with a lateral and vertical aperture angle $\alpha = 14^\circ$ and $\beta = 2^\circ$. Longitudinal range amounts 150 meters.

An example of this resulting intersection by a pitch angle of $\Delta\phi = 2^\circ$ and a roll angle of $\Delta\xi = 0^\circ$ is shown in figure 5.7.

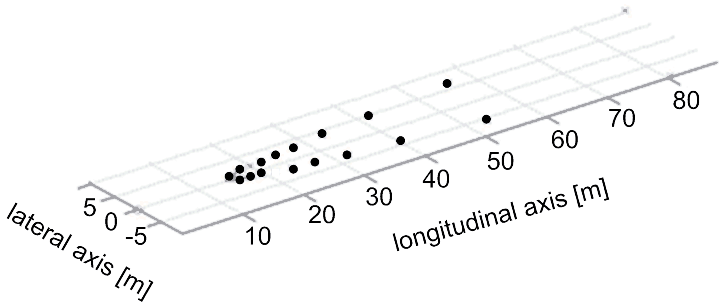


Figure 5.7: Resulting intersection between simulated lidar beam and simulated road surface.

By simulating a hypothetical ground clutter first knowledge about its properties like geometry and emerging position could be acquired. The assumption of a parabolic form could be confirmed as well as a emerging position range between 10 and 20 meters could be extracted. The main purpose of simulating a lidar beam and a road surface was not for obtaining all characteristics of ground clutters, but only for figuring out their primary properties.

Based on a experimental analysis executed in a real driving environment, geometry and emerging position of ground clutters could be confirmed. This analysis was performed using the vehicle prototype configuration described previously in section 4.1 considering exclusively the front far lidar sensor unit. As covered earlier this sensor unit consists of multiple lidar beams with a longitudinal coverage range of approximately 150 meters and lateral and vertical aperture angles of respectively 14.2 and 2 degrees. Figure 5.8 illustrates the 3-D and 2-D amplitude power diagram of a ground clutter generated by an intentionally abrupt break in a real driving environment without the presence of other objects like cars, trucks, motorcycles, etc. An approximation of a parabolic form can be identified by the amplitude power reaching values of approximately 300×10^{-12} Watt. Furthermore the emerging range of this ground clutter fall within the longitudinal range between 10 and 20 meters while the vertical range could be found around $\pm$ 3 meters.

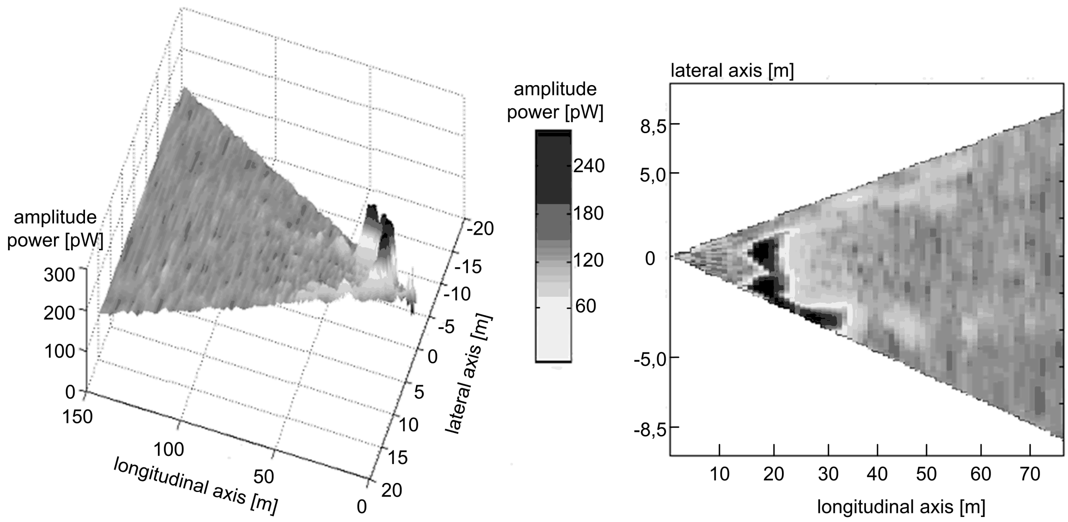


Figure 5.8: Amplitude power diagram of a ground clutter generated by an intentionally abrupt break in a real driving environment without the presence of other objects like cars, trucks, motorcycles, etc.

Based on the analysis of several experimental results the parabola dynamics dependence upon vehicle's pitch and roll motion could be confirmed as well. According to the changes of pitch and roll angles, parabola's position and aperture can be identified. For example a big pitch angle implies small aperture values and according to the conic sections it induces parabola's vertex to appear near to the vehicle.

Confirming the evidences that pitch and roll vehicle's motion influence ground clutter geometry gives basis knowledge for building a state space model for estimating the parabola's behavior. A valuable approach for this would be a ground clutter estimator based on Kalman filters. However the main goal of this specific fault detection is an approach based only on the information of the distance based sensors themselves. Additionally computational efforts caused by Kalman filters processing can not be justified by the frequency of occurrence and duration of ground clutters. According to the experimental results this specific fault lasts approximately 4 duty cycles of the used sensor unit. For these reasons an exclusive geometric fitting for the detection and identification of ground clutter could be confirmed as the best approach.

The algorithm for performing the detection of this specific faults consist thus of a parameter estimation task for nonlinear static processes. The process equation is then described by a parabola as follows:

$$x_p = Ay^2 + By + C + n \tag{5.1}$$

where the parameters A, B and C are determined with the aid of sensor measurements describing the position of the ground reflexions with the attributes x and y. The variable n describes the random noise involved in the process. The scheme illustrated in figure 5.9 shows a bird view perspective of a parabola representing a ground clutter in the driving environment along with the parameters to be estimated.

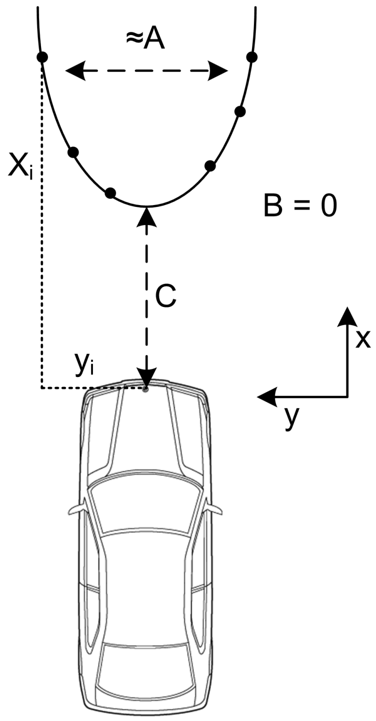


Figure 5.9: Scheme illustrating the bird view perspective of a parabola representing a ground clutter in the driving environment.

According to the experimental results the parameter B of the ground clutter parabola can be neglected. This means that the parabola geometry generated by ground clutters is essentially determined directly in front of the measuring sensor unit. Then the process equation for performing the estimation of these parable parameters considering the sensor measurements involved can be summarized as follows:

$$\underline{x}_p = \overline{U}\underline{p} + \underline{n} \tag{5.2}$$

where $\underline{x}_p$ represents the vector of longitudinal position measurements of ground clutters while matrix $\overline{U}$ describes the process model with the parabola equations and $\underline{p}$ consists of the parameter vector. The additional term $\underline{n}$ symbolizes the random noise vector associated with each parabola equation. Thus the vectorized terms of this process model assuming the absence of parameter B are formalized in equation 5.3.

$$\begin{bmatrix} x_p(0) \\ \vdots \\ x_p(i-1) \end{bmatrix} = \begin{bmatrix} 1 & y^2(0) \\ \vdots & \vdots \\ 1 & y^2(i-1) \end{bmatrix} \begin{bmatrix} C \\ A \end{bmatrix} + \begin{bmatrix} n(0) \\ \vdots \\ n(i-1) \end{bmatrix} \tag{5.3}$$

A generic scheme for parameter estimation applied for detecting ground clutters is depicted in figure 5.10. It illustrates the comparison of measured and calculated signals by building a residual vector e symbolizing the discrepancy between these two signals.

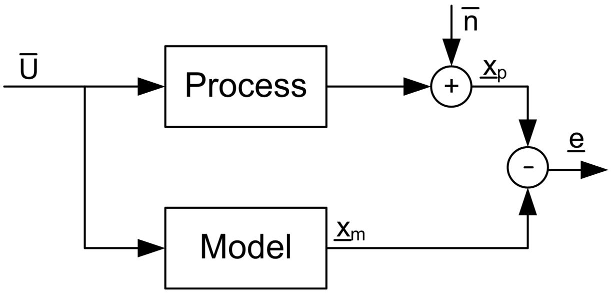


Figure 5.10: A generic scheme for parameter estimation applied for detecting ground clutters.

Once the process model is built the estimation of the parabola parameters can be achieved by minimizing the system quadratic loss function. This function is determined by rearranging, adapting and simplifying equation 5.2 as follows:

$$\begin{aligned} L &= \underline{e}^T\underline{e} = [\underline{x}_p^T - \overline{U}^T\underline{p}^T][\underline{x}_p - \overline{U}\underline{p}] \\ &= \underline{x}_p^T\underline{x}_p - \overline{U}^T\underline{p}^T\underline{x}_p - (\overline{U}^T\underline{x}_p)^T\underline{p} + \underline{p}^T\overline{U}^T\overline{U}\underline{p}. \end{aligned} \tag{5.4}$$

Thus minimizing loss function L results in:

$$\begin{aligned} \frac{dL}{dp} &= -2\overline{U}^T\underline{x}_p + 2\overline{U}^T\overline{U}\underline{p} = 0 \\ \frac{dL}{dp} &= -2\overline{U}^T[\underline{x}_p - \overline{U}\underline{p}] = 0 \end{aligned} \tag{5.5}$$

and the estimating equation is obtained from equation 5.5 as follows:

$$\underline{\widehat{p}} = [\overline{U}^T\overline{U}]^{-1}\overline{U}^T\underline{x}_p, \text{ for } \det[\overline{U}^T\overline{U}] \neq 0. \tag{5.6}$$

According to the estimating equation 5.6 sensor measurements membership to the ground clutter hypothesis can be checked. Decision criterion consists of comparing the longitudinal and lateral deviation between obtained measurements and the predicted parabola. A relevant aspect for this decision is the preselection of sensor measurements for assuring the right hypothesis assertion as well as saving computational efforts. The following measurement properties are examples of preselection criteria applied to ground clutters:

- Emerging Position: is related to the selection of measurements acquired in front of the own vehicle within a delimited field with x and y varying between 10 and 40 meters and ± 5 meters respectively.
- Measurements Segmentation: corresponds to the attempt of only selecting measurements related to ground clutters. Sometimes measurements describing relevant objects like vehicles may be associated erroneously to the searching parabola due to their parabolic shape. One method to minimize these wrong assertions is selecting measurements, which are within a fixed standard deviation of the parabola's vertex and can not be associated to valid objects.

In order to validate the algorithm for detecting and classifying ground clutters a statistical approach under several circumstances like weather conditions and traffic density were investigate. This validation was performed within the scope of a diploma thesis in Issa [2007] and the results are represented by means of the so called confusion matrix [Kohavi and Provost, 2004]. Confusion matrices provide an evaluation mechanism for classification systems by quantifying the relation between actual and predicted states. It describes numerically for a two-class classification example following terms:

- True Positive (tp): represents the number of positive classifications in case of the actual state being positive.
- False Negative (fn): represents the number of negative classifications in case of the actual state being positive.
- True Negative (tn): represents the number of negative classifications in case of the actual state being negative.
- False Positive (fp): represents the number of positive classifications in case of the actual state being negative.

Figure 5.11 shows the evaluation of the ground clutter filter according to the confusion matrix approaches with a distinct the number of object members. This validation was performed for three different categories: 6, 8 and 10 lidar measurements associated to the parabola respectively. The algorithm is executed in every duty cycle.

According to the diagram illustrated in figure 5.11 results are interpreted as follows:

- sensitivity: means the true positive rate and corresponds the percentage of right classifications of ground clutters. Sensitivity reached a maximum value of 71% of all analyzed situations. Furthermore the algorithm shown no big difference when varying the minimum

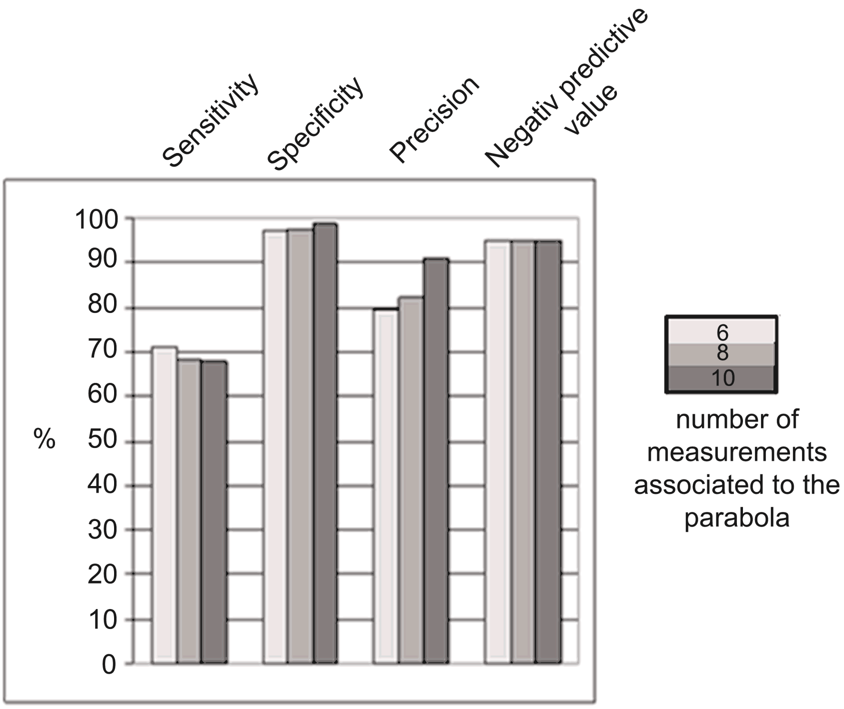


Figure 5.11: Evaluation of the ground clutter algorithms according to the confusion matrix approach.

amount of sensor measurement associated to the parabola. According to simulation and experimental tests an extension of this ground clutter model by considering pitch and roll angle measurements might increase the true positive rate.

- specificity: describes the true negative rate and means the percentage of classification where the absence of ground clutters is correct identified. In this case valid objects were considered, which should be theoretically classified as ground clutters due to their emerging position or parabolic form.

- precision: describes the probability that positive identified ground clutters is actually positive. It calculates the proportion of correct identified ground clutters against the total positive assumptions. Here a significant improvement by associating 10 sensor measurements results approximately 90%.

- negative predictive value: describes the probability that negative identified or non classified ground clutters is actually negative. It calculates the proportion of potential ground clutters (valid objects due to they emerging position or parabolic form) against the total negative assumptions.

5.1.2 Unspecific Sensor Faults

As mentioned previously unspecific sensor faults are related to sensor anomalies usually generated due to measuring principles. To these kind of faults all types of anomalies that can be detected but are not able to be identified will be associated. That is why a local analysis over the sensor measurements is performed. According to experimental tests, most of these faults can be associated to bad weather conditions, signal multipath fading or poor angle resolution (see section 4.2). In the following sections two examples of local analysis will be covered. They consist of the effects called shadowing area and measurement appearance in coverage area.

Shadowing Area

The term shadowing area corresponds to the effect generated when confirmed object hypotheses, with high believes, are determined in the driving environment. It consists of assuming a shadowing area behind and in front of a possible vehicle hypothesis. Once a sensor measurement is acquired within this virtual area, this measurement may assume three hypothesis states describing a new valid object, the object generating this shadowing area or a ghost target. Figure 5.12 represents schematically the shadowing effect.

Based on experimental tests the premise assumed consists of treating measurements appearing between the own vehicle and confirmed objects within this shadowing area as improbable ones (ghost targets). According to its appearance, in the middle or on the edge of this correspondent shadowing area, measurement confidence will decrease or increase respectively. This means if a sensor measurement is acquired direct in the middle of a shadowing area it will be considered by this analysis as a ghost target, otherwise its confidence will increase. In doing so cut-in situation can be considered. Delimiters for measurement appearance position are the α and β angles. These angles describe a sort of shadow intensity. This kind of position confidence assertion is not determined binary, but by means of a fuzzy approach characterized by membership functions (figure 3.8) and thus following the concept proposed in section 3.

On the other hand the premise assumed for treating measurements appearing in the shadowing area far away from the own vehicle depends on the type of sensor unit being analyzed. According to experimental tests radar sensors are able to acquire measurements of vehicles driving ahead in a sort of column traffic due to their reflexive properties (section 4.2.1). Therefore the shadowing effect has a stronger influence in validating measurements from lidar sensor units.

Beside the measurement position within the shadowing area regarding the angles α and β, the proximity relative to the confirmed object hypothesis will be taken into account. This also means that measurements appearing far away from the object responsible for this shadowing area will be asserted with a low confidence level. Again this is performed according to the concept proposed in section 3 referring to a soft decision process with membership functions.

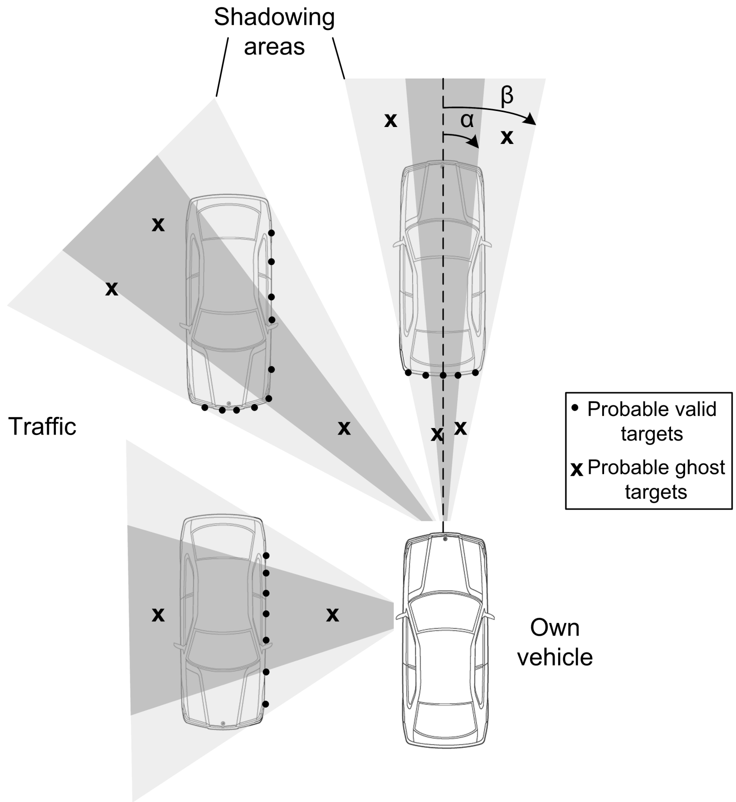


Figure 5.12: Schematically representation of the shadowing effect illustrating the correspondent shadowing areas along with probable valid and ghost targets.

Reference basis for this object hypothesis generating shadowing areas are past calculated tracks. According to the belief associated to these tracks in the past, the shadowing areas influence in the measurement validation will play a more or less important role. That is why this part of the track based feature extraction approach. This approach follows the theory proposed in chapter 3 and will be calculated according to the algorithms presented in the section measurement layer (3.2.1).

Appearance in Coverage Area

Analyzing the appearance position of measurements in their correspondent coverage area belongs to the category of sensor based feature extraction described earlier in section 3.2.1. By means of this approach the evidences for determining the confidence values of sensor measurements can be extracted as well.

The premises assumed are based on the sensor units functionality covered in section 4.2 as well as on experimental tests. For it the coverage area of the different sensor units are delimited and the appearance position of sensor measurement are evaluated according to sensor intern properties. Main aspects are related to the emission power of sensor units and plausibility criteria for the entrance of objects into sensor's coverage area. A knowing fact based on sensor specifications (section 4.2) is that emission power decays on the edge of sensors coverage area. This means, objects can be acquired with a higher level of confidence if they are measured in the center of the correspondent coverage area. With regard to the measurement position this assumption might be reverted. Isolated measurements acquired in the center of a sensor coverage area may assume a lower level of confidence than the one acquired on its edge. This can be interpreted by the fact that except for very seldom circumstances objects are not able to land abruptly in the middle of a traffic road. They should come from the edge of a sensor coverage area and then possibly move to the center.

Due to this two assumptions being contradictory a tradeoff between them has to be established. By means of experimental investigations following the categories and procedures could be confirmed:

- isolated appearance in the center of a coverage area: decreases measurements confidence due to the their probability of representing a ghost target by landing in the middle of the driving environment.
- joint appearance in the center of a coverage area: increases measurements confidence based on the assumption that the same object has been "seen" by other measurements. Additionally they are located on the most favorable position regarding sensors emission power characteristics.
- isolated appearance on the edge of a coverage area: does not influences measurement confidence due to the lack of certainty associated to this measurement. It can either describe an object entering in the sensor coverage area or a ghost target due to sensor power emission
- joint appearance on the edge of a coverage area: increases measurements confidence based on the assumption of a new object entering in the sensor coverage area and of also being "seen" by another measurements.

A scheme illustrating a virtual coverage area for evaluating sensor measurements is shown in figure 5.13.

Again the measurement position relative to the center or edge of a coverage area will be determined by means of "soft" decision principles based on membership functions according to the algorithms presented in the section measurement layer (3.2.1).

5.1.3 Attributes Similarity

Attributes similarity feature is related to the deviation between sensor measurements and modeled objects trajectory. The main aspect is to make use of filter structure and mathematical model already applied for target tracking in the driving environment.

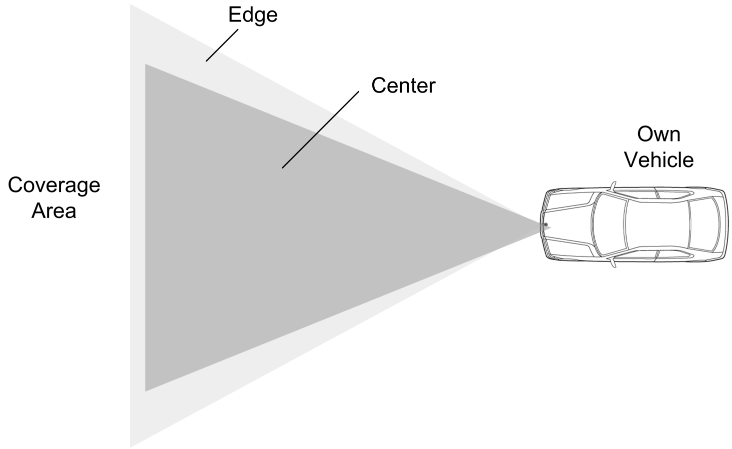


Figure 5.13: A scheme illustrating a virtual coverage area for evaluating sensor measurements.

The same residual for correcting the trajectory of object hypothesis is applied for extracting symptoms and thus allowing the evaluation of sensor measurements. The implemented model example consists of 6-D kinematic state model illustrated in figure 5.14.

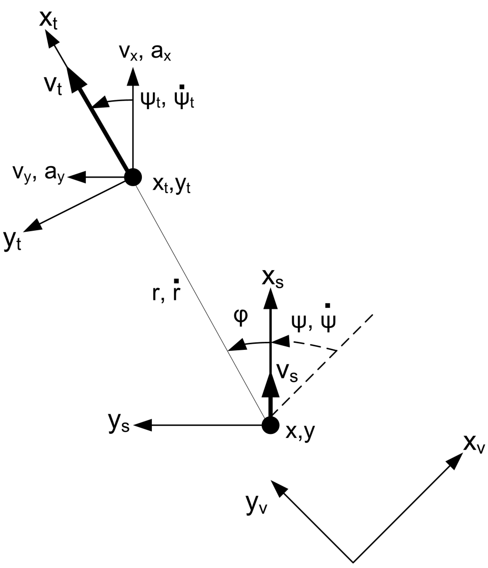


Figure 5.14: Object model depicting own vehicle, sensor, and target (object) coordinate systems.

In figure 5.14 the upmost coordinate system (x_t, y_t) represents the measured target, while the below ones depict the sensor (x_s, y_s) and vehicle (x_v, y_v) coordinate systems respectively. The aim is to obtain target attributes in their own coordinate system and represent them over the sensor and vehicle coordinate system respectively. The parameters illustrated in figure 5.14 are outlined in table 5.1. These attributes represent the 6-D state model for tracking multiple targets in the driving environment with the aid of distance based sensors like radar and lidar.

According to the sensor functionality described in section 4.2 the measurements for supporting the observability of this presented model are depicted in table 5.2.

x	target longitudinal position $[m]$
v_x	target longitudinal velocity $[m/s]$
a_x	target longitudinal acceleration $[m/s^2]$
y	target lateral position $[m]$
v_y	target lateral velocity $[m/s]$
a_y	target lateral acceleration $[m/s^2]$

Table 5.1: Parameters of 6-D kinematic state model used for tracking targets as well as extracting symptoms for the detection of sensor faults.

r	radial position $[m]$
ϕ	position angle $[rad]$
$\dot{r}$	relative radial velocity $[m/s]$

Table 5.2: Measurement obtained from distance based sensors in polar coordinates. Lidar sensor are just able to measure position attributes (e.g. r and ϕ) while radar units support additionally the measuring of the velocity component.

By means of this description of the environment the mathematical modeling can be accomplished by the linear differential equation outlined in this case in a continuous time representation:

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{v}_x \\ \dot{a}_x \\ \dot{y} \\ \dot{v}_y \\ \dot{a}_y \end{bmatrix}}_{\underline{\dot{x}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\overline{A}} \underbrace{\begin{bmatrix} x \\ v_x \\ a_x \\ y \\ v_y \\ a_y \end{bmatrix}}_{\underline{x}} - \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & r \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\overline{B}} \underbrace{\begin{bmatrix} v \\ \dot{\psi} \end{bmatrix}}_{\underline{u}} \tag{5.7}$$

and the nonlinear measurements are described by:

$$\underbrace{\begin{bmatrix} x \\ y \\ \dot{r} \end{bmatrix}}_{\underline{y}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \cos\phi & 0 & 0 & \sin\phi & 0 \end{bmatrix}}_{\overline{C}} \underbrace{\begin{bmatrix} x \\ v_x \\ a_x \\ y \\ v_y \\ a_y \end{bmatrix}}_{\underline{x}} \tag{5.8}$$

The vector $\underline{x}$ represents the state variables (table 5.1), which are, after discretized, updated in each operation cycle (10 - 50 Hz). The system matrix $\overline{A}$ represents the system by describing the relationship between the state variables and their variation in time in a continuous form. Via system matrix it is assumed that the jerk (i.e. the rate of change of the acceleration, j) of the analyzed targets is near to zero and therefore can be neglected. That is why these parameters are represented by $\dot{a}_x = 0$ and $\dot{a}_y = 0$.

The input matrix $\overline{B}$ acts as an interface between the state variables and the input parameters: the vehicle's own velocity v and yaw rate $\dot{\psi}$. By means of equation 5.8 the target measurements (table 5.2) are related with the state variables, where $\overline{C}$ represents the system measurement matrix.

Once the plant is build in a process model the residual between associated measurements and tracks determined by this model can be calculated. As mentioned before this residual is used for both filtering track trajectory and extracting symptoms for sensor faults. Due to measurements and model can be well represented by Gaussian properties the Kalman filter is applied for it. The correspondent association of measurement hypotheses and consequently filtering will be performed according to the approach proposed and discussed in section 3.3. Figure 5.15 shows a simplified scheme illustrating the extraction of fault symptoms by means of residual generation (i.e. attributes similarity).

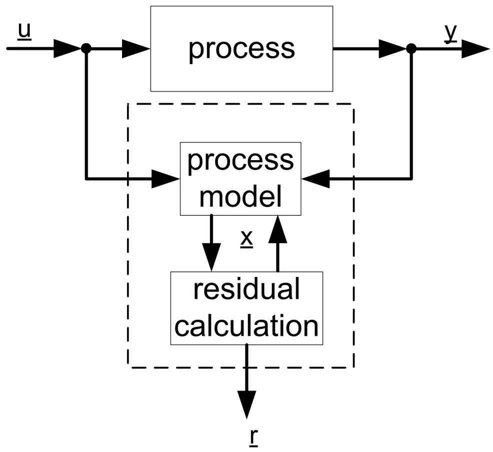


Figure 5.15: Simplified scheme illustrating the extraction of fault symptoms by means of residual generation (i.e. attributes similarity).

Thus the figurative representation of the attribute deviation feature can be performed by a hyperellipsoid, whose dimensions are determined by track attributes calculated with aid of the process model and the Kalman filter approach. These dimensions vary according to the number of attributes associated to a sensor measurement. In the case study covered here (section 4) radar and lidar sensor units supply two till three attributes illustrated in table 5.2. Then the evaluation is performed by sharing this ellipse or ellipsoid in membership areas that vary from the center to their edge representing respectively maximum to minimum confidence according to this feature. Again the measurement position relative to the center or edge of a coverage area will be determined by means of "soft" decision principles based on membership functions according to the algorithms presented in the section measurement layer (3.2.1). Figure 5.16 shows a exemplified scheme with two attributes for determining the attribute deviation feature. According to the deviation intensity between measurements and tracks attributes the complete level of confidence according to the this feature can be determined. Basis for this evaluation is the reliability of the correspondent track validated in the past cycles.

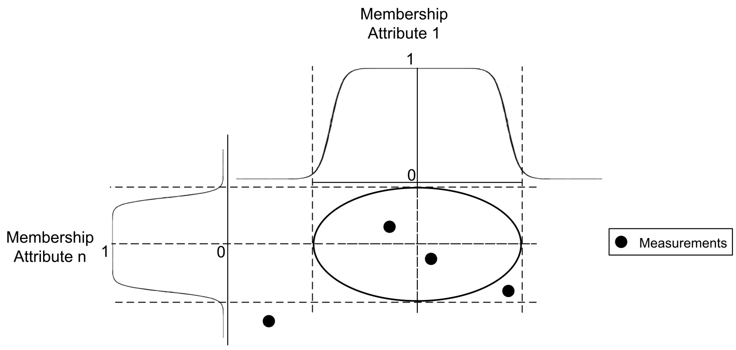


Figure 5.16: A exemplified scheme with two attributes for determining the attribute similarity feature

5.1.4 Measurement Evaluation

The final evaluation of measurements consists of determining the influences of the extracted features covered before like ground clutters, local analysis and the attributes deviation feature. This final evaluation is performed for every single measurement. Then it will be categorized as one of mutual hypotheses states described previously and illustrated in figure 5.3, namely existent track, a new track or a sensor fault. Main premise for evaluating sensor measurements is shared in two assumptions. First they cannot be described by features associated to sensor faults and second they are associated to valid objects in form of existent tracks with high degree of belief or plausible new tracks.

Thus the main challenge after extracting measurement features is to determine the amount of influence that this correspondent feature have in the assertion of the four hypothesis states. As described in section 3 the proposed approach foresees the usage of prior knowledge or learning strategies. If prior knowledge is available it is the best choice. Learning or training strategies are usually involved with high time efforts. Furthermore the chosen training field may be incomplete and thus inducing to partial correct or even wrong results. For the sake of complexity and considering the difficulty of reaching the complete training field for learning features relevancy, the influence for the described features is exclusively based on prior knowledge.

In order to facilitate measurement hypotheses assertion the influence of these different features are built and represented in a dependence model matrix discussed already in the chapter 3 where the proposed approach was introduced. An example of a dependence model for the feature shadowing area is shown in table 5.3.

hypotheses ↘ evidences	INNEAR	INFAR	OUT
$Existent\ Track_1$	0.8	0.2	0.0
$Existent\ Track_2$	0.1	0.3	0.6
$New\ Track$	0.1	0.3	0.6
$Sensor\ Fault$	0.1	0.7	0.2

Table 5.3: An example of a dependence model for shadowing area.

Table 5.3 reflects the influence of the evidences states of the feature shadowing area in the measurements hypothesis states assertion. Then the evidences states are interpreted as follows:

- INNEAR: represents the state of a measurement if it is positioned inside the shadowing area of the correspondent track 1 and also in the immediately vicinity of this track.
- INFAR: represents the state of a measurement if it is positioned inside the shadowing area of the correspondent track 1 but far away from it.
- OUT: represents the state of a measurement if it is positioned completely outside of the shadowing area of the correspondent track 1.

This means that if a measurement position relative to a certain track is determined in the INNEAR field this measurement will related to this track by an amount of 80% and 10% to the other hypothesis states. By evaluating the shadowing area of track 2 the rows in the dependence model will be alternated. The row of existent track 1 will be described by existent track 2 and vice versa. The preselection of tracks 1 and 2 is performed according to the two smallest aperture angles relative to the investigated measurement.

The influence calculation will be performed for every extracted feature for the assertion of these measurement hypotheses according to the respective evidence states. Additionally extra features can be easily and independently included in the whole system by determining their influence on the measurements assumptions. Once the whole influence is computed the correspondent measurement is evaluated according to the Bayesian approach covered in section 3.

For the sake of complexity a systematic validation by means of confusion matrix or Receiver Operating Characteristic curve (ROC) was not performed. This validation will be performed along with the track evaluation methods in the next section.

5.2 Track Evaluation

Track evaluation consists of a probabilistic network which comprises dynamic and static nodes. Dynamic nodes are represented by measurements that can be associated to this track according to the previous measurement evaluation. The static part are own track features which does not change their connection with the correspondent tracks. Referring the static part the node connections described by the measurements associated to tracks according the hypotheses asserted in the measurement layer will vary from one duty cycle to another. Figure 5.17 illustrate the probabilistic network for evaluating tracks.

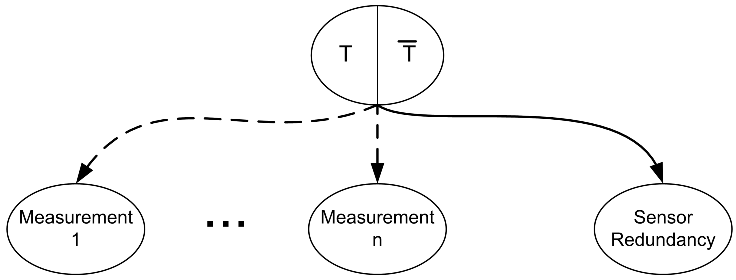


Figure 5.17: An example of a probabilistic network for determining the track evaluation

In figure 5.17 the dashed lines illustrate the dynamic connections between measurement and tracks symbolizing that this measurements just contribute for the evaluation in the current duty cycle. The connection with a solid line describes an example of a static property that will be taken into account during the whole period in which the related track exists. Measurements are associated to tracks according to their probability to belong to the track calculated on the measurement layer and discussed in the previous section.

In order to determine the amount of influence of a specific measurement in a determined track a influence model between them is built as well. Again the use of prior knowledge was privileged. Table 5.4 shows an example of a dependence model between measurements and a track.

hypotheses ↘ evidences	Track 1	Track 2	New Track	Sensor Fault
Track	0.8	0.067	0.067	0.067
No Track	0.067	0.067	0.067	0.8

Table 5.4: An example of a dependence model between measurement and track.

This table means if a measurement was previously classified as track 1 it will contribute with 80% for the hypothesis state track and with 2.5% for the remaining state no track. As illustrated in table 5.4 this measurement could also be partially associated to a neighbour, or to a complete new track of even to a sensor fault, what will reduce the contribution of this specific measurement for the current track confidence or belief. For every single measurement associated to a specific track in the sensor measurement evaluation its contribution will be accomplished.

Another important fact contributing for the determination of track's belief are the own track features that can be extracted from its own characteristics. One valuable example is the sensor redundancy feature. It consists of analyzing sensor overlapping areas and validating these regions. Figure 5.18 depicts an example of this overlapping field by means of two sensor coverage areas.

Chapter 5

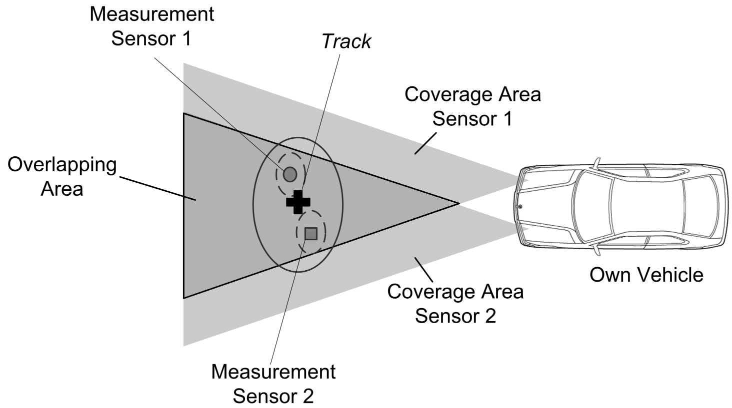


Figure 5.18: Scheme illustrating the overlapping area of two sensor's coverage areas

The premise assumed consists of investigating the origin of the measurements associated to tracks within different coverage areas. In the example illustrated in figure 5.18 tracks generated inside the delimited area are expected to be updated by both of the two sensors. Once only one sensor can confirm this evaluated track the cause for it has to be investigated in more details and can be shared as follows:

- open view: represents the situation in which there is no obstacle between sensor and the evaluated track. The effect of a track in the overlapping area not receiving an update of the correspondent sensor in a open view situation can lead back to a sensor failure like blindness, misalignment or to a wrong sensor update time.
- blocked view: represents the situation in which there is a obstacle between sensor and the evaluated track and therefore this missing measurement in the common overlapping area is justified.
- sensor time update: represents the situation in which a sensor can not deliver an update of the object in the common overlapping area due to a drift in a sensor duty cycle. The system is expecting updates for this sensor under a determined rate, but because of sensor unit intern exceptions it could not update the analyzed track.

For track evaluation just the absence of sensor redundancy under open view circumstances will influence its confidence or belief. This feature will be shared in three main evidence states: high, low and no redundancy. The evidence states high and low are direct related to the number of measurements from the expected sensors associated to the evaluated track and are determined based on fuzzy membership functions following the proposed approach (see section 3). Another important consideration for this open view situation consists of its relevancy for the identification of sensor failures like blindness or misalignment explored in the following sections.

Once the occurrence of this three redundancy states can be determined, the influence of this feature can be taken into account for evaluating the correspondent track. Again this influence model is accomplished by means of prior knowledge through observations of real driving situations. Table 5.5 gives an example of the dependence model between track's confidence or belief and the sensor redundancy feature. The terms high, low and no redundancy are related to the number of associated measurements acquired by the expected sensors within an overlapping area.

hypotheses ↘ evidences	High Redundancy	Low Redundancy	No Redundancy
Track	0.7	0.3	0.0
No Track	0.0	0.3	0.7

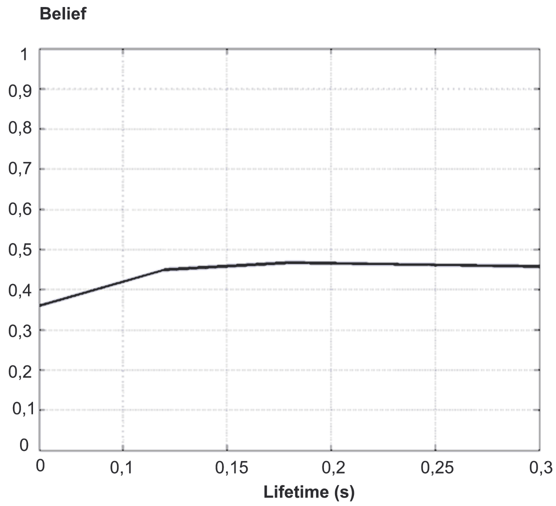
Table 5.5: An example of a dependence model between sensor redundancy feature and track.

Thus calculating the influence of all available features in the evaluated tracks enables the determination of its confidence or belief and so supplying several ADAS with the relevancy of the identified objects in the driving environment. The belief of track will be performed with aid of the probabilistic network approach discussed in section 3.2.1. In order to finalize the track belief calculations the track history information is taken into account as well. This will be performed according to the approach proposed in section 3.2.1 by means of a filter strategy. The usage of a low-pass filter enables a fair track evaluation by considering the influence of the extracted features more strongly if this track is very "young". As soon as this determined track reaches its maturity, its history will play a more strong role than the extracted features themselves.

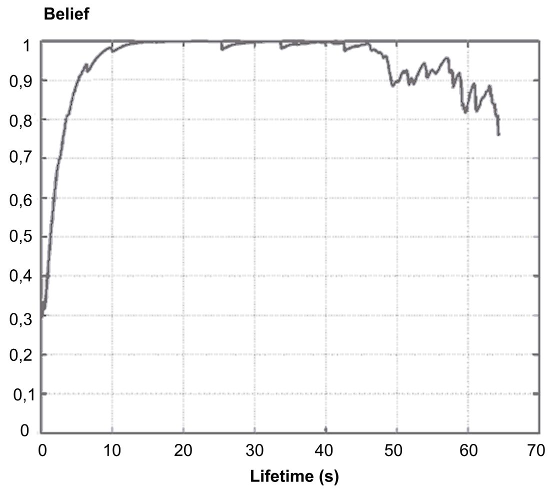
In order to validate track evaluation several experimental tests were performed in the real driving environment under different traffic conditions. Figure 5.19 illustrates examples of typical scenes in the driving environment with different traffic densities. It shows different objects which are acquired by means of reflexions points obtained from sensor measurement units. These measurements will generate new or be associated to existent object hypotheses in form of tracks. To each of these tracks a belief value will be associated according to process described previously.

Figure 5.19: Examples of typical scenes in the driving environment with different traffic densities where object will be measured by means of different sensor units, tracked and validated by the MFTT approach.

An illustrative example of the belief distribution of two tracks is shown in figure 5.20.



(a) Belief distribution of a track representing a ground clutter in the driving environment.



(b) Belief distribution of a track representing a vehicle in the driving environment.

Figure 5.20: An illustrative example of the belief distribution of two tracks in the driving environment represent for tracks describing a ground clutter and a vehicle.

The intention of these two examples of track belief distribution is not to substitute the complete concept validation, but indeed to give an idea on how belief distributions for valid and irrelevant tracks elapse. Figure 5.20(a) represents the belief distribution of a track originated by a ground clutter. During the whole track lifetime the belief distribution do not cross the 50% barrier for this irrelevant object. This can deal as reference for different ADAS to not react to this kind of false alarms. Although a 50% certainty level can be considered by most applications as insufficient to be taken into account, this decision may vary according to the requirements applied to each specific ADAS.

Figure 5.20(b) depicts an example of track belief distribution originated by a vehicle in the driving environment. It shows the initial belief of this vehicle track starting from 30% and reaching rapidly the mark of 90% until almost 100% due to the positive features associated to it. The oscillations on the distribution curve are related to measurement misses because of obstruction of the sensor field of view by valid objects.

In order to perform the concept validation as a whole 20 test runs with 2 minutes under different conditions where the most relevant valid and irrelevant objects were labelled and compared with the results obtained with the proposed approach. The results are represented by means of a Receiver Operating Characteristic (ROC) curve by a track life time of 10 seconds in figure 5.21. The ROC curve consists of a two dimensional measure for the performance of classification algorithms. It depicts the relation between benefits (true positives) and costs (false positives). Essentially a ROC curve is built by calculating several confusion matrices for distinguished threshold levels. A detailed overview on ROC curves is given in Fawcett [2005].

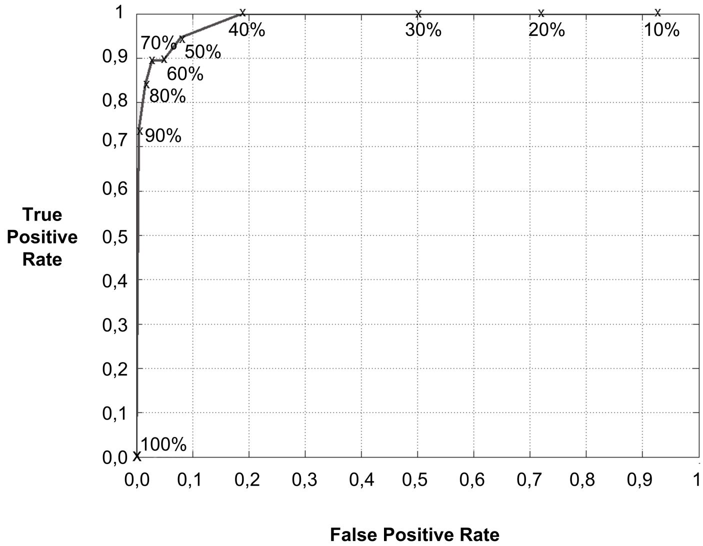


Figure 5.21: ROC curve for track evaluation by a lifetime of 10 seconds.

Due to tracks in the driving environment have different lifetimes it is difficult to achieve a fair evaluation method to verify the results of the proposed approach. In order to give an idea of how effective this approach works a ROC curve for a track life time with ten seconds was depicted in figure 5.21. It represents the performance of good classification tool by comparing the obtained results with the labelled ones and calculating true and false positive rates.

A more strength method for verifying the results of this approach is proposed by performing a multidimensional ROC analysis. It consists of building multiple ROC curves for different track lifetime and so achieving a fairly evaluation mechanism for this approach. For it figure 5.22 illustrates a multidimensional ROC curve for track evaluation varying 0 till 100 seconds lifetime. As expected it show that time play a very important role in this classification task. While ground clutter and other ghost targets can be precisely classified in a short period of time most of valid object like car, trucks and cyclist require more time for a precise identification. The time for a stable classification depends of course on the features available and may vary strongly from one object to another. Furthermore the good classification performance could be confirmed by comparing the obtained results with the labelled ones and calculating true and false positive rates as well.

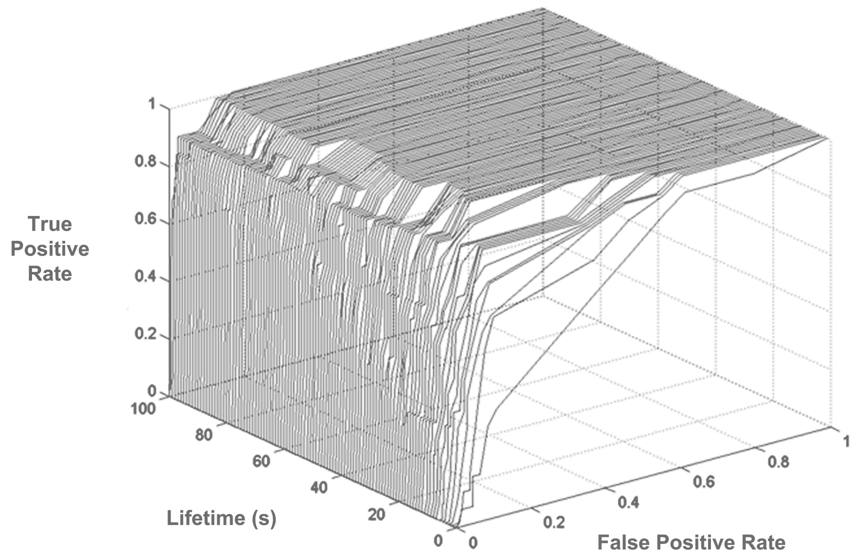


Figure 5.22: Multidimensional ROC curve for track evaluation by a varying lifetime between 0 and 100 seconds.

5.3 Sensor Failure Evaluation

Considering the definition assumed in section 2.2.1 that:

"a failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions",

the premises for evaluating sensor failures can be achieved. Sensor failures consist of anomalies that usually require the evaluation of the driving environment as a whole and depends strongly on a constant observation of the extracted features. Due to this kind of failures assume such a quasi-static behavior the cyclic results obtained from the evaluation of measurements and tracks are used here as inputs.

Similar to the detection of sensor faults the first step for performing the detection and identification of typical radar and lidar failures covered in section 4.2 is the analysis of cause-consequence effects. The essence of sensor failures are led back to anomalies that can be also noticed in the driving environment in form of ghost targets (artefacts). However artefacts generated by sensor failures are usually permanent and therefore a more deeply data analysis is required and also possible than by sensor faults. Besides the appearance of ghost targets the absence of expected measurements is an essential evidence characterizing sensor failures. Figure 5.23 shows a scheme for evaluating sensor failures in form of a case consequence analysis.

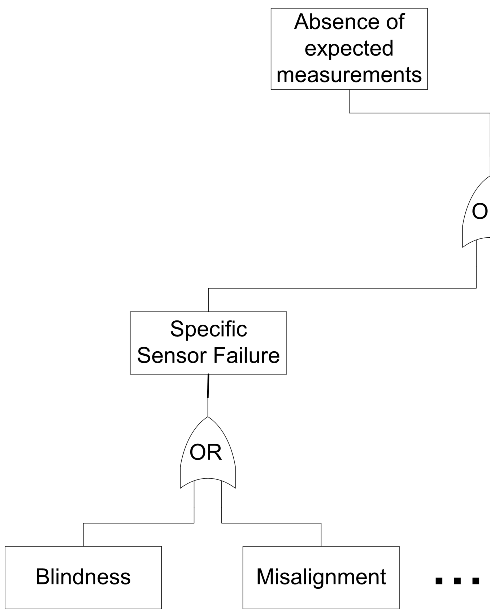


Figure 5.23: Scheme for evaluating sensor failures in form of a case consequence analysis.

Another relevant aspect is that both detection and identification of these anomalies are necessary. Detecting sensor failures may avoid ADAS to perform erroneous decisions in the driving environment by not reacting to these artefacts. While identifying these anomalies may deliver means for system counteractions. For instance the detection and identification of a misalignment failures may be used for correcting measurements position. However the unambiguous identification of failures depends directly on the availability and significance of the extracted symptoms. Then failures that can be detected and identified will be characterized as specific ones while unknown anomalies belong to the category of unspecific sensor failures. This basic type of failures is the cause for a permanent existence of ghost targets and absence of expected measurements in the driving environment (figure 5.23). Because of sensor failures are not mutually exclusive, multiple failures can take place, specific probabilistic networks are designed for each failure (section 3.2.2).

Due to the relevancy of detecting and identifying sensor failures only specific ones will be discussed in details in this work. Examples of typical specific radar and lidar failures are blindness and sensor misalignment. A detailed overview about these two specific anomalies will be given in the remain of this chapter.

5.3.1 Blindness Failure

As previously discussed in section 4.2 complete or partial sensor blindness consists of the absence of expected measurements and consequently object hypotheses in form of tracks in the driving environment. By evaluating the causes for this effect some relevant evidences can be extracted and thereby a probabilistic model for performing this failure check can be built. The main approach consists of investigating tracks density distribution over specific sensors coverage area in the driving environment. For it the coverage area of the applied sensors will be shared in different blindness sectors. In doing so not only a complete, but also a partial blindness identification is possible. Figure 5.24 gives an overview about the scheme illustrating this blindness sectors feature.

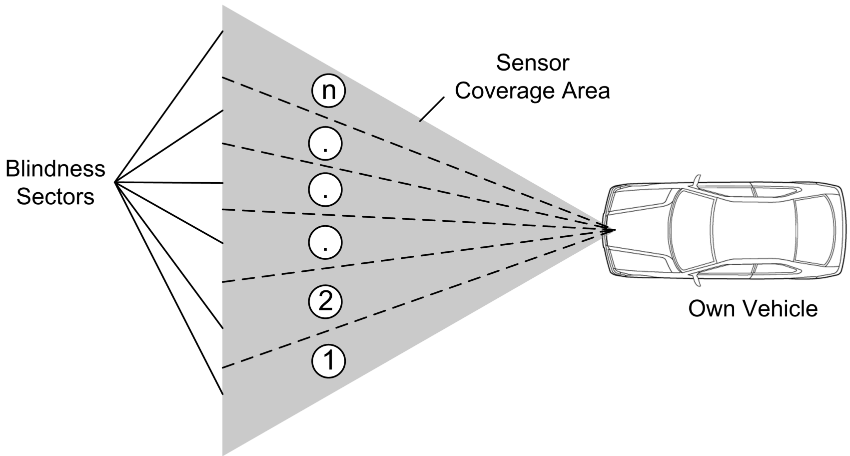


Figure 5.24: Scheme illustrating the blindness sector feature. The coverage area from the applied sensors is shared in multiple sectors for investigating tracks density.

Tracks density in the specific coverage area sectors consists of analyzing the overall number of tracks updated or eliminated within this hypothetical sectors. The assumptions for evaluating these tracks are shared as follows:

- track update: consists of investigating the generation and update of tracks with sensor measurements from specific blindness sectors. The miss rate per sector is calculated over a determined period of time and tested against a threshold value. In case of threshold transgression this evidence will be taken into account according to proposed approach in section 3.2.2.
- track elimination: consists of investigating the elimination of tracks within the specified blindness sectors. Only reliable tracks with high belief values are taken into account. The miss rate per sector is also calculated over a determined period of time and tested against a correspondent threshold value. In case of threshold transgression this evidence will be also treated according to proposed approach in section 3.2.2.

An important aspect to be considered is that according to specific sensor properties the generation, update and elimination of tracks can strongly vary among the blindness sectors. That is why correspondent cost or penalty factors for the different sector are applied depending on sensor characteristics. For instance sectors describing the inner part of coverage areas will be subject to stronger penalties for miss rates than the ones in the edge. This is because of decaying power emissions in the border of sensor coverage areas as well as due to valid objects being almost outside of the field of view from the correspondent sensor.

A further possible evidence contributing for the detection and identification of sensor partial and total blindness is recurring to the sensor redundancy effect discussed by the topic track evaluation (section 5.2). For it the overlapping field between multiple sensor coverage areas will be investigated. The assumption mentioned earlier is based on the fact that a track within this determined overlapping field should be updated by the correspondent sensor measurements. In case of missing measurements an extra verification is performed to determine the cause of it. If the described open view situation (see section 5.2) occurs and the measurement are still not available a measurement miss is assigned to the correspondent blindness sectors. Again the miss rate per sector by tracks with high believes is calculated over a determined period of time and tested against a correspondent threshold. If threshold values are exceeded the influence of this feature for blindness identification is taken into account. The amount of influence is first obtained by fuzzy membership functions (figure 3.16) and then assembled in a dependence model. An example of a dependence model for the sensor redundancy feature is illustrated as follows:

hypotheses $\searrow$ evidences	Redundancy	No Redundancy
$\mathit{Sensor\ Blindness}$	0.4	0.6
$\mathit{No\ Sensor\ Blindness}$	0.6	0.4

Table 5.6: An example of a dependence model between sensor redundancy feature and sensor blindness hypothesis states.

Similar to dependence models covered previously it will be also determined by prior knowledge methods. The specific blindness from each sector will not be distinguished for the sensor failure determination. However this information is available for detailed diagnosis procedures. Furthermore the uncertainty about this feature is expressed by the influence values in table 5.6. They amount 60% and 40% due to the uncertainty of this missing redundacy being originated by a possible sensor misaglignment failure.

A valuable feature to be considered is the sensor status information itself. In most of the current sensor units intern fault detection and diagnosis mechanisms are still available. Instead of investigating features occurrence over a period of time, sensor status information will be automatically taken into account as soon as available. For this feature an example of dependence model may be illustrated as in table 5.7.

The dependence values in table 5.7 express the relevancy of this feature for the identification of sensor blindness. The prompt indication of failure by the sensor status information will influence the failure hypothesis state to 100%.

hypotheses ↘ evidences	Blindness Status	No Blindness
$Sensor\ Blindness$	0.9	0.1
$No\ Sensor\ Blindness$	0.1	0.9

Table 5.7: An example of a dependence model between sensor status information and sensor blindness hypothesis states.

Once the dependence of all available feature is calculated, the final belief for this failure hypothesis states can be accomplished. This is performed according to the Bayesian principles discussed along with the proposed approach section in section 3.2.2. Figure 5.25 shows the proposed probabilistic network for classifying sensor blindness.

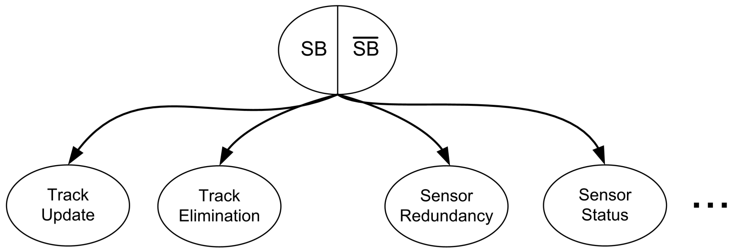


Figure 5.25: Scheme illustrating the proposed probability network for classifying sensor blindness from one specific sensor. The abbreviations SB and $\overline{SB}$ represent the states sensor blindness and no sensor blindness respectively.

By means of experimental tests the proposed concept along with the extracted features could be verified. For it several test runs were performed in the driving environment and sensor blindness was simulated by masking incoming sensor measurements from different blindness sectors in the sensor coverage area. Figure 5.26 shows a diagram illustrating the belief distribution from a example of blindness failure for a test run.

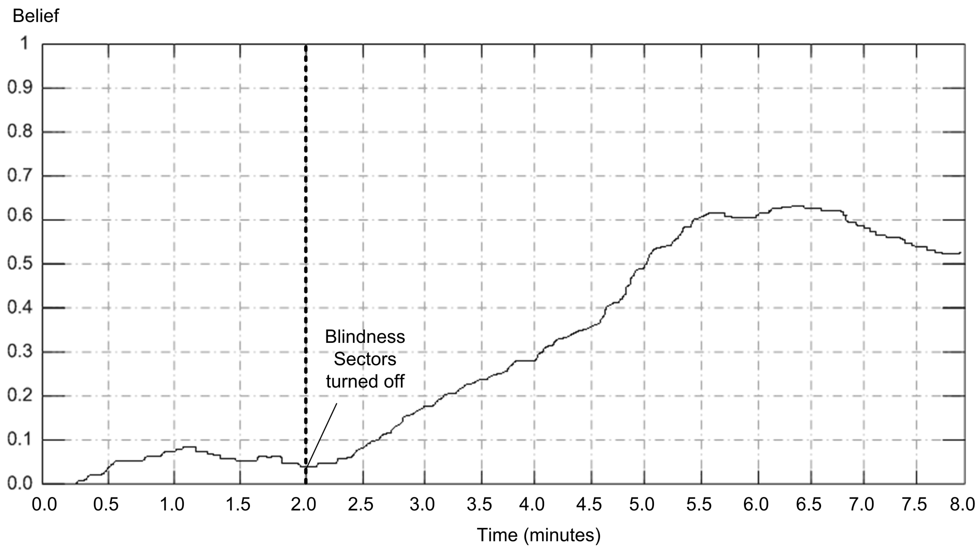


Figure 5.26: Diagram illustrating the belief distribution from a specific sensor failure, blindness, for a test run.

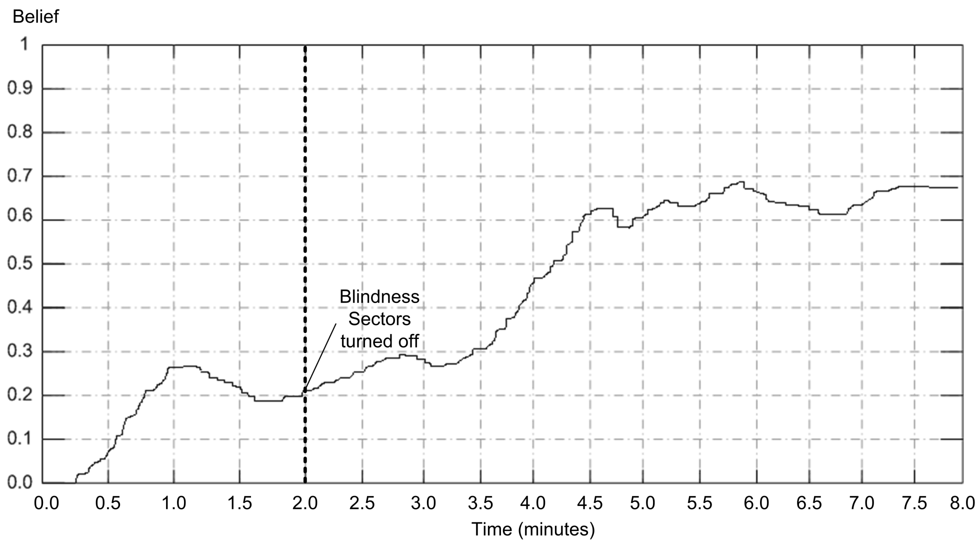


Figure 5.27: Diagram illustrating the belief distribution from a specific sensor failure, blindness, for a test run.

According to figure 5.26 an ascent of the belief value could be confirmed by masking incoming sensor measurements from three distinguished blindness sectors. The presented snapshot from a time slice of this test run shows a maximum belief value between 60% and 70%. So an explicit identification of blindness could be achieved. In order to improve the certainty of this classification a tradeoff has to be determined between the completeness and complexity of dependence models and probabilistic network structure.

A further belief distribution of another test run is depicted in figure 5.27. Here an explicit identification of sensor blindness could be achieved. A belief value between 60% and 70% could be confirmed as well. The main challenge by classifying sensor blindness as well as other failures consists on reducing the influence of false alarms that corresponds to blindness symptoms even in the absence of failures.

5.3.2 Misalignment Failure

Sensor misalignment consists of detuning effects in the alignment position of a sensor relative to its assembly pitch and yaw angles. Although the proposed concept can be applied to detect and identify misalignments in both degrees of freedom, this investigation focuses only on disturbances caused by a lateral (sensor azimuth angle) misadjustment. Essentially the method for extracting features or symptoms in order to detect and identify sensor misalignments is shared in two approaches. The first one consists of extracting features of the analyzed sensor independently from others while the second one is based on the information obtained from partially redundant and sometimes dissimilar sensor units.

A possible effect based on the analysis of at least one sensor is based on the geometric analysis of object hypothesis in form of tracks. This assumption consists of investigating the behavior of standing tracks by the straight approximation of the own vehicle respectively the analyzed sensor (figure 5.28).

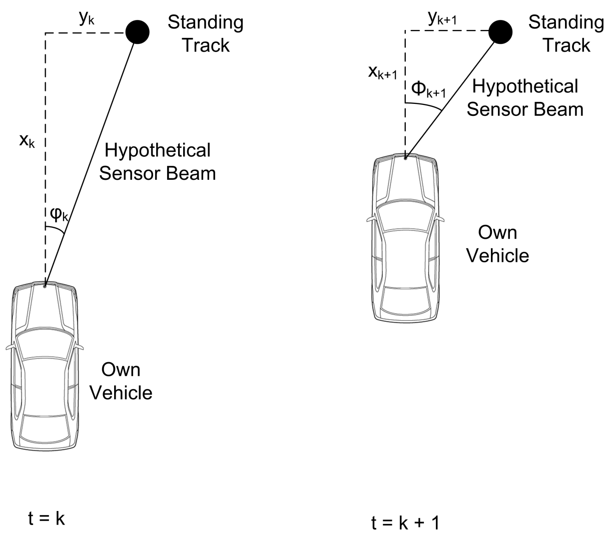


Figure 5.28: Scheme illustrating the geometric effect by a vehicle approaching straight to standing object a hypotheses represented by a track. In case of no lateral misalignment the temporal changes of the parameters will follow: $y_k = y_{k+1}$, $x_{k+1} < x_k$ and $\phi_k < \phi_{k+1}$.

Figure 5.28 illustrates the geometric effect by a vehicle approaching straight to a standing object hypotheses represented by a track from time interval $t = k$ to $t = k + 1$. Then the expected behavior of a not misadjusted sensor with respect to its own yaw angle should respect the following physical principles:

$$\psi_{vehicle} \cong 0 \tag{5.9}$$
$$y_k \cong y_{k+1} \tag{5.10}$$
$$x_{k+1} < x_k \tag{5.11}$$
$$\phi_k < \phi_{k+1} \tag{5.12}$$

This means that by a straight motion of the evaluated sensor, in this case also of the own vehicle, the vehicle yaw angle will follow $\psi_{vehicle} \cong 0$. According to this the lateral offset y_{k+1} should remain constant as well as the longitudinal and angle offsets, x_{k+1} and ϕ_{k+1}, should decrease by standing tracks. Based on the deviation between this premise and the parameters obtained in the driving environment a soft membership determination to two evidence states like *geometry* and *no geometry* may be performed with aid of a sigmoidal function discussed in section 3.2.2. As soon as the membership can be accomplished the influence of this feature on the identification can be represented by a dependence model exemplified in table 5.8.

hypotheses ↘ evidences	Geometry	No Geometry
Sensor Misalignment	0.0	1.0
No *Sensor Misalignment*	1.0	0.0

Table 5.8: An example of a dependence model between the geometry effect for standing tracks and sensor misalignment hypothesis states.

Again the dependence values in table 5.8 are composed with aid of prior knowledge by the analysis of experimental tests. According to the approach proposed to detect and identify sensor failures, features influence are just taken into account when their occurrence exceed determined threshold values.

Analog to the detection and identification of sensor blindness the sensor status information about its own misalignment is a valuable information and is taken into account. As soon as this feature is available it will contribute for the identification of the correspondent sensor misalignment following the probabilistic principles and the proposed architecture.

Investigating the information of dissimilar and partially redundant sensors provides a further source for the extraction of sensor misalignment symptoms. In doing so the behavior of object hypotheses in form of tracks evaluated in the previous track layer are analyzed and tested against plausibility models. A relevant effect observed in the driving environment in case of sensor misalignment is illustrated in figure 5.29. It exemplifies the appearance of inexistent object hypotheses due to sensor misalignments. Figure 5.29 shows the original coverage area of two hypothetical sensors as expected and configured for the complete system. Additionally it illustrates the actual sensor coverage area if sensor 1 is misadjusted. Due to this misalignment failure a object hypotheses that should only be acquired by sensor 2 will be wrongly reproduced in the coverage area of sensor 1.

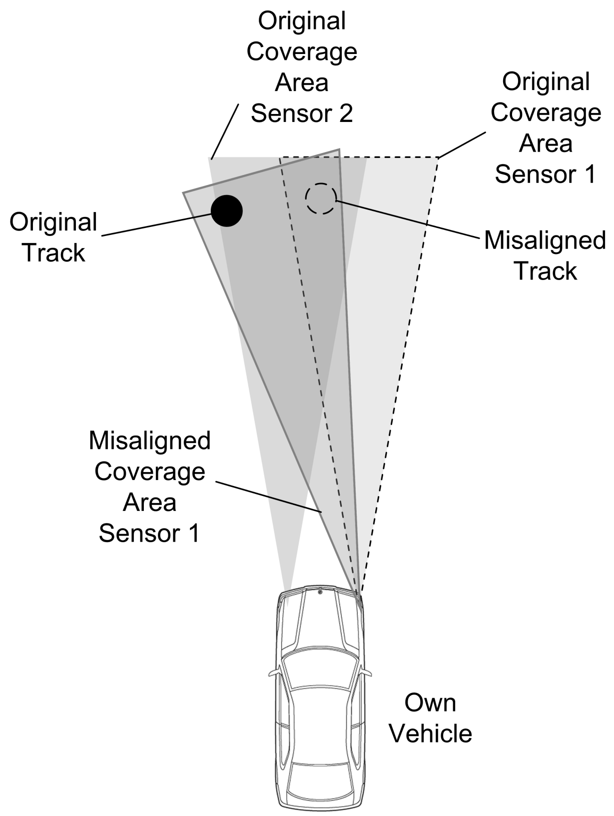


Figure 5.29: Scheme exemplifying the appearance of ghost tracks due to misalignment failures.

The misaligned track illustrated in figure 5.29 will be generated wrongly by the sensor fusion mechanism due to the original sensor assembly position do not match with the current one. Thus this misalignment failure will induce the generation of parallel tracks, which will be acquired in the driving environment within a relative separation angle between each track. This angle corresponds the amount of misalignment of the involved sensors.

Once the occurrence of parallel tracks within a constant separation angle exceeds a specific threshold value, sensor misalignment failure can be detected for the whole system. However the explicit assignment of this failure to a specific sensor is not feasible with this extracted symptoms. A further evidence that is used to support this explicit assignment is based on considering tracks confidence in the overlapping coverage area of several sensors. According to the symptoms discussed by evaluating tracks, a low belief level will be assigned to it in case of the absence of sensor redundancy (section 5.2). This means that tracks generated in this overlapping coverage have to be updated by the correspondent sensors in order to be considered reliable. Then parallel tracks with low confidence level may support the explicit identification of sensor misalignment failure. Although the effectiveness of this symptom can just be confirmed within overlapping coverage areas, it contributes significantly for the sensor configuration implemented in the case study for this thesis (see section 4.1).

According to the deviation between the premise of parallel tracks with low confidence and the parameters obtained in the driving environment, a soft membership determination per sensor can be performed. Again this membership is achieved with aid of a sigmoidal function discussed in section 3.2.2 for the two evidence states: *parallelism* and *no parallelism*. As soon as the membership can be accomplished the influence of this feature on the identification can be represented by a dependence model exemplified in table 5.9. Here the uncertainty of this symptoms by influencing the failure hypothesis states are represented by the influence value of 70%.

hypotheses $\searrow$ evidences	Parallelism	No Parallelism
$Sensor\ Misalignment$	0.7	0.3
$No\ Sensor\ Misalignment$	0.3	0.7

Table 5.9: An example of a dependence model between the parallelism effect for unreliable tracks and sensor misalignment hypothesis states.

After calculating the dependence of all available features, the final belief for this failure hypothesis states can be accomplished. This is performed according to the Bayesian principles discussed along with the proposed approach section in section 3.2.2. Figure 5.30 shows the proposed probabilistic network for classifying sensor misalignment.

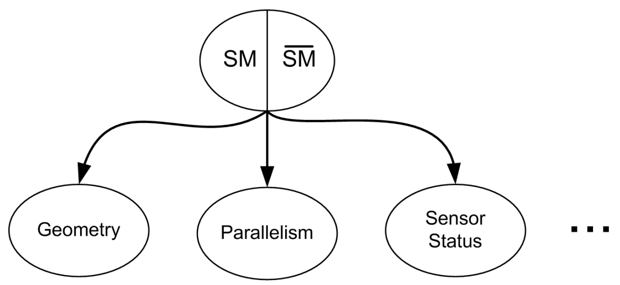


Figure 5.30: Scheme illustrating the proposed probability network for classifying sensor misalignment from one specific sensor. The abbreviations SM and $\overline{SM}$ represent the states sensor misalignment and no sensor misalignment respectively.

Based on experimental tests the proposed concept along with the extracted features could be verified. For it several test runs were also performed in the driving environment and sensor misalignment was simulated by misaligning incoming sensor measurements. This is achieved by modifying the actual acquired lateral angles. Figure 5.31 shows a diagram illustrating the belief distribution from a example of misalignment failure for a test run.

By means of the diagram in figure 5.31 an explicit identification of a simulated misalignment failure could be achieved. This snapshot from a test run shows in the belief distribution that even before misalignment simulation has started, the uncertainty by identifying this failure is reflected by a belief value up to 40%. This can lead back to the occurrence of misalignment symptoms even by the absence of this failure. Again in order to improve the certainty of this classification a tradeoff has to be determined between the completeness and complexity of dependence models and probabilistic network

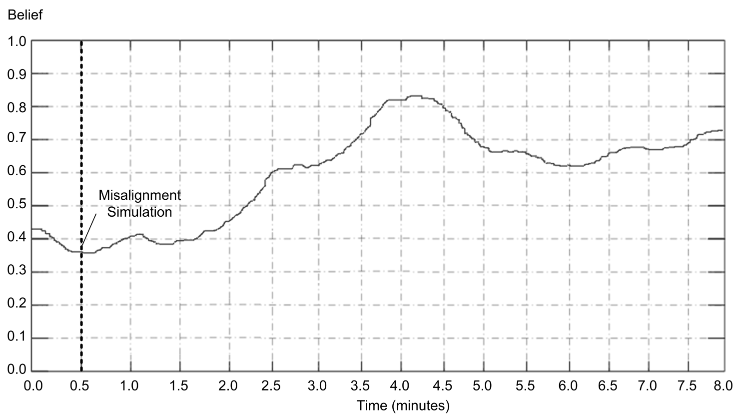


Figure 5.31: Diagram illustrating the belief distribution from a specific sensor failure, misalignment, for a test run.

A further belief distribution of another test run is illustrated in figure 5.32. As in the diagram shown in figure 5.31 the explicit detection an identification of sensor misalignment from a specific sensor could be achieved by a belief value exceeding 80%. Also here the main challenge by classifying sensor misalignment consists of reducing the influence of false alarms that corresponds to misalignment symptoms even in the absence of failures. Furthermore the extraction of further relevant symptoms, which allow the undubious classification of misalignment could stabilize even more the belief distribution.

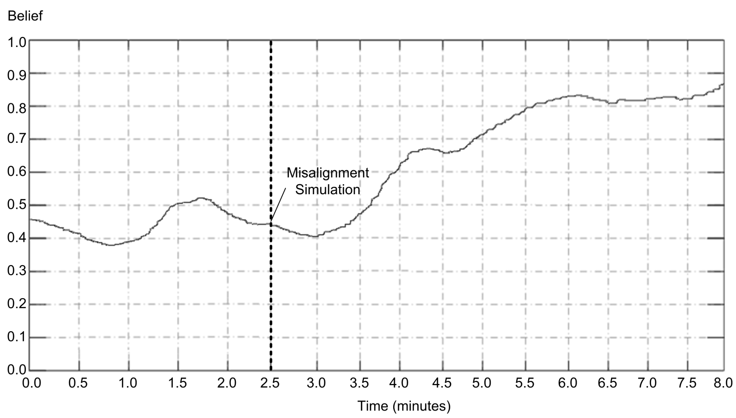


Figure 5.32: Diagram illustrating the belief distribution from a specific sensor failure, misalignment, for a test run.

5.4 Summary

This chapter presented some examples of experiments for the validation of the proposed MFTT concept. It covered the whole approach from the investigation of sensor faults and the evaluation of sensor measurement through the validation of tracks until the detection and identification of sensor failures.

The evaluation of sensor faults was performed primarily with the intention to achieve a quality measure to sensor measurements. In most of the cases the detection, but not the identification of sensor faults is relevant from the point of view of ADAS. In other cases available features or symptoms from these kind of faults are not sufficient to identify sensor faults unequivocally. In case of sensor faults being relevant for determined ADAS implementation and the symptoms for an explicit identification can be extracted, this is performed. A valuable example of a specific sensor fault, especially for lidar sensors, is represented by ground clutters. In section 5.1.1 the nature of ground clutters were investigated by means of simulations and experimental tests. The approach validation was performed by several test runs under different weather and traffic conditions. The algorithms expected performance was confirmed by the experimental test in the real driving environment. Thus combining the evidences obtained by the specific and unspecific sensor faults contribute for the evaluation of the measurement regarding its reliability. For the sake of feasibility a extensive validation of this measurement evaluation with labelled data was not performed.

As inputs for the evaluation of tracks in correspondent layer, the results obtained in the measurement layer were applied. Along with the evidences obtained from measurement specific track evidences like the sensor redundancy symptom were extracted. Again the validation of this proposed approach were performed by several test runs in the real driving environment. A good performance of the algorithms could be confirmed and the obtained track confidence values of belief worked as basis for the detection and identification of sensor failures.

The final step of the evaluation mechanisms consists of detecting and identifying sensor failures. Thus the examples of sensor blindness and misalignment were investigated. By an extensive analysis of the evidences that can originate this kind of failures experimental test runs in the real driving environment were performed as well. For most of the investigated situations the symptoms delivered a good performance. However the robustness of the identification mechanisms could be strongly increased by means of a further dissimilar sensor like a camera system.

6

Conclusion and Outlook

The main goal of future advanced driver assistance systems consists of increasing convenience, comfort and safety of driving. They might either predict a critical situation by warning the driver, or they may start automatic procedures in order to reduce accident severities. By means of dissimilar and to some extent redundant sensors a description of the events in the environment around the vehicle can be performed. Therefore the reliability of this information and of its processing mechanisms and methods have to be assured. An important aspect consists on how to combine the information of dissimilar sensors in order to obtain a reliable view of the events in the driving environment. Sensor data fusion mechanisms can connect several sources of information so that sensors can act in a cooperation, complementation and supervision form. Another crucial point is how the system should deal by the occurrence of sensor faults and failures, which might affect the system performance drastically. Fault detection and identification strategies offer different knowledge-based procedures that are based on analytical and heuristic information in order to quickly detect the occurrence and identify sensor faults and failures.

According to these premises the proposed approach in this work consists of further developing and combining of sensor data fusion, multiple target tracking (MTT) and fault detection and identification (FDI) methods. The developed and validated concept describes a new technique on how to extract the maximum potentialities of the three mentioned strategies. It aims at the evaluation of the driving environment information regarding its reliability by performing the detection and to some extent the identification of sensor faults and failures. The proposed MFTT approach is based on the extraction of symptoms that enable the evaluation of sensor measurements, tracks and sensor units themselves. These symptoms are extracted with the aid of mathematical models, which may assume white, grey or black box characteristics. Here a parallel between MTT and FDI could be performed. The residual between process and observer used to execute the target tracking might deal as a symptom for sensor fault and failures. Once specific fault or failures symptoms could be extracted their influence amount can be determined. These influence calculations are based on soft decision criteria based on human decision process under uncertain circumstances. In a first step fault and failure symptoms are categorized by fuzzy membership functions. Afterward their influence on the final or intermediary hypothesis assumption is determined by means of dependence models. The final hypothesis assertion is performed by means of probabilistic networks. It enables a graphic visualization of the dependence between symptoms and hypotheses as well as the calculation of hypothesis probabilities with aid of the Bayes' theory.

Evaluations with experimental test runs in a real driving environment show that the proposed concept deliver precise results by the evaluation of object hypotheses in form of tracks. Furthermore the detection and identification of ground clutters as an example of specific sensor faults could be accurately detected as well. With regard to sensor failures, blindness and misalignment could be precisely detected and identified with the available symptoms. The proposed architecture along with the probabilistic network approach offer a tradeoff between the complexity of symptoms as well as dependence models and the precision and correctness of the results. Additionally the proposed concept architecture enable a flexible extension of symptoms, faults and failure hypotheses. This can be performed by adding new nodes to the probabilistic network and build new symptom and dependence models.

Possible further application might consist of the integration of sensors with different measurement principles in order to enable the extraction of features related to object dimensions or consistency. In doing so not only a more specific classification of object hypotheses in form of track can be achieved, but also a more precise tracking approach might be performed. Moreover the achieved sensor failure identification might be applied to correct or ignore sensor measurements an thus improving results reliability. Additional tests with different sensor configurations might prove the stability and scalability of the algorithms. A further considerable alternative for the extension of the proposed approach is to develop suitable models and antimodels for objects in order to fit the MFTT method into different application areas (e.g. engine control).

Bibliography

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50, NO. 2:174–188.

Ayoubi, M. (1996). Fuzzy systems design based on hybrid neural structure and application to the fault diagnosis of technical processes. *Control Engineering Practice*, 4:35–42.

Bak, T. (2000). Lecture notes - estimation and sensor information fusion. Technical report, Aalborg University - Department of Control Engineering.

Bar-Shalon, Y. (1990). *Multitarget-Multisensor Tracking: Advanced Applications*. Arthech Hause, Norwood MA.

Bar-Shalon, Y. and Fortmann, T. E. (1988). *Tracking and Data Association*, volume 179. Academic Press Inc. (London) Ltd.

Barton, D. K. and Leonov, S. A. (1998). *RADAR Technologie Encyclopedia*. Artech House, Norwood - MA.

Basir, O. and Youan, X. (2007). Engine fault diagnosis based on multi-sensor information fusion using dempster-shafer evidence theory. *Information Fusion*, 8, issue 4:379–386.

Basseville, M. (1988). Detecting changes in signals and systems - a survey. *Automatica - Journal of IFAC*, 24:309–326.

Beard, R. V. (1971). *Failure Accommodation in Linear Systems through Self Re-organization.* Massachusetts Institute of Technology.

Bedworth (1992). Probability moderation for multilevel information processing. *Personal Communication*.

Bedworth, M. D. (1999). Source diversity and feature-level fusion. *Defence Evaluation and Research Agency*.

Betta, G. and Pietrosanto, A. (2000). Instrument fault detection and isolation: State of the art and new research trends. *IEEE Transaction on Instrumentation and Measurement*, 49:100–107.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Buntine, W. (1996). *A Guide to the Literature on Learning Probabilistic Networks from Data*. IEEE Educational Activities Department.

Chen, J., Xing, Y., Xi, G., Chen, J., Yi, J., Zhao, D., and Wang, J. (2007). *Advances in Neural Networks - ISNN 2007: A Comparison of Four Data Mining Models: Bayes, Neural Network, SVN and Decision Trees in Identifying Syndromes in Coronary Heart Disease*, volume 4491/2007. Springer Berlin / Heidelberg.

Chow, E. Y. and Willsky, A. S. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29:603–615.

Clark, R. N. (1978a). Instrument fault detection. *IEEE Transactions on Aerospace and Electronic Systems*, 14:456–465.

Clark, R. N. (1978b). A simplified instrument detection scheme. *IEEE Transactions on Aerospace and Electronic Systems*, 14:558–563.

Dasarathy, B. V. (1997). Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*.

de Castro Bonfim, C. (2004). Flexible multisensor fusion. Master's thesis, TU Munich.

Dickmanns, D. (2005). Realis Core: Bildfolgenauswertung und Datenfusion. Technical report, BMW Group.

Dickmanns, E. D. (1997). Vehicles capable of dynamic vision. *International Joint Conference or Artificial Intelligence IJCAI'97*.

DoD, U. S. D. o. D. (1992). *Data Fusion Lexicon*. Artech House, Boston a.o.

Doucet, A., De Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo methods in practice*. Springer Verlag.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley Interscience.

Duma, D. W. and Krieg, K. J. (2005). Dod guide for achieving reliability, availability and maintainability. *Department of Defense - United States of America*.

Dunia, R., Qin, S. J., Edgar, T. F., and McAvoy, T. J. (1997). Identification of faulty sensor using principal component analysis. *AIChE Journal*, 42, Issue 10:2797–2812.

Durrant-Whyte, H. F. (1988). Sensor models and multisensor integration. *International Journal of Robotics Research*.

EDINIEC50-191-15 (1994). Reliability and quality of service - development-related terms. Technical report, IEC.

EFNMS (1998). Draft proposal on maintainance terminology. Technical report, European Federation of National Maintenance Societies.

Eschermann, B. (2004). Safety analysis and standards. Technical report, EPFL - Industrial Automation.

Fawcett, T. (2005). An introduction to ROC analisys. *Pattern Recognition Letters*, pages 861–874.

Frank, P. M. (1987). Advanced fault detection and isolations schemes using nonlinear and robust observers. *10th IFAC World congress - Munich*, 3:63–68.

Gertler, J. (1991). Analytical redundancy methods in fault detection and isolation. *IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden, Germany*, 1:9–21.

Hiifling, T. and Isermann, R. (1996). Fault detection based on adaptive parity equations and single-parameter tracking. *Control Eng. Practice*, 4:1361–1369.

Hilbert, D. and Cohn-Vossen, S. (1999). *Geometry and the Imagination*. AMS Chelsea Publishing, New York, 2 edition.

IEC61508 (1997). Functional safety and iec 61508. Technical report, International Electrotechnical Commission, IEC, Geneva, Switzerland.

IECTR61508-0 (2005). Functional safety and iec 61508. Technical report, International Electrotechnical Commission, IEC, Geneva, Switzerland.

Isermann, R. (1992). *Identifikation Dynamischer Prozesse*, volume 1 and 2. Springer-Verlag, Berlin, 2nd edition.

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods - an introduction. *Control Engineering Practice*, 59:639–652.

Isermann, R. (2005). *Fault-Diagnosis Systems - An Introduction from Fault Detection to Fault Tolerance*, volume 1. Springer-Verlag.

Isermann, R. (2006). *Fahrdynamik-Regelung*. Springer-Verlag.

Isermann, R. and Balle, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5:709–719.

Isermann, R. and Ulieru, M. (1993). Integrated fault detection and diagnosis. *IEEE / SMC Conference Systems Engineering in the Service of Humans*, 1:743–748.

Issa, B.-R. (2007). Entwurf eines Bodenechofilters für Lidar-Sensoren im Bereich der Fahrerassistenzsysteme. Master's thesis, TU Munich.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag, New York.

Jones, H. L. (1973). *Failure Detection in Linear Systems*. PhD thesis, Department of Aeronautics, MIT - Cambridge, MA.

Kawasaki, N. and Kiencke, U. (2004). Standard platform for sensor fusion on advanced driver assistance system using bayesian network. *IEEE Intelligent Vehicles Symposium*, pages 252–255.

Kohavi, R. and Provost, F. (2004). *Applications of Data Mining to Electronic Commerce*. Springer-Verlag, Netherlands.

Kokar and Kim (1993). Review of multisensor data fusion architectures. *Proc. IEEEE*.

Kopischke, S. (2000). *Entwicklung einer Nobremsfunktion mit Rapid Prototyping Methoden*. PhD thesis, TU Braunschweig, Germany.

Krishnaiah, P. and Rao, C. R. (1988). *Handbook of Statistics*. North-Holland Publishing Co.

Lendaris, G., Pap, R., Saeks, R., Thomas, C., and Akita, R. (1994). Hardware neural network implementation of tracking system. *Neural Networks for Signal Processing - 1994 IEEE Workshop*.

Leonhardt, S. and Ayoubi, M. (1997). Methods of fault diagnosis. *Control Engineering Practice*, 5:683–692.

Leuschen, M. L., Walker, I. D., and Cavallaro, J. R. (2005). Fault residual generation via nonlinear analytical redundancy. *IEEE Transaction on Control Systems Technology*, 13:452–458.

Liang, J. and Wang, N. (2003). Fault detection and isolation based on pca: An industrial reheating furnace case study. *IEEE International Conference on Systems, Man and Cybernetics*, 2:1193–1198.

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.

Luo, R. C. and Kay, M. G. (1992). *Data Fusion and Sensor Integration: State-of-the-Art 1990s, in Data Fusion in Robotic and Machine Intelligence*. Academic Press Inc., -, 2nd. edition.

Maybeck, P. S. (1979). *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. Academic Pr.

Mehra, R. K. and Peschon, J. (1971). An innovative approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7:637–643.

Mehrotra, K., Mohan, C. K., and Ranka, S. (1996). *Elements of Artificial Neural Networks*. The MIT Press.

Naab, K. (2004). Sensorik- und Signalverarbeitungsarchitekturen fuer Fahrerassistenz und Aktive Sicherheit. *FAS TU Munich*, pages 1–9.

Omana, M. and Taylor, J. H. (2007). Fault detection and isolation using generalized parity vector technique in the absence of an a priori mathematical model. *Proceedings IEEE Conference on Control Applications - Singapore*.

Pacini, P. J. and Kosko, B. (1992). Adaptive fuzzy systems for target tracking. *Intell. Syst. Eng.*, 1(1):3–21.

Park, S. and Lee, C. S. G. (1993a). Fusion-based sensor fault detection. *International Symposium on Intelligent Control*, pages 156–161.

Park, S. and Lee, C. S. G. (1993b). Uncertainty fusion of sensory information using fuzzy numbers. *Proc. of the 5th IFSA World Congress, Seoul, Korea*, pages 1001–1004.

Patton, R. J., Frank, P. M., and Clark, R. N. (2000). *Issues of Fault Diagnosis for Dynamic Systems*. Springer-Verlag.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Potocnik, P., Govekar, E., and Grabec, I. (2005). Application of psychoacoutic filtering for machine fault detection. *The 8th International Conference of the Slovenian Society for Non-Destructive Testing*, pages 293–298.

Rasshoffer, R. and Gresser, K. (2004). Radar- und Lidarsysteme fuer den Einsatz in Kraftfahrzeugen der Naechsten Generation. *Kleinheubacher Tagung*.

Rothman, P. L. and Denton, R. V. (1991). Fusion or confusion: Knowledge or nonsense? *SPIE Data Structures and Target Classification*.

Rougee, A., Basseville, M., Benveniste, A., and Moustakides, G. V. (1987). Optimum robust detection of changes in the ar part of a multivariable arma process. *IEEE Transactions On Automatic Control*, AC-32:1116–1120.

Satish, L. (1998). Short-time fourier and wavelet transforms for fault detection in power transformers during impulse tests. *IEE Proceedings - Science, Measurement and Technology*.

Sauer, T. (2001). RB-ACC-Radarsensor der 1. Generation. Technical report, BMW AG.

Schramm, U. (1991). Multilayered perceptrons as classifiers for automatic inspections. *IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden*, pages 417–422.

Senega, S. (2006). Eigenschaften von Lidar-Systemen unter Unguestigen Wetterbedingungen. Technical report, TU Munich, Germany.

Sharma, R. K., Kumar, D., and Kumar, P. (2005). Systematic failure mode effect analysis (fmea) using fuzzy linguistic modeling. *International Journal of Quality & Reliability Management*, 22:986–1004.

Silva, R. G., Reuben, R. L., Baker, K. J., and Wilcox, S. (1998). Tool wear monitoring of turning operations by neural network and expert system classification of a feature set generated from multiple sensors. *Mechanical Systems and Signal Processing*, 12:319–332.

Smith, L. I. (2002). A tutorial on principal component analysis. Technical report, University of Otago - New Zealand.

Sorsa, T. and Koivo, H. (1991). Application of artificial neural networks in process fault diagnosis. *IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden*, pages 423–428.

Tsui, C.-C. (1993). A general failure detection, isolation and accommodation system with model uncertainty and measurement noise. *12th IFAC World congress - Sydney*, 6:231–238.

Wald, L. (1998). A european proposal for terms of reference in data fusion. *International Archives of Photogrammetry and Remote Sensing*.

Waltz, E. and Llinas, J. (1990). *Multisensor Data Fusion*. Artech House, Norwood, Massachusetts, 2nd. edition.

Wan, E. A. and Merwe, R. v. d. (2000). The unscented kalman filter for nonlinear estimation. *Oregon Graduate Institute of Science and Technology*.

Willsky, A. S. (1976). A survey of design methods for failure detection systems. *Automatica*, 12:601–611.

Wright, R. G. and Kirkland, L. (1995). *An Intelligent Approach to Sensor Fusion-Based Diagnostics*. IEEE CNF.

Yu, Z.-J., Dong, S.-L., Wei, J.-M., Xing, T., and Liu, H. T. (2007). *Neural Network Aided Unscented Kalman Filter for Maneuvering Target Tracking in Distributed Acoustic Sensor Networks*. IEEE Computer Society.

Zhang, R. and Bivens, A. J. (2007). *Comparing the Use of Bayesian Networks and Neural networks in Response Time Modeling for Service-oOriented Systems*. ACM, New York, NY, USA.

Zheng, B., Chang, Y.-H., and Wang, X.-H. (1999). Comparison of artificial neural network and bayesian belief network in a computer-assisted diagnosis scheme for mammography. *IJCNN'99 International Joint Conference on Neural Networks*, pages 4181–4185.