Cranfield University

Behaviour Monitoring: Investigation of Local and Distributed Approaches

Author:
Dario Turchi

Supervisors:
Dr. Hyo-Sang Shin
Prof. Antonios Tsourdos

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

Centre of Cyber-Physical Systems

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

"Laugh, and the world laughs with you; Weep, and you weep alone."

Ella Wheeler Wilcox

CRANFIELD UNIVERSITY

Abstract

School of Aerospace, Transport and Manufacturing (SATM)
Centre of Cyber-Physical Systems

Doctor of Philosophy

Behaviour Monitoring: Investigation of Local and Distributed Approaches

by Dario Turchi, Hyo-Sang Shin, Antonios Tsourdos

Nowadays, the widespread availability of cheap and efficient unmanned systems (either aerial, ground or surface) has led to significant opportunities in the field of remote sensing and automated monitoring. On the one hand, the definition of efficient approaches to information collection, filtering and fusion has been the focus of exremely relevant research streams over the last decades. On the other hand, far less attention has been given to the problem of 'interpreting' the data, thus implementing inference processes able to, e.g., spot anomalies and possible threats in the monitored scenario. It is easy to understand how the automation of the 'target assessment' process could bring a great impact on monitoring applications since it would allow sensibly alleviating the analysis burden for human operators. To this end, the research project proposed in this thesis addresses the problem of *behaviour assessment* leading to the identification of targets that exhibit features "of interest".

Firstly, this thesis has addressed the problem of distributed target assessment based on behavioural and contextual features. The assessment problem is analysed making reference to a layered structure and a possible implementation approach for the middle-layer has been proposed. An extensive analysis of the 'feature' concept is provided, together with considerations about the target assessment process. A case study considering a road-traffic monitoring application is then introduced, suggesting a possible implementation for a set of features related to this particular scenario. The distributed approach has been implemented employing a consensus protocol, which allows achieving agreement about high-level, non-measurable, characteristics of the monitored vehicles. Two different techniques, 'Belief' and 'Average' consensus, for distributed target assessment based on features are finally presented, enabling the comparison of consensus effects when implemented at different level of the considered conceptual hierarchy.

Then, the problem of identifying targets concerning features is tackled using a different approach: a probabilistic description is adopted for the target characteristics of interest and a hypothesis testing technique is applied to the feature probability density functions. Such approach is expected to allow discerning whether a given vehicle is a target of

interest or not. The assessment process introduced is also able to account for information about the context of the vehicle, i.e. the environment where it moves or is operated. In so doing the target assessment process can be effectively adapted to the contour conditions. Results from simulations involving a road monitoring scenario are presented, considering both synthetic and real-world data.

Lastly, the thesis addresses the problem of *manoeuvre recognition* and *behaviour anomalies detection* for generic targets through pattern matching techniques. This problem is analysed considering motor vehicles in a multi-lane road scenario. The proposed approach, however, can be easily extended to significantly different monitoring contexts. The overall proposed solution consists in a trajectory analysis tool, which classifies the target position over time into a sequence of 'driving modes', and a string-matching technique. This classification allows, as result of two different approaches, detecting both a priori defined patterns of interest and general behaviours standing out from those regularly exhibited from the monitored targets. Regarding the pattern matching process, two techniques are introduced and compared: a basic approach based on simple strings and a newly proposed method based on 'regular expressions'. About reference patterns, a technique for the automatic definition of a dictionary of regular expressions matching the commonly observed target manoeuvres is presented. Its assessment results are then compared to those of a classic multi-layered neural network.

In conclusion, this thesis proposes some novel approaches, both local and distributed, for the identification of the 'targets of interest' within a multi-target scenario. Such assessment is solely based on the behaviour actually exhibited by a target and does not involve any specific knowledge about the targets (analytic dynamic models, previous data, signatures of any type, etc.), being thus easily applicable to different scenarios and target types. For all the novel approaches described in the thesis, numerical results from simulations are reported: these results, in all the cases, confirm the effectiveness of the proposed techniques, even if they appear to be open to interpretation because of the inherent subjectivity of the assessment process.

Acknowledgements

To my supervisors Hyo-Sang Shin and Antonios Tsourdos goes all thankfulness, for giving me the opportunity of undertaking my PhD research and for having been such good mentors. Because of their vast experience and knowledge in the field of autonomous systems they have represented to me role models since the first days of my PhD. With their insight, they have helped me in developing creative solutions and believing in my work, which has not always been easy. I especially thank them for constantly setting me a bit further my comfort zone, which made me grow a lot as researcher. Beyond this, I can't thank them enough for their patience and friendliness.

I am really grateful to Richard Theobald (Selex ES), who has significantly contributed to my research with insightful and critical advices. He has spent a lot of time helping me addressing my research and providing valuable suggestions to improve the quality and presentation of my work.

A special word goes to Pau, the person who has been with me in this adventure since the beginning. He has always been inspiring, supportive, cheerful and sympathetic. An example of positive attitude. A real friend.

I am thankful to my colleagues: Chang How, Sangjun, Inmo, Serena, Kostas and all the other researchers that have joined my group over the years. They have been there for me in the good days and in the bad ones (which was a lot), trying to be helpful and heroically standing my frequent feeling blue.

Of course, thank you to my parents: they were thousands miles away but they have always made me feel their presence during these years, which was really helpful.

Lastly, thanks to all the people who have been close to me during my whole life. Thank you, friends met and lost. Thank you, girlfriends. Thank you, professors and teachers. Thank you Nicola, Marco, Marta, Costantino, Roberta, Simone, Elisa, Stefano, Andrea, Maria Chiara, Diletta, Antonella and all the others. If I have succeeded in this huge step is because of the person I am today, and you all have played a great role in it.

Contents

List of Figures

List of Tables

Acronyms

AB Average-Based. 50, 51, 61, 65, 66, 68, 73

BER Bit Error Rate. 20

BN Bayesian Network. 30, 33, 35

CBAA Consensus-Based Auction Algorithm. 20

CBBA Consensus-Based Bundle Algorithm. 20

CDF Cumulative Distribution Function. 78, 89

CDTT Consensus-Based Distributed Target Tracking. 22

DFA Deterministic Finite Automata. 125

DHT Distributed Hypothesis Testing. 24

DM Driving Mode. 108, 109, 112, 121, 123, 125, 127

ELRT Extended Likelihood-Ratio Test. 88, 93, 105

EO Electro-Optical. 55, 70, 73

FHWA Federal Highway Administration. 56

FIS Fuzzy Inference System. 35, 61, 62

FTMS Freeway Traffic Monitoring Systems. 34

GMM Gaussian Mixture Model. 35

GMTI Ground Moving Target Indication. 55, 70, 73, 74, 95

GP Gaussian Process. 64, 65, 66

GT Ground Truth. 10, 11, 15, 16, 70, 71, 74, 96, 97, 104, 123, 125, 133, 135, 136

HMM Hidden Markov Model. 13, 34, 35

HSI HyperSpectral Imaging. 30

ICF Information-Weighted Consensus Filter. 22

IDS Intrusion Detection System. 24, 30

ISR Intelligence, Surveillance and Reconnaissance. 1

JPDA Joint Probabilistic Data Association. 22

KB Knowledge-Based. 50, 51, 61, 62, 64, 73, 115, 116, 125, 126, 130

KCF Kalman Consensus Filter. 20, 22

KF Kalman Filter. 20, 95

LB Learning-Based. 116, 135

LHS Left-Hand Side. 89

LRT Likelihood-Ratio Test. 87

LTI Linear Time-Invariant. 43, 46

MAP Maximum A Posteriori. 24

MF Membership Function. 61, 62, 63, 64, 68, 77

MLE Maximum Likelihood Estimators. 84

ML-NN Multi-Layered Neural Network. 30

MRF Markov Random Field. 34

MTA Measurement-to-Track Association. 15

MTIC Multi Target Information Consensus. 22

NGSIM Next Generation SIMulation. 53, 56, 57, 58, 104, 128, 143

NN Neural Network. 17, 30, 116, 118, 119, 126, 128, 130

NTP Network Time Protocol. 25

OS Operating System. 30

PCA Principal Components Analysis. 30

PDF Probability Density Function. 78, 79, 82, 84, 85, 88, 98, 102

regex Regular Expression. 17, 108, 112, 113, 114, 115, 116, 118, 119, 121, 123, 124, 125, 126, 127, 128, 129, 130, 131, 134

RF Radio Frequency. 1

RHS Right-Hand Side. 91

SA Situational Awareness. 49

SD Standard Deviation. 72, 73

SDS Sleepiness Detection System. 33

SIA Stochastic, Indecomposable and Aperiodic. 138

SRNP Self Routing Network Protocol. 26

SVM Support Vector Machine. 30, 31

T2TA Track-to-Track Association. 15

TOI Target Of Interest. 11, 17, 30, 50, 51, 66, 68, 69, 70, 75, 77, 86, 87, 92, 93, 95, 96, 97, 101, 104, 105, 107, 108, 116, 119, 120, 133, 134, 135

TOIW Target Of Interest Warning. 95, 96, 119

TSK Takagi-Sugeno-Kang. 61, 68

UAS Unmanned Aerial System. 1, 8

UAV Unmanned Aerial Vehicle. 55, 70, 73, 95

Chapter 1

Introduction: Background and Motivations

1.1 Autonomous Vehicles in Monitoring Applications

Recently, there has been a noticeable increase in the interest about the study of networked systems of unmanned vehicles: this is due to the progressive, simultaneous reduction in costs and increase in embedded computational resources of autonomous vehicles. The introduction of cooperation in teams of agents can greatly improve operational effectiveness, both for civilian and military operations: groups of autonomous vehicles that operate on the basis of coordination principles are characterised by greater efficiency and operational capabilities with respect to agents performing solo missions. Furthermore, redundancy in the number of agents allows for a more reliable mission accomplishment and greater robustness to faults, implementing, for example, a task-reallocation procedure if a vehicle become unavailable. In order to obtain such a robustness, vehicles behaviour must be defined in a distributed manner, on the basis of information both locally known and provided by other active agents within the communication range.

Potential applications for single- and multi-vehicle systems include combat, perimeter patrol, intelligent transportation, contaminant cloud extension estimation, hazardous material handling, forest fires monitoring, cooperative robotics, distributed estimation, space-based interferometers and dynamic networks of sensor.

A relevant problem, currently of great interest, is the development of *Intelligence, Surveillance and Reconnaissance* (ISR) capabilities in urban, extra-urban and maritime scenarios, both for civilian and military applications. An *Unmanned Aerial System* (UAS) performing ISR tasks and communicating with a ground station is illustrated in

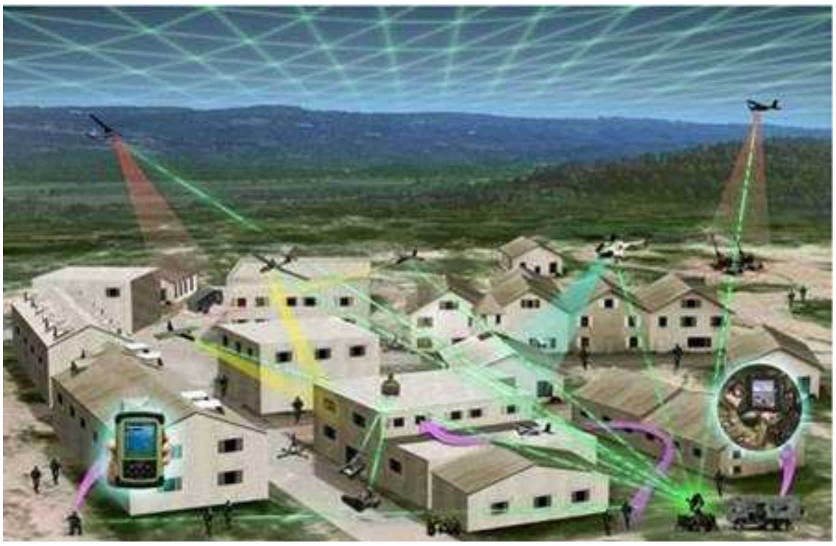


FIGURE 1.1: Representation of a heterogeneous unmanned aircraft systems, [168]

Fig. 1.1. ISR applications should be capable of monitoring, e.g., relocations of enemy troops on a battlefield or anomalous behaviours of generic targets (cars, boats or persons) in a civilian context. Other than tracking and monitoring targets, ISR systems should also be capable of inferring additional information on the basis of actual measurements and additional data previously provided: tasks commonly required are target classification, assessment of the observed behaviours and evaluation of the threat levels associated with the targets. Monitoring tasks usually consist of long, monotonous and perhaps dangerous (e.g. in the case of a military scenario) missions, which are thus inadvisable, or even impossible to accomplish for a human pilot. Such tasks, on the other hand, seems perfectly suitable to a UAS comprising multiple vehicles equipped with similar or heterogeneous sensors. This kind of multi-agent systems, if properly managed, results indeed more versatile and sensibly cheaper than solutions based on manned vehicles. With reference to crowded environments like cities and towns, classical sensing approaches, based for example on radar measurements, could be unsatisfactory or even impractical because of the great number of targets, their proximity and reflections for *Radio Frequency* (RF) signals due to buildings. Even if correct localisation of a target would be possible, typical ISR applications, as mentioned before, are required to extract additional information, providing target identification and, possibly, threat assessment. More sophisticated measurement and tracking techniques are thus required, involving information pertaining to different domains, and thus collected by different types of sensors (e.g., radar, audio and vision systems).

1.2 Consensus Problem

The *'consensus problem'* is the problem of getting a set of agents having different information and different perspective to agree on some information, e.g. a data value, a course of action or a decision. The mentioned agents could be, for example, experts [1] providing their interpretations of an event or interacting nodes of a decentralized/distributed system, which usually perform autonomous sensing and guidance operations.

In the previous section, numerous applications for autonomous vehicles have been presented, with special emphasis on the cases where multiple agents collaborate in order to perform a task. For the implementation of such applications, it is required to develop advanced coordinated control capabilities, including rendezvous, flocking, swarming, attitude alignment, formation control, cooperative search, payload transport, task and role assignment and air traffic control. The implementation of these capabilities can be facilitated if the agents in the team share a consistent view of the mission scenario and objectives. For example, considering a 'formation keeping' task, agents have to exchange information about their relative position and to modify their dynamics accordingly, so that a predefined formation shape can be achieved and preserved. Numerous techniques have been proposed over the years for addressing the formation keeping problem, based on, e.g., leader-followers schemes ([26, 33, 55, 112, 121, 142, 143, 150]), virtual leader methods ([73, 173, 174]) and pure-behavioural approaches ([12, 56]). With reference to these solutions, an actual consensus about quantities of interest is not achieved, but rather the necessary formation-keeping action is exerted by means of techniques like artificial attractive/repulsive forces, potential fields or simple multi-loop control systems.

A slightly different case is the one where the interactions between agents concern actual information rather than dynamics aspects. In such cases, the consensus problem has been largely investigated and addressed by means of algebraic (linear averaging) and stochastic (linear/logarithmic '*opinion pools*') tools; examples are provided in [11, 18, 30, 167], where medical and topographic problems are addressed. The problem with these solutions is that they implement a *'centralized'* approach where all the information is available to a single agent, which can then take a decision. Consider however the scenario where multiple assets take measurements about a target: information is clearly generated in a decentralised manner, and it requires to be aggregated, or fused, for obtaining a unique, seamless reading of the situation monitored. Such process might not be performed by a central node for reliability (single point of failure) or communication burden reasons, and a decentralised architecture for data elaboration and fusion could thus be required. The use of 'message passing' or 'flooding' techniques permits the nodes

[1] Actual people or expert systems.

to share the information they have with the rest of the network, allowing all the agents to independently execute a centralized solution. Once the information has been distributed, if all the nodes perform the same elaboration process, consensus within the network is guaranteed. This approach, which can be somehow considered decentralised, even if preventing some of the issues associated with centralized solutions, e.g. the single point of failure, still has serious drawbacks. Examples are the relevant communication burden, the inefficient use of computational resources[2] and, most important, the necessity to exactly know how many and which nodes are involved in the process.

With reference to both the cases here briefly described[3], an elegant, distributed (not symply decentralised) solution has been identified and largely adopted for allowing co-operating agents to achieve some sort of agreement about relevant information for the problem addressed. These solutions consist in dedicated iterative interaction rules denoted as '*consensus protocols*' (or algorithms), which allow, e.g., taking a shared decision, achieve coordinated motion or perform distributed estimation. The distributed nature of these protocols makes it possible to overcome some of the drawbacks previously mentioned, that is to reduce the communication burden (each node communicates with a reduced subset of agents), calculation optimisation (each node perform a partial, different component of the overall computation) and, most important, knowledge about the structure of the whole network of agents is not required, thus allowing nodes to join and detach without any effect on the effectiveness of the consensus process. For these reasons, consensus protocols will be the tool leveraged in the remaining of the thesis for achieving agreement among agents.

1.2.1 Consensus Protocols: Brief History and Applications

The consensus problem has a long history in computer science, especially within the field of *automata theory* and *distributed computing* ([84]). In this context a well-known problem is the 'transaction commit problem', which arises in distributed database systems ([34, 53, 68]). The problem is for all the data manager processes that have participated in the processing of a particular transaction to agree on whether to install the transactions results in the database or to discard them (if some data managers were, for any reason, unable to carry out the required transaction processing). Regardless of the taken decision, it is essential for all of the data managers to agree on that in order to preserve the consistency of the database. Another problem in the computer science field where consensus between nodes in a network is required, is that of clock synchronisation: some

[2]all the nodes consume energy for performing the exactly same calculation

[3]Dynamic synchronisation and information agreement.

of the most relevant protocols addressing such a problem are *Reference Broadcast Synchronization* (RBS, [37]), *Timingsync Protocol for Sensor Networks* (TPSN, [52]) and the *Precision Time Protocol* (PTP, [36]).

Other than computer science field, consensus problem is closely linked to the study of *self-organising networked systems*, that are composed of locally interacting mobile or static nodes equipped with sensing, computing and communication devices: such nodes will be referred in the following as *agents*. With reference to this kind of networks, the consensus problem was introduced, even if indirectly, with the pioneering works of Reynolds ([134]), describing distributed cooperative models for flock, herds and schools, Vicsek ([160]), where collective motion of a swarm of agents is obtained through local agreement about vehicle's heading, and Jadbabaie ([62]), that provides a theoretical explanation of the cooperative behaviour achieved from the Vicsek model. The theoretical framework for posing and solving consensus problems for networked dynamic systems was introduced by Olfati-Saber and Murray ([107, 139]) on the basis of the earlier work of Fax and Murray ([46, 47]). These work, together with the results from Moreau ([93]) and Ren and Beard ([130]), paved the way for more recent advances in study of consensus algorithms and their convergence characteristics under various conditions for the underlying communication network and for consensus subject of various nature. Consensus principles have been applied for solving some classical problems in teams of agents, such as rendezvous ([76, 77, 88]), formation control ([47, 67, 71, 80, 122]), flocking ([32, 94, 103, 151, 153, 159]), attitude alignment ([70, 124, 128]) and distributed estimation ([100, 109, 145, 176]). A detailed review of the available literature for the consensus problem is provided later in Section 2.1

1.2.2 Reasons and Basic Principles

As stated, in the recent years, there has been a huge surge of interest in the study of multi-agent networked systems: further to reliability reasons, this was mainly due to the whole new task capabilities of such systems, which are achieved by means of cooperative approach. A key topic in recent research on autonomous vehicles has thus been the analysis of either centralised or decentralised cooperation techniques allowing the multi-agent system to operate in a reliable and efficient manner.

Among the numerous solution proposed, consensus protocols are iterative interaction rules that enable a network of agents to agree upon quantities of interest, which are usually critical for achieving a coordination/cooperation result. Such process can be applied to generic information: e.g., sensor measurements, estimations, local decisions and assessment or even kinematic and dynamics quantities. The current local value of this information is the *consensus* (or *information*) *state*, and it is updated by the consensus protocol until agreement is obtained. Examples of quantities that could be used as information state are:

- Rendezvous time;
- Centre position and shape of a formation;
- the direction of motion for a multi-vehicle swarm;
- Length of a perimeter being monitored;
- Probability that a certain event has happen (e.g., a military target has been destroyed).

A consensus algorithm defines which type of information is shared among neighbours, i.e. the nature of the consensus state, and how each node updates its state on the basis of the information received, i.e. which *update (consensus) function* to adopt. The update function can be a continuous-time or discrete time expression and defines for each agent the new state or the state time derivative respectively, on the basis of its current state and state information received from agents within a set referred as *Neighbourhood*. The neighbourhood of an agent can be defined on the basis of communication range limitations or by design, thus limiting the number of exchanged messages and minimizing the communication effort ([40, 41]). The distributed approach based on consensus finds indeed its main difference and advantage respect to classical decentralised approaches in the fact that each agents do not have to communicate with all the other nodes in the network but just with a subset of them, still preserving a consistent coordinated behaviour for the entire team. In most of the works available in literature, the consensus state is expressed through a scalar value or a vector of real numbers, thus allowing for the application of consensus protocols performing either linear or non linear operations on the states. In the former case the updated state for an agent (or its time derivative) can be a linear combination of its previous state and the states of neighbouring agents; such algorithms are referred as *linear-consensus protocols*. The most common example is that of the *average-consensus* where consensus about the possibly weighted average of the initial states of the nodes is achieved performing only linear combinations of the state values over time. Otherwise, when general operations are applied to not necessarily real-valued states, a *nonlinear-consensus protocol* is adopted, which is the case, e.g., of:

Max-Min Consensus: consensus result is the the maximum or minimum of the initial states ([28, 114])

Product Consensus: reach an agreement about the product of the initial states ([110])

Power Mean Consensus: consensus is obtained about the (weighted) power mean of the initial states ([28])

***f*-consensus:** each agent can calculate a family of functions of the initial states ([28, 57])

Nonlinear consensus protocol can be used for obtaining agreement about states of peculiar nature, like logical values ([40, 41]) or sets ([39, 42]). Such types of states result useful when the consensus process involves a shared decision rather than a distributed estimation.

To the best of the author's knowledge, consensus problems have always been considered as problems on graphs, since these seems to be the simplest and most natural representation of the communication network connecting the agents. By adopting a graph representation, each networked agent is considered as a graph node and the presence of an edge between two nodes represents the possibility for the agents to communicate. Directed and undirected graphs are considered in the case of unidirectional or bidirectional communication between the agents respectively. The most general case of consensus problem can be addressed by adopting a weighted switching topology graph. Edges weights allow modelling reliability, time-delays and bandwidth limitations characterising the underlying communication network. Furthermore, the time-varying structure of the graph accounts for the fact that communication links could be introduced and removed over time due to agents movements and communication range limits. Graph theory has a crucial role in the analysis of consensus protocols since both algorithms convergence properties and speed of convergence depend on topological characteristics of the associated graph. Roughly speaking, a necessary condition for the convergence of a consensus protocol is the presence of a spanning tree in the underlying graph. This intuitively means that information can flow from an agent to any other in the network in a finite time (or number of steps). In the case of linear consensus this condition is also sufficient ([105, 107]). Convergence speed for linear consensus protocols (and also for some nonlinear ones) usually depends on the *algebraic connectivity* of the graph ([110, 131]), which is measured as the the second-smallest eigenvalue of the Laplacian matrix associated with the graph. Another parameter of interest is the diameter of the graph, which defines the rate of convergence for some consensus problems operating on discrete variables ([39, 40, 42]), that is when the quantity of interest for the protocol is, e.g., an integer, a boolean or the element of a set. All the notions about graph theory referred in the thesis are presented in Appendix B in detail.

Continuous-time consensus protocols can be further classified on the basis of their *order*: second or higher-order consensus protocols are characterised by a vectorial state where each component is the derivative of the previous variable within the vector, i.e. $X = [x, \dot{x}, \ddot{x}, \dddot{x}, \ldots]$. This definition corresponds with the 'multiple integrator' model, commonly adopted for representing the kinetic/dynamic state of a moving vehicle. For this reason, a natural application for this class of protocols is to "sinchronyze" the movements of networked mobile agents. More specifically, assuming to adopt a consensus protocol with order equivalent to that of the considered vehicle dynamics, higher-order consensus can be applied for obtaining a coordinated dynamics for the agents in a team: by adopting a consensus state composed of states from the dynamic model, agreement between nodes implies the achievement of same position[4], speed and acceleration (depending on the actual order of the protocol) for all the vehicles in the network. Analysis of higher-order consensus protocols and their convergence properties is presented in ([127, 179, 183]). In the case of higher-order linear consensus protocols, the presence of a spanning tree in the graph does not represent a sufficient condition anymore (yet necessary), but more complex conditions on the linear multi-agent dynamical system are required. For example, in the case of second and third order algorithms, agreement can be obtained if and only if the matrix associated with the dynamic of the whole multi-agent system has two or three zero-valued eigenvalues respectively. This rule can not be generalized to protocols with order greater than three: in such cases, algorithm convergence is guaranteed under specific conditions for all the non-zero eigenvalues of the Laplacian matrix.

1.3 Target Assessment: Qualitative approach and Validation

1.3.1 Operator-Support Role

A specific problem that can be framed in the general surveillance context is the monitoring of targets to recognise peculiar behaviours representing evident or hidden threats ([99]). In such case, a relevant drawback of most of the solutions proposed in the past consists in the fact that the UAS only implements tracking functionalities, and a human operator is thus required for assessing and classifying the behaviour of the targets. This approach, requiring numerous and highly skilled human operators, appears to be expensive and unsustainable, especially in cases with a large number of targets. Alleviating the work burden for human operators appears thus as an extremely relevant objective: this can be obtained by signalling to the ground station only when the vehicles that

[4]Of course, reference is here made to individual components of the position vector, e.g. the altitude, and not to the overall position, causing otherwise collisions between agents.

represent a potential menace, possibly together with the level and nature of the associated threat. In so doing, operators are provided with a filtered, summarising picture of the monitored scenario and thus relieved from lots of menial analysis. Such intended supporting role allows considering the monitoring tool as non-critical, in the sense that no actual damages are to be expected in the cases where the system misses a detection or produces a false alarm. Regardless of these soft-constraints for its functioning, a monitoring system can be considered effective, and thus useful, only if it can spot the majority of the actual threats and, most important, it does not produce an excessive number of false alarms. In this last case, indeed, an operator would be tempted to simply ignore the suggestions provided by the assisting monitoring tool. For this reason, the evaluation of the assessment process performances in terms of correct detections and false alarms ratio appears to be an extremely relevant issue that needs to be thoroughly analysed. The operator-supporting tool just described requires the implementation of an automated assessment process that is able to reproduce the inference processes naturally undertaken by a human operator. One of the key issues that arises in these conditions is the need to replicate a not mathematical, and thus not rigorous, assessment approach as the one performed by human beings. These indeed are usually in charge of detecting generic, ‘unusual’ aspects of the behaviour of the monitored targets. Given these considerations, it is not surprising that currently behavioural assessment is predominantly based on the experience and intuition of human operators. Two relevant reasons for this are the lack of mathematical models to refer to and the inability for the system designer to provide an analytical interpretation of the monitored scenario.

1.3.2 The Need for a Qualitative Assessment

In the previous section, the problem has been introduced of target assessment and threat detection, emphasising the great advantages connected with an automated implementation of such tasks.

The key problem in trying to automate the generic-purpose monitoring action provided by a human operator is the subjectivity of the assessment process. This is due to the lack of strict applicable mathematical models and objective rules to implement. An assessment process is indeed qualitative ‘by definition’: any attempt to quantitatively describe it, by means of whatever analytical tool, necessarily leads to the introduction of subjective assumptions and approximations.

By proposing an assessment ‘mechanization’, a model is defined on the basis of the available knowledge about the considered problem. Such knowledge, however, cannot be assumed as the only one ‘valid’. The subjective interpretation involved in the assessment process is evident thinking of the case where, e.g., a monitoring task is assigned to two different human operators: their assessments would probably not be completely matching as each of them would have its own interpretation of what actually represents a threat. Intuitively, under these circumstances, the assessment process can hardly be framed within an 'exact science' approach since it can not be assumed as indisputable but rather opened to different interpretations.

Furthermore, the knowledge leveraged for modelling the assessment process is not necessarily valid under all the circumstances. An effective assessment approach should be capable of taking into account the information about the environment where the targets operate/move (referred in the following as '*context information*'): behaviours totally acceptable in a given context could indeed be suspicious under different conditions. Include context information in the assessment process thus allows achieving a more flexible and reliable classification.

The subjective nature of the assessment process has been one of the most challenging issues encountered during the research project. In the solutions proposed within the thesis, this issue has been addressed by adopting a general, non-strict approach to the detection of irregular behaviours suggesting possible threats . This has been done mainly by:

- providing a qualitative characterisation of the target features of interest;
- defining testing policies based on simple, adaptable thresholds;
- taking context information into account within the testing process;
- introducing a flexible approach for identifying abnormal behavioural patterns.

1.3.3 Validation Issues for Assessment Approaches

In the previous section, the subjectivity issue of the assessment task has been introduced. Such condition can be ascribed to the lack of a *Ground Truth* (GT), which is a univocal description of the reality underlying the monitored scenario. In a classic estimation problem, from the simulation point of view, the real state of the target is known, allowing the definition of error measures and proper performance indexes. However, this is not the case of the generic assessment problem, for which an indisputable classification of the target based on its behaviour, to be adopted as GT and compared to assessment results, is not available.

The main issue with the considered assessment process is not that this has to be carried out in on the basis of common-sense and empirical rules, rather that in most cases a quantitative evaluation of assessment performance can hardly be provided. The assessment case is sensibly different from the one where non-rigorous techniques based on non-specific knowledge (like fuzzy systems) are applied , e.g., for implementing a control system. In the latter case indeed, even if the resulting stability or control performances can not be guaranteed analytically, the effectiveness of the control system together with actual performances can anyway be tested and measured given the existence of a clear, indisputable GT (system state). This usually does not hold for the assessment problem, where being a 'suspect' vehicle does not correspond to a measurable physical state. A key problem that is necessary to tackle when dealing with an automated assessment framework is thus to define how to evaluate the performances of the implemented techniques.

It is possible to conclude that trying to provide an objective performances evaluation for an assessment framework would not be extremely meaningful. This is due to the inherent subjectivity of the framework outcomes, which prevents any common approach to system validation. For this reason, the approach taken in the remainder of the thesis consists in providing numeric results that should not be considered 'absolute' but related to a synthetically-defined GT. This makes reference to a specific interpretation of the problem and is thus, to a certain extent, subjective.

1.4 Aims and Objectives

Given the subjectivity and validation issues described in the previous section, the definition of automated assessment techniques appears as a complex and tricky task. Nonetheless, this problem undoubtedly represents an interesting research field, which could bring great advantages in the automation of repetitive, dangerous or time-consuming surveillance tasks. Furthermore, the application of consensus techniques seems an effective way to easily accomplish the assessment task when more agents are involved.

The general objective of the proposed research is to define a novel approach for the automated identification of target that appears to be 'misbehaving' and thus to be considered a 'possible threats'. These will be indicated in the following as *Targets Of Interest* (TOIs). The proposed solution has to be effective, adaptable to different detection policies, accounting for different types of knowledge and fit to be tailored to various monitoring scenarios. The main necessary steps for accomplishing this results consists in:

- Defining a hierarchic and modular structure for the intended assessment framework, accounting for measurement collection, data processing and information inference;
- Designing assessment techniques implementing different approaches, based, e.g., on field-specific knowledge and learning tools respectively;
- Define reasonable performance metrics for evaluating the framework outcome;
- Verify the effectiveness and the efficiency of the proposed approaches on the basis of the adopted metric;
- Investigate multi-agent scenario cases, where the target assessment has to be produced in a distributed manner.

The main problem with most of the solutions proposed in the past for assessing the targets behaviour (usually solely on the basis of its trajectory and dynamics) is that those approaches seem not easily generalizable to different fields of application. They indeed usually leverage either specific and rigorous mathematical models for representing the expected target behaviour or learning tools for classifying what is "regular" from what is not. Both these approaches, however, are closely linked to the monitoring scenario, the behavioural aspect of interest or the available data. If any of these changes, the considered techniques require to be significantly revised.

The intended contribution of this thesis is the definition of a generic hybrid framework for threat assessment based on a mixture of analytical and empirical techniques. Such approach is intended to be decoupled respect to any specific model or information dataset, allowing for an easy adaptation to new scenarios, possibly involving different target types, without affecting the basic assessment approach.

1.5 Monitoring Approach

This section describes the approach adopted for implementing the monitoring task and for addressing some relevant issues.

1.5.1 Layered Monitoring Structure

The main address of the proposed work is the application of behaviour monitoring techniques to generic targets. This problem can be tackled by making reference to a multi-layered data processing structure involving multiple monitoring agents: a representation of the proposed approach is provided in Fig. 1.2. Each node of the graph is an agent, whose functionalities are classified on three different levels. The three layers may not make reference to the same communication network topology, addressing the case where interactions between agents do not depend only on the communication range[5], but also on some routing principle.

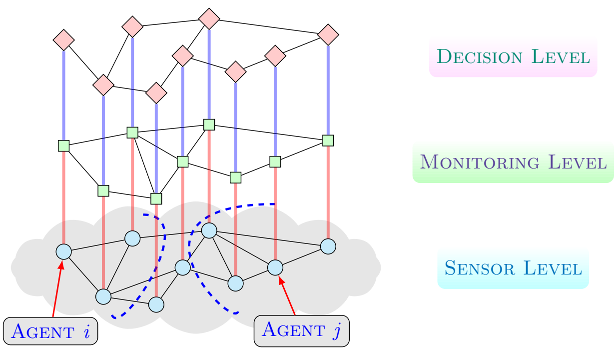


Figure 1.2: Hierarchical structure for data processing

The first layer is referred as '*Sensor Level*', and represents the lowest level of information aggregation: here is where the measurements produced by the sensors placed on different agents can be fused for obtaining single, more accurate, information.

The second, intermediate, level is referred as '*Monitoring Level*': its intended role is to implement distributed monitoring and behaviour analysis, providing a common situational awareness for all the agents in the network. Multiple information sources seeking agreement can be considered within this level, entailing that the consensus subject could be function of heterogeneous data.

The third layer, referred as '*Decision Level*', is intended to implement high-level functions like behaviour modelling, machine learning and decision-making policies, where the last could lead to efficient task allocation and resource deployment. The decision level, for example, could gather information and assessments provided by the lower level, assigning to each agent a reliability score. These can then be used as weights for influencing the consensus protocols executed at the lower layer. More in general, this layer is meant to provide to the monitoring level:

- feedbacks on its performance;
- high-level information about the mission scenario that is not directly measurable and must be thus inferred from the history of measurements or decisions.

[5] Communication ranges are represented for agents i and j with a blue dashed line.

For all these reasons, the most suitable techniques for data processing at this level seem to fall within the computer-science field, like, e.g., *Hidden Markov Model* (HMM), neural networks or fuzzy systems.

1.5.2 Consensus Implementation Within the Different Layers

In the cases where the monitoring action is performed through a team of networked cooperating agents, suitable consensus policies need to be identified. These protocols allow the monitoring agents to undertake their task in a distributed manner and should be defined taking into account the layered approach proposed in the previous section.

The application of consensus protocols to the measurements taken by the single agents (thus at sensor level) has the clear advantage of enabling the implementation of a simple distributed sensor fusion solution. In this case, the system is able to automatically adapt to network topology changes since no fixed fusion scheme is defined. On the other hand, with the consensus approach, the fusion result depends on the network structure, providing probably poorest outcomes respect to a properly designed distributed fusion algorithm based, e.g., on a Bayesian framework. The implementation of consensus technique within the sensor level usually leads to an agreement about a real value, this being the data type associated with most sensor readings.

If implemented at monitoring level, consensus protocols have the objective of reaching an agreement about the target features or final assessment. Unlike consensus at the sensor level, agreement protocols at the monitoring level can return various types of values, such as boolean (e.g., a positive or negative assessment), a distribution (interpreted as stochastic representation of a feature) or set-valued (e.g., agents agree about a set of possible values).

Given these differences, the identification of the most suitable level for the implementation of consensus techniques appears to be a relevant investigation topic. This choice could indeed possibly affect the effectiveness and the efficiency of the overall assessment task. Examples of the implementations of consensus techniques at different levels are provided later in the thesis.

1.5.3 Monitoring Level Implementation

The outlined multi-layer architecture has been considered because it allows implementing a modular approach to the monitoring problem. In so doing, indeed, it is possible to decouple the 'data fusion' and 'information inferring' tasks (both pertaining to the monitoring level) from sensors management[6] and high-level monitoring capabilities (implemented by sensor and decision levels respectively).

The behaviour assessment problem can be considered as a particular case of the more generic and well-know 'anomalies detection' topic. One of the most distinguishing characteristic of the assessment process is, however, that the policies and techniques to be implemented do not deal directly with measurements, but rather with more abstract aspects of the target activities over time. Considering, for example, anomalies within a spectral imagery reading, they can be identified by applying a classification technique directly to the pixel values; such classification can then be verified or disproved by the GT associated with the image. On the other hand, the identification of a misbehaving vehicle usually can not be achieved by means of simple measurements[7]: the threat level associated with a vehicle does not depend only on its actual state, but also on previous trajectories, on the conditions of neighbouring vehicles and on the current environment. This makes necessary to adopt an overall, comprehensive assessment for the observed manoeuvre. This assessment, as already mentioned in Section 1.3 is necessarily subjective, since no absolute, indisputable classification of what is legit and what is not can be provided. Given the need to work with such possibly-abstract features, the most suitable level for the implementation of behaviour assessment technique appears to be the 'Monitoring Level'. This entails to consider information and to provide results within a higher conceptual-level respect to, e.g., measurements and estimations. For this reason, the work presented in this thesis neglects implementation details pertaining to the sensor level (i.e., measurements filtering, *Measurement-to-Track Association* (MTA), *Track-to–Track Association* (T2TA) and estimation) for focusing on the high conceptual-level features characterising the targets behaviour. At the same time, consensus protocols at this level are supposed to reach an agreement about abstract features of the targets or peculiar aspects of the mission scenario, rather than, e.g., simple measurements. On the basis of such considerations, the approach proposed in this thesis is to leverage consensus techniques for implementing distributed assessment and decision-making, rather than for calculating physical quantities about the targets in a distributed manner .

1.5.4 Subjectivity and Validation Issues

The arbitrariness involved in the decision process and the lack of objective performance metrics appear as challenging, tightly coupled issues. The approach taken in this thesis to

[6] Involving the measurement collection, filtering, fusion and estimation processes.

[7] It is here made reference to the case where complex behaviours are the subject of the monitoring action. Of course, simple behaviours like a speeding vehicle can be identified by means of simple sensor readings.

address this matter consists in techniques able to easily adapt to the different subjective interpretations of the problem.

The subjectivity involved in the assessment process has been mainly taken into consideration by adopting a flexible approach to the classification of the targets as possible 'threats'. The accuracy of the assessment so obtained is then evaluated by means of a synthetically-defined GT that is closely related to a specific interpretation of the problem or by analogy with the outcomes of different, assumed reliable, techniques. More specifically, concerning the subjectivity issue and performance evaluation, the key solutions adopted in the thesis consist of:

- providing a qualitative, probabilistic characterisation of the target features of interest;
- defining testing policies based on simple, adaptable thresholds;
- taking context information into account within the testing process;
- introducing a flexible approach to the identification of anomalous behavioural patterns.
- defining a controlled and fully ground-truthed environment for basic performance evaluation
- Defining a GT consisting of the classification outputs provided by well-established learning techniques.

1.6 Contribution

This thesis addresses the problem of target assessment on the basis of high-level behavioural aspects, with reference to both single or multiple cooperating agents. In particular, the thesis outline a distributed assessment framework focusing on three key aspects: (i) Implementation and analysis of consensus protocols within a monitoring environment, (ii) Target characterisation and assessment on the basis of high-level behavioural features and (iii) Target manoeuvre detections by means of pattern matching.

The specific contributions of this thesis can be summarised as follows:

Consensus Protocols Integration: The application of consensus protocols within the monitoring level has been investigated, in order to implement agreed decision-making and assessment. A monitoring application based on high conceptual-level features characterising ground vehicles traffic has thus been taken into consideration: results from performed simulations have shown sensible advantages in implementing the consensus protocol within the monitoring level respect to the sensor level.

Target Behavioural Features In the monitoring application considered for consensus analysis, the feature values have been obtained by means of simple fuzzy-like sigmoid functions. With the purpose of defining a more analytical approach to feature evaluation, a stochastic representation of the target features has been then taken into account. This led to the definition of a novel approach for high conceptual-level features description, based on a hierarchy of probability distributions. Finally, a probabilistic decision policy based on joint features distributions for the classification of targets as TOI has been proposed. Simulations results have confirmed the effectiveness of the method, for both synthetic and real-world data.

Manoeuvres Detection The problem of identifying TOIs in a multi-target scenario has also been addressed by taking an approach that is based on pattern matching techniques. Unlike the other solutions presented in the thesis, where features values are related to the target state at a single time instant, the motion of the targets is now taken into consideration by analysing their position and velocity in time and trying to detect not commonly observed trajectories. A novel approach has been proposed, where manoeuvres of interest (reference patterns) are codified by means of *Regular Expressions* (regexes). Identification performances improvement respect to a classic, well-established pattern matching technique has been then verified by means of numerical simulations.

Lastly, the regex-based matching technique has been extended by means of a dictionary of regular expressions defined leveraging a supporting learning tool, in the proposed case a *Neural Network* (NN). This dictionary contains sets of regular expressions representing behaviours that are considered ‘regular’ to different extents and its use allows the framework to recognise as TOIs vehicles either (i) behaving in a known-unregular manner or (ii) behaving in a manner that is not considered regular. Empirical tests have been performed also for this technique, achieving identification performances similar to those of the supporting NN, but requiring sensibly shorter computation times.

1.7 Thesis Outline

The remainder of this thesis is organised as follows. Chapter 2 presents a comprehensive literature review about the main problems considered within the thesis: consensus achievement in networks of cooperating agents, anomalies detection and target monitoring. Chapter 3 provides the necessary background for the implementation of agreement techniques, by introducing the concept of graph and the fundamental consensus protocols. Chapter 4 outlines the main characteristics of the proposed assessment framework together with the details of a case study that will be taken into consideration for simulations in the remainder of the thesis. In Chapter 5 a basic implementation for the assessment framework is proposed. A distributed approach based on consensus is taken into consideration, providing also a comparison between the implementation of consensus about information at different levels of abstraction. In Chapter 6 a comprehensive implementation approach is proposed for the assessment framework, which is based on a stochastic representation of the behavioural features. An hypothesis-testing decision policy is furthermore provided, and the effectiveness of the proposed approach is tested by means of simulations involving both synthetic and real-world data. Chapter 7 describes pattern matching techniques applied to the behaviour monitoring problem. A 'knowledge-based' and a 'learning-based' approaches are presented, both leveraging regular expressions for the detection of patterns of interest. The performances of the proposed approaches are finally compared against well-established techniques. Finally, Chapter 8 provides conclusions about the proposed work and indications for possible future works.

1.8 List of Publications

The following papers have been published in relation to the research project described in this thesis:

(1) Dario Turchi, Hyo-Sang Shin, and Antonios Tsourdos. Airborne behaviour monitoring using regular-expressions based pattern matching. *IFAC-PapersOnLine*, 48(5): 2934, 2015.

(2) Dario Turchi, Hyo-Sang Shin, and Antonios Tsourdos. Feature-based probabilistic target assessment. Under review at *IEEE Transactions on Intelligent Transportation Systems*.

(3) Dario Turchi, Hyo-Sang Shin, and Antonios Tsourdos. Behaviour Monitoring Using Learning Techniques and Regular-Expressions Based Pattern Matching. Waiting for submission to *IEEE Transactions on Intelligent Transportation Systems.*

Chapter 2

Literature Overview

2.1 Consensus

2.1.1 Early Works

One of the earliest works, and probably the most relevant, addressing the '*Consensus Problem*' is [62], where Jadbabaie et al. analyse the flocking model that has been presented in [161] and provide a theoretical explanation for the observed particles behaviour. The convergence of autonomous particles headings to a common value is here proved, using (undirected) graphs and systems theory. In the paper is furthermore presented a model for leader-followers interaction, proving the convergence of the followers headings to that of the leader. The consensus problem is not explicitly named yet, but it is clear from the purpose of the model provided, i.e. the agreement about a common heading for a group of particles.

2.1.2 Protocols Convergence Analysis

In [107] consensus protocols convergence analysis is provided for directed graphs with switching topology or communication time-delays. The work is focused on balanced graphs, a specific class of directed graphs that allow to solve the average-consensus problem. The relation between the convergence rate and the network topology is taken into account, also considering the necessary trade-offs between performances and the robustness to time-delays. Moreau in [93] provides a strong theoretical framework for the convergence analysis of a coordination process in a network of agents that interact via time-dependent communication links. The agents behaviour model considered in the paper is simple but appealing, and finds application in a variety of field, including synchronisation, swarming and *distributed* decision making. Necessary or necessary and sufficient conditions (depending on the network topology) for the convergence of a given consensus protocol are provided, leveraging graph and system theory tools. In [130] the problem of consensus achievement is addressed for time-varying network topologies with time-varying communication links weights, in both the continuous and discrete time cases. This work extends the results given in [62] to directed graph topologies: the consensus protocol convergence under unidirectional information exchange is here taken into account, which is necessary for applications where bidirectional communication or sensing is not available. On the basis of the problem formulation given in [130], the work in [175] provides convergence conditions in the case of time-varying communication delays: a necessary and sufficient condition is given for the case of fixed communication topologies, while only a sufficient condition is considered for time-varying graphs. In the same paper is showed how the problem of asynchronous consensus can be seen as a special case of time-delayed communication. In [131] an overview of consensus strategies is presented: recent results from literature, application examples and graph and matrix tutorials are briefly yet clearly described. Olfati et al. in [105] present key results about the theory and implementation of consensus techniques within networked systems, such as synchronisation of coupled oscillators, flocking, formation-control, fast consensus, Markov processes, rendezvous in space and distributed sensor fusion. These results are based on the work described in five key earlier papers: [47, 62, 93, 107, 130].

2.1.3 Task Allocation

In [25] consensus algorithms are adopted for robust decentralised tasks allocation to a team of cooperating autonomous vehicles. In this work *Consensus-Based Auction Algorithm* (CBAA) and *Consensus-Based Bundle Algorithm* (CBBA) are presented, providing the algorithm outline, the convergence condition and numerical simulation results. In [115] a comprehensive analysis is provided of CBBA and its possible extensions, among which the most relevant is maybe *CBBA With Relays*, which allows allocating tasks efficiently while preserving the communication network connectivity; this is done by utilising idle agents as communication links. This last version of CBBA estimates communication capabilities of agents solely on the basis of euclidean distances, while in real-world applications communications are influenced by many causes such as path loss, presence of obstacles, multipath fading and packets routing. In [60] an extension of CBBA with relays is presented, which adopts a more sophisticated and realistic communication model, evaluating the network connectivity on the basis of the data-rate and *Bit Error Rate* (BER) thresholds. The paper provides the algorithm outline and numerical simulations confirming the effectiveness of the proposed approach.

2.1.4 Dynamic Consensus

The works referred so far tackle the problem of *Static Consensus*, i.e reaching an agreement on a time-invariant quantity, as function of agent initial states and network connection topology. However, the implementation of decentralised control or estimation usually requires the coordination (and thus some form of agreement) of agents within a dynamic environment: this field of research is referred as *Dynamic Consensus* or *Consensus Tracking*. Spanos et al. in [146] consider the problem of obtaining agreement about time-varying quantities: the consensus algorithm proposed is implemented through a dynamic model acting as an high-pass filter on the local variables that correspond to the consensus states. The authors provide two variants of this dynamic model, accounting for either arbitrary communication time-delays or network 'splitting' and 'merging' events. In [109], *Consensus Filters* are introduced: these are distributed algorithms that enable the calculation of average-consensus about time-varying signals. In the paper, consensus filters are exploited as a tool for distributed sensor fusion and estimation in sensor networks: a dynamic model acting as a distributed low-pass filter on noisy signals is presented, and the convergence of the consensus on the target estimation is proved. The main drawback of the approach adopted in [109] is the fact that sensor fusion is made with no regard to noise statistics, i.e. without considering the covariance of the measurement error. Olfati in [100] adopts consensus filters for obtaining a distributed *Kalman Filter* (KF) from local KFs in *information form*, which are locally executed by the agents. The error covariance and the state estimation are updated using the outcome values from two different consensus filters, which reach an agreement on measurements and covariance innovation respectively. The same author in [101] introduces three novel Kalman filtering algorithms, with the purpose of solving two serious drawbacks of the approach proposed in [100]: the KF proposed in that work requires the sensors in the network to have identical sensing models (thus not supporting heterogeneous sensor fusion) and has a relatively weak performance, comparable with the collective estimation error of n non-cooperative local KFs (this problem was noticed by other researchers and reported in [20]). The Kalman filtering algorithms proposed in [101] implement measurements/covariances consensus (via consensus filters) or state estimates consensus (directly within the Kalman estimate-update step), showing that the best performances are obtained in the latter case: this is referred as *Kalman Consensus Filter* (KCF). In [104], a formal derivation is provided of optimal consensus gains for the KCF, which result however in a covariance propagation algorithm that is not scalable respect to the number of cooperating agents. In the same work a suboptimal scalable version of the KCF is presented, and stability proof for this novel Kalman-like filter is provided using Lyapunov-based technique. It is worth to be mentioned that the problem of *decentralised* Kalman filtering was firstly solved by Speyer [148] and then by Rao et al. [118], but both these methods require a completely connected network, that is with all-to-all links. These solutions are thus not scalable and clearly less efficient respect to the properly distributed Kalman approaches adopted in [100, 104, 109], where each agent communicates only with its neighbours. Consensus Filtering has been considered also by other authors: Yu et al. in [182] present consensus filtering algorithms that allow reaching an agreement on the target state estimation with guaranteed error bounds. In this work, consensus filtering is applied also in the cases of 'Pinning Controllers' (when some network nodes can not be controlled [24, 165, 181]) and 'Pinning Observers' (when some network nodes can not measure all the target states); numerical simulations results are finally provided for both these two methods.

2.1.5 Higher-Order Protocols

When consensus techniques are aimed at achieving coordinated motion for networked agents, it could be useful to implement consensus algorithms of order higher than one. Ren and Atkins in [127] propose a second-order protocol for information consensus, providing necessary and/or sufficient conditions under which agreement can be achieved in directed networks; an example where a consensus protocol is exploited for the altitude alignment of rotary-wing air vehicles is furthermore presented. The same authors in [126] extend their previous work considering also the case of switching communication topology, applying the second-order consensus algorithm to the formation-control problem of nonlinear dynamic agents. Second order consensus tracking algorithms are considered in [123]: consensus achievement in the case of bounded control inputs or directed switching network topology is there proved. Simulations for the formation-control problem involving actual ground robots have been performed, and numerical results confirming the effectiveness of the consensus algorithms are given. In [184] the problem is considered of second-order consensus, in the case that a nonlinear dynamic is required for the agents, i.e the control input is a nonlinear function leading to time-varying velocity for the agents. In the paper is showed that the agreement on consensus variables can be obtained under specific conditions for the '*generalized algebraic connectivity*' of the graph associated with the network. This result is obtained for directed network topology, but the case of time-varying communication link is not considered in this work. Yu et al. in [183] derive some necessary and sufficient conditions for achieving consensus in the case of generic higher-order protocols and undirected communication topology. Here, the concept of '*Stable consensus regions*' is exploited for performing the stability analysis and an application of the generic m-th order consensus to the leader-followers case is presented. This work extends some of the results obtained in [179] and [133], where consensus protocols of second and third order respectively are considered.

2.1.6 Consensus in Tracking Applications

Information consensus has been widely applied to the target tracking problem, trying to exploit advantages offered by a multi-vehicle system for improving the estimation performance. In [108], the authors apply the distributed estimation method presented in [101] to a sensor network of fixed agents with limited sensing range. Olfati et al. in [106] exploit Kalman consensus filter ([101, 102, 104]) and the flocking models described in [103] for implementing an approach where target tracking is coupled to agents deploying, trying to maximise an information theoretic metric; this technique is referred as '*Information-Driven Flocking*'. The stability analysis of the coupled consensus and flocking dynamics is performed, resulting in an asymptotically stable structural dynamics of the flock and estimation consensus asymptotically reached. In [114] Petitti et al. describe a fully decentralised estimation scheme, named *Consensus-Based Distributed Target Tracking* (CDTT), dedicated to heterogeneous sensor networks, where single agents have different sensing ranges. In the proposed scheme, the agents perform their own estimation about the target (correction-prediction for actually sensing agents, simple prediction otherwise), and then a max-consensus algorithm is applied. In so doing, at the end of the consensus step, all the agents in the network will share the most accurate target estimate. The simulation results show improved performance respect to KCF, which does not deal properly with the case of not sensing agents. The problem of distributed target tracking in a sensor network of fixed heterogeneous agents is tackled also in [186]: here an unbiased, optimal estimation filter with a consensus term is derived, both for sensing and non-sensing agents. Simulations results are reported that prove the effectiveness of the method, but no comparison with other techniques is provided. In [64], Kamal et al. exploit average-consensus for implementing a distributed 'maximum a posteriori' estimation method called *Information-Weighted Consensus Filter* (ICF): consensus is achieved about local estimates, and the name refers to the fact that the estimations provided by non-sensing agents (thus obtained by means of simple prediction) have less influence on the result of the consensus algorithm. Simulation results show that the proposed method perform better than KCF and that its performances tend to those of a centralised 'maximum a posteriori' estimator as the consensus step number increases. The same authors in [63] adopt a similar solution for the problem of multi-target tracking problem. The well known *Joint Probabilistic Data Association* (JPDA)([6]) technique is implemented in a distributed manner by means of average-consensus. This approach is referred as *Multi Target Information Consensus* (MTIC) and its efficiency has been proven by simulation results comparing the proposed method with both a different distributed and a centralised solutions.

2.1.7 Nonlinear Consensus

In literature, great interest has been dedicated also to nonlinear forms of consensus. In such cases agreement is not obtained on a numeric value as a linear combination of information from different agents, but nonlinear functions or even different types of information (e.g., binary or set values) are involved in the process. This kind of approach represents a more general form of consensus, which can be useful in many control and robotics problems.

Olfati-Saber et al. in [110] tackle the problem of *Distributed Hypothesis Testing* (DHT) introducing a novel consensus algorithm for belief propagation, denoted as '*Belief Consensus*'. The proposed protocol achieves agreement on the likelihood of the different hypotheses of a test, thus enabling the distributed computation of the product of n belief values (conditional probabilities for a hypothesis) locally produced by n different nodes of a network. Convergence proof is provided for the novel product-consensus protocol leveraging the equivalence respect to classic linear average-consensus techniques. Some considerations are furthermore provided about the algorithm speed of convergence, which results to be exponential and related to the algebraic connectivity of the communication graph. The novel consensus algorithm is finally exploited for the definition of a distributed *Maximum A Posteriori* (MAP) estimator, which allows all the agents in the network to take a shared decision. An application is then presented, where the introduced framework is used for the distributed detection of multi-target formations. In the proposed example, each agent in the network performs its own measurements on the targets and calculates five feature values, on the basis of which the likelihoods of some known formation types are evaluated. Agreement is then obtained about these likelihoods (or beliefs) through the novel protocol and a unanimous decision about the most likely formation is eventually taken.

Fagiolini et al. in [42] exploit a consensus framework within a robotic multi-agent system for implementing a distributed *Intrusion Detection System* (IDS), i.e., a system capable of detecting faulty or malicious agents. These are identified as robots that do not behave consistently with a common set of rules, which are respected by all the other agents in the network. Since each robot has limited knowledge of the system state, decisions on the behavioural correctness of other agents can be taken only in a distributed, cooperative manner, thus requiring consensus between all the monitoring agents. In the paper, a different approach is considered to the consensus problem respect to classic linear protocols: the shared information corresponds to confidence intervals for the presence or absence of vehicles around the monitored agent, and the consensus algorithm reaches an agreement on these intervals by performing set operations. If the agreed confidence intervals cannot explain the behaviour of the monitored agent, this vehicle is classified as faulty/intruder. An analytical proof of convergence for consensus algorithms performing only commutative, associative and idempotent operations in presence of a connected communication graph is given, with guarantees on the maximum number of consensus steps required. The described framework has been applied to a scenario where 5 mobile robots are travelling along a multi-lane road with different maximum speed and may want to reach different desired positions: consensus about the faulty behaviour of one agent is obtained, proving the effectiveness of the proposed solution.

In [39] the clock synchronization between networked agents is obtained as solution of a consensus problem based on set-valued information. In the paper, an extended centralised version of the Marzullo's algorithm ([89]) for time synchronization, which is at the basis of *Network Time Protocol* (NTP), is initially described. Considering a scenario where each node, or agent, expresses its information as a confidence interval, the considered algorithm allows identifying the smallest confidence interval that is contained in the largest number of consensus states, solely by means of operations on sets. A distributed version of the Marzullo's algorithm is then introduced, where each agent computes the shared confidence interval on the basis of information received from the nodes within its neighbourhood. The proposed method operates under the hypothesis of 'bounded inconsistency': only a limited amount of agents can broadcast a measure that is not consistent with the final agreed solution (i.e., empty intersection). It is proved that, under given conditions on the communication graph, the distributed approach converges to the same solution of the centralised Marzullo's algorithm, and agreement is obtained for all the agents in the network. The number of nodes allowed to have not consistent initial information depends on the connection level of the underlying graph, which becomes thus a robustness index for the consensus protocol.

The problem of consensus on logical values is considered in [40] from Fagiolini et al. . In this work the networked agents locally perform event detection and decision tasks on the basis of their own measurements and information provided by neighbouring agents. Each agent can communicate with a subset of the nodes in the network (as usual for consensus problems) and can detect a limited amount of the events of interest, which means that it can directly influence a limited portion of the information state vector. This, in the considered case, is defined as a bit string representing a minimal encoding of the events list. A linear consensus algorithm is proposed that guarantees agreement if, for each event, any agent that can not directly detect the event has a path to an agent that can perform such measurement. The key point of the proposed consensus protocol is the definition of an optimal communication matrix, which allows obtaining agreement in the minimum time (minimum number of steps or *rounds*) and to minimize the number of messages to be exchanged. Consensus is, however, not guaranteed in the case where a faulty node is involved. A second, robust, consensus protocol is then presented, accounting for the presence of misbehaving agents, i.e., agents that perform wrong event detections. In this case, the consensus protocol is not linear and, through the use of an increased number of exchanged messages and redundant minimum-length communication paths, robustness respect to a given number of faulty nodes is obtained. Agreement is thus guaranteed for all the agents in the network, except for the faulty ones. Finally, the two proposed consensus protocols are applied to a distributed intrusion detection problem, both in the case of presence and absence of faulty nodes, proving the effectiveness of the proposed algorithms.

Fagiolini et. al in [41] extend the logical consensus protocol described in [40] introducing an algorithm that allows the agents in the network to define the optimal communication matrix (which defines the message routing protocol used for the communications) in a distributed manner. The proposed *Self Routing Network Protocol* (SRNP) is able to elaborate the optimal communication matrix in a finite time, with $\mathcal{O}(n^2)$ complexity and without any need for centralised elaboration. It is worth to be noticed that the distributed computation of the communication matrix that will be used for achieving the logical consensus is a form of consensus itself, which is executed, in the considered application, just in the initial phase and when a new node joins the network.

2.1.8 Various Works

The problem of obtaining consensus among agents over a finite time-horizon is considered by Wang and Xiao in [164]: in this work, a continuous input command is adopted (thus improving earlier works where discontinuous control actions are applied, [29]), obtaining finite time agreement in the case of undirected time-varying or directed fixed topologies. This work has been extended in [48] taking into account directed switching communication links. In [19] Cao et al. consider the problem of finite-time consensus in the case of undirected fixed connection topology, where the networked agents are characterised by an inherent nonlinear dynamics.

Cortés in [28] considers the problem of designing continuous-time coordination algorithms allowing the network agents to agree on the value of a desired function of their initial states; such function is indicated as χ, and the general problem is referred as 'χ-consensus'. This study has been undertaken by modelling the communication network between the agents in the most general way, i.e. as a switching topology directed weighted graph. Necessary and sufficient conditions for the χ-consensus to be reached are given, with respect to the smoothness of the consensus function and the coordination algorithm, and to the class of allowed weighted digraphs. Later, a systematic approach for defining distributed coordination algorithms with reference to a class of smooth χ-functions is provided: for these algorithms the convergence of the agents states to the desired function is proved, assuming a weakly-connected, weight-balanced communication digraph. A coordination algorithm performing the 'power-mean consensus' [1] is then defined by following the proposed approach. Finally, coordination algorithms for max- and min-consensus are described, but these are not obtained through the general law defined in the paper since max and min functions do not fit the class of smooth χ-functions considered. This paper introduces an interesting, systematic approach to consensus algorithms definition, which allows reaching agreement about quite general functions of the agents initial states. The main drawback is that only continuous-time coordination algorithms are considered, requiring, in real applications, for numeric integration. In [57], Goldenbaum et al. provide algorithms for implementation of f-consensus, obtaining agreement on a generic function of initial states of nodes. The techniques proposed in this work are completely different from the consensus approach used in the previously cited work and deals directly with communication policies and with the communication channel access. Further insight into the problem of distributed calculation of generic functions in a sensor network is provided in [8],

Consensus algorithms, with appropriate selection of the consensus states , are used in [122] for the formation-control issue, both during the phases of 'formation approach' and 'formation keeping'. In this work, it is shown that some of the most common formation control techniques (leader-follower, behavioural and virtual structure) could be thought of as special cases of consensus-based formation-control strategies, where different consensus subjects are defined. In this paper a general second-order consensus algorithm and its stability analysis (based on graph and matrix theory) are initially provided; this protocol is used then as base for the definition of consensus algorithms in the three different formation cases previously mentioned. In all the cases, a second-order dynamic is assumed for the vehicles thus permitting the use of a consensus variable as direct (acceleration) command for the vehicle. Simulation results are provided for the behavioural approach case, considering a scenario with four vehicles characterised by point-mass kinematic, second-order dynamics and saturation on the control effort. Three conditions have been tested: presence of a spanning tree in the communication network, absence of the spanning tree, and presence of the tree together with low gain for the consensus protocol. Results show that only in the first condition the formation shape is preserved during the agents motion.

In [49], Fischer et al. give an analytical proof that consensus between networked agents can not be obtained in presence of a non-cooperating node. This work considers totally

[1] In the proposed case the χ-function is the power mean of the initial states.

asynchronous models of computation and communication that are based on a messages-exchange framework. It is assumed that messages can be received with a certain delay, but it is guaranteed that messages sent by non-faulty nodes will be eventually received from other non-faulty nodes. The considered consensus problem make reference to distributed commitment algorithms, a common issue in distributed computing and distributed database systems. The very simple main result, "*No consensus protocol is totally correct in spite of one fault*", means that any consensus algorithm is prone to a 'window of vulnerability', i.e. an interval of time during the execution of the algorithm in which the delay or inaccessibility of a single process/agent can cause the entire algorithm to wait indefinitely (impossibility to reach consensus), confirming a widely believed tenet in the literature. The results provided in the paper do not imply that such problem cannot be solved in practice, instead, they point up the need for more refined models of distributed computing that better reflect realistic assumptions about processor and communication network capabilities.

Protocol Features	Communication Network Characteristichs			
	Undirected Graph	Directed Graph	Time-Varying Graph	Communication Delay
Linear	[62, 64, 103, 106, 114, 139, 152, 176, 183]	[67, 78, 79, 92, 93, 100, 101, 104, 105, 107, 109, 117, 122, 125, 127, 129–133, 175, 179, 180]	[78, 79, 93, 105, 107, 117, 125, 126, 130, 131, 175, 176]	[78, 79, 92, 105, 107, 117, 131, 139, 152, 175, 179, 180]
Non-Linear	[8, 19, 25, 29, 39, 42, 110, 124, 139, 164, 182]	[28, 40, 41, 48, 124, 164, 184]	[19, 29, 48, 164]	[139]
Continuous Time	[8, 19, 29, 100, 103, 106, 109, 124, 139, 164, 182]	[28, 48, 67, 79, 92, 105, 107, 109, 117, 122, 124, 125, 127, 129–133, 164, 179, 180, 184]	[19, 29, 48, 79, 105, 107, 117, 125, 126, 130, 131, 164]	[79, 92, 105, 107, 117, 131, 179, 180]
Discrete Time	[25, 39, 42, 62, 64, 106, 110, 114, 152, 176, 183]	[40, 41, 78, 93, 101, 104, 105, 107, 131, 175]	[78, 105, 107, 131, 175, 176]	[78, 152, 175]
First Order	[8, 19, 25, 29, 39, 42, 62, 64, 100, 103, 106, 109, 110, 114, 124, 139, 152, 164, 176, 182]	[28, 40, 41, 48, 67, 92, 93, 105, 107, 124, 129–131, 164, 175],	[19, 29, 48, 105, 107, 130, 131, 164, 175, 176],	[92, 93, 105, 107, 131, 139, 152, 175]
Higher Order	[127, 183]	[78, 79, 117, 122, 125, 132, 133, 179, 180, 184]	[78, 79, 117, 125, 126]	[78, 79, 117, 179, 180]

Table 2.1: Classification of the considered literature for the consensus problem

2.2 Anomalies Detection

The *Target Of Interest* (TOI) identification task can intuitively be framed within the more generic problem of '*anomaly detection*', which is a relevant research field extensively analysed over the last century. In its most general definition, an anomaly is a single data point or a points pattern that does not comply with a definition of 'regularity' either provided by experts in the field or inferred from a set of data representing the available information. In the latter case, which is the most considered by current researchers because of the large data-availability provided by modern computer systems, any pattern that stands out with respect to the other data points within the considered data-set is considered as an anomaly. Numerous approaches involving different techniques, usually based on learning processes, have been used for discriminating anomalies from regular data points: classification of patterns by means of *Neural Networks* (NNs) ([82, 154]), *Bayesian Networks* (BNs) ([66, 170]), *Support Vector Machines* (SVMs) ([120]) or *rule-based systems* ([45]), data clustering ([51]), distance or density analysis respect to the *k-th nearest neighbour* ([38, 74]), statistical approaches based on parametric models (*Gaussian\regression models*, [5, 156]) or *Kernel Functions* ([96]), information-theoretic techniques based on *entropy* ([95]) or *Kolmogorov complexity* ([162]), spectral analysis performed, e.g. , by means of *Principal Components Analysis* (PCA) ([35]) or *wavelet transform* ([154]). Anomaly detection is an essential operation for a huge variety of applications: telephonic and credit card fraud detection ([21, 44, 45]), fault detection and diagnosis for mechanic machinery or electronic sensors ([7, 141]), road accidents and relative precursors detection ([27, 96, 154]), vehicle trajectories monitoring ([51, 66]), image processing ([35, 147]), medical diagnostic ([147, 170]), intrusion and malicious activity detection in networked computer systems ([74, 95, 162]), etc. .

Lokman et al. in [82] propose a solution for anomalies detection and target recognition based on *HyperSpectral Imaging* (HSI) and *Multi-Layered Neural Networks* (ML-NNs). With this approach, the individual pixels from spectral imagery data are submitted to a properly trained NN capable of recognising background elements from anomalies, signalling the latter case with a '1' output. If an anomaly is detected, the pixel is fed to a second NN, which is able to classify the anomalous pixel as a specific target type within a predefined set.

Kruegel et al. in [66] face the anomaly detection issue considering an IDS applied to system calls within a Unix-like *Operating System* (OS). In this work, the detection task is implemented by means of a Bayesian network modelling the relations between ordinary/malicious requests and a predefined set of features associated with the structure and content of the system calls. The proposed BN further accounts for correlation between features, confidence levels about these and additional information provided by users or complementary detection systems. The outputs of the individual feature models are likelihood values in the [0,1] range, which are then discretized into semantic values (‘Normal’, ‘Uncommon’, ‘Irregular’,‘Suspicious’ and ‘Very Suspicious’) by means of arbitrary thresholds and finally leveraged for inferring the final classification for the considered system call. A similar problem had been previously tackled in [50, 166] by Forrester et al., with the difference that in these works anomalous behaviours were detected by considering the sequence of observed system calls instead of the individual request characteristics.

Intrusion detection, intended as the attempt to maliciously access a computer system, is also considered in [74] by Liao et al. . The authors show that an anomalous program execution can be identified by considering the system calls it generates as words of a text document and by applying a set of well-established text analysis tools. More specifically, each process is represented as function of the frequencies of the requested system calls and then a k-NN (*Nearest Neighbour* of order k) classifier is applied. This identifies the k training processes with smallest distance respect to the test entry and then classifies this as ‘regular’ only if the aggregated distance is below a given threshold.

In [120], the task of detecting anomalies and outliers is cast from the authors into a one-class classification problem, which is solved by means of both an SVM and a boosting algorithm. On the basis of the well-known equivalence of the two approaches, the authors further provide a general recipe for translating SVMs into boosting-like algorithms.

Fawcett et al. in [45] consider the problem of telephone fraud detection proposing a rule-based system relying on learning tools. In this work user-related rules indicating possible frauds are extracted from a dataset of labelled telephonic records. These rules are then leveraged for selecting the inputs to be provided to analysis tools indicated as ‘monitors’. These components of the detection framework are in charge of measuring the deviation of observed call patterns from what it is considered regular for the specific user, and are trained, or ‘profiled’, on the basis of known-legit calls entries. Finally, a classifier for the aggregated monitors outputs is defined through a training approach.

Data clustering as learning tool for achieving anomaly detection is considered in [51] by Fu et al.. In this work a dataset of actual vehicle trajectories over a specific road section is clustered by means of a hierarchic two-layers clustering technique. Real-time anomalies detection is then achieved by associating newly observed trajectories to one of the previously defined clusters and by checking some conditions (e.g., the distance between the current datapoint and the associated cluster-centre or the fact that the trajectory is assigned to different clusters over time).

Outliers rejection as specific case of anomalies detection is considered in [156] by Torr et al.. Given an orthogonal regression problem, the authors propose a technique for identifying a regression hyperplane not influenced by eventual outliers within the data points. This is achieved by leveraging the ‘eigenvector perturbation theory’ for defining metrics that allow measuring the influence of individual data points on the final regression solution. Then, an iterative algorithm is applied that at each step removes the most influencing input until a termination criterion is met.

Nychis et al. tackle the problem of anomalies detection with reference to computer network traffic in [95]. Their work provides an entropy-based interpretation of the distributions associated with relevant features of the monitored traffic (source and destination ports, addresses, packet sizes, communication in- and out-degree for a node etc.). The authors provide useful insights about how variations in the entropy time-series for the features suggest the presence of anomalous traffic, also considering the meaning of features correlation.

A relevant topic related to anomaly detection is that of discriminating anomalies from novelties in the observed behaviours or, more generically, in the inputs provided to a trained classification or monitoring tool. With reference to this problem, in [15], Bishop describes a statistical approach for identifying novel values provided as input to a neural network: in so doing, the network can provide a confidence value for the output associated with a novel input. This allows discarding unreliable outputs and thus, somehow, validating the neural network behaviour.

A comprehensive survey on the anomaly detection problem is provided by Chandola et al. in [22], while Bolton et al. in [17] and Patcha et al. in [113] present surveys focused on fraud detection and intrusion detection respectively. A further survey mainly focused on intrusion detection as specific case of anomaly detection has been proposed by Patcha et al. in [113]. Finally, Markou et al. in [86, 87] present a review on novelty detection techniques.

2.3 Target Monitoring

In literature, a well-established approach to threat detection is based on the concept of 'target classification', and thus on the assumption that some given types of vehicles represent a threat, while others do not. Some of the most widely adopted classification techniques are based on target feature extraction from imagery, like shapes ([72, 81]) or movements ([58]), measurable physical target characteristics, e.g. radar cross section ([23]) or acoustic data ([165]), kinematic aspects, like speed and acceleration ([3, 137]), or analysis of the exhibited trajectories ([185]).

A slightly different problem is that of assessing the threat level associated with a target regardless of its 'physical' typology, solely due to its behaviour. Some useful references and guidelines on this topic are provided by Liebhaber et al. in [75], where the cognitive process involved in the threat assessment task is outlined with reference to the aeronautic field. Examples of automatic assessment systems are provided in [83], where the threat assessment problem is considered with reference to a battleground scenario, in [90], where expert systems are used for assessing the state (e.g., fatigue condition) of a human being, and in [99], where the threat assessment is implemented by detecting a given sequence of manoeuvres exhibited by a road-constrained ground vehicle. Rigolli et al. in [136] present a multi-lane traffic simulator where the driving agents are modelled by taking into account the subjective interpretation of human drivers about the road/-traffic conditions. With this approach, a set of 7 parameters is used for translating the “Objective World” (where, e.g., the car in front is 5 meters distant) into the “Subjective World” (where the same vehicle is or is not, e.g., “close enough to start an overtaking manoeuvre”). Different sets of values for these parameters allow simulating different driving behaviours, thus providing a variegate representation of human drivers. The proposed simulation model has not been validated by means of any real-world dataset, yet it is able to reproduce phenomena commonly observed in real traffic conditions, e.g. ‘compression waves’ in vehicles density. No monitoring actions are performed in this work, but the simulator appears to be a useful tool on top of which assessment procedures can be built, e.g., clustering techniques based on the behavioural parameters.

The problem of safety-improvement for ground vehicles by means of online-monitoring has been firstly addressed by taking into consideration the emotional and physiological conditions of the driver. Malta et al. in [85] propose a system for inferring the driver’s frustration state on the basis of a set of physiological, behavioural and environmental information. Actual driving tests have been performed during which gas/brake actuation, electrodermal activity, speech recognition errors and video footage focused on facial expressions have been recorded. Lastly, ground truth information about its frustration state has been directly provided at the end of each session by each of the 30 drivers involved in the experiment. The data collected has allowed the authors to define and train a Bayesian network capable of recognising up to 80% of the frustration condition states for the drivers. In [140] Sandberg et al. address the problem of detecting sleepiness in car drivers. As first thing, a variety of sleepiness indicators from literature has been considered, and suitable parameters for each indicator have been retrieved by means of a stochastic optimization method (genetic algorithms). The data leveraged for this process have been obtained from a study that has involved 12 test subjects at the moving-base driving simulator at the Swedish National Road and Transportation Research Institute. The data were derived from 12 1-h driving sessions for each test subject, with varying degrees of sleepiness. The resulting indicators have then been aggregated into *Sleepiness Detection Systems* (SDSs), implemented as linear combination of indicators or by means of neural networks, and combined with a mathematical model of sleepiness. Such approach has lead to a resulting performance score up to 0.83%, demonstrating the effectiveness of the proposed approach. A study similar to the previous one has been performed by Rigas et al. in [135]. Here, a real-time methodology for the detection of stress events while driving is presented. This task is performed by analysing physiological signals, i.e. electrocardiogram, electrodermal activity and respiration, as well as information about the user driving behaviour, e.g. average speed, average magnitude of heading change, average throttle. Two Bayesian networks are then defined allowing the detection of stress conditions: the fist BN only relies on physiological features, while the second considers also behavioural features. The dataset used for the BNs training consists of information recorded in real driving conditions, involving 13 subjects of mixed age and gender. Similarly to the works in [85, 140] the correct detection ratio is about 80% for the BN based on solely physiological information, while it reaches 96% for the more complex solution accounting for driving events. ([169]). Such solutions, however, mostly rely on intrusive sensing devices that need to be applied directly to the driver (e.g. heart-rate monitor) or on the vehicle (e.g., cameras or kinematic/dynamics information sensors), thus being hardly suitable for implementation on large scale. The use of in-car data recorders is the basis of the experimental study conducted by Wouters et al. in [171]. The authors here analyse the effects on road accident numbers of recording driving actions and providing behavioural feedback to the drivers. More specifically, drivers from heterogeneous vehicle fleets have been informed about the recording device equipping their vehicle and have been then provided with a feedback about their driving behaviour. The road accident rates for the recorder-equipped vehicles have been compared with those relative to standard vehicles during the same period and accident rates characterising previous and successive periods respect to the actual test, so that inherent trends can be isolated from the test effects. Regardless of the assumptions and approximations introduced in the design of the statistic experiment, this test, overall involving 1100 vehicles over 3100 vehicle-years, provides really interesting results, indicating that the use of data-recorder could lead to a reduction in the accidents rate up to 30%. A relevant aspect that is not addressed by the experiment, however, is if the accidents reduction should be ascribed to the sole presence of the recording device (influencing the driver to adopt more conscientious behaviours) or to the actual feedback provided from the system. In the two cases, indeed, different actions would be required to maintain a decreasing accidents ratio over time.

A different and probably more feasible approach to road-safety improvement exploits roadside infrastructure[2], also denoted as *Freeway Traffic Monitoring Systems* (FTMS): these are equipped with, e.g., camera systems, loop detectors and other sensing devices that allow quantifying the traffic flow at discrete positions along the road ([27]). With reference to this approach, Oh et al. in [96] propose a real-time monitoring system that collects traffic flow information and effectively manages to detect traffic conditions suggesting a likely imminent accident; this allows authorities to warn drivers by means of dedicated road signals or other communication devices. This result is achieved by means of historic data about the accidents that have happened in that specific road section (which is used for inferring accident likelihood functions) and a statistic decision

[2]Such infrastructures have been already deployed in many cities around the world and provide information on the overall traffic flow instead of the single vehicle. This solution addresses the sensing issue that affects the 'physiological' approach, even if more coarse-grain information is provided.

policy. The idea behind this work is to adopt a learning technique so that incident precursors, which are anomalies in vehicular traffic possibly leading to an accident event and thus to traffic disruption, can be identified. A similar approach is taken in [154] by Thajchayapong et al., who present a monitoring framework capable of detecting accident precursors by analysing microscopic traffic variables. These quantities (e.g. relative speed and distance between vehicles) are obtained by means of roadside infrastructures or calculated in a distributed manner by local devices. The classification of the traffic conditions as 'regular', 'transient anomaly' or 'incident precursor' is finally obtained by feeding a spectral representation of the traffic variables to a multi-layer neural network and then by applying a decision policy based on a varying threshold. In [65], Kamijo et al. present a system for the monitoring of traffic flow at intersections. The proposed approach is based on video information and considers the individual vehicles behaviour rather than global flow analysis . For this purpose, a tracking technique based on spatio-temporal *Markov Random Fields* (MRFs) is firstly developed, which has proved to be effective up to 95% of the cases for occluded subjects and 99% for vehicles not causing occlusion. This technique has been used then for implementing an event-recognition system, in charge of detecting anomalous events at at road intersection, e.g. bumping accidents, passing and jamming. This system consists of a *Hidden Markov Model* (HMM) trained by means of the Baum-Welch Algorithm on the basis of video observations sequences. The effectiveness of this method, however, has been tested only on a small number of cases, due to the limitation of accident events during our observation period.

A different approach to the monitoring problem has been taken in other works, where the focus has been placed on the behaviour of a single vehicle rather than on general traffic conditions. Rodriguez Gonzalez et al. in [138] propose a technique for assessing whether the driving-behaviour of a vehicle can be considered 'aggressive', and thus to some extent dangerous for the traffic flow, or not. This is done by recording kinematic data from a vehicle performing a limited number of 'loops' on a road segment when being driven by some different drivers, both in an 'aggressive' and 'regular' manner. The data collected is then used for producing *Gaussian Mixture Model* (GMM) distributions for predefined 'driving features' and for empirically verifying the initial hypothesis that the effect of 'aggressiveness' can be modelled as a linear filter on the 'regular' variables distributions. The risk associated with a single vehicle behaviour is analysed also in [172] by Wu et al., who define a hierarchic assessment framework based on HMMs and fuzzy logic. Three Markov models for three assumed classes of behaviours (lateral, frontal and 'car following') are defined and trained by means of experimental kinematic data: in so doing the Markov models become capable of recognising a finite set of driving events and their 'degrees'. The outputs from the Markov models are provided to a *Fuzzy Inference System* (FIS) that calculates the membership functions for different linguistic variables and applies then one of two hierarchic rule-sets. This finally allows establishing if the risk level associated with a set of kinematic measurements should be considered low, medium or high. In [69], Lane et al. address the problem of threat assessment in the maritime environment on the basis of alarming behavioural aspects. This is achieved by means of a BN that assesses the threat likelihood of the behaviours exhibited by the monitored targets, without however proposing a decision policy that would enable to actually differentiate alleged threats from regular targets. Benavoli et al. in [10] present an approach to threat assessment based on *evidential networks*, an inference tool defined within the *theory of the evidence*; the limit for this technique is, however, to consider discrete values for the features characterising the target. The description of a generic framework for threat assessment accounting for multi-modal data is provided in [9] by Beaver et al.. The main focus of this work is the introduction of a taxonomy for the threat condition and a qualitative description of an approach for implementing the 'decision-level' of the information fusion process; an actual definition of the techniques to adopt is however not provided.

Chapter 3

Consensus Protocols Theory Background

3.1 Graph Theory

3.1.1 Communication Network Representation

Graphs are probably the simplest and most common models adopted in literature for representing the communication network connecting a team of agents. A graph can be considered either directed or undirected, depending on the capabilities of the agents' communication devices or on some design specification. A directed graph topology could be adopted, for example, in the case where some agents have shorter communication range than other (e.g., agent v_i can send messages to agent v_j but not vice-versa) or if, for reducing the number of exchanged data, a minimal number of communication links is established. In the most general case, weighted edges could be considered, accounting, for example, for a greater reliability of information broadcasted by certain nodes. An example of a generic weighted digraph is shown in Fig.3.1, which accounts also for *self-loops*. This type of edge, connecting a node to itself, will, however, be neglected in the remainder of the thesis (only *simple* graphs will be considered).

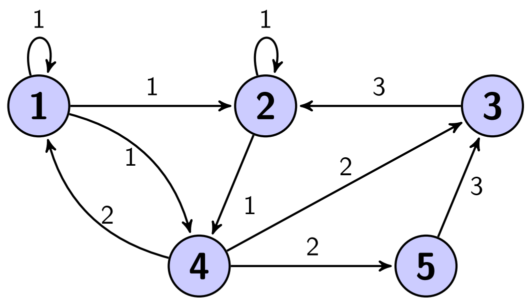


FIGURE 3.1: Example of weighted directed graph

3.1.2 Definitions

A *Graph* is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, ..v_n\}$ is a finite nonempty set of *nodes* (or *vertices*) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of *edges* (or *arcs*) connecting the graph nodes. Intuitively nodes correspond to network agents, while edges represent active communication links. The pair $(i, j) \in \mathcal{E}$ if exists an edge linking node i and j. Self-edges are not allowed, so pairs $(i, i)\ \forall i$ are not valid elements of $\mathcal{E}$. The *Graph order* n corresponds to the number of nodes in the graph, i.e. $n = |\mathcal{V}|$.

A *Directed Graph*, or *Digraph*, is a graph where edges are ordered pairs, i.e $(i, j) \in \mathcal{E}$ indicates that node j can be reached from node i, but not necessarily vice versa: in such case, i is referred as the *parent node* and j as the *child node*. An *Undirected Graph* is a graph where edges have no orientation. The edge (i, j) is the same of (j, i), i.e., they are not ordered pairs. This type of graph can be seen as a special case of a digraph, where $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$

Given a (di)graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a *(Directed) Path* p between nodes v_i and v_j is a sequence of edges of the form $\Big(\left(v_{p1}, v_{p2}\right), \left(v_{p2}, v_{p3}\right), \ldots, \left(v_{p_{k-1}}, v_{p_k}\right)\Big)$, where $i = p_1$, $j = p_k$ and $v_{p_q} \in \mathcal{V}\ \ \forall q$. A (di)graph is *(Strongly) Connected* if there is a (directed) path between every pair of distinct nodes.

The *Neighbor Set* of node v in graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as $\mathcal{N}_v = \big\{j \in \mathcal{V} : (j, v) \in \mathcal{E}\big\}$.

A *Weighted (Directed) Graph* is a triple $(\mathcal{V}, \mathcal{E}, w)$ where $(\mathcal{V}, \mathcal{E})$ is a (di)graph and $w : \mathcal{E} \to \mathbb{R}_+$ is a map associating each arc (i, j) to a strictly positive weight w_{il}. For undirected graphs, edges (i, j) and (j, i) are allowed to assume different weights.The weights map w is usually expressed by the adjacency matrix $A = \left[a_{ij}\right]$, i.e $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$. It is thus clear that a generic graph can be defined as function of the only adjacency matrix, i.e $\mathcal{G} = \Gamma(A)$, since it indicates the number of nodes, presence of connection between the latter and weights associated with the arcs.

All the further relevant definitions about graphs that are adopted in this work are provided in Appendix B.

3.2 Algebraic Tools

This section provides the basic algebraic tools necessary for analysing consensus protocols over graphs. All the required definitions that are not provided in this section can be found in In Appendix A.

3.2.1 Algebraic Representation of the Communication Network

A graph $\mathcal{G}$ can be completely described by means of a matrix $A \in \mathbb{R}^{n \times n}$, usually referred as the *Adjacency matrix*. More specifically, $\mathcal{A}(\mathcal{G}) = A = [a_{ij}]$, where $n = |\mathcal{V}|$, and

$$a_{ij} = \begin{cases} w_{ji}, & (j,i) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}$$

where $w_{ji} \neq 0$ is the weight associated with the directed edge connecting node j to node i. If no weights are defined $a_{ij} \in \{0,1\}\ \forall i,j$. This general definition can be specialised for non-weighted graph assuming $w_{ij} = 1, \quad \forall i,j \in \mathcal{V}$, and for non-directed graph imposing $a_{ij} = a_{ji}, \quad \forall i,j \in \mathcal{V}$. Thanks to the assumption about self-loops, it is possible to state that $a_{ii} = 0, \quad \forall i \in \mathcal{V}$. For an undirected graph, A is symmetric by definition.

The *Neighbour set* of agent i is the set of agents that are supposed to communicate with agent i: in the general case, this is a proper subset of $\mathcal{V}$, and can be defined on the basis of the adjacency matrix, as

$$\mathcal{N}_i = \{j \in \mathcal{V} : a_{ij} \neq 0\}.$$

3.2.2 Laplacian Matrix

A relevant matrix for the study of the dynamic of linear consensus protocols is the *Laplacian Matrix* $\mathcal{L}(\mathcal{G}) = L = [l_{ij}]$, with

$$l_{ij} = \begin{cases} -a_{ij}, & i \neq j \\ \sum_{k \neq i} a_{ik}, & i = j \end{cases}$$

For an undirected graph, L is symmetric and positive semidefinite. Considering indeed the *Laplacian Potential*

$$\Phi_G(x) = \frac{1}{2} \sum_{i,j=1}^{n} a_{ij}(x_j - x_i)^2 \tag{3.1}$$

in the case of symmetric Laplacian, i.e. obtained from a symmetric adjacency matrix, it holds ([139]):

$$x^T L x \geq \Phi_G(x) \geq 0.$$

By definition, in both the cases of directed or undirected graph, any row-sum of the Laplacian matrix is zero: L has thus a zero-valued eigenvalue associated with the right eigenvector

$$w_r = \mathbf{1} = (1, 1, \ldots, 1)^T.$$

Under these conditions, the 'Rank-Nullity' theorem guarantees that $\text{rank}(L) \leq n-1$. Furthermore, L is diagonally dominant and has non-negative diagonal entries. On the basis of these proprieties for the Laplacian matrix, it can be proved ([59]) that

- For an undirected graph, all of the non-zero eigenvalues of L are positive;
- For a directed graph, all of the non-zero eigenvalues of L have positive real parts.

Given an undirected graph, the i^{th} smallest eigenvalue of L is denoted as $\lambda_i(L)$ with

$$\lambda_n(L) \geq \ldots \geq \lambda_2(L) \geq \lambda_1(L) = 0$$

$\lambda_2(L)$ is the *algebraic connectivity* of the graph $\mathcal{G}$, which quantifies the convergence rate of a relevant class of consensus algorithms.

Given the previous definitions, the following theorems hold:

Theorem 3.1. (Laplacian Rank for Directed graph) *Let $\mathcal{G}$ be a weighted digraph with Laplacian L. If $\mathcal{G}$ is strongly connected, then* $\text{rank}(L) = n-1$.

Proof. Proof is given in [107], Appendix A □

Theorem 3.2. (Laplacian Rank for Undirected Graph) *Let $\mathcal{G}$ be a weighted graph with Laplacian L.* $\text{rank}(L) = n-1$ *if and only if $\mathcal{G}$ is connected.*

Proof is provided in [14].

Theorems 3.1 and 3.2 give conditions under which zero is a simple eigenvalue of the Laplacian matrices associated with directed and undirected[1] graphs. For digraphs, algebraic multiplicity of the zero-valued eigenvalue can be checked also through the following proposition:

Proposition 3.3. *Let G denote a (loopless) digraph. Then, zero is an eigenvalue of algebraic multiplicity one for the Laplacian L if and only if G has a rooted directed spanning tree.*

Proof is given in [67].

[1] To be noticed that, for an undirected graph, having a simple zero-valued eigenvalue implies having a positive algebraic connectivity.

3.2.3 Perron Matrix

The *Perron Matrix* associated with a graph enables the analysis of discrete-time consensus algorithms. This matrix is defined as:

$$\boldsymbol{P} = \boldsymbol{I} - \epsilon \boldsymbol{L}$$

where ϵ is denoted as 'Perron matrix parameter', while $\boldsymbol{L}$ and $\boldsymbol{I}$ are respectively the Laplacian and identity matrix. The Perron matrix just defined can be thought of as a special case of the general form

$$\boldsymbol{P} = e^{-\epsilon \boldsymbol{L}} = \sum_{k=1}^{\infty} \frac{(-\epsilon \boldsymbol{L})^k}{k!},$$

where the series is limited to the first-order term since the algorithm uses just the information provided by the first-order neighbours [2]. Some relevant characteristics of the Perron matrix are stated in the following theorem:

Theorem 3.4. *Let $\mathcal{G}$ be a digraph with n nodes and maximum in-degree $\Delta = \max_i \deg_{in}(v_i) = \max_i \sum_{j \neq i}(a_{ij})$. Then, the Perron matrix $\boldsymbol{P}$ with parameter $\epsilon \in (0, 1/\Delta]$ satisfies the following properties:*

i) *$\boldsymbol{P}$ is a row stochastic nonnegative matrix with a trivial eigenvalue of 1;*

ii) *All eigenvalues of P are in a unit circle;*

iii) *If $\mathcal{G}$ is a balanced graph, then $\boldsymbol{P}$ is a doubly stochastic matrix;*

iv) *If $\mathcal{G}$ is strongly connected and $0 < \epsilon < 1/\Delta$, then $\boldsymbol{P}$ is a primitive matrix.*

The proof is given in ([105])

Remark 3.5. From part i) of the theorem it comes directly that an eigenvector associated with the trivial eigenvalue 1 is $\mathbf{1} = (1, \ldots, 1)^T$

Remark 3.6. If the communication graph is strongly connected, $\boldsymbol{P}$ results to be an *irreducible* and *primitive* matrix. From Perron-Frobenius theorem it is possible to say that the *Perron-Frobenius eigenvalue*, i.e. the eigenvalue corresponding with the spectral radius, is simple. Since from Theorem (3.4) the spectral radius of the Perron matrix has value 1, these results can be thought of as the corresponding condition of having a simple zero-valued eigenvalue in the Laplacian matrix.

Remark 3.7. The condition $\epsilon < 1/\Delta$ in part iv) is necessary. If an incorrect step-size, i.e. $\epsilon = 1/\Delta$, is used, then $\boldsymbol{P}$ is no longer a primitive matrix, making multiple eigenvalues with modulus 1 possible.

[2] m^{th}-order neighbours are indicated from the m^{th} power of the adjacency matrix, A^m.

The following result will be useful for the convergence analysis of a widely-applied discrete-time consensus algorithm:

Theorem 3.8. (Perron-Frobenius) *Let $\boldsymbol{P}$ be a primitive nonnegative matrix with w and v, respectively, left and right eigenvectors associated with the trivial eigenvalue* 1*, satisfying $\boldsymbol{P}\boldsymbol{v} = \boldsymbol{v}$, $\boldsymbol{w}^T\boldsymbol{P} = \boldsymbol{w}$ and $\boldsymbol{v}^T\boldsymbol{w} = 1$. Then:*

$$\lim_{k\to\infty} \boldsymbol{P}^k = \boldsymbol{v}\boldsymbol{w}^T$$

The proof is provided in ([59])

3.3 First Order Linear Consensus Algorithms

In this section linear consensus protocols applied to scalar information states are considered. The basic idea behind a consensus algorithm is to achieve similar values for the information states of the agents in a team. The update law for the states can be modelled as a differential equation when the communication network allows communicating in a continuous manner or if the communication bandwidth is sufficiently large. If, otherwise, the information exchange is implemented at given time steps utilising discrete data packets the consensus protocol is defined as a difference equation. Recall that, in linear consensus protocols, the state update for a given agent is obtained as a linear combination of its own state and the information states received from the agents in its neighbourhood.

3.3.1 Continuous-Time Protocols

The most common continuous-time consensus algorithm is given by

$$\dot{x}_i(t) = -\sum_{j=1}^{n} a_{ij}\left(x_i(t) - x_j(t)\right) \tag{3.2}$$

where n is the number of agents in the network, a_{ij} is the (i, j) entry of the adjacency matrix and $x_i(t)$ is the information state of the i^{th} vehicle at a time t. Given the definition previously provided for the elements of the adjacency matrix, the sum $\sum_{j=1}^{n}$ could also be written as $\sum_{j\in\mathcal{N}_i}$. On the basis of the definitions introduced in Section 3.2.2, Eqn. 3.2 can be written as:

$$\dot{\boldsymbol{x}}(t) = -\boldsymbol{L}(t)\boldsymbol{x}(t), \tag{3.3}$$

where $\boldsymbol{x} = [x_1, \ldots, x_n]$, and $\boldsymbol{L}(t) = [l_{ij}] \in \mathbb{R}^{n\times n}$ is the Laplacian of the underlying communication graph. Regardless of the notation, agreement in a team of vehicles is achieved through the adoption of a consensus protocol if, for any initial collective state $\mathbf{x}(0)$:

$$\lim_{t\to\infty} |x_i(t) - x_j(t)| = 0, \quad \forall i, j = 1, \ldots, n.$$

This condition[3] is equivalent to "proving that the agreement space characterised by $\boldsymbol{x} = \alpha\mathbf{1},\ \alpha \in \mathbb{R}$ is an asymptotically stable equilibrium of system" (3.2), ([105]). Assuming an invariant communication topology, the Laplacian matrix is a constant, i.e. $\boldsymbol{L}(t) = \boldsymbol{L}$, and thus, with reference to Eqn. 3.3 the behaviour of the consensus algorithm for the whole team of agents is represented by a *Linear Time-Invariant* (LTI) system, whose dynamic matrix is $-\boldsymbol{L}$. The convergence of the proposed consensus algorithm is thus completely determined by the location of the eigenvalues of the Laplacian matrix. As previously noticed, all non-zero eigenvalues of matrix $-\boldsymbol{L}$ have negative real part. Furthermore, vector $\mathbf{1}$ is an eigenvector of $\boldsymbol{L}$ and thus $\mathrm{Span}(\mathbf{1}) \in \mathrm{Ker}(\boldsymbol{L})$. If 0 is a simple eigenvalue of $\boldsymbol{L}$ the following statements hold:

- The system described by Eqn. 3.3 is *Marginally stable*, since the real part of any pole in the system transfer function is non-positive, and all poles with zero real value are simple roots;
- Given a scalar value $\bar{x}$, $\bar{x}\mathbf{1}$ is an equilibrium of the system (3.3), and thus $\boldsymbol{x}(t) \to \bar{x}\mathbf{1}$ when $t \to \infty$, meeting the definition of convergence to a consensus.

The convergence analysis for the considered consensus protocol focuses therefore on proving that zero is a simple eigenvalue of $\boldsymbol{L}$. Assuming indeed that the algebraic multiplicity of the zero-valued eigenvalue is greater than one, if the associated geometric multiplicity is one, the marginal stability is not achieved, while, otherwise, if the geometric multiplicity is greater than one the dynamical system accepts equilibrium states that are not contained in $\mathrm{Span}(\mathbf{1})$, and thus the convergence to a consensus state is not guaranteed. Necessary and sufficient conditions for the Laplacian of a graph for having a simple zero-valued eigenvalue, with reference to graph topology, are provided in Section 3.2.2 for both the directed and undirected graph cases. It is worth to remark that in presence of a single zero-valued eigenvalue, even if the dynamic system associated with the consensus protocol is classified as marginally stable, the states of the nodes asymptotically converge to the consensus value $\bar{x}$. This means that the equilibrium $\bar{x}\mathbf{1}$ is globally exponentially stable.

Assuming the convergence of the consensus protocol, some considerations about the final equilibrium achieved, i.e. the scalar value $\bar{x}$, are provided in the following. In the case of undirected graph, since $a_{ij} = a_{ji} \quad \forall i, j$, it can be noticed that the sum of the states of all the nodes is an invariant quantity, i.e. $\sum_i \dot{x}_i = 0$. Applying this condition at times

[3]The condition makes reference to the cases where the information state is a scalar value. It can be however generalised to the case of vectorial information state by adopting an appropriate norm instead of the 'absolute value' operator.

$t = 0$ and $t = \infty$ the following equation is obtained:

$$\bar{x} = \frac{1}{n}\sum_i x_i(0).$$

This result means that, for an undirected graph, the application of the consensus protocol in (3.2) leads, if a consensus is asymptotically reached, to a collective decision that is equal to the average of the initial state of all nodes. A consensus algorithm with this specific invariance property is called an *average-consensus* and can be thought of as a special case of f-consensus where the function $\mathrm{f}(\boldsymbol{x})$ of the initial states is the arithmetic mean.

For the convergence analysis in the case of a directed graph, the following theorem is proposed in [105]:

Theorem 3.9. (Weighted Average-Consensus) *Consider a network of n agents represented by a directed graph $\mathcal{G}$: suppose $\mathcal{G}$ is strongly connected, and let $\boldsymbol{L}$ be the Laplacian of $\mathcal{G}$, with a left eigenvector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$ associated with the zero-value eigenvalue, thus satisfying $\boldsymbol{\gamma}^T\boldsymbol{L} = 0$. Then, applying the consensus algorithm in (3.2) the following statements hold:*

i) *A consensus is asymptotically reached for all initial states;*

ii) *The algorithm solves the f-consensus problem with the linear function $f(\boldsymbol{z}) = (\boldsymbol{\gamma}^T\boldsymbol{z})/(\boldsymbol{\gamma}\mathbf{1})$, i.e. the group decision is*

$$\bar{x} = \sum_i w_i x_i(0)$$

with $\sum_i w_i = 1$

iii) *if the digraph is weight-balanced, an average-consensus is asymptotically reached and*

$$\bar{x} = \frac{1}{n}\sum_i x_i(0)$$

The theorem proof is given in [105].

Remark 3.10. The condition of strong connectivity required in Theorem 3.9 could be substituted with the weaker condition of presence of a rooted spanning tree (preposition 3.3)

Remark 3.11. Convergence for the consensus algorithm applied to an undirected graph can be thought of as a special case of the directed case, since an undirected graph is balanced by definition: part iii) of Theorem 3.9 provides indeed the same result previously obtained for an undirected graph.

Theorem 3.9 shows that, if the consensus is achieved, the equilibrium state is a weighted average of the initial information states in the network. However, some of the components of $\boldsymbol{\gamma}$ may be zero, implying that the information states of some of the vehicles do not contribute to the equilibrium. For strongly connected graphs it can be proved that $\gamma_i \neq 0, \quad \forall i$ and thus all of the initial information states contribute to the consensus equilibrium. On the other hand, when the directed communication topology contains a rooted directed spanning tree but is not strongly connected, the consensus equilibrium is the weighted average of the initial conditions of just those vehicles that have a directed path to all of the other agents. The presence of a rooted directed spanning tree is a less stringent condition than requiring a strongly connected graph. However, as shown above, in this case, the consensus equilibrium is less meaningful since some initial conditions are entirely neglected during the agreement process.

Convergence proof for the consensus algorithm in (3.2) has been provided under topological conditions for the underlying communication graph, both in the cases of directed and undirected graph. Note that the considered protocol does not allow to define a desired final state for the information vector. In general, it is possible to guarantee only that the agreement value is a convex combination of the initial information states.

3.3.2 Discrete-Time Protocols

Various versions of a linear discrete-time consensus protocol have been proposed in literature: e.g., [62, 93, 105, 130, 131]. In the following, the approach described by Olfati et al. in [105] is considered since it is more similar in the structure and notation to the previously described continuous-time algorithm. The protocol update-law is:

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in \mathcal{N}_i} a_{ij} \left(x_j(k) - x_i(k) \right) \tag{3.4}$$

where

$$\begin{aligned} x_i &: \text{state of } i^{\text{th}} \text{ agent} \\ \epsilon &: \text{step size} \quad (> 0) \\ \mathcal{N}_i &: \text{Neighbour set of agent } i \\ a_{ij} &: (i, j) \text{ entry of the adjacency matrix} \end{aligned}$$

Only the case of digraph will be considered in the following since an undirected graph can be considered as a symmetric, and thus balanced, directed graph. Algorithm (3.4) can be written in matrix form as:

$$\boldsymbol{x}(k+1) = \boldsymbol{P}\boldsymbol{x}(k) \tag{3.5}$$

where $\boldsymbol{P}$ is the Perron Matrix with parameter ϵ associated with the communication graph (see Section 3.2.3). From this formulation, as well as for the continuous-time case, it is clear that the discrete-time consensus algorithm in (3.4), applied to all the agents in the network, can be represented by an autonomous discrete LTI system, whose stability analysis relies only on the properties of its dynamic matrix $\boldsymbol{P}$. Convergence analysis for the proposed algorithm is provided by the following theorem ([105]):

Theorem 3.12. *Consider a network of agents with communication topology represented by the strongly connected digraph $\mathcal{G}$, and implementing the distributed consensus algorithm in (3.4). Assume $0 < \epsilon < 1/\Delta$, where Δ is the maximum in-degree of the nodes in the network. Then:*

- *A consensus is asymptotically reached for all initial states;*
- *The group decision value is $\bar{\boldsymbol{x}} = \sum_i w_i x_i(0)$ with $\sum_i w_i = 1$*
- *If the digraph is balanced (or $\boldsymbol{P}$ is doubly stochastic), an average-consensus is asymptotically reached and $\bar{\boldsymbol{x}} = \sum_i x_i(0)/n$.*

The proof for the theorem is provided in [105] and here reported for convenience.

Proof. The marginal stability of the dynamic system in (3.5) can be proved by applying Theorem 3.4 to a strongly connected graph: all the eigenvalues of $\boldsymbol{P}$ result to be within the stability region for a discrete-time linear system (unitary circle), and only one eigenvalue is on the boundary of the region. Marginal stability is thus guaranteed. An equivalent result can be obtained by noting that $\boldsymbol{x}(k) = \boldsymbol{P}^k\boldsymbol{x}(0)$, and thus the system (3.5) results stable if $\boldsymbol{P}^k$ admits a limit: the existence of such limit is guaranteed by Theorem 3.8. The convergence of the consensus protocol can be proved considering that, under the given hypotheses, $\boldsymbol{P}$ is an irreducible primitive matrix (Theorem 3.4 part iv)), and thus from Theorem 3.8:

$$\lim_{k\to\infty} \boldsymbol{x}(k) = \boldsymbol{v}(\boldsymbol{w}^T\boldsymbol{x}(0)).$$

Remark 3.5 allows affirming $\boldsymbol{v} = \boldsymbol{1}$, leading to:

$$\bar{x} = \lim_{k\to\infty} \boldsymbol{x}_i(k) = \sum_j \boldsymbol{w}_j^T\boldsymbol{x}_j(0)), \quad \forall i,$$

which is the agreement value produced by the consensus algorithm. Further, $\sum_j \boldsymbol{w}_j = 1$ because $v^T w = 1$ (again, Theorem 3.8). For the last part of the theorem, if the graph is balanced, $\boldsymbol{P}$ is a column stochastic matrix with a left eigenvector of $\boldsymbol{w} = (1/n)\mathbf{1}$, since $w_i = w_j \forall (i,j)$ and $\sum_j \boldsymbol{w}_j = 1$. The group decision becomes equal to $\bar{x} = (1/n)\mathbf{1}^T x_i(0)$ and average-consensus is thus asymptotically reached $\square$

Remark 3.13. The result of the consensus algorithm is a weighted average of the initial states of the agents. The actual weights are defined from the elements that compose the left eigenvalue of matrix $\boldsymbol{P}$ associated with eigenvalue 1.

3.4 Belief Consensus Protocol

'Belief Consensus' has been introduced by Olfati et al. in [110]. This consensus approach is implemented by means of a nonlinear update rule that allows propagating local 'belief values' denoted as π and achieving a final agreement about this value. The algorithm is defined as follow:

$$\pi_i(k+1) = \pi_i(k)^{\beta_i} \prod_{j \in \mathcal{N}_i} \pi_j(t)^{\gamma}, \tag{3.6}$$

where i is the agent index, k is the consensus step, γ is the consensus protocol gain, $\mathcal{N}_i$ is agent i neighbourhood, Δ_i is the in-degree of node i and $\beta_i = 1 - \gamma\Delta_i$. By implementing this protocol, it is proven that the consensus states π_i will tend to a common value that complies with the Bayes rule for posterior probability:

$$\lim_{k\to\infty} \pi_i(k) = p\left(h_t | Z\right) = \alpha p\left(h_t\right) \prod_{j=1}^{n} p\left(z_j | h_t\right) \quad \forall i = 1, \dots, n \tag{3.7}$$

where h_t is the hypothesis "of interest", $Z = [z_1, z_2, \dots, z_n]$ is the set of measurements about the target and n the number of agents involved.

By defining the *likelihood of the belief* of agent i as $l_i = \log(\pi_i)$, the considered belief consensus corresponds to the discrete-time average-consensus algorithm described in Section 3.3.2 applied to an information vector composed of belief likelihood values. This result is proved by the following lemma.

Lemma 3.14. *The belief consensus protocol*

$$\pi_i(k+1) = \pi_i(k)^{\beta_i} \prod_{j \in \mathcal{N}_i} \pi_j(t)^{\gamma}, \tag{3.8}$$

is equivalent to the discrete-time linear consensus protocol

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in \mathcal{N}_i} a_{ij} \left(x_j(k) - x_i(k)\right) \tag{3.9}$$

under the following conditions:

- $a_{i,j} \in \{0,1\} \ \forall i, j$
- $\epsilon = \gamma$
- $\boldsymbol{X}(k) = \big[x_1(k), x_2(k), \ldots, x_n(k)\big]^T = \big[l_1(k), l_2(k), \ldots, l_n(k)\big]^T \ \ \forall k$

Proof. The lemma can be proved by simply replacing in Eqn. 3.9 the information states x_i with the belief likelihood l_i and taking the exponential of both the sides of the equation:

$$\exp\Big(\log\big(\pi_i\left(k+1\right)\big)\Big) = \exp\left(\log\big(\pi_i\left(k\right)\big) + \epsilon \sum_{j \in \mathcal{N}_i} \Big(\log\big(\pi_j\left(k\right)\big) - \log\big(\pi_i\left(k\right)\big)\Big)\right) \tag{3.10}$$

and thus:

$$\pi_i(k+1) = \pi_i\left(k\right)\exp\left(-\epsilon\Delta_i \log\big(\pi_i\left(k\right)\big) + \epsilon \sum_{j \in \mathcal{N}_i} \log\big(\pi_j\left(k\right)\big)\right) \tag{3.11}$$

where $\log\big(\pi_i\left(k\right)\big)$ has been taken out of the summation by multiplying its value by the number of nodes in agent i neighbourhood. This value is referred as the *in-degree* of node i and is denoted by Δ_i. Then:

$$\begin{aligned} \pi_i(k+1) &= \pi_i\left(k\right)^{1-\epsilon\Delta_i} \exp\left(\sum_{j \in \mathcal{N}_i} \epsilon \log\big(\pi_j\left(k\right)\big)\right) \\ &= \pi_i\left(k\right)^{1-\epsilon\Delta_i} \prod_{j \in \mathcal{N}_i} \pi_j\left(k\right)^{\epsilon} \end{aligned} \tag{3.12}$$

Noticing that $\epsilon = \gamma$ by hypothesis, the proof is completed by defining $\beta_i = 1 - \gamma\Delta_i$ □

Given the equivalence proved by Lemma 3.14, the convergence conditions for belief consensus are the same outlined in Theorem 3.12.

Chapter 4

Monitoring Framework Structure

4.1 Introduction

As extensively described in Section 2.1, most of the works about consensus available in literature pursues agreement about some measurable physical quantities. This is usually done in order to either cooperatively produce a representation of the monitored environment or for achieving coordinated motion for the networked agents. In the latter case consensus protocols are directly embedded within the agents guidance law, thus not representing a case of interest for the intended monitoring application. On the other hand, making reference to the former case, consensus protocols are usually adopted for implementing distributed averaging of measures or estimations : this, again, do not really fit with the research aims, that is the implementation of distributed assessment and decision-making techniques. For this reason, it is considered more convenient to consider data-processing and consensus seeking at a higher conceptual-level, involving possibly abstract features characterizing the targets. This will allow providing a comprehensive description and a working implementation of the monitoring level introduced in Section 1.5.1. In this chapter further details and functionalities of the monitoring level are presented, together with an example case study that will be leveraged in the remainder of the thesis for testing the proposed approaches. Actual implementations of this framework are provided in the following chapters.

4.2 Monitoring Level

With reference to the hierarchical structure proposed in Section 1.5.1, this thesis is mainly focused on application fields for consensus protocols that lie within the second layer, at a medium level of abstraction. Agreement at this level has been used in previous works, e.g., to recognise misbehaving vehicles in a team ([42, 43]), to cooperatively assess the shape of a target formation ([110]), to detect intrusions in a security perimeter ([41]) or to define the alarm status for a multi-zone area on the basis of multiple sensor readings ([40]). This section describes the structure and components of the monitoring layer, proposing a distributed, consensus-based solution for the *Situational Awareness* (SA) problem.

4.2.1 Features Classification

The approach adopted in order to identify possible TOIs is based on '*features*': these are pieces of information that try to quantify peculiar aspects of either the behaviour exhibited by the target or the environment where the target is moving. The main aspects of these two broad classes of features are described in the following sections.

4.2.1.1 Behavioural Features

Information about the target behaviour is leveraged for the definition of 'Behavioural Features': these can represent simple or complex characteristics of a target. The former come (almost) directly from the measurement or estimation process (e.g., the difference between the target speed and the speed limit), while the latter are obtained from an 'interpretation' of the target track and require some kind of elaboration (e.g., the fact that the target is moving along a not allowed traffic lane). Irrespective of the 'simple' or 'complex' nature of the features, these can be divided in two classes:

***Knowledge-Based* (KB) Features:** are defined on the basis of pieces of knowledge provided by experts in the field, establishing the parameters of interest and their acceptable/unacceptable values. These features are strictly application-dependant with reference to their definition and assessment, but are necessary for detecting the particular behaviours/manoeuvres of interest;

***Average-Based* (AB) Features:** leverage the idea that similar behaviours are expected from targets that have something in common, e.g. cars moving along the same road-section. AB features do not require specific knowledge in the field, but simply try to detect patterns that differ from normality: for this reason such features appear more general-purpose and will not need to be redefined unless the application field of the framework drastically changes.

4.2.1.2 Context Features

The second type of information considered, '*context information*', enables the definition of 'Context Features', which have the twofold purpose of:

- Characterizing the environment where the targets move, by means of a syntax that is understandable by all the agents in the team;
- Conditioning the target assessment.

The main idea behind the use of context information is that the significance of target behaviours can depend on the actual context. For example, a car exhibiting a high rate of overtaking manoeuvres in condition of heavy traffic could be normal. However, if the same amount of manoeuvres would be accomplished in the case of very light traffic, it would represent something anomalous, maybe a vehicle waving from a lane to another without any reason. Context features can be defined:

- Independently from each agent of the team;
- As the result of an agreement process among all the platforms (Fig. 4.1).

The latter case is of particular interest when dealing with heterogeneous networks, where different agents are equipped with different sensors, and can thus measure different aspects of the environment of interest. Finally, further information about the context, which cannot be sensed or inferred, could be provided by an external operator.

4.2.2 Target Assessment Implementation

Given these different forms of information, a technique for data and information fusion must be identified, together with assessment and classification policies that allow achieving comprehensive and seamless knowledge about the targets.

The theoretical tools adopted for implementing such assessment could depend on the nature of the features (AB/KB) and on the accuracy of the knowledge about the monitored events. A learning technique, e.g. Gaussian processes, seems to be an ideal tool for implementing 'Average-Based' behavioural features: by adopting a variable-in-time training set, the system can be made auto-updating, and thus capable of following recent trends in the behaviour of the targets. For KB features, on the other hand, the identification of a unifying approach capable of describing all the possible features seems to be rather challenging, since the features itself are usually application-dependent. Notice that the analytical tool to be adopted seems to be strictly related to the syntactic form in which the knowledge is expressed, and, as already mentioned, to the degree of accuracy for the knowledge itself. For example, approaches based on the Bayesian or possibility theory could be implemented if an exact formulation of the rules regulating the monitored environment is available, while more flexible methodologies, e.g. fuzzy logic, could be instead leveraged in the case of a more general, and thus less accurate, knowledge of the application field. The result of the target assessment process can be, e.g., a continuous, discrete or binary value indicating if and to what extent the monitored vehicle should be considered a TOI.

Fig. 4.1 depicts the reference structure for the proposed assessment process: there, the interactions among tracking, behavioural features detection, context features detection and target assessment processes are clearly shown, as well as the separation between sensor and monitoring layers (upper and lower boxes respectively).

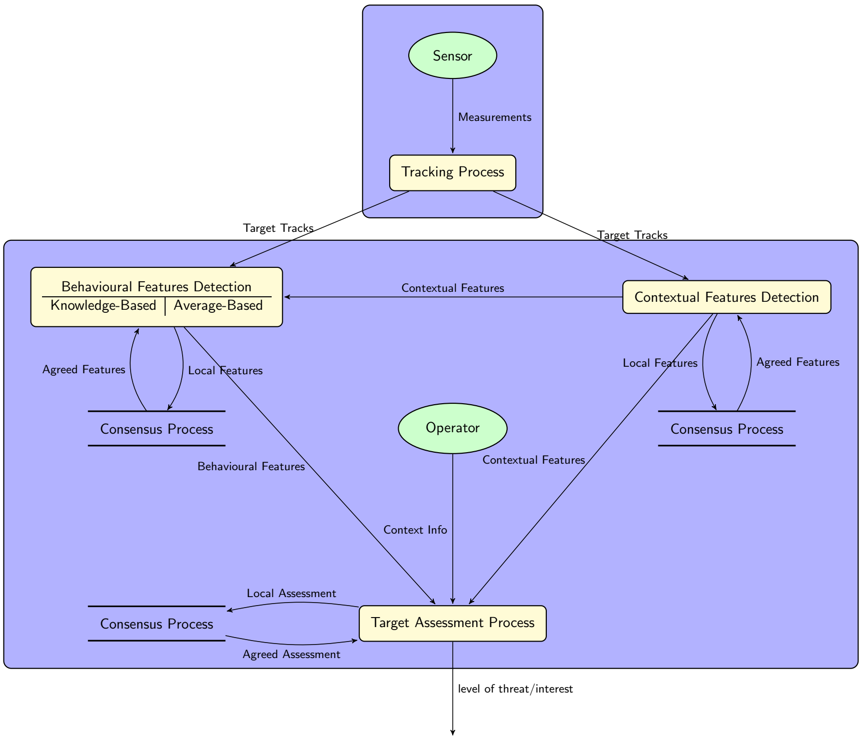


FIGURE 4.1: Reference Structure for the proposed assessment approach

4.2.3 Consensus Approaches

With reference to the target assessment problem, two different policies are possible for the implementation of consensus techniques:

Consensus on local assessments: local assessments individually produced by the agents in the team can be fed to a consensus algorithm ([105, 110, 131, 139]), allowing all the nodes to share a unique tactical picture of the monitored scenario. The type of agreement to be achieved, and thus the particular consensus protocol, depends on how the assessment value is defined: possible solutions could be average-consensus, max-consensus or belief-consensus ([105, 110, 131]).

Consensus on features: the consensus algorithm can be applied to the single behavioural features, allowing all the agents in the network to agree about specific characteristics of the targets. It should be noted that such approach leads to higher computational and communication burdens respect to achieving an agreement about the local assessments. Nonetheless, this solution could be preferable in some cases since the agents share not only their assessments but also the information on the basis of which the assessment is produced.

4.3 Case Study

A specific case study has been considered for the implementation of the assessment approach generically described in the previous section. In the remainder of the thesis, a road monitoring scenario is considered, where behaviours exhibited by ground vehicles moving along a motorway section are observed, tracked, analysed and assessed.

With reference to this case study, both synthetic data, produced by means of a traffic simulator, and real-world traffic data collected for the *Next Generation SIMulation* (NGSIM) research project has been taken into consideration. This twofold approach is adopted since it allows:

- To test the assessment process within a well-known scenario (when the simulated data is leveraged);
- To generalise the assessment capabilities to actual vehicles in real traffic conditions (by making all the necessary assumptions).

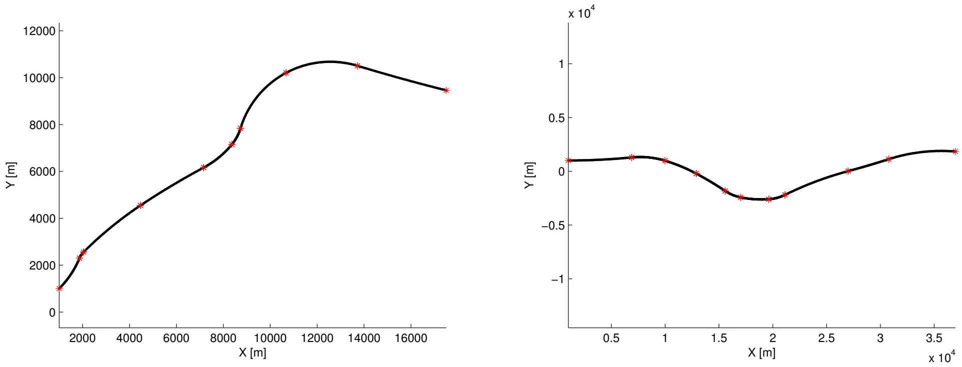
It is clear that, by considering a simulated scenario, an extreme simplification of the actual problem has been introduced. This allows, however, focusing on a controlled environment where the manoeuvres exhibited by the vehicles are to some extent known, and thus for which some form of ground truth about the targets is known. In this case, the assessment outcomes can be rated by means of appropriated performance metrics. On the other hand, the use of real-world data allows to capture most of the peculiarities of the monitored phenomena, but the lack of a ground truth against which to compare the assessment results makes the performance evaluation step not straightforward. For these reasons, the simplest and most intuitive approach to take seems to check whether the assessment trends obtained for real data follows those provided by the synthetic environment: this would somehow allow generalising the performances measured in the latter case to the former.

4.3.1 Traffic Simulator

This Section provides a brief, generic introduction to the considered simulation environment. More specifically, it describes how a generic road layout can be taken into account, the nature and capabilities of the monitoring agents and the characteristics of the different types of vehicles accounted for in the implemented dynamic simulator.

4.3.1.1 Road Representation

The approach taken in the adopted simulator represents the considered road section by means of a sequence of circular arcs, leveraging simple geometric techniques ([98]). By varying the radius length and the angular value associated with each segment it is possible to obtain a wide variety of road geometries, thus allowing the approximation of generic road sections (Fig. 4.2). The only simplifying assumption made about the road is the absence of intersections or roundabouts, and thus that the road represents a continuous path for the vehicles, with no possibility for direction change[1].



(a) (b)

FIGURE 4.2: Examples of road segments approximated as sequences of circular arcs

A nonlinear algorithmic transformation has been defined to translate generic Cartesian coordinates into a reference system relative to the road. This reference system, in each point of the road, has an axis aligned with the direction of the road, while the other, perpendicular, axis is orientated towards the right of the road. In so doing two 'curvilinear' coordinates (s, r) are being considered, where s indicates the actual position of the vehicle along the road, while r represents the lateral position on the road and thus, to some extent, the lane occupied by the vehicle. With such approach it is possible to consider a "straight road abstraction" for each generic road segment, as depicted in Fig. 4.3.

4.3.1.2 Agents Characterisation

The monitoring tasks are performed by three different types of agent:

[1]This assumption seems realistic for short and mid-length sections of a motorway (between two junctions).

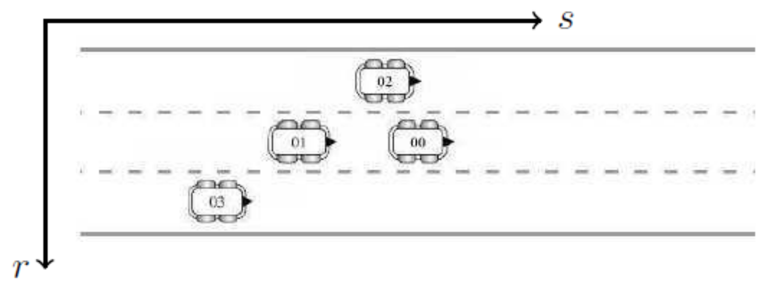


FIGURE 4.3: Abstract representation of a road segment

Fixed surveillance cameras along the road section : Arrays of surveillance cameras are used for obtaining piecewise continuous tracking on the vehicles. An array of surveillance cameras is defined as a sequence of cameras without any gap between their fields of view, whose resultant images are fed to a single tracking process; this, in the end, produces a unique set of tracks for the vehicles in the combined coverages of the camera array. From a simplistic point of view, an array of cameras can be represented as a single camera with extended coverage and sensing range. Image processing and tracking algorithms are assumed to be executed by the camera system, thus producing position and velocity estimation for each vehicle in the field of view. More precisely, the tracking process should provide the following information:

- Along-road position of the vehicle ([m]);
- Along-road speed of the vehicle ([m/s]);
- Lateral displacement of the vehicle with respect to the centre line ([m]).

It is assumed that this information is shared with monitoring *Unmanned Aerial Vehicles* (UAVs) and/or the ground station by appropriated communication links.

UAVs equipped with *Electro-Optical* (EO) sensors: The video coverage for the road section under consideration is also derived from the use of UAVs equipped with electro-optical sensors. From a functional point of view, their contribution is similar to that of fixed cameras; the assumed differences being a smaller sensing range and varying position over time.

UAVs equipped with *Ground Moving Target Indication* (GMTI) radars : The third class of agents involved in the scenario consists of UAVs equipped with GMTI radars. Each UAV is assumed to be capable of tracking all vehicles in its sensing range, thus performing:

- Measurement-to-track association;
- Track-to-track association;
- Position and velocity estimation.

Under these assumptions, for each vehicle on the road segment, the following information is available:

- Cartesian position (x, y) ([m]);
- Cartesian velocity vector $(\dot{x}, \dot{y})$ ([m/s]).

4.3.1.3 Targets Characterisation

The vehicles moving along the considered road section, which are the targets of the monitoring action, can be classified as follow:

Regular driver : Maintains a speed below the limit and behave correctly respect to the overtaking policy, the stopping distance and the lane occupation;

Cowboy driver : Can travel at a speed up to the limit, overtakes on both the side and can occupy the second and third lane even if not overtaking;

Snail driver : Similar to the type 1 driver, but maintains a speed significantly under the limit;

Cautious driver : Similar to the type 1 driver but keeps a greater stopping distance;

Racer driver : Can overtake on both the sides and is capable of travelling at speed higher than the limit;

4.3.2 Real-World Data

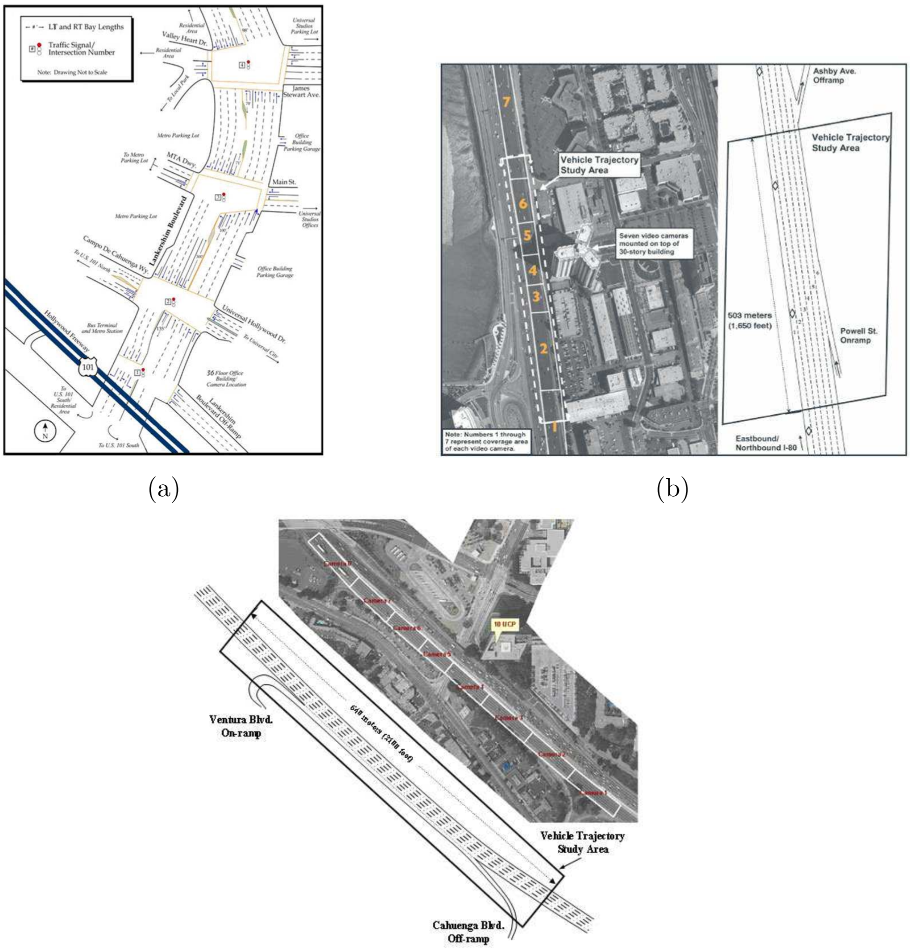
For the purpose of testing the proposed assessment approach in real-world conditions, the data collected for *Federal Highway Administration* (FHWA) NGSIM project ([2, 158]) has been taken into consideration. This is available to researchers in the field of transportation and traffic flow theory and incorporates a collection of single-vehicle related data, including processed trajectories in global and road-local reference frames. The main reasons behind the use of this dataset are the following:

- It represents an established approach since the dataset has been adopted in numerous previous studies, e.g., for:
  - validating traffic flow models;
  - interpreting traffic phenomena;
  - calibrating traffic/vehicle models.
- It is one of the very few available datasets providing microscopic information about the traffic flow (single vehicle trajectories);

- The data representation perfectly fits the reference frame adopted within the simulator that has been used for producing synthetic data: this allows applying the monitoring technique presented in this thesis to the NGSIM vehicle trajectories.

The information collected for the NGSIM project is split into three datasets, each of which is relative to a different road segment located in the United States (Fig. 4.4):

- Lankershim Boulevard in the Universal City neighbourhood of Los Angeles, CA;
- Eastbound I-80 highway in the San Francisco Bay area in Emeryville, CA;
- Southbound US 101, also known as the Hollywood Freeway, in Los Angeles, CA.



(a) (b) (c)

FIGURE 4.4: Schematic drawing and aerial photographs showing the lanes, traffic signals, junctions and intersections configurations within the study areas of Lankershim Boulevard (4.4a), I-80 (4.4b) and US-101 (4.4c).

Note that only the last two datasets have been taken into consideration in the following since Lankershim Boulevard does not match the highway layout assumed for the road in Section 4.3.1.1 . The complete structure for the NGSIM datasets entries is reported in Appendix C.1. For testing purposes, however, only a subset of the fields is taken into consideration, more specifically those providing the following information:

- Vehicle ID;
- Time;
- Local X position;
- Local Y position;
- Speed;
- Acceleration.

This information is sufficient to reconstruct the target trajectory with respect to the position, speed and acceleration within a 2-dimensional reference frame local to the road.

Since its release, NGSIM data has been widely leveraged in traffic studies: nonetheless, some analyses ([116]) have shown that the original data from the project is not totally reliable, exhibiting indeed a low degree of accuracy for the following criteria:

Internal Consistency: The consistency between space travelled and velocity and acceleration profiles;

Platoon Consistency: Physical consistency of the inter-vehicle spacing resulting from the individual trajectories of the vehicles.

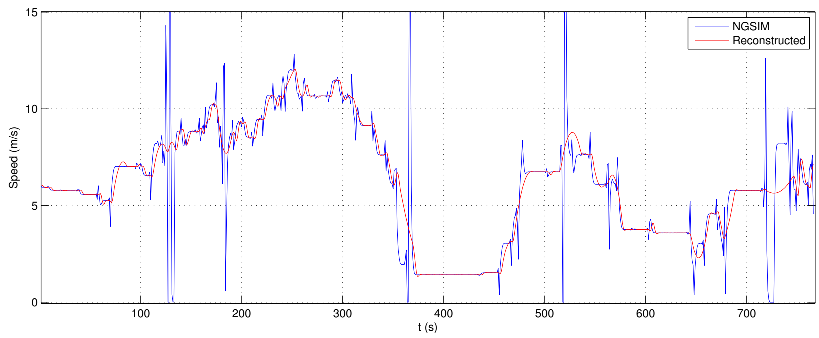


FIGURE 4.5: Example of NGSIM trajectory, original and reconstructed speed profiles

A hybrid filtering and reconstructing approach has been proposed in [91] with the purpose of fixing the position, velocity and acceleration profiles so that the two previous criteria can be better satisfied. The proposed solution aims at:

- Rejecting the outliers, which give rise to unphysical accelerations, decelerations and jerks;

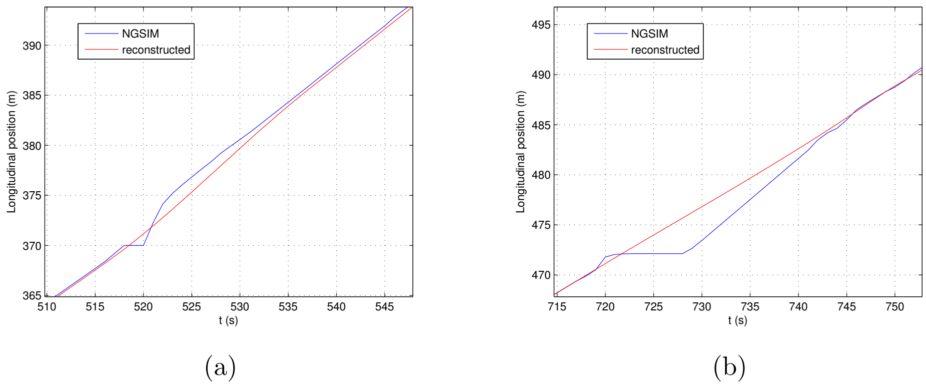


FIGURE 4.6: Examples of position profile correction for NGSIM trajectory

- Smoothing out random disturbances in the data.

This, however, should be achieved bearing in mind the need for preserving the driving dynamics (stopping vehicles, shifting gears, etc.) and maintaining the internal consistency of the trajectory.

Further details about the correction process applied to the trajectories are reported in Appendix C.2 . Some examples of the procedure outcomes are here reported, with reference to trajectory 1882 within the I-80 dataset:

- the reconstructed speed profile is depicted in Fig. 4.5, where it is possible to note the effective outlier and high-frequency noise removal, yet preserving the target dynamics;

- Since the correction of position errors cannot be appreciated plotting the whole trajectory, in Fig. 4.6 two details are reported for the longitudinal position in time, where sudden deviations clearly due to measurement errors have been corrected leading to a smooth profile.

Chapter 5

Distributed Traffic Monitoring Example

5.1 Introduction

In the previous chapter, the proposed general assessment framework has been outlined in terms of basic ideas, fundamental components and data-sources characterisation. In this chapter, a basic implementation example of such framework is presented. This makes reference to the simulation scenario described in the previous chapter. The main purpose of this work is to provide some early-stage indications about the assessment performances achievable by the proposed conceptual-features based approach. Consistently with the data-sources previously described, the specific application here considered is the monitoring of road traffic by multiple cooperating agents. The proposed assessment process simply consists in checking the value of high-level behavioural features by means of a "fuzzy-like", or hybrid, approach. This definition is due to the fact that for AB features no fuzzification process is applied to crisp inputs: instead, learning tools have been adopted for evaluating such features. From a functional point of view, these tools are seen from the rest of the system as black boxes representing a semantic input variable characterised by a single fuzzy set, thus producing outputs that comply with those of a *Membership Function* (MF). In the following, an extremely simple implementation of a FIS is presented, where semantic variables account for at most a single MF and a single output rule is taken into account. Since a single MF is considered for each semantic variable, the former indicates to what extent the concept associated with the variable is "verified" by the given crisp input. More specifically, a *Takagi-Sugeno-Kang* (TSK) fuzzy model ([149]) is adopted, where each KB/AB feature is represented as a semantic variable. The values associated with these are produced by means of MFs (KB case) or leveraging learning tools producing outputs in the range $[0, 1]$. Regardless of how they are generated, the values for the semantic variables are used for calculating the output associated with the only rule considered, which clearly corresponds the output of the inference/assessment process. The structure of the proposed assessment process is represented in Fig. 5.4. The outlined approach, based on fuzzy logic, has been adopted for the following reasons:

- It allows to easily implement a simple assessment process which can be used for obtaining some early-stage indications about the considered problem;
- It appears as the most intuitive solutions given the not accurate, common-sense knowledge for the assessment issue.

The simple implementation here considered has been used for investigating the effects of consensus techniques implementation within the proposed framework. More specifically, the assessment performances associated with two different consensus protocols, agreement on measurements and agreement on features evaluation, are taken into consideration and compared. Bearing in mind the layering introduced in Section 1.5.1, this corresponds to comparing the implementation of consensus protocol within the sensor or the monitoring level respectively.

5.2 Features

Several features of interest are here presented. These make reference to the traffic monitoring scenario and the classification scheme introduced in Chapter 4.

5.2.1 KB Behavioural Features

Behavioural features are related to behaviours that could raise the level of interest in a vehicle. Given a qualitative, non-specific, knowledge of the traffic dynamics problem, the approach taken for the definition of behavioural features is pretty simplistic, based on MFs. The considered KB features, which are associated with the semantic input variables of the FIS, are:

Irregular Speed: Indicates that the speed of the vehicle is above allowed limit:

$$f_{is} = \mathrm{smf}\left(x, -0.1l, 0.1l\right) \tag{5.1}$$

where x is the difference between the actual speed and the limit l and and smf() is a "s-shaped" MF defined as follow:

$$\mathrm{smf}(x,a,b)=\begin{cases}0 & x\leq a\\ 2\left(\frac{x-a}{b-a}\right)^2 & a\leq x\leq\frac{a+b}{2}\\ 1-2\left(\frac{x-b}{b-a}\right)^2 & \frac{a+b}{2}\leq x\leq b\\ 1 & x\geq b\end{cases} \tag{5.2}$$

Frenetic Driver: This feature is related to rate of overtaking manoeuvres over a certain time-window:

$$f_{fd}=\mathrm{smf}\left(x,1,2\right) \tag{5.3}$$

where x is the number of overtaking manoeuvres per minute[1]

Wrong Lane: This feature indicates the likelihood that the target is occupying a wrong lane, that is, it should move to the lane on its right. The value for this feature is obtained as the product[2] of three fuzzy MFs:

$$f_{wl}=f_{dp}\cdot f_{df}\cdot f_{dl} \tag{5.4}$$

with

$$f_{dp}=\mathrm{smf}\left(d_p,d_s,2d_s\right) \tag{5.5}$$

$$f_{df}=\mathrm{smf}\left(d_f,d_s,2d_s\right) \tag{5.6}$$

$$f_{dl}=\mathrm{smf}\left(d_l,w_l/2,w_l\right) \tag{5.7}$$

where d_p and d_f are the distances between the target and preceding and following vehicles respectively, d_s is the assumed safety distance, w_l the lane width and d_l is the lateral displacement with respect to the preceding (or following) vehicle. The functions f_{dp}, f_{df} and f_{dl} can be thought of as the marginal likelihood of the 'Wrong Lane' feature based only on the distance respect to the preceding vehicle, the distance respect to the following vehicle and the lateral distance respectively.

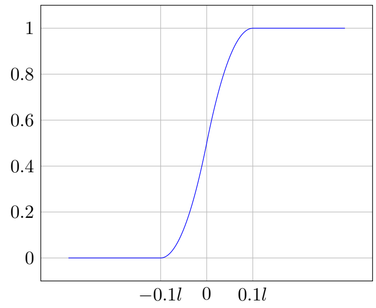
All the considered membership functions are depicted in Fig. 5.1.

With reference to the features just introduced, it should be notice that:

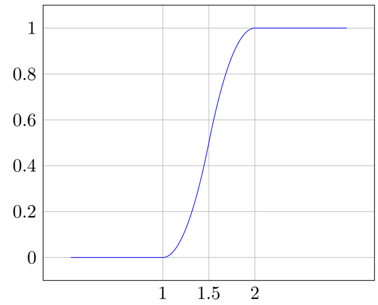
- The actual thresholds characterising the membership functions could be different for each agent, on the basis, e.g., of the agent confidence about the related measurements, making the membership function more or less selective;

[1]Overtaking manoeuvres can be detected, e.g., by means of the pattern-matching technique proposed by the authors in [157].

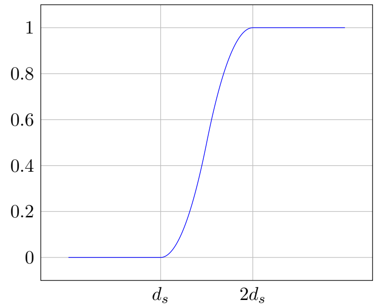
[2]The 'product' function has been adopted in this case for implementing a fuzzy T-norm.



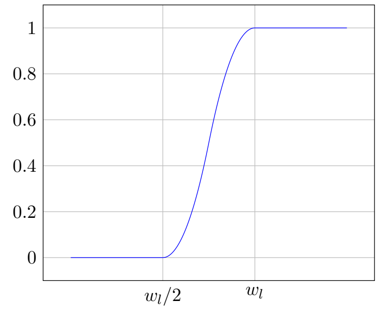
(a) MF for ‘Irregular Speed’, f_{is}



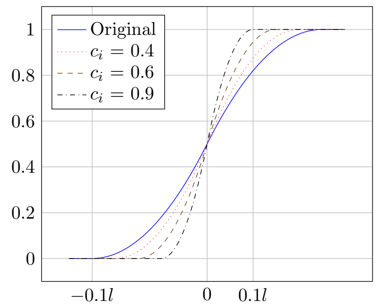
(b) MF for ‘Frenetic Driver’, f_{fd}



(c) MF for ‘Preceding Distance’ and ‘Following Distance’, f_{dp}, f_{df}



(d) MF for “Lateral Distance”, f_{ld}



(e) Adaptation of MF f_{is} on the basis of the “Congestion Index”

FIGURE 5.1: Adopted fuzzy membership functions

- Different agents could have different types of information about the targets, e.g., in the case where the local tracks make reference to different informative contexts, case not considered at this stage. The assessment logic implemented by each agent could thus rely not only on different thresholds for the membership functions, but even on completely different variables[3].

[3]Consider, for example, the case of security cameras assessing the ‘Wrong Lane’ feature: instead of geometrically evaluating from the image the distances between the vehicles (d_p,d_f), the agents could just analyse the pixels surrounding the target on the left side, for assessing if other vehicles are present.

5.2.2 AB Behavioural Features

The purpose of average-based features is to assess given aspects of the target behaviour on the basis of what is perceived as "normal" and what is not. The great advantage of this approach, conceptually more complex than the one adopted for KB features, is that no field-specific knowledge is required. By adopting behavioural features, there is no need to encode the experts' knowledge in the form of a rules set: the system is made able to monitor the targets and learn what is normal by itself, being thus capable of detecting anomalous behaviours. The fuzzy approach adopted in Section 5.2.1 for KB features cannot be leveraged for this different task: specific knowledge of the considered problem is indeed required for defining the MFs. This problem could be partially overcome by automatically retrieving the coefficients[4] for the Sugeno-type output function on the basis of labeled data (if available). Nonetheless, a characterization of the MFs involved in the process would be still required, making this solution not the most suitable for the considered problem. For this reason, the field of *machine learning* has been taken into consideration, and the actual tools adopted in the following are *Gaussian Processes* (GPs), ([119]). These have been chosen because they allow retrieving an analytical stochastic model for the behaviour of interest, in terms of mean value and variance. Respect to other learning tools that allow to predict the output associated with a new input without however providing any indication about the underlying model, e.g. neural networks, the use of GPs leads to a better understanding of the considered phenomenon and uncertainty bounds.

A GP is a stochastic process used for modelling an unknown, usually referred as '*latent*', function on the basis of a set of sample measurements. The implementation of '*GP Regression*' or '*GP Classification*' techniques requires a two-step approach:

- **Training Step**: a set of training points (each consisting of a domain point and the corresponding function value) is used to identify a set of parameters, denoted as '*hyperparameters*', which maximises the likelihood of the training points with respect to the stochastic process;
- **Test Step**: given the optimised hyperparameters, the stochastic model approximating the latent function is completely defined. This can then be leveraged for:
  - producing a prediction of the function value at a point not included in the training set;
  - calculating the likelihood of a given function value for a domain point.

With reference to the considered scenario, two relevant AB features have been identified:

[4]Assuming linear or polynomial output functions, which is usually the case.

Anomalous Speed: this feature aims to assess whether the vehicle is moving at a "reasonable" speed or not, on the basis of the speed of other vehicles in the proximities. The purpose of this feature is to identify vehicles that are not responding properly to unaccounted traffic obstacles, e.g. localised and unpredictable slow-down in the traffic flow. This function is implemented through the Gaussian process regression technique (Fig. 5.2), defining a reference speed profile along the road, iteratively trained using the collected data and thus self-adapting.

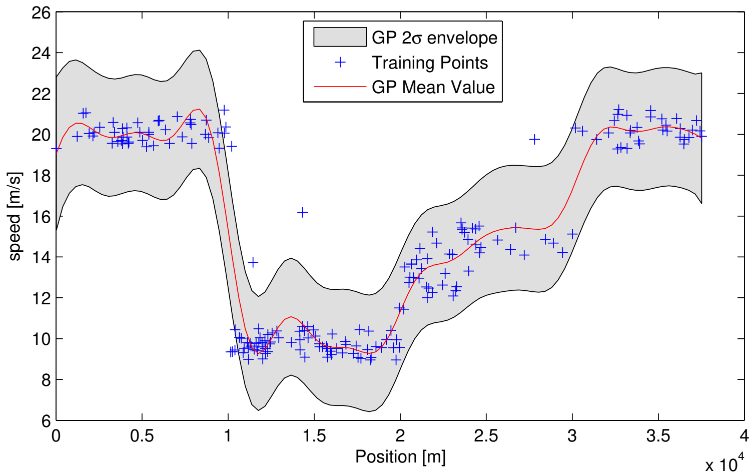


FIGURE 5.2: Gaussian process regression for the speed profile along the road

Frenetic Driver : this feature relates the 'overtaking ratio' (number of overtaking manoeuvres detected during a time interval) to the congestion state of the road, quantified by the dedicated context feature (Section 5.2.3). The objective is to classify the targets as "regular" or "frenetic" drivers, where the latter exhibit a significantly higher overtaking ratio respect to the former, depending also on the congestion index for the road. The idea is to consider a bi-dimensional surface where the coordinates are the congestion index (x) and the overtaking ratio (y), and to obtain a partitioning of such surface in two zones, which enclose the points associated with regular or frenetic drivers, respectively. This can be obtained using a Gaussian process classifier, which, once trained, estimates the likelihood for each point on the plane of being related to a frenetic[5] driver. Also in this case the training data for the GP is periodically updated for coping with a dynamic scenario where traffic condition can change over time. An example of Gaussian process classifier is depicted in Fig. 5.3.

For simplicity, only this AB version of the 'Frenetic Driver' feature is considered in the following

[5]The likelihood for the point to be related to a regular driver is obviously the complementary value to 1.

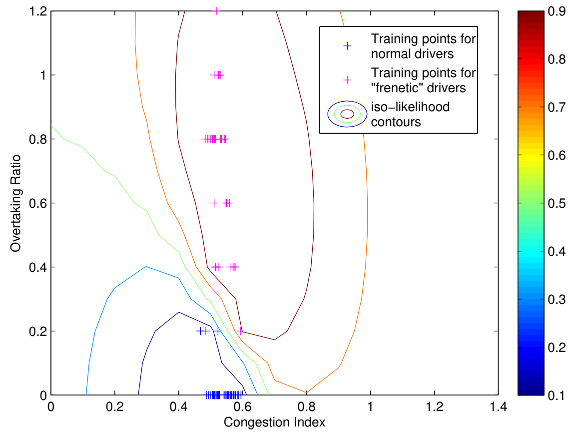


Figure 5.3: Likelihood contours produced by GP analysis

5.2.3 Context Features

The most intuitive example of contextual feature for the considered scenario is the congestion level of the monitored road section: vehicles moving at a speed under the allowed limit but not adequate to the actual congestion could indeed still represent a TOI. Congestion indexes are thus defined as follow:

$$c_j = \sum_{k \in \mathcal{N}_+(j)} \left\| p_j - p_k \right\|_2 \tag{5.8}$$

$$c = \begin{cases} \left(w_c - \frac{1}{n_v} \sum_{j=1}^{n_v} c_j \right) / w_c & \text{if } n_v > 1 \\ 0 & \text{if } n_v \in \{0, 1\} \end{cases} \tag{5.9}$$

where c and c_j are the congestion indexes calculated by the monitoring agent (overall and relative to target j respectively), while $\mathcal{N}_+(j)$ is the set of vehicles in front of target j at a distance smaller or equal to a threshold w_c, p_k is the position of vehicle k, and n_v is the number of vehicles tracked. The overall index c normalises the average of the vehicle-related indexes respect to the window dimension, and lies between 0 (no congestion at all) and 1 (heavy congestion).

5.3 Local Target Assessment

5.3.1 Context Information Integration

As mentioned when context features have been introduced, the rationale behind their use is that the interpretation of some behaviours is usually not "absolute", but may change depending on the surrounding environment. Context features can thus be thought of as "weights" for the behavioural context-dependant features, capable of highlighting, e.g., behaviours that would be considered legit in standard traffic condition, but are alarming in a congested scenario. For this reason, it is necessary to modify the membership functions for context-dependent features on the basis of the context information:

$$\begin{cases} f'_k = g\left(f_k, c\right), & \text{if } f_k \text{ is influenced by context} \\ f'_k = f_k & \text{otherwise} \end{cases}$$

where $k = 1 \ldots, n_f$ is the index for the considered features, f_k is the MF associated with feature k, c a context-related index (e.g., the one defined in Eqn. 5.9) and g is an adaptation function. An example of adaptation is provided in Fig. 5.1e, for the case of the "Irregular Speed" MF.

5.3.2 Local Belief Index

The last step involved in the local target assessment consists in the combined analysis of the possibly adapted behavioural features: the case where the monitored vehicle should be classified as a TOI is identified with the "presence" of at least one of the feature considered. For this reason, the following semantic rule is considered:

"If the vehicle has an irregular speed or a speed that differs greatly from neighbouring vehicles, or it moves on a wrong lane, or it is moving in a frenetic manner, then it is a TOI".

Within the fuzzy-like approach adopted, this rule can be stated as:

"IF X_1 is 'Irregular Speed' OR X_2 is 'Wrong Lane' OR X_3 is 'Anomalous Speed' OR X_4 is 'Frenetic Driver' THEN $u = 1$"

Some considerations about the rule are necessary:

- X_i are the crisp inputs associated with the features;
- In classic TSK models, an inference rule lists crisp inputs and MFs. However, in the proposed rule, semantic input variables are specified instead of MFs. This can be done because at most a single MF is associated with a semantic variable (avoiding any ambiguity[6]) and because it allows taking into consideration the case of AB features, where no actual MF is leveraged;
- The rule component *"X_2 is 'Wrong Lane' "* has been adopted for legibility and compactness reasons, but it should be expanded as "$X_{2,1}$ is f_{dp} AND $X_{2,2}$ is f_{df} AND $X_{2,3}$ is f_{dl}", as described in Section 5.2.1.

[6]The rule complies with the standard notation assuming that the MF associated with a semantic variable has the same name of the latter.

The rule 'firing strength', or weight, w is calculated by translating the OR conjunctions with the max operator (T-conorm). In the considered case, the context information is assumed to be applied to 'Irregular Speed' and 'Frenetic Driver' features only, and thus:

$$w = \max\left(f'_{is}, f_{as}, f'_{fd}, f_{wl}\right) \tag{5.10}$$

The final inference output can then be calculated by applying the rule weight to the rule output. In the simple case considered, because of the constant output for the only rule adopted, this can be written as:

$$f(u, w) = w \tag{5.11}$$

The assessment result is thus defined as:

$$\pi_j = \max\left(f'_{j,is}, f_{j,as}, f'_{j,fd}, f_{j,wl}\right) \tag{5.12}$$

where π_j represents the overall local belief of the agent in the fact that the target j exhibits at least one of the behaviours of interest, and thus represents a TOI. More generally:

$$\pi_j = \max_k f'_{j,k} \tag{5.13}$$

where k spans over the set of considered features.

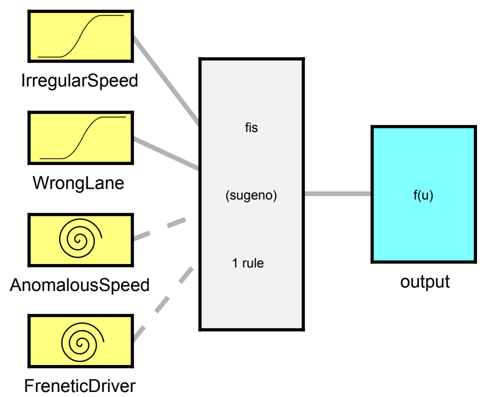


FIGURE 5.4: Conceptual structure of the assessment/inference process

5.4 Distributed Target Assessment

The desired outcome of the monitoring process is a distributed assessment of each target, obtained as result of an agreement process among all the agents in the team. Such result can be achieved by means of the 'belief consensus' algorithm presented in Section 3.4. With this approach, the local assessment obtained by means of (5.13) is interpreted as the local belief about the fact that the vehicle is a TOI and is used as consensus state within the following update rule:

$$\pi_{i,q}(k+1) = \pi_{i,q}(k)^{\beta_i} \prod_{j \in \mathcal{N}_i} \pi_{j,q}(t)^{\gamma} \tag{5.14}$$

This is the same of Eqn.3.6, where the index q has been introduced for denoting the considered target.

5.5 Simulations

With reference to the example scenario described in Section 4.3, a simulation framework implementing the features presented in this chapter has been developed. Numerous simulations have been performed and the numerical results are provided in the remainder of the chapter. The proposed simulations are aimed at testing the effectiveness of the different components of framework rather than the overall system. This approach has been taken since the complexity of the proposed assessment system suggests that, in order to fully understand the framework outputs, it is necessary to have deep insight into the performances and limits of the individual components. For this reason, in the remainder of the chapter the effectiveness of the distributed identification of just two individual features is tested, and not the final classification as TOI. Furthermore, two different consensus approaches for the features distributed evaluation are implemented and compared.

5.5.1 Simulation Description

5.5.1.1 Scenario Considered

Complying with the case study outlined in Section 4.3, a three-lanes motorway is here considered with multiple vehicles moving along, monitored by three different agent types: EO UAVs, GMTI UAVs and fixed cameras. The performed simulations take into account only a subset of the features introduced in Section 5.2.1: the objective of the agents is indeed to assess the two KB features 'Wrong Lane' and 'Frenetic Driver' in a distributed manner.

5.5.1.2 Performance Assessment

As an established approach, the framework ability to detect TOIs is evaluated by comparing the obtained results with some form of *Ground Truth* (GT). However, unlike a common estimation problem, where the GT is known, the objective of the proposed work is to identify and assess complex behaviours; in this case the GT is not directly available in numeric form and a more subtle definition is required. In the considered case, targets trajectories are obtained as result of a dynamic simulation: the GT is not known "a priori", but can be defined as the result of centralised assessment policies applied to perfect data. Obviously, for having a consistent comparison between the decentralised assessments and the GT, the centralised policies must be similar to those locally adopted by the agents. For this reason, the GT assumed valid within the following simulations consists of the outcomes of assessment logics similar to those adopted by the single agents but centrally applied to "perfect"[7] data. At each time step the results of the centralised assessment are compared with those obtained cooperatively by the agents, and the distributed performance is thus quantified as the percentage of time steps where the two methods provide the same results. More precisely, a threshold value is defined for each feature and, on the basis of these values, for each time step, vehicle and feature one of the following cases occurs:

- Correct detection, if the distributed and centralised approaches provide beliefs both above or under the threshold;
- False positive, if the distributed approach leads to a belief value above the threshold while the GT is below the threshold;
- False negative, if the distributed approach leads a belief value under the threshold and the centralised assessment above the latter.

The results shown in the Section 5.5.3 make reference to the total number of correct, false negative and false positive detections obtained during the entire simulation, averaged over all the vehicles.

5.5.1.3 Comparison with Average-Consensus

With the purpose of having a comparison term, in addition to the belief consensus outlined in Section 5.4, a basic average-consensus approach has been implemented. This consensus protocol is capable of reaching an agreement on track values instead of belief values, averaging position and velocity information provided by all the agents. Once all the agents share the same tracks, the assessment process is locally undertaken , on the basis of common logics, thus providing identical resulting belief values for all the agents. These results are then compared with the assumed GT (centralised assessment on perfect data) for having a performance indicator comparable with the one used for belief consensus. In summary, the two consensus approaches adopted are:

[7]This means to have an exact knowledge of position and velocity for all the vehicles in the scenario.

Belief Consensus : Consensus is reached about the likelihood of a feature, that is the feature value. This solution is conceptually slightly different from the approach described in Section 5.4, where the consensus protocol is applied to the overall local assessments, yet it still represents a correct implementation of the technique: a feature value is indeed interpreted as the belief of the monitoring agent in the fact that target is exhibiting the characteristic associated with that feature. The protocol in Eqn. 3.6 has been adopted for this case.

Track Consensus : The agents reach an agreement about the track values (position and velocity) for the targets, and calculate then the features values on the basis of the agreed track. For this purpose, the protocol in Eqn. 3.4 has been used.

Lemma 5.1. *Assuming a connected undirected topology for the graph associated with the communication network, both 'Belief Consensus' and 'Track Consensus' converge to an agreed value for all the agents in the network.*

Proof. The lemma is proved by simply noticing that a connected undirected graph can be considered as a strongly connected digraph. Then, Theorem 3.12 guarantees the convergence of both the algorithms. □

5.5.2 Simulation Parameters

The parameters adopted for the simulations are the following:

- Time parameters:
  - Simulation step time: 0.5 s;
  - Simulation time: 50 min;
- Scenario parameters:
  - Road segment length: ∼20 Km;
  - Number of lanes: 3;
  - Max number of vehicles: 30;
  - Vehicles speed range: 10–30 m/s;
- Consensus algorithms parameters ('Belief' and 'Average'):
  - Consensus Steps: 10 ;
  - Consensus Gain:

$$\big(1/(n-1)\big) * 0.8 = 0.26$$

    where $n = 4$ is the number of agents involved in the consensus process;
- Fixed Camera Parameters:
  - Sensing Range [8] : 1000 m;
  - Coverage factor [9]: 1/3 or 1 (full coverage);

[8]Maximum distance for a trackable vehicle.
[9]Portion of monitored road respect to the total length.

  - Longitudinal position error *Standard Deviation* (SD): 2 m;
  - Lateral position error SD: 0.5 m;
  - Speed tracking error SD: 0.5 m/s;

- UAV Camera Parameters:
  - Sensing Range: 400 m;
  - Number of camera-equipped UAVs : 2;
  - Longitudinal position error SD: 2 m;
  - Lateral position error SD: 0.5 m;
  - Speed tracking error SD: 0.5 m/s;

- GMTI Radar Parameters:
  - Number of GMTI-equipped UAVs: 1;
  - Position tracking-error SD: 0.5 m;
  - Velocity tracking-error SD: 0.5 m/s.

The network between the GMTI UAV, the two EO UAVs and the array of fixed cameras is assumed to enable bidirectional communications and to contain a spanning tree in the associated graph. Under these conditions Lemma 5.1 is verified, guaranteeing consensus for all the agents.

5.5.3 Simulation Results

The simulations evaluate the individual values of a KB and a AB feature, 'wrong lane' and 'frenetic driver', respectively. The numeric results here presented are relative to two different conditions for the video coverage provided by fixed cameras: a first case of partial coverage (1/3 of the road length) and a second case of full coverage; all the other parameters are the same for the two cases. It is important to notice that, even if a single simulation for each condition has been run, the results provided are statistically relevant since a huge number of vehicles have been monitored during the 50 minutes of simulation. All the numeric results are summarised in Table 5.1.

TABLE 5.1: Numeric results for the comparison of the two consensus approaches

		33% Coverage Case		100% Coverage Case	
		Frenetic Driver	Wrong Lane	Frenetic Driver	Wrong Lane
Correct	Belief Cons.	86	87	92	95
	Track Cons.	74	63	56	93
False positive	Belief Cons.	5	13	4	5
	Track Cons.	17	36	38	6
False Negative	Belief Cons.	9	< 1	4	< 1
	Track Cons.	9	< 1	6	< 1

5.5.3.1 Results with 1/3 of Video Coverage

The percentages of correct, false positive and false negative detections obtained by means of belief consensus and track consensus in the case where fixed cameras cover one-third of the overall road length are depicted in Fig. 5.5.

Under these conditions, the approach proposed, which is based on consensus about local assessments, provides results highly consistent with the GT (first two columns of Table 5.1): the method outcomes correspond indeed to those provided by the centralised assessment process in more than 85% of the cases. The effectiveness of this method is then confirmed comparing these results with those obtained applying average-consensus to the tracks values under identical simulation conditions. From this comparison, it can be noticed that belief consensus achieve better results[10] than track consensus for both the considered features. For example, in the case of ‘Wrong Lane’ feature, the percentage of correct detections for the track consensus technique is considerably lower respect to that provided by consensus on assessments (63% against 87%).

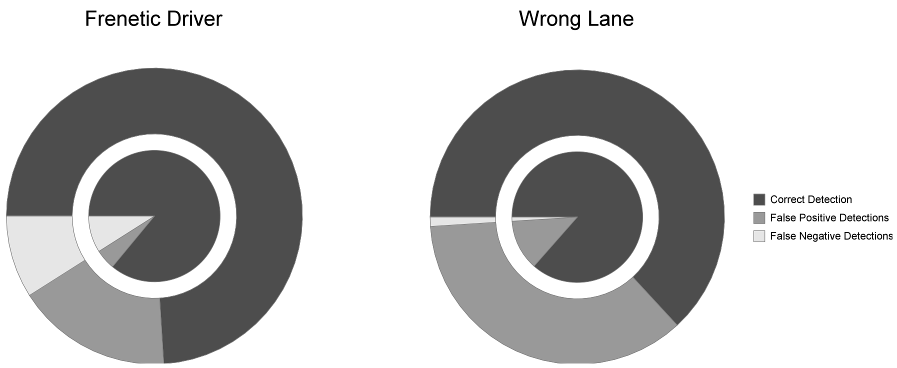


Figure 5.5: Detections fractions for the considered features. Results provided by belief consensus (inner circle) and track consensus (outer ring) in the case of 1/3 of video coverage.

5.5.3.2 Results with Full Video Coverage

A simulation similar to that of Section 5.5.3.1 has been performed in the case of full video coverage, for the purpose of investigating the relevance of imagery information. Results are depicted in Fig. 5.6 for both the consensus approaches and both the features.

Simulation results for this case (last two columns of Table 5.1) confirm the greater effectiveness of the belief consensus approach respect to track consensus, providing better results for both the considered features. By comparing the detection ratios obtained under this simulation condition respect to the ‘1/3 coverage’ case, it is possible to note that for belief consensus a performance improvement is obtained in the assessment of

[10]i.e., higher correct detections and lower false positive detections ratios

both the considered features. On the other hand, in the case of the average-consensus approach, worst performances can be observed when the 'Frenetic Driver' feature is taken into account. The reasons for this increase in false alarms (and, obviously, for the associated decrease of correct detections) are not immediate but can be probably explained considering that video measurements from fixed cameras are less accurate respect to those produced by GMTI sensing on the longitudinal direction. This can lead to errors in the calculation of the congestion index, which sensibly affects the 'Frenetic Driver' feature: as can be seen in Figure 5.3, indeed, in presence of high values for the congestion index the target is classified as "frenetic" even when small numbers of overtaking manoeuvres are detected. In such scenario, the agreement on local assessment seems to be capable of better rejecting the described type of error, while simple averaging is more sensitive to the increased quantity of "low-quality" information collected.

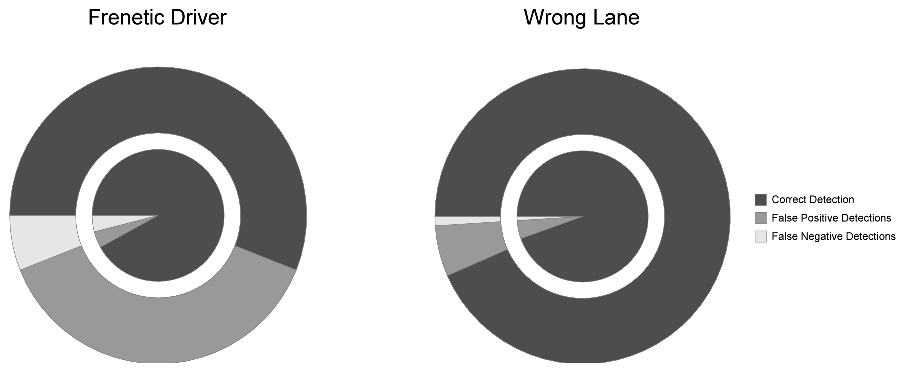


Figure 5.6: Detections fractions for the considered features. Results provided by belief consensus (inner circle) and track consensus (outer ring) in the case of full video coverage.

5.6 Conclusions

In this chapter the distributed target assessment problem has been considered, proposing an approach that complies with the conceptual layering for a monitoring framework that has been previously outlined (Sections 1.5 and 4.2). Focusing on the functionalities of the monitoring layer, a set of high-level target features allowing to accomplish the assessment task have been theoretically and practically[11] described. Finally, two different approaches to distributed assessment, based on two different consensus protocols, have been compared by means of numerical simulations.

The reported simulations show that distributed assessment based on consensus techniques is a feasible approach for implementing a framework dedicated to traffic monitoring and, more generally, to behaviour monitoring and TOI identification. From the results obtained, where assessments based on features-consensus are better than those

[11]With reference to a specific case-study.

provided by measurements-consensus in all the cases, the idea of implementing common situational awareness within a team of cooperating autonomous agents by means of consensus at high conceptual-level seems reasonable and motivated.

Other than the performance improvement described in Section 5.5, consensus at high conceptual-level (e.g., on local assessments) has the advantage of sensibly limiting the communication burden in cases where the amount of information leveraged for assessing the vehicle behaviour is considerable. Furthermore, it seems a particularly suitable solution for heterogeneous teams, where different agents can measure different characteristics of the target: in such case a policy based on agreement about measurements could indeed be not feasible, while, on the other hand, each agent would probably still be able to assess certain limited aspects of the target and share this information with the cooperating agents.

Chapter 6

Behaviour Assessment

6.1 introduction

In the previous chapter a basic implementation for an automated monitoring and assessment framework has been presented. This implementation characterises the features by means of fuzzy MFs and derives the assessment about a target on the basis of a t-conorm aggregation operator. Such a simplistic approach has been adopted for the following reasons:

- For obtaining immediate indications about the effectiveness of the proposed distributed approach to target assessment;
- For testing at what level of abstraction it is more convenient to implement a consensus process among cooperating agents.

Given the promising results provided by the simple fuzzy-like approach considered so far, the analysis of assessment techniques based on high conceptual-level features appears as worthy of further investigation. More precisely, it seems necessary to define:

- A more rigorous approach to the characterisation and evaluation of behavioural features;
- A proper test policy allowing to jointly account for for multiple features involved in the assessment process .

In order to address these requirements, this chapter provides a novel stochastic description of behavioural features. This description relies on a hierarchical architecture capable of dealing with features of arbitrary conceptual complexity. Furthermore, an hypothesis-testing based decision policy has been developed allowing for the classification of a target as TOI on the basis of the multivariate distribution associated with the set of considered features.

Lastly, the assessment approach based on these novel analytical tools has been tested through an extensive set of simulations. These have taken into account both a simulated environment and real-world data and have provided valuable insights about the effectiveness of the proposed techniques.

6.2 Probabilistic Representation for the Behavioural Features

6.2.1 Features Properties and Classification

One of the main contributions of the thesis is to propose an assessment approach where a probabilistic description is adopted for the behavioural features of a target. These are assumed to have a dimensionless value in the range [0, 1], where 0 indicates that the considered feature is not exhibited at all by the target, while 1 means that the feature is fully consistent with the target behaviour. The lack of an actual physical process to model enables a certain degree of freedom in the choice of the distributions family to adopt. Beta distributions (Fig. 6.1) have been chosen for representing the likelihood of the feature values since:

- Their domain matches the feature value range;
- They offer great flexibility in the definition of the probability curve, allowing, e.g., exponential-like functions and both symmetric and asymmetric bell-type curves.

The fact that a random variable X follows the beta distribution is denoted in the following as:

$$X \sim \text{beta}\left[\alpha, \beta\right] \tag{6.1}$$

where α and β are the distribution parameters. Each feature is assumed to be independent from the others[1], and the joint *Probability Density Function* (PDF) will be thus considered[2] $f_{\Xi}(\xi^1, \xi^2) = f_{\Xi^1}(\xi^1) f_{\Xi^2}(\xi^2)$ for the case of two random variables Ξ^1 and Ξ^2. The nature of a feature can be:

[1] They likely represent distinct and uncorrelated aspects of the target behaviour.

[2] In the following, for the purpose of avoiding a cumbersome notation, when referring to a PDF or CDF, the considered random variable will not be explicitly indicated, but it will be implicit in the argument: $f_{\Xi^i}\left(\xi^i\right) \rightarrow f\left(\xi^i\right)$, $F_{\Xi^i}\left(\gamma_i\right) \rightarrow F\left(\gamma_i\right)$.

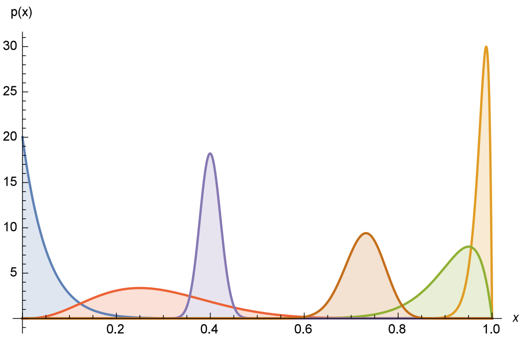


FIGURE 6.1: Examples of beta distributions with different parameters

- **Instantaneous**: the feature makes reference to any target condition that can be evaluated at a single time-instant, e.g., its speed;
- **Non-instantaneous**: the feature makes reference to coordinated and persistent conditions for the target, e.g. an overtaking manoeuvre.

'Instantaneous' features are described and analysed in Section 6.2.2, while the others ("Non-instantaneous") are not further considered within this chapter. The focus of this is, indeed, on assessing whether the proposed approach is suitable to the problem of threat assessment, and such analysis can reasonably be performed by considering a minimal feature set of reduced complexity. Nonetheless, the problem of detecting coordinated (time-correlated) behaviours of interests is addressed in Chapter 7.

6.2.2 Features Description

This section describes the proposed procedure for deriving the *Probability Density Functions* (PDFs) associated with instantaneoustarget features. Two possible approaches to features definition are here presented, depending on its conceptual complexity. Then, two feature examples making reference to behavioural aspects of interest for the problem of traffic monitoring, 'Irregular Speed' and 'Wrong Lane' occupation, are presented.

6.2.2.1 Layered Description

The features considered in the proposed work represent characteristics of a target at both low and high level of abstraction. Some features can indeed make reference to straightforward and directly measurable aspects of the target behaviour, while others could require a more complex reasoning, possibly involving multiple conditions. For this reason, two different approaches are proposed for the probabilistic definition of a feature: a single-layer method and a hierarchic, multi-layer approach. The proposed layered representation for behavioural features has been preferred to other approaches, e.g. conditional probabilities, because it provides a more immediate and intuitive description of abstract concepts for which the definition of proper multivariate PDFs would not be easy to derive and motivate.

Single-Layer Features This category includes features that can be defined by means of a single beta distribution, on the basis of a generic number k of measurements indicated as $\mathbf{Z} = [z_1, z_2, \ldots, z_k]$. Considering the generic feature i for a target[3], its value corresponds to a random variable ξ^i distributed as follow:

$$\xi^i \sim \text{beta}\left[\mathrm{g}^{\alpha,\mathrm{i}}\left(\mathbf{Z}\right), \mathrm{g}^{\beta,\mathrm{i}}\left(\mathbf{Z}\right)\right] \tag{6.2a}$$

$$\mathbf{Z} \sim f\left(\mathbf{Z}^*\right) \tag{6.2b}$$

where $\mathbf{Z}^*$ is the vector of the actual quantities measured, f is a generic distribution representing the measurement error, $g^{\alpha,i}$ and $g^{\beta,i}$ are functions such that $\mathbb{R}^k \rightarrow \mathbb{R}$, which specify the values for the parameters α and β, respectively. Note that, even if a two-level hierarchy of distributions is considered, this case is referred to as 'single-layer' since only one level deals with the problem of behaviour assessment (Eqn. 6.2a), while Eqn. 6.2b simply describes the sensor model[4] .

Multi-Layer Features A multi-layer description is suggested when the feature assessment is a complex task, involving multiple measurements and requiring more conditions to be considered at the same time. In such cases the adoption of a hierarchical stochastic description is proposed, where the characterisation of a random variable at a given level is provided as a function of the random variables introduced at the previous level[5]. At the top of the hierarchy there is the feature of actual interest (6.3a), indicating to what extent a target is compatible with the feature concept, while the bottom layer (6.3g) corresponds with the sensor/estimation level. All the distributions at levels 2 to $n-1$ (6.3f-6.3b) are referred in the following as *sub-features* and represent partial aspects involved in the feature assessment. The analytic description of the described

[3]Target index is omitted for simplicity.
[4]Or the accuracy of the related estimation process if such step is performed.
[5]Makes an exception, of course, the first (bottom) layer.

hierarchy is the following:

$$\xi^i \sim \text{beta}\left[\mathrm{g}_{\mathrm{n}}^{\alpha,\mathrm{i}}\left(\boldsymbol{\xi}_{n-1}^i\right), \mathrm{g}_{\mathrm{n}}^{\beta,\mathrm{i}}\left(\boldsymbol{\xi}_{n-1}^i\right)\right] \tag{6.3a}$$

$$\xi_{n-1,m_{n-1}}^i \sim \text{beta}\left[\mathrm{g}_{\mathrm{n\text{-}1,m_{n\text{-}1}}}^{\alpha,\mathrm{i}}\left(\boldsymbol{\xi}_{n-2}^i\right), \mathrm{g}_{\mathrm{n\text{-}1,m_{n\text{-}1}}}^{\beta,\mathrm{i}}\left(\boldsymbol{\xi}_{n-2}^i\right)\right] \tag{6.3b}$$

$$\vdots$$

$$\xi_{n-1,1}^i \sim \text{beta}\left[\mathrm{g}_{\mathrm{n\text{-}1,1}}^{\alpha,\mathrm{i}}\left(\boldsymbol{\xi}_{n-2}^i\right), \mathrm{g}_{\mathrm{n\text{-}1,1}}^{\beta,\mathrm{i}}\left(\boldsymbol{\xi}_{n-2}^i\right)\right] \tag{6.3c}$$

$$\vdots$$

$$\xi_{3,1}^i \sim \text{beta}\left[\mathrm{g}_{3,1}^{\alpha,\mathrm{i}}\left(\boldsymbol{\xi}_{2}^i\right), \mathrm{g}_{3,1}^{\beta,\mathrm{i}}\left(\boldsymbol{\xi}_{2}^i\right)\right] \tag{6.3d}$$

$$\xi_{2,m_2}^i \sim \text{beta}\left[\mathrm{g}_{2,\mathrm{m}_2}^{\alpha,\mathrm{i}}\left(\mathbf{Z}\right), \mathrm{g}_{2,\mathrm{m}_2}^{\beta,\mathrm{i}}\left(\mathbf{Z}\right)\right] \tag{6.3e}$$

$$\vdots$$

$$\xi_{2,1}^i \sim \text{beta}\left[\mathrm{g}_{2,1}^{\alpha,\mathrm{i}}\left(\mathbf{Z}\right), \mathrm{g}_{2,1}^{\beta,\mathrm{i}}\left(\mathbf{Z}\right)\right] \tag{6.3f}$$

$$\mathbf{Z} \sim f\left(\mathbf{Z}^*\right) \tag{6.3g}$$

where i denotes the considered feature, m_j is the number of sub-features defined at level j with $j = 2, \ldots, n-1$, $\boldsymbol{\xi}_j^i = \left[\xi_{j,1}^i, \ldots, \xi_{j,m_j}^i\right]$ is the vector of level j sub-features and n is the total number of layers involved in the feature definition. Parameters α and β for the distributions in the hierarchy are defined by means of two families of functions $\mathbf{g}^{\alpha,i} = \left[\mathbf{g}_2^{\alpha,i}, \ldots, \mathbf{g}_n^{\alpha,i}\right]$, $\mathbf{g}^{\beta,i} = \left[\mathbf{g}_2^{\beta,i}, \ldots, \mathbf{g}_n^{\beta,i}\right]$ where[6] $\mathbf{g}_j^i : \mathbb{R}^{m_{j-1}} \rightarrow \mathbb{R}^{m_j}$, while $\mathbf{g}_2^i : \mathbb{R}^k \rightarrow \mathbb{R}^{m_2}$. The single elements of a vectorial function $\mathbf{g}_j^i$ are indexed as $\mathrm{g}_{j,h}^i$ with $h = 1, \ldots, m_j$.

6.2.2.2 Feature Examples

Two examples of target feature are here considered, 'Irregular Speed' and 'Wrong Lane', which can be related to the two typologies previously introduced (single and multiple layer, respectively).

'Irregular Speed' Feature The Irregular Speed feature characterises vehicles that exceed the local speed limit. A speed limit entails a rigid, binary condition (either the limit is respected or it is not), and thus a probabilistic approach for such assessment could be considered inappropriate. Nonetheless, a probabilistic description of the feature is here used for two main reasons:

[6] Superscripts α and β are dropped in the remainder of the section for avoiding a cumbersome notation.

1. It is relevant to provide a simple example of how to define a probabilistic description of a feature. This approach will be valid for various different aspects of a moving vehicle;

2. Features are defined with the final purpose of detecting drivers representing possible threats: when moving slightly faster than allowed, a vehicle does not represent a more likely threat respect to when its speed is barely under the limit. Therefore, reduced sensitivity to speed deviations near the speed limit is required in the feature definition in order to have a reasonable assessment, by "smoothing" the binary condition associated with the concept of limit.

The definition for the Irregular Speed feature is quite immediate, since it considers only the measured speed and a known limit: the feature can thus be described on the basis of the single-layer approach introduced in Section 6.2.2.1. The PDFs considered for this specific feature are the following:

$$\xi^{is} \sim \text{beta}\left[\frac{\kappa}{\sigma_v^2}\,\text{Exp}\left(\frac{s_d}{\nu}\right), \frac{\kappa}{\sigma_v^2}\,\text{Exp}\left(-\frac{s_d}{\nu}\right)\right] \tag{6.4a}$$

$$s_d \sim \mathcal{N}\left[s^* - s_l, \sigma_v^2\right], \tag{6.4b}$$

which are the proposed implementation for Eqn. 6.2a and 6.2b, respectively. In Eqn. 6.4a and 6.4b, s^* is the real speed of the target, s_l the known limit and the following parameters are adopted:

- κ, modulus adaptation: introduces a degree-of-freedom in the definition of the resulting selectiveness of the PDF;

- ν, tolerance adaptation: defines the correspondence between the input data and the distribution mode position. This parameter thus defines if some given data should lead to a "high" or "low" value for the feature, regardless of the selectiveness of the associated distribution.

- σ, measurement accuracy: reflects the uncertainty in the measurement/estimation process. Similarly to k, this parameter symmetrically increases or reduces α and β, altering the selectiveness of the function without changing the distribution mean.

The definition provided in (6.4a) clearly indicates the effect of the measured s_d on the PDF shape: when $s_d = 0$, that is the target is moving almost at the traffic speed limit, α and β are the same and the feature PDF is centred in 0.5 (target is "in between" respecting the limit and exceeding it). By increasing s_d, α increases and β decreases, while the opposite, decreasing α and increasing β, happens when the difference s_d is reduced. This is due to the fact that as s_d tends to either positive or negative values, α and β will have opposite derivatives (one increases, the other decreases), shifting the distribution to the left or right. Regarding the functions used for defining the beta distribution parameters, the exponential form has been chosen on the basis of two requirements:

- α and β should be always positive;
- The modulus of α and β should grow quickly for $s_d > 0$. Parameter ν can then be used for "slowing down" the growth of the exponential term, and thus to delay the "alarm" condition. This leads to a less rigid assessment of the feature, where drivers are allowed to slightly exceed the limit before the feature assumes high values.

'Wrong Lane' Feature The Wrong Lane feature characterises vehicles that are moving along the overtaking lane without an apparent valid reason, that is they do not have any close preceding or following vehicle, and thus they cannot be considered in the middle of an overtaking manoeuvre. The feature is described making reference to a straight road and a local reference system with the s axis in the direction of the road and the r axis orthogonal to the former. A detailed description of such road representation and how it can be considered representative of generic road segments is reported in Section 4.3.1.1.

Wrong Lane assessment is not a straightforward task, and it is thus implemented by means of the multi-layer approach described in 6.2.2.1. In this case a 3-levels hierarchy has been adopted[7]: level 1 (bottom level), corresponding to the sensing/estimation process (6.3g), level 2 (middle level), implementing a layer of sub-features (6.3f-6.3e) and level 3 (top level), representing the feature of actual interest (6.3a).

Level 1 (bottom level): A stochastic description of the initial quantities leveraged for the feature assessment is here provided:

$$\delta r_p \sim \mathcal{N}\left[r_v^* - r_p^*, 2\sigma_r^2\right] \tag{6.5a}$$

$$\delta r_f \sim \mathcal{N}\left[r_v^* - r_f^*, 2\sigma_r^2\right] \tag{6.5b}$$

$$\delta s_p \sim \mathcal{N}\left[s_p^* - s_v^*, 2\sigma_s^2\right] \tag{6.5c}$$

$$\delta s_f \sim \mathcal{N}\left[s_v^* - s_f^*, 2\sigma_s^2\right] \tag{6.5d}$$

where σ_r^2 and σ_s^2 are the error variances for lateral and longitudinal measurements, respectively, while r^* and s^* are the real coordinates of the considered (v subscript), preceding (p) and following (f) vehicles. δr_p, for example, is a random variable

[7]For the sake of notational simplicity and for having a more explicative nomenclature, the feature name is not reported in the sub-feature symbols, and the numerical indexes in (6.3) have been replaced by names denoting the aspect represented by each sub-feature.

describing the lateral distance between the considered vehicle and the preceding one, while δs_p provides a stochastic description of the longitudinal distance between the two same vehicles. Similar considerations apply to δr_f and δs_f.

Level 2 (middle level): This level defines random variables representing various conditions on the relative position between adjacent vehicles. The sub-feature $\xi_{\delta r_p}$, for example, indicates to what extent the lateral distance between the considered vehicle and the preceding one matches the condition of being on a wrong lane; the same is valid for the other variables, where the letters r, s, p, f indicate lateral distance, longitudinal distance, preceding and following vehicle, respectively.

This level of the hierarchy is implemented by the following equations:

$$\xi_{\delta r_p} \sim \text{beta}\left[\frac{\kappa}{\sigma_r^2}\,\text{Exp}\left(\frac{(\delta r_p - \Delta_1)(\Delta_2 - \delta r_p)}{\nu}\right), \frac{\kappa}{\sigma_r^2}\,\text{Exp}\left(-\frac{(\delta r_p - \Delta_1)(\Delta_2 - \delta r_p)}{\nu}\right)\right] \tag{6.6}$$

$$\xi_{\delta r_f} \sim \text{beta}\left[\frac{\kappa}{\sigma_r^2}\,\text{Exp}\left(\frac{\left(\delta r_f - \Delta_1\right)\left(\Delta_2 - \delta r_f\right)}{\nu}\right), \frac{\kappa}{\sigma_r^2}\,\text{Exp}\left(-\frac{\left(\delta r_f - \Delta_1\right)\left(\Delta_2 - \delta r_f\right)}{\nu}\right)\right] \tag{6.7}$$

$$\xi_{\delta s_p} \sim \text{beta}\left[\frac{\kappa}{\sigma_s^2}\,\text{Exp}\left(\frac{(\delta s_p - \Delta)}{\nu}\right), \frac{\kappa}{\sigma_s^2}\,\text{Exp}\left(-\frac{(\delta s_p - \Delta)}{\nu}\right)\right] \tag{6.8}$$

$$\xi_{\delta s_f} \sim \text{beta}\left[\frac{\kappa}{\sigma_s^2}\,\text{Exp}\left(\frac{\left(\delta s_f - \Delta\right)}{\nu}\right), \frac{\kappa}{\sigma_s^2}\,\text{Exp}\left(-\frac{\left(\delta s_f - \Delta\right)}{\nu}\right)\right] \tag{6.9}$$

where Δ is a threshold value for the longitudinal vehicle separation, while Δ_1 and Δ_2 are threshold values for the minimal and maximal lateral vehicle separation, respectively[8]. From a numerical point of view, the PDFs for the variables $\xi_{\delta r_p}, \xi_{\delta r_f}, \xi_{\delta s_p}$ and $\xi_{\delta s_f}$ can be easily obtained by replacing the variables $\delta s_f, \delta s_p, \delta r_f$

[8] In all the formulas reported in this section the effects of parameters κ, σ and ν are the same of those presented for the Irregular Speed feature, but the actual values could be different for each case; no subscripts have been added to the parameters labels for the sake of notational simplicity.

and δr_p with the respective *Maximum Likelihood Estimators* (MLE) , simply provided from arithmetic operations on the assumed vehicles coordinates:

$$\widehat{\delta s_f} = s_v - s_f \tag{6.10}$$

$$\widehat{\delta s_p} = s_p - s_v \tag{6.11}$$

$$\widehat{\delta r_f} = r_v - r_f \tag{6.12}$$

$$\widehat{\delta r_p} = r_v - r_p \tag{6.13}$$

where r and s are the assumed (directly by measurements or by an estimation process) lateral and longitudinal coordinates of the considered (v subscript), preceding (p) and following (f) vehicles.

Level 3 (top level): At the top of the hierarchy there is the feature of actual interest, ξ^{wl}, indicating to what extent the target is compatible with the Wrong Lane condition:

$$\xi^{wl} \sim \text{beta}\left[\kappa\left(\text{Exp}\left(\frac{\xi_{\delta r_p}\xi_{\delta r_f}\xi_{\delta s_p}\xi_{\delta s_f}}{\nu}\right) - \eta\right), \kappa\frac{1}{\text{Exp}\left(\frac{\xi_{\delta r_p}\xi_{\delta r_f}\xi_{\delta s_p}\xi_{\delta s_f}}{\nu}\right) - \eta}\right] \tag{6.14}$$

where $\eta \simeq 1$ is a parameter that allows shaping the PDF when the distribution mode is close to 0. The numerator of the exponential functions arguments , $\xi_{\delta r_p}\xi_{\delta r_f}\xi_{\delta s_p}\xi_{\delta s_f}$, is the product of the sub-features introduced at the previous level of the hierarchy. This is due to the fact that all the conditions represented by the sub-features are required to be verified for considering the vehicle on a wrong lane (high value for the feature ξ^{wl}). As soon as one of the $\xi_{\delta\#_*}$ approaches zero, indeed, $\alpha \to 0$ and $\beta \to \infty$, leading to a beta distribution completely shifted to the left. In Fig. 6.2 an example of the distribution for ξ^{wl} is depicted, when $\xi_{\delta r_p} = 1$, $\xi_{\delta r_f} = 0.9$, $\xi_{\delta s_p} = 0.8$, $\xi_{\delta s_f} = 0.7$ and $\eta = 0.9$.

Considering the definitions provided for the α and β parameters in Eqn. 6.14, note that when even a single $\xi_{\delta s*_\#}$ is close to zero (i.e. one of the four triggering conditions is not satisfied) the argument of the exponential tends to zero and thus the exponential tends to 1. In this case, the η parameter causes α and β to be "small" and "big",respectively, thus leading to a beta distribution shifted to the left (feature not recognised). In the absence of η , α and β would be very similar, leading to a centred distribution (mode close to 0.5): this, however, would not be consistent with the common-sense knowledge (if a single condition

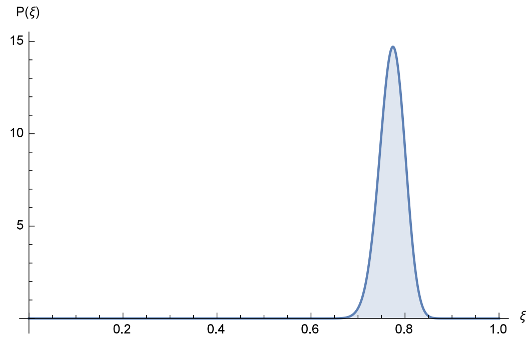


FIGURE 6.2: Distribution example for the Wrong Lane feature

on lateral/longitudinal distance is not met the Wrong Lane assertion should be considered false).

The actual α and β parameters for the beta distribution at this level must be defined on the basis of numerical values for the variables $\xi_{\delta r_p}$, $\xi_{\delta r_f}$, $\xi_{\delta s_p}$, $\xi_{\delta s_f}$. Since no real measurements are available for the sub-features, a single numerical value must be extracted from each of the known PDFs, which could be considered as 'pseudo-measurements'. For this purpose, possible solutions are, e.g., the mean value, the distribution mode, a single sample or the average of multiple samples.

6.3 Hypothesis Testing For Target Assessment

6.3.1 Decision Space and Test Policy

A hypothesis testing approach has been adopted for assessing whether or not the target should be considered as a possible threat, i.e. a TOI. Within this approach, two states for a target are considered: either it is a TOI (characterised by the value θ_1 for the decision parameter θ) or it is not (θ_0). In so doing, reference is made to a finite parameter space $\Theta = \Theta_0 \cup \Theta_1$, where $\Theta_0 = \{\theta_0\}$ and $\Theta_1 = \{\theta_1\}$. The 'null' and 'alternative' hypotheses can thus be formulated as following:

$$\begin{cases} H_0 : \theta = \theta_0 \\ H_1 : \theta = \theta_1 \end{cases}$$

that is, the target is recognised as a regular or suspicious (possible threat) driver, respectively.

The approach adopted for testing the two hypotheses is based on *the posterior odds ratio*. This can be considered as a likelihood ratio test accounting also for the prior probabilities of the two hypotheses. The priors can be leveraged to introduce a subjective bias in the decision process or for simply modelling any further knowledge affecting it. With reference to the Bayesian theory, the posterior odds ratio ρ^* represents the ratio between the odds given to the two hypotheses on the basis of the prior knowledge and the new information ([13]), and it is defined as

$$\rho^* = \rho \cdot B \tag{6.15}$$

where $B = f\left(x|\theta_0\right)/f\left(x|\theta_1\right)$ is known as *Bayes factor* and $\rho = \pi_0/\pi_1$ is the *prior odds ratio*, that is the ratio between the assumed prior probabilities for the two hypotheses. The considered test policy simply consists in checking whether the posterior odds ratio is greater or smaller than a given threshold (1 in the simplest case), thus suggesting the null or alternative hypothesis, respectively.

Given a decision space and a test policy, it is necessary to define the statistical models for the feature values associated with the two classes of vehicles; these models correspond to the likelihood functions for (pseudo)measurements taken from the target. The key problems to cope with within the outlined decision process are:

- Since an assessment based on "high-level" features is being performed, the hypotheses to be tested ('target is a TOI', 'target is ordinary')are "semantic" concepts that are not explicitly related to the measurements likelihood function;
- The likelihood functions for the two hypotheses do not exist since direct measurements about the features cannot be taken.

These issues are addressed in the following sections for different numbers of considered features.

6.3.2 Single Feature Case

This section describeds how the decision policy can be applied within the considered assessment framework.

In the cases where a direct measurement z about the feature ξ and simple[9] hypotheses are available, a well-established and optimal decision criterion would be the *Likelihood-Ratio Test* (LRT), [31, 178], defined as following:

$$\Lambda(z) = \frac{P(z|H_0)}{P(z|H_1)} > T \implies H_0. \tag{6.16}$$

In Eqn. 6.16, P is the probability function, $\Lambda(z)$ is referred as *likelihood ratio*, T is a threshold value that can be calibrated following various approaches ([1]), and the symbol $\implies$ denote that the indicated hypothesis is accepted. Noticing that the likelihood ratio

[9] That is $\Theta = \bigcup_i \Theta_i$, with $\Theta_i = \{\theta_i\}$, $\forall i$.

corresponds to the Bayes factor in the case of simple hypotheses, the posterior odds ratio test can be written in a very similar form respect to (6.16), that is:

$$\Lambda(z) > T' \frac{\pi_1}{\pi_0} \implies H_0 \tag{6.17}$$

where T' is a threshold value not necessarily equal to the one in (6.16).

In the considered case, however, the feature value is not measured (since it is an "abstract" concept expressed as an dimensionless coefficient) but described as a probability density function $f(\xi)$. The necessary steps for the implementation of the described assessment technique are then:

- To subjectively define reasonable likelihood functions for the feature values given the two considered hypotheses;
- To extend the test policy reported in (6.17) to cope with information about the feature expressed as a PDF instead of a single measurement.

For the first step, considering a generic single feature ξ, a threshold value γ can be defined, which separates the feature values that classify the target as a threat ($> \gamma$) from those characterising regular targets ($\leq \gamma$). The threshold value can be defined on the basis of the knowledge provided by human experts or by means of a learning process (in case the necessary information is available). It should be pointed out that the threshold values are coupled to the definitions adopted for the features PDF: only when these are defined it is possible to establish the thresholds on the basis of the available knowledge. Assuming equal likelihood[10] for all the feature values smaller or greater than the threshold, it is possible to define the following likelihood functions:

$$P_0(\xi) = P(\xi|\theta_0) \sim \mathcal{U}\,[0, \gamma] \tag{6.18}$$

$$P_1(\xi) = P(\xi|\theta_1) \sim \mathcal{U}\,[\gamma, 1] \tag{6.19}$$

where $\mathcal{U}\,[a, b]$ indicates a uniform probability distribution over the interval $[a, b]$. These likelihood functions are depicted in Fig. 6.3 for the case of $\gamma = 0.8$, together with an example for the feature PDF.

The second step has been addressed considering the PDF values as weights for the likelihood values and integrating over the feature space. The resulting ratio is taken into account for the definition of a test policy referred in the remainder of the thesis

[10] Uniform distributions with constant likelihood across the considered interval have been chosen since, on the basis of the available information, there is no reason for stating that, e.g., given the fact that the target is "regular" a feature value of 0.5 is more likely than 0.2. From a simplistic analysis of the problem, bell-type PDFs appears as not motivated, while sigmoidal functions could be appealing. The latter, however, are not commonly leveraged for expressing probabilities distributions.

as *Extended Likelihood-Ratio Test* (ELRT)[11]. The decision test in (6.17) can thus be written as

$$\begin{cases} H_0 & \text{if} \quad \frac{\int_0^1 P_0(\xi) f(\xi) d\xi}{\int_0^1 P_1(\xi) f(\xi) d\xi} > \frac{\pi_1}{\pi_0} \\ H_1 & \text{otherwise} \end{cases} \tag{6.20}$$

where the arbitrary threshold T' has been set to 1 for the sake of simplicity without affecting the generality of the method[12]. Considering the definitions provided for the likelihood functions, the *Left-Hand Side* (LHS) of the inequality in (6.20) can be written as:

$$\begin{aligned} \frac{\int_0^1 P_0(\xi) f(\xi) d\xi}{\int_0^1 P_1(\xi) f(\xi) d\xi} &= \frac{\int_0^\gamma \frac{1}{\gamma} f(\xi) d\xi}{\int_\gamma^1 \frac{1}{1-\gamma} f(\xi) d\xi} \\ &= \frac{\frac{1}{\gamma} F(\gamma)}{\frac{1}{1-\gamma}\left(1 - F(\gamma)\right)} \end{aligned} \tag{6.21}$$

thus leading to the following test inequality:

$$F(\gamma) > \frac{\pi_1 \gamma}{(1-\gamma)\,\pi_0 + \gamma \pi_1} \implies H_0 \tag{6.22}$$

where $F(\gamma)$ is the *Cumulative Distribution Function* (CDF) for the random variable Ξ at the threshold value γ: $\int_0^\gamma f(\xi)\, d\xi$.

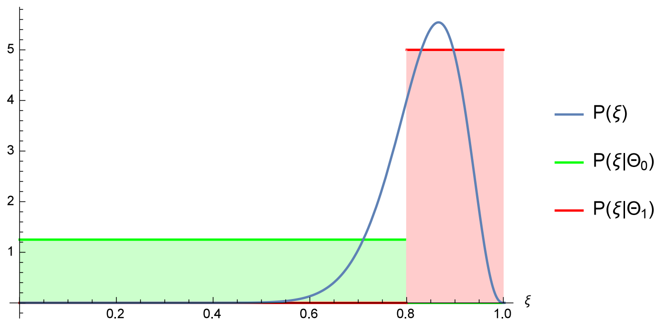
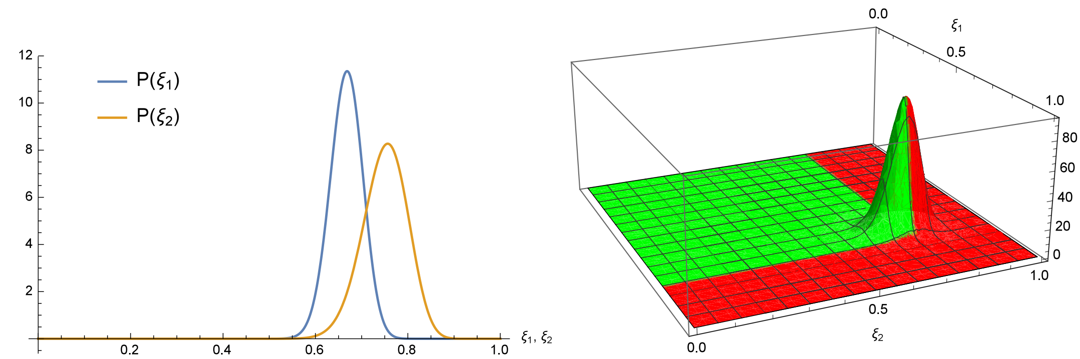


FIGURE 6.3: Uniform likelihood functions and feature distribution

[11] To be noticed that the ELR has the same form of the the Bayes factor that would result from considering a continuous decision space ($\Theta = [0, 1]$) and the feature distribution as prior information.

[12] Different values for T' can be enforced by simple multiplication of the final test threshold.



(a) Single feature distributions (b) Joint distribution for the features

FIGURE 6.4: Example distributions for the two-features case

6.3.3 Two Features Case

The approach outlined in Section 6.3.2 can be easily extended to account for two features. In the two-features case, the feature space is a 2-dimensional surface $\Psi = \left\{\left(\xi^1, \xi^2\right) \mid 0 \leq \xi^1, \xi^2 \leq 1\right\}$ divided into two sub-regions: $\Psi_0 = \left\{\left(\xi^1, \xi^2\right) \mid \xi^1 < \gamma_1, \xi^2 < \gamma_2\right\}$ and $\Psi_1 = \Psi - \Psi_0$ (Fig. 6.4b). Given two features, their conditioned[13] probabilities can be reasonably considered independent, while the joint likelihood function cannot be considered as the simple product of the two previously defined likelihoods. The joint likelihood functions for the two hypotheses are again members of the uniform distributions family, and can be defined as follows:

$$P(\xi^1, \xi^2|H_0) = P_0(\xi^1, \xi^2) = \mathcal{U}\left[\{0, \gamma_1\}, \{0, \gamma_2\}\right] = \phi_0 \tag{6.23}$$

$$\begin{aligned} P(\xi^1, \xi^2|H_1) &= P_1(\xi^1, \xi^2) \\ &= \mathcal{U}\left[\{\{\gamma_1, 1\}, \{0, 1\}\} \cup \{\{0, \gamma_1\}, \{\gamma_2, 1\}\}\right] \\ &= \phi_1 \end{aligned} \tag{6.24}$$

where

$$\phi_0 = \frac{1}{\gamma_1 \gamma_2} \tag{6.25}$$

$$\phi_1 = \frac{1}{(1 - \gamma_1) + \gamma_1 (1 - \gamma_2)} \tag{6.26}$$

The extended likelihood ratio over the joint probability for the features is calculated in (6.27).

[13]Conditioned respect to the information provided from measurements, $f\left(\xi|Z\right)$. This conditioning has not been made explicit in the features PDFs definition for the sake of notational simplicity, but is obviously present.

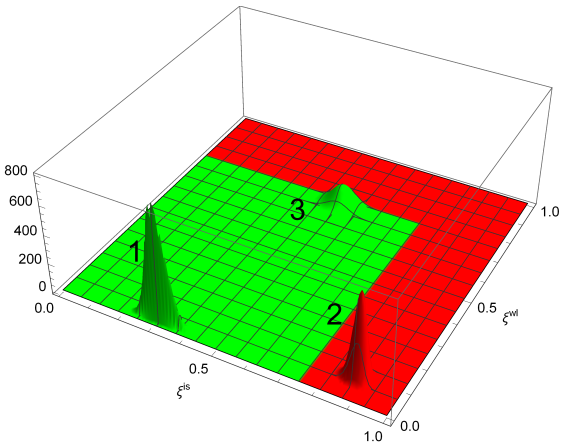


FIGURE 6.5: Joint PDF examples associated with three different targets

$$\frac{\iint\limits_{\boldsymbol{\xi}} P_0\left(\xi^1,\xi^2\right) f\left(\xi^1,\xi^2\right)\, d\boldsymbol{\xi}}{\iint\limits_{\boldsymbol{\xi}} P_1\left(\xi^1,\xi^2\right) f\left(\xi^1,\xi^2\right)\, d\boldsymbol{\xi}} = \frac{\phi_0 \int_0^{\gamma_1}\int_0^{\gamma_2} f\left(\xi^1\right) f\left(\xi^2\right)\, d\xi^2 d\xi^1}{\phi_1\left[\int_0^1\int_{\gamma_2}^1 f\left(\xi^1\right) f\left(\xi^2\right)\, d\xi^2 d\xi^1 + \int_{\gamma_1}^1\int_0^{\gamma_2} f\left(\xi^1\right) f\left(\xi^2\right)\, d\xi^2 d\xi^1\right]}$$
$$= \frac{\phi_0 F\left(\gamma_1\right) F\left(\gamma_2\right)}{\phi_1\left[\int_0^1 f\left(\xi^1\right)\left(1-F\left(\gamma_2\right)\right)\, d\xi^1 + F\left(\gamma_2\right)\int_{\gamma_1}^1 f\left(\xi^1\right)\, d\xi^1\right]}$$
$$= \frac{\phi_0 F\left(\gamma_1\right) F\left(\gamma_2\right)}{\phi_1\left[\left(1-F\left(\gamma_1\right)\right) + \left(1-F\left(\gamma_2\right)\right) F\left(\gamma_1\right)\right]} \tag{6.27}$$

$$\Delta = \frac{\int\cdots\int_{\boldsymbol{\xi}} P_0\left(\boldsymbol{\xi}\right) f\left(\boldsymbol{\xi}\right)\, d\xi^1\ldots d\xi^n}{\int\cdots\int_{\boldsymbol{\xi}} P_1\left(\boldsymbol{\xi}\right) f\left(\boldsymbol{\xi}\right)\, d\xi^1\ldots d\xi^n}$$
$$= \frac{\phi_0 \int_{\xi^1=0}^{\gamma_1}\int_{\xi^2=0}^{\gamma_2}\cdots\int_{\xi^n=0}^{\gamma_n}\prod_{i=1}^{n} f\left(\xi^i\right)\, d\xi^1\ldots d\xi^n}{\phi_1\left[\int_{\gamma_1}^1\int_0^1\cdots\int_0^1 f\left(\boldsymbol{\xi}\right)\, d\boldsymbol{\xi} + \int_0^{\gamma_1}\int_{\gamma_2}^1\cdots\int_0^1 f\left(\boldsymbol{\xi}\right)\, d\boldsymbol{\xi} + \cdots + \int_0^{\gamma_1}\int_0^{\gamma_2}\cdots\int_{\gamma_n}^1 f\left(\boldsymbol{\xi}\right)\, d\boldsymbol{\xi}\right]}$$
$$= \frac{\phi_0 \prod_{i=1}^{n} F\left(\gamma_i\right)}{\phi_1 \sum_{i=1}^{n}\left[\left(1-F\left(\gamma_i\right)\right)\prod_{j=1}^{i-1} F\left(\gamma_j\right)\right]} = \frac{\phi_0 \prod_{i=1}^{n} F\left(\gamma_i\right)}{\phi_1\left(1-\prod_{j=1}^{n} F\left(\gamma_i\right)\right)} \tag{6.28}$$

The decision test can thus be written as:

$$\frac{F\left(\gamma_1\right) F\left(\gamma_2\right)}{1-F\left(\gamma_1\right) F\left(\gamma_2\right)} > \frac{\pi_1\gamma_1\gamma_2}{\pi_0\left(1-\gamma_1\gamma_2\right)} \tag{6.29}$$

Consider, for example, the case of the feature distributions depicted in Fig. 6.4, with

$\gamma_1 = \gamma_2 = 0.75$ and prior for the null hypothesis ("ordinary" target) $\pi_0 = 0.75$. Under these conditions the test threshold for the decision-making process is 0.43, while the *Right-Hand Side* (RHS) term of the test is 0.91; on the basis of the assessment policy previously described the target is then classified as ordinary.

Fig. 6.5 shows three examples of joint probability distributions for the two features introduced in 6.2.2.2. These are relative to the state of three target vehicles at a given time instant, simulated by means of the simulation environment described in Section 6.4.1.1. Distribution 1 is relative to a regular target, moving at 5.2 km/h below the speed limit and positioned on the first lane on the left. Distribution 2 represents a vehicle moving on the same lane but exceeding the speed limit by 8 km/h. Finally, the third distribution is relative to a vehicle moving at the limit speed on the overtaking lane: the distance from the closest vehicle behind the target is 600 m, while the distance from the preceding vehicle is 120 m, which corresponds to a reasonable stopping distance. The target could be thus considered in the beginning of an overtaking manoeuvre, or waiting to achieve a greater distance from the car in the front before to move on the left-side lane. For these reasons the Wrong Lane feature assumes a medium-high value, yet not leading to a TOI assessment. For the three cases presented, the decision test described in this section classifies as TOI only target 2.

6.3.4 Generic Multiple Feature Case

The posterior odds ratio test described by Inequality (6.29) can be easily extended to the generic multi-variable case, where a set of n features $\boldsymbol{\xi} = \{\xi^1, \xi^2, \ldots, \xi^n\}$ is considered. This generalisation should be carried out bearing in mind the two main conditions previously introduced:

- The PDFs for the features values (conditioned to the measurements)are independent
$$f(\boldsymbol{\xi}) = \prod_{i=1}^{n} f(\xi^i)$$
- For each feature a threshold value γ_i is defined, distinguishing the allowed values from those indicating a possible threat. The multi-dimensional feature space can thus be divided into two disjoint sets Ψ_0 and Ψ_1 such that
$$\tilde{\boldsymbol{\xi}} \in \Psi_1 \Leftrightarrow \exists\, i : \tilde{\xi}^i > \gamma_i.$$

Under these assumptions the values for the uniform likelihood functions result:

$$\phi_0^{-1} = \prod_{i=0}^{n} \gamma_i \tag{6.30}$$

$$\phi_1^{-1} = \sum_{i=1}^{n} \left[(1-\gamma_i) \prod_{j=1}^{i-1} \gamma_j \right] = 1 - \prod_{i=1}^{n} \gamma_i \tag{6.31}$$

The ELR for the multiple-feature case is obtained in (6.28)[14]. The decision test can thus be finally written as :

$$\frac{\prod_{i=1}^{n} F(\gamma_i)}{1 - \prod_{j=1}^{n} F(\gamma_i)} > \frac{\pi_1 \prod_{i=0}^{n} \gamma_i}{\pi_0 \left(1 - \prod_{j=1}^{n} \gamma_j\right)} \implies H_0 \tag{6.32}$$

6.3.5 Context Information Influence

The influence of the information about the contextual aspects affecting the targets behaviour on the decision-making process can be modelled through the prior probabilities for the two hypotheses. A similar approach has been described by Blacknell in [16], but with reference to a standard likelihood ratio test based on actual measurements.

Considering the ELRT-based decision approach previously described, context information (e.g. traffic congestion) suggesting an increased alert state would lead to a modification in the prior probability for the 'TOI' hypothesis, that is:

$$\pi_1' = \pi_1 + \zeta\Lambda \tag{6.33}$$

$$\pi_0' = 1 - \pi_1' \tag{6.34}$$

where the parameter ζ indicates the maximum modification accepted for the prior based on the context, while the parameter $\Lambda = g(\boldsymbol{\lambda})$ is function of the context information considered. Possible implementations for the g function are the product or the maximum, depending on the desired level of conservativeness for the taken approach. Considering the numerical example presented in Section 6.3.3, and assuming $\pi_0 = 0.75$ and $\pi_1 = 0.25$, the influence of the context information on the decision-making process is shown in Fig. 6.6. This depicts the space of all the possible[15] couples (ζ, Δ) and the boundary between those leading to a classification as 'regular target' and those that classify the target as TOI (blue and white regions, respectively).

[14] In this equation and (6.31) holds $\prod_{j=1}^{0} g(i) = 1$ and $F(\gamma_i) = \int_0^{\gamma_i} f\left(\xi^i\right) d\xi^i$.

[15] Parameter ζ is limited to 0.74 by design.

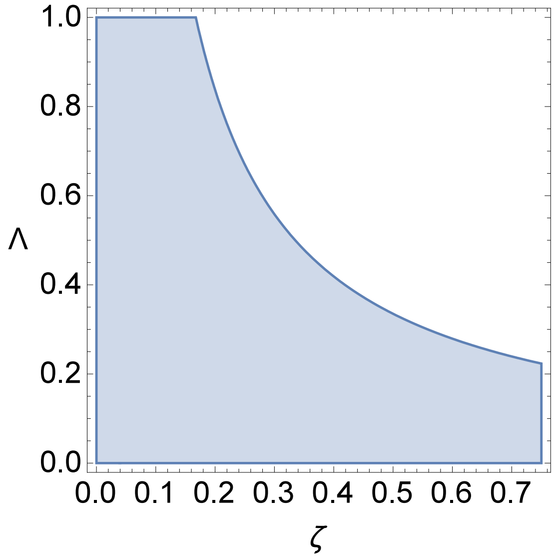


FIGURE 6.6: Effect of the context-related parameters on the target classification for a fixed feature value: switching from 'regular' assessment (blue area) to TOI (white area)

For the road-monitoring application considered, an example of relevant context information could be the traffic condition λ^t, defined as:

$$\lambda^t = \frac{\sum_{i=1}^{n} \lambda_i^t}{n} \tag{6.35}$$

where

$$\lambda_i^t = \frac{\sum_{j \in w_i} \frac{1}{d_{i,j}}}{\tilde{\lambda}} \tag{6.36}$$

$$\tilde{\lambda} = 2 \sum_{k=1}^{\bar{n}} \frac{1}{k\bar{d}} \tag{6.37}$$

$$\bar{n} = \frac{w}{\bar{d}} \tag{6.38}$$

In the previous equations, n is the number of monitored targets, w is the assumed dimension for the distance window within which the presence of other vehicles affects the conduct of a driver, w_i denotes the road section beginning at the position of target i with extension w, $\bar{d}$ is the minimum assumed distance among vehicles (still permitting a continuous traffic flow). Then, $\tilde{\lambda}$ is the sum of the inverse distances between a vehicle and all the other vehicles within its influence window in the worst congestion condition, while λ_i^t represents the same quantity for the actual vehicle i normalised by $\tilde{\lambda}$. The final traffic index λ^t is then obtained as simple average of the local indexes λ_i^t.

6.4 Validation

In this section, results are presented from an extensive set of simulations. These have been performed in order to validate the effectiveness of the proposed assessment method and its sensitivity to some of the various tuning parameters involved. Both synthetic and real-world data have been taken into account within the simulations: this has allowed testing the assessment framework under sensibly different conditions and thus for obtaining more general indications about its performances.

6.4.1 Application to Synthetic Data

6.4.1.1 Simulations Description

In the synthetic data case, the vehicles trajectories are produced by means of the traffic simulator described in Section 4.3.1. A section of a three-lane highway with a fixed number of cars moving along is considered, where the vehicles can perform overtaking manoeuvres, switch among the three lanes, accelerate, decelerate or keep constant speed. The total number of vehicles moving along the road is 50, partitioned among the following types[16] with the following probabilities:

1. Regular driver (60%): maintains a speed below the limit and behave correctly respect to the overtaking policy, the stopping distance and the lane occupation;
2. Cowboy driver (10%): can travel at a speed up to the limit, overtakes on both the side and can occupy the second and third lane even if not overtaking;
3. Snail driver (10%): similar to the type 1 driver, but maintains a speed significantly under the limit;
4. Cautious driver (10%): similar to the type 1 driver but keeps a greater stopping distance;
5. Racer driver (10%): can overtake on both the sides and is capable of travelling at speed higher than the limit;

The monitoring objective is to identify vehicles of classes 2 and 5 since they represent possible threats for the traffic. A GMTI-equipped UAV is assumed to monitor the road section, taking position and velocity measurements about the targets. The noise on these readings is assumed to be Gaussian with variance σ_p^2 and σ_v^2, respectively. The tracking task is then performed by a classic KF. It is assumed that data-to-track and

[16] The drivers characterisation is here repeated for convenience.

track-to-track association processes are performed perfectly as they are not the main focus of this study.

On the basis of the targets tracks, the assessment technique described in Section 6.3 is applied, taking into consideration the two features introduced in Section 6.2.2: Irregular Speed and Wrong Lane. The hypothesis testing approach adopted indicates whether the target should be considered a TOI with respect to just the last target state, and thus to a single time step; in the following, this event is referred to as *Target Of Interest Warning* (TOIW).

The final decision about an individual vehicle, that is to classify it as TOI or regular target, can be taken considering a threshold Γ on the number of TOIW generated by that specific vehicle. This approach has the twofold purpose of:

- Introducing some flexibility in the assessment process: one or more TOIWs can be issued to vehicles that do not represent an actual threat to the traffic flow. Consider, for example, a target that exceeds the speed limit just for a very short time interval, e.g. during an overtaking manoeuvre;
- Accounting for the common-sense, imperfect knowledge leveraged: as previously mentioned the hypothesis testing process is based on a subjective interpretation of what is "correct" and what is "wrong", but this is not necessarily the only reasonable interpretation. For example, with reference to the specific implementation proposed, a type-4 ('cautious') driver tends to maintain large distances from the preceding vehicle, and could thus start an overtaking manoeuvre with great advance: it could thus move to the overtaking lane when the distance from the preceding vehicle is so big that the assessment process (tuned on what are considered as common practises) generates a warning due to Wrong Lane condition.

6.4.1.2 Ground Truth Definition

The lack of a physical, indisputable GT against which to compare the assessment results provided by the monitoring technique is a relevant problem that has been borne in mind since the very beginning of the thesis (Section 1.3). The approach adopted in the simulations here reported is to define the GT on the basis of the actual simulation implementation rather than on the behaviours exhibited during the simulation itself. This means that if a target is capable, by its type definition, to exhibit concerning behaviours it is considered a TOI within the GT regardless whether these behaviours are manifested or not during the simulation. Such approach is implemented by comparing the number of the vehicles recognised as TOI with the number of class 2 and class 5 vehicles created by the simulator, regardless of the actual simulation evolution.

This approach clearly represents the simplest solution from a practical point of view, but the reasons for this choice involve also more subtle considerations. Even if there is no apparent reason to classify a target as a threat until it actually performs a concerning manoeuvre, it should be noted that it could be meaningful to implement a "predictive" GT where a TOI is defined as a vehicle with the "potential" to be a threat. This approach could be particularly suitable to critical contexts (e.g., counter-terrorism), where the threat manifestation and the actual damage are almost simultaneous, and thus preemptive actions are required. A further reason to adopt this "predictive" GT is that the use of the "non-predictive" type would limit the performance of the automated assessment system to that of a human operator, which indeed can classify a vehicle as TOI just after that it has exhibited some alarming behaviour. Consider the case where an automated assessment policy would be capable of identifying[17] a TOI before it explicitly manifests (from an human-operator point of view) its character: with a non-predictive GT it would be instead classified as false alarm, thus unreasonably bounding the assessment performances.

Because of the characteristics so far described for the assumed GT, the numerical results provided in the following may not be considered "absolute", but depending on a personal interpretation of the problem. This particular interpretation is reflected in the γ_i thresholds and other features parameters and affects the final detection values.

6.4.1.3 Simulation Results

Simulation results for the scenario described in 6.4.1.1 are here reported, aiming at testing the framework capabilities under different simulation conditions. The results indicate the number of 'correct identifications' (type 2 and 5 vehicles classified as TOI) or 'false alarms' (vehicles of different types indicated as TOI) as a function of the Γ adopted for the final decision criteria. In all the reported plots, the values on the x axis (threshold Γ) have been normalised respect to the number of the simulation times steps[18] and thus lie in the range $[0, 1]$.

With this approach, it is possible to i) Easily understand the influence of the threshold Γ on the technique performances ii) Identify the most suitable threshold value for actual on-line implementations. In most of the cases, indeed, this threshold requires to be defined by means of a trade-off process between correct detections and false alarms constraints. Bearing in mind the supporting (not autonomous) role of the proposed framework, the need for modifying the threshold Γ could be signalled by a human operator in actual implementations. This could be the case, e.g., where the amount of false alarms would be higher than what expected from the system design.

[17] For example by considering multiple features at the same time.

[18] That is twice the duration of the single iteration expressed in seconds.

The simulations considered in the following are aimed at testing the assessment performances when the key parameters for the simulation and the decision policy change. The tests have been implemented by means of a Monte-Carlo approach, that is by averaging the results from a high number of independent simulation iterations; in so doing, the results are conferred with statistical relevance.

Preliminary Tests Two preliminary tests have been performed to verify the validity of the conditions adopted for the simulations that are presented in the remainder of the chapter and for ensuring the generality of the numerical results reported. As previously stated, an actual implementation of the proposed approach requires to choose a method for selecting a single numerical value from the lower-level distributions. For the simulations reported in the following, this choice has been done by comparing the outcomes provided by three different methods: distribution mode value, distribution mean value and random sampling from the distribution. The results depicted in Fig. 6.7 are obtained by averaging over 100 simulations with duration 10 mins each. These results suggest that mean and mode operators perform similarly, while an unacceptably low ratio of correct detections is obtained when random sampling is adopted. For these reasons, for the evaluation of the Irregular Lane feature, the mode operator has been adopted for inferring the level 3 PDF from the level 2 PDFs .

Then the cases of two different distributions for the vehicle types have been tested and compared:

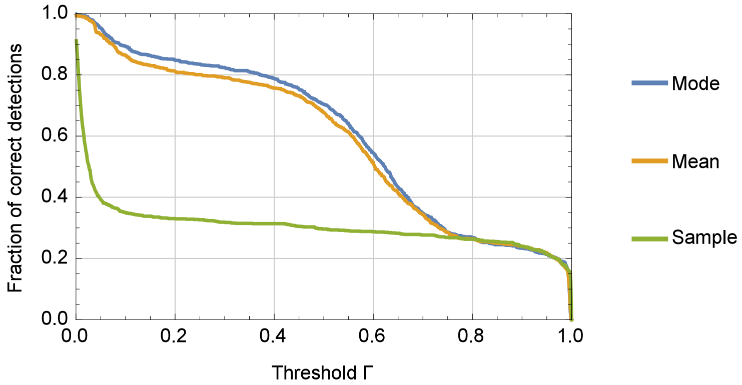
Case 1: It is the one described in Section 6.4.1.1 and adopted in all the following simulations: the vehicles capable of behaviours of interests are the 20% of the total.

Case 2: The following drivers distribution is adopted:

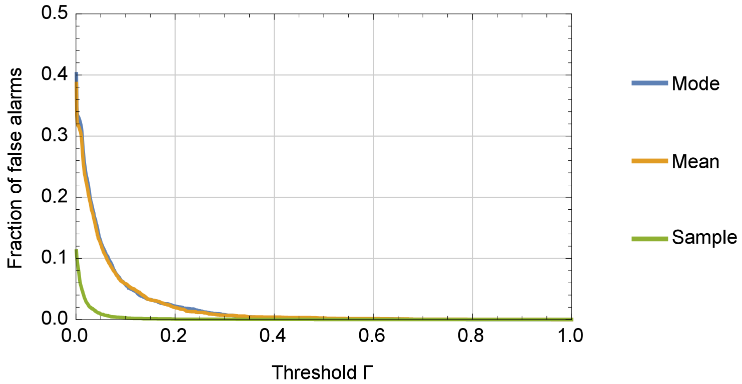
1. Regular drivers (30%);
2. Cowboy drivers (25%);
3. Snail drivers (10%);
4. Cautios drivers (10%);
5. Racer drivers (25%);

where the vehicles capable of behaviours of interests are the 50% of the total.

The differences in the assessment performances for the two cases are depicted in Fig. 6.8 and appear to be negligible since below 1/1000 over a $[0, 1]$ range. This result suggests that the simulation outcomes presented in the following sections can be considered valid for generic distributions of vehicle types and not only for the one adopted for the simulation.



(a) Correct detections



(b) False alarms

FIGURE 6.7: Results for Monte-Carlo simulations adopting different aggregation methods

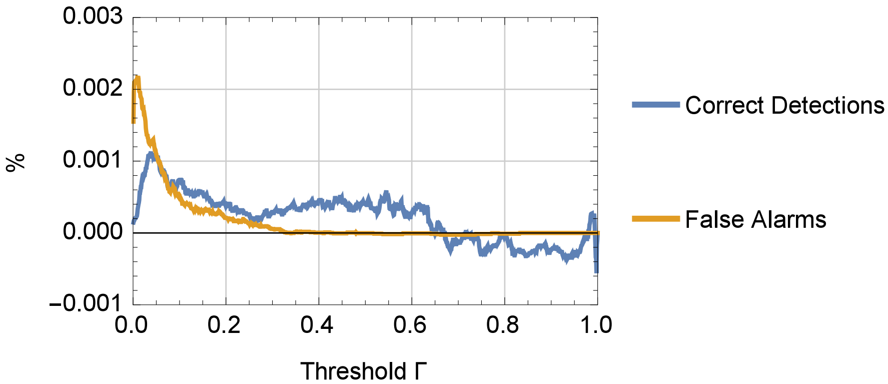
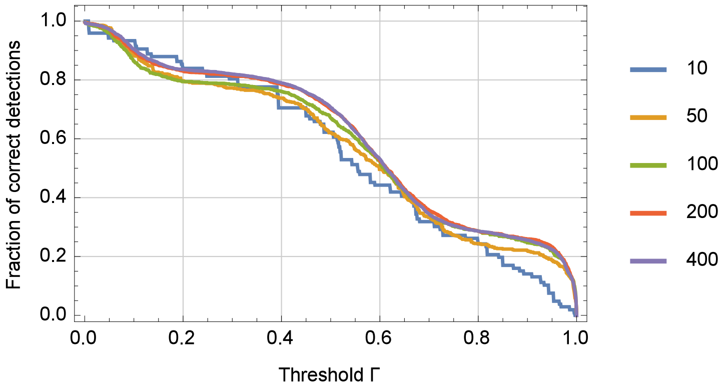


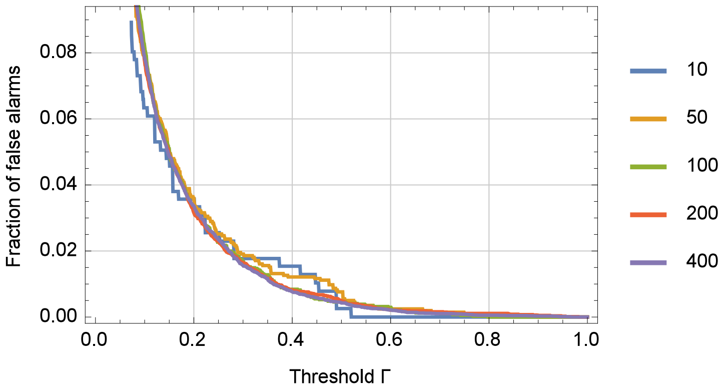
FIGURE 6.8: Performances comparison for two different distribution of vehicle types

Simulation Iterations Sensitivity The purpose of this test is to find out how many Monte-Carlo simulations appear to be necessary for having statistically relevant results. Simulation parameters adopted are:

- Simulation duration: 10 mins;
- Null hypothesis prior: 0.5;
- Position error standard deviation: 0.3 m;



(a) Correct detections



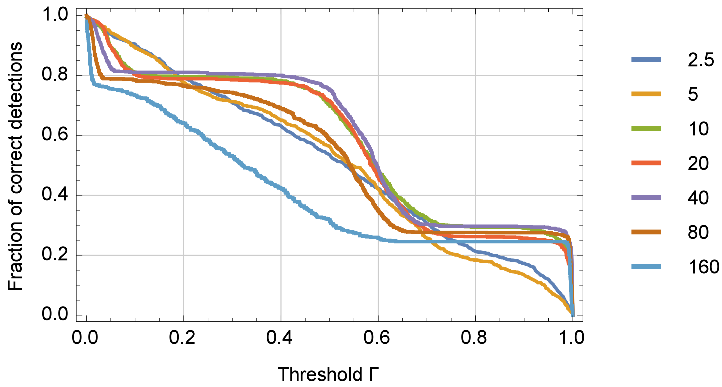
(b) False alarms

Figure 6.9: Results for Monte-Carlo simulations with different number of iterations

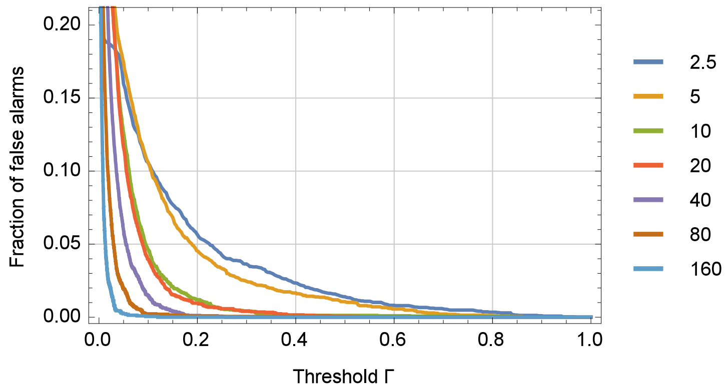
The results depicted in Fig. 6.9 are based on up to 400 simulations of 10 minutes each, thus considering 66.6 hours (2.7 days) of monitoring data. These simulations clearly indicate a trend for increasing numbers of iterations, showing that the differences for values greater than 100 are negligible. For this reason, all the following sensitivity analyses has been performed on the basis of 100 Monte-Carlo iterations, thus relying[19] on 16 hours of monitoring data. This approach is likely more relevant than a single simulation of equivalent duration, since each simulation is initialized with different random traffic conditions.

Simulation Duration Sensitivity This test is aimed at describing the relationship between the target identification ratio and the length of monitoring interval. Since a single monitoring interval is considered, its length corresponds with that of the whole simulation. Other simulation parameters are:

- Simulation iterations: 100;
- Null hypothesis prior: 0.5;
- Position error standard deviation: 0.3 m;



(a) Correct detections



(b) False alarms

FIGURE 6.10: Monte-Carlo simulations with different iteration duration ([mins])

With reference to correct detections (Fig. 6.10a) the responses for durations below 40 mins converge to a well-shaped curve exhibiting two plateaus in the intervals $[0.1, 0.5]$ and $[0.7, 1]$. By further increasing the simulation duration, the first plateau is replaced by an almost constant slope (evident in the 160 mins duration case). Results for false alarms are depicted in Fig. 6.10b and appear fairly straightforward: as the monitoring time increases the threshold Γ required for reducing the fraction of false alarms under a given level decreases.

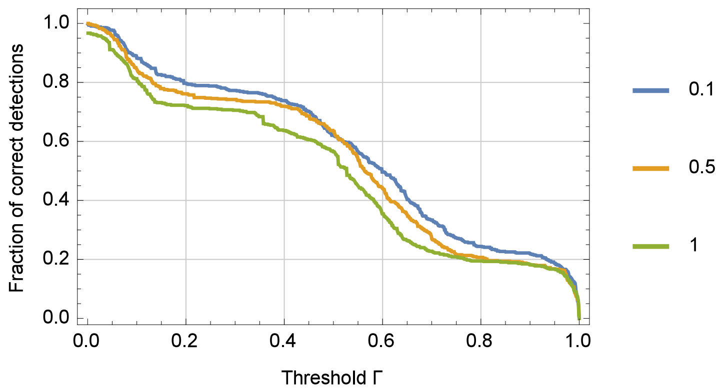
A conclusion that could be drawn from these results is that it appears convenient to adopt a monitoring interval duration such that the first plateau on the characteristic curve is still present (in the proposed case, 40, 20 or even 10 minutes), and then impose a Γ corresponding with the right extreme of such plateau. In so doing the ratio of correct detection is maintained high, while false alarms are drastically reduced. Obviously, the actual duration of the monitoring interval could result from the trade-off with further specification, e.g., the need for quick detection and intervention.

Prior Sensitivity The prior probabilities for the null and alternative hypotheses allow making the TOI identification less or more tolerant. As indicated in Section 6.3.5

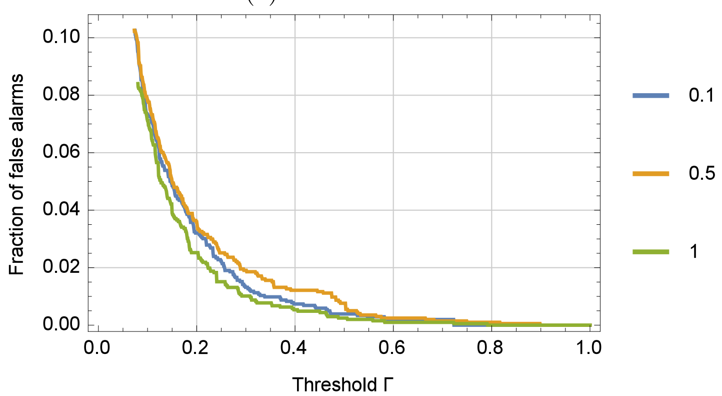
[19]Considering 10 minutes as the duration of the single iteration, which is almost always the case.

these probabilities can be leveraged to account for context information, and thus to make the assessment process dependent on the environment conditions affecting the vehicles. This test allows analysing the effect of the prior information by considering the cases where it strongly rejects, strongly suggests or gives the same odds to the null hypothesis respect to the alternative one. Other simulation parameters adopted are:

- Simulation iterations: 100;
- Simulation duration: 10 mins;
- Position measurement error standard deviation: 0.3 m;



(a) Correct detections



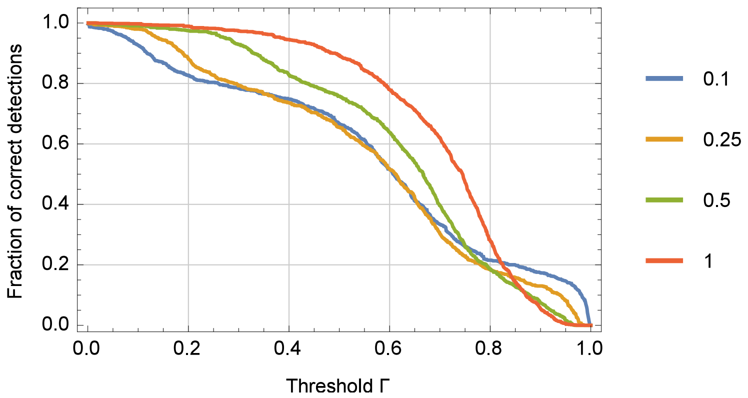
(b) False alarms

Figure 6.11: Monte-Carlo simulations with different prior probabilities for the null hypothesis

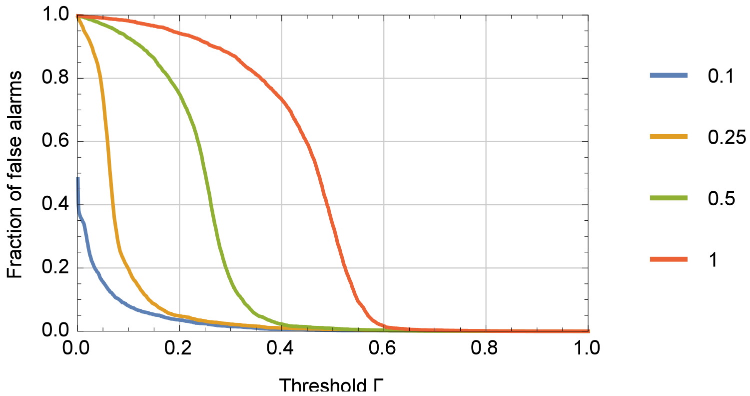
As expected, lower priors for the null hypothesis lead to higher correct detection rates (Fig. 6.11a): the curve associated with a given prior is almost always above those related to higher priors for the null hypothesis. Similar trends can be noticed for the false alarms (Fig. 6.11b). Actually, in this case, the curve for $\pi_0 = 0.5$ does not look exactly how expected within the range (0.2,0.5) since it should be lower than the one associated with $\pi_0 = 0.1$; the differences in the three characteristics, however, are so small ($< 1\%$) that they can probably be influenced by numerical aspects.

Measurement Error Sensitivity As indicated in Section 6.2.2 the information about the measurement error covariance is leveraged for qualitatively tuning the PDFs selectiveness for features affected by that specific measurement. The influence of such aspect on the final identification results is here analysed by varying the position measurement error covariance $\sigma = \sigma_r = \sigma_s$ within the range $[0.1, 1]$. Other simulation parameters are:

- Simulation iterations: 100;
- Simulation duration: 10 mins;
- Null hypothesis prior: 0.5;



(a) Correct detections



(b) False alarms

FIGURE 6.12: Monte-Carlo simulations with different standard deviations for the position measurement error ([m])

Fig. 6.12 shows that the bigger is the covariance, the higher are the identification curves, both for correct and erroneous detections. Note, however, that a relatively small increment in the correct identification ratio corresponds to a substantial raise in false alarms. For this reason, it is possible to conclude that, as expected, bigger uncertainties lead to overall worst assessment performances. From these results, it is clear that when bigger covariances are considered, the assessment framework is more keen to classify the vehicles as TOI, probably because of the minor selectiveness of the feature PDFs and the fact that noisy measurements tends to collocate the vehicles on the overtaking lanes even if they are on the first lane.

In this case, the trade-off for the definition of the Γ relies heavily on the system requirements. If it is requested to identify the majority of the targets of interest, a medium-value threshold should be adopted (between 0.3 and 0.5)[20]. On the other hand, if it is more important to not overwhelm the operator with a huge number of false alarms, Γ should be chosen so that the false detections ratio falls at least under 0.5 (between 0.5 and 0.6), inevitably leading to a reduced amount of correct detections.

6.4.2 Application to Real Data

In this section, the proposed assessment approach has been applied to data collected from a real-world scenario. For the simulation here presented 5200 trajectories provided by the NGSIM project (Section 4.3.2) have been taken into consideration. These tracks belong to the US101 dataset and take place in the period of time between 7.50am and 8.35am. Before undertaking the assessment process, the original NGSIM data have been corrected by applying the technique described by Montanino et al. in [91], which allow removing outliers and unfeasible acceleration profiles due to measurement errors.

Similarly to the synthetic data case, also for the NGSIM data no GT is available about the behaviours of interest (e.g., to occupy an overtaking lane for no good reason). Furthermore, the target trajectories are not generated within a controlled environment anymore[21]. This implies that for the real-world case is not possible to distinguish between correct detections and false alarms. What is possible to do, however, is to check whether the overall detections amount generated in the case of NGSIM data follows the trends seen for the synthetic data case or not. More specifically, the aim of the test is to compare the actual detections obtained by assessing the NGSIM trajectories with the number of detections expected from the percentages indicated, e.g., in Fig. 6.9 (400-iterations case) applied to the new number of vehicles. The expected detections amount is calculated as:

$$d_e = n_v \left(0.2\varphi p_c + (1 - 0.2\varphi)\, p_f \right) \tag{6.39}$$

where p_c and p_f are the percentages, as a function of Γ[22], for correct detections and false alarms, respectively, and φ is an adaptation coefficient associated with the fraction of vehicles representing a TOI. The resulting d_e is the number of expected detections

[20] With specific reference to the case of $\sigma = 1$.

[21] The fact that, in the previous case, the data was generated within a controlled environment has allowed defining a sort of "expected" GT, which was based on how the vehicles are expected to behave.

[22] Similarly to the case of synthetic data, Γ is expressed as fraction of the overall monitoring time for the vehicle.

expressed as a function of the threshold Γ adopted within the assessment process. The use of the coefficient φ is due to the fact that, within the simulator of Section 6.4.1.1, it has been arbitrarily decided for the 20% of the vehicles to be possible TOIs: this, however, does not necessarily hold for the real-world data. By simple trials, it has been found out that the trend for actual detections become very similar to that of the expected detections for $\varphi = 0.025$, as depicted in Fig. 6.13.

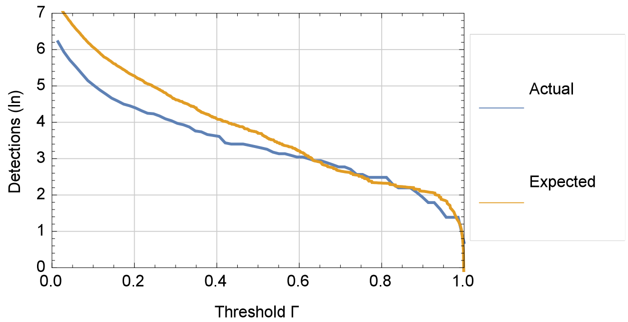


FIGURE 6.13: Detection results for NGSIM data

The existence of a difference between actual and expected detections, especially for low values of the threshold Γ, does not represent an actual problem since the purpose of this analysis is not to validate the traffic simulator on the basis of real data. What is important to note, however, is that the detections trends obtained by varying the threshold are similar for the two cases. This suggests that the trends for correct detections and false alarms reported in Section 6.4.1.3 can be considered valid, to some extent, also for real traffic data. The actual detection percentages could be significant as well for the case of NGSIM trajectories, but this cannot be stated for sure without knowing the underlying ground truth. Nonetheless, the proven similarity in the number and trend of detections generated by the analysis of real data and those expected from the results obtained for synthetic data allows having a qualitative indication of the technique performances also for actual implementations. The assessment performances varying Γ for this last case are indeed expected to follow similar trends to those reported in Section 6.4.1.3.

6.5 Conclusion

In this chapter the behaviour assessment problem has been considered, with the aim of identifying possible TOIs. These are targets exhibiting peculiar characteristics and could represent a threat to other vehicles. The proposed assessment approach is based on the concept of 'features', which are described by means of a probabilistic framework. Decisions about the most likely classification for the targets, i.e. regular target or TOI, are lastly taken utilising a hypothesis testing technique.

The main contribution here provided is the definition of a novel general approach to target assessment and TOI identification. This consists of two main components: i) a hierarchy of probability distributions representing high-level, not directly-measurable aspects of the target behaviour ii) a decision test specifically designed for dealing with the considered type of features and denoted as ELRT. The proposed approach has been applied to a road-monitoring scenario, but it seems to be extremely general since no specific model (mathematical, kinetic, etc ...) is leveraged within the assessment process. This should allow applying the proposed method also to completely different contexts, such as maritime environments or for human activities monitoring. Furthermore, the proposed approach appears to be highly flexible since it allows: (i) representing arbitrarily complex behavioural characteristics, (ii) reproducing both rigid and permissive assessment processes by tuning the feature PDFs parameters and the hypotheses prior probabilities and (iii) adapting to different conditions for the monitored scenario by accounting for context information.

The main issues that have arisen during the proposed work deal with the analytical modelling of a subjective assessment process. This has required to make assumptions about what should be considered "regular" and what "suspicious" on the basis of qualitative available knowledge. For this reason, and for the fact that a single, subjective interpretation of the reality is being considered, it appears hard to claim an "absolute" validity for the identification results provided by any type of assessment method and thus for its performances. This is also the case for the proposed assessment framework. Nonetheless, the results obtained can still be compared, and thus somehowvalidated, with a definition of ground truth based on the simulation implementation: this definition is maybe not above criticism, yet seems reasonable as it is based, to a large extent, on the actual simulation implementation. Under these assumptions, the results presented in Section 6.4 demonstrate the effectiveness of the proposed method, achieving, under specific simulation conditions, the 80% of correct detections with almost no false alarms. The application to real-world data has also been taken into account, suggesting that the numerical results obtained for synthetic data can reasonably be generalised to actual targets trajectories. Useful indications for an actual implementation are also provided from the performances analysis for different values of the threshold Γ on the number of warning events.

Chapter 7

Behaviour Monitoring Using Regular Expressions Based Pattern Matching

7.1 Introduction

In this chapter, the problem of identifying "possible threats" or, more generally, TOIs is tackled by considering the time-correlated behaviour of targets.

As mentioned in Section 2.2, a vehicle is considered "misbehaving" when it does not comply with a given definition of "normality", which is related to either pieces of knowledge provided by domain experts or inferred from a set of data constituting the available information. Given this twofold characterisation provided for the concept of "regularity", two main different approaches have been considered in literature for tackling the anomaly detection problem: the detection of particular patterns of interest in the behaviour exhibited by the target or the recognition of behaviours that do not comply with what is considered usual and thus implicitly accepted. The former approach is usually adopted in the cases where the monitoring action is focused on spotting just a few well-known behaviours of interest or when a comprehensive description of all the unacceptable manoeuvres is provided by experts. With reference to this approach, techniques widely adopted in literature include probabilistic frameworks ([4, 54, 177]), fuzzy systems ([61]) and pattern matching ([99]). When however no specific knowledge about the monitored events is available, it is necessary to implement learning techniques enabling the extraction of behaviour models from previous observations (the second case previously mentioned). In this case, any pattern that stands out with respect to the other data points within the considered dataset is classified as an anomaly. This approach is probably the most considered by current researchers because of the large availability of data provided by modern computer systems.

This chapter proposes an approach to the TOI identification problem that relies on a novel pattern matching technique based on *Regular Expressions* (regexes). These are a popular search tool that is widely adopted in the information technology field ([144]), e.g., within operating systems, programming languages, etc.. The proposed solution addresses both the anomaly detection approaches previously discussed[1] and can thus be adopted under different circumstances and monitoring conditions. As a preliminary step, a simple approach to manoeuvres detection is described: this consists in analysing the speed and curvature of target trajectories by means of differential geometry techniques ([99, 111]), enabling the classification of these as discrete values. The result of this process is then matched with either pre-defined reference patterns representing specific behavioursof interest or sets of patterns representing what is considered “normal” to various extents. These latter patterns, in the proposed solution, are retrieved by means of an automated learning process and organised as a dictionary of regexes.

The proposed solution for manoeuvres detection is intended as part of the , decentralised monitoring framework described in Chapter 4, with specific reference to the outlined case study.

The techniques outlined in this chapter are inspired by the approach introduced by Oh et al. in [99]. However, a more powerful, flexible and robust technique for pattern-matching is here developed, which is based on regular expressions. Furthermore, the approach proposed by Turchi et al in [157] is extended by introducing a whole new learning component that allows generating regexes representing commonly observed behaviours.

7.2 Trajectory Classification

In the proposed approach, the first step necessary to achieve behaviour recognition is the classification of the trajectories exhibited by the target. The purpose of this step is to translate a general trajectory into a set of predefined motion categories; these are referred in the following as *Driving Modes* (DMs). This can be done by identifying and analysing a few, simple quantities (e.g., position, velocity, heading rate, etc.) capable of characterising the trajectories and using this information for selecting the most suitable driving mode.

Two main steps can thus be identified within the trajectory classification process:

Curvature Analysis: allows extracting from the target track the quantities of interest: trajectory curvature, speed and acceleration;

[1]That is, the one based on ‘a priori’ knowledge and the one based on learning techniques.

Manoeuvre Classification: on the basis of the quantities previously extracted, this step produces a classification of the target trajectory within a predetermined set of driving modes.

7.2.1 Curvature Analysis

The proposed approach to curvature analysis is based on a moving-window-based trajectory approximation ([99]) and exploits a third-order polynomial function generating a trajectory with a virtually increased sampling frequency([97]).

Given a window composed of N_T original samples taken at time instants[2]

$$\left[0,\, T_s,\, 2T_s, \ldots, (N_T-1)T_s\right],$$

a new samples sequence is generated by means of a resampling process, taking place at instants

$$\left[0,\, T_n,\, 2T_n, \ldots, (N_T-1)cT_n\right],$$

where the final instant of the two time sequences is the same, that is

$$(N_T-1)cT_n = (N_T-1)T_s. \tag{7.1}$$

In the previous description, T_s and T_n are the original[3] and "artificial" sampling times respectively, and c is the ratio between the two times. In the following, the case is considered where $T_n < T_s$ ($c > 1$): this allows defining a sequence of data generated at 'virtually increased' sampling frequency. The resampling process has been performed with the purpose of applying the curvature analysis procedure to finer-grained, even if approximated, information, which leads to a more effective identification of DMs.

The polynomial function leveraged for the trajectory approximation has the following form:

$$p(x) = p_n x^n + p_{n-1}x^{n-1} + \ldots + p_1 x + p_0 \tag{7.2}$$

where n is the desired order, and the coefficients $p_0, \ldots, p_n$ are chosen so as to minimise the approximation error in a 'least squares' sense. Such optimisation can be formulated as:

$$\tilde{P} = \underset{\left[p_0,\ldots,p_n\right]}{\operatorname{argmin}} \sum_{k=T-m+1}^{T} \big(p\,(k) - x(k)\big)^2 \tag{7.3}$$

where $\tilde{P} = [\tilde{p}_0, \ldots, \tilde{p}_n]$ is the vector of optimal coefficients for the polynomial , T is the current time step and $X = \left[x(1), \ldots, x(m)\right]$ is the history of the target position

[2] Time instants are expressed as offset respect to the window initial time.

[3] It could be the time step of a tracking filter or simply the sampling time of a discrete measurement process.

(a single coordinate) within the considered time window of dimension m. With this approach two polynomial functions are defined for approximating the target trajectory in the two coordinates of motion[4], referred as $p_x(t)$ and $p_y(t)$. The target position at discrete time instants within the time window $[T - N_T T_s, T]$ is then obtained by sampling from the two polynomial approximating functions. This process is performed at a "higher rate" respect to the sensing frequency, that is with time step T_n instead of the original value T_s; in the following, indices k and i make reference to original and virtual samples respectively within the considered time window. After this, the velocity $(\dot{x}(i), \dot{y}(i))$ and acceleration $(\ddot{x}(i), \ddot{y}(i))$ values relative to the two coordinates of motion are derived by simple differentiation of the re-sampled position and velocity profiles respectively. These profiles are then used to calculate the forward acceleration $a_f(i)$, the orientation rate of change $\theta(i)$ and the minimum speed U of the vehicle for each i:

$$U = \min v(i) = \min \sqrt{\dot{x}(i)^2 + \dot{y}(i)^2} \tag{7.4}$$

$$\begin{aligned} \theta(i) &= v(i)\kappa(i) \\ &= \sqrt{\dot{x}(i)^2 + \dot{y}(i)^2}\frac{\dot{x}(i)\ddot{y}(i) - \dot{y}(i)\ddot{y}(i)}{\left(\dot{x}(i)^2 + \dot{y}(i)^2\right)^{3/2}} \end{aligned} \tag{7.5}$$

$$a_f(i) = \ddot{x}(i)\cos\psi(i) + \ddot{y}(i)\sin\psi(i) \tag{7.6}$$

where κ is the curvature, and $\psi = \tan^{-1}(\dot{x}/\dot{y})$ is the heading angle from North.

7.2.2 Manoeuvre Classification

Given the quantities defined in Section 7.2.1, the following driving modes ([99]) can be identified from the analysis of the data within a single time window:

- **Stopping(0)** : this state is detected when $U < U_{\min}$, indicating that the target is stationary, stopping or moving;
- **Left Turn (1)**: recognised when $\max(\theta) > \theta_{\text{th},1} > 0$ and $\max(\theta)\min(\theta) > 0$.
- **Right Turn (8)**: $\min(\theta) < -\theta_{\text{th},1} < 0$ and $\max(\theta)\min(\theta) > 0$;
- **Left Lane Change (2)**: this kind of manoeuvre is detected when

$$\begin{cases} \max(\theta)\min(\theta) < 0 \\ \max|\theta| > \theta_{\text{th},2} \\ \theta(0) > 0 \end{cases}$$

[4]Since the thesis is focused on ground vehicles, a bi-dimensional motion is assumed, but the approach proposed can be easily extended to the case of 3D trajectories.

The difference respect to a 'Left Turn' consists in the detection of a sign change[5] for the orientation rate of change;

- **Right Lane Change (7)**: characterised by

$$\begin{cases} \max(\theta)\min(\theta) < 0 \\ \max|\theta| > \theta_{\mathrm{th},2} \\ \theta(0) < 0 \end{cases}$$

The inspection of the sign change of θ is used to distinguish a lane change from a pure 'Right Turn' manoeuvre;

- **Straight (9)**: in this mode, the monitored target is moving straight at constant speed, i.e. $\max(|\theta|) < \theta_{\mathrm{th},3}$ and $\max(|a_f|) < a_{th}$;

- **Closing Gap (6)**: characterised by $a_f(0) > a_{th}$ and $\max(a_f)\min(a_f) < 0$. In the case where a target gets closer to the preceding vehicle[6], the monitored target will exhibit positive and negative accelerations in the beginning and in the end, respectively, of the considered time window;

- **Widening Gap (3)**: $a_f(0) < -a_{th}$ and $\max(a_f)\min(a_f) < 0$. Considerations about this driving mode are similar to those for the 'Closing Gap' case, with the difference that the initial acceleration is negative;

- **Accelerating Ahead (5)**: this driving mode is recognised when $\max(a_f)\min(a_f) > 0$ and $a_f(0) > a_{th}$. The sign of acceleration is positive during the whole time window;

- **Decelerating Ahead (4)**: in this case

  $\max(a_f)\min(a_f) > 0$ and $a_f(0) < -a_{th}$, i.e. the sign of the acceleration is negative over the whole time window;

- **U-Turning (A)**: detected when $\max|\theta| > \theta_{\mathrm{th},4}$. A U-turn manoeuvre, where the vehicle inverts the direction of motion through a 180° rotation, can indeed be associated with large values for the orientation change.

In the previous classification, the quantities U_{min}, $\theta_{\mathrm{th},1}$,$\theta_{\mathrm{th},2}$, $\theta_{\mathrm{th},3}$, $\theta_{\mathrm{th},4}$ and a_{th} are threshold values that need to be tuned on the basis of the estimation accuracy and expected target dynamics.

[5] As shown in Fig. 7.1, during a left lane change the curvature sign changes from positive to negative values.

[6] This vehicle is assumed to move at constant speed.

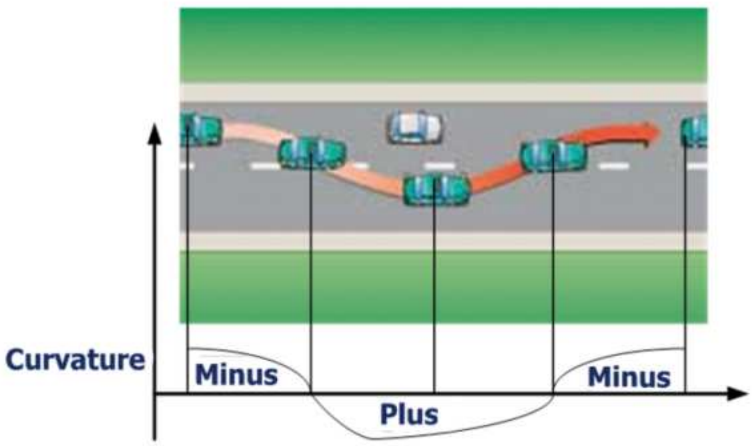


FIGURE 7.1: Right lane change and left lane change manoeuvres during an overtaking

7.2.3 Test Patterns Definition

The classification introduced in Section 7.2.2 enables the definition, at each time step k, of a driving mode m_k. This is chosen among the set of pre-defined modes $M = [0, 1, \ldots, 9, A]$ and describes the behaviour of the monitored target during the considered time window. These modes can then be leveraged for defining classes of complex behaviours that could draw the attention on the monitored target.

In the proposed approach, complex behaviours are defined as specific strings of driving modes, and the behaviour detection relies on string matching techniques. More specifically, given the modes set M, a symbolic time-series of driving modes $y_k = \{m_j \in M | j = k - N_w + 1, k - N_w + 2, \ldots, k\}$ is generated at each time-step k, where N_w is the adopted window dimension for behaviour detection. This last quantity depends on the expected duration of the events that have to be detected: assuming to be interested in events that last no longer than t_w, N_w can be calculated as follows:

$$N_w = \left\lfloor \frac{t_w}{T_n} \right\rfloor - (N_T - 1) \tag{7.7}$$

7.3 String Pattern Matching

The list of DMs extracted from the target trajectory is then compared with pre-defined strings associated with peculiar behaviours that have to be detected, referred in the following as *reference patterns*. Such comparison process is indicated as *patterns* (or *strings*) *matching*, and, in the proposed approach, it is implemented employing both basic strings of characters and regexes. These two techniques are thoroughly described and analysed in Sections 7.3.1 and 7.3.2, respectively.

7.3.1 Edit Distance Based Matching

Given a reference and an observed pattern, it is clear that the simple exact matching of the strings would represent an excessively strict policy to pattern recognition. This approach would not account for minor differences in the compared strings, which do not sensibly affect the overall interpretation of the target behaviour. The concept of 'edit distance' has been introduced as a generalised measure of the difference between two strings: this allows defining a discrete level of similarity between patterns instead of a binary matching value. In [155] the edit distance between two strings S_r and S_t (reference and tested strings respectively) is defined as the minimum total number of changes C, insertions I and deletions R required to turn string S_t into S_r, that is:

$$D(S_r, S_t) = \min_j \big[C(j) + I(j) + R(j)\big] \tag{7.8}$$

where j runs over all possible combinations of symbol variations in order to obtain S_r from S_t. The main drawback of this simple approach is that the same relevance is given to missing, excess, wrong or misplaced terms within the analysed pattern.

Considering, for example, the case where the monitoring agent is interested in detecting overtaking manoeuvres performed by a target, an example of pattern to look for could be '755992', that is the overtaking car moves to the right (overtaking) lane, accelerates for achieving a velocity greater than that of the preceding car, keeps a constant speed until it passes the other vehicle and then moves back to the left lane. Obviously, lots of similar patterns should be recognised as "overtaking manoeuvre", e.g. '755552' and '799992', both characterised by edit distance 2 respect to the reference string.

Setting a threshold on the edit distance value allows recognising these slightly different patterns, but does not consider adequately the characterising peculiarities of each manoeuvre. Paradoxically, by adopting a threshold value equal to 2, a perfectly straight trajectory '955999' would be recognised as overtaking manoeuvre. This happens because no emphasis is given to the fact that the characterising features of an overtaking are the two specular lane changes at the beginning and end of the manoeuvre. In the cases where these two driving modes are not detected, it makes no sense to consider the pattern an overtaking. A possible solution to this problem could account for different coefficients for the components of the edit distance (C,R and I). However, it is important to note that weights-tuning is often a tricky process requiring in-depth insight into the considered problem. The previous considerations suggest that it is probably worthy to investigate for new, flexible approaches to reference patterns definition.

7.3.2 Regular-Expressions Based Matching

The matching approach proposed in this chapter consists in expressing the reference patterns as regexes: this allows defining a fixed binding structure for the behavioural patterns of interest, specifying mandatory and optional terms. With this approach, a reference pattern consists of any combination of the following components:

- **Specific Symbol, e.g. 1:** indicates a driving mode that must be present in the tested string;
- **Character Class, e.g. [65]:** gives multiple possibilities, equally acceptable, for a term of the tested string (6 and 5 in the example);
- **Upper Unbounded Character Sequence, + :** one or more instances of the preceding element are allowed;
- **Unbounded Character Sequence, * :** zero or more instances of the preceding element are allowed;
- **Bounded Character Sequence, $\{n,\ m\}$:** a number between a minimum n and a maximum m of instances of the preceding element are matched.

In the cases where the reference string is expressed by means of a regex, the matching result is a value representing an extension of the previously described edit distance:

- $D_r(S_r, S_t) = -1$: the tested string does not match the reference pattern;
- $D_r(S_r, S_t) = \min_j I(j) \geq 0$: the edit distance indicates the number of insertion required for the test string to comply with the reference pattern;

With this approach, a possible reference pattern for an overtaking manoeuvre is:

$$S_{r,o} = \text{‘}\,[12345689] * 7 + [34569]\,\{4, 16\}2\text{’}$$

where the monitored vehicle moves to the overtaking (or *outer*) lane (‘7+’), maintains a straight direction for at least four time steps (regardless of the acceleration profile, ‘[34569] $\{4, 16\}$’) and then moves back to the left (or *inner*) lane (‘2’). The use of the unbounded sequence term (‘*’) allows decoupling the length of the reference and tested strings without affecting the edit distance: strings ‘99799992’ and ‘799992’ define pretty similar behaviours, returning, however, different edit distances if the classical approach is applied. By adopting regexes, instead, the symbol ‘*’ can be used to reject the influence of driving modes detected before the manoeuvre of actual interest, which entails $D_r\left(S_{r,o}, \text{‘99799992’}\right) = D_r\left(S_{r,o}, \text{‘799992’}\right)$. As suggested by this last example, the use of bounded and unbounded sequences appears to be particularly useful when the target dynamics is uncertain, and thus the length of the time interval needed by the target to exhibit the behavioural pattern of interest is not exactly known a priori. In the case of the overtaking manoeuvre, for example, the time interval between the two lane changes, and thus the number of characters between modes 7 and 2, depends on the relative velocity of the two vehicles involved. Consider the following three reference strings:

$$S_{r,o_1} = \text{‘755992’} \tag{7.9}$$

$$S_{r,o_2} = \text{‘}\,[12345689A]*7+[34569]\{4,16\}2\text{’} \tag{7.10}$$

$$S_{r,o_3} = \text{‘}\,[12345689A]*7+[34569]\{4,\}2\text{’} \tag{7.11}$$

where S_{r,o_1} is expressed with the basic approach, while S_{r,o_2} and S_{r,o_3} are defined as regexes. S_{r,o_1} clearly represents a strict reference pattern: indeed, if it would be compared only to strings of the same length, the matching approach would not be robust respect to little variations in the expected target dynamics since, for example, an overtaking that takes slightly longer, e.g. ‘75599992’, would not be recognised. Even by comparing S_{r,o_1} with strings of different length, the ability to detect the overtaking manoeuvres is drastically reduced as more the target dynamics differs from the expected: the edit distance calculated for ‘75599992’, a clear overtaking manoeuvre just a little slower than expected, would be indeed the same associated with the string ‘955999’, which is definitely not an overtaking. S_{r,o_2} is a regex where both upper and lower bounds are defined on the target dynamics: because of the term $\{4, 16\}$, a lower bound of four time-steps is imposed where the target has to move straight along the overtaking lane. Similarly, the value 16 defines an optional extension of this interval, and thus an upper bound beyond which the pattern is not matched. Similar considerations can be made about the reference pattern S_{r,o_3}, where however there is no upper bound for the duration of the manoeuvre. It is thus clear how the proposed approach leads to greater flexibility in pattern recognition respect to the classic edit distance.

7.4 Approaches to Reference Patterns Definition

7.4.1 Knowledge-Based Reference Patterns

The simplest and most intuitive way to implement the pattern matching approaches described in Sections 7.3.1 and 7.3.2 is to define one or more patterns representing particular behaviours considered “concerning” or, more generally, “of interest’. With reference to the traffic monitoring scenario discussed so far, such behaviours could consist in any manoeuvre representing a danger for the regular traffic flow: e.g., to continuously switch from a lane to another, to suddenly brake after an overtake manoeuvre or to overtake on the left. It is clear that this approach requires exploiting the knowledge of domain experts for defining the patterns that have to be spotted. On the one hand, a monitoring application designed for matching KB reference patterns is able to precisely identifying particular behaviours of interest. On the other hand, with this kind of approach it is not possible to detect general unexpected manoeuvres that differ significantly from those regularly exhibited by the targets.

7.4.2 Learning-Based Reference Patterns

To overcome the limitations due to considering only KB reference patterns, a new component is taken into consideration within the assessment process: reference patterns defined on the basis of previously observed target behaviours.

7.4.2.1 Basic Concepts

The rationale behind the use of a *Learning-Based* (LB) component is that the detection of manoeuvres that are rarely observed, even if not 'a priori' classified as 'concerning', should anyway raise the attention level on the associated vehicle. This approach appears justified assuming that a sufficiently large amount of target manoeuvres has been analysed, and thus that almost any legit manoeuvre has been observed a fair amount of time. Such approach is not conceptually different from that of Gaussian processes ([119]), where the observed samples of a process define the expected mean and covariance values for future outcomes.

With reference to the string matching approach described in Section 7.3.2, the purpose of the learning component is to infer a set of regular expressions from the observed sequences of driving modes. This set is referred in the following as '*dictionary*' and is considered capable of summarising all the "frequent", and thus assumed "regular", target manoeuvres. In so doing, when a pattern observed during the monitoring process does not match any of the expressions in the dictionary, the relative vehicle can be reasonably considered a TOI.

7.4.2.2 Learning Technique

The objects of the proposed learning process are patterns of driving modes, that is sequences of alpha-numerical characters with no ordinal meaning. Given this characterisation, the most suitable learning approach appears to be the use of a neural network. This tool, if properly trained, can accept a sequence of driving modes of fixed length as input and produce a single value denoting to what extent the pattern can be assumed "regular". Various learning techniques have been considered in addition to neural networks, e.g. Hidden Markov Model and Gaussian processes, but these did not fit satisfactorily to the examined problem.

For these reasons, in the proposed approach, a NN is leveraged for implementing the learning process and, subsequently, for defining groups of regular expressions constituting the regex dictionary. This NN is denoted in the following as '*supporting neural network*' since its role is solely to support the creation of the dictionary, and it is not meant to be further executed during the actual functioning of the assessment framework.

7.4.2.3 Neural Network Implementation and Training

The structure of the neural network adopted in the remainder of the chapter (Figure 7.2) is composed of three layers: the first one, the 'input layer', is made of n neurones, where n is the size of the input vectors. This layer has the sole purpose of 'presenting' the input data to the network[7] (that means distributing the input values to the neurones of the following layer) and thus does not entail any calculation (passive nodes). The second layer, referred as 'hidden layer', is composed of n neurones as well, while the third, or 'output', layer has a single neuron connected to all those within the hidden layer.

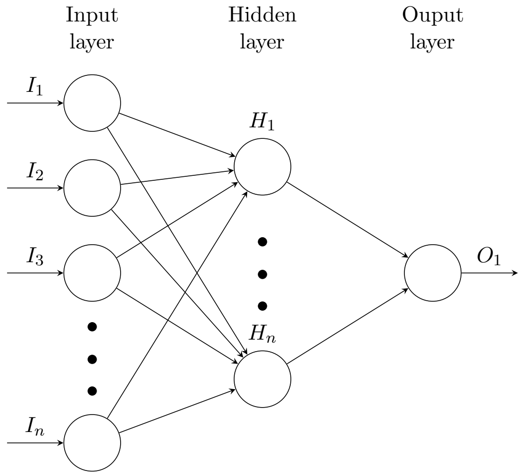


FIGURE 7.2: Structure of the adopted supporting neural network

In the proposed approach, the dataset used for training the neural network (*NN-training set*) entirely or partially consists of *synthetic data* and is composed as following:

- A set of "assumed regular" patterns, associated with output value 1. These can be obtained: i) by utilising a Markov Model, designed for producing sensible sequences of driving modes, ii) by logging patterns generated from a traffic simulator (described in Section 7.5.1.2) or iii) by analysing real traffic data. Different sources of data have been considered for testing the effectiveness of the proposed methods

[7]The presence of the input layer has more to do with the description of the neural network than with its actual implementation, and it is in some cases neglected.

under differentconditions. In the last two cases, since the traffic simulator implements a dynamic model that evolves from random initial conditions and since the real-word data is not labelled, some "irregular" patterns could be presented to the network[8]. Nonetheless, since these "irregular" sequences likely represent a tiny or non-existent part of the "assumed regular" patterns dataset, their effect on the training process can be expected to be negligible.

- A set of "irregular" sequences of driving modes, associated with the output value 0. These patterns are produced in an automated manner, by including within a random sequence of driving modes a sub-sequence representing a manoeuvre that is known to be concerning, e.g., multiple lane changes in quick succession, prolonged braking or continuously switching between acceleration and deceleration.

7.4.2.4 RegEx Dictionary Creation

Once the neural network has been trained to distinguish between "regular" and "suspicious" behaviours, it can be used to derive one or more sets of regular expressions that allow summarising and generalising the observed behavioural patterns. This can be done by providing a new set of patterns as input to the neural network. These patterns are denoted as *RD-training set* (where RD stands for *Regex Dictionary*) and are defined following the same procedure used for the NN-training set. Noticing that the network outputslie in the $[0, 1]$ range, the RD-training patterns can be split among a pre-defined number of bins on the basis of the NN output associated with each of them, as follow:

$$\mathrm{nn}\left(p_i\right) > \gamma \wedge \mathrm{nn}\left(p_i\right) \in \left[1-\left(j-1\right)s_b, \quad 1-js_b\right] \Rightarrow p_i \in b_j \tag{7.12}$$

where the 'nn' operator represents the neural network, p_i is the i-th pattern of the RD-training set , γ is the threshold[9] on the NN outputs for the patterns to be considered within the regexes creation process, $s_b = \left(1-\gamma\right)/N_b$ is the step between the values associated with two successive bins, N_b is the number of bins adopted and b_j denotes the j-th bins, with $j = 1, \ldots, N_b$.

The set of patterns associated with the j-th bin can thus be denoted as $p_{b_j} = \{p_i : p_i \in b_j\}$, and it can be used to define one or more regexes capable of expressing all the behaviours related to patterns p_{b_j}. This is done through a newly-developed algorithm that, given a set of patterns p_{b_j}, returns a set of regexes $r_{b_j} = \{r_{i,j}\}$, with $i = 1, .., n_{r_j}$ where n_{r_j} is the number or regexes generated for the j-th bin. By applying this procedure to each bin, a dictionary of regular expressions is built from the RD-training patterns and the output values provided by the supporting NN. Each of the N_b bins constituting

[8]This is why the expression 'assumed regular' has been used for this type of patterns.
[9]The whole range $[0, 1)$ is considered when $\gamma = 0$.

the dictionary is associated with a different level of "regularity" assumed for the patterns matched by the regular expressions contained in the bin.

Given a regex dictionary d structured as described, the assessment process for a generic pattern p simply consists in finding the first bin from the top (index 1) containing a regex that matches the pattern. Assuming k as the result of the search, the assessment for the considered pattern can be formulated as follow:

$$\mathrm{l}(k) = \mathrm{l}\big(\mathrm{srch}(\mathrm{p}, \mathrm{d})\big) = 1 - (k-1)\, s_b \tag{7.13}$$

where the function srch() returns the index of the first bin containing a regex that matches the input pattern.

The novel approach proposed to on-line monitoring of targets consists in querying a dictionary of regexes like if it were the supporting[10] NN. As stated, the dictionary is created 'offline' on the basis of the NN outcomes and is characterised by a granularity that is function of the number of bins N_b and the threshold γ adopted. This approach can be adopted under the assumption that

$$\mathrm{nn}(\mathrm{p}) \simeq \mathrm{l}\big(\mathrm{srch}(\mathrm{p}, \mathrm{d})\big)$$

that is, that the dictionary-query technique represents a valid approximation of the neural network. If and to what extent this assumption can be considered correct is investigated through a comprehensive set of tests in Section 7.5.2.

7.4.2.5 Real-Time Dictionary Update

A relevant advantage of using a dictionary of regexes respect to a neural network is that the former can be easily updated in real-time: this means that the system can match patterns that were not accounted for during the system design phase but appears to be of considerable importance during the operational phase. In so doing, the monitoring application can quickly adapt to different scenarios or temporary conditions influencing the targets behaviour.

For example, a prolonged slowdown on a motorway could be a priori considered as a suspicious manoeuvre, but if all the vehicles were doing that, e.g. because of a closed lane, the assessment for that pattern should be updated to "regular". Such result can be obtained by adding, during the monitoring process, new regular expressions to the dictionary or by moving an existing regex to a different bin in the dictionary hierarchy.

[10]The NN used for the dictionary creation.

7.4.2.6 Assessment Framework structure

Given the two proposed approaches to TOI detection (Knowledge-based and Learning-based), an hybrid assessment framework for supporting operator decisions can be considered, whose structure is outlined in Fig. 7.3. In this representation on-line components are depicted in violet, while the off-line part is in orange. It is possible to notice that the TOI warnings (TOIWs) produced by the pattern matching techniques are not directly forwarded to the operator but are provided to a further processing block indicated as 'Target Assessment Process': its role is to apply some kind of filtering action to the automatic TOIWs, accordingly with the operator specification and requests. From the practical point of view this component could:

- Implement the logic sum of the two input signal[11];
- Impose a threshold on the number of warnings necessary for classifying a vehicle as a TOI;
- Implement any arbitrarily complex policy.

The functionalities related to this component are not further analysed since they are strictly related to the operator choices and its personal interpretation of what represents an actual threat.

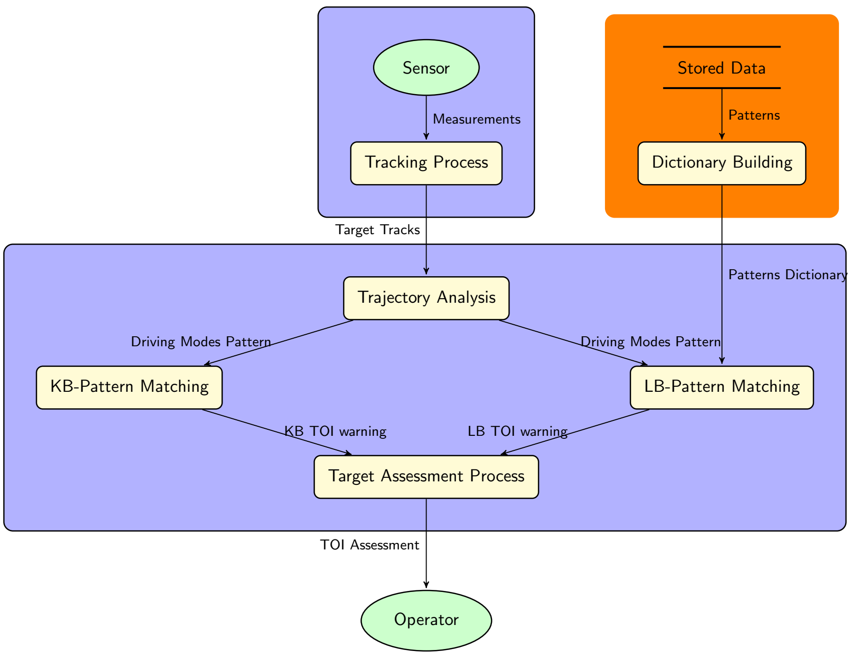


FIGURE 7.3: Schematic representation of the proposed assessment approach

[11]In the case that the operator is interested in each possible "irregular" behaviour exhibited by the targets.

Table 7.1: Markov chain for patterns generation, KB approach

<table>
<tr><th></th><th></th><th colspan="5">Driving Modes</th></tr>
<tr><th></th><th></th><th>(2)</th><th>(7)</th><th>(5)</th><th>(4)</th><th>(9)</th></tr>
<tr><td rowspan="5">Driving Modes</td><td>Left Lane Change (2)</td><td>1/2</td><td>1/6</td><td>1/6</td><td>0</td><td>1/6</td></tr>
<tr><td>Right Change (7)</td><td>1/6</td><td>1/2</td><td>1/6</td><td>0</td><td>1/6</td></tr>
<tr><td>Acceleration Ahead (5)</td><td colspan="2">1/6</td><td>1/2</td><td>1/6</td><td>1/6</td></tr>
<tr><td>Deceleration Ahead (4)</td><td colspan="2">1/6</td><td>0</td><td>2/6</td><td>2/6</td></tr>
<tr><td>Constant Speed (9)</td><td colspan="2">2/6</td><td>1/6</td><td>1/6</td><td>1/6</td></tr>
</table>

Table 7.2: Numerical result for simulations based on Markov Chain and Dynamic Model (%)

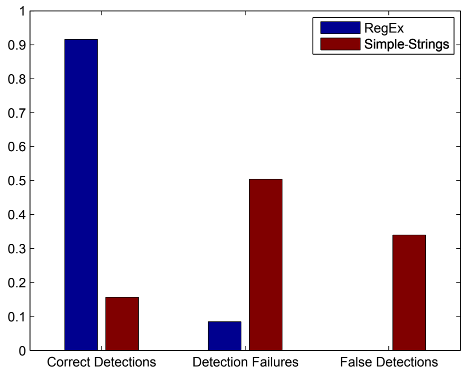
	Markov Chain				Dynamic Model	
	Consistent Knowledge		Inaccurate Knowledge		regex	Basic
	regex	Basic	regex	Basic		
Correct Detections	90.5	16	63.8	22	84.4	15.4
Missed Detections	9.5	49.5	36.2	58	15.6	84.6
False Detections	0	34.5	0	20	0	0

7.5 Simulations

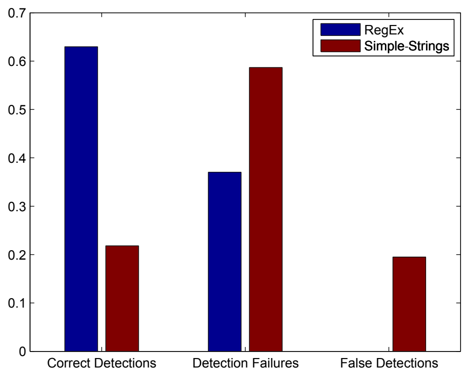
This section reports the results from an extensive set of simulations and tests performed for checking the effectiveness of the pattern matching approaches proposed in Section 7.3. Knowledge-Based and Learning-Based matching solutions have been tested separately, allowing to compare the proposed solutions with corresponding well-established techniques. The final target assessment step has not been taken into consideration in the following tests, since it entails a non-univocal interpretation of the problem and no labelled datasets or similar assessing techniques can be easily retrieved.

7.5.1 Knowledge-Based Matching

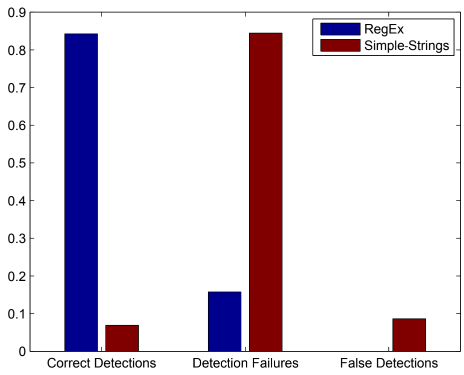
The two string-matching techniques described in the Section 7.3 have been tested and compared for assessing whether the approach based on regexes performs better than the basic technique. Two different methods have been considered for generating the strings to be tested against the reference patterns. As first approach, a Markov chain has been leveraged, which produces sequences of DMs accordingly with some reasonable probabilities. Then, a dynamic system simulating a car moving on a two-lane highway has been implemented: in this case, the strings of driving modes have been produced through the analysis technique described in Section 7.2.



(a)



(b)



(c)

FIGURE 7.4: Numeric results for regex and simple-strings based approaches: a "consistent" case simulation, b "inaccurate" case simulation, c, dynamic system simulation

7.5.1.1 Simulations Based on Markov Chain

The Markov chain described in Table 7.1 has been used for generating plausible sequences of driving modes: probability values are defined with a simple approach[12], taking into account some basic considerations:

- Manoeuvres time correlation suggests that the probability of remaining in the same state must be higher than switching to any other state;
- After the target has moved to the outer lane, it will not reduce its speed immediately;
- After the target has moved back to the inner lane, it will not reduce its speed immediately;

Lastly, notice that when in states $4, 5$ or 9 the probabilities of switching to modes 7 and 2 have been aggregated, since they depend on the lane occupied, i.e.:

$$\text{lane} = \text{left} \implies \begin{cases} \mathcal{P}\left(2\right) = 0 \\ \mathcal{P}\left(7\right) = P \end{cases}, \text{ otherwise} \begin{cases} \mathcal{P}\left(2\right) = P \\ \mathcal{P}\left(7\right) = 0 \end{cases}$$

where P is the probability reported in the table.

By means of the considered Markov chain, a sequence of one hundred thousand driving modes has been produced and analysed leveraging the two string-matching approaches, with the aim of identifying overtaking manoeuvres. The considered reference pattern for the simple-string approach is $S_{r,o1}$ = '755992', assuming a threshold value of two for the edit distance. This means that only strings with edit distance smaller or equal to two have been considered matching the reference pattern. For the regex-based approach, the regular expression adopted for describing an overtaking manoeuvre is S_{r,o_2} = '$[12345689] * 7 + [345689]\{4, 16\}2$', assuming a zero threshold on the edit distance, i.e. just exact matching have been considered. This is because the use of metacharacters and character classes already provides a sufficient degree of flexibility. Given the sequence of driving modes produced by the Markov model, the GT about the number of overtaking manoeuvres has been defined in two different cases:

Consistent Knowledge: just the sub-sequences with at least four "straight-direction" DMs between the left-lane-change and right-lane-change modes have been counted as actual overtaking manoeuvres. The term "consistent" is used since the assessment of the ground truth complies the definition of the reference patterns S_{r,o_1} and S_{r,o_2}. With this approach, the DMs sequence exhibits 3412 overtaking manoeuvres.

[12] Driving modes 'widening gap', 'closing gap' and 'U-turn', codified as '3','6' and 'A' respectively, are not considered in the following for simplicity.

Inaccurate Knowledge: sub-sequences with two or more straight-direction DMs between the two lane changes have been considered as actual overtaking manoeuvres. In this case, a non-exact knowledge of target dynamics is simulated since overtaking manoeuvres are executed much faster than expected (the reference patterns account for four straight-direction driving modes). With this approach, 6253 overtaking manoeuvres have been counted.

The results for the two cases are shown in Fig. 7.4a, 7.4b and summarised in Table 7.2: in the consistent case the correct detection rates are 90% and 16% for the regex-based and simple-string approaches, respectively. In the case of inaccurate knowledge the regex-based technique lowered its success ratio to 63.8% and the performance of the simple-string approach remains almost the same. False positive detections are not a problem concerning the regex-based technique (absent in both the cases), while in the simple-string case their fraction is not negligible: it is indeed of the same order of magnitude respect to correct detections.

7.5.1.2 Simulations Based on Dynamic Model

A second test for the two string-matching techniques has been carried out using a dynamic-system model representing an individual car moving on a two-lane highway; the car switches between the two lanes, accelerating, decelerating or keeping a constant speed. An "always-straight" road is considered for simplicity's sake; however, this can be extended to the more complex case of a generic road by making reference to the abstract representation introduced in Section 4.3.1.1.

The car trajectory has been simulated for fifty thousand seconds (corresponding to one hundred thousand driving modes, assuming a sample time $T_s = 0.5\,\mathrm{s}$), on the basis of the following procedure:

I) The car moves along the inner lane for a randomly distributed time interval $t_{\mathrm{in}} \sim \mathcal{N}(30, 16)$. The acceleration profile during t_{in} is defined dividing the interval into n_{in} randomly distributed[13] sub-intervals $t_{\mathrm{in},j} \sim \mathcal{N}\left(2,\ 6.25 \times 10^{-2}\right)$ with $j = 1, 2, \ldots, n_{\mathrm{in}}$. For each interval $t_{in,j}$ the vehicle can decide to accelerate, decelerate or maintain a constant velocity;

II) The car starts to move towards the outer lane with constant velocity;

III) The vehicle arrives on the outer lane and determines how long it will stay on this lane: $t_o \sim \mathcal{N}(5, 4)$. The acceleration profile during this period is defined using the same procedure as in step I;

IV) The car moves back to the lane on the right;

[13]The length of the last sub-interval $t_{\mathrm{in},n_{\mathrm{in}}}$ is constrained by the length of interval t_{in}.

V) The procedure restarts from the first step.

Other simulation parameters are:

- Maximum forward speed: 30 m/s;
- Minimum forward speed: 5 m/s;
- Lateral speed: ± 1.75 m/s;
- Acceleration: $\pm 2\,\mathrm{m/s}^2$

All the 'inner-to-outer lane' events followed by 'outer-to-inner lane' have been considered as overtaking manoeuvres regardless of the time interval spent on the outer lane; with this approach 1262 overtaking manoeuvres have been identified. The reference patterns for the two matching techniques are the same of the previous example (Section 7.5.1.1), and the numeric results obtained from the simulation are reported in Table 7.2 and Fig. 7.4c. Again, the regex-based approach undoubtedly outperforms the simple-string technique: 84% instead of the 15% of correct detections.

7.5.1.3 Computation Complexity Analysis

The time complexity for edit distance calculation is $\mathcal{O}(n \cdot m)$ ([163]) where n and m are the lengths of the compared strings. On the other hand, for classical regular expressions (the type here considered) the time complexity can be reduced to $\mathcal{O}(n)$, with n length of the tested string, by the use of a *Deterministic Finite Automata* (DFA), thus representing a less complex solution.

7.5.2 Learning-Based Matching

The learning approach for behavioural patterns has been tested under the same two cases considered in Section 7.5.1, and has been further applied to real-word vehicle trajectories (made available by the NGSIM project). The objective of the simulations reported in the following is to establish whether or not the use of a regex-dictionary represents a suitable solution for replacing a neural network, i.e. if these two methods have comparable outcomes. This analysis appears to be more meaningful than considering actual detection capabilities since: i) No GT is available for most[14] of the data ii) A neural network is assumed to be an effective tool for patterns recognition and thus its outputs are considered to be valid references.

[14] Only patterns produced by means of the Markov model are associated with an actual GT.

7.5.2.1 Markov Model

Similarly to the KB case, a Markov model (Fig. 7.5) has been defined for producing DM sequences associated with "regular" behaviours.

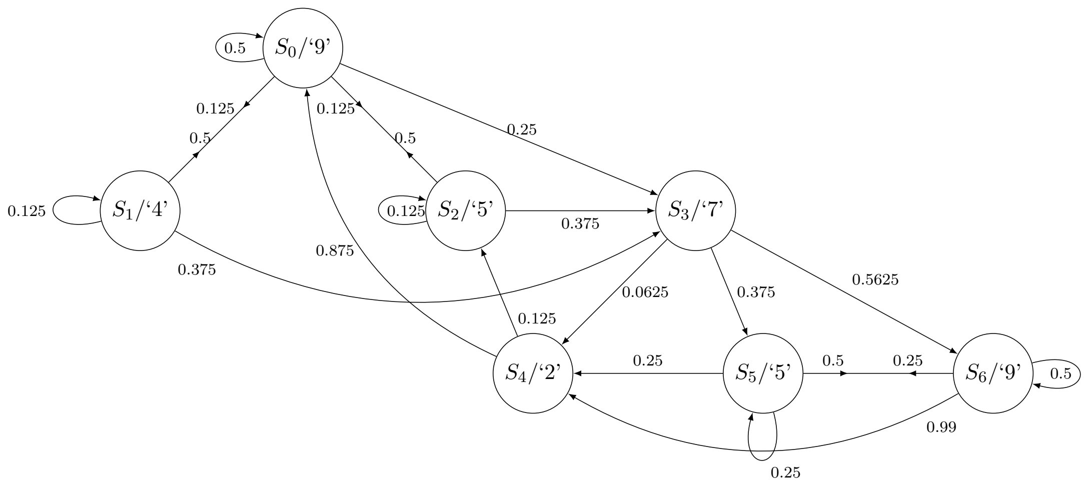


FIGURE 7.5: Markov model for patterns generation, learning-based approach

However, unlike the KB case, "irregular" patterns are also required for properly training the neural network. Each irregular pattern is generated as a random sequence of driving modes where a sub-sequence is added[15] that represent a known irregular behaviour.

The following datasets have been produced on the basis of the criteria just described:

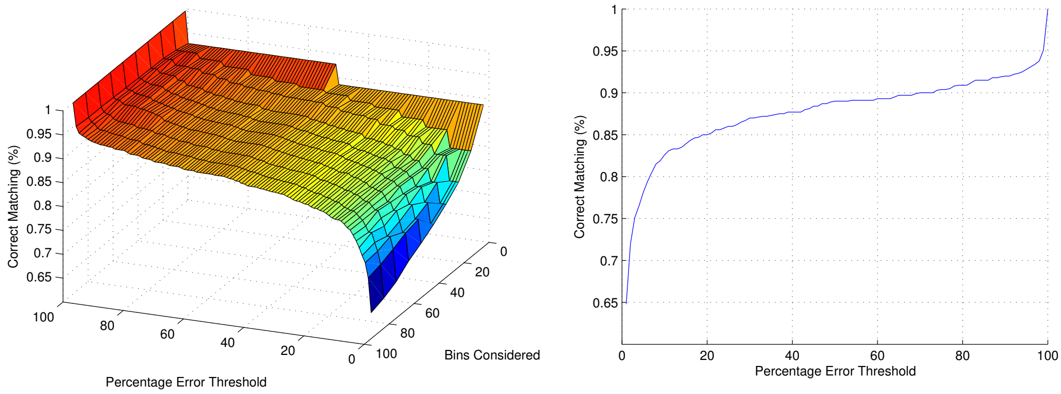
NN-training set: 20,000 patterns, used to train the neural network;

RD-training set: 20,000 patterns, used to query the trained network and populate the regex dictionary;

Test set: 2000 patterns, used to query both the neural network and the dictionary-based matching technique, thus allowing to compare the results provided by the two approaches.

All the datasets are composed of regular patterns for the 75% and irregular patterns for the remaining part. This ratio for the two components of the dataset, clearly privileging the regular patterns, has been chosen considering that the simulation focus is in learning regular, typical behaviours by examples, while irregular patterns are provided to the network just for training purposes. The proposed approach consists indeed in trying to learn what is regular and considering then everything else irregular. Using these datasets, a test accounting for various numbers of bins for the dictionary-based technique have been performed, and the results are depicted in Fig. 7.6.

[15] As example, assuming a prolonged deceleration (4) as an irregular behaviour, irregular patterns could be: '5792**444**959', '95**444**97254', '2472**444**9', etc., where **444** is the considered sub-sequence and the other driving modes are randomly chosen.



(a) (b)

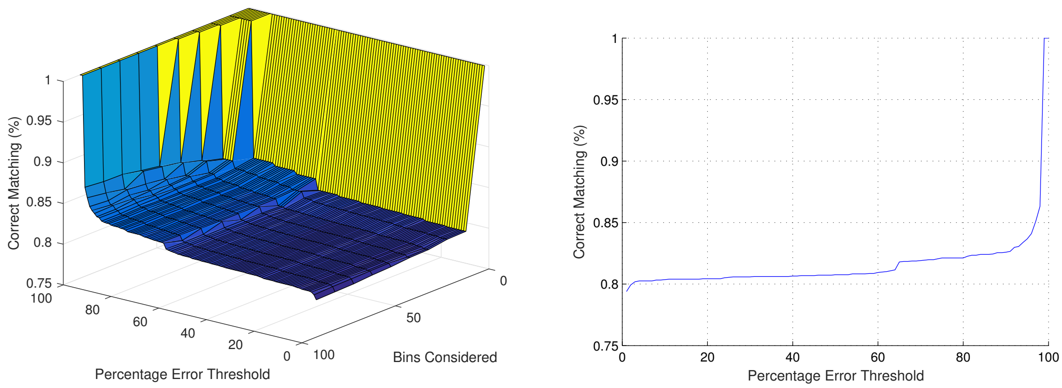
FIGURE 7.6: Results for the regex-dictionary against neural network approaches confrontation, Markov-model data: 7.6a multiple bins test, 7.6b detail of the results for the highest number of bins considered (100)

The aim of the test consists in verifying for what percentage of the input patterns the regex and the NN approaches provide the "same" result. With "same", it is here meant that the difference in the outcomes of the two methods lies within a given error threshold, which represents the second parameter of the test.

The numeric results indicate that the proposed dictionary-query approach has been able to reproduce the NN outputs for a fair portion of the test patterns: approximately 85% of correct matching assuming an error threshold of 20%.

7.5.2.2 Dynamic Model

The test described in Section 7.5.2.1 has also been performed using data from the dynamic simulator described in Section 7.5.1.2 instead of patterns from the Markov model.



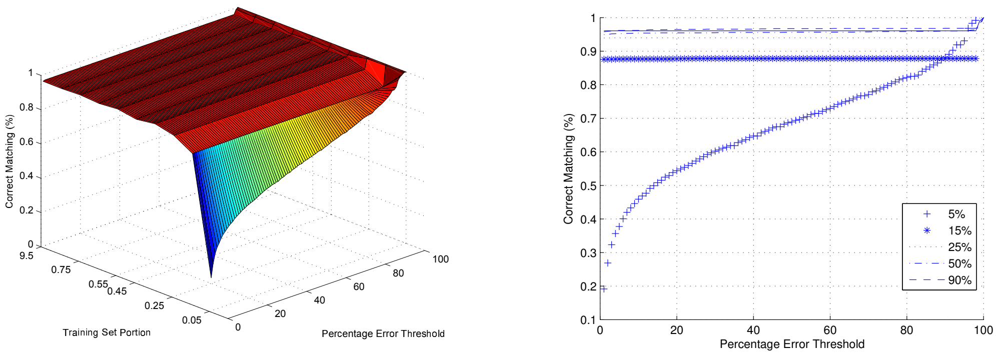
(a) (b)

FIGURE 7.7: Results for the regex-dictionary against neural network approaches confrontation, dynamic model: 7.7a multiple bins test, 7.7b detail of the results for the highest number of bins considered (100)

The test results for this different case are depicted in Fig. 7.7 and, from a qualitative point of view, are not substantially different from those obtained for the Markov model data. The resulting curve (Figure 7.7b) exhibits, indeed, a trend similar to that of Figure 7.6b, but it is possible to note a considerably less pronounced slope. This could be due to the fact that the dynamic-model simulator has probably produced a set of patterns with lower 'entropy', that is a smaller variety of DM sequences is used for training the network and thus to produce the regex-dictionary.

7.5.2.3 Real Data Analysis

In this section, the case is considered where the dictionary of reference patterns is inferred from real-word traffic data: 5200 trajectories have been extracted from the NGSIM US101 dataset, within the period of time between 7.50am and 8.35am. For the test, the original NGSIM data have been corrected utilising the technique described by Montanino et al. in [91], allowing to remove outliers and infeasible acceleration profiles due to measurement errors.



(a) (b)

FIGURE 7.8: Results for regex-dictionary against neural network approaches confrontation, real-world data: 7.8a matching ratio as a function of error threshold and training set portion, 7.8b detail of the results for different dimensions of the training set (fraction of the whole dataset).

On the basis of these target trajectories and by employing the analysis technique described in Section 7.2, 3563 driving patterns have been identified. These have been then leveraged for the definition of a dictionary accounting for 30 regular expressions distributed among 100 bins[16]. For this example test, the dictionary threshold γ has been set to 0, thus allowing to have a granularity of 1/100 in the representation of the NN outputs by means of the regex-dictionary. The test results are depicted in Fig. 7.8, where the ratio of correct matching between NN and regex-dictionary outcomes are plotted

[16] For the sake of simplicity and since the previous tests suggest that the number of bins does not significantly influence the matching ratio (causing though an underlying quantization error), the matching performances for a single value for the number of bins have been analysed.

against the training set portion[17] and error threshold. The numeric results clearly show that, even with a relatively small amount of data dedicated to the training process, the proposed regex-dictionary based approach is capable of replicating the outcomes of the neural network, exhibiting matching percentages between 90% and 95%. It is furthermore possible to notice that, apart from the case of extremely small training sets, the matching performance is not sensibly affected by the error threshold, which means that for most of the patterns in the test set the dictionary-query technique either provide a perfect match or it does not recognise the pattern at all.

7.5.2.4 Computation Time Analysis

A comparison of the computational efficiency for the dictionary-query and neural network techniques is here presented.

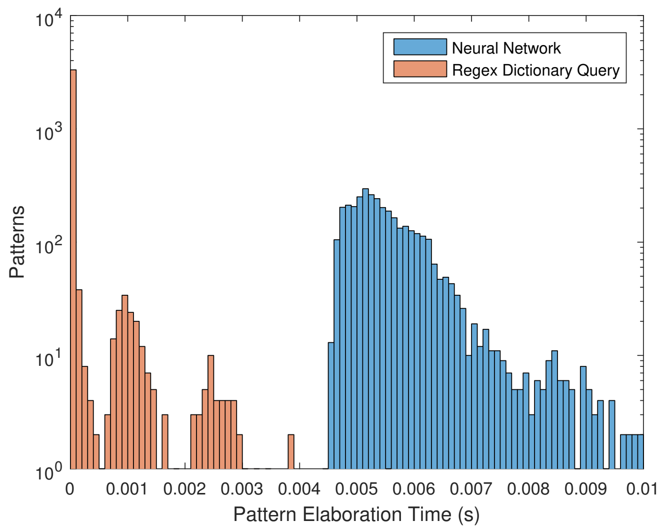


FIGURE 7.9: Distribution of the computation times for the 3563 patterns obtained from NGSIM data

The test has considered the real-word data case (Section 7.5.2.3), and the neural network and the regex dictionary have been defined on the basis of a training set portion equal to 0.5. Then, each pattern in the dataset has been assessed through both the considered techniques and the actual computation times[18] have been recorded. The neural network outputs have been calculated by means of the Matlab built-in *net()* function, while the dictionary-query approach relies on a newly-implemented function that browses the dictionary and checks for the first regex matching the pattern under assessment.

[17]Notice that, for the test here reported, the training set portion is equally distributed between the NN-training and RD-training sets.

[18]The reported time values make reference to a Matlab 2015 environment on an Intel i7-3770 16 GB RAM machine, with no parallel computation enabled.

The measured time intervals[19] for the two approaches are reported in the histogram in Fig. 7.9. This figure clearly shows that the regex-dictionary based technique is always faster than the neural network. Furthermore, it can be noticed that the use of a neural network leads to quite evenly distributed time values, while the dictionary-query has taken the minimum observed elaboration time for most of the patterns. This aspect can be attributed to the fact that, assuming to browse the dictionary from the top bin, the regular patterns, which are the most frequent, are usually matched almost immediately by a regex located in the first few bins. On the other hand, irregular patterns take longer since they require most or all of the dictionary to be browsed. Regarding the overall performance, however, this does not seem to be a problem, since irregular patterns are expected to be observed far less frequently than regular ones. The measured mean value and standard deviation for the elaboration times in the two cases are $\bar{t}_{NN} = 5.9\,\text{ms}$, $\sigma_{NN} = 1.5\,\text{ms}$ and $\bar{t}_{RD} = 0.12\,\text{ms}$, $\sigma_{RD} = 0.49\,\text{ms}$ for the neural network and regex-dictionary approaches, respectively.

7.6 Conclusions

This chapter has considered the problem of manoeuvres detection and behaviour recognition through the definition of a 2-step approach:

1. Classification of the target trajectories in sequences of driving modes;
2. Comparison of the driving modes sequences with reference patterns either defined a priori on the basis of specific knowledge or automatically learned by utilising a supporting NN.

These two different approaches to pattern matching have been tested and compared to well-established techniques in the field, leading to promising matching results.

Concerning the KB component, the proposed regex-based approach has clearly provided better results than the simple-string based method in all the tests that have been performed. The definition of reference patterns as simple-strings leads, indeed, to a lack of robustness against small variations in the actual string under assessment. The concept of 'edit distance', used as a threshold for the matching, tries to alleviate such problem, but appears to be not so effective, especially for long reference patterns. The use of regexes, on the other hand, allows achieving greater flexibility through the definition of optional and multiple-choice terms within the pattern of interest. This intuitive idea has been confirmed by all the tests presented in Section 7.5.1: the regex-based approach exhibits high rates of correct detections while the simple-string based method performances are

[19]The reported time values accounts only for pattern evaluation, and thus do not consider the network training and dictionary creation processes. These procedures are executed just once, probably off-line, and are thus not of interest for the on-line monitoring performance assessment.

always at an inadequate level. On the basis of these results, it is possible to affirm that the simple-string based technique seems not suitable to a minimally robust detection approach. On the other hand, regex-based string matching appears as an effective technique for implementing KB target assessment, allowing to detect anomalous behaviours when expressed as partially known sequences of driving modes.

For the learning-based component, a comparison has been proposed between the pattern matching results provided by the regex-based approach and the neural network. It has been shown that the use of a properly defined dictionary of regexes provides results very similar to those of a neural network in most of the cases, at far smaller computational costs. Furthermore, considering a distributed multi-agent environment, the regex-based approach allows to easily update and share the knowledge locally available to the monitoring agents (i.e. the regular expressions dictionary). Such process would be considerably more complex if a neural network would be adopted as pattern matching tool, making the regex dictionary approach represent a more flexible and practical solution for actual implementations.

The main contributions of the work presented in this chapter:

- The application of a regex-based matching technique to the field of automated monitoring, specifically to the problem of behaviour classification;
- The definition of a procedure for the automatic creation of a dictionary of regexes representing the behaviours commonly observed in a monitored scenario.

The regex-dictionary enables assessing to what extent an observed pattern can be considered regular and thus detecting uncommon and unexpected manoeuvres that should be considered possible threats.

It is worthy to remark the flexibility and generality of the proposed approach: this, by employing the pattern matching technique described, is capable of detecting a great variety of events other than the considered overtaking manoeuvre. The proposed method can thus be easily applied to various monitoring scenarios by defining appropriate reference patterns and driving[20] modes; these can depend on the specific application field (e.g., maritime, highways, urban roads) and on the behaviours of interest (e.g., fickle speed, suspicious stops, periodic behaviours).

[20] In a general context the term "driving" should be replaced by "moving", thus fitting to the problem of, e.g., boats or people monitoring.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

This thesis has investigated the problem of '*target assessment*' and '*Target Of Interest* (TOI) *identification*' taking into account solely the behavioural characteristics of the monitored vehicles. The proposed approach also enables to integrate 'context information' into the assessment process, thus taking into account environmental conditions possibly influencing the target behaviour.

The analysis approach considers the monitoring functionalities segmented into three different layers: '*sensor level*', '*monitoring level*' and '*decision level*'. The proposed work has focused on the analysis and implementation of the monitoring level, which takes into account 'refined' information (not simple sensor measurements) and makes decisions about the target classification. Because of this layered approach and the declared focus on the middle layer, the assessment of target behaviours has been performed by defining high conceptual-level (almost abstract) features. These have been utilised for quantifying the 'regularity' of complex aspects of the target dynamics by means of normalised values. The use of high-level, behavioural features has allowed decoupling the assessment process from the specific target considered: since dynamic models or prior information about the target type are not involved in the analysis, the proposed method seems to be widely adaptable to different scenarios and target classes.

The techniques proposed in the thesis have been developed with the aim of being general, flexible and adaptable to a variety of monitoring conditions and scenarios. The traffic-monitoring scenario considered within the thesis, indeed, just represents a particular case study, which has been adopted for defining clear and understandable test conditions.

Huge effort has been put into providing meaningful numerical results and performance indexes for the proposed techniques. As thoroughly explained in the thesis, the general assessment problem does not have an actual *Ground Truth* (GT) associated, and any numerical result is affected by a subjective component due to the artificially-defined, adopted GT. Nevertheless, all the assumptions at the basis of the numerical results provided within the thesis can be considered well motivated and consistent with common knowledge.

In the thesis, firstly, a basic distributed implementation of an assessment framework has been proposed, with the purpose of investigating the effects of consensus techniques applied to different layers of the monitoring hierarchy. Simulation results show that, in the considered scenario, reaching an agreement about local assessments leads to better distributed-assessment performances than having consensus on target tracks. This result is one of the main contributions of the thesis, confirming the effectiveness of the novel proposed approach based on: (i) the definition of high conceptual-level features and (ii) the implementation of consensus techniques dealing with assessments rather than individual measurements.

Then, the basic approach previously adopted for the feature characterisation and target assessment has been replaced by a more articulated and formal solution. This involves a stochastic, hierarchic representation for the features and a decision policy based on hypothesis-testing. The contribution of this novel proposed approach consists in providing: (i) An analytical description of arbitrarily complex behavioural features and (ii) An assessment criterion that jointly takes into account the whole set of considered target features . The described approach has been extensively tested trough numerous simulations: the obtained results demonstrate the efficacy of the proposed assessment technique and provide indications on the sensitivity of the method to different simulation conditions. An analysis of the framework performances on real-world data has been furthermore performed, the results of which suggest that the performance indexes obtained for simulated data can be qualitatively extended to the case of actual target trajectories

Lastly, the identification of possible TOIs has been performed utilising pattern matching techniques. Two different solutions have been presented with reference to this approach: the detection of known “irregular” patterns and the automated definition of a “dictionary” of “assumed-regular” patterns, which can be queried for assessing the regularity of observed behaviours. The main contribution, with reference to this part of the thesis, is the adoption of *Regular Expressions* (regexes) for implementing the pattern matching functionalities. This approach has allowed achieving better performances respect to well-established techniques for both the cases previously mentioned. More specifically, for the problem of matching a priori known patterns, the novel approach based on regexes has exhibited higher identification capabilities than the ‘Edit Distance’ technique. For the second case, where a list of regexes is inferred from trajectories data, the novel dictionary-based approach has provided similar performances respect to a properly trained neural network, requiring however sensibly shorter computation times.

In conclusion, general approaches for the target assessment task have been developed and analysed within the thesis. The outlined solutions are meant for a preliminary investigation of the problem, avoiding optimised and target-specific methods. For this reason, an actual implementation of the proposed framework could benefit from further analysis of the key issues underlying the assessment problem and from a refinement of the adopted techniques. This matter is considered in the next section.

8.2 Future Works

The problem of reproducing the generic human assessment process appears extremely challenging, especially when domain-specific knowledge about the monitored scenario is not available. The work presented within this thesis represents an experimental investigation of such problem, expecting to pave the way for further analyses and developments.

A first possible development of the proposed work is the application of consensus techniques to the assessment/detection processes described in Chapter 6 and Chapter 7. For the former, consensus protocols could be applied to the considered beta distributions so that an agreed stochastic representation for the behavioural features associated to a given target can be cooperatively defined by the network agents. A way for achieving this result could be, e.g., to define a distributed approach to the calculation of the *Kullback-Leibler Average*: this is indeed the distribution that minimise the sum of the *Kullback-Leibler Distances* respect to a set of input distributions. In so doing, the result of the consensus process would be the "closest" beta distribution to those suggested by the individual agents. For the TOIs detection approach introduced in Chapter 7, with reference to the *Learning-Based* (LB) matching case, agreement between agents could be simply obtained by applying an average-consensus protocol to the "regularity" values returned by the dictionary-query technique. Monitoring agents would be thus classifying a target as 'regular' or TOI on the basis of the (possibly weighted) average of the local assessments.

In the thesis some fundamental issues have been addressed, e.g. "Which granularity (i.e. level of detail) to adopt for the target representation?", "What is the most convenient approach to assessment decentralisation?", "How to represent articulated characteristics of a target behaviour?", "How to assess a target on the basis of multiple aspects of its conduct?", "How to detect anomalies in time-correlated behaviours?", etc.. The answers provided in the thesis to these questions are supported by numerical results from an extensive set of simulations, yet they cannot be considered "unquestionably true" because of the inherent subjectivity of the assessment process. For this reason, the fundamental problem to be addressed in future works is the identification of a technique that allow defining a somehow 'objective' GT. This would make it possible to actually validate the proposed approach, leading to performance indexes above criticism. Because of the subjective nature of the assessment process, a fully objective GT can maybe hardly be defined, yet the author's opinion is that some relevant steps in this direction can be done using a proper analytical approach.

A related but different problem encountered within the research project is that of performancecomparison against different assessment techniques. In this case, the subtlety consists in the fact that the underlying knowledge (that is, the subjective definition of what is regular and what is not) that defines the severity of the assessment must be the same for the two techniques considered, otherwise the comparison would not be meaningful. Two approaches could, indeed, be equally effective and yet identifying a different number of targets, since "tuned" to be severe to different extents. A relevant investigation could thus concern the 'severity equalisation' issue between different techniques.

The issues just described deal more with the "interpretation" of the assessment problem than with actual implementation details. However, with regard to the techniques presented in the thesis, future works could also be aimed at refining and extending the current functionalities.

A possible future work involves the hierarchic approach proposed for the features representation (Chapter 6). In the adopted implementation, the beta functions at any level m are defined on the basis of single numeric values, each of which is associated with a distribution located at level $m - 1$. It would be however extremely significative to develop an analytical approach for the definition of the highest-level distribution (the one associated with the actual feature) that take into account the full distributions at the lower level instead of single aggregating values.

Furthermore, an actual implementation for the dictionary-update process mentioned in Section 7.4.2.5 can be easily integrated into the framework assessment. The real, problem, again, is more "philosophical" than technical: how to guarantee that behaviours becoming frequent during the monitoring process can be assumed to be legit? In light traffic conditions, e.g., left overtaking manoeuvres are likely to be rarely observed, while their frequency could significantly rise under heavy traffic conditions: this, of course, does not entail that such behaviour should be tolerated. The availability of a reliable GT could mitigate this problem, providing feedbacks about the benefits resulting from such update action. The GT definition remains thus a binding, sensible issue, to be addressed before efficient and sophisticated technical solutions can be taken into account.

Similar considerations can be made regarding the integration of contextual information in the assessment process. This topic has been briefly addressed in the thesis, but its contribution has not been quantitatively described. Changes in the environment conditions reasonably entail a shift in the GT, but how this should alter the assessment process can hardly be evaluated in an objective manner.

Appendix A

Linear Algebra and Matrix Theory

Definition A.1. Matrix $A = \left[a_{ij}\right]$ is defined '*positive*' ($A > 0$) if $a_{ij} > 0, \forall i, j$.

Definition A.2. Matrix $A = \left[a_{ij}\right]$ is defined '*nonnegative*' ($A \geq 0$) if $a_{ij} \geq 0, \forall i, j$.

Definition A.3. Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is is said to be '*positive-definite*', '*positive-semidefinite*', '*negative-definite*' or '*negative-semidefinite*' if, respectively:

- $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0$
- $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0$
- $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} < 0$
- $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \leq 0$

for every non-zero column vector $\boldsymbol{x}$ composed of n real numbers.

Theorem A.1. *(Rank-Nullity Theorem) Let $\boldsymbol{V}$ and $\boldsymbol{W}$ be vector spaces over a field $\boldsymbol{F}$, and let $\boldsymbol{T} : \boldsymbol{V} \to \boldsymbol{W}$ be a linear transformation. Assuming the dimension of $\boldsymbol{V}$ is finite, then:*

$$\dim(\boldsymbol{V}) = \dim(\mathrm{Ker}(\boldsymbol{T})) + \dim(\mathrm{Im}(\boldsymbol{T})), \tag{A.1}$$

where dim *is the dimension of a vectorial space,* Ker *is the kernel, and* Im *is the image.*

Note that $\dim(\mathrm{Ker}(\boldsymbol{T}))$ is called the '*nullity*' of $\boldsymbol{T}$ and $\dim(\mathrm{Im}(\boldsymbol{T}))$ is denoted as the '*rank*' of $\boldsymbol{T}$.

Definition A.4. Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is said to be '*diagonally dominant*' if, for every row of the matrix, the magnitude of the diagonal entry is larger than or equal to the sum of the magnitudes of all the other (non-diagonal) entries in that row. More precisely, the matrix A is defined diagonally dominant if:

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \quad \forall i = 1, \ldots, n$$

Definition A.5. A nonnegative $n \times n$ matrix $\boldsymbol{A} = \left(a_{ij}\right)$ is defined '*row stochastic*' or '*right stochastic*' if

$$\sum_{j=1}^{n} a_{ij} = 1, \quad \forall i$$

A row stochastic matrix is referred, for the purpose of this thesis, simply as *stochastic* matrix.

Definition A.6. A nonnegative $n \times n$ matrix $\boldsymbol{A} = \left(a_{ij}\right)$ is defined '*column stochastic*' or '*left stochastic*' if:

$$\sum_{i=1}^{n} a_{ij} = 1, \quad \forall j$$

Definition A.7. A nonnegative $n \times n$ matrix $\boldsymbol{A} = \left(a_{ij}\right)$ is defined '*doubly stochastic*' if it is both row stochastic and column stochastic:

$$\sum_{j=1}^{n} a_{ij} = \sum_{i=1}^{n} a_{ij} = 1, \quad \forall (i, j)$$

Definition A.8. A nonnegative square matrix $\boldsymbol{A} = \left(a_{ij}\right)$ is said to be '*primitive*' if $\exists k \in \mathbb{N} : \boldsymbol{A}^k > 0$, i.e. if there exist a k such that all the elements of $\boldsymbol{A}^k$ are positive.

Definition A.9. A stochastic matrix $\boldsymbol{A}$ is defined '*Stochastic, Indecomposable and Aperiodic* (SIA)' if $\lim_{i \to \infty} \boldsymbol{A}^i$ is a matrix of rank 1, i.e.

$$\lim_{i \to \infty} \boldsymbol{A}^i = \mathbf{1}\boldsymbol{v}^T$$

where $\mathbf{1}$ is a column vector of ones, and $\boldsymbol{v}$ is a generic column vector, both of suitable dimension.

Definition A.10. An '*invariant subspace*' of a linear mapping $\boldsymbol{T} : \boldsymbol{V} \to \boldsymbol{V}$ from some vector space $\boldsymbol{V}$ to itself, is a subspace $\boldsymbol{W} \subseteq \boldsymbol{V}$ such that $\boldsymbol{T}(\boldsymbol{W}) \in \boldsymbol{W}$. An invariant subspace of $\boldsymbol{T}$ is also said to be T *invariant*.

Definition A.11. Given matrix $\boldsymbol{A}$, one of its eigenvalues λ and the associated eigenvector $\boldsymbol{w}$, is defined '*generalized eigenvector*' the vector $\boldsymbol{w}'$ such that:

$$\left(\boldsymbol{A} - \lambda \boldsymbol{I}\right) \boldsymbol{w}' = \boldsymbol{w}$$

where $\boldsymbol{I}$ is the identity matrix.

Appendix B

Graph Theory

Definition B.1. A '*Graph*' is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, ..v_n\}$ is a finite nonempty set of '*nodes*' (or '*vertices*') and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of '*edges*' (or '*arcs*') connecting the graph nodes. The pair $(i, j) \in \mathcal{E}$ if it exists an edge linking node i and j. Self edges are generally allowed, so pair (i, i) is a valid element of $\mathcal{E}$. The '*graph order*' n corresponds to the number of nodes in the graph, i.e. $n = |\mathcal{V}|$.

Definition B.2. A '*directed graph*' or '*digraph*' is a graph where edges are ordered pairs, i.e $(i, j) \in \mathcal{E}$ indicates that node j can be reached from node i, but not necessarily vice versa: in such case i is referred as '*parent node*' and j as '*child node*'.

Definition B.3. An '*undirected graph*' is one in which edges have no orientation. The edge (i, j) corresponds to the edge (j, i), i.e., they are not ordered pairs. This type of graph can be seen as a special case of a digraph, where $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$.

Definition B.4. A '*simple graph*' is a graph with no self-loop, i.e. $(i, j) \in \mathcal{E}$ only if $i \neq j$.

Definition B.5. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, it is defined '*adajacency matrix*' $\mathcal{A}^{'}$ of $\mathcal{G}$, the $n \times n$ matrix with elements

$$a^{'}_{ij} = \begin{cases} 1, (j, i) \in \mathcal{E} \\ 0, (j, i) \notin \mathcal{E} \end{cases}$$

where $n = |\mathcal{V}|$.

Definition B.6. The '*neighbor set*' of node v in graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as $N_v = \{j \in \mathcal{V} : (j, v) \in \mathcal{E}\}$.

Definition B.7. The '*degree*' or '*valence*' of a graph node v is the number of incident edges for that vertex, and it is referred as $\deg(v)$.

Definition B.8. The degree or valence of a node v of a digraph is the sum of the number of edges entering the node (‘*in-degree*’ $\deg_{in}(v)$) and edges exiting the node (‘*out-degree*’ $\deg_{out}(v)$).

Definition B.9. A node v of a digraph is defined ‘*topologically balanced*’ if it has the same in-degree and out-degree, i.e., $\deg_{in}(v) = \deg_{out}(v)$. Nodes in an undirected graph are topologically balanced by definition.

Definition B.10. A (di)graph is defined topologically balanced if each vertex $v_i \in \mathcal{V}$ is topologically balanced.

Definition B.11. a ‘*weighted (directed) graph*’ is a triple $(\mathcal{V}, \mathcal{E}, w)$ where $(\mathcal{V}, \mathcal{E})$ is a (di)graph and $w : \mathcal{E} \to \mathbb{R}_+$ is a map associating each edge (i, j) to a strictly positive weight w_{ij}. For undirected graphs, edges (i, j) and (j, i) are allowed to assume different weights.

The weights map w is usually expressed as a modified version of the adjacency matrix, i.e $\mathcal{A} = \left[a_{ij}\right]$ where $a_{ij} = w_{ij}\ \forall i, j : a_{ij}^{'} \neq 0$. In this thesis a weighted (directed) graph is thus referred as a triple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$. Must be noticed that the adjacency matrix introduced in definition B.5 is a particular case of the one just defined, where $w_{ij} = 1\ \forall i, j$. In this thesis, the adjacency/weight matrix $\mathcal{A}$ here defined is adopted.

Definition B.12. The ‘*weighted in-degree*’ and ‘*weighted out-degree*’ of a node v_i of a weighted digraph are respectively defined as $\deg_{in,w}(v_i) = \sum_{j=1}^{n} a_{ij}$ and $\deg_{out,w}(v_i) = \sum_{j=1}^{n} a_{ji}$.

Definition B.13. A node v in a weighted digraph is said to be ‘*weight-balanced*’ if $\deg_{in,w}(v) = \deg_{out,w}(v)$. Vertices of a weighted non-directed graph are weight-balanced by definition.

Definition B.14. A weighted (di)graph is said to be weight-balanced if each vertex $v_i \in \mathcal{V}$ is weight-balanced.

Definition B.15. a ‘*subgraph*’ $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is graph with $\mathcal{V}' \subset \mathcal{V}$ and $\mathcal{E}' \subset \left(\mathcal{E} \bigcap (\mathcal{V}' \times \mathcal{V}')\right)$.

Definition B.16. a ‘*spanning subgraph*’ $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a subgraph of $\mathcal{G}$ with $\mathcal{V} = \mathcal{V}'$.

Definition B.17. Given a collection of graphs $\{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_m\}$, is defined ‘*union*’ of the graphs in the collection, the graph $\mathcal{G}_u = (\mathcal{V}_u, \mathcal{E}_u)$, with $\mathcal{V}_u = \cup_{i=1}^{i=m} \mathcal{V}_i$ and $\mathcal{E}_u = \cup_{i=1}^{i=m} \mathcal{E}_i$.

Definition B.18. Given a (di)graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a ‘*(directed) path*’ between nodes v_{i_1} and v_{i_k} is a sequence of edges of the form $\left(\left(v_{i_1}, v_{i_2}\right), \left(v_{i_2}, v_{i_3}\right), \ldots, \left(v_{i_{k-1}}, v_{i_k}\right)\right)$ where $v_{i_j} \in \mathcal{V}\ \ \forall j$ and $(i_j, i_{j+1}) \in \mathcal{E}\ \ \forall j \neq k$.

Definition B.19. The '*geodesic distance*' $\mathrm{d}(v_i, v_j)$ between two nodes in a (di)graph is the number of edges in the shortest (directed) path connecting them.

Definition B.20. The '*eccentricity*' $\epsilon(v)$ of a vertex v in a (di)graph is the greatest geodesic distance between v and any other vertex. It can be thought of as how far a node is from the node most distant from it in the graph.

Definition B.21. The '*radius*' r of a (di)graph is the minimum eccentricity of any vertex, i.e. $r = \min_{v \in \mathcal{V}} \epsilon(v)$.

Definition B.22. The '*diameter*' d of a (di)graph is the maximum eccentricity of any vertex in the graph. That is, d it is the greatest distance between any pair of vertices: $d = \max_{v \in V} \epsilon(v)$.

Definition B.23. A '*cycle*' in a digraph is a non-trivial directed path that starts and ends at the same vertex.

Definition B.24. A node of a digraph is '*globally reachable*' if it can be reached from any other node by traversing a directed path.

Definition B.25. An '*Euler tour*' of a digraph G is a cycle that visits all the edges of the digraph exactly once. A digraph is '*Eulerian*' if it has an Euler tour.

Definition B.26. A (di)graph is '*(strongly) connected*' if there is a (directed) path between every pair of distinct nodes.

Definition B.27. A digraph is '*weakly connected*' if each pair of nodes is connected by an undirected path.

Definition B.28. A digraph is '*strongly semiconnected*' if the existence of a directed path from a vertex i to a vertex j implies the existence of a directed path from j to i. Strongly semiconnected digraphs that are also weakly connected must be strongly connected.

Definition B.29. The '*strongly connected components*' of a directed graph $\mathcal{G}$ are its maximal strongly connected subgraphs.

Definition B.30. A '*directed tree*' is a digraph where every node, except one special vertex without any parent (the '*root node*'), has exactly one parent, and the root vertex can be connected to any other node through directed paths

Definition B.31. A '*spanning tree*' of $\mathcal{G}$ is a spanning subgraph of $\mathcal{G}$ that is also a directed tree. This means that graph $\mathcal{G}$ has (or contains) a spanning tree if a subset of the edges in $\mathcal{E}$ forms a spanning tree.

Definition B.32. A '*(directed) graphs sequence* is a (di)graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ with a fixed set of nodes and a time-varying set of edges, where $t \in \mathbb{N}$.

Definition B.33. The '*weighted (directed) graphs sequence*' definition is obtained from definitions B.11 and B.32, i.e. is a triple $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t), \mathcal{A}(t))$ with time-varying edges and weights sets.

Given the nodes set $\mathcal{V}$, it can be useful to define the set of graphs that can be defined on $\mathcal{V}$, i.e $\overline{\mathcal{G}} = \{\mathcal{G}_i : \mathcal{G}_i = (\mathcal{V}, \mathcal{E}_i), \forall \mathcal{E}_i \subseteq (\mathcal{V} \times \mathcal{V})\}$

Definition B.34. A '*graph switching signal*' $\sigma(t)$ for a set of graphs $\overline{\mathcal{G}}$ defined on a common vertex set $\mathcal{V}$ is a function $\sigma : \mathbb{R} \to \mathcal{P}$, where $\mathcal{P} = \left\{i : \mathcal{G}_i \in \overline{\mathcal{G}}\right\}$ is the set of indices for graphs in the set $\overline{\mathcal{G}}$.

It must be noticed that the function σ has been defined on the graphs set $\overline{\mathcal{G}}$, previously introduced as the set of all possible graphs sharing a given nodes set $\mathcal{V}$. This has been done with the purpose of considering the most generic graphs set possible, but the function σ could be defined on any set collecting graphs that share a common node set, for example $\mathcal{Q} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_k\} \subset \overline{\mathcal{G}}$

Considering the function $\sigma(t)$ introduced in B.34, the definition for a graphs sequence (previously given in B.32 and B.32) can be reformulated as:

$$\mathcal{G}(t) = \mathcal{G}_{\sigma(t)} \triangleq \left(\mathcal{G}_i,\ i = \sigma(t)\right)$$

Definition B.35. The '*vertex connectivity*' of the (di)graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the minimum number of nodes $\kappa(G)$ whose deletion make the graph disconnected. The vertex connectivity is sometimes referred to as '*point connectivity*' or simply '*connectivity*'.

Definition B.36. A (di)graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be '*k-connected*' if there does not exist a set of $k-1$ vertices in $\mathcal{V}$ whose removal makes the graph disconnected, i.e., the vertex connectivity of $\mathcal{V}$ is $\geq k$.

Appendix C

NGSIM Details

C.1 NGSIM Data Structure

Each entry of the *Next Generation SIMulation* (NGSIM) dataset represents a single X,Y,T data point for a vehicle (specified by the vehicle identification number), with further associated information. Each data point is characterised by the following information fields:

1. Vehicle ID

   Vehicle identification number (ascending by time of entry into section)

2. Frame ID

   Frame Identification number (ascending by start time)

3. Total Frames

   Total number of frames in which the vehicle appears in this data set.

4. Global Time

   Elapsed time since Jan 1, 1970.

5. Local X

   Lateral (X) coordinate of the front centre of the vehicle with respect to the left-most edge of the section in the direction of travel.

6. Local Y

   Longitudinal (Y) coordinate of the front centre of the vehicle with respect to the entry edge of the section in the direction of travel

7. Global X

   X Coordinate of the front centre of the vehicle based on CA State Plane III in NAD83.

8. Global Y

   Y Coordinate of the front centre of the vehicle based on CA State Plane III in NAD83.

9. Vehicle Length

   Length of the vehicle.

10. Vehicle Width

    Width of the vehicle.

11. Vehicle Class

    Vehicle type: 1 - motorcycle, 2 - auto, 3 - truck.

12. Vehicle Velocity

    Instantaneous velocity of the vehicle.

13. Vehicle Acceleration

    Instantaneous acceleration of the vehicle.

14. Lane Identification

    Current lane position of the vehicle. Lane 1 is farthest left lane; lane 5 is farthest right lane. Lane 6 is the auxiliary lane between Ventura Boulevard on-ramp and the Cahuenga Boulevard off-ramp. Lane 7 is the on-ramp at Ventura Boulevard, and Lane 8 is the off-ramp at Cahuenga Boulevard.

15. Preceding Vehicle

    Vehicle Id of the lead vehicle in the same lane. A value of '0' represents no preceding vehicle - occurs at the end of the study section and off-ramp due to the fact that only complete trajectories were recorded by this data collection effort (vehicles already in the section at the start of the study period were not recorded).

16. Following Vehicle

    Vehicle Id of the vehicle following the subject vehicle in the same lane. A value of '0' represents no following vehicle - occurs at the beginning of the study section and on-ramp due to the fact that only complete trajectories were recorded by this data collection effort (vehicle that did not traverse the downstream boundaries of the section by the end of the study period were not recorded).

17. Spacing (Space Headway)

    Provides the distance between the front-centre of a vehicle to the front-centre of the preceding vehicle.

18. Headway (Time Headway)

    Provides the time to travel from the front-centre of a vehicle (at the speed of the vehicle) to the front-centre of the preceding vehicle. A headway value of 9999.99 means that the vehicle is traveling at zero speed (congested conditions).

C.2 NGSIM Data Reconstruction Process

The data-fixing procedure considered for correcting NGSIM data ([91]) is composed of four main steps:

1. Outliers Removal
   - Speed and acceleration profiles are obtained from position in time by successive differentiations (original speed and acceleration values are discarded);
   - Data points exhibiting an acceleration higher than a sensible threshold value (yet far higher than common vehicle capabilities, e.g.,) are identified and modified. This happens by replacing each identified position value with the one suggested by a spline interpolation of a given number of position values preceding and following the data point considered;
   - Velocity and acceleration profiles are then obtained by successive integration.
2. Medium- and High-Frequency Responses Removal Within the Speed Profile
   - Low-pass filtering of the current speed profile;
   - Definition of the position and acceleration profiles by integration and differentiation respectively.
3. Residual Unphysical Acceleration Values Removal
   - Identification of the data points exhibiting acceleration (in absolute value) above a threshold defined on the basis of the expected target dynamic;
   - For each of these points a time window allowing covering the same distance travelled in the current data but with a feasible acceleration profile is calculated;
   - The speed profile is reconstructed over the previously defined time-window. This is obtained by means of a polynomial interpolant defined accounting for some kinematic/dynamic constraints, for example:

  - Constraint on the space travelled;
  - Boundary condition on the derivatives;
  - Boundary condition on the sign inversions of the jerk over a given time-window.

4. Removal of the high- and Medium-frequency Responses Introduced at Step 3
   - The low pass filter introduced in Step 2 is reapplied for removing medium- and high-frequency components possibly introduced by the interpolation process (e.g. angles in the speed profile leading to discontinuities for the acceleration)
   - Definition of position and acceleration profiles by integration and differentiation respectively.

This approach to trajectory correction seems to perform better than a standard low-pass filtering for the following reasons:

- The preliminary correction of the outliers by means of spline interpolation allows adopting a wider-band low-pass filter (step 2), thus preserving quick dynamics of the vehicle (velocity profiles during stopping manoeuvres, gear shifting, etc.,)
- The proposed trajectory reconstruction approach allows removing unphysical values, guaranteeing at the same time internal consistency for the resulting trajectory (with reference to position, speed and acceleration profiles).

Bibliography

[1] Murray Aitkin. “The calibration of P-values, posterior Bayes factors and the AIC from the posterior distribution of the likelihood”. In: *Statistics and Computing* 7.4 (1997), pp. 253–261 (cit. on p. 87).

[2] Vassili Alexiadis et al. “The next generation simulation program”. In: *Institute of Transportation Engineers. ITE Journal* 74.8 (2004), p. 22 (cit. on p. 56).

[3] Donka Angelova and Lyudmila Mihaylova. “Sequential Monte Carlo algorithms for joint target tracking and classification using kinematic radar information”. In: (2004) (cit. on p. 32).

[4] P. Angkititrakul, R. Terashima, and T. Wakita. “On the Use of Stochastic Driver Behavior Model in Lane Departure Warning”. In: *IEEE Transactions on Intelligent Transportation Systems* 12.1 (2011), pp. 174–183 (cit. on p. 107).

[5] Frank J Anscombe. “Rejection of outliers”. In: *Technometrics* 2.2 (1960), pp. 123–146 (cit. on p. 30).

[6] Y. Bar-Shalom, F. Daum, and J. Huang. “The probabilistic data association filter”. In: *Control Systems, IEEE* 29.6 (2009), pp. 82–100 (cit. on p. 23).

[7] Sabyasachi Basu and Martin Meckesheimer. “Automatic outlier detection for time series: an application to sensor data”. In: *Knowledge and Information Systems* 11.2 (2007), pp. 137–154 (cit. on p. 30).

[8] D. Bauso, L. Giarr, and R. Pesenti. “Non-linear protocols for optimal distributed consensus in networks of dynamic agents”. In: *Systems & Control Letters* 55.11 (2006), pp. 918 –928 (cit. on pp. 27, 29).

[9] J.M. Beaver, R.A. Kerekes, and J.N. Treadwell. “An information fusion framework for threat assessment”. In: *Information Fusion, 2009. FUSION ’09. 12th International Conference on.* 2009, pp. 1903–1910 (cit. on p. 36).

[10] A. Benavoli et al. “An Application of Evidential Networks to Threat Assessment”. In: *Aerospace and Electronic Systems, IEEE Transactions on* 45.2 (2009), pp. 620–639 (cit. on p. 36).

[11] Jon Atli Benediktsson and Philip H Swain. "Consensus theoretic classification methods". In: *IEEE transactions on Systems, Man, and Cybernetics* 22.4 (1992), pp. 688–704 (cit. on p. 3).

[12] Derek J Bennet et al. "Autonomous Three-Dimensional Formation Flight for a Swarm of Unmanned Aerial Vehicles". In: *Journal of Guidance, Control, and Dynamics* 34.6 (2011), pp. 1899–1908 (cit. on p. 3).

[13] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013 (cit. on p. 87).

[14] Norman Biggs. *Algebraic graph theory*. Cambridge University Press, 1993 (cit. on p. 40).

[15] Christopher M Bishop. "Novelty detection and neural network validation". In: *Vision, Image and Signal Processing, IEE Proceedings-*. Vol. 141. 4. IET. 1994, pp. 217–222 (cit. on p. 32).

[16] D Blacknell. "Contextual information in SAR target detection". In: *IEE Proceedings-Radar, Sonar and Navigation* 148.1 (2001), pp. 41–47 (cit. on p. 93).

[17] Richard J Bolton and David J Hand. "Statistical fraud detection: A review". In: *Statistical science* (2002), pp. 235–249 (cit. on p. 32).

[18] Richard Bradley. "Reaching a consensus". In: *Social choice and welfare* 29.4 (2007), pp. 609–632 (cit. on p. 3).

[19] Yongcan Cao et al. "Finite-time consensus of multi-agent networks with inherent nonlinear dynamics under an undirected interaction graph". In: *American Control Conference (ACC), 2011*. 2011, pp. 4020–4025 (cit. on pp. 26, 29).

[20] R. Carli et al. "Distributed Kalman filtering based on consensus strategies". In: *Selected Areas in Communications, IEEE Journal on* 26.4 (2008), pp. 622–633 (cit. on p. 21).

[21] Philip K Chan and Salvatore J Stolfo. "Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection." In: *KDD*. Vol. 1998. 1998, pp. 164–168 (cit. on p. 30).

[22] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15 (cit. on p. 32).

[23] JS Chen and EK Walton. "Comparison of two target classification techniques". In: *Aerospace and Electronic Systems, IEEE Transactions on* 1 (1986), pp. 15–22 (cit. on p. 32).

[24] Tianping Chen, Xiwei Liu, and Wenlian Lu. "Pinning Complex Networks by a Single Controller". In: *Circuits and Systems I: Regular Papers, IEEE Transactions on* 54.6 (2007), pp. 1317–1326 (cit. on p. 22).

[25] Han-Lim Choi, L. Brunet, and J.P. How. "Consensus-Based Decentralized Auctions for Robust Task Allocation". In: *Robotics, IEEE Transactions on* 25.4 (2009), pp. 912–926 (cit. on pp. 20, 29).

[26] Jongug Choi and Yudan Kim. "Fuel efficient three dimensional controller for leader-follower UAV formation flight". In: *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*. 2007, pp. 806–811 (cit. on p. 3).

[27] Benjamin A Coifman and Ramachandran Mallika. "Distributed surveillance on freeways emphasizing incident detection and verification". In: *Transportation research part A: policy and practice* 41.8 (2007), pp. 750–767 (cit. on pp. 30, 34).

[28] Jorge Cortés. "Distributed algorithms for reaching consensus on general functions". In: *Automatica* 44.3 (2008), pp. 726 –737 (cit. on pp. 6, 7, 26, 29).

[29] Jorge Corts. "Finite-time convergent gradient flows with applications to network consensus". In: *Automatica* 42.11 (2006), pp. 1993 –2000 (cit. on pp. 26, 29).

[30] Morris H DeGroot. "Reaching a consensus". In: *Journal of the American Statistical Association* 69.345 (1974), pp. 118–121 (cit. on p. 3).

[31] Arthur P Dempster. "The direct use of likelihood for significance testing". In: *Statistics and Computing* 7.4 (1997), pp. 247–252 (cit. on p. 87).

[32] Dimos V Dimarogonas et al. "A feedback stabilization and collision avoidance scheme for multiple independent non-point agents". In: *Automatica* 42.2 (2006), pp. 229–243 (cit. on p. 5).

[33] Ji Ding and Qiongjian Fan. "A multi-UAV tight formation flight controller". In: *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*. Vol. 1. 2012, pp. 60–64 (cit. on p. 3).

[34] D Dolev and HR Strong. "Distributed commit with bounded waiting". In: *Proc. of the 2nd Symp. on Reliability in Distributed Software and Database Systems*. 1982, pp. 53–60 (cit. on p. 4).

[35] Qian Du and James E Fowler. "Hyperspectral image compression using JPEG2000 and principal component analysis". In: *Geoscience and Remote Sensing Letters, IEEE* 4.2 (2007), pp. 201–205 (cit. on p. 30).

[36] John Eidson and Kang Lee. "IEEE 1588 standard for a precision clock synchronization protocol for networked measurement and control systems". In: *Sensors for Industry Conference, 2002. 2nd ISA/IEEE*. IEEE. 2002, pp. 98–105 (cit. on p. 5).

[37] Jeremy Elson, Lewis Girod, and Deborah Estrin. "Fine-grained network time synchronization using reference broadcasts". In: *ACM SIGOPS Operating Systems Review* 36.SI (2002), pp. 147–163 (cit. on p. 5).

[38] Eleazar Eskin et al. “A geometric framework for unsupervised anomaly detection”. In: *Applications of data mining in computer security*. Springer, 2002, pp. 77–101 (cit. on p. 30).

[39] A. Fagiolini, S. Martini, and A. Bicchi. “Set-valued consensus for distributed clock synchronization”. In: *Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on*. 2009, pp. 116–121 (cit. on pp. 7, 25, 29).

[40] A. Fagiolini, E.M. Visibelli, and A. Bicchi. “Logical consensus for distributed network agreement”. In: *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. 2008, pp. 5250–5255 (cit. on pp. 6, 7, 25, 26, 29, 50).

[41] A. Fagiolini et al. “A self-routing protocol for distributed consensus on logical information”. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. 2010, pp. 5151–5156 (cit. on pp. 6, 7, 26, 29, 50).

[42] A. Fagiolini et al. “Consensus-based distributed intrusion detection for multi-robot systems”. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. 2008, pp. 120–127 (cit. on pp. 7, 24, 29, 49).

[43] A. Fagiolini et al. “Decentralized intrusion detection for secure cooperative multi-agent systems”. In: *Decision and Control, 2007 46th IEEE Conference on*. 2007, pp. 1553–1558 (cit. on p. 49).

[44] Tom Fawcett and Foster Provost. “Activity monitoring: Noticing interesting changes in behavior”. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 1999, pp. 53–62 (cit. on p. 30).

[45] Tom Fawcett and Foster Provost. “Adaptive Fraud Detection”. In: *Data Mining and Knowledge Discovery* 1.3 (1997), pp. 291–316 (cit. on pp. 30, 31).

[46] J Alexander Fax. “Optimal and cooperative control of vehicle formations”. PhD thesis. California Institute of Technology, 2001 (cit. on p. 5).

[47] J.A. Fax and R.M. Murray. “Information flow and cooperative control of vehicle formations”. In: *Automatic Control, IEEE Transactions on* 49.9 (2004), pp. 1465–1476 (cit. on pp. 5, 20).

[48] Xiao Feng, Wang Long, and Chen Tongwen. “Finite-time consensus of multi-agent systems with directed and intermittent links”. In: *Control Conference (CCC), 2011 30th Chinese*. 2011, pp. 6539–6543 (cit. on pp. 26, 29).

[49] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. “Impossibility of distributed consensus with one faulty process”. In: *Journal of the ACM (JACM)* 32.2 (1985), pp. 374–382 (cit. on p. 27).

[50] Stephanie Forrest et al. "A sense of self for unix processes". In: *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*. IEEE. 1996, pp. 120–128 (cit. on p. 31).

[51] Zhouyu Fu, Weiming Hu, and Tieniu Tan. "Similarity based vehicle trajectory clustering and anomaly detection". In: *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. Vol. 2. IEEE. 2005, pp. II–602 (cit. on pp. 30, 31).

[52] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. "Timing-sync protocol for sensor networks". In: *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM. 2003, pp. 138–149 (cit. on p. 5).

[53] Hector Garcia-Molina. "Elections in a distributed computing system". In: *Computers, IEEE Transactions on* 100.1 (1982), pp. 48–59 (cit. on p. 4).

[54] Arati Gerdes. "Automatic maneuver recognition in the automobile: the fusion of uncertain sensor values using bayesian models". In: *Proceedings of the 3rd International Workshop on Intelligent Transportation (WIT 2006)*. 2006, pp. 129–133 (cit. on p. 107).

[55] F. Giulietti, L. Pollini, and M. Innocenti. "Autonomous formation flight". In: *Control Systems, IEEE* 20.6 (2000), pp. 34–44 (cit. on p. 3).

[56] Fabrizio Giulietti, Lorenzo Pollini, Mario Innocenti, et al. "Formation flight control: A behavioral approach". In: *AIAA paper* 4239 (2001) (cit. on p. 3).

[57] M. Goldenbaum, Holger Boche, and S. Stanczak. "Nomographic gossiping for f-consensus". In: *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012 10th International Symposium on*. 2012, pp. 130–137 (cit. on pp. 7, 27).

[58] Roman Goldenberg et al. "Behavior classification by eigendecomposition of periodic motions". In: *Pattern Recognition* 38.7 (2005), pp. 1033 –1043 (cit. on p. 32).

[59] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 1985 (cit. on pp. 40, 42).

[60] Jonathan P. How et al. "Dynamic Mission Planning for Communication Control in Multiple Unmanned Aircraft Teams". In: *Unmanned Systems* 01.01 (2013), pp. 41–58 (cit. on p. 20).

[61] T. Hlnhagen et al. "Maneuver recognition using probabilistic finite-state machines and fuzzy logic". In: *Intelligent Vehicles Symposium (IV), 2010 IEEE*. 2010, pp. 65–70 (cit. on p. 107).

[62] A. Jadbabaie, Jie Lin, and A.S. Morse. “Coordination of groups of mobile autonomous agents using nearest neighbor rules”. In: *Automatic Control, IEEE Transactions on* 48.6 (2003), pp. 988–1001 (cit. on pp. 5, 19, 20, 29, 45).

[63] Ahmed T. Kamal, Jay A. Farrell, and Amit K. Roy-Chowdhury. “Information Consensus for Distributed Multi-target Tracking”. In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. 2013, pp. 2403–2410 (cit. on p. 23).

[64] A.T. Kamal, J.A. Farrell, and A.K. Roy-Chowdhury. “Information weighted consensus”. In: *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. 2012, pp. 2732–2737 (cit. on pp. 23, 29).

[65] S. Kamijo et al. “Traffic monitoring and accident detection at intersections”. In: *IEEE Transactions on Intelligent Transportation Systems* 1.2 (2000), pp. 108–118 (cit. on p. 35).

[66] Christopher Kruegel et al. “Bayesian event classification for intrusion detection”. In: *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*. IEEE. 2003, pp. 14–23 (cit. on p. 30).

[67] Gerardo Lafferriere et al. “Decentralized control of vehicle formations”. In: *Systems & Control Letters* 54.9 (2005), pp. 899–910 (cit. on pp. 5, 29, 40).

[68] Butler Lampson and Howard Sturgis. *Crash recovery in a distributed data storage system*. Xerox Palo Alto Research Center Palo Alto, California, 1979 (cit. on p. 4).

[69] R.O. Lane et al. “Maritime anomaly detection and threat assessment”. In: *Information Fusion (FUSION), 2010 13th Conference on*. 2010, pp. 1–8 (cit. on p. 35).

[70] Jonathan R Lawton and Randal W Beard. “Synchronized multiple spacecraft rotations”. In: *Automatica* 38.8 (2002), pp. 1359–1364 (cit. on p. 5).

[71] Jonathan RT Lawton, Randal W Beard, and Brett J Young. “A decentralized approach to formation maneuvers”. In: *Robotics and Automation, IEEE Transactions on* 19.6 (2003), pp. 933–941 (cit. on p. 5).

[72] Dah-Jye Lee et al. “Shape-based human detection for threat assessment”. In: vol. 5438. 2004, pp. 81–91 (cit. on p. 32).

[73] N.E. Leonard and E. Fiorelli. “Virtual leaders, artificial potentials and coordinated control of groups”. In: *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*. Vol. 3. 2001, 2968–2973 vol.3 (cit. on p. 3).

[74] Yihua Liao and V Rao Vemuri. “Use of k-nearest neighbor classifier for intrusion detection”. In: *Computers & Security* 21.5 (2002), pp. 439–448 (cit. on pp. 30, 31).

[75] Michael J Liebhaber and Bela Feher. *Air threat assessment: Research, model, and display guidelines*. Tech. rep. DTIC Document, 2002 (cit. on p. 32).

[76] Jie Lin, A Stephen Morse, and Brian DO Anderson. "The multi-agent rendezvous problem". In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*. Vol. 2. IEEE. 2003, pp. 1508–1513 (cit. on p. 5).

[77] Jie Lin, A Stephen Morse, and Brian DO Anderson. "The multi-agent rendezvous problem-the asynchronous case". In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 2. IEEE. 2004, pp. 1926–1931 (cit. on p. 5).

[78] Peng Lin and Yingmin Jia. "Consensus of second-order discrete-time multi-agent systems with nonuniform time-delays and dynamically changing topologies". In: *Automatica* 45.9 (2009), pp. 2154 –2158 (cit. on p. 29).

[79] Peng Lin et al. "Distributed leadless coordination for networks of second-order agents with time-delay on switching topology". In: *American Control Conference, 2008*. June 2008, pp. 1564–1569 (cit. on p. 29).

[80] Zhiyun Lin, Bruce Francis, and Manfredi Maggiore. "Necessary and sufficient graphical conditions for formation control of unicycles". In: *Automatic Control, IEEE Transactions on* 50.1 (2005), pp. 121–127 (cit. on p. 5).

[81] Alan J Lipton, Hironobu Fujiyoshi, and Raju S Patil. "Moving target classification and tracking from real-time video". In: *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*. IEEE. 1998, pp. 8–14 (cit. on p. 32).

[82] Gurcan Lokman and Gurkan Yilmaz. "A new method for anomaly detection and target recognition". In: *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE. 2014, pp. 577–583 (cit. on p. 30).

[83] Carl G Looney and Lily R Liang. "Cognitive situation and threat assessments of ground battlespaces". In: *Information Fusion* 4.4 (2003), pp. 297–308 (cit. on p. 32).

[84] Nancy A Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996 (cit. on p. 4).

[85] L. Malta et al. "Analysis of Real-World Driver's Frustration". In: *Intelligent Transportation Systems, IEEE Transactions on* 12.1 (2011), pp. 109–118 (cit. on pp. 33, 34).

[86] Markos Markou and Sameer Singh. "Novelty detection: a reviewpart 1: statistical approaches". In: *Signal processing* 83.12 (2003), pp. 2481–2497 (cit. on p. 32).

[87] Markos Markou and Sameer Singh. "Novelty detection: a reviewpart 2:: neural network based approaches". In: *Signal processing* 83.12 (2003), pp. 2499–2521 (cit. on p. 32).

[88] Sonia Martınez, Jorge Cortes, and Francesco Bullo. “On robust rendezvous for mobile autonomous agents”. In: *16th IFAC World Congress, Praha, CZ.* 2005 (cit. on p. 5).

[89] Keith Ansel Marzullo. “Maintaining the time in a distributed system: An example of a loosely-coupled distributed service”. PhD thesis. Stanford University, 1984 (cit. on p. 25).

[90] S. Modi et al. “A Socially Inspired Framework for Human State Inference Using Expert Opinion Integration”. In: *Mechatronics, IEEE/ASME Transactions on* 16.5 (2011), pp. 874–878 (cit. on p. 33).

[91] Marcello Montanino and Vincenzo Punzo. “Making NGSIM data usable for studies on traffic flow theory: Multistep method for vehicle trajectory reconstruction”. In: *Transportation Research Record: Journal of the Transportation Research Board* 2390 (2013), pp. 99–111 (cit. on pp. 58, 104, 128, 145).

[92] L. Moreau. “Stability of continuous-time distributed consensus algorithms”. In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on.* Vol. 4. 2004, 3998–4003 Vol.4 (cit. on p. 29).

[93] L. Moreau. “Stability of multiagent systems with time-dependent communication links”. In: *Automatic Control, IEEE Transactions on* 50.2 (2005), pp. 169–182 (cit. on pp. 5, 19, 20, 29, 45).

[94] Nima Moshtagh and Ali Jadbabaie. “Distributed geodesic control laws for flocking of nonholonomic agents”. In: *Automatic Control, IEEE Transactions on* 52.4 (2007), pp. 681–686 (cit. on p. 5).

[95] George Nychis et al. “An empirical evaluation of entropy-based traffic anomaly detection”. In: *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement.* ACM. 2008, pp. 151–156 (cit. on pp. 30, 32).

[96] Cheol Oh, Jun-Seok Oh, and Stephen G Ritchie. “Real-time hazardous traffic condition warning system: framework and evaluation”. In: *Intelligent Transportation Systems, IEEE Transactions on* 6.3 (2005), pp. 265–272 (cit. on pp. 30, 34).

[97] Hyondong Oh. “Towards autonomous surveillance and tracking by multiple UAVs”. In: (2013) (cit. on p. 109).

[98] Hyondong Oh et al. “Airborne behaviour monitoring using Gaussian processes with map information”. In: *IET Radar, Sonar & Navigation* 7.4 (2013), pp. 393–400 (cit. on p. 54).

[99] Hyondong Oh et al. “Behaviour recognition of ground vehicle using airborne monitoring of unmanned aerial vehicles”. In: *International Journal of Systems Science* 45.12 (2014) (cit. on pp. 8, 33, 107–110).

[100] R. Olfati-Saber. “Distributed Kalman Filter with Embedded Consensus Filters”. In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC ’05. 44th IEEE Conference on*. 2005, pp. 8179–8184 (cit. on pp. 5, 21, 22, 29).

[101] R. Olfati-Saber. “Distributed Kalman filtering for sensor networks”. In: *Decision and Control, 2007 46th IEEE Conference on*. 2007, pp. 5492–5498 (cit. on pp. 21, 23, 29).

[102] R. Olfati-Saber. “Distributed Tracking for Mobile Sensor Networks with Information-Driven Mobility”. In: *American Control Conference, 2007. ACC ’07*. 2007, pp. 4606–4612 (cit. on p. 23).

[103] R. Olfati-Saber. “Flocking for multi-agent dynamic systems: algorithms and theory”. In: *Automatic Control, IEEE Transactions on* 51.3 (2006), pp. 401–420 (cit. on pp. 5, 23, 29).

[104] R. Olfati-Saber. “Kalman-Consensus Filter : Optimality, stability, and performance”. In: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. 2009, pp. 7036–7042 (cit. on pp. 21–23, 29).

[105] R. Olfati-Saber, J.A. Fax, and R.M. Murray. “Consensus and Cooperation in Networked Multi-Agent Systems”. In: *Proceedings of the IEEE* 95.1 (2007), pp. 215–233 (cit. on pp. 7, 20, 29, 41, 43–46, 52).

[106] R. Olfati-Saber and P. Jalalkamali. “Collaborative target tracking using distributed Kalman filtering on mobile sensor networks”. In: *American Control Conference (ACC), 2011*. 2011, pp. 1100–1105 (cit. on pp. 23, 29).

[107] R. Olfati-Saber and R.M. Murray. “Consensus problems in networks of agents with switching topology and time-delays”. In: *Automatic Control, IEEE Transactions on* 49.9 (2004), pp. 1520–1533 (cit. on pp. 5, 7, 19, 20, 29, 40).

[108] R. Olfati-Saber and N.F. Sandell. “Distributed tracking in sensor networks with limited sensing range”. In: *American Control Conference, 2008*. 2008, pp. 3157–3162 (cit. on p. 23).

[109] R. Olfati-Saber and J.S. Shamma. “Consensus Filters for Sensor Networks and Distributed Sensor Fusion”. In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC ’05. 44th IEEE Conference on*. 2005, pp. 6698–6703 (cit. on pp. 5, 21, 22, 29).

[110] Reza Olfati-saber et al. “Belief consensus and distributed hypothesis testing in sensor networks”. In: *Network Embedded Sensing and Control. (Proceedings of NESC05 Worskhop), volume 331 of Lecture Notes in Control and Information Sciences*. Springer Verlag, 2006, pp. 169–182 (cit. on pp. 6, 7, 24, 29, 47, 50, 52).

[111] Barrett O'neill. *Elementary differential geometry.* Academic press, 2006 (cit. on p. 108).

[112] Meir Pachter et al. "Tight formation flight control". In: *Journal of Guidance, Control, and Dynamics* 24.2 (2001), pp. 246–254 (cit. on p. 3).

[113] Animesh Patcha and Jung-Min Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends". In: *Computer networks* 51.12 (2007), pp. 3448–3470 (cit. on p. 32).

[114] A. Petitti et al. "Consensus-based distributed estimation for target tracking in heterogeneous sensor networks". In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on.* 2011, pp. 6648–6653 (cit. on pp. 6, 23, 29).

[115] Sameera S. Ponda. "Robust distributed planning strategies for autonomous multi-agent teams". PhD thesis. Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics, Aug. 2012 (cit. on p. 20).

[116] Vincenzo Punzo, Maria Teresa Borzacchiello, and Biagio Ciuffo. "On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data". In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pp. 1243–1262 (cit. on p. 58).

[117] Jiahu Qin, Huijun Gao, and Wei Xing Zheng. "Second-order consensus for multi-agent systems with switching topology and communication delay". In: *Systems & Control Letters* 60.6 (2011), pp. 390 –397 (cit. on p. 29).

[118] B.S.Y. Rao, H.F. Durrant-Whyte, and J.A. Sheen. "A Fully Decentralized Multi-Sensor System For Tracking and Surveillance". In: *The International Journal of Robotics Research* 12.1 (1993), pp. 20–44 (cit. on p. 22).

[119] Carl Edward Rasmussen. "Gaussian processes for machine learning". In: (2006) (cit. on pp. 65, 116).

[120] Gunnar Rätsch et al. "Constructing boosting algorithms from SVMs: an application to one-class classification". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.9 (2002), pp. 1184–1199 (cit. on pp. 30, 31).

[121] Wei Ren. "Consensus based formation control strategies for multi-vehicle systems". In: *American Control Conference, 2006.* 2006, 6 pp.– (cit. on p. 3).

[122] Wei Ren. "Consensus based formation control strategies for multi-vehicle systems". In: *American Control Conference, 2006.* IEEE. 2006, 6–pp (cit. on pp. 5, 27, 29).

[123] Wei Ren. "Consensus Tracking Under Directed Interaction Topologies: Algorithms and Experiments". In: *Control Systems Technology, IEEE Transactions on* 18.1 (2010), pp. 230–237 (cit. on p. 22).

[124] Wei Ren. "Distributed attitude consensus among multiple networked spacecraft". In: *American Control Conference, 2006*. IEEE. 2006, 6–pp (cit. on pp. 5, 29).

[125] Wei Ren. "Second-order Consensus Algorithm with Extensions to Switching Topologies and Reference Models". In: *American Control Conference, 2007. ACC '07*. 2007, pp. 1431–1436 (cit. on p. 29).

[126] Wei Ren and Ella Atkins. "Distributed multi-vehicle coordinated control via local information exchange". In: *International Journal of Robust and Nonlinear Control* 17.10-11 (2007), pp. 1002–1033 (cit. on pp. 22, 29).

[127] Wei Ren and Ella Atkins. "Second-order Consensus Protocols in Multiple Vehicle Systems with Local Interactions". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2005 (cit. on pp. 8, 22, 29).

[128] Wei Ren and Randal Beard. "Decentralized scheme for spacecraft formation flying via the virtual structure approach". In: *Journal of Guidance, Control, and Dynamics* 27.1 (2004), pp. 73–82 (cit. on p. 5).

[129] Wei Ren, Randal W Beard, and Timothy W McLain. "Coordination variables and consensus building in multiple vehicle systems". In: *Cooperative Control*. Springer, 2005, pp. 171–188 (cit. on p. 29).

[130] Wei Ren and R.W. Beard. "Consensus seeking in multiagent systems under dynamically changing interaction topologies". In: *Automatic Control, IEEE Transactions on* 50.5 (2005), pp. 655–661 (cit. on pp. 5, 20, 29, 45).

[131] Wei Ren, R.W. Beard, and E.M. Atkins. "Information consensus in multivehicle cooperative control". In: *Control Systems, IEEE* 27.2 (2007), pp. 71–82 (cit. on pp. 7, 20, 29, 45, 52).

[132] Wei Ren, K. Moore, and Yang-Quan Chen. "High-Order Consensus Algorithms in Cooperative Vehicle Systems". In: *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*. 2006, pp. 457–462 (cit. on p. 29).

[133] WEI REN, Kevin L MOORE, and YANGQUAN CHEN. "High-order and model reference consensus algorithms in cooperative control of MultiVehicle systems". In: *Journal of dynamic systems, measurement, and control* 129.5 (2007), pp. 678–688 (cit. on pp. 22, 29).

[134] Craig W. Reynolds. "Flocks, Herds and Schools: A Distributed Behavioral Model". In: *SIGGRAPH Comput. Graph.* 21.4 (Aug. 1987), pp. 25–34 (cit. on p. 5).

[135] G. Rigas, Y. Goletsis, and D.I. Fotiadis. "Real-Time Driver's Stress Event Detection". In: *Intelligent Transportation Systems, IEEE Transactions on* 13.1 (2012), pp. 221–234 (cit. on p. 33).

[136] M RIGOLLI and M BRADY. *Towards a behavioural traffic monitoring system, AAMAS'05: Proceedings of the fourth international joint conference on Autonomous agents and multi-agent systems, 2005.* 2005 (cit. on p. 33).

[137] Branko Ristic, Neil Gordon, and Amanda Bessell. "On target classification using kinematic data". In: *Information Fusion* 5.1 (2004), pp. 15–21 (cit. on p. 32).

[138] A.B. Rodriguez Gonzalez et al. "Modeling and Detecting Aggressiveness From Driving Signals". In: *Intelligent Transportation Systems, IEEE Transactions on* 15.4 (2014), pp. 1419–1428 (cit. on p. 35).

[139] R.O. Saber and R.M. Murray. "Consensus protocols for networks of dynamic agents". In: *American Control Conference, 2003. Proceedings of the 2003.* Vol. 2. 2003, pp. 951–956 (cit. on pp. 5, 29, 39, 52).

[140] D. Sandberg et al. "Detecting Driver Sleepiness Using Optimized Nonlinear Combinations of Sleepiness Indicators". In: *Intelligent Transportation Systems, IEEE Transactions on* 12.1 (2011), pp. 97–108 (cit. on pp. 33, 34).

[141] Javier Sanz, Ricardo Perera, and Consuelo Huerta. "Fault diagnosis of rotating machinery based on auto-associative neural networks and wavelet transforms". In: *Journal of Sound and Vibration* 302.4 (2007), pp. 981–999 (cit. on p. 30).

[142] Shay Segal, Joseph Z Ben-Asher, and Haim Weiss. "Derivation of formation-flight guidance laws for unmanned air vehicles". In: *Journal of Guidance, Control, and Dynamics* 28.4 (2005), pp. 733–742 (cit. on p. 3).

[143] Pete Seiler, Aniruddha Pant, and Karl Hedrick. "Analysis of bird formations". In: *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on.* Vol. 1. IEEE. 2002, pp. 118–123 (cit. on p. 3).

[144] Reetinder Sidhu and Viktor K Prasanna. "Fast regular expression matching using FPGAs". In: *Field-Programmable Custom Computing Machines, 2001. FCCM'01. The 9th Annual IEEE Symposium on.* IEEE. 2001, pp. 227–238 (cit. on p. 108).

[145] Demetri P Spanos, Reza Olfati-Saber, and Richard M Murray. "Distributed sensor fusion using dynamic consensus". In: *IFAC World Congress.* 2005 (cit. on p. 5).

[146] Demetri P Spanos, Reza Olfati-Saber, and Richard M Murray. "Dynamic consensus on mobile networks". In: *The 16th IFAC World Congress, Prague, Czech.* 2005 (cit. on p. 21).

[147] Clay Spence, Lucas Parra, and Paul Sajda. "Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model". In: *Mathematical Methods in Biomedical Image Analysis, 2001. MMBIA 2001. IEEE Workshop on.* IEEE. 2001, pp. 3–10 (cit. on p. 30).

[148] Jason L. Speyer. "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem". In: *Automatic Control, IEEE Transactions on* 24.2 (1979), pp. 266–269 (cit. on p. 21).

[149] Michio Sugeno. *Industrial Applications of Fuzzy Control.* New York, NY, USA: Elsevier Science Inc., 1985 (cit. on p. 61).

[150] Min-Jea Tahk, Chang-Su Park, and Chang-Kyung Ryoo. "Line-of-sight guidance laws for formation flight". In: *Journal of Guidance, Control, and Dynamics* 28.4 (2005), pp. 708–716 (cit. on p. 3).

[151] Herbert G Tanner, Ali Jadbabaie, and George J Pappas. "Stable flocking of mobile agents, Part I: Fixed topology". In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on.* Vol. 2. IEEE. 2003, pp. 2010–2015 (cit. on p. 5).

[152] H.G. Tanner and D.K. Christodoulakis. "State synchronization in local-interaction networks is robust with respect to time delays". In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on.* 2005, pp. 4945–4950 (cit. on p. 29).

[153] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. "Stable flocking of mobile agents part I: dynamic topology". In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on.* Vol. 2. 2003, 2016–2021 Vol.2 (cit. on p. 5).

[154] Suttipong Thajchayapong, Edgar S Garcia-Trevino, and Javier A Barria. "Distributed classification of traffic anomalies using microscopic traffic variables". In: *Intelligent Transportation Systems, IEEE Transactions on* 14.1 (2013), pp. 448–458 (cit. on pp. 30, 35).

[155] S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Elsevier Science, 2008 (cit. on p. 113).

[156] Philip HS Torr and David W Murray. "Outlier detection and motion segmentation". In: *Optical Tools for Manufacturing and Advanced Automation.* International Society for Optics and Photonics. 1993, pp. 432–443 (cit. on pp. 30, 31).

[157] Dario Turchi, Hyo-Sang Shin, and Antonios Tsourdos. "Airborne Behaviour Monitoring Using Regular-Expressions Based Pattern Matching". In: *IFAC-PapersOnLine* 48.5 (2015), pp. 29–34 (cit. on pp. 63, 108).

[158] U.S. Department of Trasportation, FHWA. *NGSIM-Next Generation SIMulation.* 1999. (Visited on 06/10/2016) (cit. on p. 56).

[159] JJP Veerman et al. "Flocks and formations". In: *Journal of Statistical Physics* 121.5-6 (2005), pp. 901–936 (cit. on p. 5).

[160] Tamás Vicsek et al. "Novel type of phase transition in a system of self-driven particles". In: *Physical Review Letters* 75.6 (1995), p. 1226 (cit. on p. 5).

[161] Tamás Vicsek et al. "Novel Type of Phase Transition in a System of Self-Driven Particles". In: *Phys. Rev. Lett.* 75 (6 Aug. 1995), pp. 1226–1229 (cit. on p. 19).

[162] Arno Wagner and Bernhard Plattner. "Entropy based worm and anomaly detection in fast IP networks". In: *Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on.* IEEE. 2005, pp. 172–177 (cit. on p. 30).

[163] Robert A. Wagner and Michael J. Fischer. "The String-to-String Correction Problem". In: *J. ACM* 21.1 (1974), pp. 168–173 (cit. on p. 125).

[164] Long Wang and Feng Xiao. "Finite-Time Consensus Problems for Networks of Dynamic Agents". In: *Automatic Control, IEEE Transactions on* 55.4 (2010), pp. 950–955 (cit. on pp. 26, 29).

[165] Xiao Fan Wang and Guanrong Chen. "Pinning control of scale-free dynamical networks". In: *Physica A: Statistical Mechanics and its Applications* 310.34 (2002), pp. 521 –531 (cit. on pp. 22, 32).

[166] Christina Warrender, Stephanie Forrest, and Barak Pearlmutter. "Detecting intrusions using system calls: Alternative data models". In: *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on.* IEEE. 1999, pp. 133–145 (cit. on p. 31).

[167] Susan C Weller and N Clay Mann. "Assessing rater performance without a" gold standard" using consensus theory". In: *Medical Decision Making* 17.1 (1997), pp. 71–79 (cit. on p. 3).

[168] Wikipedia. *Heterogeneous Aerial Reconnaissance Team — Wikipedia, The Free Encyclopedia.* [Online; accessed 9-March-2014]. 2013 (cit. on p. 2).

[169] M. Wollmer et al. "Online Driver Distraction Detection Using Long Short-Term Memory". In: *Intelligent Transportation Systems, IEEE Transactions on* 12.2 (2011), pp. 574–582 (cit. on p. 34).

[170] Weng-Keen Wong et al. "Bayesian network anomaly pattern detection for disease outbreaks". In: *ICML*. 2003, pp. 808–815 (cit. on p. 30).

[171] Peter IJ Wouters and John MJ Bos. "Traffic accident reduction by monitoring driver behaviour with in-car data recorders". In: *Accident Analysis & Prevention* 32.5 (2000), pp. 643–650 (cit. on p. 34).

[172] Bing-Fei Wu et al. "Reasoning-based framework for driving safety monitoring using driving event recognition". In: *Intelligent Transportation Systems, IEEE Transactions on* 14.3 (2013), pp. 1231–1241 (cit. on p. 35).

[173] Xiaorui Xi and E.H. Abed. "Formation Control with Virtual Leaders and Reduced Communications". In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*. 2005, pp. 1854–1860 (cit. on p. 3).

[174] Xiaorui Xi and Eyad H Abed. "New formation control designs with virtual leaders". In: *Proc. 16th IFAC World Congress*. 2005 (cit. on p. 3).

[175] F. Xiao and L. Wang. "State consensus for multi-agent systems with switching topologies and time-varying delays". In: *International Journal of Control* 79.10 (2006), pp. 1277–1284 (cit. on pp. 20, 29).

[176] Lin Xiao, Stephen Boyd, and Sanjay Lall. "A scheme for robust distributed sensor fusion based on average consensus". In: *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE. 2005, pp. 63–70 (cit. on pp. 5, 29).

[177] Yiguang Xuan and B. Coifman. "Lane Change Maneuver Detection from Probe Vehicle DGPS Data". In: *2006 IEEE Intelligent Transportation Systems Conference*. 2006, pp. 624–629 (cit. on p. 107).

[178] G Alastair Young and Richard L Smith. *Essentials of statistical inference*. Vol. 16. Cambridge University Press, 2005 (cit. on p. 87).

[179] Wenwu Yu, Guanrong Chen, and Ming Cao. "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems". In: *Automatica* 46.6 (2010), pp. 1089 –1095 (cit. on pp. 8, 22, 29).

[180] Wenwu Yu, Guanrong Chen, and Ming Cao. "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems". In: *Automatica* 46.6 (2010), pp. 1089 –1095 (cit. on p. 29).

[181] Wenwu Yu, Guanrong Chen, and Jinhu L. "On pinning synchronization of complex dynamical networks". In: *Automatica* 45.2 (2009), pp. 429 –435 (cit. on p. 22).

[182] Wenwu Yu et al. "Distributed Consensus Filtering in Sensor Networks". In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 39.6 (2009), pp. 1568–1577 (cit. on pp. 22, 29).

[183] Wenwu Yu et al. "Distributed Higher Order Consensus Protocols in Multiagent Dynamical Systems". In: *Circuits and Systems I: Regular Papers, IEEE Transactions on* 58.8 (2011), pp. 1924–1932 (cit. on pp. 8, 22, 29).

[184] Wenwu Yu et al. "Second-Order Consensus for Multiagent Systems With Directed Topologies and Nonlinear Dynamics". In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 40.3 (2010), pp. 881–891 (cit. on pp. 22, 29).

[185] Xi Yu and Mahmood R Azimi-Sadjadi. "Neural network directed bayes decision rule for moving target classification". In: *Aerospace and Electronic Systems, IEEE Transactions on* 36.1 (2000), pp. 176–188 (cit. on p. 32).

[186] Shanying Zhu, Cailian Chen, and Xinping Guan. "Distributed optimal consensus filter for target tracking in heterogeneous sensor networks". In: *Control Conference (ASCC), 2011 8th Asian*. 2011, pp. 806–811 (cit. on p. 23).