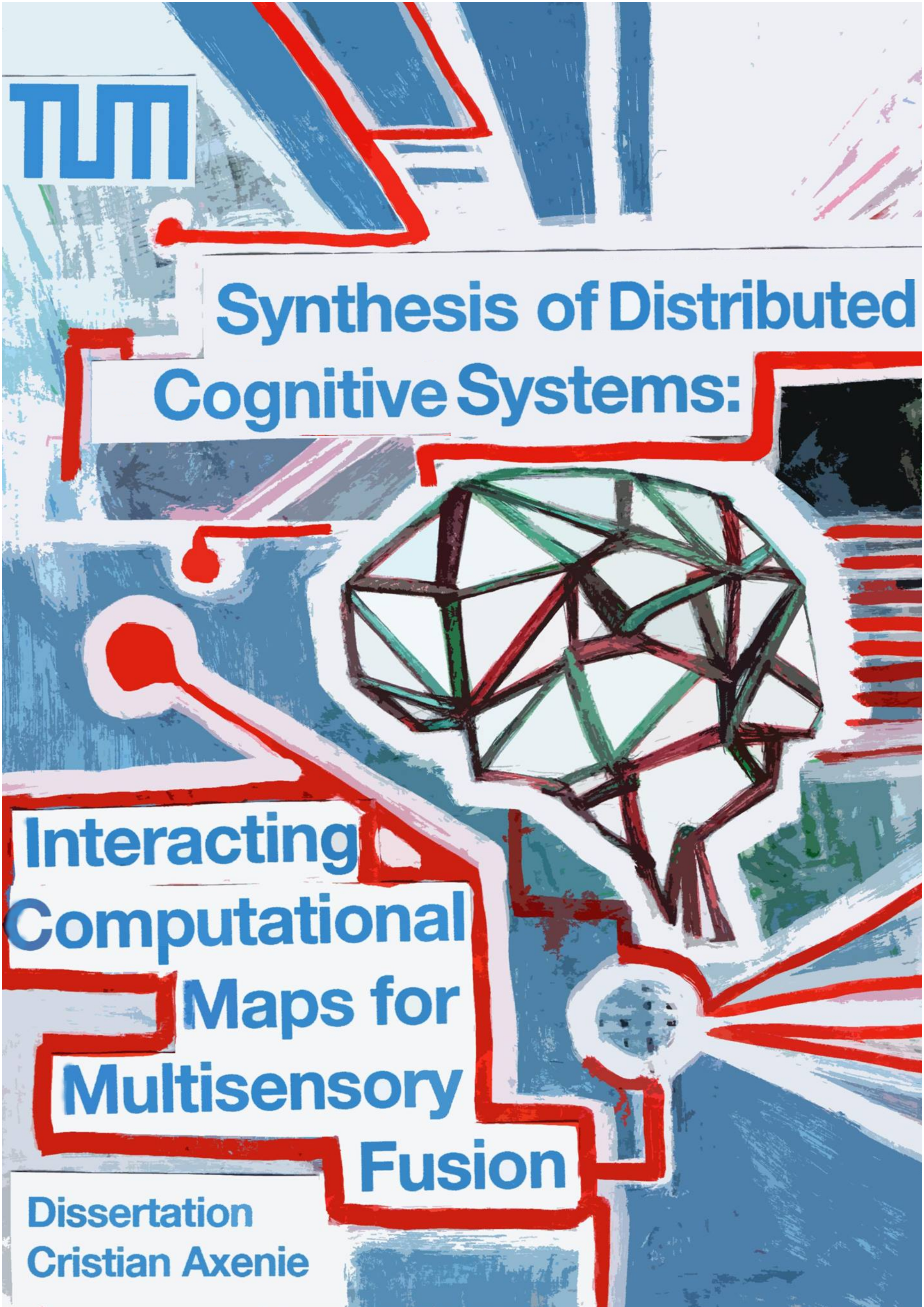


Synthesis of Distributed Cognitive Systems: Interacting Computational Maps for Multisensory Fusion

Dissertation

Cristian Axenie

Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München
Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

Synthesis of Distributed Cognitive Systems: Interacting Computational Maps for Multisensory Fusion

Cristian Axenie

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

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Samarjit Chakraborty

Prüfer der Dissertation:

1. Prof. Dr. Jörg Conradt
2. Prof. Jeffrey Krichmar, Ph.D.,
   University of California, Irvine / USA

Die Dissertation wurde am 24.11.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 31.03.2016 angenommen.

Foreword

Paraphrasing Helmholtz, ”perception is our best guess as to what is in the world, given our current sensory input and our prior experience”. Indeed, one of the most enduring quests in neuroscience and engineering alike concerns the perception of the external world. There are still fundamental questions to tackle about the possible mechanisms underlying the ability to perceive, yet some important principles were derived. These principles enable both biological and technical systems to leverage their capability to interact with their environment. This is where it all started.

I could never have imagined that after spending many years among electronic circuits, control theory, and robots, I will finally work on investigating processing subtleties of neural systems. Joining the Neuroscientific System Theory Group (NST) fed my curiosity and opened a new world of possibilities. Not too far from the great feeling of designing and building robots, I have been given the opportunity to go one step further, to understand information processing in neural systems, to develop novel algorithms inspired by brain functionality, and, finally, to transfer these to robots.

A roboticist dream? Well, this thesis quantifies my effort to respond this question and moreover summarizes the research I conducted within Neuroscientific System Theory Group at the Technische Universität München since late 2011. The amazing journey from the first neurobiology courses and mathematically dense neural computation compendiums, the first ideas sketched on the whiteboard, the first proof-of-concept implementations, to finally testing the hypotheses in real-world robotic scenarios, has been an exciting one. Looking back in time I realise that I could not have undertaken it without the great support of many people.

First of all, I would like to thank my advisor Prof. Dr. Jörg Conradt. Starting from our first discussion on distributed processing, he opened a new perspective for the fresh robotics graduate I was, a new exciting niche, neuromorphic robotics. Combining a sharp and scientifically rigorous mindset with a pragmatic engineering approach to problems, he supported me in every aspect of my research. Enabling me to freely explore this truly multidisciplinary area, he carefully supported me towards tackling challenging problems I encountered during my PhD years.

An exotic mixture of amazing people, NST was genuinely a creativity pool. Years spent among this great team will definitely put a mark on both my scientific and personality profiles. Randomizing the order, I would like to thank Dr. Viviane Ghaderi, for her great moral support, her pragmatic view on the outcome of research, and her permanent and contagious enthusiasm. Patient enough, yet really active in analysing data and formalism, Dr. Christoph Richter's mentorship helped me to leave those painful states in which research didn't really progressed. Always a mature presence, Dr. Marcello Mulas was the person to tame my enthusiasm and bring value to my scientific approach to problems and paper writing. Never ending discussions on cortical circuits, inference, or more earthly topics, Mohsen Firouzi was always the mathematically correct opinion to have. Ranging from self-construction algorithms, to programming subtleties, from sports, to obscure music genres, Nicolai Waniek was always the perfect man to talk to and to already sketch a solution for problems. Finally, I would like to thank Susanne Schneider, who took care of all administrative details and made it all transparent for me.

Over the years, I have been involved in a number of projects and collaborations. I especially thank Dr. Matthew Cook from INI, University of Zürich and ETH Zürich for fruitful discussions shaping my perspective on neural computation. Moreover, during my stay at the Neuroscientific System Theory Group, I became a member of the neuromorphic engineering community, and an active participant in both Telluride Neuromorphic Cognition Engineering Workshop and Capocaccia Cognitive Neuromorphic Engineering Workshop. I thank this great community for the opportunity to extend my basic formation with extensive experience in neuromorphic hardware and neural computation.

Finally, this work would not have been possible without the great support from the Bavarian Elite Network which generously funded my research and offered me the possibility to share my results within the community.

Besides research, a lot of people have made daily life a perfect balance. I would like to thank my ”local group” of friends who unconditionally listened to all my ”network convergence issues” and helped me to detach from such research problems. Thank you Xenia Sabodash, Meisi Zhan, Milla Widmer, Izabela Maria Kolodziej (credits for the cover), and all my ”remote” Romanian friends for their patience and support.

Last but not least, I would like to thank my parents for believing in me and my work and their tireless guidance even from 3000 km away. Thank you for supporting my adventure of being a student along with its perks and drawbacks.

Munich, November 2015 Cristian Axenie

Contents

Notations

Abbreviations

ABCA	Alpha-Beta Divergence Correlation Analysis
ANN	Artificial Neural Network
BCM	Bienenstock-Cooper-Munro Model
BSS	Blind Source Separation
CCA	Canonical Correlation Analysis
CJPDA	Cheap Joint Probabilistic Data Association
CSP	Constraint Satisfaction Problem
DBN	Dynamic Bayesian Network
DKF	Distributed Kalman Filter
EKF	Extended Kalman Filter
EM	Expectaction Maximisation
FG	Factor Graph
GPS	Global Positioning System
ICA	Independent Component Analysis
ILD	Interaural Level Difference
ITD	Interaural Time Diference
JCBB	Joint Compatibility Branch and Bound
JPDA	Joint Probabilistic Data Association
LIDAR	Light Detection and Radar
LLE	Locally Linear Embedding
LMS	Least Mean Squares
MAP	Maximum Aposteriori Estimate
MCMC	Markov Chain Monte Carlo
MHA	Metropolis-Hastings Algorithm
MHT	Multiple Hypothesis Test
MHT-D	Distributed Multiple Hypothesis Test
MLE	Maximum Likelihood Estimator
MSE	Mean Squared Error
NLCCA	Nonlinear Canonical Correlation Analysis
NN	Nearest Neighbours
PCA	Principle Component Analysis
PDA	Probabilistic Data Association
RANSAC	Random Sample Consensus
RBM	Restricted Boltzmann Machine
RGB-D	Red-Green-Blue - Depth (cameras)
RMSE	Root Mean Squared Error
RPY	Roll-Pitch-Yaw

SCNN	Sequential Compatibility Nearest Neighbour
SFA	Slow Feature Analysis
SIFT	Scale Invariant Feature Transform
SIR	Sequential Importance Resampling
SIS	Sequential Importance Sampling
SLAM	Simultaneous Localisation and Mapping
SMC	Sequential Monte Carlo
SNR	Signal to Noise Ratio
SOM	Self-Organising-Maps
UKF	Unscented Kalman Filter
WTA	Winner-Take-All

Conventions

x or X	Scalar
$\underline{x}$	Vector
$\underline{X}$	Matrix
$\underline{X}^T$	Transposed of $\underline{X}$
$\underline{X}^{-1}$	Inverse of $\underline{X}$
$f(\cdot)$	Scalar function
$\underline{f}(\cdot)$	Vector function
$\hat{x}$	Estimated or predicted value of x
$\overline{x}$	Average value of x
$\|\cdot\|$	norm
$\underset{x}{argmax}$	the argument of the maximum
$\propto$	proportional to

Abstract

Biological and technical systems live in a rich environment for which, due to the multimodal nature of incoming sensory streams and variety of motor capabilities, there is no single representation and no singular unambiguous interpretation. Furthermore, there is no single neural process or engineering algorithm to interpret sensorimotor streams for all possible scenarios in which a system might operate in.

In this work we proposed an alternative computational architecture, inspired by the distributed macro-architecture of the mammalian cortex. The underlying computation is performed by an interconnected network of computational maps, each representing a different sensory quantity. All the different sensory streams enter the system through multiple parallel channels and the system aligns, and given incoming observations, combines them into a coherent representation. In biological systems sensory representation and interpretation are flexible and context dependent operations underlining the use of dynamically adaptive sensory integration mechanisms. These mechanisms are learned and result as the outcome of a developmental process. Along these lines, the second component of this work focuses on the mechanisms underlying self-creation and learning of the functional relations between the computational maps encoding sensorimotor streams directly from the sensory data.

Depicting a synthetic view of our contribution, Figure 1 introduces a novel approach to representing, learning, and processing various sensory streams for multisensory fusion.

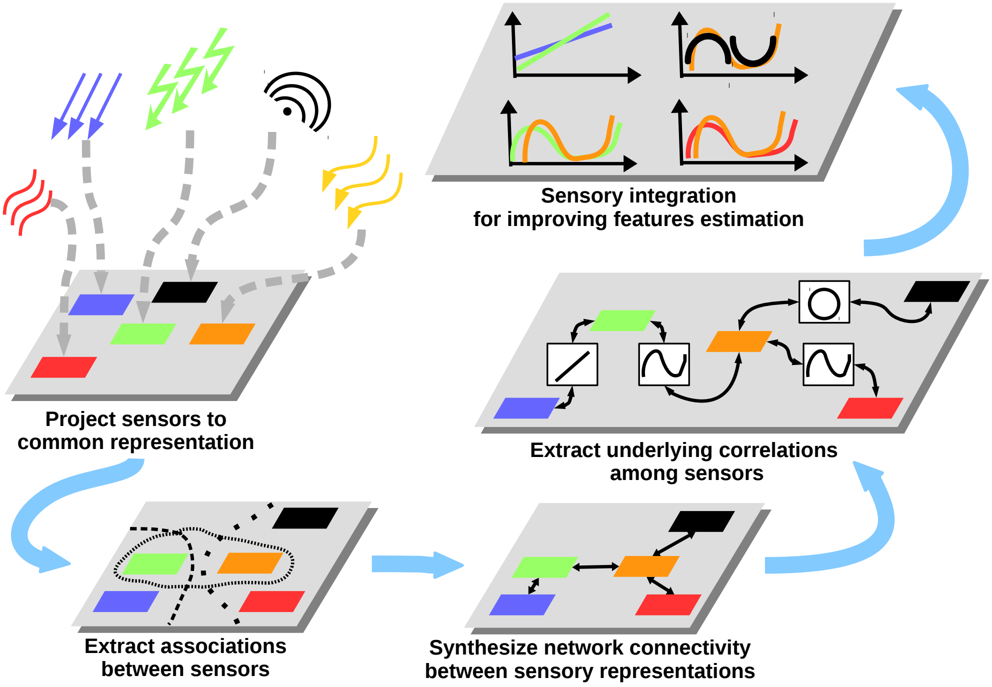

Sensory integration
for improving features estimation
Project sensors to
common representation
Extract underlying correlations
among sensors
Extract associations
between sensors
Synthesize network connectivity
between sensory representations

Fig. 1: Distributed cognitive systems for multisensory fusion: from sensory representations, to integration and cross-sensory learning for precise representations using autonomous synthesis processes.

Combining principles of distributed cortical computation for generic data representation and inference, our framework supports the change of paradigm towards neurally-inspired sensory processing. The results of our preliminary instantiations in various robotic scenarios (i.e. 2D mobile robot motion estimation, 3D attitude estimation on a quadrotor) make our approach a promising candidate for robust real-time multisensory fusion in robotic systems because of intrinsic scalability/parallelisation and automatic adaptation to unforeseen sensory perturbations.

Zusammenfassung

Biologische wie technische Systeme müssen sich in einer sensorisch ausgesprochen reichhaltigen Umgebung zurechtfinden. Die multimodale Natur der eingehenden Sensorströme und die mannigfaltigen Bewegungsmöglichkeiten machen eine eindeutige Repräsentation oder zweifelsfreie Interpretation unmöglich. Es gibt weder einen einzelnen bekannten neuronalen Prozess, noch einen technischen Algorithmus, der sensomotorische Signalströme in beliebig gewählten Szenarios, in denen ein System operieren kann, zu interpretieren vermag.

In dieser Arbeit stellen wir eine alternative Datenverarbeitungsmethode vor, die von der verteilten Makro-Architektur des Säugetierkortex inspiriert ist. Die zugrundeliegenden Rechenoperationen werden von einem Netzwerk miteinander verbundener "Rechenkarten"durchgeführt. Jede dieser Karten repräsentiert hierbei eine andere Sensormodalität. Die unterschiedlichen Sensorströme fließen parallel und zeitgleich in die Rechenkarten ein. Sie beeinflussen das System so, dass es sich an die gegebenen Beobachtungen anpasst, und sie zu einer kohärenten Repräsentation kombiniert.

In biologischen Systemen sind sensorische Repräsentation und Interpretation flexibel und abhängig vom jeweiligen Kontext. Biologische Sensorintegrationsmechanismen sind also adaptiv. Sie werden erlernt oder ergeben sich aus natürlichen Entwicklungsprozessen. Hierauf beruht die zweite große Komponente dieser Arbeit: Die selbstorganisierte Entstehung und das Erlernen von funktionalen Zusammenhängen zwischen unterschiedlichen Rechenkarten.

Figure 1 gibt eine schematische Darstellung unserer Methode wieder, vielschichtige Sensorströme zu repräsentieren, zu verarbeiten, aus ihnen zu lernen und sie hiermit letztlich sinnvoll zu vereinigen.

Die vorgestellte Kombination kortikaler Datenverarbeitungsprinzipien mit einem universellen Ansatz zur Datenrepräsentation ermöglicht einen Paradigmenwechsel in praktisch relevanten Anwendungsfeldern der Sensorverarbeitung. Wir zeigen dies anhand ausgewählter Einsatzbeispiele aus der Robotik: Zweidimensionale Bewegungsschätzung einer fahrbaren Plattform und dreidimensionale Fluglagebestimmung einer Schwebeplattform. Die Beispiele belegen weiterhin, dass unser Ansatz naturgemäß skalier- und parallelisierbar ist, was ihn zu einem vielversprechenden Kandidaten für echtzeitfähige und ausfallsichere Sensorfusion für Roboter macht.

1 Problem formulation

1.1 Preamble

The reciprocity between the environment and the perceiver, be it a biological or artificial entity, builds itself as a mutual interactive system. This perspective assumes that the operating environment provides opportunities, resources for decision-making, and actions, as well as information for what is to be perceived to guide actions [Gibson et al., 2003]. Moreover, actions have consequences that provide more useful and informative content to the perceiver, so as to properly describe a rich internal representation of the environment.

In this context, at any given moment, both biological and technical systems need to process multiple inputs from their different sensory modalities. Deciphering this broad array of sensory information is by far a non-trivial problem and both biology and engineering came up with successful approaches to make sense of the multisensory world. Different sensors are tuned to different forms of energy, each giving rise to a qualitatively and quantitatively different perceptual experience. *A proactive physical system needs to constantly combine a plethora of sensory information and moreover track and anticipate changes in one or more of the incoming streams in order to consistently create an internal representation.* This representation subsequently provides the base to build autonomous behaviour and flexible interaction with the environment [van Atteveldt et al., 2014].

In order to properly set up the context and define the problem, we need to extract the key principles governing multisensory fusion systems in both biological and technical systems. By far a pragmatic perspective, we will try to focus on the main principles of information representation and computation known to occur in neural systems and transfer these principles in technical systems. For validating this approach we use robotic systems as a flexible experimentation platform to develop and test hypotheses.

Either for environment interpretation or self-state estimation, both biological and technical systems need the capability to disambiguate perception by using different sources of sensory data. This subsequently guides their behaviour. We can already extract a fundamental principle governing multisensory fusion, the need for a coordinated interplay of available senses to properly interact with the environment [Kayser et al., 2015].

In order to make sense of the environment and own state, given all available sensory data, the system needs to solve several computational problems. For instance, given the available sensory observations, the system needs the capability to compensate for uncertainty and noise, assuming inherent redundancy in sensory observations. Moreover, systems have to infer new quantities from existing sensory observations, given known or learned causal relations. Finally, to optimise efficient processing of incoming sensory streams, systems should constantly generate predictions about future events. Multisensory cues play an important role in anticipation, as some sensory cues will often predict what will happen in other sensory cues. This principle is supported by the fact that different sensory modalities have different timing properties, which can enhance the predictive capacity across modalities. These features are usually a trademark of active sensing, where sensory events enter the system as a result of motor activity the system is generating. Active sensing defines a fundamental component of autonomous systems.

Another computational problem that a multisensory fusion system should handle, is related to the capability of inferring temporary degraded or missing information in sensory modalities. Exploiting data redundancy and intrinsic alignment mechanisms, the multisensory fusion scheme should handle inconsistencies and imperfections, assigning judicious confidence levels to contributing quantities. The goal is to exploit this feature in order to use the different detection / discrimination capabilities of each sensor, to extract a precise estimate in a timely manner.

Due to heterogeneous nature of the incoming sensory streams, a multisensory fusion system should align multiple scales and representations to improve precision in the estimated features. Moreover, the system must align different data types and accommodate high-level representations emerging from low-level sensory data processing. This capability of extrapolating from the data space to decision space is crucial in complex perception-action-cycles found in autonomous systems.

Focusing more on the computational substrate, distributing processing and representation provides a powerful paradigm for multisensory fusion. This paradigm ensures that, by maintaining only local knowledge of observed features, and by mutually exchanging information between distributed sensory representations, a consistent global representation can be constructed. In such distributed architectures, observations from each sensory source are processed locally before being fused. The fusion mechanism balances contributions from all available sources such that each sensory representation provides a local view of the observed feature. This distributed representation is then combined in a global view by the fusion mechanism.

Finally, in order to combine available sensory streams, the multisensory fusion system needs to exploit the structure in the sensory data and extract spatio-temporal associations from it. These associations yield an adaptive layer for sensory processing towards reducing uncertainty and judging environment's causal structure for subsequent decision making. Disentangling the impact that sensory data statistics have in the fusion process is still a challenge, but its heavily contributing to increase the flexibility and robustness of the system.

To sum up, the aforementioned principles provide basic design requirements for robust and flexible multisensory fusion systems. The proposed work builds upon these principles in order to bridge progress in neuroscience, involving modelling formalism and computational paradigms, with robotics, providing a unified conceptual framework to build adaptive technical systems. Today's engineered multisensory fusion implementations use a mature iterative design methodology involving mathematical models, simulation, analysis, and experimentation, yet lack the flexibility and adaptation capabilities proven by biological systems. Notwithstanding their excellent results they are highly constrained and dedicated to the operation scenario. On the other side, biomimetic designs provide an approach that seeks sustainable solutions for multisensory fusion by emulating nature's time-tested patterns and strategies [Passino, 2005]. The primary focus of this work is not to model, emulate, or analyse (neuro-)biological systems, but rather to use "bio-inspiration" for injecting new ideas, techniques, and perspectives into the engineering of complex autonomous systems. Embarking in such a challenge draws some important questions we will try to tackle throughout the thesis. What is the minimal description level useful in developing robust, flexible, and general brain-inspired multisensory fusion mechanisms for robotics? Furthermore, how complex and biologically plausible the models should be, and given the validation in a real-world scenario, do we feed back to neuroscience? Finally, given that today's multisensory fusion architectures are fast and precise, but specific and inflexible, and (neuro-)biological systems bring more complex, and not well understood, but flexible and robust models, where do we place ourselves when aiming at providing tractable solutions for real-world technical systems? An interesting view trying to provide a plausible approach and solution on these issues was formulated as a set of "universal laws and architectures" [Doyle et al., 2011]. This rather holistic perspective was described using various case studies to illustrate concepts like robustness, complexity, and architecture under an integrated theory. One central theme of this theory is that there is a balance between robustness and efficiency, marked as trade-offs and constraints, in both biological and technical systems, Figure 1.1. In-line with the aforementioned perspective, we guided our design towards a robust approach, while keeping the complexity at a reasonable level. *Using relatively simple computation, given by the physics of the sensors (e.g. formulated as mathematical functions), our model "does its best" in combining individual sensory contributions, described by different reliabilities, noise patterns and uncertainty. The model aims at providing its best interpretation of a perceived feature, given available sensory streams and their relations by reaching consensus between all sensory contributions as fast as possible.* Relaxing the condition to converge to an optimal solution, our model provides a flexible approach (not being bound to a certain scenario) to extract a plausible interpretation of the incoming multisensory streams of information.

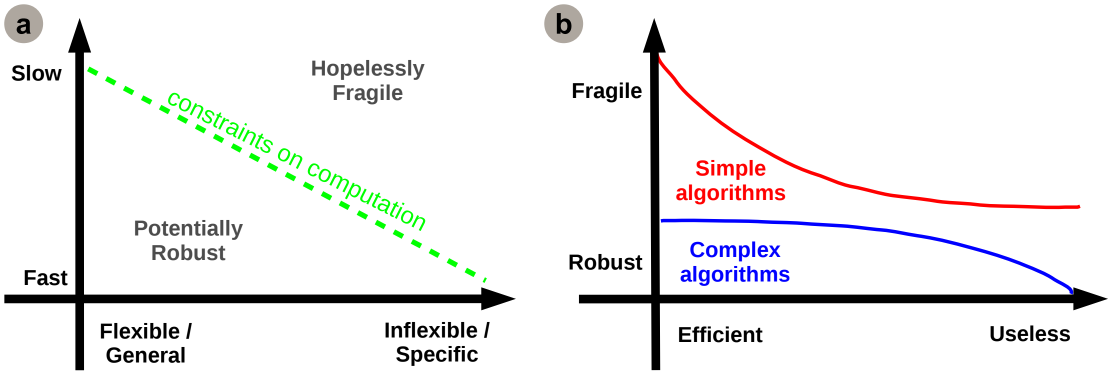


Fig. 1.1: Universal "conservation laws" (constraints) and universal architectures (constraints that deconstrain) [adapted from [Doyle et al., 2011]]: a) Perspective on system design and trade-offs given computational constraints; b) General unifying perspective.

1.2 Structure of the thesis

This section provides a brief overview on the thesis structure. Providing insight on state-of-the-art methods for multisensory fusion and the mathematical apparatus formalising them, Chapter 2 also introduces specific computational models known to explain multisensory processing in the mammalian brain. This parallel description was introduced in order to extract the main principles followed in the thesis for the design and implementation of a robust and flexible multisensory fusion mechanism. The second chapter ends by revamping the perspective and motivation behind the proposed work.

In line with the goal of the thesis, to strengthen the bridge between neuroscience and engineering, Chapter 3 introduces the formalism and functionality of the proposed multisensory fusion model. Focusing on concepts, and using dynamical systems analysis, the chapter introduces the key ingredients of the model: distributed representation and computation, concurrent dynamics to reach consensus, and mutual interaction, between computing units encoding different quantities, towards a generic consistent representation.

Chapter 4 introduces a sample instantiation of the proposed framework in a motion estimation scenario. Investigating 2D egomotion estimation for omnidirectional mobile robots, an in-depth analysis of the capabilities of the model and comparison with state-of-the-art implementations is provided. Following a neurally inspired distributed processing scheme, the parallelisation capabilities of the model are further investigated. From sequential or parallel implementations on standard PCs, to massively parallel neuromorphic computing hardware, we investigate the real-time operation capabilities in real-world scenarios. Taking advantage of existing low-power parallel hardware, we evaluate the model for estimation accuracy and performance on such an embedded platform.

Chapter 5 introduces an extension of the proposed model for multisensory integration, namely by introducing a learning process, similar to the one taking place during the development of a biological nervous system. This extension enables our system to extract mappings between sensory cues, instead of manually creating the network architecture given prior knowledge about sensory configuration. The employed learning model is able, given various sensory inputs, to converge to a state providing a coherent representation of the sensory space and the cross-sensory relations defining the fusion process dynamics.

Focusing on self-construction and learning, Chapter 6 introduces sample model instantiations in order to test its applicability and performance in real-world scenarios. Alleviating the need for tedious design and parametrisation, the model is capable to learn sensory data statistics and distribution for efficient representation and computation. We analyse model's capabilities in a 3D egomotion estimation scenario on a quadrotor.

Chapter 7 summarizes the work and emphasizes the main advantages brought by the proposed framework for multisensory fusion. The discussion session analyses the most important design principles guiding the proposed work along with an objective analysis of its advantages and limitations in the instantiated real-world scenarios. Revamping the proposed perspective over multisensory fusion by emphasizing the need for an adaptive substrate to extract the underlying sensory statistics for improved fusion, the discussion ends with proposing a series of extensions which might contribute to obtain a more mature and versatile framework.

2 Introduction and context

Biological and artificial systems (e.g. robotic systems) alike need a way to leverage their sensing capabilities in order to autonomously adapt their behaviour. Through evolution, biological systems refine their adaptation capabilities and are able to robustly represent, and interact with, their environment. On the other side, today's technical systems lack the capability to adapt to uncertainty and dynamic changes in their noisy, ambiguous, and sometimes partially observable environment. *In order to disambiguate their perception, physical systems use a complex pattern of interactions to act upon the environment which reciprocally influences their state.* These interactions are flexible and context-dependent due to the large number of possible actions to take and the variety of complementary environment features to sense. Moreover, these interactions underline the need for an adaptive processing substrate to handle all context-dependent variations in the incoming perceptual streams.

As a result of this continuous interaction cycle with the environment, sensory input is acquired through, or modulated by, motor routines, such that perception itself becomes a sensorimotor process. Despite immediate effect on the system's state, sensorimotor cues contribute to the incremental development of experience by adapting and learning correlations between available cues. *Using its past exposure to the environment and new observations, a physical system can build more precise representations which subsequently contribute to understanding, and adapting to, new contexts.*

This view suggests that the interaction capabilities with the environment, and the interpretation of its perceptual representations builds upon a developmental trajectory, in which a physical system learns and continuously adapts it's internal environment representations and sensorimotor processing mechanisms.

2.1 Perception and multisensory processing

Environment unfolds itself as a rich multisensory percept continuously contributing to the system's state changes. The coexistence of different sensory modalities enhances a system's likelihood to survive and the direction in which it can develop. The multiple sources of simultaneous inputs free the system from environmental constraints extending its perception of the environment, such that its internal representations are rich and decision making is more robust.

Different senses are tuned to different forms of energy, and give rise to a qualitatively and quantitatively different perceptual experience which the system must disambiguate. In this process the system must constantly combine all available information and moreover track and anticipate changes in one or more of these streams. The problem of maintaining a coherent internal representation of the environment and own state, given complementary percepts of the environment is not trivial and expects considerable adaptive capabilities from the physical system.

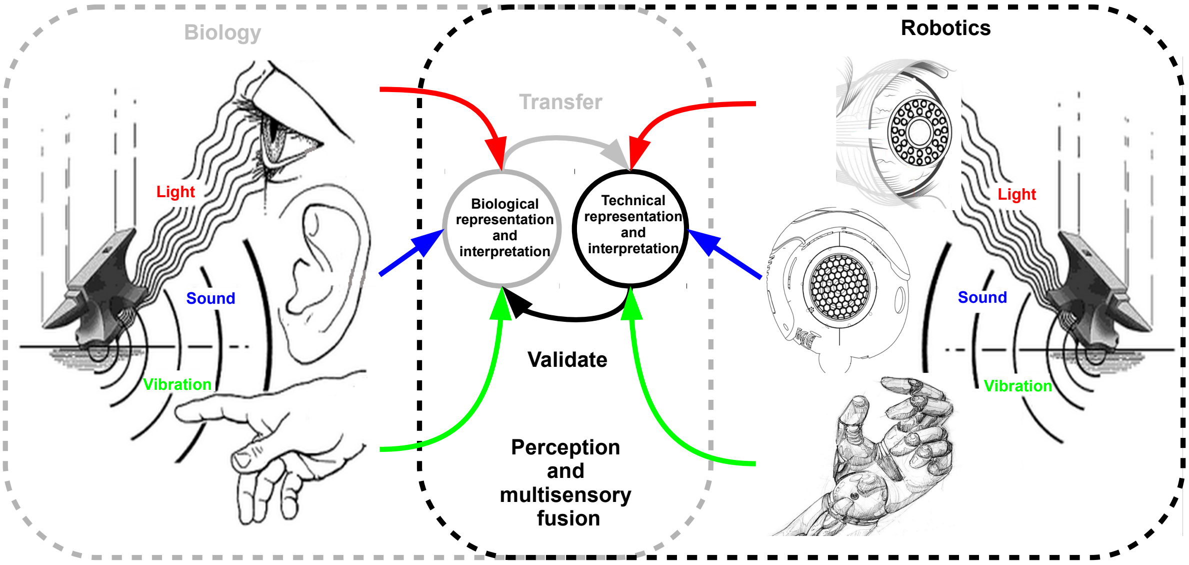

Biology
Robotics
Transfer
Light
Sound
Vibration
Biological representation and interpretation
Technical representation and interpretation
Validate
Perception and multisensory fusion
Light
Sound
Vibration

Fig. 2.1: Transfer principles from biology to robotics. Validate models from biology through results in robotics. (adapted from [Meredith et al., 2012])

Generally, the process responsible of combining information from a number of different sources of information to provide a robust and complete description of the environment and/or own state, is termed multisensory fusion. Instantiating this process allows the system to extend the range and variety of features it can detect and experience. This abstract and general perspective on environment representation and interpretation subsumes the fundamental principles guiding the design of multisensory fusion systems for autonomous systems. The current work aims at providing a bridge between neuroscience and technical systems, in order to enhance the adaptation and robustness of today's technical systems, by means of transferring principles of mammalian neural substrate processing to technical implementations.

In order to set up the context and to properly place the proposed work, the rest of this chapter is dedicated to a review of state-of-the-art methods for multisensory fusion. Not aiming at providing an exhaustive overview, the section will basically focus on extracting the main principles which guide current approaches for designing multisensory fusion systems. This overview is complemented by a review on the computational principles known to enable models of multisensory fusion in the mammalian brain.

Starting with a simple functional classification, the overview will then switch to multisensory fusion architectures in both engineered and neural systems. The chapter will provide a broad overview on the formal mathematical models and techniques employed in engineering and also models known to describe fusion processes in neural systems. The dual perspective motivates the framework proposed in the current thesis and how bridging the two areas is beneficial to enhance the capabilities of today's technical systems. A synthetic description of the goals of the current work is given in Figure 2.1.

2.2 Multisensory fusion: classifications, functionality and architectures

Viewed as "a multi-level process dealing with the association, correlation and combination of data and information from single and multiple sources to achieve refined position, identify estimates and complete timely assessments of situations, threats and their significance" [White, 1991] or either as a technique to "combine data from multiple sensors to achieve improved accuracy and more specific inferences than could be achieved by the use of a single sensor alone" [Hall et al., 1997], multisensory fusion is an important component for all physical systems, enabling more complex environment perception, interpretation, and interaction. In the following section instead of following an exhaustive overview, [Khalengi et al., 2013, Castanedo, 2013], the focus falls on those relevant aspects for the proposed work. We analyse and discuss multisensory fusion techniques, architecture classifications, and functional aspects for state estimation and data association. The overview will emphasize design principles which govern today's implementations.

Providing improved confidence and reliability, as well as a reduction in data ambiguity, while extending spatial and temporal coverage, multisensory fusion mechanisms can be divided according to the relations between the input data sources; according to the abstraction level of the employed sensory data; or according to the input and output data types and their nature.

From the perspective of the relations between the sources of sensory data, multisensory fusion can be complementary, redundant or cooperative [Hall et al., 1997]. The three different fusion mechanisms are briefly presented in Figure 2.2. In the complementary scheme the data provided by sensory inputs represent different parts of the perceived scene and thus can be used to extract more complete information [Bagher et al., 2011, Asnath et al., 2014]. Conversely, the redundant scheme assumes that the sensory inputs provide data about the same perceivable feature or quantity and thus can be used for improving confidence and accuracy [Scherba et al., 2005]. Finally, in the cooperative scheme, the sensory sources are combined into new information which is typically more complex than the original data [Kubelka et al., 2014, Huerta et al., 2014]. These schemes reflect fundamental principles behind multisensory fusion, namely the capability to disambiguate perception by using different sources of sensory data to augment environment representation or own state; the capability to compensate for failures and / or to infer missing quantities due to redundant observations; and the capability to infer new quantities from existing observations.

Taking one step back from the functional relations between the input data, one can analyse multisensory fusion mechanisms from the abstraction level of the employed data, [Luo et al., 2002]. There are three main levels on which multisensory fusion systems operate: sensory observations, characteristics or decisions. This architectural classification is synthetically depicted in Figure 2.3, where we exemplify sensor fusion levels separation for quadrotor pitch angle estimation using inertial data. The first abstraction level is the signal level, in which the system combines the individual sources to provide more accurate data (a lower signal-to-noise ratio). This level uses raw data from the sensors, without preprocessing, eventually only de-noised. Successful implementations of low-level multisensory fusion were developed in assistive robotic systems. For example, gait control perfor-

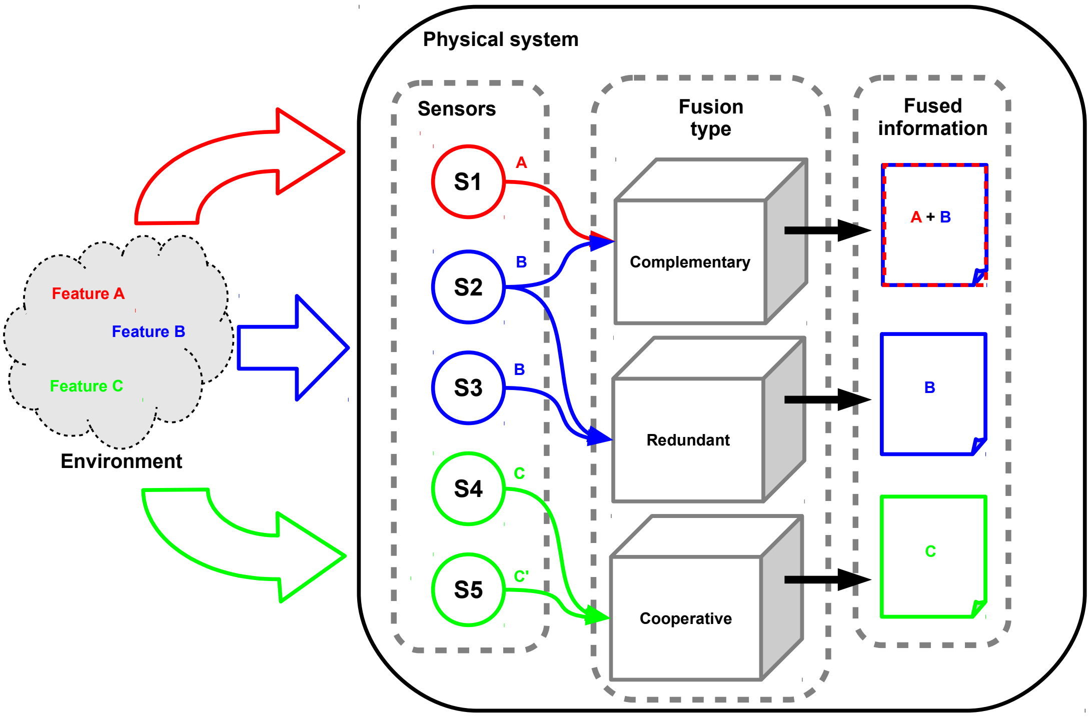


Fig. 2.2: Multisensory fusion architectures: different relations between input data sources.

mance of a robotic walker was improved through a feedback loop based on reaction forces and gait kinematics estimated from low-level signal regularities [Cifuentes et al., 2014]. In another scenario, targeting wearable robotics, primary modalities like electromyography, electroencephalography, and mechanical sensors, were fused for optimal assistance and

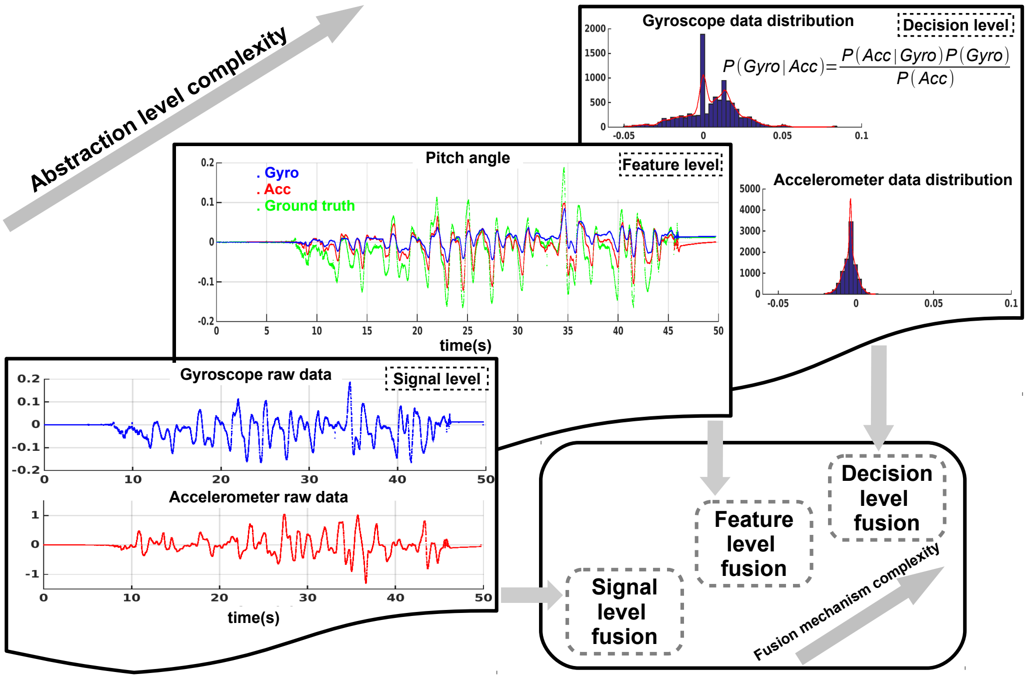


Fig. 2.3: Multisensory fusion architectures: different abstraction levels of the employed data.

quick reactions to perturbations or changes in user intentions [Novak et al., 2014].

Going higher on the abstraction scale, the pixel level fusion, operates at an aggregated level between the signal level and the feature level, usually embedded in the feature level. Fusing data using this aggregate representation, [Liu et al., 2014] obtained improved performance in adaptive image fusion based on wavelet transform in trinocular vision of picking robots. Furthermore, in a real-time visual environment mapping for quadrotors for both indoor and outdoor operation [Zhou et al., 2014] obtained a precise representation of the environment and loop closure.

Generally speaking, feature level multisensory fusion is applied when characteristics or features are readily available from the sensory signals. This abstraction level employs the use of more concrete quantities (e.g. shape, velocity, depth, optic flow) useful for direct recognition, as in the case of mobile robot vision guided localization and navigation [Siagian et al., 2014] or data fusion for biometric protection [Chin et al., 2014].

The highest level where multisensory fusion mechanisms are employed, typically in large scale systems, is the decision level, or the symbol level. Symbol-level fusion is commonly employed in applications where multiple sensors are of different nature or observe different features of the environment. Pattern recognition is the best example. In this case feature information is extracted from sensor data, defining a point in the feature space. This point is mapped to a symbolic representation of the environment based on that symbol's neighbourhood in the feature space. Such a neighbourhood function may be defined using probability theory [Soumalya et al., 2014], Dempster-Shafer's theory of evidence [Denoux et al., 2014], fuzzy logic [Santos et al., 2015], or neural networks [Dani et al., 2014].

In a second functional classification, we extend, with a finer granularity, the fusion process [Dasarathy, 1997], such that the outcome of the multisensory fusion system takes into account the type and nature of the input data. This functional classification is synthetically described in Figure 2.4 emphasizing that the fusion mechanism is able to align multiple scales and representations to improve precision, or to infer new quantities in a visual scene interpretation scenario [Cook et al., 2011]. Following a data in-data out (DAI-DAO) paradigm, a multisensory fusion algorithm for ambient noise estimation in wireless sensor networks was developed [Polastre et al., 2004]. Using a moving average filter the fusion mechanism was able to provide more accurate and reliable output data estimates given raw input data. Employing a slightly different paradigm in which multisensory fusion uses raw data from different sources to extract features or attributes that describe an entity (i.e. data in-feature out, DAI-FEO) improved real-time digital image stabilization was obtained [Erturk, 2002].

In this same context, multisensory fusion can be applied on a set of features to improve or refine another feature, or extract new ones (i.e. feature in-feature out, FEI-FEO). In such a scenario [Singh et al., 2006] provided an adaptive learning mechanism for mobile sensing networks for environmental monitoring. Creating feature maps from aggregated sensory data, the algorithm was able to geographically describe the distribution of a sensed parameter.

Going further from the low-level of raw sensory data, we reach the decision level in which we combine features and decisions in a feature in-decision out scheme (FEI-DEO). In this scheme the multisensory fusion system takes a set of features described using a

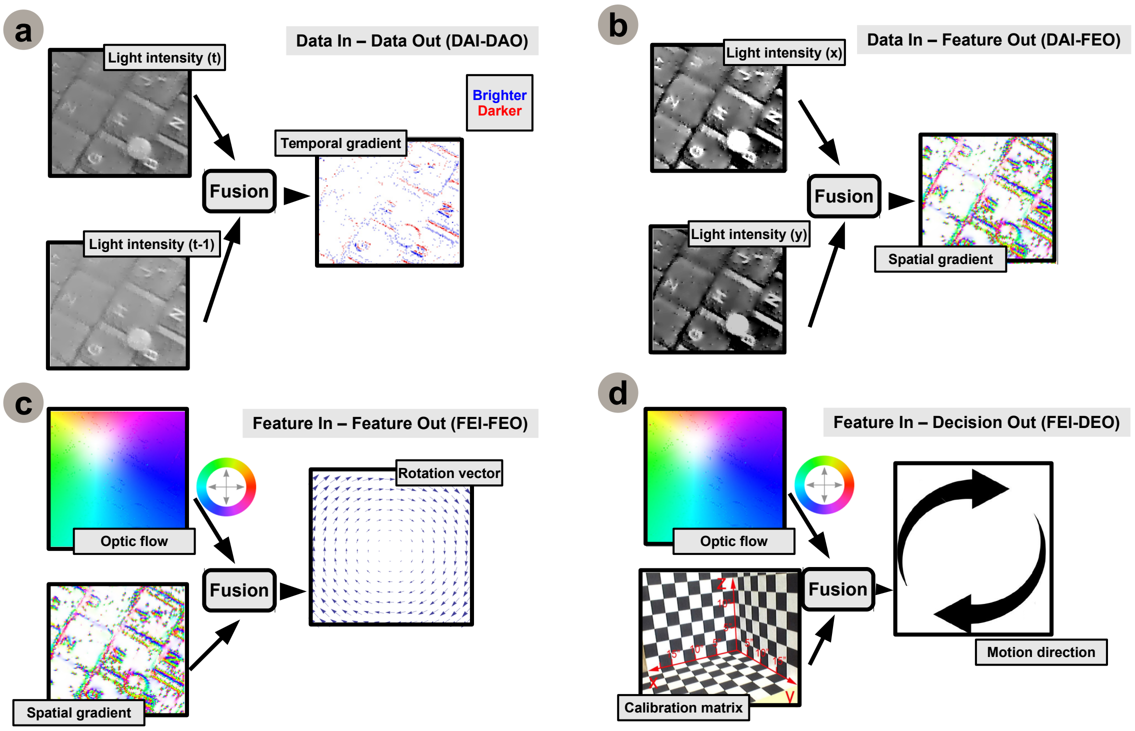


Fig. 2.4: Multisensory fusion architectures: different type and nature of the input data. a) Input streams based on temporal raw data from sensors combined to infer a third temporal sensory quantity; b) Input streams based on spatial raw data from sensors combined to extract a spatial feature; c) Combining features from different modalities and representation to infer a new feature; d) Combining features from different modalities to discriminate / describe a decision. (adapted from [Cook et al., 2011])

symbolic representation or a decision and infers new quantities. Such approach proved its advantages in fusing features describing the data transmission traffic decay to infer node failures in sensor networks [Nakamura et al., 2005, Luo et al., 2006].

This classification based on heterogeneous inputs and outputs strengthens the idea that the fusion system must also align different data types and accommodate high-level representations emerging from low-level sensory data processing. This capability of extrapolating from the data space to decision space is crucial in complex perception-action-cycles found in autonomous systems.

Focusing on data related aspects in sensor fusion, the last taxonomy emphasizes different approaches in handling inherent problems of the data to be fused. Combining different levels of abstraction, the fusion mechanism should handle data irregularities such as inconsistencies (i.e. conflicts and outliers) and imperfections (i.e. uncertainty, imprecision and granularity) [Kumar et al., 2006]. Exploiting data redundancy and intrinsic alignment mechanisms, the multisensory fusion scheme should be able to handle inconsistencies and imperfections assigning judicious confidence levels to contributing quantities [Smets, 2007].

The aforementioned functional classifications analyse the internal data handling and

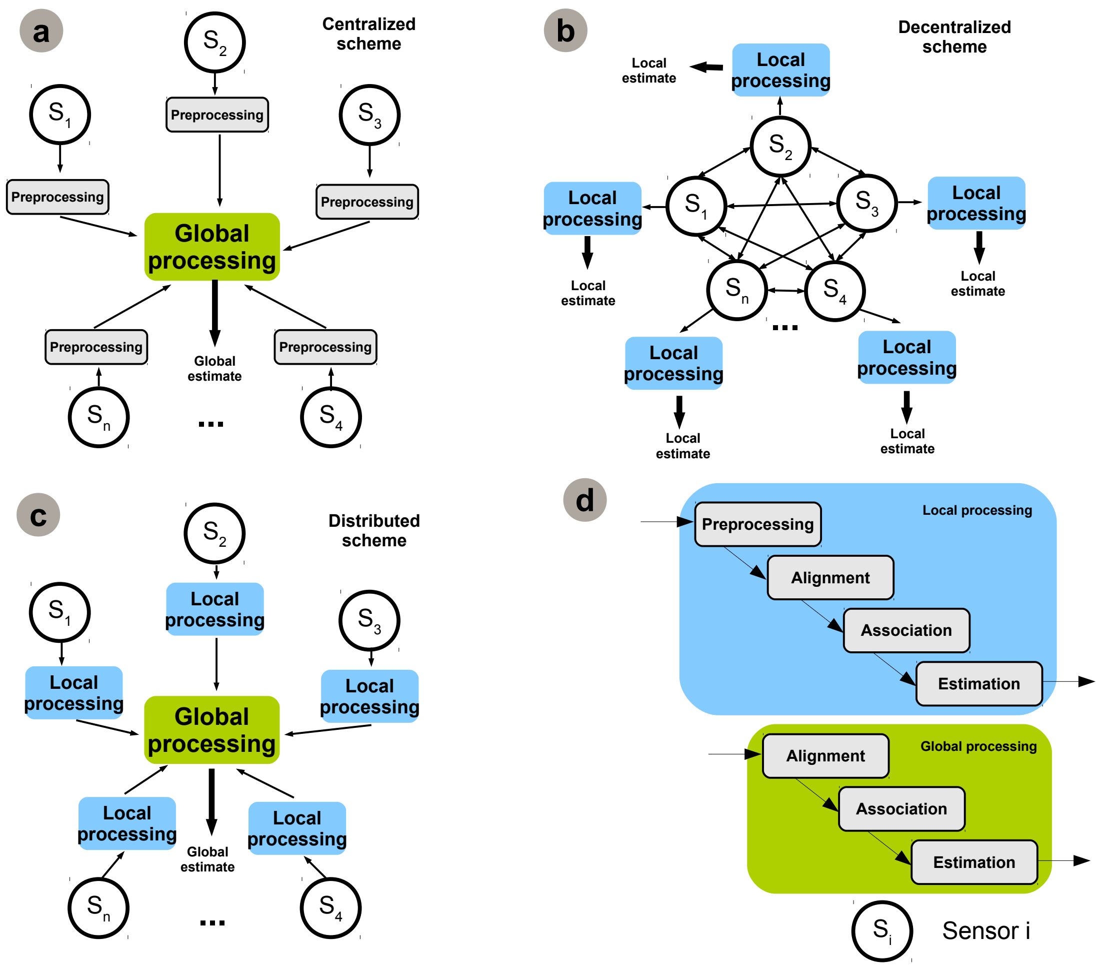


Fig. 2.5: Multisensory fusion architectures: different processing schemes. a) Centralized scheme: different modalities undergo local preprocessing before fused in a central node; b) Decentralized scheme: Fully interconnected network of processing nodes for each of the modalities each one providing its own estimate; c) Distributed scheme: local modality processing of a feature combined in a global representation and processing node. d) Local or global processing stages.

combination mechanisms of multisensory fusion systems. To understand how these principles are deployed in real-world systems, and how processing is performed, we introduce a new classification, based on the type of architecture of the multisensory fusion system. A principled depiction of most important multisensory fusion architectures is given in Figure 2.5. The basic architecture of a multisensory fusion system is the centralized architecture, Figure 2.5a. In this paradigm the fusion process is handled by a central processing unit interfacing with all sensory data sources. This simple processing scheme, in an ideal case of correct data alignment, data association and negligible data communication time, proves to be an optimal approach.

However, these assumptions do not hold in real-world scenarios, due to differences in time delays from the different sensors to the central processor. To counteract the disadvantages of the centralized scheme, a decentralized architecture was proposed.

Alleviating the need to concentrate communication and processing on a single processing unit, this scheme uses a network of nodes with local processing and communication capabilities, Figure 2.5b. Making the process autonomous, each unit is able to locally fuse its readouts and the information from its peers. Albeit its attractive distribution of processing load, communication costs increase proportionally with the number of nodes.

In order to overcome the drawbacks of the decentralized architecture, various adaptive schemes were applied to shape the communication load. Using adaptive data transfer mechanisms in a multi-robot scenario [Rajesh et al., 2014] proposed an algorithm to generate precise maps of the environment subsequently used for planning. Moreover, using parallelisable sparse approximations of Gaussian processes for spatio-temporal prediction, a decentralized data fusion scheme was efficiently used for active sensing with mobile sensors [Chen et al., 2012]. Suffering from scalability issues, this architecture was modified towards a distributed architecture, Figure 2.5c.

The main difference is that in the distributed architecture, observations from each source node are processed locally before being sent to a central fusion node. This central node balances contributions from all the nodes in the architecture. Basic data association and filtering are performed locally, at the source node, each node providing a local view of the observed feature. This distributed representation is then combined in a global view in the central fusion node. Being able to parallelise computation and distribute the representation, this scheme provided a suitable candidate for a large number of robotics applications for environment representation and intelligent ambient interaction [Pennisi et al., 2014]. Due to the the fact that the distributed architecture is based on local interactions, it was successfully used for extracting sensorimotor models for robotic proprioceptive and exteroceptive sensory calibration in cluttered environment navigation [Kelly et al., 2014]. In order to combine the advantages of both decentralized and distributed architectures hierarchical schemes were developed. These hybrid architectures ensure that the fusion process takes place at different levels in the hierarchy. This paradigm provides the means to reduce the necessary communication and computational costs because aggregation and computation are performed in a distributed fashion.

The last section provided an insight in the architectural details of multisensory fusion systems, focusing on the practical processing schemes. *Distributing processing and representation, while maintaining only local knowledge of observed features and mutual exchange of information to extract the global representation, provide important design principles.*

After providing a commonly regarded view of different strategies and their underlying functional details for multisensory fusion, the focus shifts towards the classes of problems for which multisensory fusion is applied. The two classes of multisensory fusion mechanisms that we address in the thesis are state estimation, and data association and correlation extraction. This is the core part of the current chapter as it will formally introduce the techniques currently employed in multisensory fusion systems. Along with state-of-the-art models we will also introduce the neural models known to explain multisensory integration mechanisms in the brain. This comparative approach serves as a means to frame the proposed work and emphasize its motivation and advantages.

2.3 Multisensory fusion: state-of-the-art techniques

In the upcoming section we will focus on methods and the mathematical apparatus typically used in multisensory fusion systems for state estimation. The models will be complemented by their biological counterparts to emphasize and extract the functional principles, and at the same time, frame the model proposed in the thesis.

2.3.1 State estimation

State estimation mechanisms aim at determining the changes in the state of a target or a certain perceived quantity, given sensory observations or measurements. In its general form, state estimation accounts for a tracking technique, in which it is not guaranteed that the target observations are relevant, measurements being affected by noise and uncertainty.

State estimation is an important multisensory fusion mechanism that aims at providing a global target state given the available sensory observations. From a functional point of view, this process typically accounts in finding the set of parameters that provide the best fit to the acquired redundant observations. As sensory observations are generally corrupted by errors, uncertainty and the propagation of noise in the measurement process, state estimation mechanisms are able to alleviate these problems by integrating prior knowledge and incoming observations.

Most of the state estimation methods employ the probability theory to describe and extract a state estimate from sensory measurements. The most commonly used estimation methods, including maximum likelihood estimation, maximum a posteriori estimation and Bayesian programming, Kalman filter (standard formulations and distributed version) and the particle filter (standard formulation and distributed version) are introduced in this section. Supported by mature implementations in engineering, the probabilistic approach, gained an important role and became a tool also in neuroscience, such that models of perceptual processing have been found to obey Bayesian inference rules. Furthermore, there is a growing body of experimental evidence consistent with the idea that animals are somehow able to represent, manipulate, and ultimately make decisions based on probability distributions.

Maximum Likelihood Estimation

Far from the constraints and assumptions of the laboratory, real-world environment features enable different sensory cues to capture its structure and dynamics. In real systems, sensory cues are often imperfectly related to the physical environment feature they measure due to measurement errors and the variability in the mapping between the cue value and the feature. This implies an unknown probability distribution describing the state variable or the quantity of interest. Probabilistic estimation provides an appropriate solution to this problem. Let's assume that θ is the state or quantity being estimated and $\underline{z}$= $(z_1, z_2, ..., z_n)$ a sequence of n previous sensory observations of θ. The likelihood function λ is defined as a probability density function of the sequence of observations $\underline{z}$ given the true value of θ,

$$\lambda(\theta) = p(\underline{z}|\theta). \tag{2.1}$$

The Maximum Likelihood Estimator (MLE) finds the value of θ that maximizes the likelihood function, such that:

$$\hat{\theta}(k) = \underset{\theta}{argmax}\ p(\underline{z}|\theta) \tag{2.2}$$

The model expresses the probability of the observed sensory data and requires the existence of analytical or empirical models of the sensors to compute the likelihood function. This relatively simple linear mathematical framework has been shown to explain and describe basic cue integration processes in the brain [Landy et al., 2012].

However, in most perceptual problems encountered in the natural world, sensory cues are often imperfectly related to physical environmental properties due to the variability in the mapping between the sensory cue value and the measured property, hence special assumptions must be considered. A typical instantiation of this framework is the linear cue integration model of maximum reliability. Basically, this model accounts for an averaged sum of all sensory contributions. Given z_i samples of n independent Gaussian sensory variables Z_i, with same mean, η, and variance, σ_i^2, the minimum variance, unbiased estimator of η is a weighted average:

$$\hat{\underline{z}} = \sum_{i=1}^{n} w_i z_i, \tag{2.3}$$

where the weights, w_i, are proportional to the cue reliabilities, r_i,

$$r_i = \frac{1}{\sigma_i^2}, w_i = \frac{r_i}{\sum\limits_{j=1}^{n} r_j} \tag{2.4}$$

The global estimate reliability is computed as

$$r = \sum_{i=1}^{n} r_i \tag{2.5}$$

and, given that the cues are conditionally independent, unbiased values, will be smaller than individual cues reliabilities and never worse than the least reliable. Despite its simplicity, the model verified interesting predictions in psychophysical experiments for optimal cue integration of vision and haptic information in estimating size, shape and position of objects, [Ernst et al., 2002], as depicted in Figure 2.6. For estimating the size of a real world object, S_W, visual (S_V) and touch information (S_H) were considered. Considering typical sensory models assuming unbiased sensory signals, with normally distributed independent noise components, this scenario provided that integration is beneficial to disambiguate the scene. This is due to a weighting scheme based on each cue's reliability, such that the variance of the combined estimate from vision and touch is smaller than individual estimates fed to the fusion process (i.e. in this case weighted averaging). From the basic models in neuroscience, the MLE has been extensively used as a mechanism for multisensory integration in technical systems. Targeting mobile robot localization in robot teams, fusing only local information about robot position, such that each robot is able to estimate the relative pose of nearby robots together with changes in its own pose, [Howard et al., 2002] obtained precise results without any external cues. Furthermore, using a dynamic grid representation, which improved the maximum likelihood estimation mechanism [Feng et al., 2014] provided a rapidly converging solution for dynamic self-localization in a robotic navigation scenario.

In order to characterize sensory cue integration in real world scenarios the linear model by itself is not sufficient. It can systematically underestimate the variance of the likelihood such that it biases the estimate for insufficient sensory observations. A powerful framework which can characterize more complex problems is Bayesian estimation and decision theory.

Bayesian Maximum a Posteriori Estimation

Bayesian theory offers a more generic and flexible framework for cue integration. Within this framework, information reliability provided by sensory observations, $\underline{z}$, of a certain feature of interest, θ, is represented by a ”posterior” probability distribution,

$$P(\theta|\underline{z}) = \frac{P(\underline{z}|\theta)P(\theta)}{P(\underline{z})}, \tag{2.6}$$

where $P(\theta|\underline{z})$ is a posterior probability distribution which describes how true are the values of θ given the sensory data $\underline{z}$. A narrow (i.e. low standard deviation) probability distribution indicates reliable data, while a broader probability distribution represents unreliable data. In this formulation prior knowledge about sensory data distribution, $P(\theta)$, provides information on how likely are some values of θ to be found in the environment.

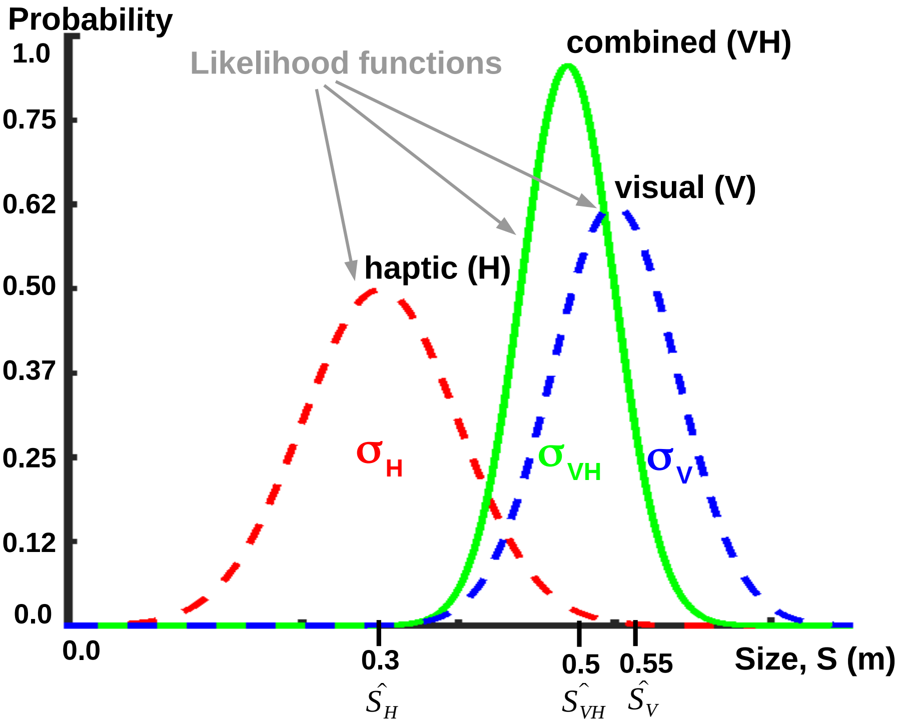


Fig. 2.6: Likelihood functions in maximum likelihood estimation of visual-haptic integration for size estimation. The combined visual-haptic estimate $\hat{S_{VH}}$ is a weighted average of the individual visual and haptic estimates $\hat{S_V}$, $\hat{S_H}$. The variance associated with the combined estimate is less then either of the two individual estimates.

Furthermore, to quantify how likely different values of θ can give rise to observed data $\underline{z}$ the model uses a likelihood function, $P(\underline{z}|\theta)$. To normalize the posterior probability distribution the product between prior sensory data distribution and likelihood is divided by $P(z) = \int_{-\infty}^{\infty} P(\theta|\underline{z})d\theta$.

The most straightforward Bayesian theory approach to cue integration assumes that the sensory cues are conditionally independent. If this condition is ensured, the likelihood function of all sensory cues is given as a product of all individual cues likelihoods,

$$P(z_1, z_2, ..., z_n|\theta) = \prod_{i=1}^{n} P(z_i|\theta). \tag{2.7}$$

Employing this formulation and ignoring the constant normalization denominator, the posterior probability distribution $P(\theta|z_1, z_2, ..., z_n)$ can be computed using

$$P(\theta|z_1, z_2, ..., z_n) \propto P(z_1, z_2, ..., z_n|\theta)P(\theta). \tag{2.8}$$

In order to optimally extract the estimate contained in the inferred posterior probability distribution, the maximum a posteriori estimate (MAP) is used. This estimate extracts the value of θ that maximizes the posterior probability distribution $P(\theta|z_1, z_2, ..., z_n)$,

$$\hat{\theta}(k) = \underset{\theta}{argmax}\ P(\theta|z_1, z_2, ..., z_n). \tag{2.9}$$

The Bayesian approach offers a more general formulation than the linear model, such that it replaces averaging with prior-likelihoods multiplications and point representations of perceptual estimates with probability distributions.

Using the representation provided by Bayes' formalism and relating sensory variables to each other over adjacent time steps inside a Dynamic Bayesian Network (DBN), a generic sensor fusion system was developed [Besada-Portas et al., 2002]. The proposed model was able to extract a homogeneous and formalized way of capturing the dependencies that exist between a robot location and the state of the environment. To achieve this the algorithm fused sensory data from a magnetometer, wheel encoders, a beacon, and ultrasonic sensors on-board the platform. Extending the basic Bayesian formalism [Ferreira et al., 2012] proposed a neuromimetic Bayesian programming framework for multimodal active perception. The model was able to deal with uncertainty and ambiguity in a multisensory fusion scenario for fast egomotion processing in a behavioural relevant fashion, using visual, auditory and inertial data. Using rather global cues in each sensory modality (i.e. visual-disparity, auditory-binaural time difference, and vestibular-angular velocities) the proposed framework in [Ferreira et al., 2012] was extended to use distributed representations of the data and features in the form of perceptual maps, given as input for Bayesian programs [Ferreira et al., 2013].

Albeit good results in both describing psychophysical results predictions and technical implementations, the two basic models (e.g. MLE, MAP) for multisensory fusion follow strong modelling assumptions. If sensory noise sources are independent, described by Gaussian probability distributions, and individual sensory cues are unbiased and redundant, then the integration mechanism is optimal, because it provides the lowest possible variance of its combined estimates [Ernst et al., 2012]. The aforementioned assumptions are not always fulfilled in real-world scenarios and limit models' capabilities. This limitation motivates the need for more robust and adaptive mechanisms in precise multisensory fusion for state estimation.

The Kalman Filter

As an exceptional case of the Bayes filter, the Kalman filter is the most popular multisensory fusion technique for state estimation. Enforcing simplifying constraints on the system dynamics, linear measurement and system models, and zero-mean Gaussian noise affected observations, the Kalman filter is widely used due to its relatively simple structure, ease of implementation, and optimality (i.e. minimal MSE).

Using a discrete state space model of a dynamical system, the Kalman filter, estimates the state, $\underline{x}$,

$$\underline{x}(k+1) = \underline{A}(k)\underline{x}(k) + \underline{B}(k)\underline{u}(k) + \underline{w}(k), \tag{2.10}$$

given the sensory observations, $\underline{z}$,

$$\underline{z}(k+1) = \underline{H}(k)\underline{x}(k) + \underline{v}(k), \tag{2.11}$$

where $\underline{A}(k)$ is the system's state transition matrix, $\underline{B}(k)$ is the input transition matrix, $\underline{u}(k)$ is a control signal, $\underline{H}(k)$ is the sensory observations matrix, and $\underline{w}(k)$ and $\underline{v}(k)$ are system and measurement noise respectively. Both system and measurement noise signals are considered zero-mean Gaussian noise signals described by covariance matrices $\underline{Q}(k)$ and $\underline{R}(k)$ respectively. Given the system parameters, the Kalman filter relaxes to a solution after an iterative prediction-correction process, as depicted in Figure 2.7. Despite its capability to progressively adapt, the Kalman filter is sensitive to data corrupted by outliers,

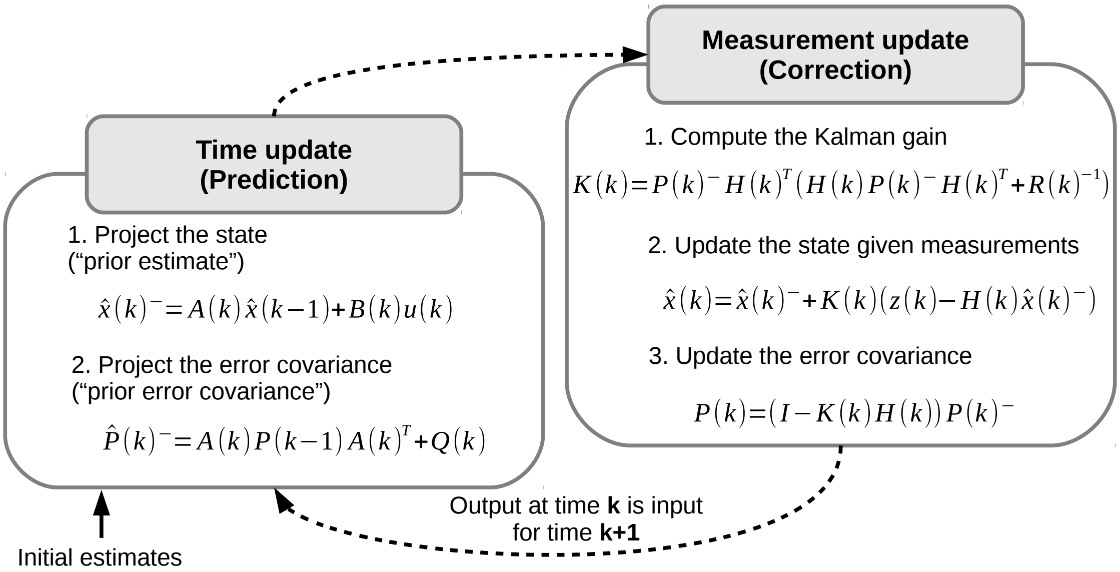

Measurement update (Correction)
1. Compute the Kalman gain
$K(k)=P(k)^{-}H(k)^{T}(H(k)P(k)^{-}H(k)^{T}+R(k)^{-1})$
2. Update the state given measurements
$\hat{x}(k)=\hat{x}(k)^{-}+K(k)(z(k)-H(k)\hat{x}(k)^{-})$
3. Update the error covariance
$P(k)=(I-K(k)H(k))P(k)^{-}$
Time update (Prediction)
1. Project the state ("prior estimate")
$\hat{x}(k)^{-}=A(k)\hat{x}(k-1)+B(k)u(k)$
2. Project the error covariance ("prior error covariance")
$\hat{P}(k)^{-}=A(k)P(k-1)A(k)^{T}+Q(k)$
Output at time **k** is input for time **k+1**
Initial estimates

Fig. 2.7: Generic Kalman filter processing scheme.

and becomes inappropriate for scenarios in which the error signal is not parametrized. To address these drawbacks, as well as nonlinear dynamical models and nonlinear observation models, the Kalman filter has been extended using first-order (i.e Extended Kalman Filter, EKF) and second-order (i.e. Unscented Kalman Filter, UKF) approximations as Taylor expansion of the state estimates.

The EKF is mainly used in multisensory fusion for robotic applications. In an indoor quadrotor control scenario, the EKF model in [Engel et al., 2012] fused camera, inertial and altitude data, to implement a monocular SLAM algorithm for stabilization. The algorithm was robust to temporary loss of visual tracking and significant delays in the communication process and was able to eliminate drift using SLAM. In a slightly different scenario [Erdem et al., 2015], visual information from a camera was fused with inertial cues to estimate 3D egomotion. The proposed model showed clear advantages in fusing both gyroscope and accelerometer during correction stage in EKF, to gain position tracking accuracy (due to acceleration data) and orientation tracking accuracy (due to gyroscope data). Using a different, rather high-level representation of the environment, [Barczyk et al., 2015] proposed a multisensory fusion scheme based on EKF, combining on-board motion sensors with readings from point clouds from a depth sensor for indoor robot localization. The EKF was improved by using invariant (symmetry-preserving) observers, such that the resulting Invariant EKF used its non-linear structure to take advantage of the geometry of the problem and provide a more robust solution.

Although used in a widespread range of scenarios, the EKF has some disadvantages due to the computation of the Jacobian matrices, slowing down the entire processing scheme. Attempts to alleviate this problem have been attempted, mainly aiming at linearising the model, but this introduced large errors in the filter, driving to instability. Having the capability to avoid the linearisation steps and the errors in the EKF, the UKF gained much attention in the robotics community.

Employing deterministic sampling of nonlinear functions to capture and recover the mean and covariance of sensory observations, the UKF extracts the minimum set of points of interest around the mean value of the state. Easily parallelisable, the UKF has been successfully used in visual guidance for robotic surgery, by providing real-time pose estimates of surgical instruments [Vaccarella et al., 2013]. Combining optical tracking and electromagnetic tracking data, the model was able to provide robust estimates of position during robot navigation given marker occlusions, and magnetic field distortions. In a robotic egomotion estimation scenario [Bloesch et al., 2014], a UKF was developed to fuse optic flow and inertial measurements, minimizing the dimensionality of the state space, allowing a fast implementation.

Going away from technical implementations there are several behavioural experiments suggesting that the mammalian nervous system uses an internal model of the dynamics of the body to implement a close approximation to a Kalman filter [Deneve et al., 2007]. The proposed neural implementation of the Kalman filter involved recurrent basis function networks with attractor dynamics, a kind of architecture that can be readily mapped onto cortical circuits. Taking advantage of a distributed representation and relatively simple computations, the proposed model embedded additional information about the input sensory streams and made use of their statistics when fusing the data.

Using experimental evidence suggesting that the brain is capable of approximating

Bayesian inference in the face of noisy input stimuli, [Wilson et al., 2009] proposed a neural network whose dynamics mapped directly to a Kalman filter. For small prediction errors the model was able to precisely behave as a Kalman filter but switched to an optimal Bayesian model as soon as the prediction error was large. The model supported the way in which sensory data probability distributions are encoded and used in the brain.

In line with the goal of extracting the computational principles and design principles of state-of-the-art approaches we also consider the distributed version of the Kalman filter (Distributed Kalman Filter, DKF). Extending the basic model by using different unsynchronized sensory data sources, the DKF needs additional time synchronization mechanisms to ensure consistent prediction-correction. Synchronization for subsequent sensory observations ensures that the model attains global consensus. Using these principles the DKF computes local state estimates using global sensor models, which are usually not optimal given local sensory observations. In order to avoid this problem and make it practical to implement [Chong et al., 2014] developed a method for de-biasing the covariance in DKF making tractable for real world applications [Olfati-Saber et al., 2011]. Here, in a distributed estimation and motion control scenario, mobile sensor networks employed DKF for collaborative target tracking. The algorithm optimized a Fisher mutual information metric to ensure that sensing agents seek to improve the information value of their sensed data, while maintaining a safe-distance from other neighbouring agents. Dealing with distributed information processing in sensor networks [Reinhardt et al., 2012] used recursive local estimates for consensus and reached optimality when assumptions about the global measurement uncertainty were met.

Particle Filters

Efficiently coping with non-Gaussian noise and nonlinear sensor dynamics, Monte Carlo simulation based techniques are employed in multisensory fusion as an alternative to Kalman filters. Either used as Sequential Monte Carlo (SMC) or Markov Chain Monte Carlo (MCMC), this technique is amongst the most powerful and popular methods for approximating probabilities associated with sensory data. Particle filters are a powerful method, in fact a recursive version of the SMC, able to represent probability distributions in a distributed manner. This method builds the posterior probability distribution using a weighted ensemble of randomly drawn samples (particles) as an approximation of the probability density of interest. The extracted probability distribution is obtained as a weighted sum of random samples resulting from the combination of sampling (i.e. Sequential Importance Sampling, SIS) and resampling (i.e. Sequential Importance Resampling, SIR) during particle propagation in time. In the standard algorithm the first phase (i.e. prediction) is responsible with modifying each particle with respect to the sensor model and simulate the noise effect on the estimate. The second phase (i.e. update) the weight of each particle is updated using the last sensory observation, and particles with low weights are removed. A schematic depiction of the particle filter functionality is given in Figure 2.8. Due to its attractive capabilities the particle filter was used in various robotics multisensory fusion applications. Being able to extract a multi-modal probability distribution over the target state space, a particle filter was used for precise event-based robot simultaneous localization and mapping (SLAM) [Weikersdorfer et al., 2012]. Extending the standard algorithm to use single measurements from a neuromorphic embedded dynamic vision sensor (eDVS) the

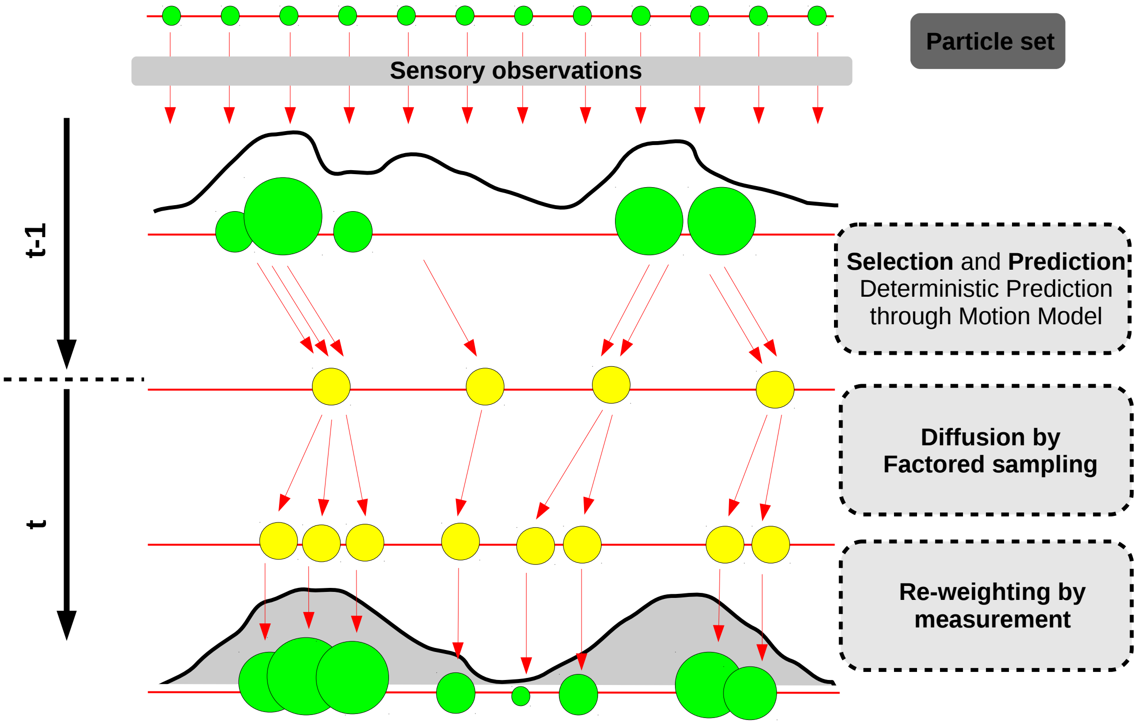


Fig. 2.8: Generic Particle Filter processing scheme. Given sensory observations, start with initial state value following data distribution. Draw samples to represent (minimal prediction error) the current state given the motion model. Define the weights for the particles using resampling after diffusion. Re-weight the new particles.

need for complete measurements and re-sampling steps in fixed intervals was alleviated, enhancing real-time processing capabilities. The developed algorithm was able to surpass the Kalman filter by handling occlusions and measurement ambiguities. In a robot assembly scenario [Thomas et al., 2007] a particle filter based multisensory fusion mechanism was developed for integrating force, torque and vision in order to ensure precise chaining plan execution in the assembly. Furthermore, in an attempt to organise initially disconnected sets of sub-maps in a complex environment, the particle filter model in [Fallon et al., 2012] achieved rapid multi-floor indoor map building using a human body-worn sensor system fusing information from RGB-D cameras, LIDAR, inertial, and barometric sensors.

Despite its good results in various applications, the particle filter has some disadvantages. One such disadvantage resides in the fact that the particle filter needs a large number of particles to obtain a small variance in the probability density estimate, and a large number of particles increases the computational costs significantly. Furthermore, in multisensory fusion problems involving a high-dimensional state space the number of particles increases exponentially with the dimensionality which makes it intractable.

In order to take advantage of the distributed representation of the estimated probability densities, the basic particle filter model has been extended to a distributed processing scheme [Bashi et al., 2003, Hlinka et al., 2013]. The emphasis was on distributed implementations on multiprocessor systems using three schemes for distributing the computations of generic particle filters, including re-sampling and, optionally, a Metropolis-Hastings algorithm (MHA) step (responsible to obtain a sequence of random samples from multi-dimensional distributions). Results obtained in target tracking scenarios supported the distribution of computation to provide a solution to the use of a large number of particles. In another real-time robotic scenario, for laser tag game playing, a distributed particle filter was implemented to perform decentralized sensor fusion [Rosencrantz et al., 2003]. The employed particle filter model considered that each particle can be viewed as an entire history or trajectory, and the set of all particles represents an approximation of the posterior probability distribution over trajectories. This consideration made the model well-suited for the type of posteriors required by the constrained decentralized selective communication scheme in the proposed scenario. Using a similar extension of the particle filter [Montemerlo et al., 2002] provided a fast SLAM algorithm that recursively estimated the full posterior distribution over robot pose and landmark locations, scaling with the number of landmarks in the map. This approach made use of a factored implementation of the particle filter for managing the number of landmarks and incorporating each sensory observation in the re-sampling step.

2.3.2 Data association

In many real-world scenarios the sensory data available to the system, be it biological or artificial, must be coherently extracted from the noisy, and sometimes partially observable environment. Data association is crucial for providing a precise environment or self-state interpretation. Formally, data association is defined as the process of assigning and computing the weights that relate sensory observations (or their temporal evolution) from one sensor to sensory observations of another sensor (or its temporal evolution) [Hall et al., 1997].

In a typical scenario, data association takes place before state estimation, due to the impact data associations, their coherence and accuracy have on the performance of the state estimation. Although an exhaustive search of data associations for a given scenario grows exponentially with the number of considered sensory modalities, various methods to extract data association have been developed. These techniques span from classical clustering algorithms, to probabilistic methods, statistical learning, and neural networks. At the same time, various neurally inspired processing models for learning data association were proposed, following experimental data and with different degrees of biological plausibility.

In this section we will introduce the main design stages of such systems, emphasizing the main principles governing data association extraction in both technical and biological models. In some multisensory fusion applications sensory data is complementary. The goal is to exploit this feature in order to use the different detection / discrimination capabilities of each sensor, to extract a precise estimate in a timely manner. Sensors are coupled through their observations stemming from the same object, feature, or motion at a certain moment in time. The different detection / discrimination capabilities lead to ambiguities when trying to match observations from multiple sources. In order to counteract this problem the system needs to take into account and exploit the diversity in the data and extract spatio-temporal associations from the sensory data. As previously mentioned, various methods were developed for extracting data associations (ranging from probabilistic inference, to clustering, machine learning, and graphical models). Closest to the proposed approach in the thesis, neurally inspired methods were also proposed and will provide a view on how the brain might solve the data association problem for efficient multisensory fusion.

Probabilistic Data Association

Data association techniques using probability theory encode the problem of extracting the correlation of multiple sensory streams in probability distributions. The basic probabilistic data association (PDA) algorithm [Bar-Shalom et al., 1975] associated probability distributions to hypotheses based on valid sensory observations. The algorithm, also termed probabilistic data association filter, is suboptimal, and uses the association probabilities for the latest sensory observations. The key idea of PDA is that a weighted average of all validated observations, where probabilities are used as weights, provides input for the fusion algorithm [Kirubarajan et al., 2004]. The basic assumption is that the state is normally distributed according to the latest state estimate and covariance matrix [Abolmaesumi et al., 2004].

In a typical scenario, valid sensory observations, $\underline{Z}(k)$, at time k, were extracted from those samples falling in a validation window (gate), γ, given their covariance gain, $\underline{S}(k)$, using

$$\gamma \geq (\underline{Z}(k) - \hat{\underline{z}}(k|k-1))^T \underline{S}^{-1}(k)(\underline{z}(k) - \hat{\underline{z}}(k|k-1)) \tag{2.12}$$

In the basic formulation, PDA comprises a prediction and an update step, similar to the Kalman filter. For the prediction step, given the sensor model $F(k-1)$ at moment $k-1$, the state is computed as

$$\hat{\underline{x}}(k|k-1) = \underline{F}(k-1)\hat{\underline{x}}(k-1|k-1). \tag{2.13}$$

Linearising the measurement matrix, $\underline{H}(k)$, the measurement prediction is given by

$$\hat{\underline{z}}(k|k-1) = \underline{H}(k)\hat{\underline{x}}(k|k-1), \tag{2.14}$$

and contributes to the computation of the innovation of the $i-th$ sensory observation,

$$v_i(k) = z_i(k) - \hat{z}(k|k-1). \tag{2.15}$$

Following the same update scheme as in the Kalman filter, the total update of the covariance is given by

$$v(k) = \sum_{i=1}^{m_k} \beta_i(k) v_i(k), \tag{2.16}$$

$$P(k) = K(k)(\sum_{i=1}^{m_k} \beta_i(k) v_i(k) v_i^T(k) - v(k)v^T(k))\underline{K}^T(k), \tag{2.17}$$

where m_k is the number of valid observations at time k, $\beta_i(k)$ is a weighting factor and $\underline{K}(k)$ is a gain factor. Finally, the association probability of $i-th$ measurement is computed using,

$$p_i(k) = \begin{cases} \frac{(2\pi)^{\frac{M}{2}}\lambda\sqrt{|S_i(k)|}(1-P_dP_g)}{P_d}, \ \textit{if i=0} \\ e^{-0.5v^T(k)S^{-1}(k)v(k)}, \ \textit{if } i \neq \textit{0} \\ 0, \ \textit{otherwise} \end{cases}$$

where

$$\beta_i(k) = \frac{p_i(k)}{\sum\limits_{i=0}^{m_k} p_i(k)}, \tag{2.18}$$

and M is the size of the input sensory observations vector, λ is the clutter density in the environment, P_d is the detection probability of the correct observation, and P_g is the validation probability of a detected value.

In PDA the association detection process is based on computing the association probabilities which are subsequently used as weights for each sensory source. Although this method provides good results in scenarios in which the estimated feature doesn't make abrupt changes, it will encounter problems in the case in which the observed features suddenly change.

One such scenario is mobile robot 3D visual SLAM [Gil et al., 2010]. While the robot moved in the environment, images from the two cameras were acquired and combined such that the algorithm needed to decide whether new observations come from an already seen landmark in the map or it is a new landmark that should be initialized. The PDA scheme for this problem starts with making observations from the two sensors. Using prior observations, the algorithm predicts the two time evolutions of the sensory observations (i.e. tracks) and uses them to predict incoming observations. This step allows the definition of an area in sensory space where to expect an observation. This expectation window (i.e. validation gate) narrows the search space, making the algorithm tractable. Subsequent sensory observations are then checked against the validation gate and validated if they are consistent matching / pairing candidates. A synthetic depiction of the algorithm is given in Figure 2.9.

Given that each landmark (e.g. L1, L2) is described by a visual descriptor (i.e. motion trajectory descriptor), for each new observation $o(k)$ composed of a distance measurement, z_{target} and a visual descriptor d_{target}, the algorithm must decide whether the observation corresponds to one of the known landmarks or is a new landmark. The decision is based on a distance metric which must be minimized taking into account the current map layout (i.e. current landmark configuration). The PDA framework received a lot of attention due to its uncertainty representation capabilities and many variants were developed for various multisensory fusion scenarios. In [Gil et al., 2006] an improved PDA algorithm was proposed for mobile robot visual SLAM in a typical office environment. Using scale invariant feature transform (SIFT) output as features and applying a filtering technique to concentrate on a reduced set of distinguishable, stable features from different views of the stereo vision system, precise position estimates were extracted. Whenever a feature was selected, the algorithm computed a representative feature given the previous sensory observations, using a squared Euclidian descriptors distance, improving data association and reducing the number of landmarks that needed to be maintained in the map.

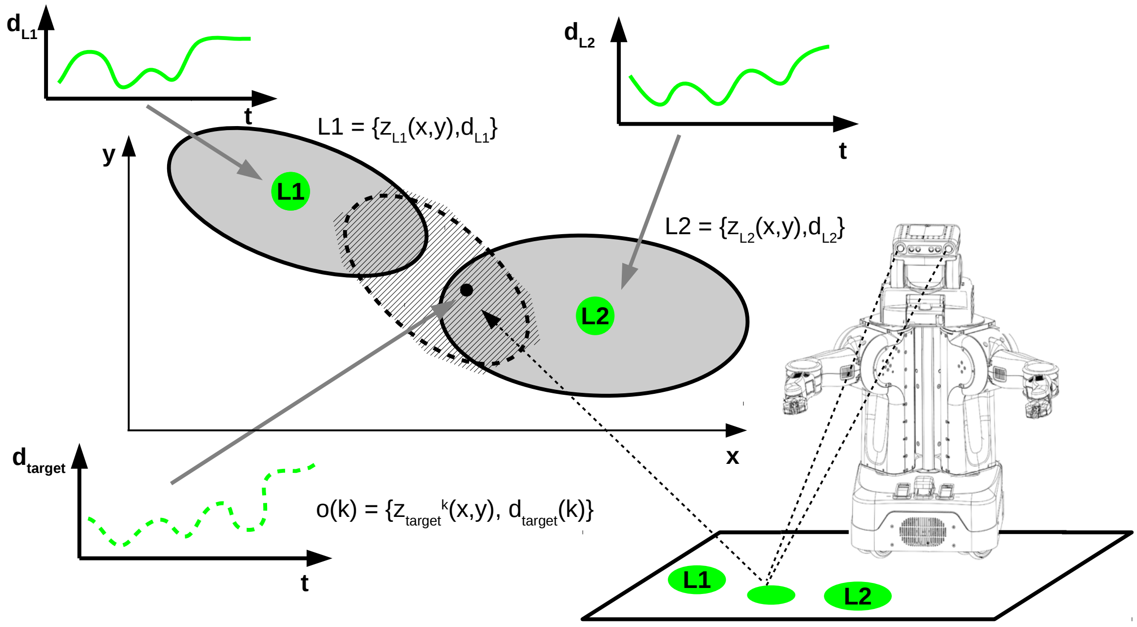

d_{L1}
t
$L1 = \{z_{L1}(x,y),d_{L1}\}$
y
L1
d_{L2}
t
$L2 = \{z_{L2}(x,y),d_{L2}\}$
L2
x
d_{target}
$o(k) = \{z_{target}^{k}(x,y),\ d_{target}(k)\}$
t
L1
L2

Fig. 2.9: Data association process for visual landmarks during SLAM. Each landmark is described by a descriptor (i.e. its temporal evolution) and given incoming observations the algorithm decides if the observations correspond to existing / new landmark based on a topological metric.

Joint Probabilistic Data Association

In order to provide a global and more consistent representation of the perceived scene using the available sensory streams, the standard PDA was extended to the Joint Probabilistic Data Association (JPDA). In JPDA the association probabilities are computed using all the observations coming from all the sensors. This extension allows the algorithm to consider various hypotheses and combine them.

Computing the probability that an observation comes from a certain sensor is based on the fact that this hypothesis excludes the others, in a mutually exclusive manner. The method uses the available observations (i.e. the most recent set) for a known number of sensors to evaluate the hypotheses and extract the associations. The method uses all available measurements in a vicinity of the sensor expected value to update the estimated value by using a weighted sum of measurement innovations.

Providing attractive capabilities in terms of handling high densities of false observations [Tchango et al., 2014] developed a computationally efficient multimodal tracking scenario using JPDA. The basic method was improved in terms of extracting an approximate interaction graph between the available sensory modalities on the fly, such that a function modelling the sensors' evolution and their mutual interactions was available. In a more complex scenario [Yangming et al., 2014] used an extended JPDA approach for fast and robust data association for mobile robot SLAM. Using a posterior-based joint compatibility test scheme, which alleviates known problems in typical methods (i.e. high computational cost, sensitivity to linearisation errors, prior knowledge of the full covariance matrix of state variables) the approach was able to outperform some classical algorithms, such as sequential compatibility nearest neighbour (SCNN), random sample consensus (RANSAC), and joint compatibility branch and bound (JCBB), in terms of precision, efficiency, and robustness. Without prior information regarding the initial relative pose in a team of collaborating robots [Indelman et al., 2014] used Expectation Maximization (EM) to efficiently infer individual robot pose and solve the multi-robot data association problem defined in the PDA framework. For any pair of robots in the team, the data association problem was defined as a constraint identification strategy for inlying and outlying position estimates extracted from the posterior probability distribution of robot trajectories. In a slightly different application scenario, [Jianqin et al., 2014] developed an efficient localization and tracking algorithm for robot sensor fusion in an intelligent house. Using JPDA to fuse data from static laser range finders and cameras, the algorithm enhanced detection and localization in an intelligent space extending the perceptive ability of the robot and its computing power to the environment itself.

Although explicitly treated in dedicated applications [Indelman et al., 2014], the basic JPDA algorithm cannot initialize new sensory modalities or remove their contributions. Furthermore, when applied to scenarios in which there is a high number of different sensory modalities and consequently a high number of hypotheses, JPDA proves to be intractable due to high computational costs. To alleviate this drawback [Gorji et al., 2007] provided a modified JPDA filter to combine multiple sensors for efficiently tracking multiple mobile robots during object manoeuvring movements. Extending [Gorji et al., 2007] work in [Schultz et al., 2003] designed a sampled version of the JPDA for flexible people motion tracking using mobile robots, but only considering Gaussian sensory data distributions and linear sensor dynamics. Extending the capability of the JPDA to multiple sensors and arbitrary sensory data distributions [Vermaak et al., 2005] proposed the Monte Carlo JPDA filter, but only tested in a synthetic tracking scenario.

In order to surpass computational costs for sequential execution of the algorithm, and taking into account the constantly increasing capabilities of multiprocessor and networked systems, distributed versions of the JPDA were developed.

Starting from a distributed sensor network with peer-to-peer communication protocol and distributed processors, [Battistelli et al., 2014] developed a JPDA association for multisensory fusion by processing local sensor measurements, exchanging data with the neighbours, and fusing local information with information from the neighbours. The approach proposed a Cheap Joint Probabilistic Data Association (CJPDA) filter by devising suitable distributed consensus-based procedures for sensor fusion for surveillance applications. Due to its concurrent implementation capabilities this approach depends on the correlation between individual hypotheses and reflects the influence of current observations in the joint hypotheses. Furthermore, to make this approach feasible, real-world implementations need to make sure that node communication exists after every sensory observation, and there are acceptable approximations when communication is sporadic or when there is a high amount of noise in the sensory contributions.

Multiple Hypothesis Test

Using more than two consecutive sensory observations, unlike the PDA and JPDA, the Multiple Hypothesis Test (MHT) minimizes the probability to generate an error and extracts associations more precisely. This method evaluates all hypotheses and maintains new hypotheses in each iteration.

In its initial formulations [Reid, 1979, Morefield, 1977] the MHT was developed as an iterative algorithm which, starting from a set of correspondence hypotheses given as a collection of sensory observations windows, computed predictions. Subsequently, the predictions were compared with incoming observations given a certain metric. The base to create new hypotheses in each iteration is given by the set of extracted associations in the current iteration. For each incoming sensory sample MHT maintains various correspondence hypotheses for each sensory modality.

For a given hypothesis $\underline{H}(k)$ at time k, $\underline{H}(k) = [h_l(k)], k = 1, ..., n$, the probability of the hypothesis $h_l(k)$ is given by

$$P(h_l(k)|\underline{Z}(k)) = P(h_g(k-1), a_i(k)|\underline{Z}(k)), \tag{2.19}$$

where $h_g(k-1)$ is the hypothesis of the complete set of sensory observations until time $k-1$; $a_i(k)$ is the $i-th$ possible sensory association; and $\underline{Z}(k)$ is the set of sensory observations.

The MHT can also detect new sensory modalities used in the fusion process, while maintaining the hypotheses tree structure, using a Bayesian decision model techniques for data association and fusion

$$P(\lambda|\underline{Z}) = \frac{P(\underline{Z}|\lambda)P(\lambda)}{P(\underline{Z})}, \tag{2.20}$$

where $P(\underline{Z}|\lambda)$ is the probability of acquiring the set of sensory observations $\underline{Z}$ given the new signal λ, $P(\lambda)$ is the prior probability distribution of the new sensory modality, and $P(\underline{Z})$ is the probability of obtaining the set of observations $\underline{Z}$.

MHT is an exhaustive approach, as it considers all hypotheses, and computes the possibility of association after each acquired sensory sample without assuming a fixed number of sensory modalities. The main disadvantage of this data association method is the computational cost, which has been shown to grow exponentially with the number of sensors and observations. An interesting approach using MHT to extract associations in large heterogeneous datasets was proposed in [Rahnavard et al., 2013]. The model was able to handle datasets of mixed data types: categorical, binary, continuous. Rather than checking all possible associations, the model prioritized computation such that only statistically promising candidate variables are tested in detail. Finally, this approach was able to limit false associations and loss of statistical power attributed to multiple hypothesis testing. An illustrative overview of the model is depicted in Figure 2.10. Practical implementations usually extend the basic formulation to judiciously exploit processing and storage capabilities of today's computing platforms. Trying to take advantage of the MHT capabilities [Joo et al., 2007] proposed a mechanism which associated sensory measurements and sensory cues in a many-to-many fashion. Using combinatorial optimization the algorithm tried to extract the best set of association hypotheses, outperforming other methods providing only approximations. Using a similar principle [Coraluppi et al., 2011] used a recursive hypotheses processing algorithm over a class of associated hypotheses instead of on a single hypothesis. Building a robust hierarchical multiple hypothesis tracker for tracking multiple objects in videos [Zulkifley et al., 2012] dealt with the problems of merging, splitting fragments and occlusions, combining camera measurements from foreground segmentation and

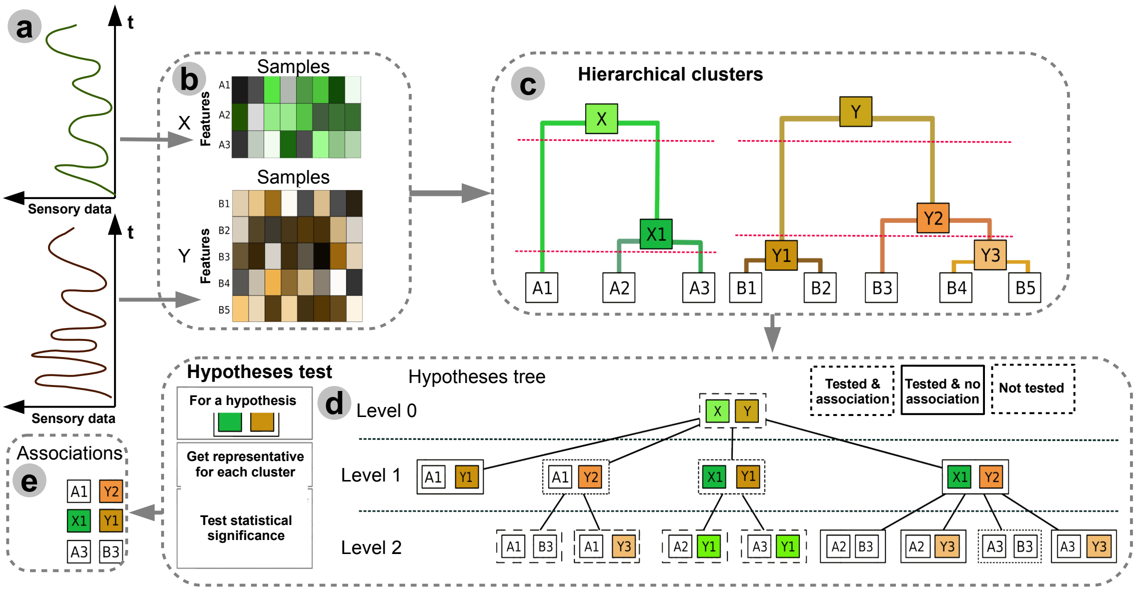


Fig. 2.10: Data association process using MHT. a) Sensory data; b) Data representation (features vs. samples); c) Hierarchy of clusters to select features according a metric; d) Hypotheses tree formation through tests of similarity metric and statistical significance; e) Report significant associations. (adapted from [Rahnavard et al., 2013])

clustered optical flow. The method used two levels of association. The first level focused on obtaining stable velocity values while multiple associations are utilized for better observation assignment, whereas the second, the occlusion predictor was used to distinguish merge, occlusion and brief interference. In a slightly different scenario, [Tsokas et al., 2012] presented an adaptation to the MHT method, which unlike classic MHT, allowed for one-to-many associations between sensory cues and observations in each hypothesis production cycle. The method provided good results in a multi-robot tracking scenario involving multiple sensors. Finally, [Brekke et al., 2015] proposed a multi-hypothesis solution to the simplified problem of simultaneous localization and mapping (SLAM) that arises when only two measurement frames are available. The model provided a Gaussian mixture approximation of the posterior density of pose displacement. Data association using MHT has been incorporated in the model in order to make this approximation as reliable and efficient as possible.

Similar to JPDA, the MHT method has been extended towards a distributed representation, the Distributed Multiple Hypothesis Test (MHT-D). In the first stage of the algorithm (i.e. the hypothesis formation) for each hypothesis to be fused a new association is created, based on observations coming from all distributed sensory nodes. Subsequently, in the second stage (i.e. hypothesis evaluation) the likelihood of the possible associations and the obtained estimation at each association are calculated. Although the main disadvantage of MHT-D is the relatively high computational cost when facing a high number of associations and a high number of sensory variables to be estimated, generic improvements were developed [Lawson et al., 2015]. Given a visually guided robot manipulation task, the developed model used clustering-based extension to MHT data association, providing more efficient and precise approximations compared to existing approaches and using a fraction of the computation time.

The MHT framework was also found to describe processes of perceptual organisation known to occur in the brain [Feldman, 2013]. This study postulated that a single proximal stimulus is consistent with an infinity of possible scenes of which only one is perceived. Following a Bayesian framework, the model proposed that our brain is able to fast and robustly build an internal representation of the external stimulus using the available information from the sensors and a process defined as unconscious inference. This concept was first defined by the German physicist and polymath Hermann von Helmholtz to describe an involuntary, pre-rational and reflex-like mechanism which is part of the formation of visual impressions. Bayesian formalism was used to infer the most plausible interpretation of the sensory data.

Given sensory data, $\underline{D}$, which has a variety of hypothetical causes, $H_1, H_2, ..., H_n$, then the algorithm checks if a hypothesis H_i is plausible. The considered hypothesis must be plausible in proportion to the product between the probability that for different hypotheses H_i in the environment being true we can recover the sensory data $\underline{D}$. Furthermore, the model recovers the prior knowledge about the hypothesis H_i (its statistics) and how likely the hypotheses can be found in the environment. This quantity is then divided by the prior probability distribution of the sensory data $\underline{D}$, given as

$$P(\underline{D}) = \sum_{i=1}^{n} P(\underline{D}|H_i)P(H_i). \tag{2.21}$$

In this formalism, the posterior distribution, reflecting the belief that the hypotheses are true given the data, is continuously refined using the incoming sensory data,

$$P(H_i|\underline{D}) = \frac{P(\underline{D}|H_i)P(H_i)}{\sum\limits_{i=1}^{n} P(\underline{D}|H_i)P(H_i)}. \tag{2.22}$$

The "likelihood swamps the prior" such that the influence of the likelihood over the prior distribution increases, limiting prior's contribution to the posterior given incoming observations.

This Bayesian approach for perception is a mean of quantifying the degree, the strength of belief in any hypothesis (if at all), under the presence of uncertainty. The problem that usually occurs when using Bayes' rule to describe how plausible some hypotheses are given the data, is defining the prior, as sometimes this information is not accessible at all.

Graphical Models

Graphical models define a series of techniques, built upon graph-theoretic representations, for describing intrinsic relations between the states in large probabilistic models. Moreover, these models are useful for providing efficient data representations for inference, prediction, and fusion. In its basic formulation a graphical model represents the conditional decomposition of a joint probability distribution into a product of factors. Each factor depends on only a subset of variables.

Two major classes of graphical models were developed. One is capable to encode causal relations hidden between random variables (i.e. directed graphical models: Bayesian networks) and the second one is able to encode soft constraints between random variables (i.e undirected graphical models: Markov random fields). The powerful representation and processing capabilities of graphical models, given sensory uncertainty, were efficiently used in solving data association problems arising in multiple target tracking with distributed sensor networks [Chen et al., 2005, Chen et al., 2006]. After considering the problem in terms of statistical dependencies between random variables encoding sensory contributions, the data association problem resumed to an inference problem solved efficiently by belief propagation through local message-passing algorithms.

This technique solves optimization problems in a distributed manner by exchanging information among neighbouring nodes on the graph. Furthermore, a re-weighted version of the max-product algorithm [Weiss et al., 2001], was able to solve the inference problem, yielding provably optimal data association. In a more complex scenario, for rapid multi-floor indoor map building, using a body-worn sensor system fusing information from RGB-D cameras, LIDAR, inertial, and barometric sensors, [Fallon et al., 2012] used a graphical model to handle and to organise initially disconnected sets of sub-maps in the environment. Using an extended Factor Graph (FG) formulation to encode sensor measurements with different frequencies, latencies, and noise distributions [Han-Pang et al., 2014] proposed a real-time navigation approach that is able to integrate many sensor types while fulfilling performance needs and system constraints. In a visual tracking scenario [Castaldo et al., 2014] developed a system based on graphical models (i.e. Bayesian Factor Graph) which fused real-time data coming from sensors, along with estimates coming from the tracked object models. Sensory information was merged within environmental constraints in order to provide the best estimate of the state of a moving object. Factor graphs allowed the information to flow bidirectionally, to predict future values, and to strengthen the knowledge of the past in a challenging automatic localization of moving objects. Using factor graphs, [Indelman et al., 2013] developed a new sensory data association and fusion mechanism for high-rate information fusion in inertial navigation systems, that usually have a variety of sensors operating at different frequencies. The flexibility of the model was provided by the fact that the joint probability of all states was represented using a factor graph. This approach fully exploited the system sparsity and provided a plug-and-play capability to easily accommodate the addition and removal of measurement sources. The model presented a generic approach for using graphical models in data association, exploiting the underlying correlation structure of the sensory sources, as shown in Figure 2.11. The model was validated using real IMU and vision data that was recorded by a ground vehicle. In their model a factor represented the general concept of an error function that should be minimized. This approach to design a measurement model that predicts a sensor measurement given a state estimate is common in robot navigation literature. The factor captured the error between the predicted measurement and actual measurement and was able to register and un-register sensors based upon their availability.

Using the intrinsic capabilities of distributing computation, various distributed schemes for data association using graphical models were developed. By exchanging messages between sensory source nodes (e.g. n nodes) in parallel, each sensory source has n possible

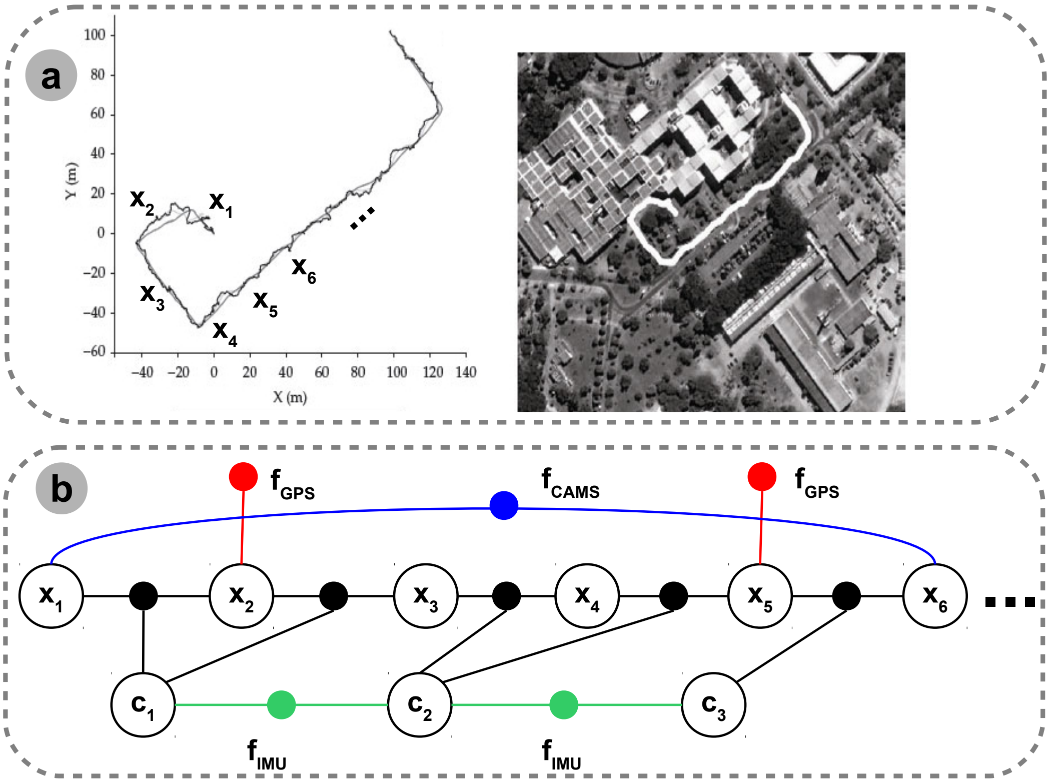


Fig. 2.11: Data association process using Factor Graphs for inertial, geolocation (GPS), and vision data fusion for ground vehicle navigation. a) Sample travelled path and navigation states; b) Factor graph for data association and fusion. Factors $f_{GPS}, f_{IMU}, f_{CAMS}$ connect navigation nodes x_i (i.e. states comprising position, velocity and orientation of the robot at time t) and bias/calibration nodes c_i. Factors have formulations for different measurement model, specific to each sensor.

combinations of associations. If there are M variables to estimate the complexity is just $O(n^2M)$, which is lower than the typical MHT-D approach (i.e. $O(n^M)$). Furthermore, linking parallelisation capabilities of graphical models to previously introduced architectures for multisensory fusion, [Makarenko et al., 2009] provided an in-depth analysis of graphical models approach to decentralised data fusion. The analysis provided a graphical model description for decentralized data fusion systems in large networks of sensors subject to rapidly varying topology changes and to issues of data delay. Interestingly, the work proposed an implementation that assembled the network using a decentralised spanning tree algorithm to enlarge the types of sensory models to hybrid distributions. The model also accommodated sparse feature descriptions, non-linear relationships, and supported generic applications such as SLAM.

Canonical Correlation Analysis

Another technique combining statistical analysis and data space properties of the input sensory data is Correlation Analysis. Using Canonical Correlation Analysis (CCA) [Mandal et al., 2013] proposed a model capable to extract out the relationship between two sets of multi-dimensional random variables, focusing on the correlation between a linear combination of the variables in one set and another linear combination of the variables in the other set. In the simplest scenario the method considered two variables encoding sensory data. From observations of the two random variables the method found the two weight vector directions such that the distance metric (i.e. the alpha-beta divergence) between the joint distribution of each variable times weight and the product of marginal probabilities of variables is maximized. The algorithm's goal was to find the weight vectors from the observed variables as a result of an optimization process, such that each variable times the corresponding weight vector are as much dependent as possible, maximizing the divergence. Finally, the method was able to reconstruct both hidden linear and non-linear relationships between the weighted variables, even in the presence of moderate amounts of noise. Trying to extend the capabilities of the CCA adaptive learning rules were proposed [Becker et al., 1996]. Moreover, combining the optimization process with gradient descent, to ensure convergence, various neurally inspired [Lai et al., 1999] and machine learning [Lai et al., 2000, Pezeshki et al., 2003] algorithms were developed.

Probabilistic and Possibilistic frameworks

In order to combine uncertainty representation capabilities of probabilistic models and evidence representation of possibilistic models, hybrid approaches were developed. Providing a generic view over data association for sensor fusion [Appriou, 2014] investigated the use of a hybrid probabilistic and fuzzy logic model to represent and infer matching sensory observations originating from different streams while considering uncertainty and efficient knowledge propagation. The perceived domain of each sensor was described as a resolution cell (i.e. highest probability density of meaningful observations in the given range of the perceived feature). In order to extract the association pattern, the model proposed to find the most likely singleton (i.e. probability mass) in the set of distributions given by the intersections of resolutions cells from all sensory modalities. The conceptual framework is depicted in Figure 2.12 for a simple object detection task.

In this scheme, sensory complementarity enriches the information content in terms of similarity information. Moreover, it exploits the dependency which might exist between sensory sources when they perceive the same feature or react to the same event. This procedure assumes extracting the probability mass function (i.e. singleton in the set of distributions) and use it simultaneously to handle spatial and temporal associations. If signals resemble each other, in the sense of a relation characterized previously on the basis of the physics at play, the sensory observations describe the same object or feature, so they can be associated in the current frame of discernment. The dependency will typically take the form of a belief function built upon a joint probability distribution and a fuzzy relation. Although the method proposed specific procedures in processing all possible intersections in the sensory space, the core idea is to compute the likelihood relating to the presence of a sensory observation at the intersection of the data distribution ranges (i.e. resolution cells) and a reliability score associated with that. For the scenario depicted in Figure 2.12b, for each of the two modalities (i.e. stereo vision and laser range finder) resolution cells, $\underline{x}_1^n$ and $\underline{x}_2^m$ are encoding the data distribution which determine the likelihood C^{nm} and a reliability score q^{nm}, relating the presence of a target at their intersection $\underline{x}^{nm}$, as shown in Figure 2.12a.

Given the current frame of discernment, E^{nm}

$$E^{nm} = \{H_0^{nm}, H_1^{nm}\}, \tag{2.23}$$

where $H_0^{nm} = \textit{no target in } x^{nm}$ and $H_1^{nm} = \textit{one target in } x^{nm}$. The mass function, $\mu^{nm}(.)$ on E^{nm} for each intersection x^{nm} of resolution cells is given by:

$$\mu^{nm}(H_1^{nm}) = 0, \tag{2.24}$$

$$\mu^{nm}(H_0^{nm}) = q^{nm}(1 - C^{nm}), \tag{2.25}$$

$$\mu^{nm}(E^{nm}) = 1 - q^{nm} + q^{nm}C^{nm}. \tag{2.26}$$

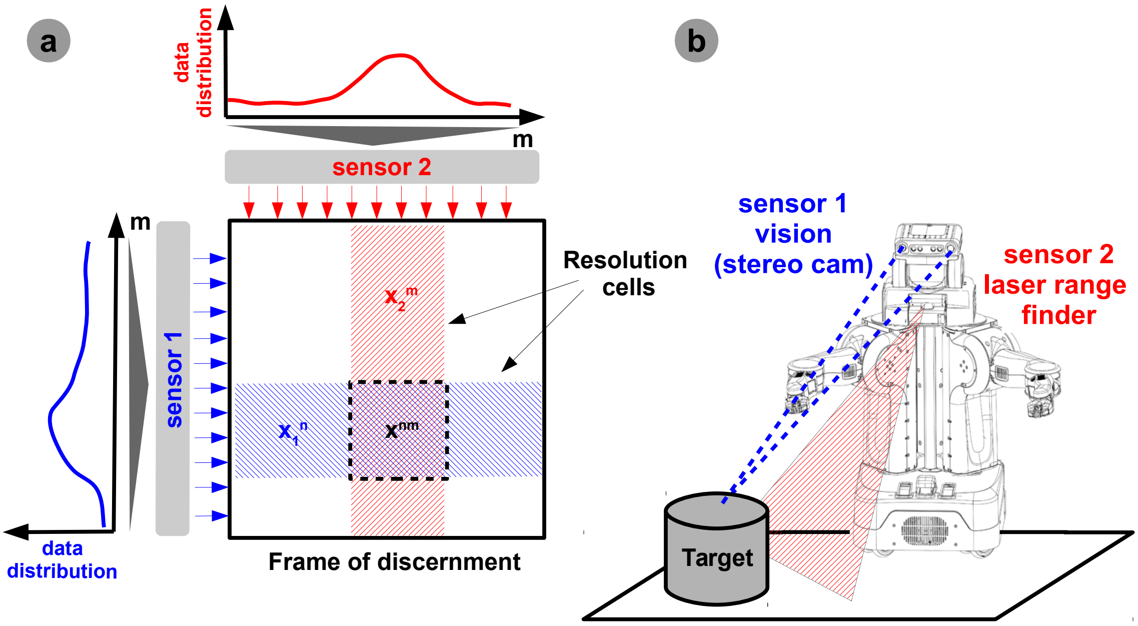


Fig. 2.12: Data association process in a probabilistic-possibilistic framework: an object detection scenario. Vision data acquired from a stereo camera is represented in the frame of discernment by a n-dimensional resolution cell. Range data coming form the laser is encoded in an m-dimensional resolution cell. In order to extract the position of the target, the algorithm decodes the overlap of the two resolution cells in the frame of discernment.

This approach ensured that the probability mass function can be used directly to handle spatio-temporal associations in the frame of discernment.

Clustering mechanisms

Cluster analysis and clustering mechanisms provide a powerful tool to explore intrinsic relationships in sensory data. Due to their heuristic approach, their application in real-world scenarios is fraught with potential biases. In a more broad view, data scaling, the selection of similarity metrics, choice of clustering algorithm, and even the order used in presenting the sensory observations, might considerably influence the extracted clusters. A typical processing flow for cluster analysis is depicted in Figure 2.13. The simplest data clustering technique is the nearest neighbours (NN) method. This algorithm provides a simple way to select or group similar values according to how close is a measurement to another given a certain distance metric. Usually, the type of metric is provided by the designer and is specific to the problem (e.g. absolute distance, Euclidian distance, statistical function of the distance). A big advantage of the NN method is that it can provide a solution or an approximation, in a timely manner. Sometimes, in the case of noisy observations and of complex cluttered environments it could provide erroneous results (i.e. false associations) which will determine the propagation of the error. In order to overcome the drawbacks in the NN algorithm, K-Means (Lloyd's) algorithm was developed. Basically, the K-Means algorithm finds the correct position of each of the K clusters centroids through an iterative process:

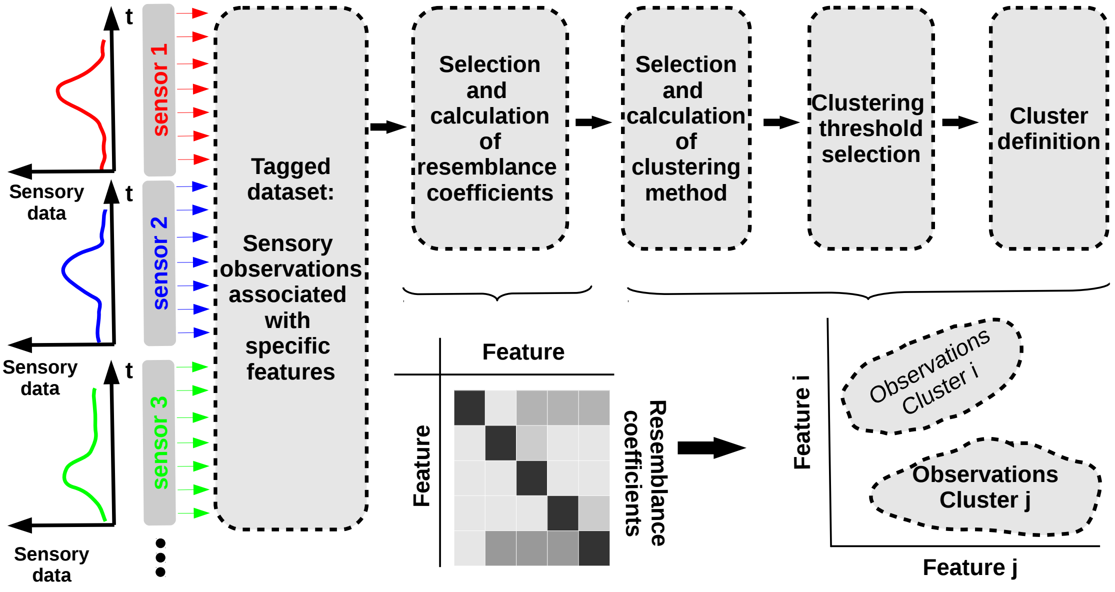


Fig. 2.13: Processing flow of cluster analysis mechanism for data association (adapted from [Hall et al., 2004]).

1. Get sensory observations and number of clusters;
2. Randomly assign the position of the centroid for each cluster;
3. Compare each observation with the centroid of each cluster;
4. Move the cluster centres to the centroid (mean) of the cluster;
5. If the centres still move (changes are bigger than a threshold), go to step 3.

The algorithm can be also seen as composed of two main steps: assignment (steps 1, 2, 3 in the process) and update (steps 4, 5 in the process). The assignment step is also referred to as expectation step, whereas the update step as maximization step, making this algorithm a variant of the generalized expectation-maximization (EM) algorithm. Since both assignment and update optimize a within-cluster sum of squares objective function, and there only exists a finite number of such partitions, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm, and the fact the number of clusters must be known a priori, limits its capabilities. Furthermore, even if improving the basic algorithm (e.g. modify the initial number of clusters, using fuzzy clustering assignments, or Bayesian techniques) most versions need to iterate through the dataset of observations in order to converge to a reasonable solution. This is a major disadvantage in real-world applications.

In order to cope with limitations in the NN algorithm [Shindler et al., 2011] proposed a fast and precise algorithm to simultaneously extract the structure of the input data while reducing the dimensionality of the input space. Extending K-Means algorithms to converge towards clusters with smaller number of centroids for any density of sensor networks , [Park et al., 2007] proposed an advanced optimization algorithm for sensor network clustering. Using the proposed clustering algorithm, redundant cluster centres are eliminated, and unnecessarily overlapping clusters are merged. The algorithm handled dynamic changes like node addition or die-out, while the network was in working state.

In some cases the number of clusters is not known. For data fusion the association algorithm should extract by itself the number of clusters and subsequently perform clustering. For such scenarios an already established neurally inspired algorithm was proposed, namely Self-Organizing-Maps (SOM). Providing a relatively simple dimensionality reduction technique, the SOM is able to extract the probability distribution of the input space while keeping its topological representation. Using this algorithm [Wan et al., 2000] provided a model to explore discrimination information from the data itself. The model had the capability to extract and represent high-order statistics of high-dimensional data from disparate sources in a non-parametric, vector-quantized fashion. The model targeted remote sensing applications under various sensory data sources providing good data clustering and joint spatio-temporal classification capabilities. Using relatively similar principles [Leivas et al., 2010] proposed a model for sensor fusion based on multi-Self-Organizing Maps for SLAM, while in a biologically inspired model of sensor fusion [Bauer et al., 2012] proposed an algorithm that learned sensors' reliabilities for different points in space, and used their associations and reliabilities to perform fusion.

Going away from engineered approaches for data association, but still using the same mathematical apparatus for clustering, [Mayor et al., 2010, Althaus et al., 2013] analysed and proposed models of cross-modal interactions in early word learning in human infants. Using relatively similar clustering mechanisms (i.e SOM) for categorization and labelling, the models proposed candidates for categorical perception from early audio-visual interactions that could play a role in the facilitation of infants' categorization through verbal labelling. Both models offered efficient generalisation of word-object associations such that the association between the paired object and its corresponding sound pattern was generalised, automatically building associations between all objects in its category to all sound patterns of the appropriate type.

Neurally inspired data association

Multisensory fusion influences many aspects of human perception, cognition and behaviour. Data association plays an important role in current models of multisensory fusion and determined two parallel research directions. The first direction is based on the idea that one sense "educates" another and provided the ground for sensory dominance research. A second approach focused on the low-level neurophysiological evidence of sensory association and combination at the neuron level following principles of temporal synchrony, spatial congruency and inverse effectiveness (i.e. as the responsiveness to individual sensory stimuli decreases, the strength of multisensory integration increases) [Spence, 2012].

Bringing together and evaluating evidence concerning how the brain attempts to organise the perceptual scene across sensory modalities [Spence et al., 2012], we will highlight some studies that have investigated how perceptual organisation in one sensory modality is used to organise the information that is simultaneously perceived in another sensory modality. This overview aims at extracting the principles behind data association and correlation learning in neural systems as a base for multisensory integration.

Using biologically plausible mechanisms [Cook, Jug et al., 2010] proposed a model of unsupervised learning of functional relationships from sensory data. After learning, the model inferred missing quantities, given the learned association relations and available sensors. Moreover, due to recurrent connectivity, the sensory representations were continuously refined, de-noising the encoded real-world variable. Finally, due to the constraints imposed by the learned relations, the model was able to combine consistent and correlated data and discriminate and penalise inconsistent data contributions. In a slightly different scenario [Weber et al., 2007] proposed a model for extracting coordinate transformations in a robot navigation task. Inspired by sensorimotor transformations in the prefrontal cortex, the algorithm produced invariant representations and a topographic map representation of the scene, guiding robot's behaviour. Finally, in an attempt to counteract the drawbacks in probabilistic data association techniques based on canonical correlation analysis (CCA) and principle components analysis (PCA) for clustering, [Hsieh, 2000] proposed a neural network model able to implement nonlinear canonical correlation analysis. The model was able to extract the underlying nonlinear structures between two sets of variables under moderate noise conditions. The proposed model treated the input variables evenly, in that they are both inputs, and no causality is assumed. This approach offered the capability to perform inference in the case one variable is missing by using the learned data association.

Extending the problem of extracting sensory data associations to the extraction of invariant features of temporally varying signals [Stone et al., 1995] proposed an invariance extraction learning algorithm based on a linear combination of Hebbian and anti-Hebbian synaptic changes, operating simultaneously upon the same connection weights but at different time scales. The model was inspired by the fact that inputs to retinal photoreceptors tend to change rapidly over time, whereas physical parameters underlying these changes vary more slowly. Accordingly, if a neuron codes for a physical parameter then its output should also change slowly, despite its rapidly fluctuating inputs. This model has been shown to be sufficient for unsupervised learning of simple spatio-temporal invariances.

Guided by similar principles, more recent work [Wiskott et al., 2002] proposed the slow feature analysis (SFA) method to learn invariant or slowly varying features from vectorial input signals. The method was based on a nonlinear expansion of the input signal and the application of PCA to this expanded signal and its time derivative. In order to extract relations between sensory streams the method was applied hierarchically to process high-dimensional input signals and extract complex features. Presented as a simple model of the visual system, the algorithm learned translation, size, rotation, contrast, and, to a lesser degree, illumination invariance for one-dimensional objects, depending on only the training stimulus.

Using a different neurally inspired substrate, [Taylor et al., 2010] proposed a model for learning consistent features from understanding video data. The model learned latent representations of image sequences from pairs of successive images. The convolutional architecture of the network model allowed it to scale to realistic image sizes whilst using a compact parameterization and providing an extension to another unsupervised learning algorithm, the Restricted Boltzmann Machine (RBM). Extracting the underlying spatio-temporal features in the sensory streams, the model learned to represent optical flow and performed image analogies being able to perform human activity recognition.

Going into more detailed neural analysis and psychophysical studies, [Tonia et al., 2001] provided insight in the temporal dynamics of functional segregation at the basis of visuomotor associative learning in humans, isolating specific learning-related changes in neurovascular activity across the whole brain. The findings proposed by this study suggest that specific cortical areas are critical for integrating perceptual information with executive processes given learned visuomotor associations.

Coming back to neural computational models for sensory correlations and association learning [Seung et al., 2000] proposed a change in paradigm in terms of perceptual representation such that computational power is leveraged: the manifold ways of perception. The paradigm proposed reducing dimensionality of the perceptual problem by finding low-dimensional structure in it using measures of local geometry of a manifold. Using this principled description and representation [Saul et al., 2003] introduced the locally linear embedding (LLE), an unsupervised learning algorithm that computed low dimensional, neighbourhood preserving embeddings of high dimensional sensory data. In this context, high-dimensional sensory data was mapped into a single global coordinate system of lower dimensionality in which computation was simpler. The model was successfully used in extracting primitives from images of faces, lips, and handwritten digits.

Finally, in a more recent study, [Law et al., 2008], focusing on perceptual learning in a visual discrimination task, it has been shown that perceptual learning does not appear to involve improvements in sensory representation, but rather how sensory representations are interpreted to form the decision that guides behaviour.

2.4 Summary

Handling the wealth of available sensory modalities yields an adaptive and robust substrate for representing, extracting, and processing the underlying information encoded in the perceived streams. Either for estimating certain (salient) features in the environment given sensory observations, or for extracting associations among available sensory cues to guide behaviour, multisensory fusion unravels as a complex process. Furthermore, it requires a suitable architectural substrate and processing paradigm. This section introduced the basic types of processing architectures, along with their algorithmic substrate, and various sample implementations to clearly emphasize the rich design space.

Starting by exploring the underlying relations between input sensory sources, we analysed multisensory fusion mechanisms exploiting complementarity, redundancy, and cooperation to fuse data from various sources at the signal level. Going away from the low-level signal representation of sensory data, we further analysed architectures employing various abstraction levels, such that the informative content of the sensory data was extracted and propagated to higher representation levels for decision-making. Finally, we investigated how these high-level (feature) representations, subsequently combined with low-level representations in complex architectures, provide a nexus platform for multisensory fusion capable of inference, discrimination, and decision-making.

The initial overview over sensory data representations provided the framework to introduce and evaluate the capabilities of the various architectures typically used in multisensory fusion applications. Alternating between centralised, pipelined processing architectures and fully distributed architectures, we introduced relevant sample applications in which various sensory modalities were combined using local and global operations applied on perceived low-level signals or high-level features. Balancing the advantages and disadvantages of every scheme, we extracted important aspects valuable at the design stage. Moreover, we identified those core principles that fully exploit the architectural, processing, and data representations for robust and adaptive multisensory fusion.

The introductory section was completed with a formal analysis of the computational substrate in state-of-the-art multisensory integration algorithms. Starting with the investigation of standard algorithms for state estimation, we focused on extracting those driving principles in current designs and provided a putative view of the underlying formalism. Utilising Bayesian theory as a unifying framework to represent uncertainty and process probabilities, we analysed the basic MLE, MAP models, as well as the Kalman filter, and the powerful Particle Filters in various scenarios, emphasising their main strengths and advantages in real-world real-time implementations. Likewise ”engineering highlights”, we extended our evaluation towards neurally inspired approaches and implementations, focusing on their applicability and advantages in real-world scenarios.

The overview on the algorithmic substrate was complemented by a formal introduction and analysis of representative approaches for data association. Providing a comparative formal description of both engineered and neural approaches, we focused on emphasizing the need for an adaptive substrate capable to learn the underlying regularities in concurrent sensory streams and exploit this highly informative cue to improve the quality and precision of the integration process. From methods like PDA and MHT, to graphical models and from CCA to SOM and PCA, we delineated those fundamental principles underlying the detection, extraction, and interpretation of underlying inter-sensory correlations supporting the fusion process.

After providing an overview of state-of-the-art approaches, we now turn towards formalising and analysing the capabilities of our novel approach to multisensory fusion.

3 Formalising a model for multisensory fusion

The current chapter introduces the motivation and the functional details behind the proposed framework. Starting from the new computational paradigm employed in the framework, its (neuro-)biological inspiration and advantages, we will further introduce all those principles which differentiate it from traditional approaches to computation.

Nowadays we experience the ubiquitous power and success in problem solving of the ”traditional” approach to computation, as pioneered by von Neumann. Using precise mathematical descriptions of the input-output transformations, these systems provide excellent solutions to a large range of problems. However, trying to extract and interpret useful information from the noisy real-world data has been resistant to straightforward solutions. In order to cope with this limitation, elaborate theoretical reasoning, algorithmic complexity, and significant processing resources are required.

In our work, we propose an alternative computational architecture, inspired by the high-level architecture of the mammalian cortex, where computation is performed in a widespread network of interconnected units, each representing a different type of information about a feature or quantity of a system, or the state of the environment in which the system operates. The connectivity between the units describes known formalized relations (e.g. equations) and computation takes place by each unit trying to be consistent with the other units it is connected to. This system is able to generate a coherent, but distributed, representation of the current feature or state of interest, given the noisy and uncertain percept.

In contrast to traditional computational architectures where a central processor executes precise instructions over data available in memory, we propose a paradigm in which processing and storage is local, distributed, and intermeshed. This blending of information with local dedicated processing is inspired by the brain and provides the core principle of our approach. We show that this new computational architecture enables real-time multisensory fusion and interpretation capabilities, while being fast, robust, and scalable compared to traditional approaches.

The difference between traditional information processing systems and our approach lies largely in the completely different architectures they employ, specifically in differences at representation level, storage, and processing of information. Computers use reprogrammable, high performance CPUs to process data fetched from and stored to memory, whereas in brains neural processing and synaptic data storage are completely intermeshed, with each cortical area being responsible for both memory and processing.

3.1 Probing neurally inspired processing mechanisms

Where do sensory relations come from? We previously emphasized that in our model each sensory modality is individually represented in a network unit, whose dynamics uses formalized relations to achieve consensus given sensory contributions. This process assumes that each unit tries to be consistent with the other units it is connected to. Relational knowledge and representations which describe associations amongst sensory signals, is a hallmark of human cognition [Christie et al., 2010]. Yet, how this high-level associations are represented in the neural substrate is still unknown. There are many aspects and models known to describe cortical architecture and processing and even more unknown aspects. In our work we have abstracted a set of principles known to describe cortical processing to yield a new style of computation.

Core principles of a new style of processing

Distributing representation and processing

One of the main architectural principles validated in neuroscience is that the cortex can be divided into areas. Each area deals with a particular form of sensory information, and areas dealing with related forms of information are reciprocally connected. Providing an interesting perspective on understanding cognition through large-scale cortical networks [Bressler, 2002] proposes that characteristic adaptability of cognitive functions seems to derive from large-scale networks in the cortex. These networks are able to repeatedly change the state of coordination amongst their constituent areas on a fast timescale. The interdependence between interacting cortical areas is balanced between integrating and segregating activities. From a high-level point of view, cortical areas, through their coordination dynamics, are thought to rapidly resolve a large number of mutually imposed constraints, leading to consistent local states and a globally coherent cognition. Although specific operations reside in individual cortical areas, complex cognitive functions require the joint operation of multiple distributed areas acting in concert [Wang et al., 2014]. Starting from these cognitive implications, it is generally believed that cortical areas, because of their unique topological positions in the overall connectional structure of the cortex, process information in specialized cognitive domains. The specification of these domains may be general (e.g.: visual, auditory, tactile, motor) as well as more specific (e.g.: speech sound subdivisions, inter-aural time difference (ITD), inter-aural level difference (ILD) processors), but mark a clear classification of areas in coarse specialization areas and fine specialization areas. Supporting this view, a more formal study carried out in [Bressler, 1995] showed that inherent in the concept of the large-scale networks models is the premise that neurons in different areas become functionally connected supporting the complex operation of the network. Attributes like co-incidence, co-localization and synchronization are defining the correlated activity in the interconnected cortical areas. Furthermore, the control of the large-scale networks is based on parallel processing and achieved through efficient coordination of information transactions. Summing up, according to this study, *elementary functions (i.e. encoding, representation, de-noising) are localized in discrete cortical areas, whereas complex functions (i.e. association learning, integration) are processed in parallel in widespread cortical networks.* Control processes operating at cortical and sub-cortical levels dynamically organize and regulate activity in the large-scale cortical networks. Cortical areas in the network become functionally connected through direct recursive interaction. In an attempt to provide a unified framework describing cortical coordination dynamics and cognition, [Bressler et al., 2001] proposed an approach to understanding operational laws in cognition based on principles of coordination dynamics derived from simple and experimentally verified theoretical models. When applied to the dynamical properties of cortical areas and their coordination, these principles support a mechanism of adaptive inter-area pattern constraint that, the authors postulate, underlies cognitive operations in general. In the introduced framework, the cortical area is conceived as an organized set of locally interacting neuronal populations that receives synaptic inputs and sends axonal projections as a functional unit. The goal of the study was to address the question of how the interactions among the large number of anatomically distinct cortical areas give rise to the emergence of cognitive function in real-time, or concisely, what are the large-scale coordination dynamics of the cortex corresponding to cognitive dynamics. Finally, in supporting our first architectural principle, [Reggia et al., 2001] provided a high-level, hemispherical specialization and interactions model focusing on features like robustness and modularity. To support this high level description of inter-areal specialization and interactions, the model pointed towards hemispheric asymmetries and interactions. The model postulated that brain plasticity is a strong factor, and that the excitatory and inhibitory influences are modulating the activity in each hemisphere.

Connectivity induced functionality and mild external sensory influence

Another main architectural principle, fundamental in our framework is that, at cortical area level, the input from other areas provides only a small fraction of the input to the target area. Furthermore, most of the input to any area is internal and local, while incoming sensory information mildly influences processing, as supported by neurophysiological data (only 10% of sensory projections from thalamus project to cortex, [da Costa et al., 2011]). Finally, each local representation is structured, typically following a topographical layout (e.g. topographical visual field representation, somatotopic sensory representations). Supporting this design principle, [Passingham et al., 2002] proposed that the functions of a cortical area are determined by its extrinsic and intrinsic properties and showed that each cortical area has an unique pattern of cortico-cortical connections (the connectional fingerprint). The described approach proposes that each area has a unique set of extrinsic inputs and outputs and that this is crucial in determining which functions that area can perform. Introducing a new model for information processing in the cortex, [Knudsen et al., 1987] identified a potential hierarchical processing architecture using two types of processing units, serial and parallel maps. Depending on the dimension of a map, one can have computational maps (active uni-dimensional maps) and non-computational maps (derived multi-dimensional maps). The model is completed by the defined relationships between the maps at a functional level. In the proposed model, map generation is synonymous with a parameter evaluation process which is parametrized using the number of simultaneous mapped parameters and the parallel array of processors. Providing a detailed study on the influence of sensory exposure and structural arrangement in cortex, [Ringach, 2007] addressed visual maps formation and interactions. The basic structure of receptive fields and functional maps in the primary visual cortex is established without exposure to normal sensory experience (rather encoded in the expressed genes during development). But how the brain wires these circuits in the early developmental stages is still unknown. The proposed model is based on the idea that the blueprint of receptive fields, feature maps, and their inter-relationships may reside in the layout of the neural substrate along with the statistical connectivity scheme dictating the wiring between thalamus and cortex. The cortical map creation is focused on replying to two main questions regarding how the initial map establishment occurred and how can the activity dependent map plasticity and persistence be preserved. Focusing on visual cortex, [Thomas et al., 2004] introduced a formal analysis over the connectivity and the coupling of cortical feature maps in the visual system. Starting from the idea that topographic maps of activities of individual neurons signal the retinal location and angle of oriented elements in the visual field, the study introduced the development model of such maps.

Extending basic neuroscience principles

Finally, although neuroscience has not yet established a generic cortical processing model, we extended the aforementioned principles set, with two more additional principles which allow us to formulate a working framework. The first principle assumes that the specificity of the inter-areal connections represent the relation between the meanings encoded in the areas. The second principle states that the computation performed by each area tries to bring the encoded representation towards a state compatible with related areas. Achieving consensus ensures that the distributed representation is coherent although built upon local contributions.

Processing as constraint satisfaction towards consensus

In order to link principles previously validated by neuroscience to our additional design principles, we introduce some studies motivating the link between cortical area coordination dynamics and information processing. The study in [Bressler et al., 2001] found that in order to be effective in ongoing dynamic computation the cortex must resolve the large number of competing constraints acting on its component areas in a rapid manner. It was suggested that the cortex achieves this through a relaxation process in which it settles into a globally consistent state that satisfies the multiple constraints on its interacting component areas. A relaxation process describes the network in which the units have access to each other's responses and adjust their own responses accordingly. In this context a problem may appear. More explicitly, falling and settling in a stable state (e.g. local minima) where the dynamics is trapped into a fixed point. The cortex seems to avoid that because the cortical areas can reconcile their competing constraints through increased relative coordination. This is done without the need of explicit relaxation, rather by using an adaptive response to the current constraints on its component areas.

Processing driven by functional relations

Regarding the specificity of the inter-areal connections and the capability to encode the relation between the meanings encoded in the areas, [Knudsen et al., 1987] proposed a model of visual processing for which the interaction between the maps can be hard-wired (i.e. defined relationships) or non pre-wired interactions (activation based). Following this specific representation of inter-areal connections, [Reilly, 2001] introduced a new model for cortical computation based on collaborative cell assemblies. In the proposed model each region was "bound" to a different sensory modality and defined its representation. This representation was considered a mapping from the environment to the cell assemblies state space and contained mapped information (sensor input - motor outputs). The defined representation was context independent, composable and followed a dispositional form, supporting different levels of complexity in defining the dynamics. Furthermore, binding was introduced in the model as a collaborative interaction in which additional sources of information (e.g. features of stimulus) will constrain the identity of the represented environment.

Comparative survey on existing cortical architectures

In order to extend the discussion on cortical architectures and emphasize the relevance of the proposed approach, we survey and compare against other, neurobiologically plausible, models of cortex.

In the work of [Barbas, 2015] the emphasis falls on a structural model that relates connections to laminar differences between linked cortical areas. The core principle is that the pattern, strength, and topography of connections among cortical and subcortical structures enable a variety of functions to be realized in both excitatory and inhibitory neurons. These findings support the proposed model in terms of the functional substrate of computation with interacting maps, similar to cortical maps, implementing various functions with excitatory / inhibitory connections.

Looking directly at thalamocortical interactions and their non-homogeneous information processing pathways, [Sherman, 2012] proposed a model comprised of two main classes of processing pathways, one carrying information processing and a second one playing a modulatory role. The model describes parallel processing in cortex as modulated by thalamic inputs through relay areas responsible with both modulation and processing. This observation enforces the idea followed in the design of our model, where connectivity patterns among different representations of sensory quantities are modulated by intermediate maps responsible with relaying or enforcing the local estimate through a different relation (i.e. pathway).

Going away from the functional aspects, [Grossberg, 2007] proposed a unified theory capable to link brain mechanisms to behavioural functions. Using complementary computing and laminar computing as main ingredients, the LAMINART architecture describes how constraints can influence multiple cortical regions, and how sensory cues can work together to learn invariant categories (i.e. instantiated for visual development, learning, perceptual grouping, attention, and 3D vision).

Using a similar, high-level, description of cortical processing [Hawkins et al., 2006] proposed the Hierarchical Temporal Memory (HTM) as a machine learning technology that aims to capture the structural and algorithmic properties of the neocortex. Similar to our model, the HTM is based on different processing regions wired together in a network. Some regions receiving input directly from the senses and other regions receiving input only after it has passed through several other intermediate processing regions. Time plays a crucial role in adaptation, inference, and prediction. Both HTM and our model can infer missing quantities given the existing connectivity (e.g. relations / previously learned associations); can adapt to unforeseen changes in the input streams and keep the processes representation consistent; and finally can predict the likely values for future inputs based upon current input and immediately past inputs.

Supporting the idea of functional coupling among different processing maps in cortex, [Edelman et al., 2013] proposed reentry as a key mechanism for integration of brain functions. This mechanism describes the ongoing bidirectional exchange of signals linking two or more brain areas. The main hypothesis is that reentrant signalling serves as a general mechanism to couple the functioning of multiple areas of the cerebral cortex and thalamus and integrate functionality. Consistent with this paradigm, our model implements reentry as a process that facilitates the coordination of functionally segregated computational areas. By these means this process binds cross-modal sensory features similar to synchronized and integrated patterns of neural activity in different brain regions.

Finally, addressing the role of uncertainty in neural coding and computation, [Rao et al., 1999] proposed predictive coding as a neurobiologically plausible scheme for inferring the causes of sensory input based on minimizing prediction error. The core hypotheses supporting this perspective are: a) feedback connections among cortical areas are carrying predictions of expected neural activity in the target area while the feed-forward connections carry the differences between the predictions and the actual neural activity; b) recurrent connections are used to store and predict temporal sequences of input neural activity. The two fundamental principles are also considered in our framework through the mixed connectivity which, using local dynamics and storage, ensure global consensus, when predictions are identical with the actual local estimates. Moreover, due to the intrinsic constraints (i.e. relations) among the different maps (i.e. areas) encoding different inputs, the network is able to infer and predict missing quantities.

Based upon the comparative analysis and the core principles of this new style of processing, we now introduce the basic model of our computational framework.

3.2 From neural models to formal implementation

Supported by known neural, cognitive processing mechanisms, as well as formal problem solver implementations (e.g. CSP), our model builds upon fundamental distributed processing principles. The underlying principles make it a promising approach for multisensory fusion and support the paradigm shift toward flexible and robust processing.

As previously described, multisensory fusion assumes interactions between percepts in order to extract globally coherent representations given modalities' local interpretations. Typically, local sensory interpretations are correlated and obey constraints imposed by the physics of the sensors. Combining all these constraints in a network of possibly conflicting local interpretations and using a relaxation method to solve the inherent constraints, ensures convergence to plausible and possible global interpretations.

The brain resolves conflicting low-level visual hypotheses to obtain globally best representations from wide-spread networks of interacting local sensory interpretations [Hinton, 1976]. Usually, the difficulty derives from the fact that the local ambiguities (inherent in perception) must be resolved by finding the best global interpretation. Instead of extensive searches through the space of all combinations of locally possible interpretations, relaxation methods can be used. Easily parallelisable, this approach attains the best global interpretation, not just a good one as in a heuristic search.

Extending this view, with focus on computational aspects, representing knowledge and constraints between percepts can be viewed as a network of relations. A network of relations can, in principle, provide a deductive style of distributed computation capable of representation, learning, and generalization close to neural mechanisms for associative memory [Cook et al., 2004]. Each relation involves a given number of variables representing sensory inputs, such that any overall relationship amongst the variables treated in the network is distributed across the network. Furthermore, each relation encodes a configuration of values corresponding to the variables it relates.

Employing similar mechanisms in multisensory fusion, where some variables may represent understandable aspects of the modelled situation and some might not, ensures that all contributions and intrinsic correlation between sensory data are exploited. Handling multiple contributions and processing them in such a network is inherently distributed. This allows the network to converge to a solution by narrowing down the space of possibilities as much as possible given the input data streams. Hence, *the outcome of the relaxation process is a stable global representation of the perceived scene.* Formalising multisensory integration using this relational paradigm ensures quick and uniform convergence for any network topology, allowing networks to be as interconnected as the relationships warrant, with no independence assumptions required.

Supporting the formal basis imposed by relational networks, insight from cortical computation [Buneo et al., 2006] strengthens the view that global knowledge representations can be extracted from local interpretations and interactions. Models of cortical processing have shown that cortical neural structures, such as gain fields, appear to implement relationships between a small numbers of variables. One example is the three-way relation between two successive joint angles and the resulting composite angle, important for an animal using its body. Given any two of the values, this three-way relationship can be used to deduce the third value. Relational knowledge is definitely a hallmark of human cognition and the subject of a vast body of research [Halford et al., 1998] with an interesting focus on the processing of associations versus the processing of relations [Phillips et al., 1995]. Formalising two computational paradigms, association and relational processing, various neural net architectures were developed, with feed-forward networks implementing associative processing, while tensor product networks implemented relational processing. Relational processing has been shown to have the essential properties of symbolic processing in humans and higher animals. This supports the view that information processing capacity is not defined in terms of the number of items but in terms of the complexity of relations that can be processed in parallel.

From a formalised point of view, relation networks can be regarded as constraint satisfaction problems (CSPs). CSP provide a generic framework used for modelling and solving combinatorial problems, employing efficient algorithms to prune search spaces using a distributed paradigm. This framework can accommodate and characterize symmetries among problem entities, facilitating local changes to the solutions (interchangeability) towards reaching global consensus [Neagu, 2005]. Typical consensus networks are composed of integrating nodes (i.e. simple transfer functions, usually integrators) and static weights (which are fixed, without dynamics). In order to realize relational processing, consensus networks can be extended, such that each node contains a variable, representing its current belief of the consensus variable of the overall graph. Moreover, each node is implementing a transfer function that produces the current variable stored inside from the incoming and the outgoing flows. Node level dynamics in this framework takes steps towards minimising the mismatch between the incoming and the outgoing flows (conservation criterion of equilibrium). In order to reach consensus, incoming streams are penalised or enhanced by

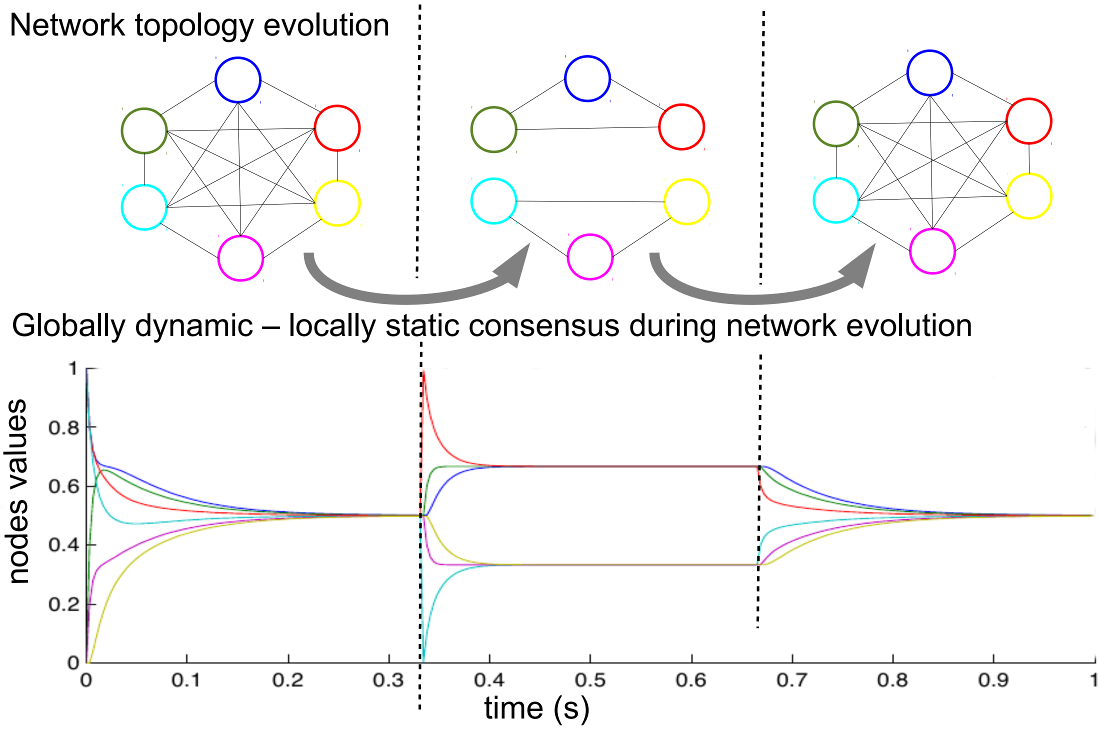


Fig. 3.1: Consensus networks: dynamical vs. static consensus protocol. Topology changes yield dynamic adaptation of the global consensus protocol, to accommodate the new connectivity pattern. For each configuration in the network's evolution, local consensus protocol is ensuring convergence.

corresponding correction or cross-correction weights responsible to increase convergence in the consensus protocol (i.e. fulfil the relations in the network). Such an approach models how global complex features can emerge from purely local rules, and how starting from initial random conditions and without any global supervision the system settles through a relaxation process, leading to the emergence of a global consensus [Kozma et al., 2008]. Finally, bringing the analysis to a higher level of generality, studying consensus over random information networks can be formulated as a quest for proving that the existence of information channel between a pair of units at each time instance is probabilistic and independent of other channels [Hatano et al., 2005]. In such a setting, the agreement protocol (i.e. configuration of relations) provides a means of coordinating the network elements towards achieving agreement on some particular parameter of interest represented in the network. Depending if the agreement protocol is fixed, and defines a state in which all elements in the network should agree on a certain value or there are probabilities on edges describing the communication channels between units, one can formulate the problem as dynamic or static. Figure 3.1 depicts the temporal dynamics of such a system which, given the locally stored quantities and the connectivity pattern, drives the global belief to consensus, such that all quantities are agreeing. Structural changes determine a change in the agreement protocol reflecting a change in the settling values of each quantity.

This analysis is relevant in the design of our model such that we need to make sure that the network connectivity exploits the relevant underlying relations in the data and uses that to achieve consensus and a global coherent representation, given incoming data and exclusively local processing.

3.2.1 Introducing the basic model

Humans can perform perceptual inference effortlessly, spontaneously, and with remarkable efficiency, given all the complex incoming sensory streams, as though these inferences are a reflex response of their cognitive apparatus [Shastri et al., 1993]. It has been postulated that the processing capacity is limited not by amount of information or number of items per se, but by the number of independent dimensions that can be related in parallel through relations [Phillips et al., 1995]. This enforces the idea that relational complexity, defined as the number of independent sources of variation (information) that are related, constitutes a major factor underlying the flexibility of higher cognitive processes.

We propose a distributed processing model which, given different input streams and the relations between them, settles in a stable state providing a coherent representation of the acquired quantities or derive new quantities.

Obeying constraints imposed by relations, each unit processes, stores, and communicates only local information, to the extent that each unit builds and refines its local belief about the represented quantity (e.g. sensory modality). Furthermore, both feed-forward and feedback processing pathways connect the units such that the mutual exchange of information is kept to a consistent state (i.e. fulfilled relations).

Each unit in our network contains a map based representation of a certain real-world quantity (i.e. perceived feature). The maps are inspired by the topographic organisation in cortical and midbrain structures for multisensory fusion [Carreira-Perpinan et al., 2005, **?**, Stein et al., 2004, Graziano et al., 2004, Swindale, 2005], and share the same functional role of mapping the sensory stimuli distribution to an internal representation. This map representation refers to a 2-dimensional topological arrangement (i.e. matrix configuration), such that adjacent values in the map encode adjacent values of the input space it represents.

In our framework the content of an individual map entry is determined by the feature space the map represents (e.g. 1D angular velocity scalar, 2D optic flow vector, 3D rotation vector). To get an idea about the map based representation and the way relationships are linking maps, we provide a toy example in Figure 3.2. Each map (i.e. a 2D structure with matrix like layout) is the basic template to encode n-dimensional features in each map cell. Cells in each map encode a multi-dimensional feature, depending on the sensory data type. As processing happens locally, the operations applied to each map are cell-wise, such that each update of the local estimate is performed independently at the cell level. The encoded quantity in each cell of a map can be represented by point estimates or using a sparse representation, encoded in neural population activity [Cook, Gugelmann et al., 2010, Pouget et al., 2004]. In a more general view, a map encodes the representation of a continuous stimulus parameter by a place-coded population response, whose peak reflects the mapped parameter [**?**]. Independent of representation (i.e. point estimate / population code) the core of the model is implementing relations that describe the connectivity of the network. Figure 3.3 introduces a canonical network that implements the identity relation between two units. Network dynamics is based on a random update process, using gradient descent. Values in each unit's map take small

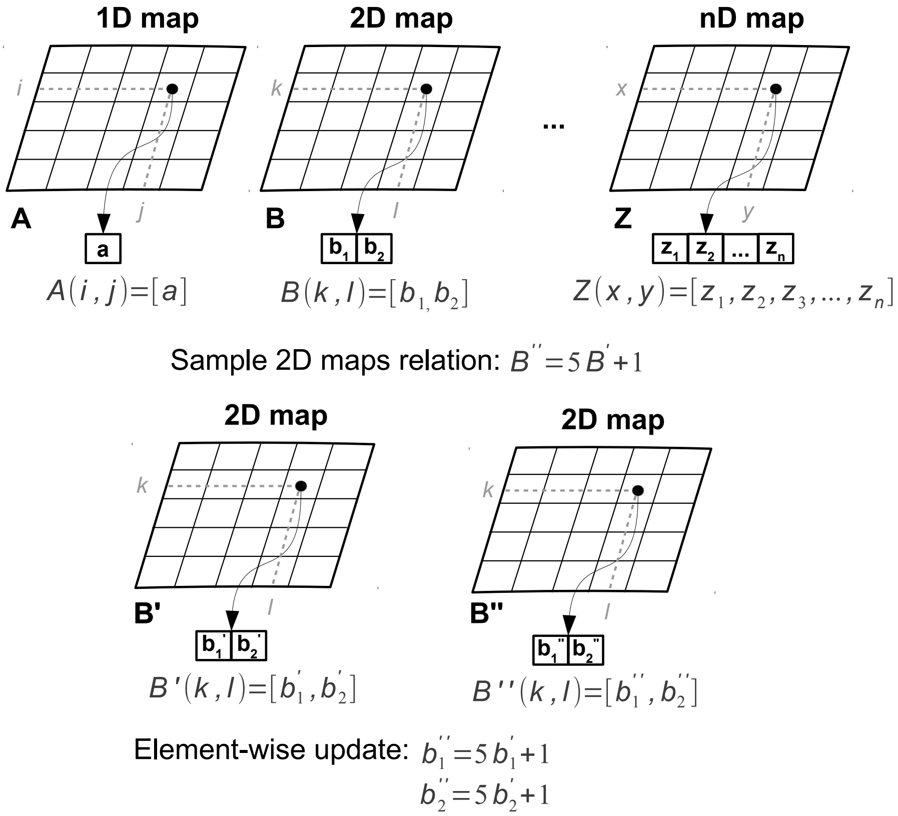


Fig. 3.2: Generic map based representation used in our model. Each map is a matrix-like structure in which each element encoding a certain sensory feature can be n-dimensional. Dimension is determined by the size of feature space a map represents. Sample implementation of an algebraic relation between two 2D maps: $B^{''} = 5B^{'} + 1$.

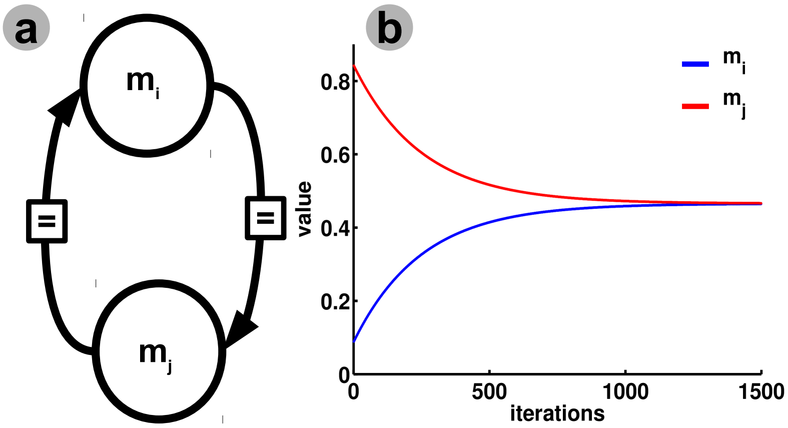


Fig. 3.3: Canonical network for identity. a) Mutual influence between two units, m_i and m_j, obeying update rules to minimise the mismatch between the values stored in each unit's map; b) Starting from initial random values the maps converge towards fulfilling the identity relation.

steps towards minimising the mismatch with the relations in which the unit is involved. In the case of absence of external sensory input (as shown in Figure 3.3), given the random initialisation of each unit's map, the network will converge to a solution in which the identity relations are fulfilled. Each unit follows update rules given by

$$\Delta m_i(t) = -\eta_{i,j}(t) \frac{\partial E_{m_i,m_j}(t)}{\partial m_i(t)} \tag{3.1}$$

$$E_{m_i,m_j}(t) = (m_i(t) - m_j(t))^2 \tag{3.2}$$

Equation 3.1 provides the update rule for map m_i. To minimise the mismatch with respect to m_j, given by $E_{m_i,m_j}(t)$, the map takes a step proportional to the mismatch, modulated by a factor $\eta_{i,j}(t)$. The mismatch (error signal) computation in Equation 3.2, is based on the squared error between the two units.

Applying relatively simple operations upon the locally stored estimate, each unit balances the influence from all the other units. Units can be linked using generic algebraic relations (e.g. summation, division, difference, or product), which can be employed to implement diverse and more complex relations. These ”atomic” operations are simple, keeping local processing fast enough to support fast network dynamics.

3.2.2 Analysis of the basic model

In this section we analyse the dynamics of the basic model. We provide an overview on convergence, precision, and adaptation capabilities in the presence external sensory input. As model systems, we consider two networks, one implementing relatively simple algebraic relations, and a second following a more complex scenario, coupling network units through highly nonlinear relations. The task the networks have to solve is to bring all quantities encoded in the network to agreement given dynamically changing external input and using only local processing and communication.

The first implementation of the model is depicted in Figure 3.4. Each unit in the network follows a connectivity pattern set by the embedded relation, such that the relation constrains the space of possible values a unit can take, given the values in the other units involved in the same relation and (eventually) the external input. External input only mildly influences the network dynamics, such that each unit balances external contributions and internal network belief, which is distributed across agreeing local estimates in each unit.

Starting from a random initialisation of the units and no external input, the network rapidly settles in a stable state. This state corresponds to a solution of each of the embedded relations. In order to converge to a solution each unit processes the values received from the other units (through bidirectional connections imposed by relations) along with its stored estimate, through mutual exchange, while exclusively processing local data.

We analyse the behaviour of the network in the case of unconstrained convergence, from initially random conditions (i.e. each unit is randomly initialized in the [0,1] interval) in Figure 3.5. Each unit updates its own local estimate such that it agrees to the units it is connected to. For units involved in more than one relation (i.e. unit m_2 and m_4) the update rules consider contributions from all related sources. We notice that convergence is

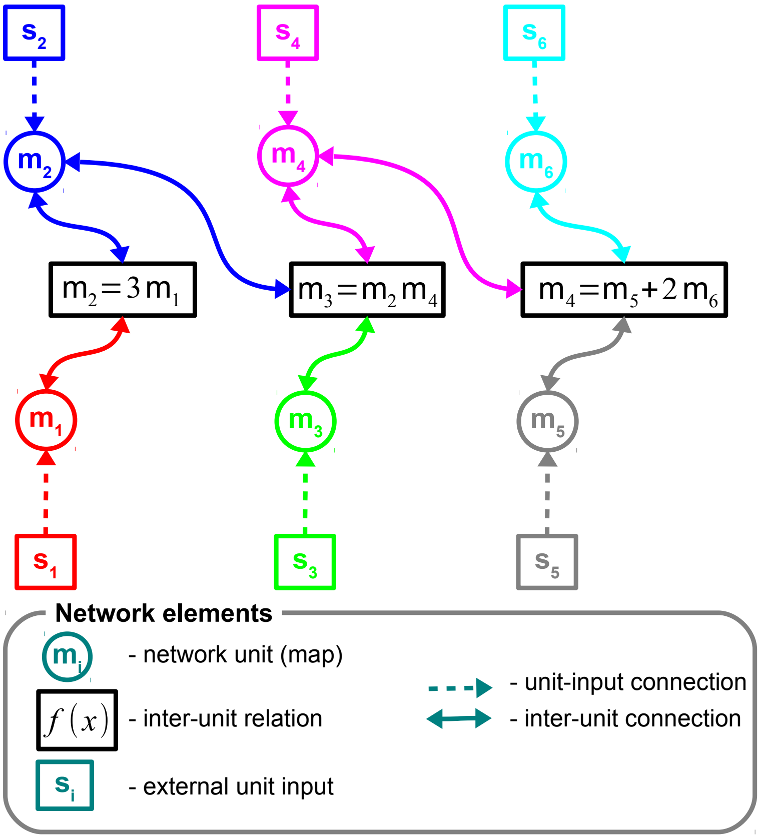


Fig. 3.4: Simple network model. Units (circles) encode a 1-dimensional quantity as a point estimate representation. Units are connected through functions (rectangles) which represent the constraints imposed on each unit estimate. External (sensory) data can be fed into the network as additional inputs to each unit (squares).

fast and each unit settles in a solution lying on the corresponding manifold in the relation space of m_1 and m_2 (Figure 3.5 lower-left panel). The convergence speed towards the constraint manifold is modulated by the relation, such that, in our scenario, m_1 is 0.3333 times slower than m_2.

Considering the same setup, now with external input, the network is constrained, such that it has less degrees of freedom (i.e. some solutions are imposed due to constraints). External input can be connected (i.e. clamped) to the network by enabling the connection between a map and an external source which continuously feeds the map with a constant value. To mark the moments at which the network is fed with external input we use appropriate labels: S_iON corresponding to the moment when the external input is clamped to the network, and S_iOFF corresponding to the moment when the external influence ceases. In all the test scenarios the external clamp is constantly feeding unit input. If we connect sensory inputs to units m_1 and m_2, network dynamics will try to reach consensus given that sensory inputs of respective units are clamped to a certain value, as shown in Figure 3.6. Figure 3.6 low-left panel shows that starting from initial conditions, the network autonomously evolves towards reaching a stable state. When the first sensory input is connected to the network (S_1ON) the network state changes such that the external

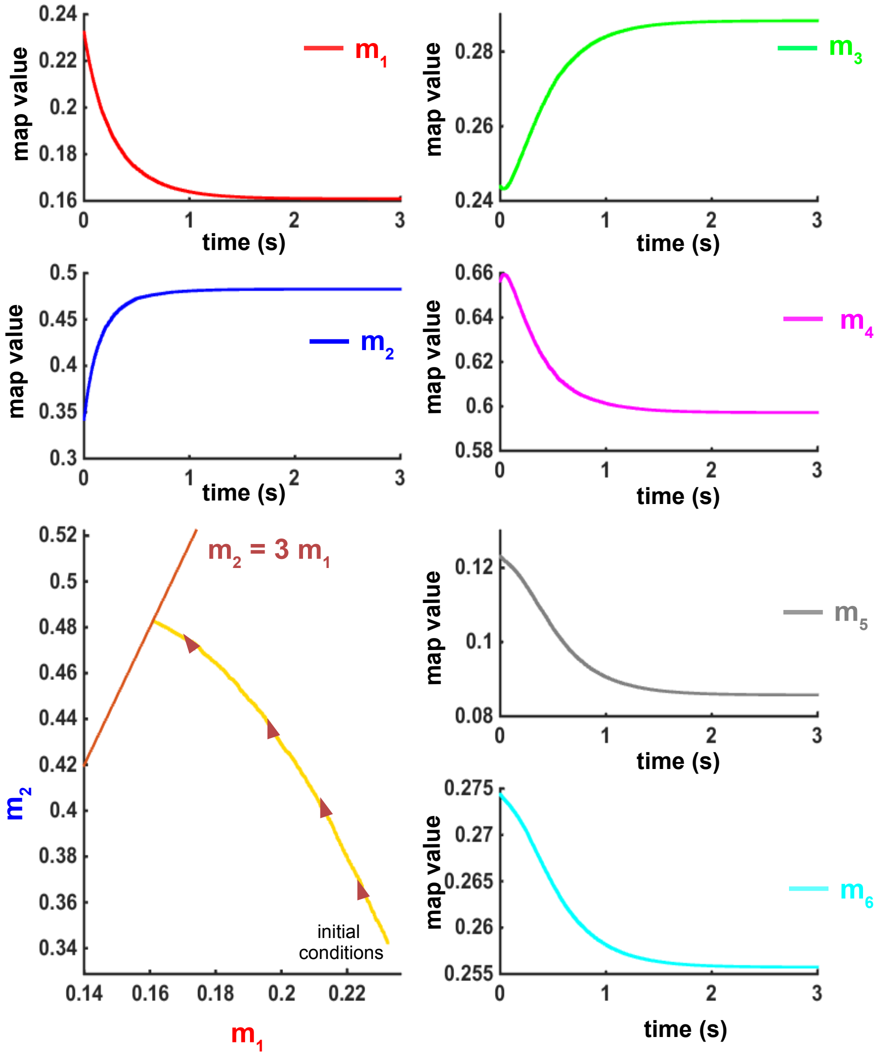


Fig. 3.5: Network analysis: first scenario. Starting from initial random conditions the network converges to a solution in which all embedded relations are fulfilled. The network is unconstrained by external input.

constraint is accommodated. The value stored by unit m_1 is now "pulled" towards the sensory input while still contributing to the overall network belief. Similarly, when m_2 starts to receive input from the sensor, it updates its state towards that value. The other maps in the network receive changing contributions from m_1 and m_2, such that constrained by the relations, they update their values accordingly. When sensory influence ends (i.e. S_1OFF, S_2OFF), the network evolves only under the influence of the internal relations, rapidly reaching consensus.

In this simple scenario, the sensors connected to m_1 and m_2 feed in 1.0 and -1.0 respectively, in the corresponding unit. We can see that once the sensors are connected, local values stored in the network units are shifting towards accommodating the external input while still obeying to the internal constraints (i.e. relations) in the network. Sensory contributions are propagated in the network through the relations such that the value con-

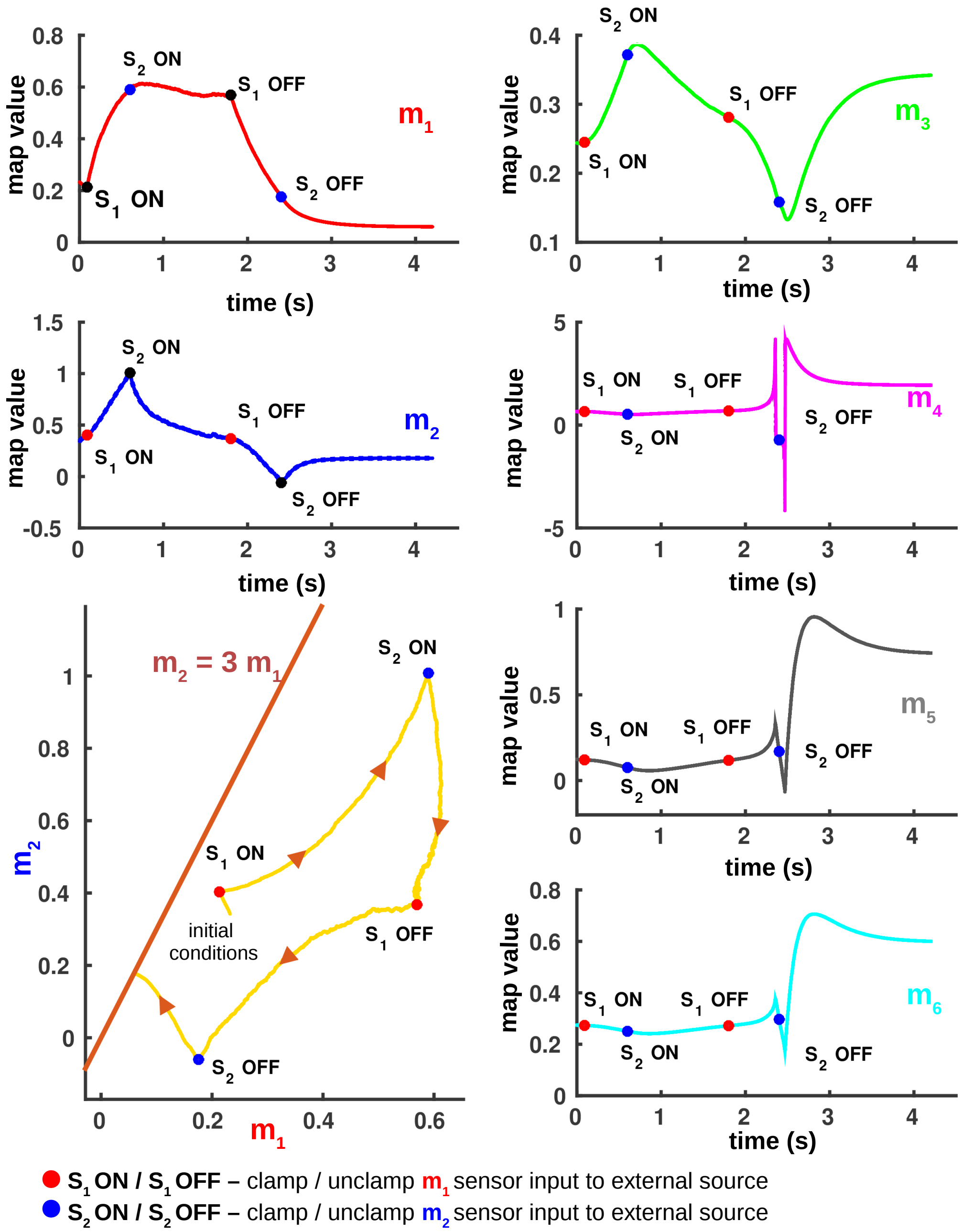


Fig. 3.6: Network analysis: second scenario. Starting from initial random conditions the network receives external input through m_1 and m_2. The maps are connected to external input at S_1ON (t=0.1s), S_2ON (t=0.6s) and disconnected at S_1OFF (t=1.7s), S_2OFF (t=2.4s) respectively. Due to the external input the network balances the contributions and accommodates new data updating its internal belief. New values are propagated through the network which updates its state towards fulfilling the relations.

tained in each unit map is taking steps towards minimising the mismatch with the relations in which it is involved. In this scenario the network is still underconstrained, such that units m_3, m_4, m_5 and m_6 are still free to settle in less restricted solutions, given that they have no external inputs. Moreover, we can see that the network reacts to exceptional cases (e.g. division-by-zero) in the case of the product relation between m_2, m_3, and m_4. Due to the fast internal dynamics, the system handles the exceptions and recovers, settling to a stable and correct value in the maps, without external constraints.

In order to analyse the mechanisms underlying the adaptation capabilities of the network, we consider the case in which we progressively connect external input to the network through each unit. By feeding different inputs to the sensors (at different rates and amplitudes) the network will do "its best" to combine the local belief of the network with all incoming sensory contributions and settle in a stable state. Moreover, the network is able to react to changes in the input space (i.e. removing an input), and using the available degree of freedom to settle in a stable solution as we can see in Figure 3.7. In order to fully constrain the network all units are now connected to their sensory inputs while still obeying the internal network constraints imposed by the relations. Figure 3.7 a (before t=1s) denotes how the network evolves rapidly to a stable state while no external input is connected, and starts to balance external contributions and internal belief towards consensus (between t=1s and t=3s), finally converging to a stable state (i.e. no more jitter in local estimates) once there are no more external constraints. Due to its internal dynamics the network will oscillate, such that each local unit estimate will jitter between sensory contribution and value imposed by the relations, Figure 3.7 b. The oscillations are determined by the network dynamics, as units are randomly updated, taking incremental steps towards minimising mismatch between local value and their input sources. Following a relaxation process the network continuously iterates, such that its internal belief is propagated across its units which locally update their state. This assumes a uniform random update process in which each unit takes steps towards minimising the mismatch to a certain incoming stream of information connected to it, be it another unit or sensor. The update process is assuming that in one network iteration all units are updated from all possible sources.

Notwithstanding its good performance in the aforementioned scenario (i.e. fast convergence to the underlying solutions of the relations), the network is also able to handle more complex, highly nonlinear relations. We explore further the capabilities of the proposed model with a network of same size but more complex and constrained functional dependencies between units. The network is depicted in Figure 3.8. In the following experiment we analyse the behaviour of the network for a temporary fully constrained context, in which it evolves freely from initial conditions, subsequently handles multiple synchronised sensory inputs, and then relaxes in a solution once external input is removed, Figure 3.9 a. In this context the network is still able to converge to a stable state fulfilling all the embedded relations. The oscillations present in each unit's evolution are given by the fact the values each unit is allowed to take values in the interval determined by the sensory input and the network estimate. Due to the mathematical constraints of the relations embedded in the network (i.e. inverse trigonometric function) we observe large jumps in the mismatch signals corresponding to zero crossings in maps values which assume illegal computation in the update rules (i.e. division by zero) or changes in sign, Figure 3.9 b, second row (E_{m_2,m_1}, E_{m_2,m_3,m_4}). Although these cases are rare in real-world scenarios, we investigated the capability of the network to react to spurious illegal values and the how fast it can handle the changes.

An interesting investigation we performed focuses on the network's capability to handle temporal relations between units. We extended the simple network architecture in Fig-

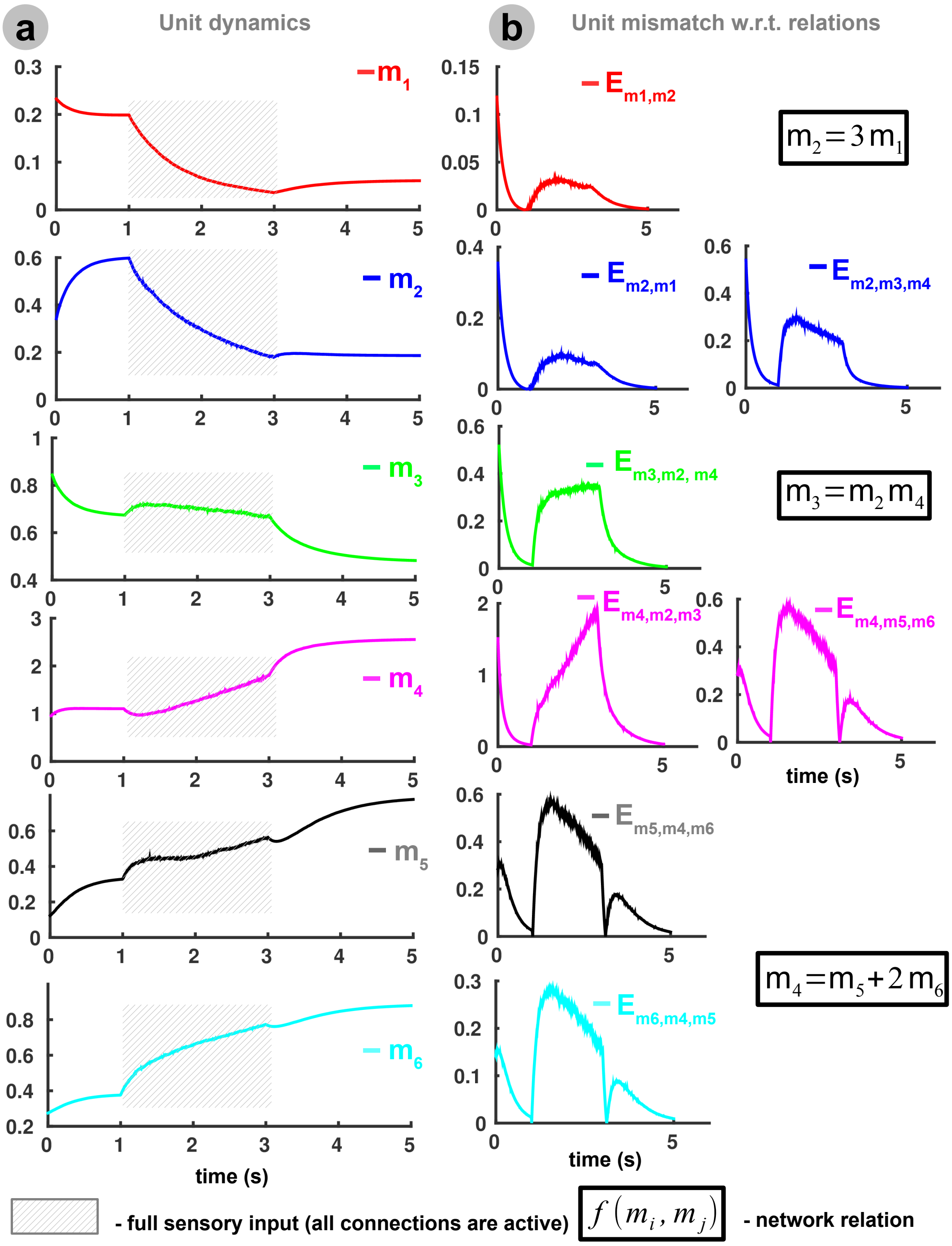


Fig. 3.7: Network analysis: third scenario. Starting from initial random conditions the network receives external input through all units starting t=1s up to t=3s. Due to the full external input the network balances the contributions and accommodates new data updating its internal belief in a fully constrained context. This is visible in the oscillations each unit's estimate has with respect to the relations it is involved in. a) Units dynamics for a fully constrained network; b) Units mismatches with respect the relations.

ure 3.4 by replacing the simple linear relations between units m_1 and $m2$ with temporal integration, such that m_2 is the temporally integrated version of m_1, Figure 3.10. We analysed the network's behaviour by connecting a switching sensory input signal to m_1, such that the signal was oscillating between -1 and 1 for around 3s. Once the sensory input

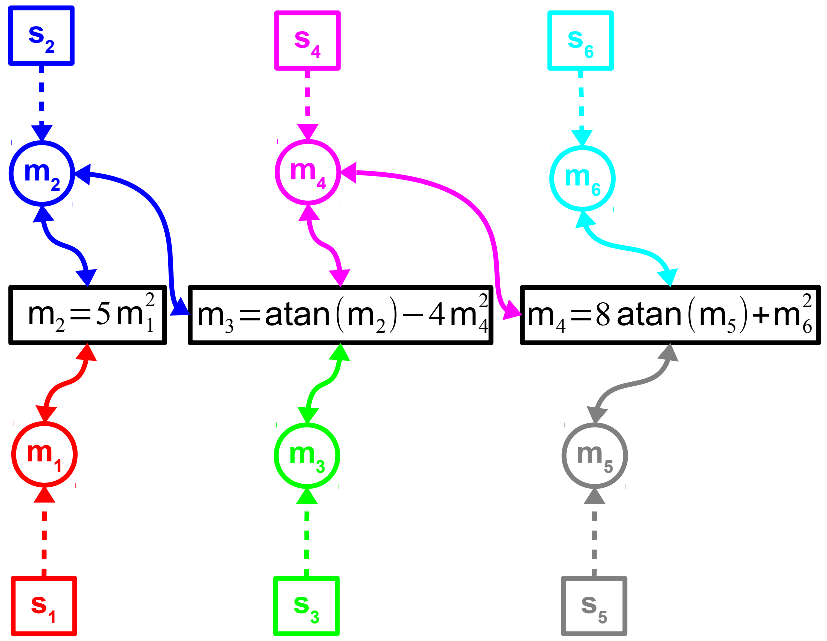


Fig. 3.8: Complex nonlinear network. Embedding highly nonlinear relations, with mathematically constrained functions.

was removed, the network evolved independently towards a stable state, Figure 3.10 a.

Given the single external input entering the network through m_1, unit m_2 accumulates (i.e. integrates) the value such that each unit in the network subsequently settles in new solutions of the embedded equations. This analysis is important as it provides insight in the speed the network can accommodate continuous changing sensory inputs, typical for real-world scenarios characterised by sensors sampled at different frequencies. In this case the network propagates incoming samples from the external source throughout the units which adapt their local estimate to be consistent with the external contribution.

In the last analysis scenario, we turn our attention to the adaptation mechanism (i.e. confidence factor) that each unit uses to weight the incoming contributions from other units or external sources. We designed a simplified version of the network introduced in Figure 3.4, so that given similar relaxation dynamics, we can analyse the adaptation mechanism on a per unit basis, and see how, through local processes, each unit is able to enhance consistent contributions and penalise inconsistent ones. The network structure we consider for this experiment is depicted in Figure 3.11. The structure used in this scenario is underconstrained, such that only two units receive external input, m_1 and m_3. Incoming sensory data is continuous, values changing in a given profile (e.g. ramp signal), and the input sequences do not overlap, such that the system can evolve towards a solution easily, as shown in Figure 3.12 a. The mismatch is rapidly compensated for due to the multiple degrees of freedom the network has in this scenario (i.e. only two external inputs connected), as depicted in Figure 3.12 b. As previously mentioned, the network benefits from an internal adaptation mechanism allowing it, at the unit level, to enhance contributions from external sources, when they are consistent with the global network

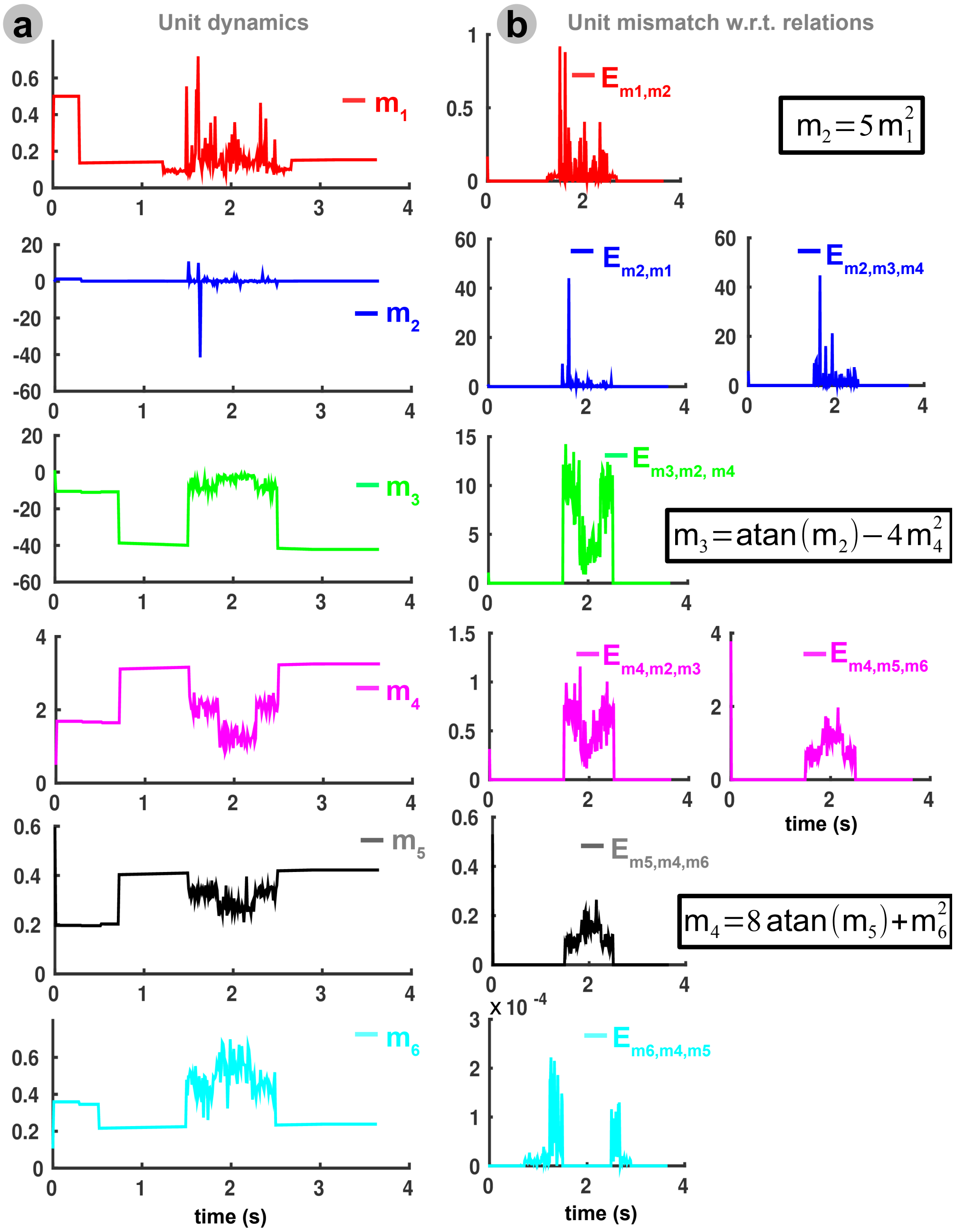


Fig. 3.9: Network analysis: fourth scenario. Starting from initial random conditions the network converges to a solution given the mathematically constrained functions in the relations. When all sensory connections are enabled ($t_{ON} = 1.5$s to $t_{OFF} = 2.5$s) the network oscillates for $t > 2.5$s due to network random update dynamics for the fully constrained space of values its units can take. Once freely evolving driven by internal dynamics, the network settles in a stable state. a)Units' dynamics for the complex network in constrained scenario; b)Units' mismatches with respect the complex relations.

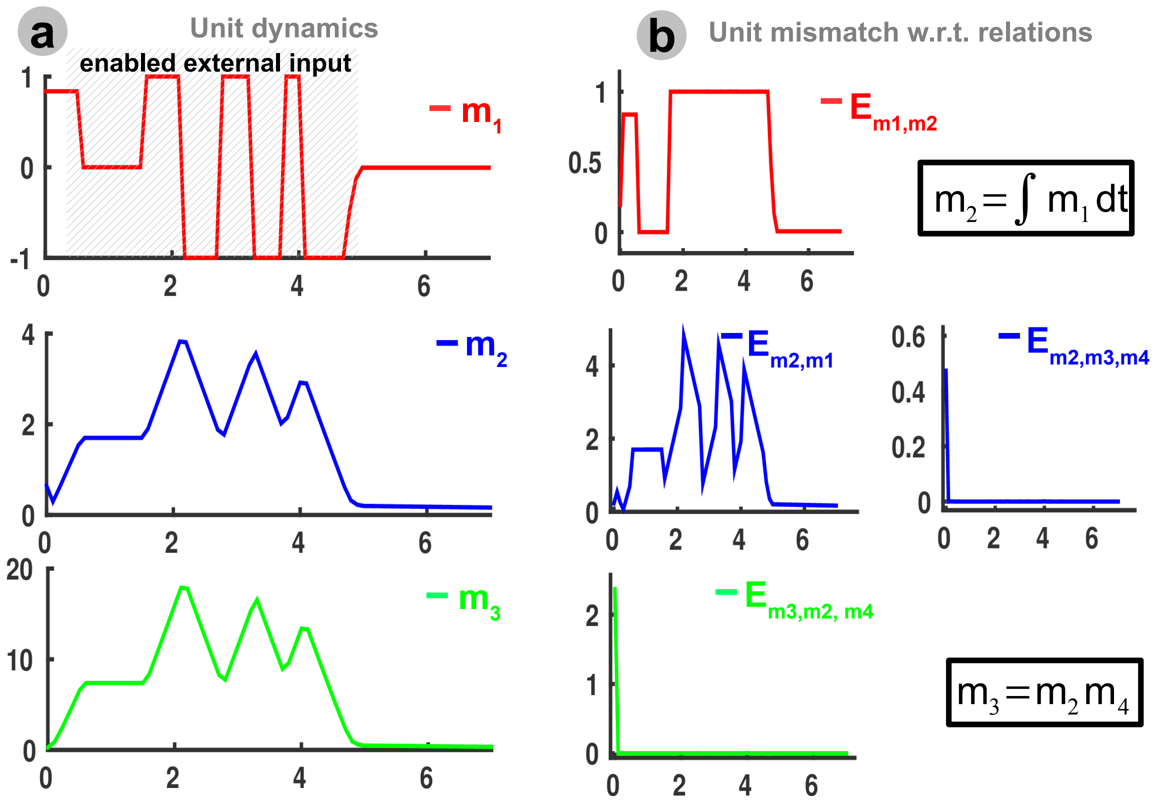


Fig. 3.10: Network analysis: fifth scenario. Network relations are a mixture of linear, nonlinear, and temporal relations. a) Units' dynamics for given sensory input and temporal accumulation in m_2; b) Units' mismatches with respect the different relations.

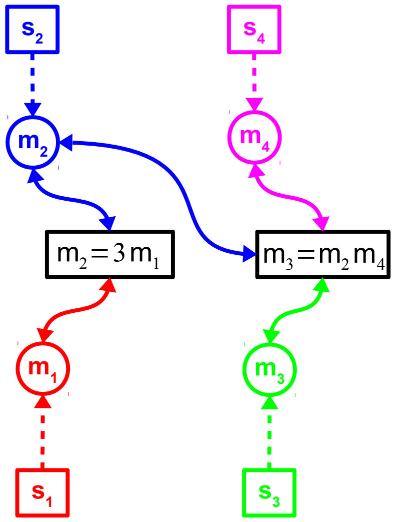


Fig. 3.11: Network model for analysing adaptation capabilities given external input.

belief, and penalise inconsistent contributions. This mechanism allows the network to detect and compensate for faulty input data and still keep stable and correct estimates in the network. Confidence factors are associated with each incoming source of information of a unit. Balancing contributions and locally computed mismatches, the network infers

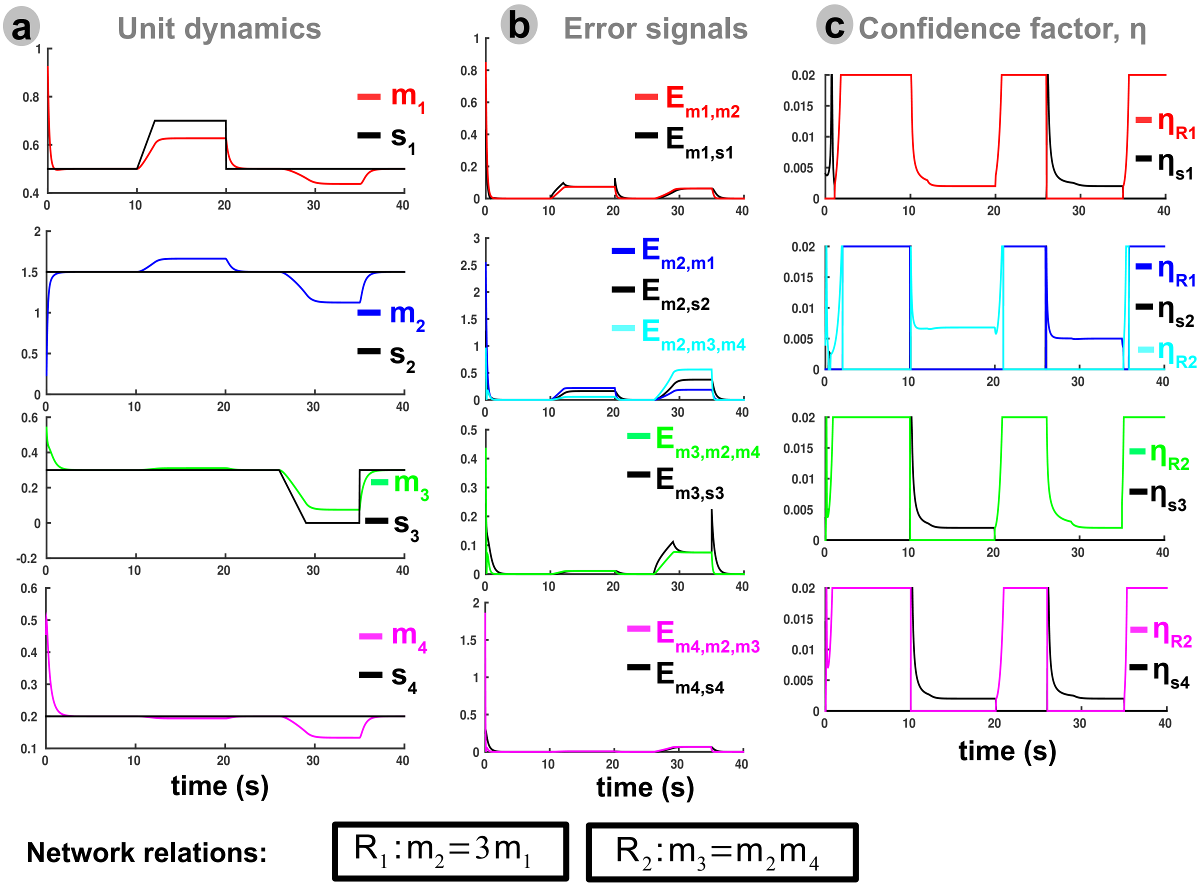


Fig. 3.12: Network analysis: sixth scenario. Simplified network structure for analysing adaptation capabilities given external input. a) Units' dynamics; b) Error signals computed locally by each unit with respect to incoming sources of information; c) Confidence factor analysis on a per unit input source basis (i.e. relative to sensor or relation).

weights for each individual source, as shown in Figure 3.12 c, such that it rapidly converges to a consistent global estimate.

3.3 Summary

We conclude our analysis with some concepts and features of our model which will be exploited in the upcoming chapters where we will focus on real-world instantiations of our framework. Figure 3.13 offers a synthetic view on how we derived our model, emphasizing the most important representation and processing mechanisms. The first aspect is extensibility. As we observed in our analysis we believe that the network can take arbitrarily large sizes due to its distributed structure, can implement arbitrarily complex relations due to simple "atomic" implemented operations, and can have arbitrarily defined connectivity patterns reflecting its dynamics. Given the generality of the update rules, the network can be flexibly extended, as local dynamics ensure global consistency between implemented constraints (relations) in the network.

A second important aspect for real-time implementations, especially when facing real-world sensory data and sensor models, is fault tolerance. We previously analysed the intrinsic confidence factor adaptation as a means to detect and weight incoming contribu-

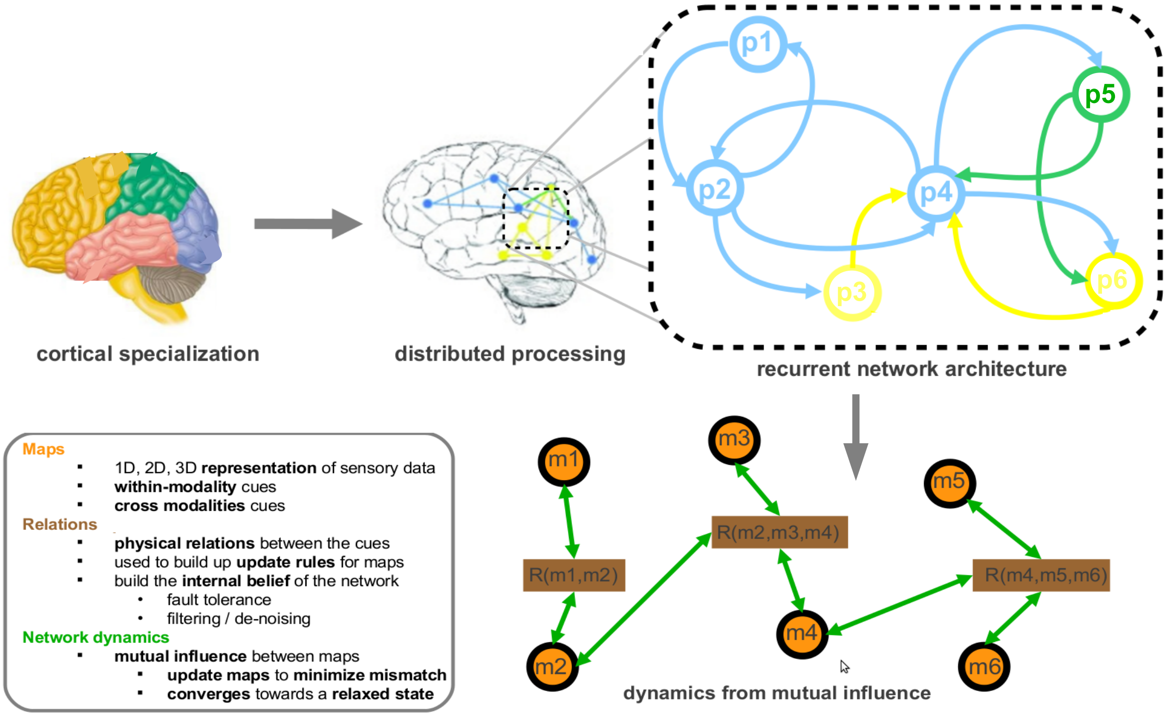

cortical specialization
distributed processing
p1
p2
p3
p4
p5
p6
recurrent network architecture
Maps
1D, 2D, 3D representation of sensory data
within-modality cues
cross modalities cues
Relations
physical relations between the cues
used to build up update rules for maps
build the internal belief of the network
fault tolerance
filtering / de-noising
Network dynamics
mutual influence between maps
update maps to minimize mismatch
converges towards a relaxed state
m1
m3
m5
R(m1,m2)
R(m2,m3,m4)
R(m4,m5,m6)
m2
m4
m6
dynamics from mutual influence

Fig. 3.13: Deriving the model: From neural substrate to multisensory representation and computation in real-world systems.

tions at the unit level. The local adaptation generates global network consistency.

In the implemented networks, units relax together. Therefore, if a sensor provides noisy data, the collective computation will analyse and weight its input in their context and will bring outlying values towards values that are consistent with the rest of the data in the network. In order to avoid problems of local minima, we also explored extra noise injections into the network using simulated annealing, to achieve better global results and faster convergence towards consensus. The noise injections assumes small amplitude increments at local level to help the network state leave the local optimum. Sensor failures are recognised as such by the equality relation that connects the sensor to its corresponding part of the network. This relation pulls the network towards the values reported by that sensor, unless the reported sensor values are so far off as to be effectively inconsistent with the rest of the network for an extended period of time (as compared with the amount of time typically required for the network to converge to a consistent state). In this situation, the confidence factor allows the sensor input to sit at its inconsistent value without further perturbing the rest of the network. If the sensor were to come back online, then the confidence adaptation mechanism automatically resumes usage of the sensor input due to the second opinion coming from the rest of the network. The mechanism is able to determine in a natural way whether the sensor is operating correctly or not.

In the upcoming chapter we will focus on specific instantiations of the developed framework in a real-world scenario, namely egomotion estimation for mobile robots. Because processing in the model is inherently parallel and asynchronous, we will also provide a thorough analysis on how the model can be distributed and executed on standard PCs taking advantage of software parallelism, or on massively parallel neuromorphic hardware architectures.

4 Instantiating the multisensory fusion model

As shown in Chapter 3, our work probes high-level processing and organization principles of multisensory fusion known to take place in the brain, and targets instantiations in robotic systems. Inspired by psychophysical and computational neuroscience models for multisensory fusion, we identified the use of an adaptive neural substrate as a support for flexible operations in adaptive sensory integration. Due to its anatomical organization, the brain allows processing of different incoming signals from sensory modalities in anatomically separate regions of the cortex. Moreover, this distributed scheme globally resembles processing at cortical level where multisensory events elicit responses from different sensors and are subsequently integrated into a unified and coherent perceptual representation of those events. Integrating multisensory events relies not only on anatomical convergence from sensory-specific cortices to multisensory brain areas, but also on reciprocal influences between cortical regions that are traditionally considered as sensory specific. We can then assume that multisensory processing is a framework designed to account for a wide variety of integrative processes that the brain constantly performs. This flexible framework yields some general principles which can be easily transferred to technical systems as an alternative to existing approaches.

The type of information processing that we propose allows seamless multisensory fusion capabilities. As we saw in Chapter 2, there is no generic framework to describe sensory integration processes, especially when supporting different sensory modalities. Our approach provides the means to develop a general solution to this problem, since each unit of the model is able to represent a different sensory modality, and extended networks can embed various types of sensory information. The capability to combine different modalities comes as a side-effect in the processing paradigm we propose, due to the mild influence sensory contributions have on different representations in the network.

Brains and computers work in very different ways, and they are good at different things. For some tasks, such as performing intense numerical calculations, memorising large lists, or precisely following predefined instructions, computers overcome human capabilities. But in other areas, such as environment interpretation, interaction, and decision making in the face of uncertainty based on whatever information is available, brains are many orders of magnitude better than computers. Humans are typically much more robust to noise in the sensory data, inhomogeneities of computational substrate, or environmental changes than current engineered systems. Using a similar processing scheme like the one employed by the brain, the proposed framework provides a solution to multisensory fusion showing that the proposed distributed processing scheme is more robust to noise, sensory failures, and uncertainty.

Finally, the proposed information processing scheme can be extended to represent and process various types of information content. As the core dynamics are based on the physics of the sensors (eventually spatio-temporal relations among physical quantities), the designer can interconnect multiple networks into a single large network, allowing it to internally represent many aspects of the environment or the estimated feature, and thus being able to incorporate many more types of sensory inputs, giving a sensible solution to the problem of multisensory fusion.

Using a distributed processing scheme based on localized intelligence that ensures asynchronous information exchange and adaptation based on external real-world sensory stimuli, the framework ensures the design of fast, robust and scalable computational architectures appropriate for real-time real-world technical applications.

4.1 Multisensory fusion network for mobile robot egomotion estimation

An essential component in motor planning and navigation, for both real and artificial organisms, is egomotion estimation. Egomotion or self-motion refers to the combined rotational and translational displacement of a perceiver with respect to the environment. During motion, organisms build their spatial knowledge and behaviours by continuously refining their internal belief about the environment and own state [Arleo et al., 2007, Heed et al., 2012, Mitchel, 2010]. Our approach is motivated by three main aspects consistent with recent results in spatial processing for navigation and perception [Mast et al., 2007], which are described in the following paragraphs.

The first aspect addresses the importance of maintaining a precise position of the self. Building an internal representation of the environment and own state implies the coherent alignment of the acquired sensory cues. As sensory cues are conveyed from both dynamic egomotion related signals such as odometry and inertial signals, and static external environmental signals, such as visual or auditory, the precise position of the self is responsible to link and keep the representation coherent. In this context a coherent representation provides the ability to recognise and define ”action possibilities” from all available sensory cues (e.g. distance to objects). Subsequently, egomotion defines the space of possible actions that impacts behaviour [Heed et al., 2012, Mitchel, 2010, Sheets-Johnstone, 2010].

A second aspect refers to the capability of a real or artificial organism to understand space itself from its own state (in space). Egomotion estimation contributes to the understanding of high-level features of the environment, like structure and layout, such that the organism can direct actions and control its movement. Typically, with respect to position, the primary question is related to distances to key objects in the environment. In order to infer correct distances, the organism must traverse the environment and distinguish between its dynamic and static features as they lead to different consequences [Warren, 1990].

The third aspect points directly to the solution offered by our model, namely how can robust egomotion perception be obtained given the complex multisensory environment. In order to handle environmental variability and complexity, continuous and simultaneous incoming sensory data streams from different sensors must be combined into a robust representation. However, sensory cues are usually complementary and redundant and is not clear how they describe the spatio-temporal properties of the environment. To disambiguate the complex scenario the global representation should combine all cues in an informative and plausible way. This sensory combination process is not trivial, as current implementations show [Ferreira et al., 2014] . The primary objective is aligning reference systems of the different, congruent, and redundant sensory cues. After alignment, depending on inferred spatio-temporal correlations, interference and conflicts between the cues need to be minimised [Heed et al., 2012, Mitchel, 2010, Lackner et al., 2004]. Finally, the multisensory fusion system should not propagate biases or errors in the final (fused) estimate, but compensate for them.

This section is organised as follows. Starting from the general neurally inspired processing model described in Section 4.1.1, we present the architecture and the specific instantiation for the mobile robot egomotion estimation in Section 4.1.2. Section 4.1.3 provides the analysis and evaluation of our model and a comparison with state-of-the-art methods, whereas Section 4.1.4 provides a thorough discussion of the obtained experimental results as well as future extensions.

4.1.1 A cortically inspired network for egomotion estimation

The model uses a distributed network in which independent neural computing nodes obtain and represent sensory information, while processing and exchanging exclusively local data, to infer an estimate of robot orientation. The scenario in [Axenie et al., 2013] is now extended to full egomotion estimation. As previously shown, our generic processing framework is inspired by the neural processing paradigm introduced in Chapter 3, where cortical areas involved in sensory processing assume rapid resolution of a large number of mutually imposed constraints (i.e. coherence / incoherence relations), leading to a globally coherent estimate of the percept. Following similar high-level cortical organization and interaction principles with our model the work in [Ferreira et al., 2013, Ferreira et al., 2011] introduces and evaluates the capabilities of a neuromimetic Bayesian framework for multimodal sensory fusion for motion estimation and 3D structure extraction, motivating the advantages of the paradigm shift towards biological inspiration.

General processing model

The proposed model is a network of processing units whose connectivity is provided by relations defined between the units (e.g. given by physics of the sensors or inter-sensory interactions). The relations between the units can be explicitly encoded in the network or obtained as a result of a learning process (as we will show in Chapter 5). A more detailed description of this computational paradigm is given in Chapter 3. Even though in the current stage relations are embedded in the network at design time, the dynamics and integration capabilities of the network are the same for both hand-crafted and learned relations. Hence, our model separates relaxation dynamics (convergence towards a solution) from learning (connectivity set-up). A similar principle was successfully used for fast visual interpretation in [Cook, Gugelmann et al., 2010]. The system proposed in this work interpreted input from a neuromorphic vision sensor by means of recurrently interconnected areas, each of which encodes a different aspect of the visual interpretation, such as light intensity, optic flow and camera calibration. This network of interacting maps is able to maintain its interpretation of the visual scene in real time. A similar approach has been also used in [Ferreira et al., 2013, Ferreira et al., 2011] modelling midbrain and cortical sensory integration sites, and employing iterative Bayesian programming to refine spatial maps (i.e. probability distributions). Following the computational map based representation described in Chapter 3, a sample network to implement sensory averaging is introduced in Figure 4.1. This network is used extensively in our model as a core mechanism to quantify the relative mismatch (offset) between the quantities inferred in the network. The network computes the average activity of all other connected units, m_{j}, in the network, stores it in *net*, and isolates the contribution of a certain unit, m_{i}, to compute its relative offset, m_{ioff}. The offset is inferred in the network, in a separate unit, and each main unit has an associated offset node following the next generic update rules:

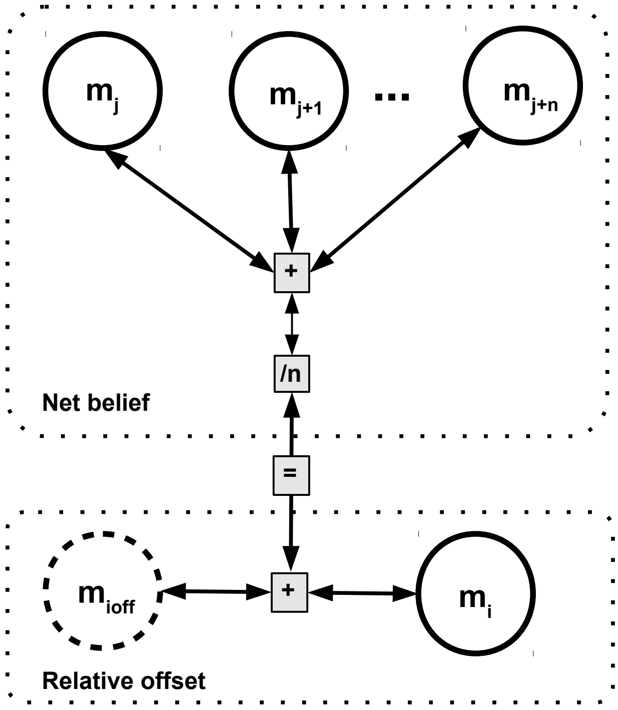


Fig. 4.1: Network for generic algebraic relations. Implementing summation, division, or difference using the proposed model is straightforward. Particular implementation of offset computation network.

$$net(t) = \frac{\sum_{k=0}^{n} m_{j+k}(t)}{n+1} \tag{4.1}$$

$$\Delta m_{ioff}(t) = -\eta_{m_i,net}(t)\frac{\partial E_{m_{ioff},net}(t)}{\partial m_{ioff}(t)} \tag{4.2}$$

$$E_{m_{ioff},net}(t) = (m_{ioff}(t) - (net(t) - m_i(t)))^2 \tag{4.3}$$

The offset nodes quantify the relative mismatch among units depending on the type of unit, and have an impact on different time-scales (e.g. faster for integrating sensors, slower for absolute sensors). This type of information can be used to define reliability regions in the space such that the system judiciously modulates the weighting of each contributing cue. Defined as prior information, this quantity can further enhance local estimates of angle and position. During operation, the network brings all quantities in agreement by satisfying all relations. The amplitude of the update step that each unit takes towards the correct value is computed on-line, on a per map basis, and modulated by the confidence factor, η. This factor accounts as a measure of reliability of a certain source of incoming information into a unit. Using this mechanism, the network is able to penalise strongly conflicting sources of information (a mechanism that improves fault tolerance) and enhance the contribution of consistent sources. Each contribution to a unit, m_i from another unit, for example m_j, in a network with n units, is modulated by the confidence factor, adapted using

$$\Delta\eta_{m_i,m_j}(t) = \eta_{m_i,m_j}(0)\frac{\bar{E}_{m_i,m_k}}{(n-1)E_{m_i,m_j}(t)}, \bar{E}_{m_i,m_k} = \frac{\sum\limits_{k=1,k\neq j}^{n} E_{m_i,m_k}(t)}{(n-1)} \tag{4.4}$$

Assuming that all n units in a network should contain the same value, Equation 4.4, computes the confidence factor, η, for map m_i when receiving influence from map m_j, by comparing the expected mismatch (i.e. average error of m_i with respect to the network) $\bar{E}_{m_i,m_k}$, with the error between m_i and mj, $E_{m_i,m_j}(t)$.

Although many possible relations can be implemented using only the basic algebraic operations, in order to handle sensory data we also need relations which encode temporal dependencies. One example is temporal integration and temporal differentiation. As the network should combine contributions from different sources, they should be aligned and represent the same type of values (i.e. rate of change or absolute values). Our model implements temporal integration, locally, using two units, one which maintains the (persistent) absolute value, and one which provides the rate update. This persistence mechanism is necessary to keep a coherent absolute value in the presence of inputs from other units. A sample integration network used in our model is depicted in Figure 4.2. The preprocessing unit, m^{pp}, which maintains a persistent value in unit m_i, obeys the same update rules with other units, the only difference is just its limited connectivity. The update rules for m_i in the canonical integration network depicted in Figure 4.2 are given by

$$\Delta m_i(t) = -\eta_{m_i,m^{pp}}(t)\frac{\partial E_{m_i,m^{pp}}(t)}{\partial m_i(t)} \tag{4.5}$$

$$E_{m_i,m^{pp}}(t) = (m_i(t) - \int_0^t m^{pp}(t)\mathrm{d}t)^2 \tag{4.6}$$

Following the generic update rule for units in the network, Equation 4.5 computes the new value of m_i by using the mismatch from the relation with m^{pp} (Equation 4.6). One can also use an error signal given by $E_{m_i,m^{pp}}(t) = m_i(t) - \int_0^t m^{pp}(t)\mathrm{d}t$ with differences in the magnitude of the update. The value in m_i converges to the accumulated absolute value of the sensory input s_i. As m_i is connected to m_j, there is an influence towards m_i from m_j unit, and the amplitude of the update, given in Equation 4.7 is equal to the mismatch

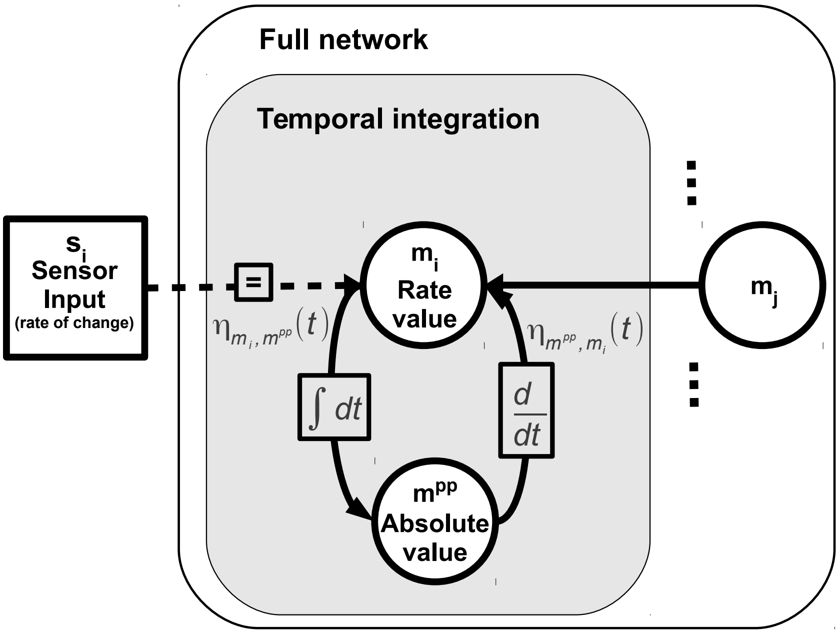


Fig. 4.2: Canonical network for temporal integration. The local integration network maintains a persistent accumulated value of sensor data in m_i. In the absence of sensory data, the preprocessing unit, m^{pp} ensures the proper rate update, avoiding large updates induced by other connected units, m_j.

between the value in m_i and the value in m_j, as shown in Equation 4.8.

$$\Delta m_i(t) = -\eta_{m_i,m_j}(t)\frac{\partial E_{m_i,m_j}(t)}{\partial m_i(t)} \tag{4.7}$$

$$E_{m_i,m_j}(t) = (m_i(t) - m_j(t))^2 \tag{4.8}$$

The differentiation map, m^{pp}, is updated using a single set of update rules (Equations 4.9, 4.10) due to the unique connection to m_i. The m^{pp} map converges to the rate of change of s_i, given by $\frac{\mathrm{d}}{\mathrm{d}t}m_i(t)$, and provides a persistent contribution to m_i.

$$\Delta m^{pp}(t) = -\eta_{m^{pp},m_i}(t)\frac{\partial E_{m^{pp},m_i}(t)}{\partial m^{pp}(t)} \tag{4.9}$$

$$E_{m^{pp},m_i}(t) = (m^{pp}(t) - \frac{\mathrm{d}}{\mathrm{d}t}m_i(t))^2 \tag{4.10}$$

As previously mentioned, one main feature of the neural substrate for multisensory fusion is robustness. Our model exhibits robustness at the unit level through the confidence factor. The confidence factor provides a fault tolerance mechanism which detects errors in incoming noisy raw sensory streams. By adapting the confidence factor according to the measured mismatch (from the expected value and the current value) the network penalises conflicting sources and enhances congruent sources.

Using these canonical circuits as building blocks we instantiate our model for egomotion estimation. Taking advantage of relatively simple processing stages at unit level the network can rapidly relax in a solution, fulfilling all relations between the fused quantities.

4.1.2 Mobile robot egomotion estimation

We instantiated our framework in a real environment using an omnidirectional mobile robot depicted in Figure 4.3. In the basic scenario the robot moves in an uncluttered environment while an overhead camera tracking system keeps track of its position and orientation. The robot is equipped with an inertial measurement unit, consisting of a 3-axis gyroscope and a 3-axis magnetometer which acts as vestibular input; wheel encoders acting as proprioceptive input; motor driver providing an efferent copy of the PWM signal; and a camera for visual input. Raw sensor data is fed to the network which updates its internal belief and infers an estimate of robot's position and orientation. The main architecture of our network for egomotion estimation is depicted in Figure 4.4. There is no explicit input or output in/from the network and sensor data just mildly influences the activity in the network. Based on the embedded relations, the network is able, in the absence of one or more sensors, to infer the missing quantities.

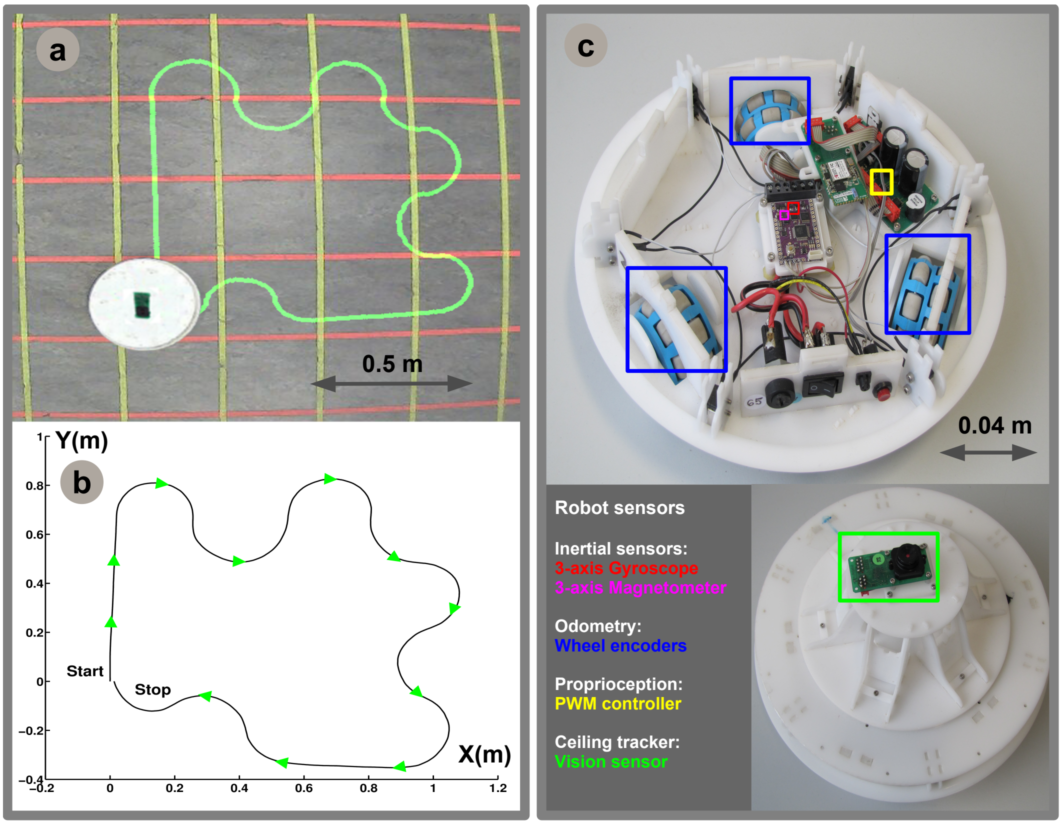
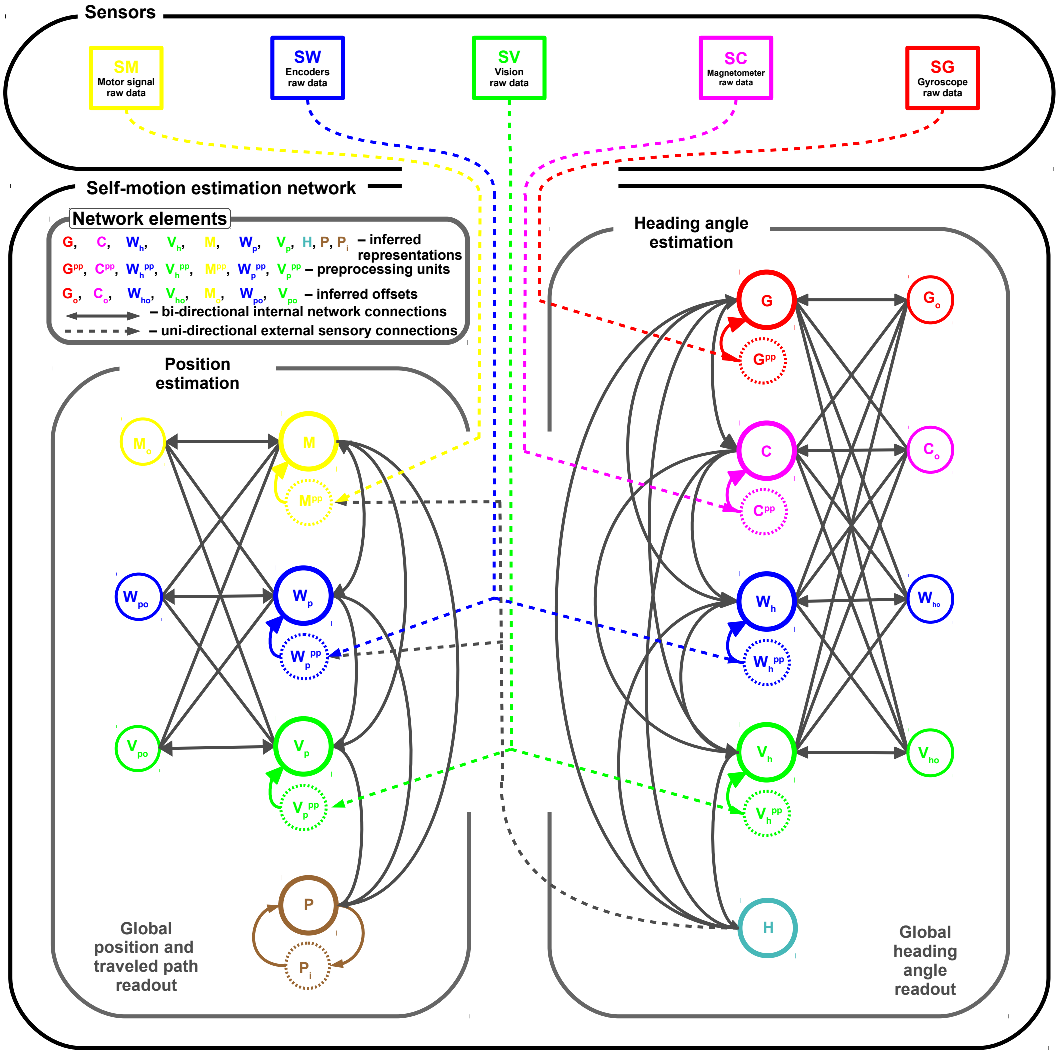


Fig. 4.3: Robot architecture and experimental setup: mobile robot and test trajectory. a) Overhead tracker trajectory; b) Robot reference trajectory; c) Mobile robot sensors.

In order to properly visualise the network connectivity we split the embedded functionality with respect to the task, namely position and orientation estimation. Most connections within the network are bidirectional and elicit mutual influence between the units linked by relations. The only unidirectional connections are the ones coming from the sensors, as the network cannot influence sensory readings. In order to quantify the performance of our model, we also added two readout units which provide an average of the estimated quantities. These units obey the same update rules and dynamics as all other inferred quantities in the network.



Sensors
SM Motor signal raw data
SW Encoders raw data
SV Vision raw data
SC Magnetometer raw data
SG Gyroscope raw data
Self-motion estimation network
Network elements
G, C, W_h, V_h, M, W_p, V_p, H, P, P_i – inferred representations
G^{pp}, C^{pp}, W_h^{pp}, V_h^{pp}, M^{pp}, W_p^{pp}, V_p^{pp} – preprocessing units
G_o, C_o, W_{ho}, V_{ho}, M_o, W_{po}, V_{po} – inferred offsets
– bi-directional internal network connections
– uni-directional external sensory connections
Position estimation
M_o
M
M^{pp}
W_{po}
W_p
W_p^{pp}
V_{po}
V_p
V_p^{pp}
P
P_i
Global position and traveled path readout
Heading angle estimation
G
G^{pp}
G_o
C
C^{pp}
C_o
W_h
W_h^{pp}
W_{ho}
V_h
V_h^{pp}
V_{ho}
H
Global heading angle readout

Fig. 4.4: Network architecture for egomotion estimation. Distributed fully-connected network composed of interconnected sub-networks for heading and position estimation. All connections are bidirectional except those coming from the sensors.

Experimental setup

The egomotion network consists of two main interacting components, one for orientation estimation, and the other for position estimation. Functionally the sensory input fed to the network is clamped to the corresponding maps encoding a certain sensory quantity. As mentioned, the network dynamics pulls all quantities towards agreement given the intersensory relations and external input. Based upon the basic gradient mechanism described in Chapter 3, each map takes steps towards minimising the mismatch with the other maps it is connected to. Gradient descent steps are proportional with the relative mismatch of a map with respect to all its input sources. Practically, each input sample coming from the sensors is presented to the network, which asynchronously iterates all the quantities towards consensus. Due to the fact that the network adapts for each incoming sample, updating only simple algebraic relations, it is suitable for real-time implementations. In the current instantiation, sensory data was sampled at 25 Hz, and each sample was presented to the network for 100 iterations, the time the network needed to converge (i.e. no more changes in the maps' values).

Heading estimation network

We first introduce the heading estimation network, depicted in Figure 4.5. This network is comprised of:

- 4 main units (G, C, $\mathrm{W_h}$, $\mathrm{V_h}$) encoding representations of modalities' heading angle estimates (gyroscope, compass (magnetometer), wheels encoders, and camera) functionally similar to neural visual-somatosensory-vestibular integration models in [Mergner et al., 1990, Wertheim, 1990, DeAngelis et al., 2012];

- 4 preprocessing (pp) units ($\mathrm{G^{pp}}$, $\mathrm{C^{pp}}$, $\mathrm{W_h}^{\mathrm{pp}}$, $\mathrm{V_h}^{\mathrm{pp}}$) which transform raw sensor data performing offset compensation (for vision and magnetometer) or temporal integration (for gyroscope and encoders odometry);

- 4 heading angle offset (ho) units ($\mathrm{G_o}$, $\mathrm{C_o}$, $\mathrm{W_{ho}}$, $\mathrm{V_{ho}}$) which quantify sensors bias or drift and act upon the absolute estimate on different time-scales (e.g. faster for integrating sensors and slower for absolute sensors);

- 1 global readout unit, H, which provides an average of the inferred quantities (accumulates and propagates a snapshot of instantaneous angle estimates for updating integration processes in the position network).

After internally preprocessing raw sensory data, each main unit stores an estimate of absolute heading angle and tries to be consistent with the values in the other units. The main units in the network internally represent an internal model the system has about the sensed quantity (i.e. heading angle) and how to extract it from the transducer (i.e. angle from angular velocity, angle from wheel encoders). Moreover, sensory observations are acquired by the preprocessing units. The global process governing network's dynamics is to integrate predictions (provided by the internal model the designer encodes in the network) and sensory observations, to infer a belief about the perceived motion component. As the network infers multiple estimates of heading angle from different sources, it ensures that all yield the same value. In fact the fusion process assumes that all complementary modalities are combined, yielding a more precise estimate than individual estimates. As each modality provides its own estimate of absolute angle, the network combines all contributions judiciously. Mutual influence between units is modulated by the confidence factor. Each

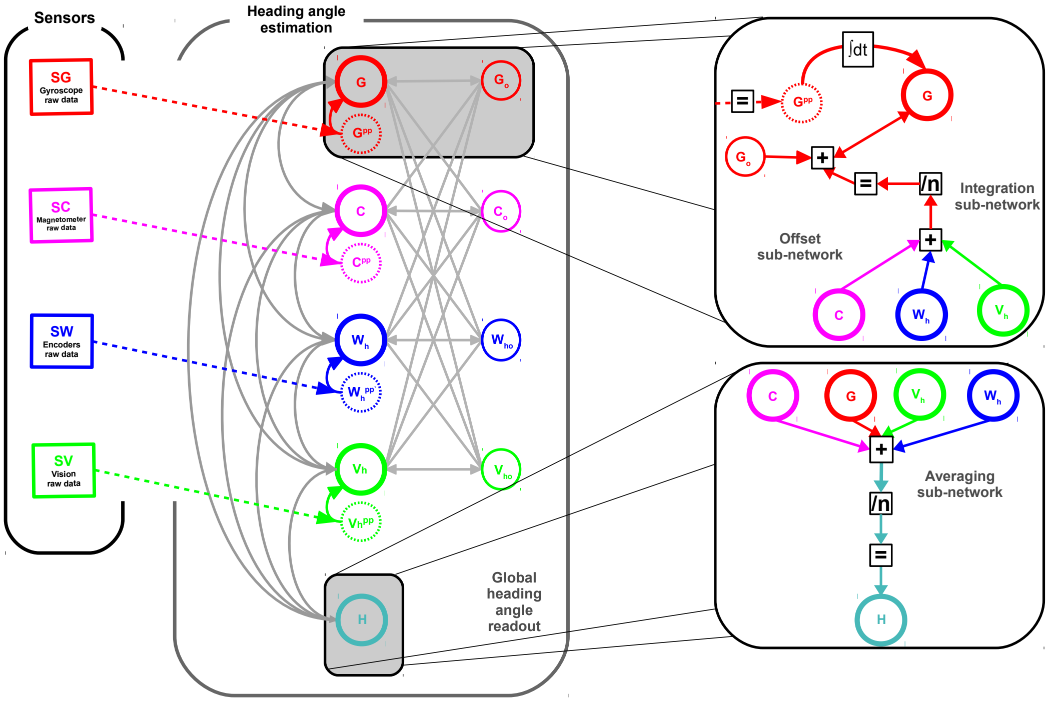


Fig. 4.5: Heading estimation network. Sensory data from gyroscope (SG), compass (SC), wheels encoders (SW) and camera (SV), flows in the network mildly influencing its activity. Local networks implement preprocessing, using G^{pp}, C^{pp}, $W_h{}^{pp}$, $V_h{}^{pp}$ units, while all-to-all connections between main units determine interactions yielding a fused estimate.

interaction pathway of a unit has an associated confidence factor which adapts according to the level of trustworthiness of a source of information to which the unit is connected. Hence, the network benefits of a mechanism to detect and compensate for faults and abrupt changes in sensor data. The distributed local representations inferred in the network nodes are integrated in the readout node, which provides an average of network inferred quantities, and can be used to quantify performance. The readout node contribution to each of the main units is proportional to the contribution of that specific unit in the global estimate, modulated by the confidence factor. The processing steps behind the global readout node are depicted in the lower right panel of Figure 4.5.

Position estimation network

The other component of the egomotion network, dedicated to position estimation, computes a global estimate of robot position in the 2D plane as well as the travelled path. Using an integration scheme functionally similar to mechanisms presented in [Sheets-Johnstone, 2010, Wiener et al., 2011], the network, shown in Figure 4.6, is composed of:

- 3 main units encoding representations of different modalities' 2D position (p) estimates (V_p - camera estimate, W_p - encoders estimate, M - position from efferent copy of motor command estimate);

- 3 position offset (po) units (M_o, W_{po}, V_{po}) encoding the mutual mismatch between the inferred quantities;
- 3 sensory preprocessing (pp) units (M^{pp}, ${W_p}^{pp}$, ${V_p}^{pp}$) which are tightly coupled with the main units and perform temporal integration or simple transformations from robot reference frame to world reference frame;
- 1 global readout node, P, providing an average of the network belief about robot's position (x, y) (accumulates a snapshot of instantaneous position estimates);
- 1 global readout node, P_i, providing the travelled path, by accumulating changes in the P unit.

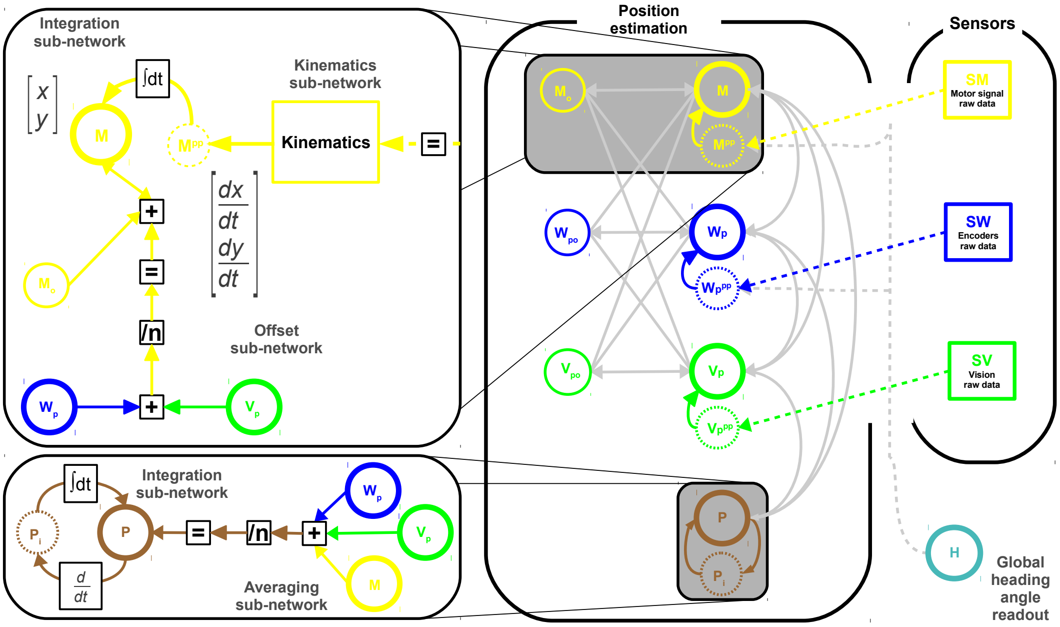


Fig. 4.6: Position and travelled distance estimation network. Sensors provide raw data to the network. Local circuits preprocess raw data by using algebraic or temporal (differentiation) relations to maintain a position estimate in each main unit.

The position estimation network in Figure 4.6 infers a 2-dimensional estimate of position, given as (x, y) coordinates, and a 1-dimensional estimate of travelled path, using sensor data, the heading network estimate, and inter-sensory relations. Moreover, the efferent motor copy is used to propagate in the network the reference signal responsible to generate motion. The use of this copy of the motor command is motivated by the fact that it provides an additional fault tolerance mechanism (e.g. if odometry sensors are broken).

The internal link between the two sub-networks is based on the transformation between robot reference system and world reference system. Computing the rotation matrix necessary to perform the coordinate transformation assumes that the absolute heading angle is known. Hence, the heading angle estimate from the heading network is fed into preprocessing units (of integrating sensors: odometry, efferent motor copy) of the position estimation network, such that the quantities are re-encoded in the world reference frame. This coordinate transformation is necessary to evaluate the performance of the proposed model in the world centred reference system representation. An important aspect is that the position estimate is not computed separately for incremental updates in x and y, rather we use the coupled vector (x, y) such that the position describes the continuous trajectory of the robot. At the current stage we performed our experiments in a relatively simple, uncluttered environment. Yet imagining a more complex scenario is straightforward as long as there are additional cues to measure and (if needed) internally build a map of occupancy. As a concrete idea, depth or ultrasonic sensors can be added to provide distance to objects. In this scenario the distance to objects will be computed in the egocentric reference frame of the robot, and will provide another cue which the network will fuse with the computed travelled distance (i.e. from the odometry, vision, and efferent motor copy). This way the network can infer a more precise travelled distance given also the occupancy information of the environment.

4.1.3 Experimental results

Heading angle estimation analysis

We analysed the behaviour of the heading estimation network for the complex trajectory depicted in Figure 4.7 a. After being randomly initialized in a stable state in which all relations are fulfilled, the heading estimation network receives raw sensory data samples from gyroscope, compass, wheel encoders and camera. In order to align the sensory data, the network preprocesses the raw samples to obtain an absolute heading angle, depicted in Figure 4.7 b. We observe that the inferred absolute angle values are not perfectly matching.

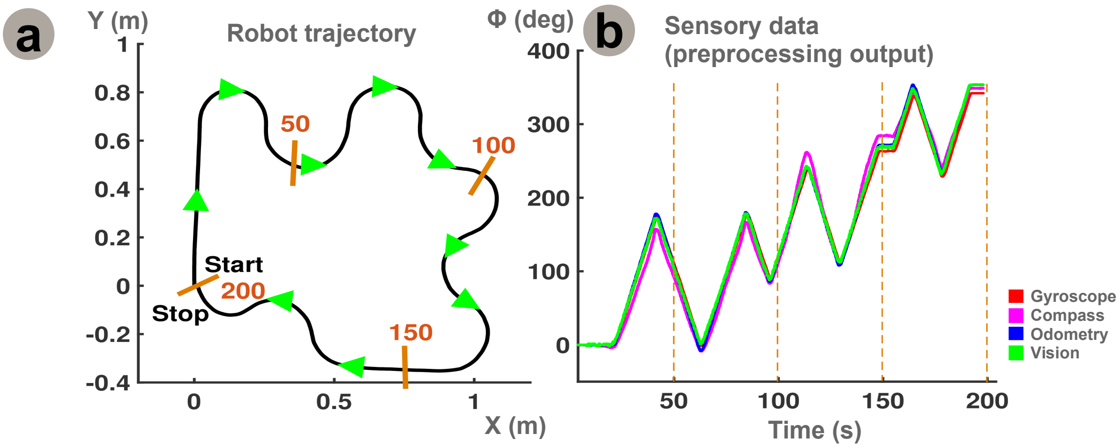


Fig. 4.7: Heading estimation network. a) Robot trajectory; b) Measured heading angle from sensors (preprocessing output).

This is due to the measurement noise and errors that are typical for angle measurements.

The gyroscope is affected by drift which accumulates over time, the magnetometer is affected by strong magnetic fields in the environment or the robot motors; odometry is affected by systematic errors like wheel slippage or imprecise sizes of the wheels; and the camera tracking is affected by changing illumination, strong dependency on the robot's velocity and is, in general, of low accuracy. The heading angle estimates do not wrap-around at a top value as sensors react differently to changes, and a wrap-around would determine a large mismatch (e.g. 360 degrees for a 2π radians wrap-around), compromising the network estimate. Figure 4.8 d shows the mismatch between the computed heading angle from sensory data along with sensory data and ground truth, emphasizing the network capability to improve, by fusion, the global estimate. When sensory data is continuously fed in, the network accommodates new observations by updating its own belief about the current state. At the low level, each unit modulates the influence from the units and sensors is connected with, such that only consistent data is used for updating its state. The network combines all contributions and enforces that all the quantities are close to the same value, Figure 4.8 e, f.

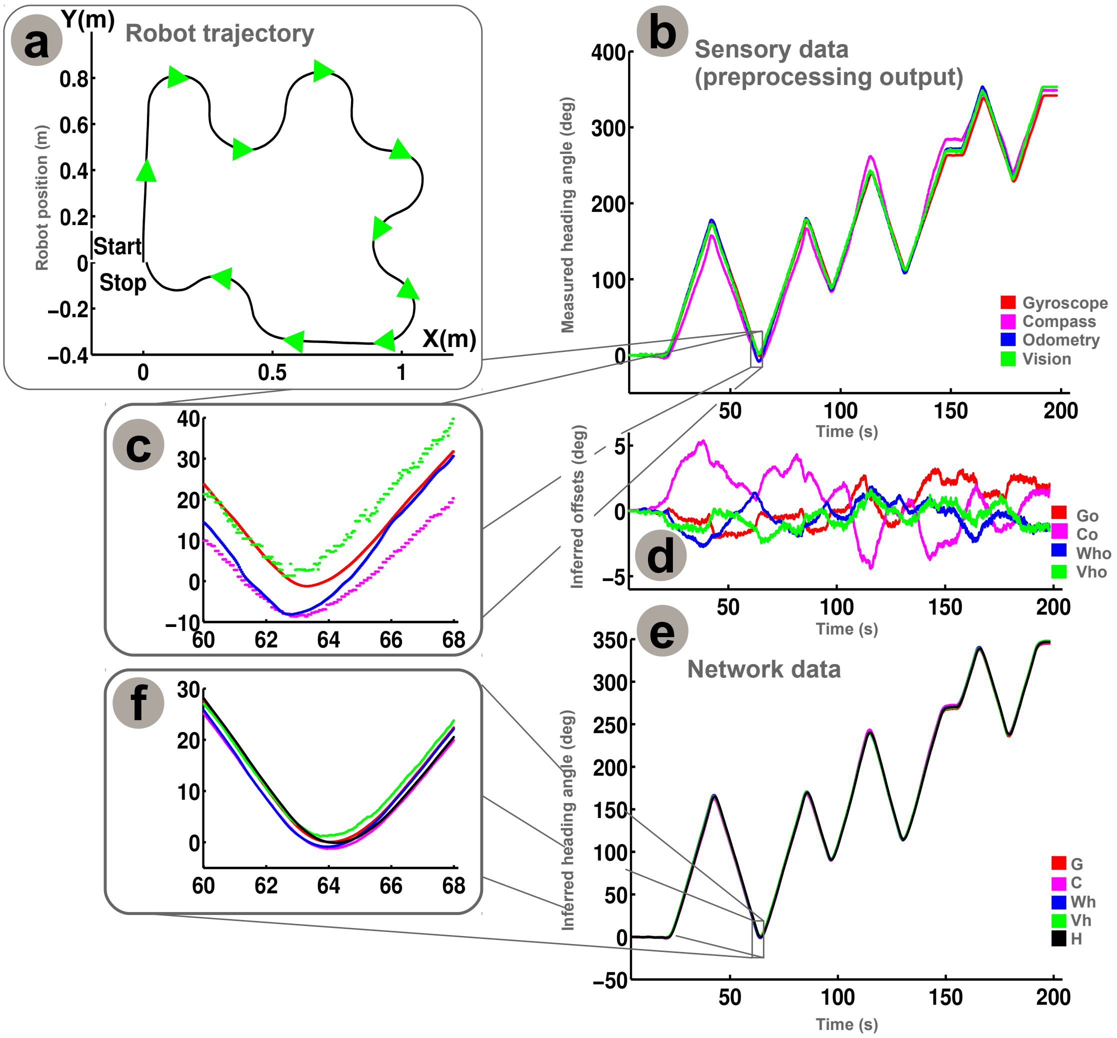


Fig. 4.8: Heading estimation network dynamics. a) Robot trajectory; b),c) Measured heading angle from sensors (preprocessing output); e),f) Inferred network quantities; d) Inferred offset values. When the network relaxes, inferred quantities fulfill the relations given external sensory data.

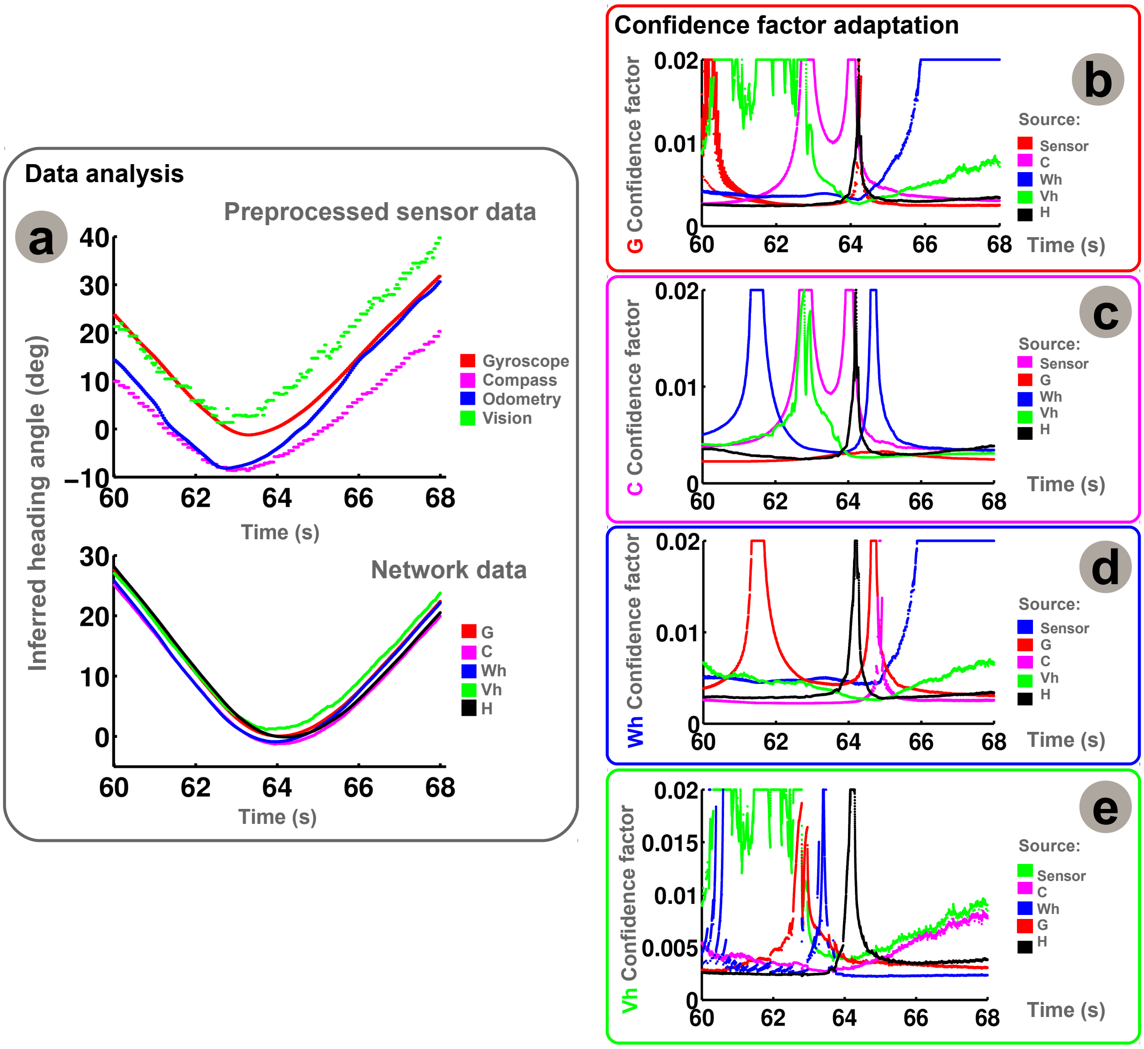


Fig. 4.9: Heading estimation network dynamics. a) Measured heading angle from sensors (preprocessing output) vs. Inferred network quantities; b), c), d), e) Confidence factor adapts according to the mismatch between local and incoming information in a unit.

The confidence factor adaptation is presented in Figure 4.9 b-e, on a per map basis. For each inferred main map (G, C, W_h, V_h) the confidence factor is computed as previously shown in Equation 4.4, such that each source of information contributing to a unit's update is compared against the other sources. Depending on the mismatch, the confidence factor is adjusted proportionally. In order to get a better intuition on the confidence factor adaptation we briefly analyse the sample update rule behaviour for a short time window of 10ms during operation in Figure 4.10. As previously mentioned this adaptation process defines the belief of the system on how consistent one contribution is to the overall network estimate. In this simple example we analyse the impact the magnetometer sensor has upon the network belief, in terms of it's contribution impact. Given the relative mismatch the sensor has with respect to the other sensors estimates in the network (i.e. G, W_h, V_h units) and it's own belief (i.e. C unit) the confidence value η_{C,S_C} will be lover than for example the contribution of the wheel encoders estimate (i.e. W contribution) given by η_{C,S_C} and higher than the contribution from the gyroscope estimate (i.e. G contribution). In order

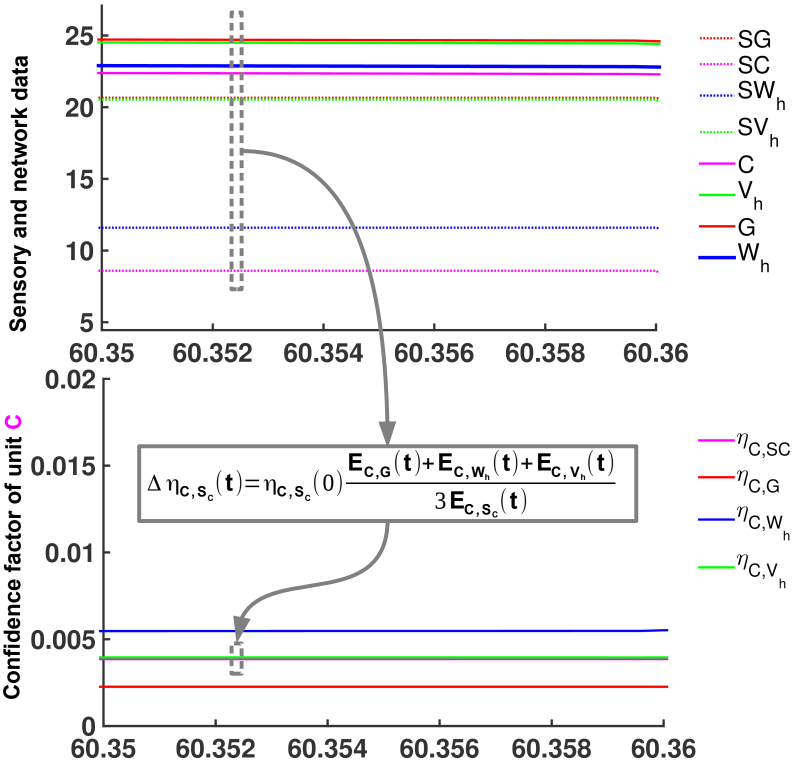


Fig. 4.10: Confidence factor analysis for the impact of the magnetometer upon network belief.

to ensure convergence in the real-time scenario, we set a lower limit on the confidence factor such that the network accommodates incoming sensory samples and settles in a solution. Moreover, in order to avoid uncontrolled increases of the confidence factor in extreme cases (e.g. high values of mismatch among a source and network belief), we clamp the confidence values to a maximum preset value. Figure 4.8 e,f depicts the inferred values for the G, C, W_h, V_h. The network brings these quantities into agreement with respect to the respective relations (imposed by the internal model). Changes in confidence factor support that. For example, if we consider unit G between t_1 = 60s and t_2 = 64s, one can see that the confidence factor with respect to V_h is high (saturated to a maximum imposed value) and all the others are low. This behaviour is supported by the graph in Figure 4.8 c, where we see that between t_1 and t_2, G and V_h values are overlapping while after t_2 maps store slightly different values. In order to assess the importance of an adaptive confidence factor we also performed experiments with fixed values of the confidence factor. Previous experiments have shown that the adaptive confidence factor is suitable for such a scenario considering dynamically changing contributions. Moreover, other variants of adaptation rules were explored (e.g. using the direction and amplitude of the relative error of a unit with respect to all other; using a fixed increment/decrement taking into account the amplitude of contributions mismatch). As expected, using a fixed weighting scheme (i.e. confidence values are identical) each unit was "pulled" with the same amount towards consensus. Yet, the global estimate proved to be less precise than the adaptive scheme, due to uniform trust level which allocated the same weight to all sensors even if their contribution was not consistent with the others. When using the fixed weighting mechanism, the weight (i.e. confidence factor) each input source gets is identical and based on an average standard deviation of each source estimate, measured prior to the experiment. In the adaptive scheme the weights are adjusted depending on the statistics of individual sources, such that higher deviations determine lower values and small deviations determine higher values of the weight, as shown in Figure 4.9 b-e.

An additional quantity inferred by the network, useful in detecting and compensating sensor errors or biases, is the offset, depicted in Figure 4.8 d. Each main map in the network has an associated offset node, which will store a relative mismatch with respect to the other units. The offset node update depends on the type of its respective unit, a faster update for integrating sensors and a slower update for absolute sensors. Moreover, this quantity is used to provide an additional source of consistent data, that will use only knowledge inferred internally in the network and contribute to a unit's update. Given the sensory data the network finds a solution which fulfils all the embedded relations, and keeps all inferred quantities in agreement, as shown in Figure 4.8 e.

Extended analysis on special cases

In order to further extend the analysis of the network capabilities in the 2D egomotion scenario we revisit offset node computation and confidence adaptation in a simple (one-loop) scenario and a complex (multi-loop) scenario, respectively. As also shown in the initial experiment the robot moved on a predefined trajectory (i.e. a square) and sensory data was acquired and fed to the network for heading and position estimation. As on-board sensors react differently to the robot's motion (i.e. different transducing techniques) there's a clear final misalignment in the sensory estimates of the motion components, Figure 4.11 a, f. In these extended experiments we follow the same methods and procedures, and we feed the data in the network such that each sample is presented to the network until the network converges to a solution given all cross-sensory constraints. Offset nodes are connected to each map and estimate the global offset (mismatch) to the other maps in the network. An offset node receives input from its corresponding map and all the other maps in the network and feeds back to the unit, such that the unit can compensate for it (i.e. minimise another global source of mismatch between it and the other maps). The offset nodes have an impact on the long-term effects of sensory anomalies (e.g. gyro drift, odometry offset) by providing a new source of information at the map level. This will also impose a constraint, forcing the local estimate to minimize the offset to the other maps. If we analyse the dynamics of the offset nodes we see that they obey same dynamics as all the other maps in the network, Figure 4.11 b,c, towards ensuring that there's no mismatch between their corresponding maps Figure 4.11 e. As soon as the network maps converge to a value, given sensory input and mutual influence, the offset maps will have decreasing values, Figure 4.11 c around t=1s. Overall, if we analyse the network behaviour, we see that the offsets, visible in Figure 4.11 f, are minimised through the network dynamics, Figure 4.11 g, such that network estimates are closer to each other, Figure 4.11 h. Offset values are signed such that they provide also the direction to take towards minimising the mismatch. For each input sample presented to the network, there are a number of update rules to fire, each rule corresponding to the update of a map from an input source. Each rule ensures that each map in the network receives updates from all its sources, and after one iteration the input sample was propagated through the network.

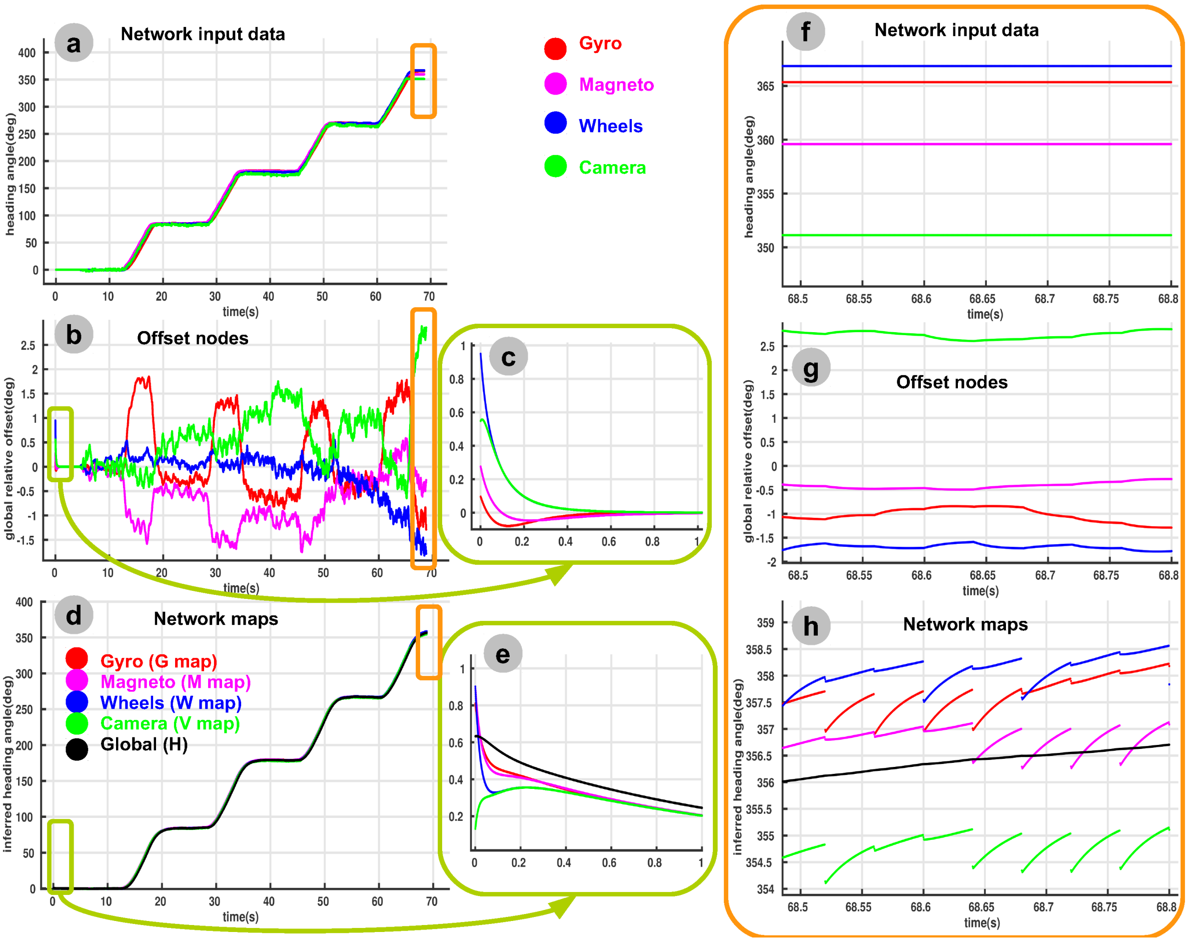


Fig. 4.11: Single loop scenario: Network analysis. a) Input data; b) Offset nodes; c) Offset nodes dynamics; d) Network data; e) Network data dynamics; f) Input data at end of trajectory; g) Offset nodes at end of trajectory; h) Network data at end of trajectory.

This process is performed until all relations between the units are fulfilled (100 network iterations). The sudden changes in the maps values, Figure 4.11 h, are present due to the impact of an input source upon the local map estimate (i.e. a source consistent with the current value will be enhanced). Moreover, we see that all values are pulled towards agreement. The global trend is consistent (Figure 4.11 h, underestimating maps are pulled upper and overestimating ones lover), such that given the most trustworthy source at each iteration the maps moves towards it. Another special case we investigate is the operation in a multi-loop scenario. In this experiment, the robot followed the prescribed trajectory for around five times, accumulating up to 1900 degrees in heading angle, as shown in Figure 4.12 a. We observe that all sensory cues agree and at some moment (t=140s) there's a glitch in the data acquisition such that the odometry overestimates the heading angle while all the other cues follow consistently the motion profile. This analysis is focusing on investigating the fault tolerance capabilities of the network and adaptation for efficient integration. As previously mentioned, each source of information that a certain map in the network receives is modulated by a confidence factor. This factor is just a quantification of how similar one source is with respect to the local map estimate, and is based on a comparison with all the other contributing inputs. If we analyse the confidence

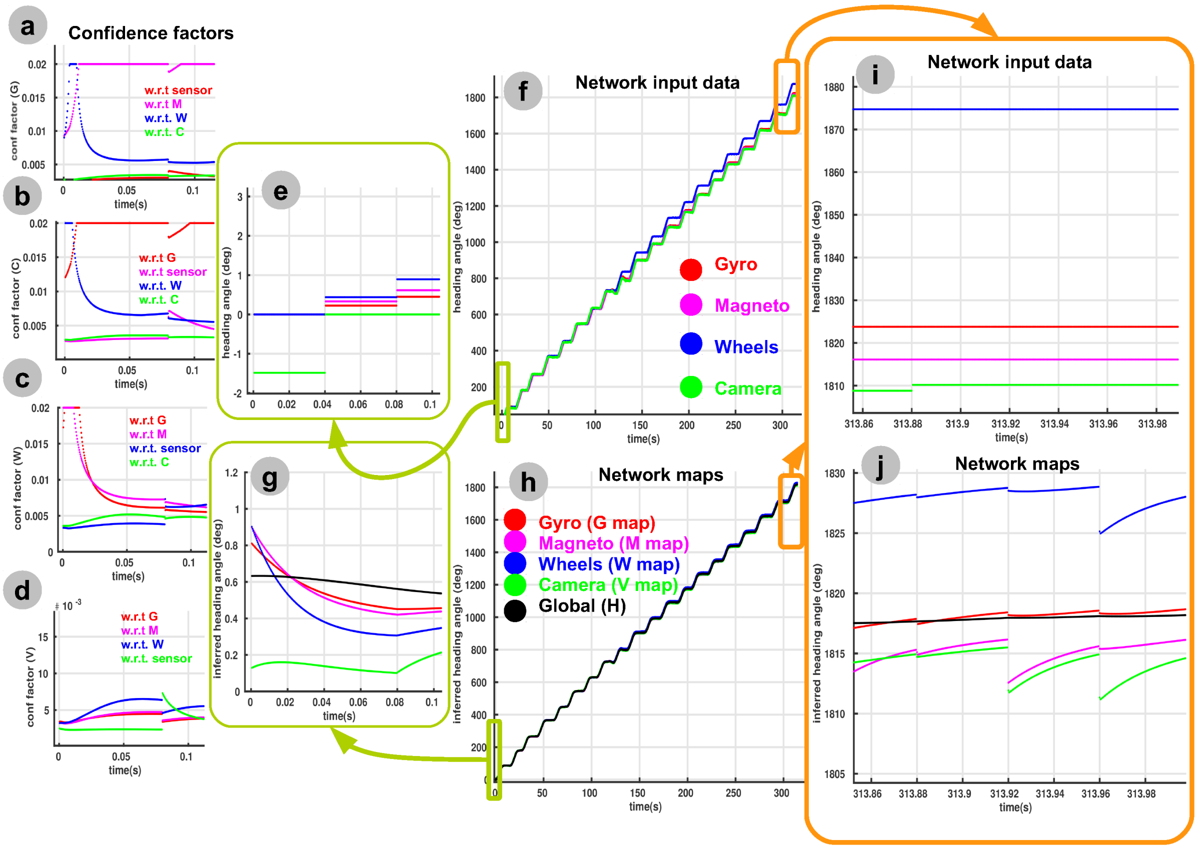


Fig. 4.12: Multi loop scenario: Network analysis. a) Confidence factor analysis for map G; b) Confidence factor analysis for map M; c) Confidence factor analysis for map W; d) Confidence factor analysis for map V; e),f) Input data; g),h) Network data; i)Input data at end of trajectory; j) Network data at end of trajectory.

factor adaptation at each map level (Figure 4.12 a to d) we see that each source, be it the sensor (Figure 4.12 e) or the other maps (Figure 4.12 g) contribute to the local estimate proportionally to the level they agree / disagree. For example, if we consider the G map confidence factor adaptation, in Figure 4.12 a, and analyse it's behaviour between t1 = 0.5s and t2 = 1.0s, we see that there's a high confidence in the magnetometer map (M) supported by the close estimates in Figure 4.12 g; the confidence values with respect to the odometry map (W) and vision (V) are low, and proportional to the distance to the local G estimate; finally the confidence in the G map sensor (the gyroscope) is minimised due to the mismatch between the current value (at t=0s maps are randomly initialised, G 0.8 deg) and the 0 input from the sensor. This effect is explained by the confidence adaptation rule, which is based on a voting scheme. If one source is far from the mean mismatch to all the other sources of information it is penalised (low confidence factor), and if consistent it is enhanced. Confidence factor adaptation is a process which acts upon network dynamics on short a short timescale allowing the network to converge towards a globally consistent value. In order to analyse network's convergence behaviour in the presence of inconsistent sensory data and using all the underlying processes (i.e. confidence adaptation, network dynamics with offset compensation) we analyse the settled state at the end of the robot operation, as depicted in Figure 4.12 i, j. We can see that the network pulls the W map value towards a global consistent value dictated by all the agreeing maps (G, M, V maps), Figure 4.12 j, given inconsistent sensory input (mismatch 50 deg), as shown in Figure 4.12 i.

Position estimation analysis

In the current section we analyse the behaviour of the second component of our model, the position estimation network. Similar to the heading estimation network, raw data from wheel encoders, a copy of the PWM signal and vision data, are presented to the network. Subsequently, the network preprocesses raw data to obtain an estimate of 2D position. The position network operation is depicted in Figure 4.13. As one can observe in Figure 4.13

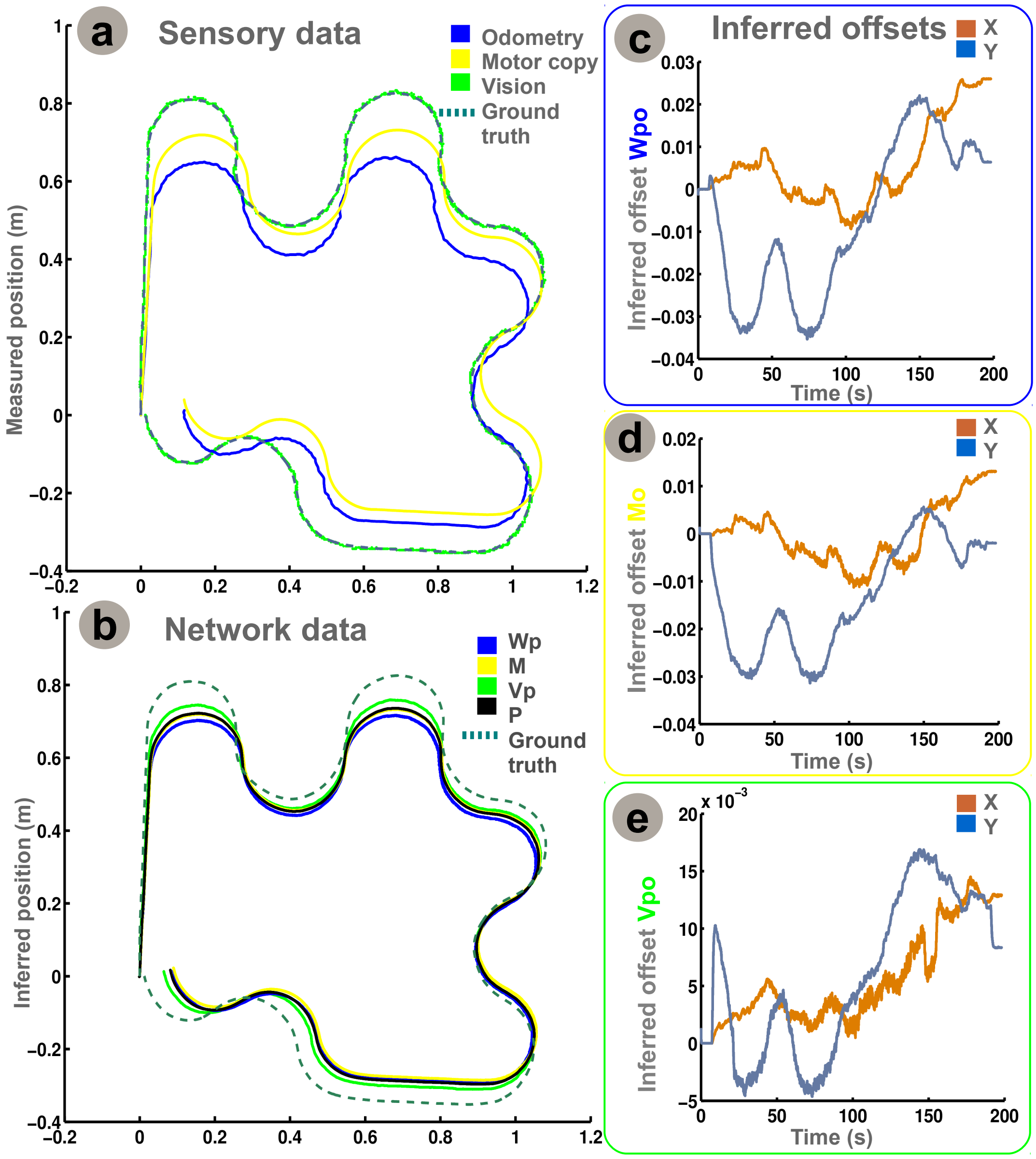


Fig. 4.13: Position estimation network dynamics. a) Comparison between measured position from sensors and ground truth data; b) Network inferred quantities and ground truth data; c), d), e) Inferred offsets for each map.

a, no modality estimate is able to provide a precise position estimate. Despite strongly conflicting estimates, the network brings all quantities in agreement, such that a more precise global estimate is inferred. Figure 4.13 b depicts the network data. Following the same principle with the heading angle estimation network, each map penalises contradictory sources of information (providing a low confidence factor) and enhances contributions from consistent sources (high value of confidence factor). In order to assess how each modality main map (V_p, W_p, M) is updated under the influence of the sensory data and local network belief, we can analyse the inferred offsets in Figure 4.13 c-e as they quantify a relative mismatch between the units. One can observe a mismatch in the inferred quantities which emerges due to the fact that each sensor has a different response time and the network cannot influence (by prediction and correction) the sensory readings explicitly.

In order to measure the performance of our model we compare it with two state-of-the-art methods: the Kalman filter and the Maximum Likelihood Estimator (MLE). State-of-the-art methods need already preprocessed data (i.e. absolute angle values) due to the fact that they lack explicit mechanisms to handle raw data. Hence, for heading estimation both Kalman filter and MLE receive four sources of absolute angle. Using sensory observations, each model updates the modelled system state representation such that we can directly read out an estimate of heading angle, similar to typical implementations [Durrant-Whyte et al., 2008, Thrun et al., 2005]. Figure 4.14 compares the results of the

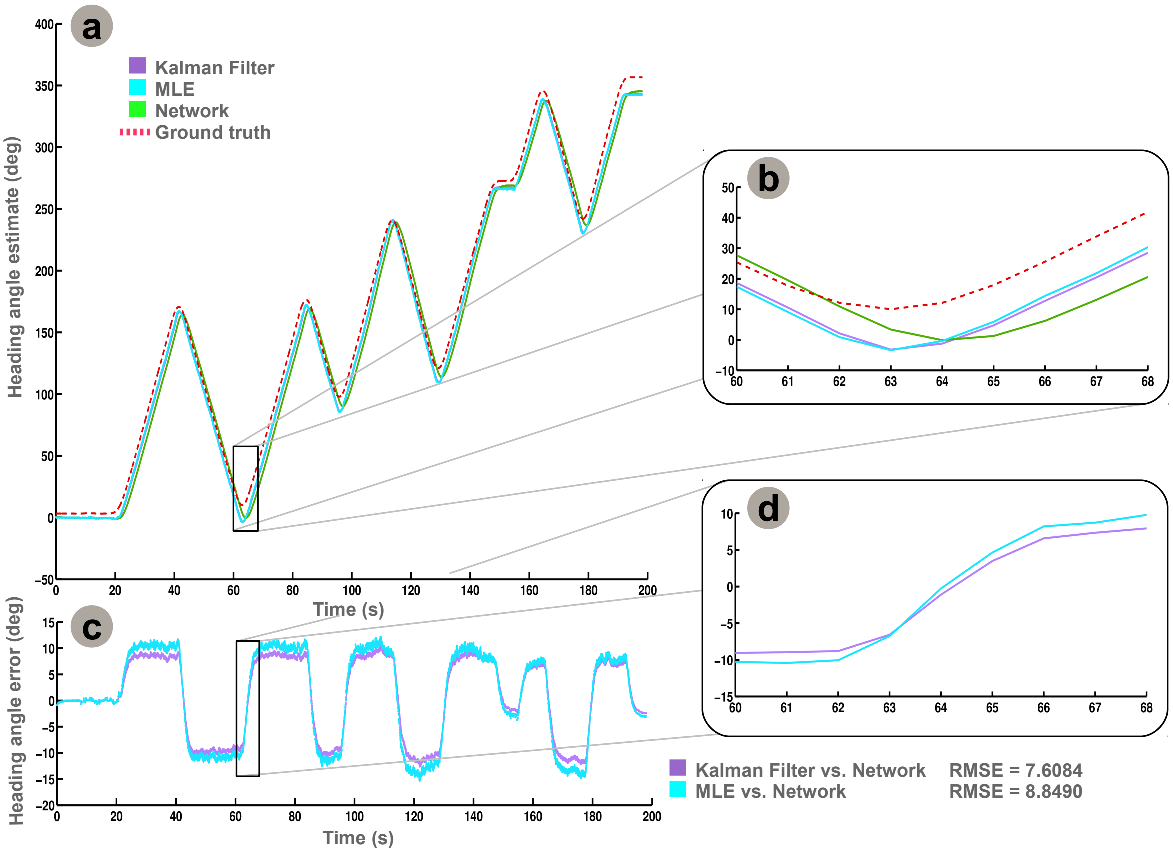


Fig. 4.14: Heading estimation evaluation. a),b) Comparison between network estimate, Kalman filter estimate, MLE estimate, and ground truth data; c),d) Heading angle estimation error and RMSE values.

network response with the other algorithms. We observe that the network response is slower than both Kalman filter and MLE approaches. This is due to the additional operations to preprocess the data internally. This internal preprocessing stage (temporal integration and offset compensation) is performed in the network while new sensory observations are received and it takes place at the same time with the estimation, obeying same dynamics. This effect is not present when implementing the network on parallel hardware in Section 4.2. In Figure 4.14c it can be noticed that the network overestimates or underestimates the values from the other estimators when significant changes in orientation take place. This is motivated by the fact that in the considered scenario the robot often changed direction with different angles and the network needed time to accommodate the changes and balance different sensor contributions. As sensors react differently to changes, the network identifies which contributions are consistent and accommodates them as new observations are available. Albeit the network does its best in balancing the represented quantities autonomously, it can also accommodate externally imposed constraints (i.e. user can set a preferred value) to exhibit slower responses with higher accuracy or faster responses with lower accuracy. In order to quantify the network performance, the RMSE was calculated against Kalman filter and MLE estimates with respect to ground truth data. In our scenario a smaller RMSE value describes a better estimation. Given available sensory data, our network estimate is comparable with state-of-the-art estimators with <10% RMSE, as shown in Figure 4.14 c,d. For position estimation, both Kalman filter and MLE receive three sources of (x, y) position, inferred from a copy of pulse width modulated (PWM) motor command, wheel encoders data and vision data. The network infers a global estimate comparable with state-of-the-art estimators given the available sensory data, as shown in Figure 4.15 a,b. Figure 4.15 c,e show that our network is comparable to both Kalman filter and MLE as the measured RMSE values for position estimation are smaller than 1%. Furthermore, one can see in the decoupled analysis on each axis shown in Figure 4.15 d,f that the network is close to estimates of the two state-of-the-art methods. Another quantity inferred by the network is the travelled path, P_i, computed as an accumulation of all intermediate position estimates, P. The P_i unit integrates the average position provided by the global position estimate, P. Furthermore, to quantify the precision of the main maps in the network (V_p, W_p, M) we computed the corresponding travelled path values, as given by the integration of the successive position estimates in each of the maps, and the individual modality path deviation from ground truth. The computed values are given in Table 4.1 and one can see that the values are globally consistent (with an error smaller than 3 cm).

Network unit	Inferred travelled path (m)	Path deviation (m)
Vision, V_p	4.212	0.052
Odometry, W_p	4.160	0.030
Motor efference, M	4.177	0.041
Global average, P_i	4.183	0.040
Ground truth	4.190	-

Tab. 4.1: Inferred travelled distances and path deviations from individual modalities.

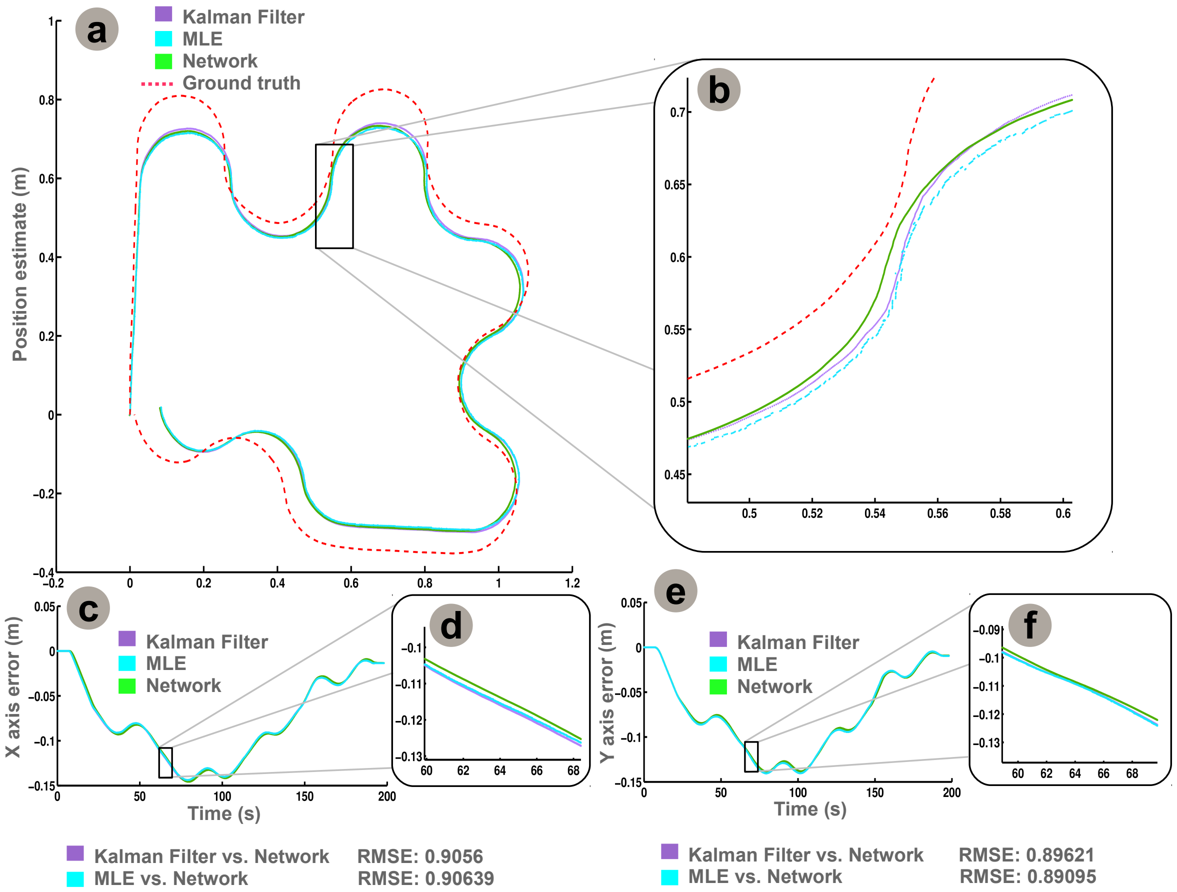


Fig. 4.15: Position estimation evaluation. a) Comparison between network estimate, Kalman filter estimate, MLE estimate, and ground truth data. c),d) X axis position estimation errors for each estimator and relative RMSE values; e),f) Y axis position estimation errors for each estimator and relative RMSE values.

In order to assess the fault tolerance capabilities of our network we performed an additional set of experiments in which we tested the network in the presence of faulty sensors. We clamped the sensor (i.e. magnetometer) readings to 0 for a certain amount of time during operation. After some time we re-activated the readings simulating just a temporary failure. Figure 4.16 a illustrates the sensory data and the faulty transient for the magnetometer. Moreover, in Figure 4.16 b we can see that even if the sensor is not usable, the network does its best to infer its representation from the other available sensors. This is fully supported by the dynamics of the confidence factor of map C (encoding the magnetometer representation for heading angle) with respect to the incoming sensory data. The confidence factor decreases immediately as the sensor is faulty at $t_1 = 50$s and is restored to a high value once the readings are consistent with the network belief at $t_2 = 150$s. One interesting aspect to mention is that during the faulty transient the confidence factor changes dramatically (around $t_3 = 65$s) when, as one can see in Figure 4.16 b, the sensor is consistent with the other sensors and network belief. This experiment was meant to quantify the robustness against faulty sensors of our network and provide an insight on the simple and efficient mechanism behind it.

The current subsection presented results supporting the capability of our network to

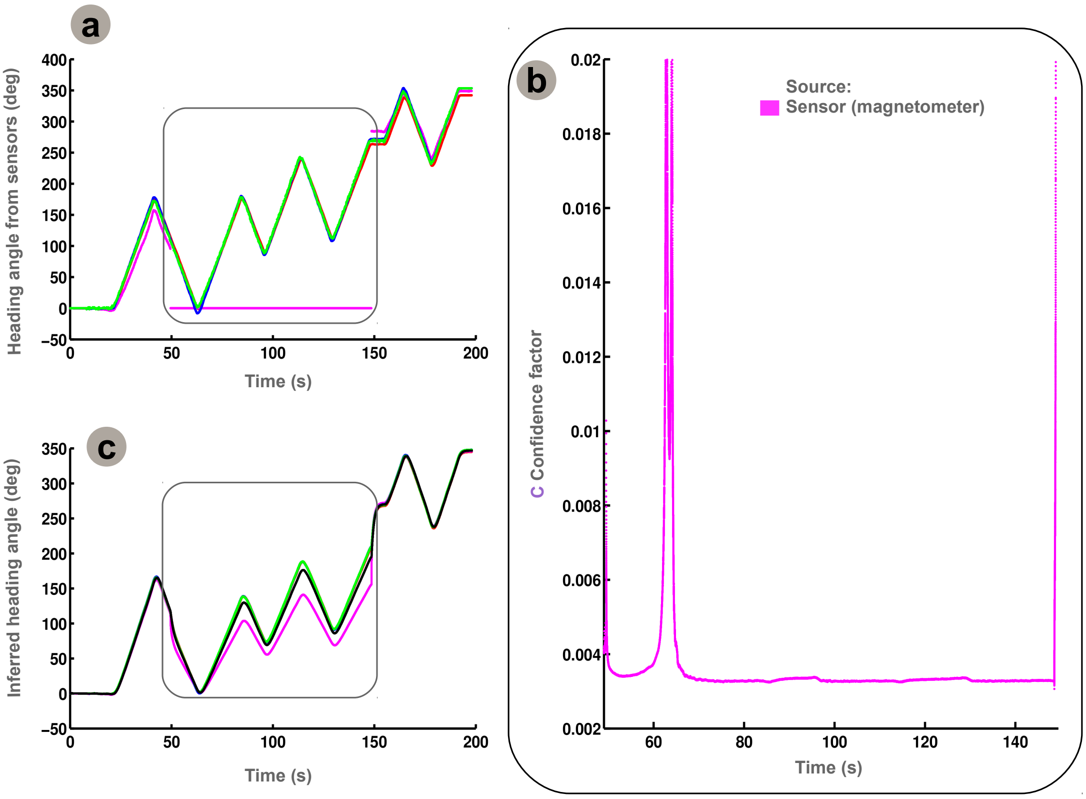


Fig. 4.16: Fault tolerance analysis. a) Heading angle values from sensors. The magnetometer is faulty between $t_1 = 50s$ and $t_2 = 150s$. b) Inferred heading angle values in the network. Even if the sensor (i.e. magnetometer) is faulty the network compensates for that, and infers its representation from the other modalities. c) Confidence factor adaptation w.r.t magnetometer.

infer a position estimate from each modality, and subsequently combine them into a global estimate more precise than individual modalities.

4.1.4 Discussion

In the previous sections we introduced a sample instantiation of our cortically inspired framework for multisensory fusion, which uses principles of neural processing to combine contributions from different sensors and infer an estimate of both position and orientation of a mobile robot. It is unanimously accepted that an organism's possible actions and movements are conditioned by the environment. Egomotion estimation contributes actively in shaping this space of possibilities.

While moving, both real and artificial organisms receive a constant flow of information from parallel sensory channels, bind and compare the stimuli with previous experience and goals, and produce motor outputs to match the current circumstances. Yet, combining sensory contributions is not a trivial task, sensory contributions must be aligned in a common reference frame depending on their spatio-temporal properties, and the resulting representation should be plausible and informative to disambiguate the scenario.

State-of-the-art methods for multisensory fusion provide good results for dedicated scenarios but lack the generality, failing to accommodate different contexts from the ones considered at design time. The complexity of real-world scenarios goes over the prepared environment of the lab, and the impact of complex environments on multisensory fusion is likely to become a major issue, as models become more and more sophisticated [Khalengi et al., 2013].

Alternatively, there is evidence that our brain is able to combine different information streams from available senses and use the combined representation in a flexible manner to robustly orient behaviour. Albeit a large number of putative models which were developed, it seems that biological systems tend to combine not only exteroceptive, and interoceptive cues, but also psychological and cognitive cues when integrating senses [Rowland, 2012].

In order to build and maintain a precise representation of the self in the environment, sensory cues conveyed from egomotion must be combined to precisely guide actions. Psychophysical studies in human spatial cognition hypothesize that behaviour can arise from perception. A unifying theory introduced in [Mitchel, 2010] provides a possible mechanism that recognizes and/or creates spatial identity or similarity between various sensory experiences (e.g. kinesthetic, visual) to enhance cognition about the environment. Following this hypothesis, our model combines contributions from different sensors for estimating robot's position and orientation. For heading angle estimation the model enforces identity between the individual absolute angle estimates computed from the raw gyroscope, magnetic compass, wheel encoders and vision sensor data depicted in Figure 4.7 b. Fusing the different sensors' contributions enhances the global estimate over individual estimates, because the integration process compensates for sensor errors and noise in individual measurements, as shown in Figure 4.8 e, f. In order to estimate position, the model receives wheel velocities from the wheel encoders, 2D position from vision and a motor command copy. The raw data is preprocessed by the network which infers three different sources of robot absolute position as shown in Figure 4.13 a. These individual estimates are kept in agreement by the network which computes also an average estimate from all contributions as shown in Figure 4.13 b. The global average is a quantification of network's belief and can be used in planning more precise motor commands.

Egomotion estimation provides the organism the capability to understand the environment from its own state. Evidence in motion psychophysics and kinesthesis [Sheets-Johnstone, 2010] enforce the hypothesis that humans build their perceptions and conceptions of space when they learn their bodies and move based on some innate kinetic dynamics. Furthermore, they develop more complex notions of space (e.g. connectivity, distances to objects, occlusions, objectification, [Sheets-Johnstone, 2010]) useful in conceiving themselves to spatial bounds and layout. The current instantiation of our model is able to estimate both egomotion components, position and orientation, from available sensory data. Derived quantities (i.e. do not have an associated sensor) can be also inferred by combining other quantities in the network. For example, each position estimate provided by the network is accumulated in a global travelled distance unit (i.e. $\mathrm{P_i}$) and can be used to compute distances to objects in the environment. In a sample use case, the first step is to combine global (P) and camera ($\mathrm{V_p}$) positions estimates, such that the network determines which areas of the environment were already traversed by simply matching the positions from the two sources. In the second step, the travelled distance ($\mathrm{P_i}$) provides the absolute distance to occupied areas of the environment detected in the first step. To support higher level representations or derived quantities, the network accommodates new sensors by simply defining additional connections. This is beneficial as the network can be easily extended such that complementary sensors can be fused to yield an occupancy map, which along with the self-motion cues can provide a complete description of the environment (e.g. SLAM).

The perception of the environment and body orientation are influenced by multiple sensory and motor systems [Lackner et al., 2004]. In order to handle the variability and complexity of the environment, available sensory are combined such that interference and conflicts between the individual measurements are minimized and a more precise estimate is obtained. For technical systems, in order to build such a representation the system designer must precisely describe a) the system model, b) the prior information about the sensory observations and the system, and c) the preprocessing steps, as shown in [Durrant-Whyte et al., 2008, Thrun et al., 2005]. In order to relieve the system designer from the difficult task of describing the aforementioned aspects, our model simplifies the representation and fusion mechanism by using a different processing paradigm inspired by cortical computation principles. Basic mathematical relations link different processing units which use feed-forward and feedback connections to exchange information, as shown in Figure 4.4. The network tries to keep all quantities stored in the units in agreement given noisy sensory data, Figure 4.8 b, c. Despite the fact that each source of information is affected by noise or systematic errors, the network is able to detect abnormal changes in sensory data, such that there is a small impact over its internal belief. Preprocessing is performed inside the network, such that sensory contributions are aligned to a common representation, without increasing the network complexity. The preprocessed data flows into the network and each unit balances contributions from all its connections. An adaptive mechanism (i.e. confidence factor) modulates the influence of external information sources, to penalize strongly conflicting estimates and enhance consistent values, as shown in Figure 4.9 b-e. Moreover, the network uses relatively simple dynamics for unit update such that it converges rapidly to a solution, shown in Figure 4.8 e, f and Figure 4.13 b, given the constraints imposed by the embedded relations and sensory data. As sensory data mildly influences the network activity, in the absence of one sensory modality the network can recover the missing quantity based on the other modalities and the connectivity. Relevant experimental results are illustrated in Figure 4.16 b. This inference capability accounts for a fault tolerance mechanism. Assuming that temporarily a sensor doesn't provide any measurements, its value will be continuously inferred by the network such that when it will become online it's impact will be modulated by the network belief and progressively accommodated in the network, as shown in Figure 4.16 c. Continuously refining its own belief given available sensory data, the network provides an estimate which is comparable with state-of-the-art methods, as shown in Figure 4.14 c,d and Figure 4.15 c-f. Although for the current scenario the network relations were hard-coded by the designer, we expect to extend this model such that relations emerge from learned correlations in sensory data. The network will no longer need a predefined structure as the incoming stream of sensory data will shape it's connectivity given the cross-modal interaction patterns. These extensions are inspired by the underlying neural circuits for multisensory fusion in cortex and their experience based development and plasticity. Another intuitive extension focuses on self-organising-maps learning and specialisation principles, through competition and cooperation. This extension can be accommodated in the existing structure as representation and processing capabilities of the units can be modified without altering the network level dynamics. The proposed processing scheme provides many advantages, in terms of implementation and complexity, being able to distribute computation, evaluate and balance contributions of the fused sensory data, while using only relatively simple operations representing cross-sensory relations. Chapters 5 and 6 provide insight in the extension capabilities of the network using a formal model supported by application scenarios.

Given noisy and sometimes conflicting sensory data, multisensory fusion, is crucial for precise egomotion estimation. Our model introduces a new approach for multisensory fusion. Using a cortically inspired processing paradigm our model provides results comparable with optimal state-of-the-art methods. Without precise modelling and parametrisation of the system model, our network is able to combine information from multiple sensors into a global estimate, more precise than individual estimates. Balancing external sensory contributions with its internal belief, the network is able to detect and compensate for sensor inconsistencies. By distributing computation, such that each unit processes and stores only local information using only basic mathematical relations, complexity is reduced. The current instantiation of the model for egomotion estimation provides comparable results with state-of-the-art methods in terms of estimate accuracy, but with less design challenges. Finally, our network is highly parallelisable, making it suitable for implementations on massively parallel hardware architectures for real-time robotics applications. Given its generality, computational efficacy, and ease of implementation our model is a promising candidate for multisensory fusion in robotic applications.

4.2 Probing model parallelization: Multisensory fusion network for mobile robot egomotion estimation on massively parallel hardware

In the previous section and in Chapter 3, we described our model as a distributed network in which independent neural computing nodes obtain and represent sensory information, while processing and exchanging exclusively local data, to infer an estimate of robot orientation and position. In order to take advantage of the parallel processing capabilities of the network, we explored the implementation [Simlinger et al., 2015] on a massively-parallel computing platform SpiNNaker [Furber et al., 2013]. Inspired by the fundamental structure and function of the human brain, which itself is composed of billions of simple computing elements, in SpiNNaker computing cores communicate and process only locally available data. Given various sensory inputs, and simple relations defining intersensory dependencies, the model takes advantage of the inherent hardware parallelism of the SpiNNaker platform to ensure convergence into a consistent interpretation of the perceived motion. In order to evaluate the performance of our approach we also implemented a standard version of the Kalman filter as well as a distributed Kalman filter. Next section introduces our choice of the computing platform, describes the network allocation and partitioning techniques on the hardware, as well as an evaluation against other computing platforms (i.e. standard PC, multi-core PC).

4.2.1 Massively parallel neuromorphic hardware

SpiNNaker is a novel massively parallel computer architecture, inspired by the fundamental structure and function of the human brain. This novel hardware architecture provides a platform for high-performance computation suitable for simulating neural models in real-time, and provides a great research tool for both neuroscience and robotics. Due to its distributed, asynchronous, and low-power embedded processing capabilities, SpiNNaker was chosen as a suitable candidate for probing the parallelization capabilities of our model.

The experiments were conducted on the SpiNN-3 model of the SpiNNaker family, Figure 4.17 a, which features four chips with 18 ARM968 cores per chip (16 core usable, 1 core for management, 1 spare core) as displayed in Figure 4.17 b. The ARM cores clock at 200 Mhz such that the board requires a 5V at 1A power supply, and everything is packaged in a 9x8cm form-factor embedded board. The internal architecture of each chip and core is depicted in Figure 4.18 a and Figure 4.18 b respectively.

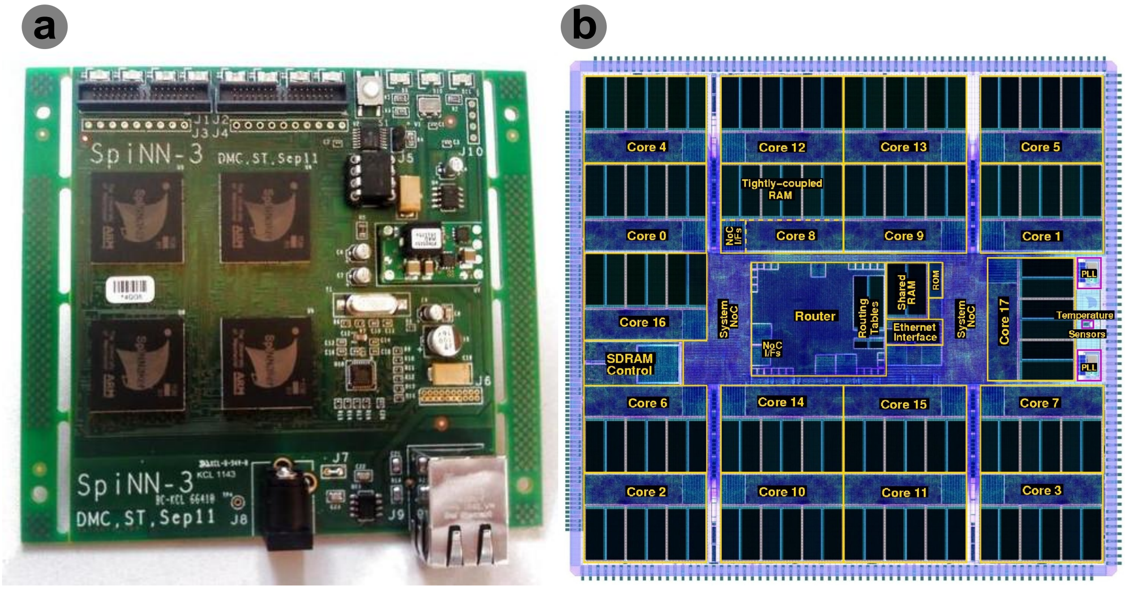


Fig. 4.17: Massively parallel hardware platform used in experiments: a) SpiNN-3 board; b) Hardware layout.

4.2.2 Mapping the neural model to hardware

As shown in previous sections, in our framework, maps represent a uni-/multi-dimensional representation of a sensory quantity (e.g. scalar, vector, field, matrix). Different sensor modalities are encoded using maps and the network dynamics tries to settle in an agreement state by exchanging exclusively local information. Based on this asynchronous and continuous data exchange, the maps update their local belief in a gradient descent fashion until the network converges to a relaxed state (i.e. global consensus).

In order to take advantage of the asynchronous address-event-representation (AER) of the SpiNNaker architecture, we partitioned the network on the hardware such that we exploited the intrinsic capabilities of the model. Using a similar mapping strategy for the heading as well as the position estimation network, we evaluated the network performance in terms of full egomotion estimation precision, as shown in Figure 4.20. We

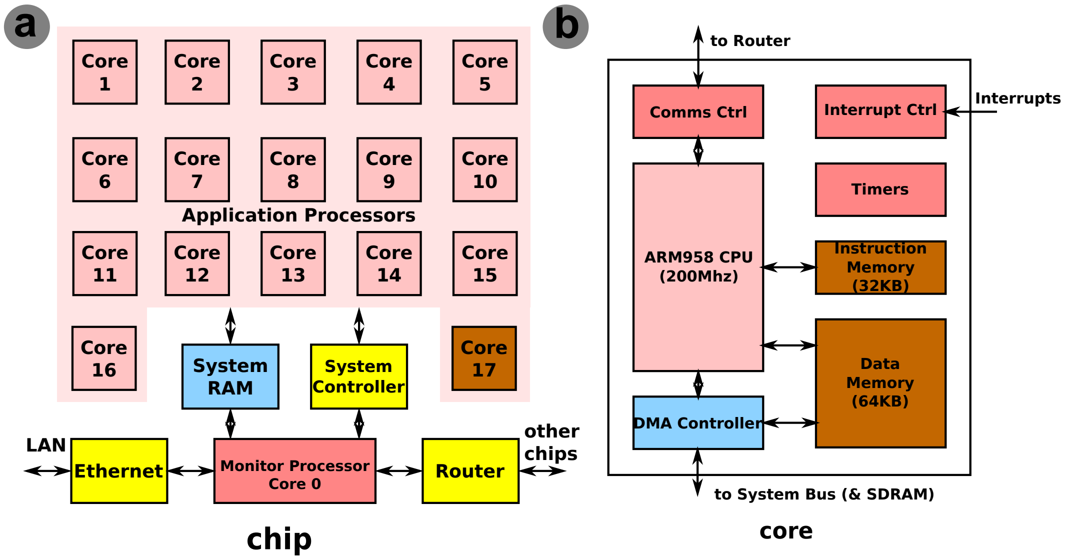
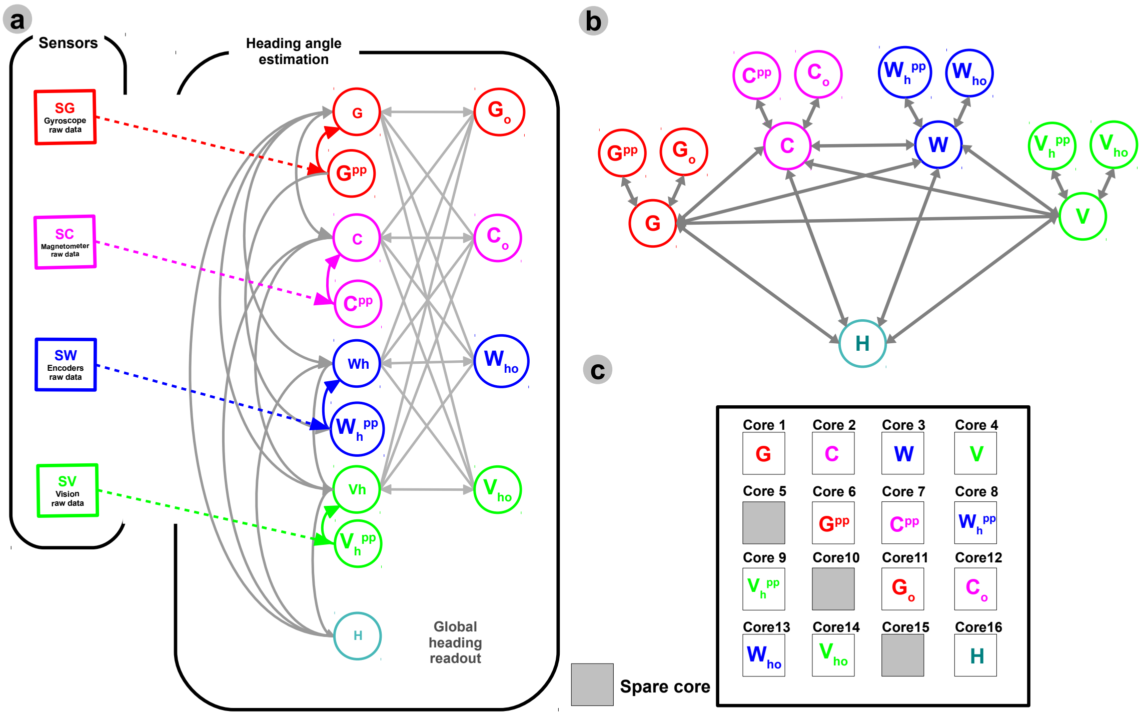


Fig. 4.18: Massively parallel hardware platform used in experiments: a) Internal architecture a the chip; b) Internal architecture of a core.



Core 1	Core 2	Core 3	Core 4
G	C	W	V
Core 5	Core 6	Core 7	Core 8
	G^{pp}	C^{pp}	W_h^{pp}
Core 9	Core10	Core11	Core12
V_h^{pp}		G_o	C_o
Core13	Core14	Core15	Core16
W_{ho}	V_{ho}		H

Fig. 4.19: Mapping the model on hardware: Sample dispatching for heading estimation network. a) Heading angle estimation network; b) Processing map associated with the network: bidirectional, asynchronous message passing; c) Map partitioning on hardware cores.

observe that the precision of the network is high for both heading angle, Figure 4.20 a, and position, Figure 4.20 b, and that the partitioning and resource allocation preserves the core functionality of the network. In this experiment, instead of using a sequential update of all sensory quantities in the network (results in previous section, using an embedded microcontroller or a desktop PC) the network obeys an asynchronous message passing protocol provided by the hardware platform. This update scheme makes inference more robust and flexible while ensuring precise results.

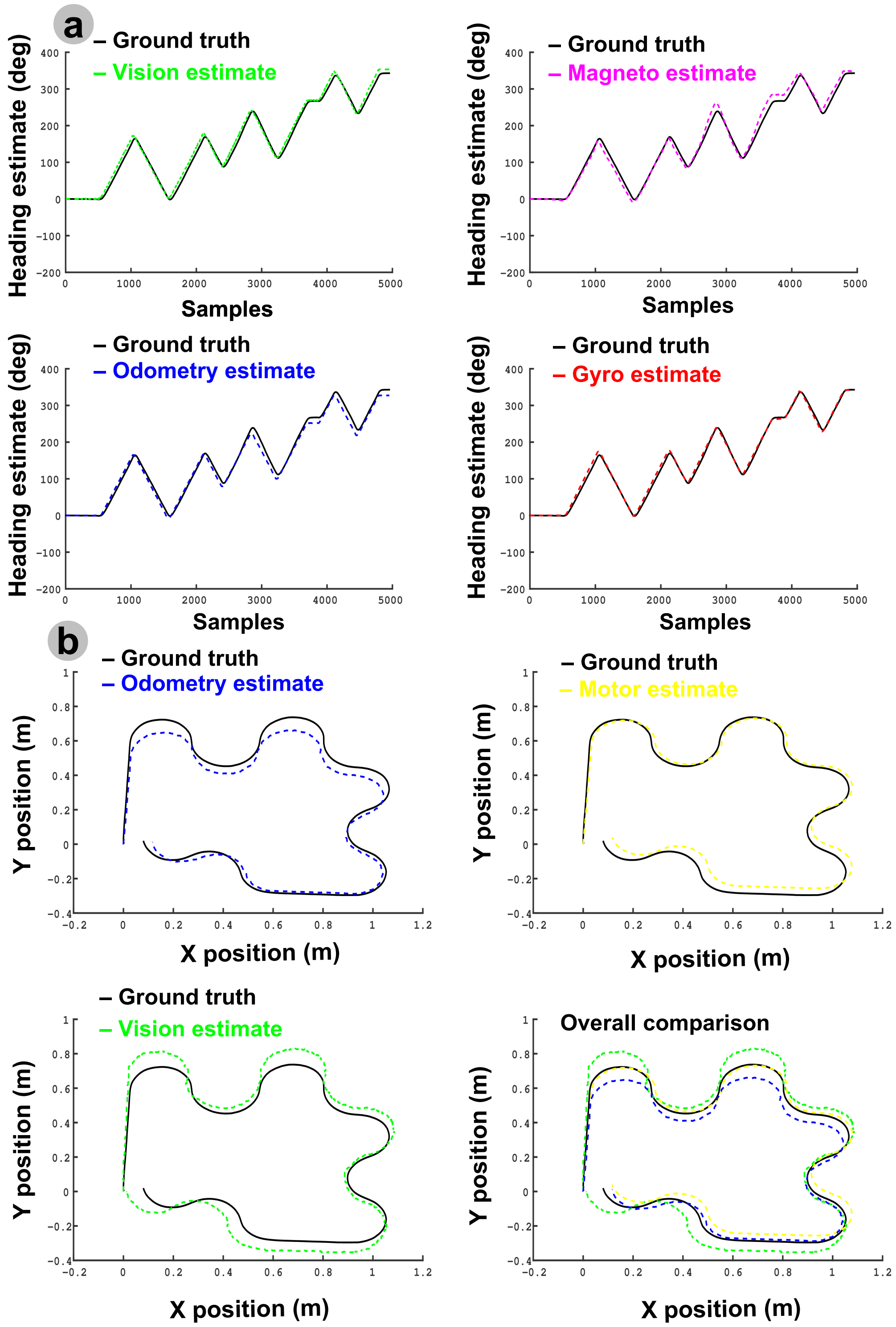


Fig. 4.20: Evaluating the SpiNNaker implementation of the neural model. a) Heading estimation; b) Position estimation.

4.2.3 Mapping the Kalman filter to hardware

One state-of-the-art mechanisms widely employed in sensor fusion is the Kalman filter. The basic formulation as well as its variants were introduced in Chapter 2. Even though the theoretical background is rather complex, due to its straight-forward application, the Kalman filter has quickly become one of the most widely applied state estimation algorithms. But the Kalman filter obeys special assumptions and special care must be taken in scenarios where these are not fulfilled. In order to compare the approach with our model, an extension of the Kalman filter for distributed computation, called Covariance Intersection [Julier et al., 1997], is introduced. This model alleviates some of the limitations of the distributed Kalman filter in practical applications.

Assume we have a system equipped with two sensors, both measuring the same physical quantity. Because technical characteristics of the two sensors will in reality never be equal, the accuracy of the first sensor will differ from the other. This can easily be expressed through their respective means (a, b) and covariances ($\underline{\mathrm{A}}$, $\underline{\mathrm{B}}$). If the two measurements are statistically independent, the signals can be fused through a convex combination, i.e. maximum likelihood estimation (MLE) [Chong et al., 2001]:

$$\underline{C} = (\underline{A}^{-1} + \underline{B}^{-1})^{-1} \tag{4.11}$$

$$c = \underline{C}(\underline{A}^{-1}a + \underline{B}^{-1}b) \tag{4.12}$$

Equation 4.11 represents the covariance of the two fused estimates, while equation 4.12 represents the fused mean. This result is only optimal, if the cross-covariance between the two sensor signals is zero. If the cross-covariance is exactly known, the signals can be fused with the BLUE algorithm [Chong et al., 2001], which in case of Gaussian signals provides the maximum a posteriori estimate (MAP). In our experiments we assume that the cross-covariance amongst the signals of our system is unknown. The assumption of unknown cross-covariance is mainly motivated by the fact that the correlation of multiple sensors, induced by changes in temperature and especially by vibrations of the robot, is usually very hard to acquire. Another motivation is the need for a consistent fusion algorithm that can be used in a distributed fusion architecture. This yielded the use of a more robust method, namely the Covariance Intersection method.

As shown in Chapter 2, various configurations of architectures for sensor fusion were developed, spanning from centralized, to decentralised, and distributed architectures. In order to cope with the drawbacks brought by standard centralised and decentralised designs, Covariance Intersection (CI) brings a relatively compact representation and computation mechanisms making it a suitable candidate for implementation on massively parallel hardware.

In the aforementioned scenario, the joint covariance of two sensor signals, represented by their respective mean and covariance $(a, \underline{A})$ and $(b, \underline{B})$, is given by:

$$\underline{C} = \begin{bmatrix} \underline{A} & \underline{X} \\ \underline{X}^T & \underline{B} \end{bmatrix} \tag{4.13}$$

The requirement to have precise knowledge of the cross-covariance $\underline{X}$ can be avoided by finding a covariance matrix $\underline{M}$, whose diagonal blocks $\underline{M}_A > \underline{A}$ and $\underline{M}_B > \underline{B}$, such that

$$\underline{M} \geq \begin{bmatrix} \underline{A} & \underline{X} \\ \underline{X}^T & \underline{B} \end{bmatrix} \tag{4.14}$$

$$C = \begin{bmatrix} A & X \\ X^T & B \end{bmatrix} \tag{4.15}$$

for any possible instantiation of the unknown cross-covariance $\underline{X}$.

This is illustrated in Figure 4.21a on the basis of the two sensor signal covariances (blue and red traces) and multiple results of MLEs, with different cross-covariances. As we can see, the MLE estimates always lie in the intersection of the two covariance ellipsoids. The CI algorithm finds the matrix $\underline{M}$ through the following equations:

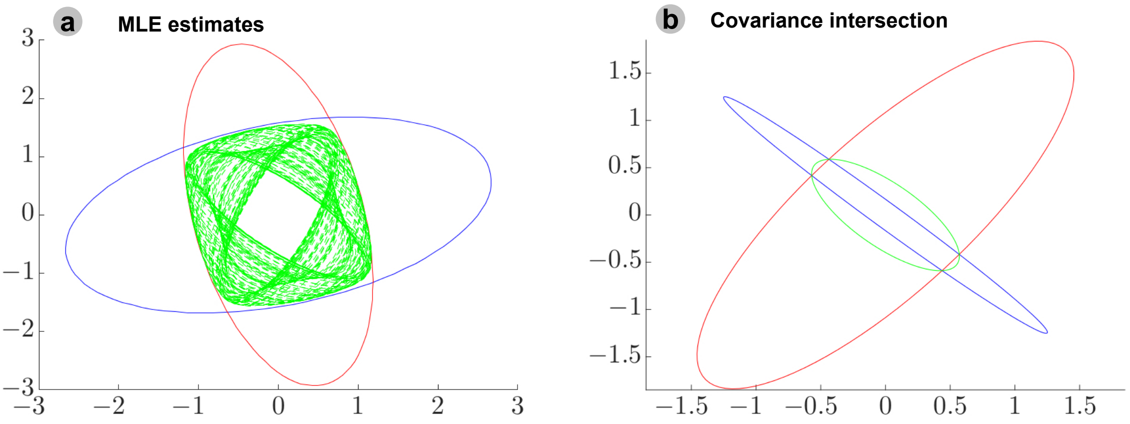


Fig. 4.21: Geometrical interpretation of the CI algorithm for a setup with two sensors with signal covariances (blue and red traces) 0.5 confidence ellipsoids of Gaussian distributions. a) MLE estimates (always lie in the intersection); b) Covariance intersection.

$$\underline{M}^{-1} = \omega \underline{A}^{-1} + (1 - \omega)\underline{B}^{-1} \tag{4.16}$$

$$m = \underline{M}(\omega \underline{A}^{-1} a + (1 - \omega)\underline{B}^{-1} b) \tag{4.17}$$

Comparing the above equations with 4.11 and 4.12 reveals the fact that CI performs a weighted maximum likelihood estimation. The parameter ω in 4.16 and 4.17 serves as a weighting between the two measurements and can be found with respect to some performance criterion on $\underline{M}$, i.e. the minimization of the trace or determinant of $\underline{M}$. Figure 4.21 b illustrates the function of CI (adapted from [Sequeira et al., 2009], the ellipsoids correspond to 0.5 confidence level of a Gaussian normal distribution). The covariances of the two sensors are:

$$\underline{A} = \begin{bmatrix} 0.8 & -0.7 \\ -0.7 & 0.8 \end{bmatrix}^2, \underline{B} = \begin{bmatrix} 0.3 & 1.2 \\ 1.2 & 1 \end{bmatrix}^2$$

Sensor A (blue) is highly accurate along the measurement direction but has a wide detection aperture, while Sensor B (red) has poor accuracy. The fused covariance (green) is calculated from Equation 4.16, with a minimal trace of M as a performance measure (resulting $\omega = 0.1557$). As can be seen in Figure 4.21b, the CI algorithm locks onto the more accurate sensor A, if the two signals are highly uncorrelated. This behaviour is very useful in the decentralized fusion network, where the estimates from other nodes should be fused weighted by their accuracy. Comparing CI to the MLE reveals how CI covariance encloses any MLE estimate and therefore serves as a conservative and robust fusion technique. This covariance can then be consistently used in the Kalman filter equations and distributed on the SpiNNaker hardware as shown in Figure 4.22. The CI algorithm finds the weight ω in Equation 4.16 by minimizing the trace of $\underline{M}$. The minimizer chosen

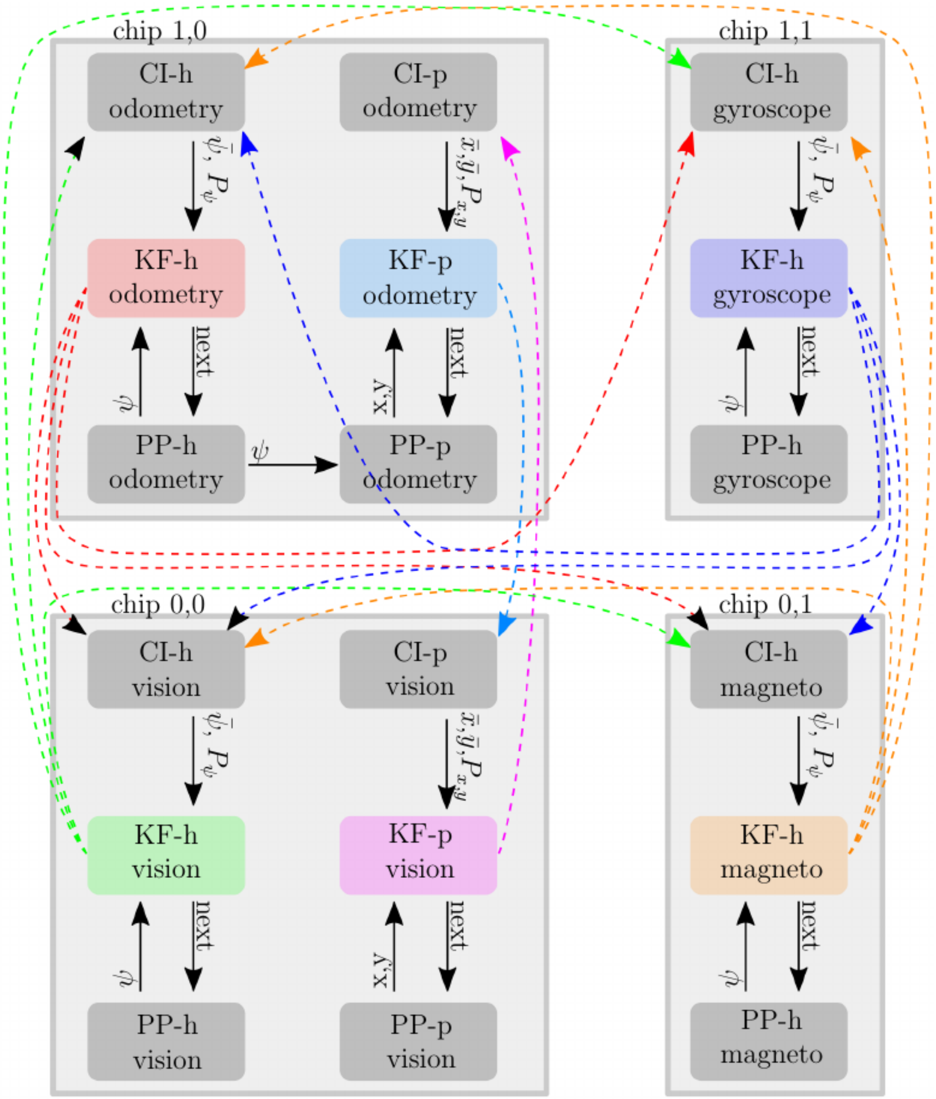


Fig. 4.22: Mapping the Kalman filter on hardware: Sample dispatching for egomotion estimation network. Covariance Intersection heading/position: CI-h/CI-p ; Kalman filter heading/position: KF-h/KF-p ; Preprocessor heading/position: PP-h/PP-p ; Communication lines: Fast (System-BUS, plain lines) / Multicast (Router, dashed lines).

for the implementation on SpiNNaker is Brent's method [Brent, 2013], which can optimize a function without using derivatives. The overall infrastructure of the implementation is illustrated in Figure 4.22. In this setup each Kalman filter unit receives a state estimate from its preprocessor and fuses it with the current belief from the covariance intersection unit. The fused quantity is then sent to one of the other CI units, which fuses this estimate obeying the CI equations. The communication among the cores is implemented via the System-BUS where possible, to ensure fast transfer of local estimates. The inter-chip communication on SpiNNaker is possible solely via multicast packages (broadcast among all cores via the router). Figure 4.23 shows the heading simulation results for the different units (Preprocessor heading: PP-h blue; Kalman filter heading: KF-h red dashed; Covariance Intersection heading: CI-h green). The odometry CI unit is obviously averaging the signal, while the other units alternate between preferring their own signal and the estimates received from the other nodes. It can be seen how all CI nodes tend to level onto a global heading estimate. The odometry heading estimation is in fact accumulating errors due to uncertainty (i.e. slipping wheels). Additionally, both odometry and gyroscope preprocessors are integrating their sensor signals which introduces another source of errors. This can be seen in Figure 4.23 a through the heading difference between the odometry/gyroscope preprocessors and the other nodes. This error accumulation becomes quite severe in the global position estimation.

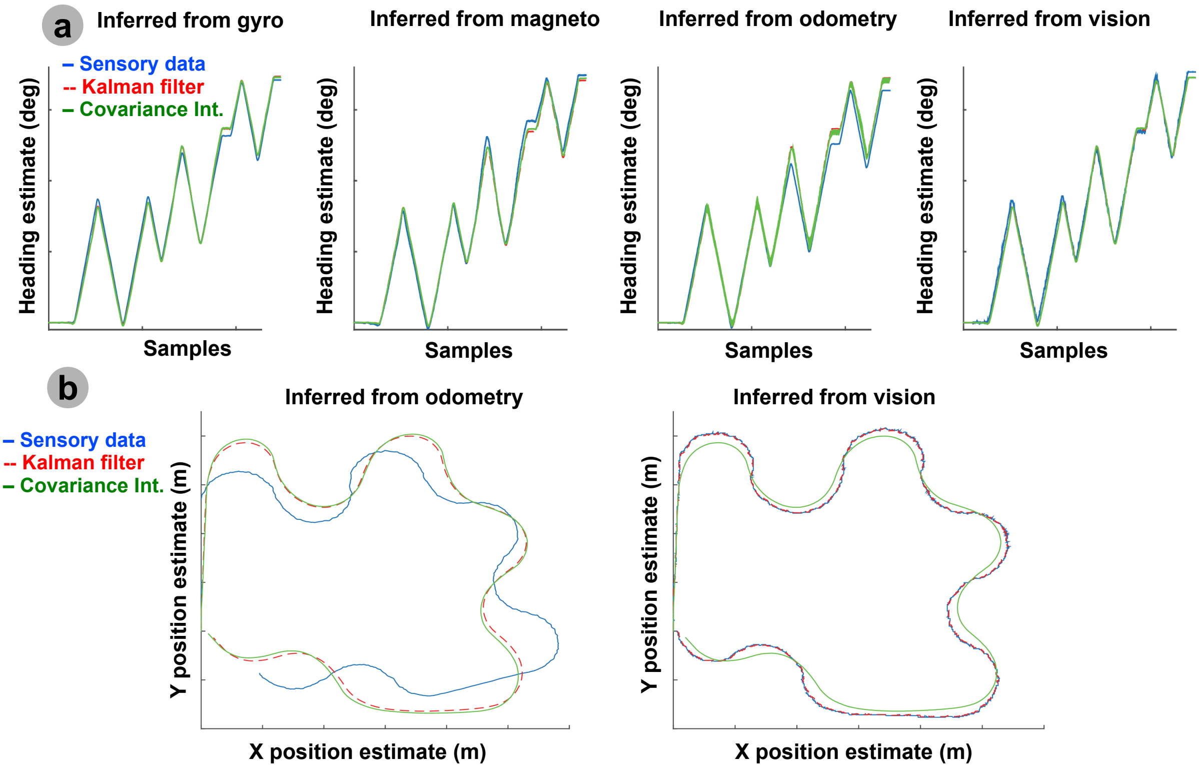


Fig. 4.23: Evaluating the SpiNNaker implementation of the CI Kalman filter model: a) Heading estimation; b) Position estimation.

Figure 4.23b shows the position estimation results of the vision and odometry node. The vision node position estimate can be expected to be very accurate, because it is

	Net M(PC)	Net C++(PC)	Net C(PC64)	Net C(SP)	KF C(SP)	CI C(SP)
F	3 GHz	3 GHz	1.8GHz	0.2 GHz	0.2 GHz	0.2 Ghz
N	1	1	64	30	4	18
M	4GB	4GB	16GB	128 MB	128 MB	128 MB
R	930 s	31 s	40s	49 s	19 ms	15 s
C	1796	6056	2500	1301	884	1070
E	No	No	Yes	Yes	Yes	Yes

Tab. 4.2: Runtime analysis (F - CPU frequency, N - number of cores, M - memory, R - run time, C - lines of code, E - extensible) for different implementations (i.e. MATLAB (M), C++, C) of the fusion mechanisms (i.e. Net-our model, KF-Kalman filter(part of CI processing), CI-covariance intersection computation) on different hardware architectures (i.e. PC-standard desktop, PC64-64core PC, SP-SpiNNaker).

calculated from a camera mounted on the moving robot and pointing at the ceiling. The odometry heading data already suffers from the aforementioned error accumulation. In addition, the errors introduced by the body-fixed to global position transformation cause further inaccuracy. Overall, the odometry position estimation can be expected to be quite inaccurate. This can be expressed by an increased measurement noise, which causes the fusion network to reject the odometry contributions continuously. The global position estimation is clearly dominated by the vision estimate, clearly visible in Figure 4.23b.

4.2.4 Evaluation and discussion

The parallelization experiments with both approaches (the neurally inspired model and distributed Kalman filter) considered sensory data acquired at 25 Hz over 198 s of robot operation. In order to evaluate the implementation and emphasize the advantage of parallelization, we considered various other sequential and parallel implementations on standard microcontroller and PC platforms.

There are several implementations of the proposed architecture. In order to benchmark the approaches, we recorded the data and fed it offline, although the numbers would hold in real-time operation. The first implementation is written in MATLAB. It served as reference code and server as early proof of concept and visualization purposes. Second, a C++ implementation, which worked as drop in for the MATLAB implementation through MATLAB's MEX interface, was considered. A C implementation was also tested on a 64-core server with the OpenMP API for parallelization tests. Finally, the architecture was implemented in C on the SpiNNaker hardware, leveraging the intrinsic parallelization capabilities of the network.

The reference MATLAB implementation requires 930 seconds to process the test data set. The drop in C++ implementation reduced the net runtime to 31 seconds. Tests of the C implementation on a 64-core server with OpenMP resulted in inferior results (40s) which is explained by the additional overhead of the multiprocessor structure and parallelization library. A global runtime evaluation is given in Table 4.2. In this section we analysed the parallelisation capabilities of our proposed model for multisensory fusion. In order to evaluate the performance of the model we also considered a distributed implementation of a state-of-art methods (i.e. DKF - distributed Kalman filter). The DKF was based on a zeroth-order Kalman filter combined with Covariance Intersection to fuse the heading and global position estimates from four different sensors available on-board the robot. The implementation exploits given hardware features, such as the asynchronous multicast communication among the cores and chips. The heading sensor fusion takes around 6s for 5000 samples acquired at 25 Hz, while the sensor fusion of position data takes around 15s. The whole experiment took the robot approximately 198 s, such that real-time sensor fusion is possible. A possible extension of this implementation, to allow it to detect inconsistencies in sensory data, is to use Covariance Union [Uhlmann, 2003].

Finally, our network's implementation on SpiNNaker hardware uses only half of the cores available which in turn allows for doubling the number of sensor inputs without any noticeable penalty in hardware requirements or runtime. The runtime depends on the slowest map type. In the current implementation base maps are the slowest map type, because they have the highest number of inter-map connections which results in an increased number of value update calculations. Focusing on the SpiNNaker implementation analysis, it is interesting to notice that our network is considerably slow, around 50s compared with both the standard Kalman filter (20ms) and the CI Kalman filter (15s). This is due to the fact that our network uses all available modalities, whereas the other state-of-the-art only use one (KF) or two (CI) modalities at once, having no explicit way to parallelize their processing. Another important aspect is the fact that the network is distributed among 30 cores, whereas in the KF there are only 4 cores used to sequentially execute the prediction and update steps for the recursive filtering. The distribution introduces a relatively high communication throughput which, for a high number of individually asynchronously updated sensory maps, slows down the overall system. This phenomena is also visible in the more complex CI KF implementation on SpiNNaker, where the execution time is comparable with our network, in the order of seconds for the 198s of robot operation. Finally, in order to conclude the analysis, we notice that the parallel SpiNNaker implementation of the network is slower than the PC implementations, for both standard desktop and multi-core platforms. This is mainly due to the fact the the PCs are equipped with high-frequency processors and considerable amount of memory but do not exploit the parallelization and extensibility capabilities of the network.

The parallel hardware which mirrors our network's architecture in combination with event-based programming form a viable solution for real-time application. Additionally, the low power consumption and form factor make it suitable for mobile applications.

4.3 Summary

Providing an instantiation of our framework introduced in Chapter 3, the current chapter focused on the core design aspects and advantages of the proposed approach for multisensory integration and its results in various real-world scenarios.

In a first scenario we investigated how our distributed network of units can be employed in a 2D motion estimation scenario for an omnidirectional wheeled mobile robot. After providing a review of state-of-the art approached for egomotion estimation we introduced the design stages and the rationale behind our approach. Distributing computation in a fully-connected network of units acquiring, representing, and processing sensory information through mutual exchange of information, the proposed model was able to combine available sensory cues into a global position and heading angle estimate. The dynamics of the model is based on the known physics of the sensors and cross-modal relations defining an internal model. Moreover, this internal model provides a prediction of the possible values a sensor can provide, which subsequently combined with sensory observations construct the system's belief of the perceived motion component. The inferred estimate is more precise that individual estimates and provided comparable results with state-of-the-art methods but with less parameterization effort.

Intrinsically distributable, our model proved a significant performance increase when executed on parallel hardware. We implemented the same model for 2D egomotion estimation on massively parallel hardware platforms (i.e. neuromorphic hardware and multicore PCs) and analysed the computational advantages and its capabilities in real-time scenarios against "traditionally sequential" approaches (e.g. Kalman filter). With simple partitioning and resource allocation, the network mapped easily on the available hardware providing great results, comparable with the state-of-the-art approaches (which required special treatment for the hardware mapping), on a variety of platforms with hardware and software enabled parallelism.

5 Formalizing a model for perceptual learning in multisensory fusion

Learning processes which take place during the development of a biological nervous system enable it to extract mappings between external stimuli and its internal state. Although the neural substrate is not well understood and formalised, the learning and development component can enhance adaptation and flexibility capabilities of today's technical systems. By alleviating the need for tedious design and parametrisation, the systems would learn the sensory data statistics and distribution. This subsequently allows for efficient representation and fast computation for environment understanding and interaction.

5.1 Probing neural models of perceptual learning and development

In this section we propose an extension of our framework, towards including biologically plausible learning mechanisms, for autonomous synthesis based on available sensory input. Rather than focusing on biologically precise descriptions of neural circuitry, we use relatively simple computational blocks, known to be widespread in the brain, and which are well formalised and understood. This approach is in line with our goal to keep the computational substrate simple enough, yet powerful and distributed, such that real-time operation, required by real-world scenarios, is still achieved.

Maintaining the generality and robustness shown previously by our model, we now redefine the problem. Without for prior analysis, and subsequent encoding the sensory relations in the model, our system is extended to learn them directly from the incoming sensory data streams. This capability leverages the applicability of the framework for those multisensory scenarios in which cross-sensory relations are complex, if at all possible to be expressed mathematically. Indeed, in some cases inter-sensory relations can be intrinsic in the data (e.g. temporal dependencies) and cannot be easily mathematically formalised in our relational framework.

In the current chapter we will focus on how can real-world sensory data be represented in neural substrate, and how a system with relatively limited initial knowledge can learn and synthesize an appropriate processing infrastructure efficiently, using only the available sensory streams. Moreover, we show that this kind of system is able, by using relatively simple computational mechanisms, to learn efficient (and sufficient) representations and make use of them for subsequent inference. Before delving into the implementation details of our model, we go back to neuroscience and provide an overview on current models of perceptual learning and development. This overview supports and motivates our approach, and at the same time, provides the framework in which we formulate our approach.

An interesting question to start with is, how cortical sensory maps emerge as functions of the parameters of the feature space or sensory modality they represent? This question marks the transition from our basic computational framework introduced in Chapter 3. As previously mentioned, local processing influences the state of the sensory representation, which influences back the processing due to the mutual interaction of the communicating areas encoding a specific sensory modality. In this context, [Cimponeriu et al., 2000] developed a dynamic model of the visual cortex based on experimental data for extracting the spatial structure of orientation and ocular dominance cortical maps. Their results showed that the ordering, and subsequently the connectivity, of cortical maps (during development) is controlled by the parameters of the feature space they represent. These results are consistent with one of our design principles, according to which each area tries to bring the encoded representation towards a state compatible with related areas connected with it.

Trying to provide a more generic view on the processes underlying perceptual learning and interpretation, [von der Malsburg, 1999] addressed three fundamental questions: how are the brain states interpreted as representations of actual situations; what are the organisation mechanisms of these states; what permanent information storage mechanism is used by the brain; and what are the mechanisms of learning? Trying to answer these questions, the study provided an analysis on sensory cue binding, focusing on the temporal scales, and time influence over the representation formation dynamics. The core observation refers to the way the correlations in the temporal signal structure arise and their influence in sensory binding. Ultimately, the purpose of temporal binding is to express significant relationships between data items, for example of causal or spatial nature, and the physical interactions establishing such relations (represented by signal correlations).

In a more recent study, [Michler et al., 2009] proposed that spatial and temporal stimuli correlations can be exploited for learning invariant representations. Spatiotemporal sensory correlations in the sensory streams were mapped from different views of objects onto a topographic representation, showing that cortical topographic maps have a functional relevance. The working hypothesis, of interest also for our design, is that correlations in input sequences can shape the neighbourhood relationships in the learned representation.

In a more broad perspective, detached from the low-level local representations, [Quiton et al., 2011] provided a new framework considering competition within the brain and interactions between assemblies of neurons. The proposed model adopted a distributed approach to cognition, and focused on a mesoscopic description scale in which the cortex is decomposed in cortical maps, themselves made of cortical columns. Using a sparse modelling scheme, contrary to traditional matrix implementations, there are no more dependencies on the number of dimensions. Hence the high performance for direct handling of high dimensional input spaces typically describing multisensory processing scenarios. Furthermore, the model predicted that sensory features and relationships defining multimodal representations may be highly dependent on the considered concept and the sensory context. This aspect is relevant in our framework as it is defining the constraints the model extracts from the incoming streams and subsequently uses to attain consistent distributed representations.

Each sensory contribution brings inherent constraints in the global representation. Due to their intrinsic coupling (e.g. through the motion of the system/body) sensory cues are correlated such that these constraints quantify how this correlation is realised. As previously mentioned, correlation is marked either by an explicit mathematical formulation or lives hidden in the data. In this context, correlation can be considered as a mathematical basis for learning and formulated as an optimization problem [Chen et al., 2007]. The learning system should have an appropriate objective criterion (e.g. function) upon which an optimisation (i.e. minimisation / maximisation) procedure is applied to compute good parameters that constitute the minimum (or maximum) of the objective criterion. Indeed, viewed in this light, our framework is basically enforcing consensus by finding solutions to the constraints imposed by contributing sensors. The optimisation procedure relies on dynamically ”pulling” each local representation to a coherent representation, which is locally stable (given the sensory input) and globally coherent (given the cross-sensory correlations). In neurobiological systems, this learning process is described by the synaptic adaptation process, towards obtaining optimal synaptic weights describing the connectivity pattern encoding a stable representation. We abstract from this principle and develop a computationally efficient model directly transferable to technical implementations.

5.2 From neural models to implementation

In the current section we introduce our synthesis model for learning sensory correlations. Learned correlations are used for subsequent multisensory fusion. Given relatively simple and well understood neurally inspired computational mechanisms, we design a model for sensory correlation extraction. Consistent with our designed paradigm, of distributing computation and representation amongst a network of computing units, we extend the sensory data representation and the cross-sensory relation encoding. Instead of using single point estimates (i.e. scalar values) to represent real-world sensory readings, we switch to a sparse representation, encoding sensory values into an activity profile of a number of topologically organised neural processing units (i.e. neurons). This representation contributes to creating more precise ”local knowledge” and simplifies computation, as all sensory cues will be ”re-coded” in the same representation space. Local processing upon local representations of the sensory quantities ensures that consistent local states (i.e. representations) converge to coherent global representations. This perspective is consistent with models and experimental evidence from human developmental science, and provides the new perspective on the synthesis of adaptive and autonomous technical systems brought by our work. The following subsections provide a formal description of our approach and a detailed analysis of its capabilities and scalability.

5.2.1 Introducing the basic model

Starting from Hebb's original postulate of learning in neurobiogical systems, various learning models were developed, spanning a wide range of sensory, motor, perceptual, and cognitive functions, including associative memory, coincidence detection, sound localization and segregation in the auditory system, topographic map formation in the visual system, feature binding for sensory perception, as well as sensorimotor control in the cerebellum.

All these models went far beyond the Hebbian postulate, included all of the three major machine learning paradigms: unsupervised, supervised and reinforcement learning, and were widely used in artificial adaptive systems capable to imitate adaptive functions of the brain.

Much attention was given to unsupervised Hebb-type learning rules, especially competitive learning, BCM learning, PCA learning, and Boltzmann learning [Chen et al., 2007], due to their innate capability to extract knowledge from the data without a "teacher" or an error signal. Furthermore, in order to take into account the intimate information structure of the data and its statistics, unsupervised information-theoretic learning methods were developed: Linsker's rule, Imax rule, BSS, ICA, and SFA [Chen et al., 2007].

For the cases in which a quantification (e.g. error criteria) of how good the learning process is, supervised learning mechanisms were developed, such as the perceptron learning rule and the LMS. These widely employed methods sometimes link to or root in traditional signal processing mechanisms.

Finally, reinforcement learning, with its variants, temporal Hebbian learning, TD learning, or even models which combine reinforcement learning and Hebbian learning, provided more insight in reward driven or reward modulated processing in the brain, marking the transition to new computational paradigms.

All of the above mentioned methods are linked to Hebbian plasticity rule and share common roots with correlation-based learning principles, having also an underlying biological motivation.

As mentioned earlier, we propose to use a sparse representation of the sensory streams in order to extract the underlying statistics and probability distribution of the sensory data. More explicitly, we use a competitive learning rule. As an important ingredient of self-organising systems, competitive learning's goal is to tune a certain number of parameter vectors (i.e. synaptic weights) in a possibly high-dimensional space. The distribution of these vectors should reflect the probability distribution of the input data. Depending on the type of activation function they use in their dynamics, competitive learning methods can be categorised as either "hard competition" (or WTA, Winner-Take-All), where each input data sample determines the adaptation of only one winning representation (i.e. the closest to the input data), or "soft competition", for which each data sample is represented with a certain probability in the system, and the local adaptation will be proportional with this probability.

The adaptive development and shaping of functional organisation in cortical areas seems to depend strongly on the available sensory inputs, which gradually sharpen their response, given the constraints imposed by the cross-sensory relations. Following this principle, we use one of the most popular forms of competitive learning, namely the Self-Organising-Maps (or Kohonen network). The underlying self-organization mechanism can be viewed as a form of Hebbian learning, in a network with competitive interactions with a decay term guaranteeing normalization. In its basic formulation the SOM is composed of a lattice (1D or 2D) of neural processing units (i.e. neurons), and each neuron has a preferred representation (e.g. 1D, 2D, ..., nD synaptic weight vector) of a 1D, 2D, ..., nD sensory feature. In order to learn, the SOM is fed with sensory data samples such that for each sample the unit with the closest representation to the input is chosen as winner. Subsequently, the winner (in "hard competition") or the winner and a predefined vicinity (in "soft competition") are "pulled" towards the input sample. This process assumes that the weight vector of the winner (and vicinity, if any) is adapted towards better representing the input sample. Iterating through available input data, the algorithm is able to separately represent features in different parts of the network and still keep the topological organisation. Close features in the input sensory space will be closely represented in the network after a sufficient number of presentations of the input dataset.

In our model samples from each input sensory modality are fed into a SOM. These networks are responsible for locally extracting the statistics of the incoming data, depicted in the simple example in Figure 5.1a, and encoding sensory samples in a distributed activity pattern, as shown in Figure 5.1b. This activity pattern is generated such that the closest preferred value of a neuron to the input sample will be strongly activated and will decay, proportionally with distance, for neighbouring units. Figure 5.2 provides a detailed depiction of processing stages which take place when sensory input samples are presented to the network. Using the SOM distributed representation, the model learns the boundaries of

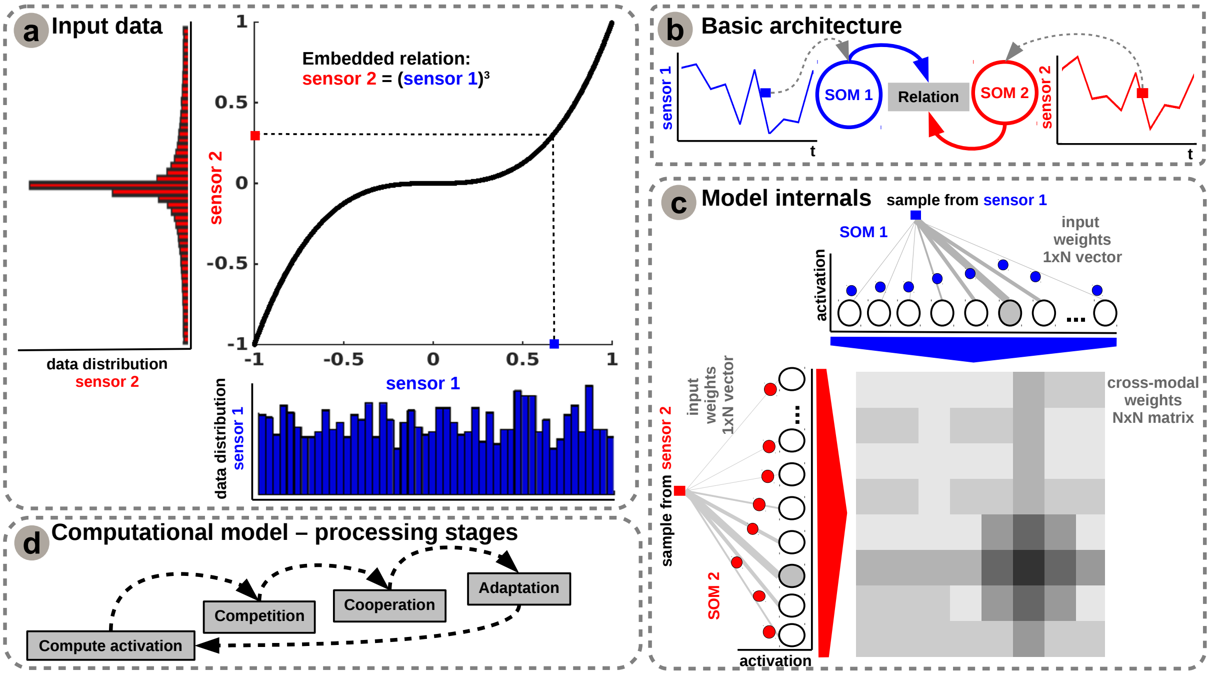


Fig. 5.1: Model architecture instantiated for a simple example: a) Input data resembling a nonlinear relation (3rd order power-law) and input data distributions; b) Basic model architecture; c) Processing stages of the model.

the input data, such that, after relaxation, the SOMs provide a topological representation of the input space. We extend the basic SOM in such a way that each neuron not only specialises in representing a certain (preferred) value in the input space, but also learns its own sensitivity (i.e. tuning curve shape). Given an input sample, $s^p(k)$ at time step k, the network follows the processing stages depicted in Figure 5.1d and explicitly presented in Figure 5.2. For each $i-th$ neuron in the $p-th$ input SOM, with the preferred value $w^p_{in,i}$ and $\xi^p_i(k)$ tuning curve size, the sensory elicited activation is given by

$$a^p_i(k) = \frac{1}{\sqrt{2\pi}\xi^p_i(k)} e^{\frac{-(s^p(k)-w^p_{in,i}(k))^2}{2\xi^p_i(k)^2}}. \tag{5.1}$$

The winner neuron of the $p-th$ population, $b^p(k)$, is the one which elicits the highest

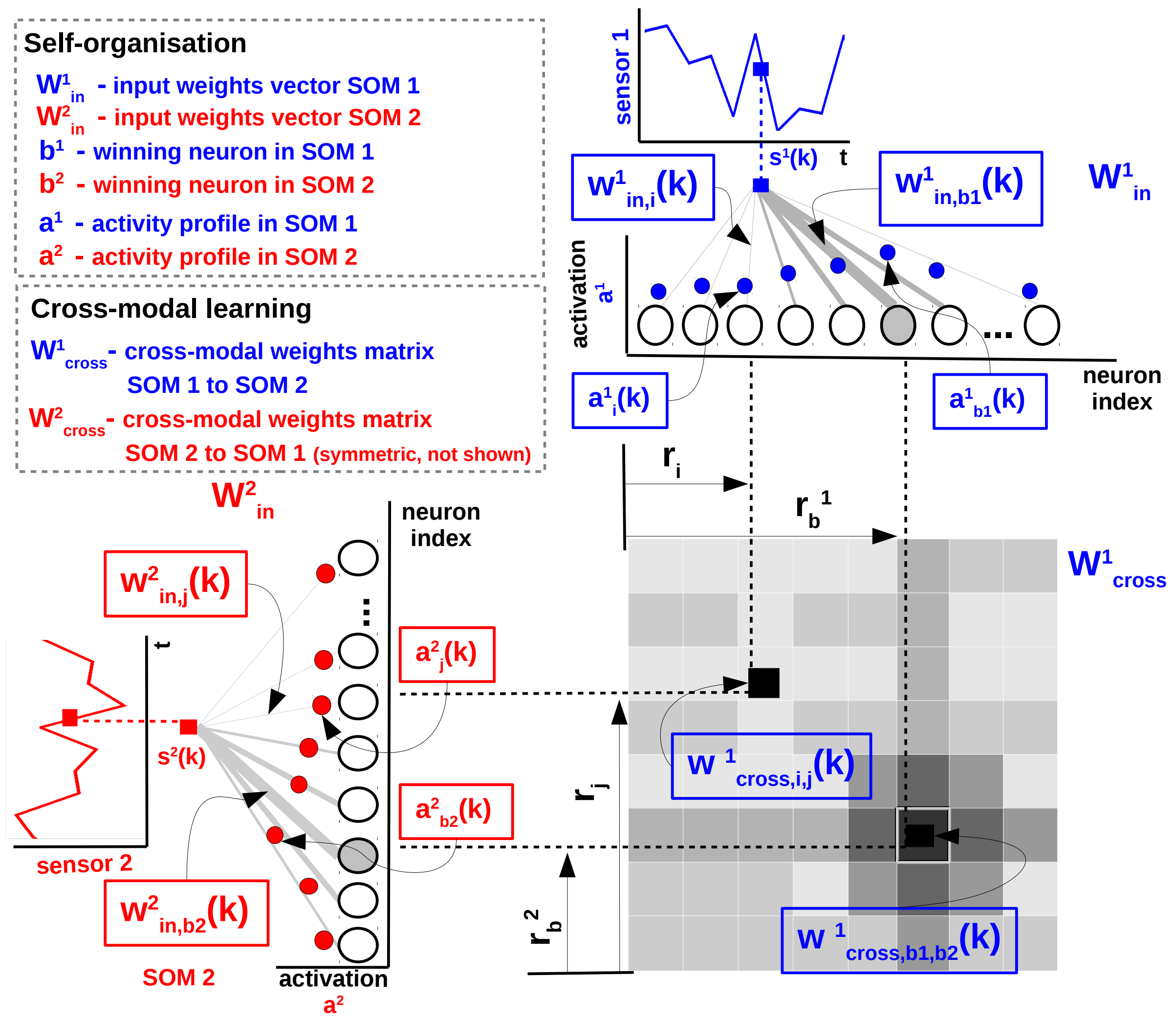

Self-organisation
W¹in - input weights vector SOM 1
W²in - input weights vector SOM 2
b¹ - winning neuron in SOM 1
b² - winning neuron in SOM 2
a¹ - activity profile in SOM 1
a² - activity profile in SOM 2
Cross-modal learning
W¹cross - cross-modal weights matrix SOM 1 to SOM 2
W²cross - cross-modal weights matrix SOM 2 to SOM 1 (symmetric, not shown)
sensor 1
s¹(k)
t
w¹in,i(k)
w¹in,b1(k)
W¹in
activation a¹
neuron index
a¹i(k)
a¹b1(k)
ri
r¹b
W¹cross
W²in
neuron index
w²in,j(k)
a²j(k)
t
s²(k)
sensor 2
a²b2(k)
w²in,b2(k)
SOM 2
activation a²
rj
r²b
w¹cross,i,j(k)
w¹cross,b1,b2(k)

Fig. 5.2: Detailed architecture of the model and processing stages.

activation given the sensory input at time k

$$b^p(k) = \underset{i}{argmax}\ \underline{a}^p(k). \tag{5.2}$$

During self-organisation, at the input level, competition for highest activation is followed by cooperation in representing the input space (second and third step in Figure 5.1d). Given the winner neuron, $b^p(k)$, the interaction kernel,

$$h^p_{b,i}(k) = e^{\frac{-||r_i - r_b||^2}{2\sigma(k)^2}}. \tag{5.3}$$

allows neighbouring cells (found at position r_i in the network) to precisely represent the sensory input sample given their location in the neighbourhood $\sigma(k)$. The interaction kernel in Equation 5.3, ensures that specific neurons in the network specialise on different areas in the sensory space, such that the input weights (i.e. preferred values) of the neurons are pulled closer to the input sample,

$$\Delta w^p_{in,i}(k) = \alpha(k) h^p_{b,i}(k)(s^p(k) - w^p_{in,i}(k)). \tag{5.4}$$

This corresponds to the adaptation stage in Figure 5.1d and ends with updating the tuning curves. Each neuron's tuning curve is modulated by the spatial location of the neuron, the distance to the input sample, the interaction kernel size, and a decaying learning rate $\alpha(k)$,

$$\Delta\xi_i^p(k) = \alpha(k)h_{b,i}^p(k)((s^p(k) - w_{in,i}^p(k))^2 - \xi_i^p(k)^2). \tag{5.5}$$

If we consider learned tuning curves shapes for 5 neurons in the input SOMs (i.e. neurons 1, 6, 13, 40, 45), depicted in Figure 5.3, we notice that higher input probability distributions are represented by dense and sharp tuning curves. Whereas lower or uniform probability distributions are represented by more sparse and wide tuning curves. Using this mechanism, the network optimally allocates resources (i.e. neurons): a higher amount to areas in the input space, which need a finer representation; and a lower amount for more coarsely represented areas. This feature, emerging from the model, is consistent with recent work on optimal sensory encoding in neural populations [Ganguli et al., 2014]. This claims that, in order to maximise the information extracted from the sensory streams, the prior distribution of sensory data must be embedded in the neural representation.

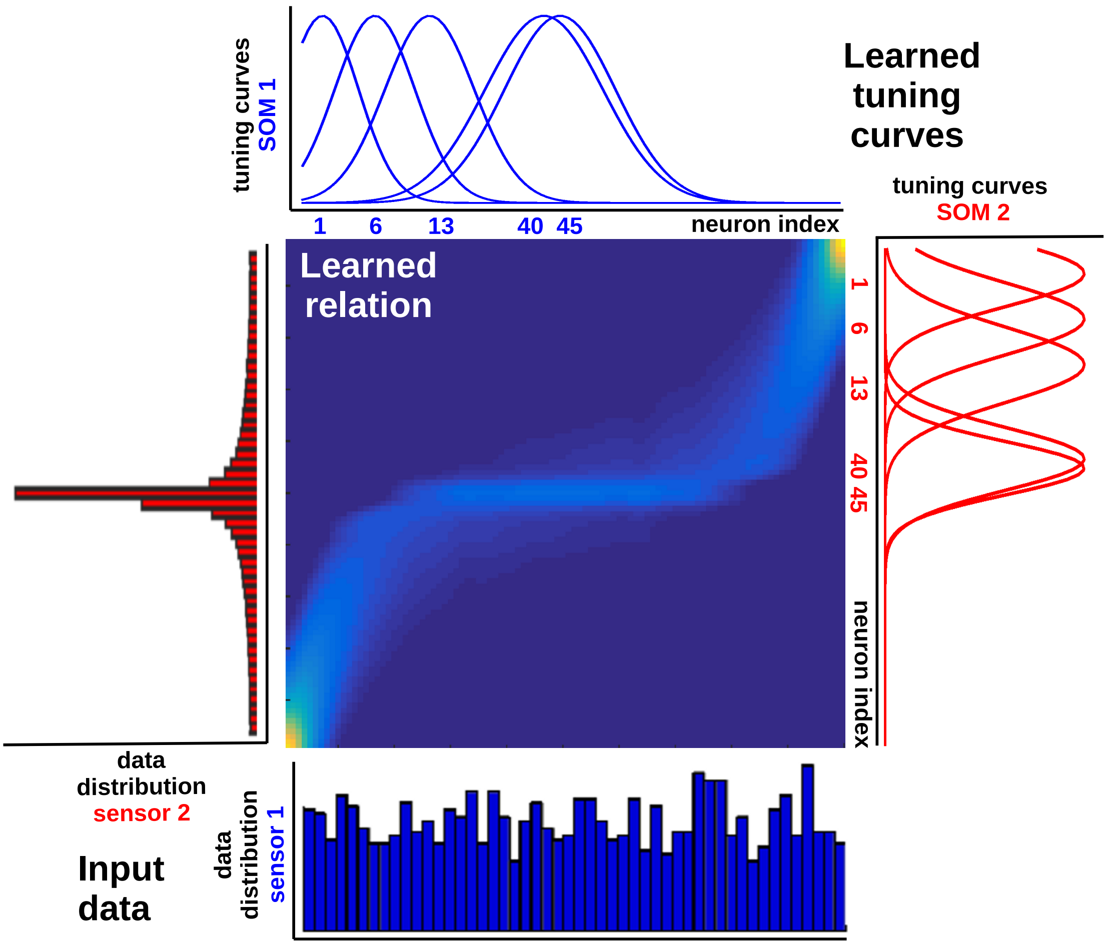


Fig. 5.3: Extracted sensory relation and data statistics using the proposed model

In order to link the two representation constructed in the two SOMs, we use a variant of the Hebbian learning rule. This rule is consistent with Hebb's postulate that the strength of the synaptic connection between two neurons, A and B, should increase in proportion to the degree to which neuron A repeatedly takes part in the firing of neuron B. However, this rule only allows for synaptic strengthening. In order to satisfy biological constraints, there must also be some mechanism for synaptic weakening. We decided to use a balancing learning rule, namely the covariance rule.

As previously mentioned, we have to couple the sensory representations in the two SOMs such that we can extract the correlation amongst them. In order to achieve that we use a Hebbian linkage, which consists of a fully connected matrix of synaptic connections between neurons in each input SOM. The Hebbian learning process is responsible for extracting the co-activation pattern between the input layers (i.e. SOMs), as shown in Figure 5.1b, and for eventually encoding the learned relation between the sensors. Hebbian connection weights, $w^p_{cross,i,j}$, between neurons i, j in each of the input SOM populations are updated using

$$\Delta w^p_{cross,i,j}(k) = \eta(k)(a^p_i(k) - \overline{a}^p_i(k))(a^q_j(k) - \overline{a}^q_j(k)), \tag{5.6}$$

where

$$\overline{a}^p_i(k) = (1 - \beta(k))\overline{a}^p_i(k-1) + \beta(k)a^p_i(k). \tag{5.7}$$

In order to prevent unlimited weight growth, we use a modified Hebbian learning rule (i.e. covariance rule, Equation 5.6) to allow for weight decreases when neurons fire asynchronously. The proposed mechanism uses a time average of pre- and postsynaptic activities, $\overline{a}^p_i(k)$, defined in Equation 5.7, such that when neurons fire synchronously in a correlated manner their connection strengths increase, whereas if their firing patterns are anticorrelated the weights decrease.

Self-organisation and correlation learning processes evolve simultaneously, such that both representation and correlation pattern are continuously refined. Moreover, the timescales of the two processes align, such that once the representations are learned in the SOMs the correlation pattern in the Hebbian connection matrix becomes sharper.

In the initial example we consider a set of values drawn from a uniform random distribution (i.e. sensor 1), Figure 5.1a, to which we apply a power-law, and we compute a second input (i.e. sensor 2) drawn from a Gaussian distribution. The network is fed with random pairs from the two datasets. After learning, the Hebbian connectivity matrix encodes the input data relation, as shown in Figure 5.3. Moreover, the tuning curves encode the input data distribution: narrower spaced for higher probability distributions and widely spaced for lower (or uniform) distributions of the input data. Learning and allocating overlapping tuning curves shapes allows the network to tile the input space representing more highly probable sensory data by a higher tuning curve density for highest probability distribution and low density for low probability distribution. The tiling properties are maintained as the tuning curves cover the entire representation space.

This learning scheme extends [Axenie et al., 2014], in which given various sensory inputs and simple relations defining inter-sensory dependencies, the model infers a precise estimate of the perceived motion. Now, by alleviating the need to explicitly encode sensory relations in the network dynamics, we introduce a model providing flexible and robust multisensory fusion, without prior modelling assumptions, and using only the intrinsic sensory correlation pattern. In this framework we see the learning and development process as a sharpening process, during which sensory projections and correlations are refined by experience. Using a sparse, population encoded representation of sensory data instead of point estimates, the model is able to embed sensory statistics and reliability such that the model is able to extract the underlying data correlations. The learning mechanisms use relatively simple cooperative and competitive circuitry, which are well explained and understood and provide an effective mechanism to learn patterns of co-activation in distributed representations of sensory data. Our model finds itself at the border between engineering and neuroscience, providing a basic structure for learning from real-world sensory data using relatively simple biologically plausible mechanisms. Various methods, ranging from neural circuitry implementations to statistical correlation analysis, have been developed to extract correlational structure in sensory data. In order to frame our work, as well as defining its advantages, we provide an overview on some selected approaches close to our work.

Other approaches for learning sensory relations

Related work in [Cook, Jug et al., 2010] used a combination of simple biologically plausible mechanisms, like WTA circuitry, Hebbian learning, and homeostatic activity regulation, to extract relations in artificially generated sensory data. The model is depicted in Figure 5.4, while the network dynamics and its evolution, for 1000 training epochs, is depicted in Figure 5.5, where the model extracted the nonlinear (i.e. power-law) relation. The structure could easily accommodate new tasks using the same substrate (i.e same network, only input differed). Real-world values presented to the network were encoded in population code representations. Each input to the network had an associated population coded representation and dynamics was provided by a continuous WTA circuit with hard-coded connectivity, Figure 5.5a. This approach is similar to our approach in terms of the sparse representation used to encode sensory values. The difference resides in the fact that in our

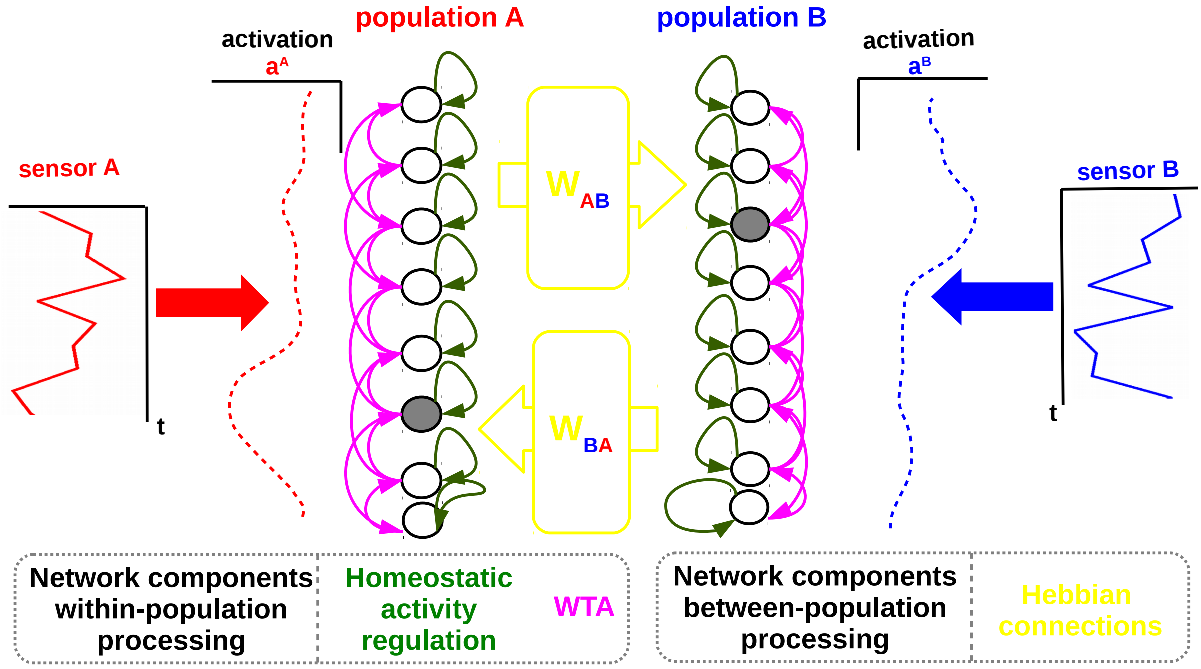

population A
population B
activation
a^A
activation
a^B
sensor A
sensor B
t
t
W_{AB}
W_{BA}
Network components within-population processing
Homeostatic activity regulation
WTA
Network components between-population processing
Hebbian connections

Fig. 5.4: Other approaches for learning sensory relations: [Cook, Jug et al., 2010]

model the input population (i.e. SOM) connectivity is learned. Using this capability, our model is capable of learning the input data bounds and distribution directly from the input data, without any prior information or fixed connectivity. Furthermore, the dynamics between each population coded input was performed through plastic Hebbian connections. Starting from a random connectivity pattern, the matrix finally encoded the functional relation between the variables which it connected, Figure 5.5b. The Hebbian linkage used between populations is the correlation detection mechanism used also in our model, although in our formulation we adjusted the learning rule to accommodate both the increase and decrease of the connection weights. Finally, the model also considered neuron level dynamics, represented through a homeostatic activity regulation process. This mechanism is responsible to ensure that all neurons in each population are used and that each neuron is used in moderation. In our model, due to the local competition and cooperation

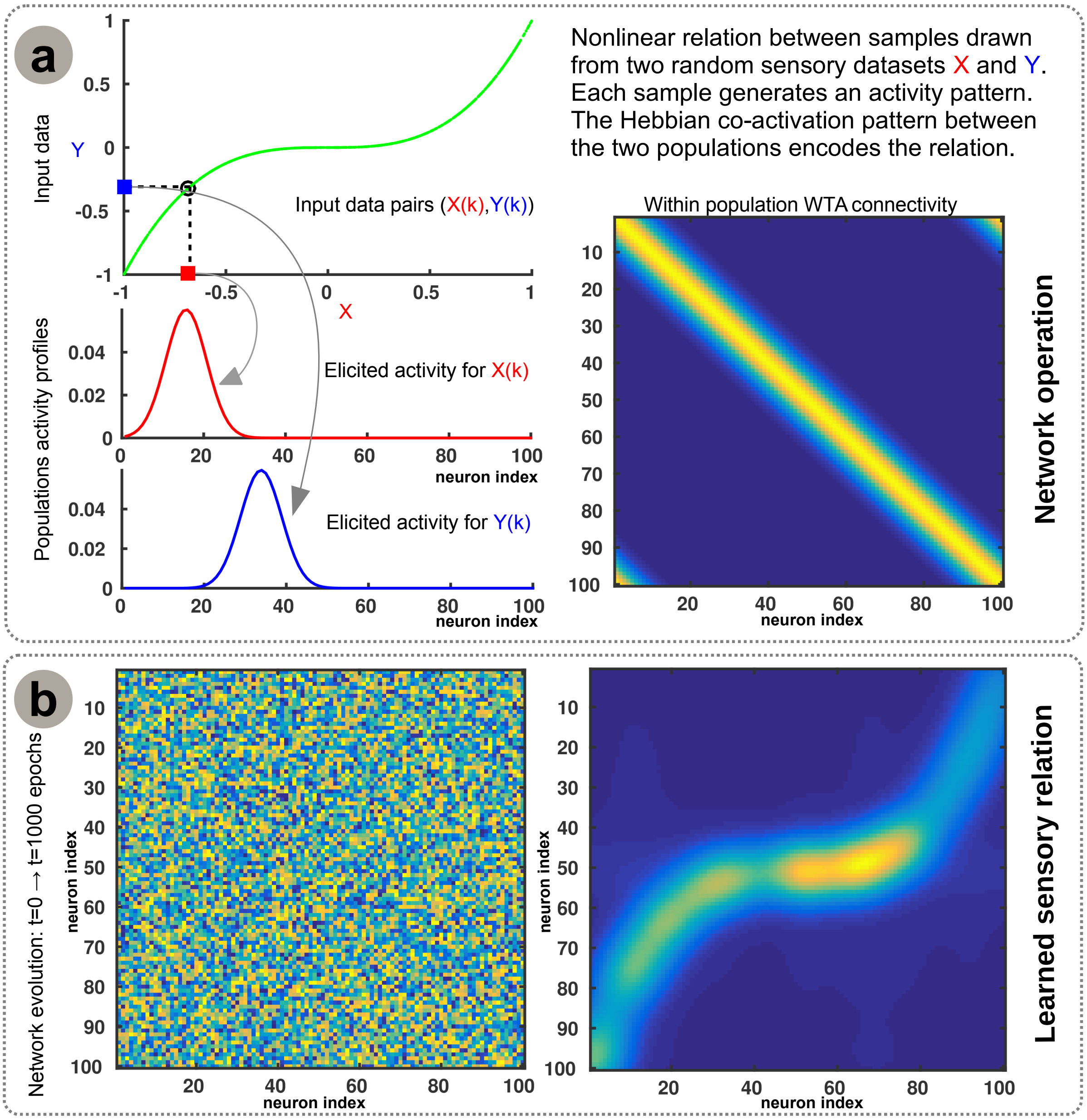


Fig. 5.5: Analysis of other approaches for learning sensory relations: [Cook, Jug et al., 2010]. a) Network operation; b) Learned sensory relation.

(i.e. SOM mechanisms), each neuron in the model is able to provide a contribution which will not saturate the overall activity pattern in the network while keeping a topological organisation.

Interestingly, the proposed model was able to exhibit different behaviours depending on the input type. After learning, the model was able to infer missing quantities given the learned relations and available sensors (i.e. inference task). Moreover, due to recurrent connectivity, the sensory representations were continuously refined and cleaned-up to precisely extract the real-world encoded variable (i.e. de-noising task). Given that the network had an all-to-all connectivity between the population encoding the inputs, the dynamics allowed the adjustment of the population codes to be consistent with each other (i.e. cue integration task). Finally, the network was able to discriminate and choose between alternative population codes when facing with consistent data.

Using a different neurally inspired substrate, [Weber et al., 2007] combined competition and cooperation in a self-organizing network of processing units to extract coordinate transformations in a robotic visual object localization scenario. More precisely, the model used simple, biologically motivated operations, in which co-activated units from population coded representations self-organized after learning in a topological map, solving the reference frame transformation between the inputs (mapping function). The basic network architecture is depicted in Figure 5.6. The representation used a n-uple based population code representation of the functional relationship encoding the reference system mapping with Hebbian links between connected populations. Similar to our model the proposed approach extended the SOM network by using sigma-pi units (i.e. weighted sum of products). The connection weight between this type of processing units is effective, if unit i of one input population is coactivated with unit j of the other input population, implementing a logical AND relation. Inspired by sensorimotor transformations in the prefrontal cortex, the algorithm produced invariant representations and a topographic map representation of the visual scene guiding a robot's behaviour.

Going away from biological inspiration, [Mandal et al., 2013] used a nonlinear canonical correlation analysis method, termed alpha-beta divergence correlation analysis (ABCA), to extract relations between sets of multidimensional random variables. The main idea in canonical correlation analysis is to first determine linear combinations of the two random variables (called canonical variables/variants) such that the correlation between the canonical variables is the highest amongst all such linear combinations. As traditional CCA is only able to extract linear relations between two sets of multi-dimensional random variable, the proposed model comes as an extension to extract nonlinear relations, with the requirement that relations are expressed as smooth functions and can have a moderate amount of additive random noise on the mapping. The model employed a probabilistic method based on nonlinear correlation analysis using a more flexible metric (i.e. divergence / distance) than typical canonical correlation analysis. A simple diagram describing the model's functionality is given in Figure 5.8. From observations of two random variables, $\underline{x}$ and $\underline{y}$ the method was able to extract the two vector directions (i.e. weight vectors $\underline{w}_x$ and $\underline{w}_y$) such that the divergence (i.e. distance metric) between the joint distribution $(\underline{w}_x^t\underline{x}, \underline{w}_y^t\underline{y})$ and the product of marginal probabilities of the variables is maximized. Assuming that there

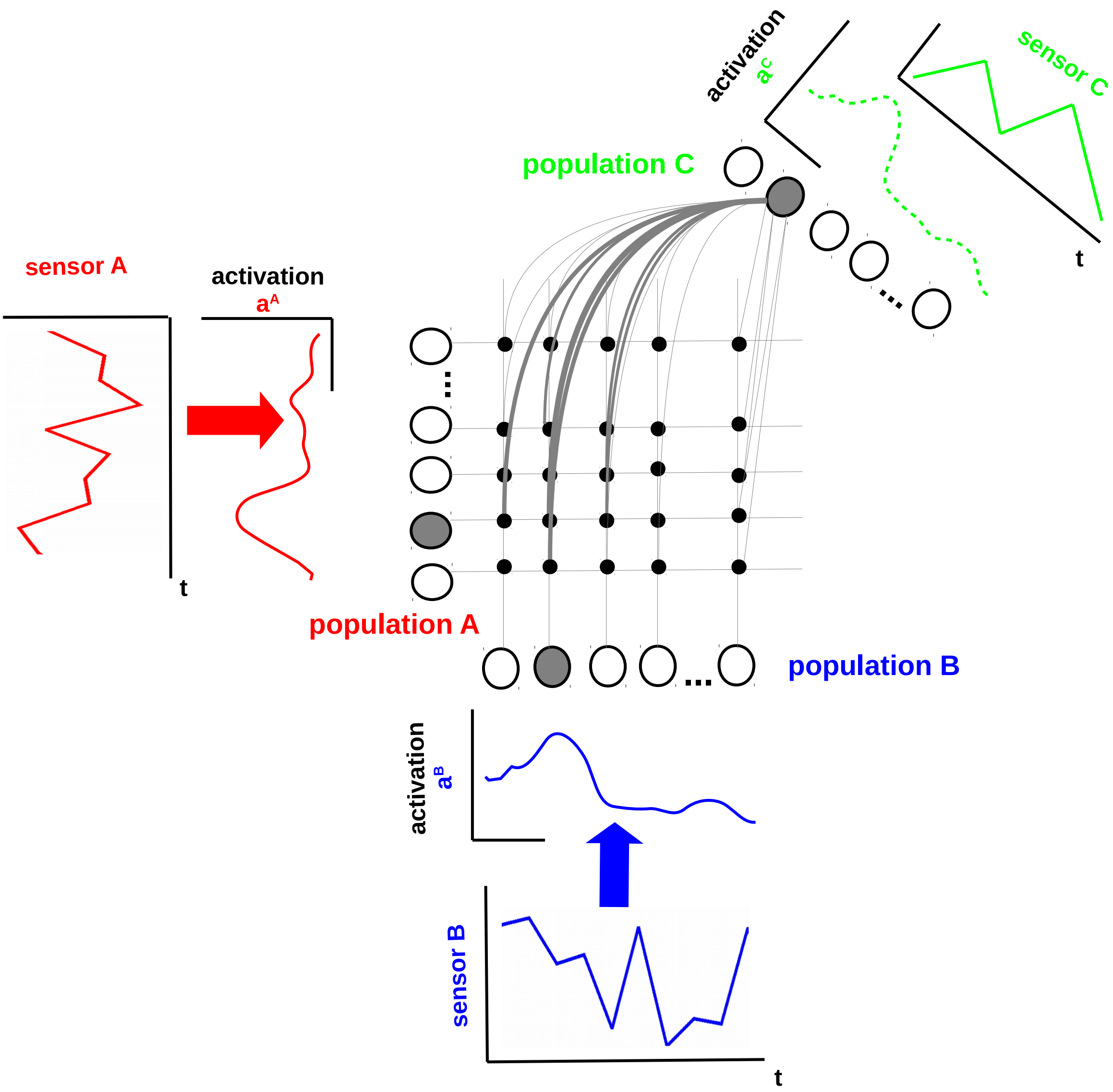


Fig. 5.6: Other approaches for learning sensory relations: [Weber et al., 2007]

is a hidden linear or nonlinear functional relation ψ of the following type:

$$\underline{w}_y^t \underline{y} = \psi(\underline{w}_x^t \underline{x}) + \epsilon, \tag{5.8}$$

the model finds $\underline{w}_x$ and $\underline{w}_y$ from observed x and y such that the canonical correlation coefficient

$$p^* = maxCorr(\underline{w}_y \underline{y}, \psi(\underline{w}_x^t \underline{x})), \tag{5.9}$$

provides the maximum possible correlation between $\underline{w}_y$ and any function of $\underline{w}_x$.

In order to illustrate the capabilities of the model we implemented a two-dimensional scenario in which each component of the input variables (i.e. $\underline{x}$ and $\underline{y}$) obeys a hidden nonlinear relation given by $\underline{y}_1 = \underline{x}_1^2$ and $\underline{y}_2 = \underline{x}_2^3$ as shown in Figure 5.9a. Using 500 pairs of randomly generated values for each variable, the algorithm provided high correlation values, 0.997 for the first dimension, and 0.996 for second dimension of the input variables. In Figure 5.9 we observe that ABCA extracts the relations quite accurately although the

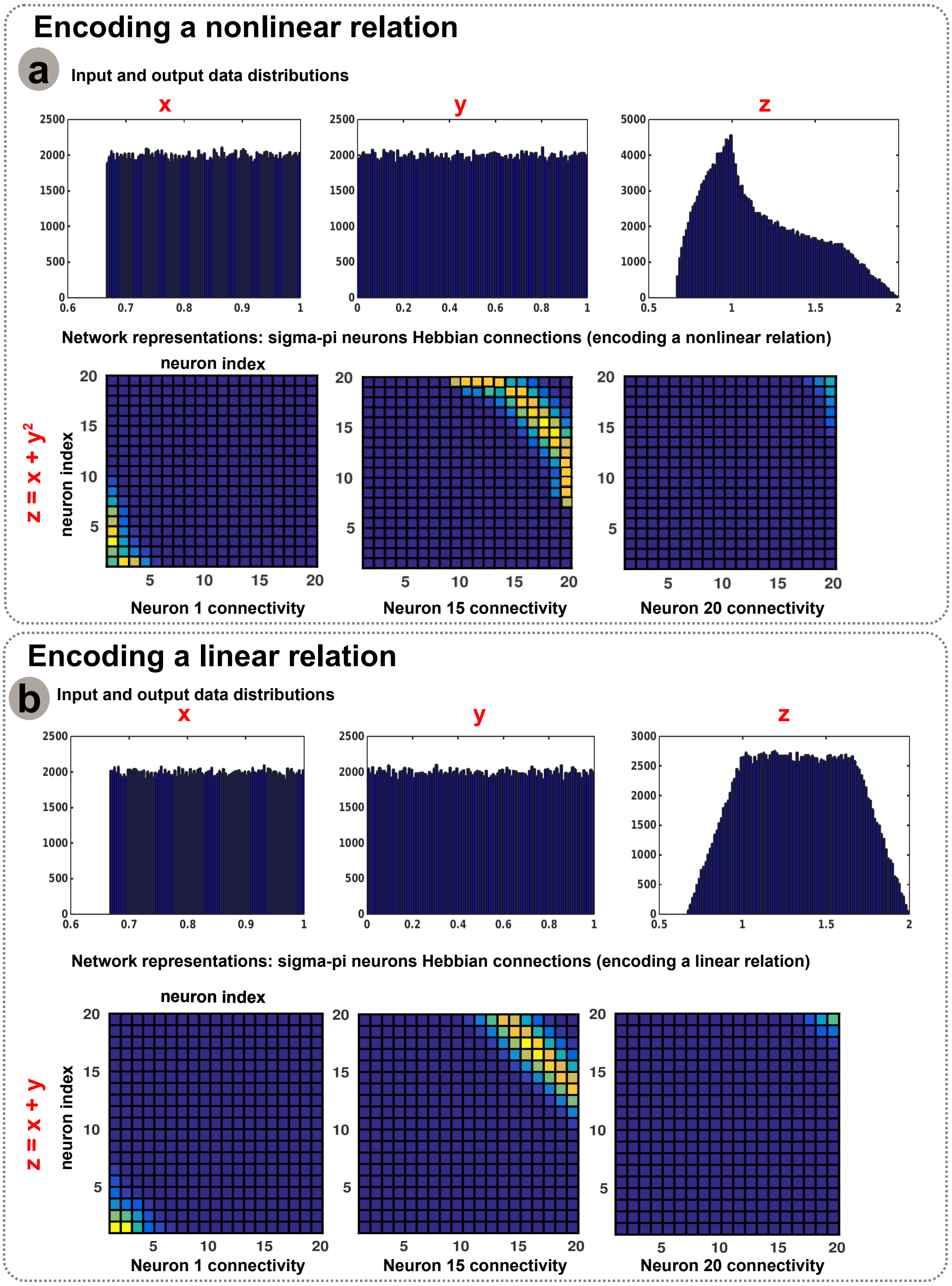


Fig. 5.7: Analysis of other approaches for learning sensory relations: [Weber et al., 2007]. a) Network analysis: encoding process and input data (nonlinear) distributions and learned nonlinear relation encoded in the network; b) Network analysis: encoding process and input data (linear) distributions and learned linear relation encoded in the network;

scale and the sign of the canonical vectors cannot be recovered. The standard CCA failed to extract them, due to a nonlinear relationship within the variables. Basically, the model implemented a change of representation from the variables input space to a new space of canonical variants, $\underline{u} = \underline{w}_y^t \underline{y}$ and $\underline{v} = \underline{w}_x^t \underline{x}$. Subsequently, the model mapped the repre-

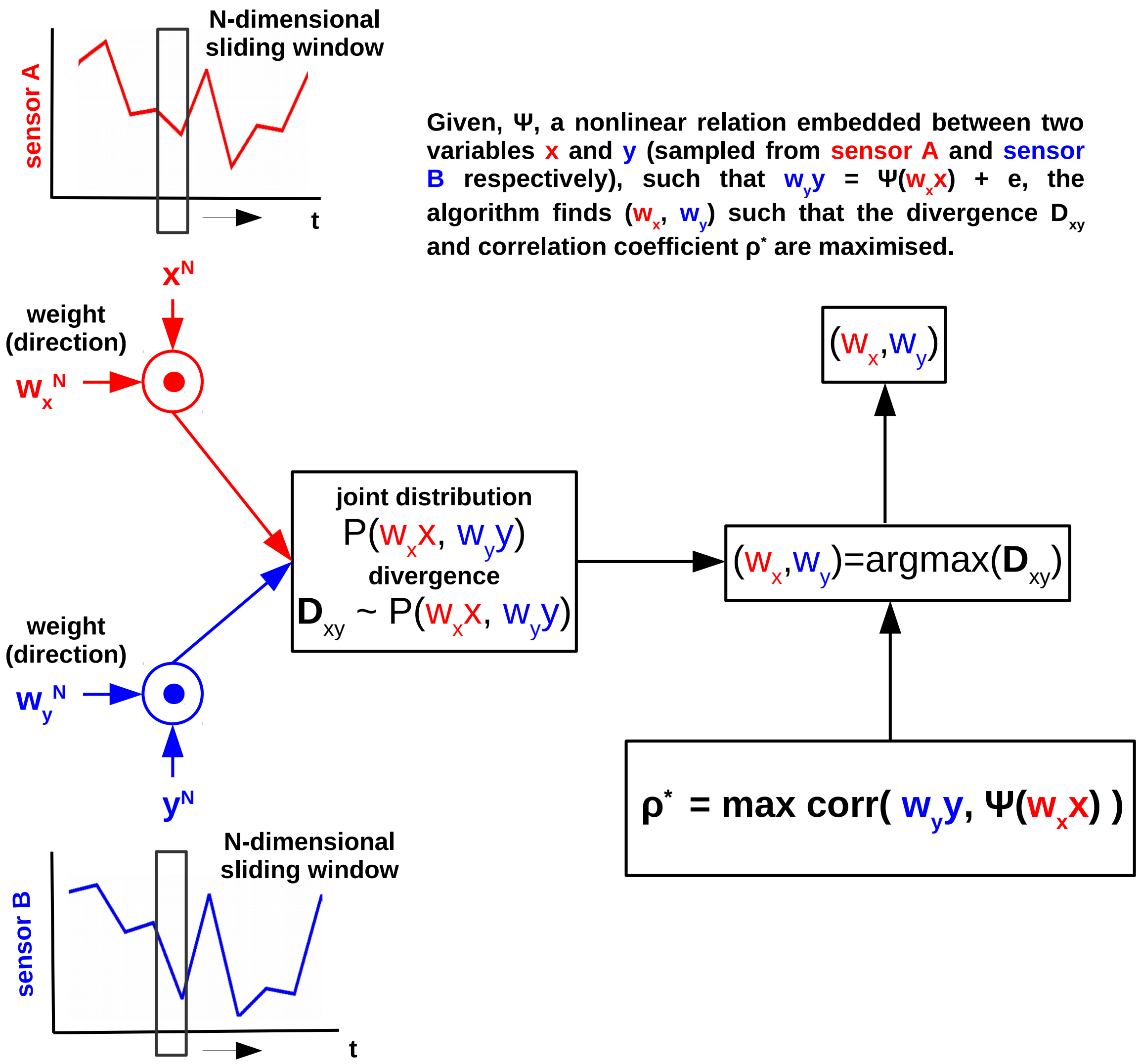

N-dimensional sliding window
sensor A
t
x^N
weight (direction)
w_x^N
Given, Ψ, a nonlinear relation embedded between two variables x and y (sampled from sensor A and sensor B respectively), such that w_y y = Ψ(w_x x) + e, the algorithm finds (w_x, w_y) such that the divergence D_xy and correlation coefficient ρ* are maximised.
(w_x,w_y)
joint distribution
P(w_x x, w_y y)
divergence
D_xy ~ P(w_x x, w_y y)
(w_x,w_y)=argmax(D_xy)
weight (direction)
w_y^N
y^N
ρ* = max corr(w_y y, Ψ(w_x x))
N-dimensional sliding window
sensor B
t

Fig. 5.8: Other approaches for learning sensory relations: [Mandal et al., 2013]

sentations back to the initial space minimising the relative mismatch between the original data and the mapping. The extracted relation was encoded in the weights configuration maximising the correlation between the canonical variants as shown in Figure 5.9b. The algorithm provided good results in extracting sensory relations in moderate noise conditions, for relatively small datasets, but with a cautious parametrisation of the divergence metric (i.e. taking into account prior information on the dataset densities). Furthermore, due to its iterative nature, the algorithm is prone to stop in local maxima, so it is needed to run the algorithm multiple times to obtain acceptable results.

Using a neurally inspired computing substrate for implementing canonical correlation analysis [Hsieh, 2000] proposed a model able to extract the underlying structures between two sets of variables under moderate noise conditions. The motivation behind this model was to counteract the limitations in the PCA to extract features or patterns in only a set of variables by looking only for modes of maximum variance. Furthermore, the model aimed at overcoming the CCA limitation to extract linear relations between two sets of (correlated) variables looking for modes of maximum variance. The proposed Nonlinear

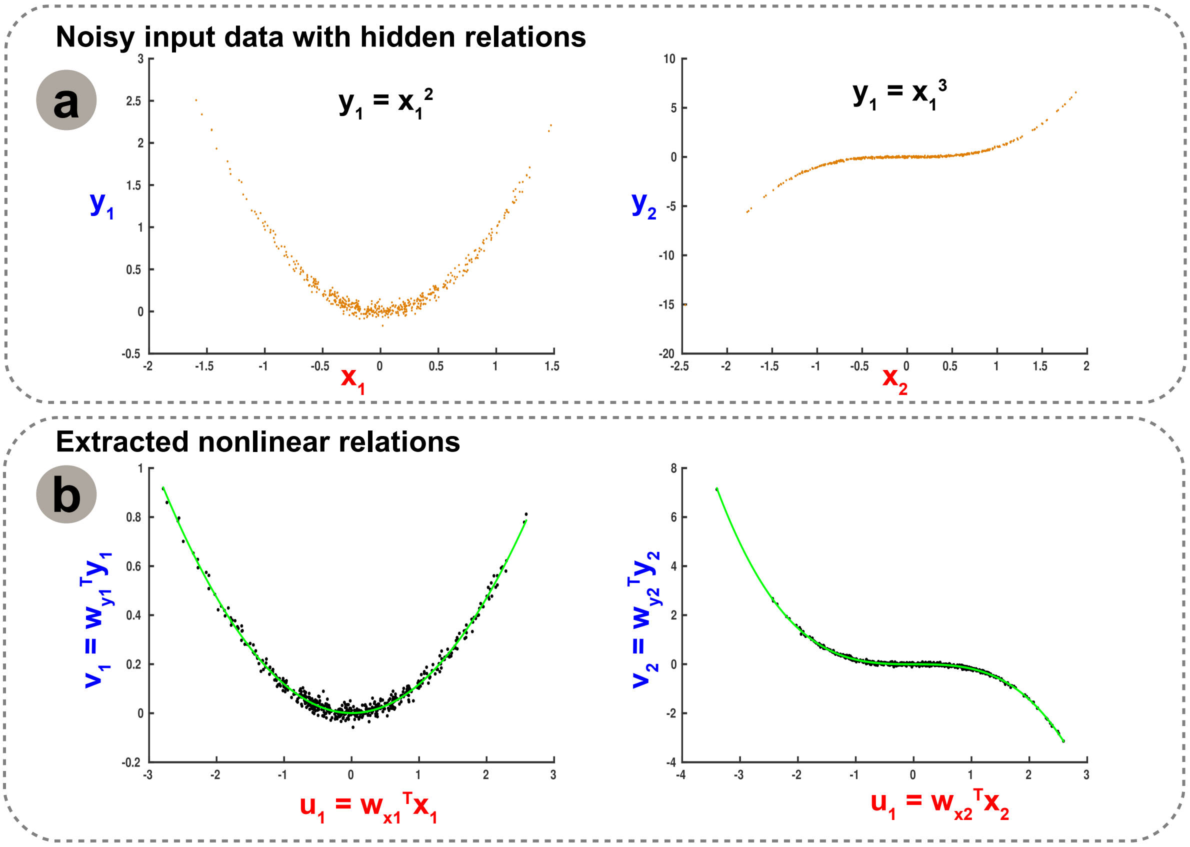


Fig. 5.9: Analysis of other approaches for learning sensory relations: [Mandal et al., 2013]. a) Input data and hidden realtions; b) Extracted nonlinear relations from the input data.

CCA (NLCCA) extended the CCA to be able to handle nonlinear mappings using an artificial neural network (ANN), more precisely mappings from input sets to canonical variants are realized by a feed-forward ANN (hyperbolic and linear transfer functions). In order to find the optimal values for the weight vectors in the CCA combinations, the network optimized (minimized) a cost function of the difference between the input variables and the mapped values (the output of the network). An interesting feature of the model is that it treats the input variables evenly, in that no causality is assumed. The model has the capability to perform inference in case one variable is missing (by using the learned relation) similar to [Cook, Jug et al., 2010]. A synthetic description of the processing stages in the model are provided in Figure 5.10. In order to test the NLCCA model the author used a variety of nonlinear functions with arguments randomly chosen from [-1, 1] interval. We implemented the model to extract the first correlated mode in the data, given that the input space is 3D. A small amount of Gaussian random noise, with standard deviation equal to 10% was added and the variables were then standardised (i.e. mean was removed, and values normalised by standard deviation). The input dataset contained 500 pairs of (x, y) values such that:

$$x_1(t) = t; x_2(t) = t^2; x_3(t) = t^3; \tag{5.10}$$

$$y_1(t) = t; y_2(t) = 3t; y_3(t) = t + t^2; \tag{5.11}$$

The model provided good results in the proposed scenario such that correlation between u and v is 0.996 in NLCCA and 0.993 in CCA. The more notable difference between

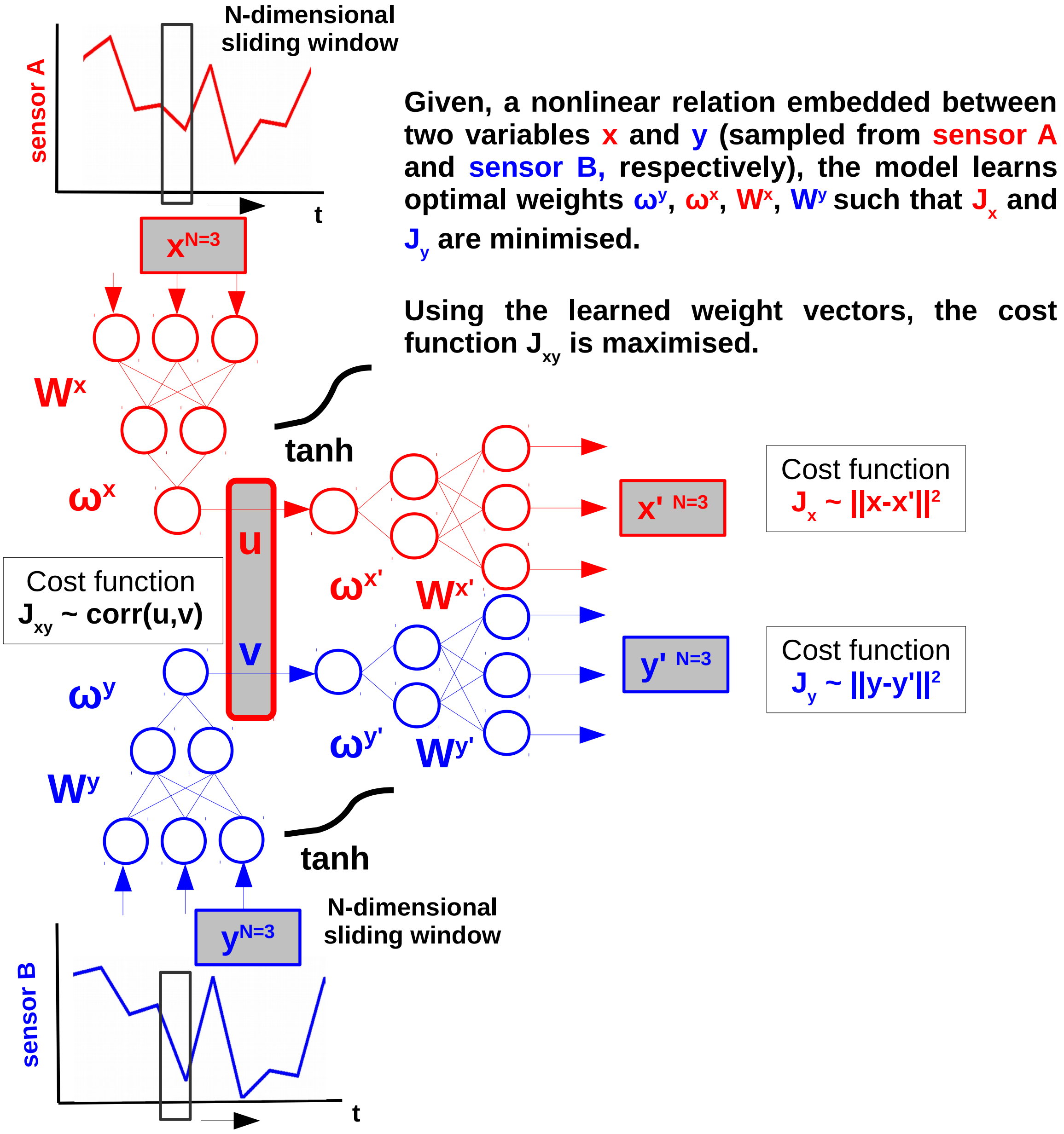

N-dimensional sliding window
sensor A
t
$x^{N=3}$
Given, a nonlinear relation embedded between two variables x and y (sampled from sensor A and sensor B, respectively), the model learns optimal weights ω^y, ω^x, W^x, W^y such that J_x and J_y are minimised.
Using the learned weight vectors, the cost function J_{xy} is maximised.
W^x
tanh
ω^x
u
v
$x'^{\ N=3}$
Cost function
$J_x \sim \|x-x'\|^2$
Cost function
$J_{xy} \sim corr(u,v)$
$\omega^{x'}$
$W^{x'}$
ω^y
$y'^{\ N=3}$
Cost function
$J_y \sim \|y-y'\|^2$
$\omega^{y'}$
$W^{y'}$
W^y
tanh
$y^{N=3}$
N-dimensional sliding window
sensor B
t

Fig. 5.10: Other approaches for learning sensory relations: [Hsieh, 2000]

NLCCA and CCA lied in the MSE in learning of x, the MSE was 0.028 for NLCCA (versus 1.165 for CCA); and for learning y, the MSE was 0.124 (versus 0.166). Although the model was able to handle high levels of noise applied to the data (up to 50% standard deviation) the precision decreased, as new neural networks structures were needed for more strong de-noising capabilities. A very interesting feature of the Hsieh model is that it is able, to some extent, to predict missing values given inferred correlation (i.e. variants correlation). Let's assume that the model has been built, and the standard deviations $std(u)$ and $std(v)$, of the canonical variates, are known, and have zero mean. If new x data becomes available, then u can be calculated, and v estimated by $ustd(v)/std(u)$, which can then be used to predict y. Similarly, x can be predicted using new y data. Providing a generalization of canonical correlation analysis through the use of feed-forward

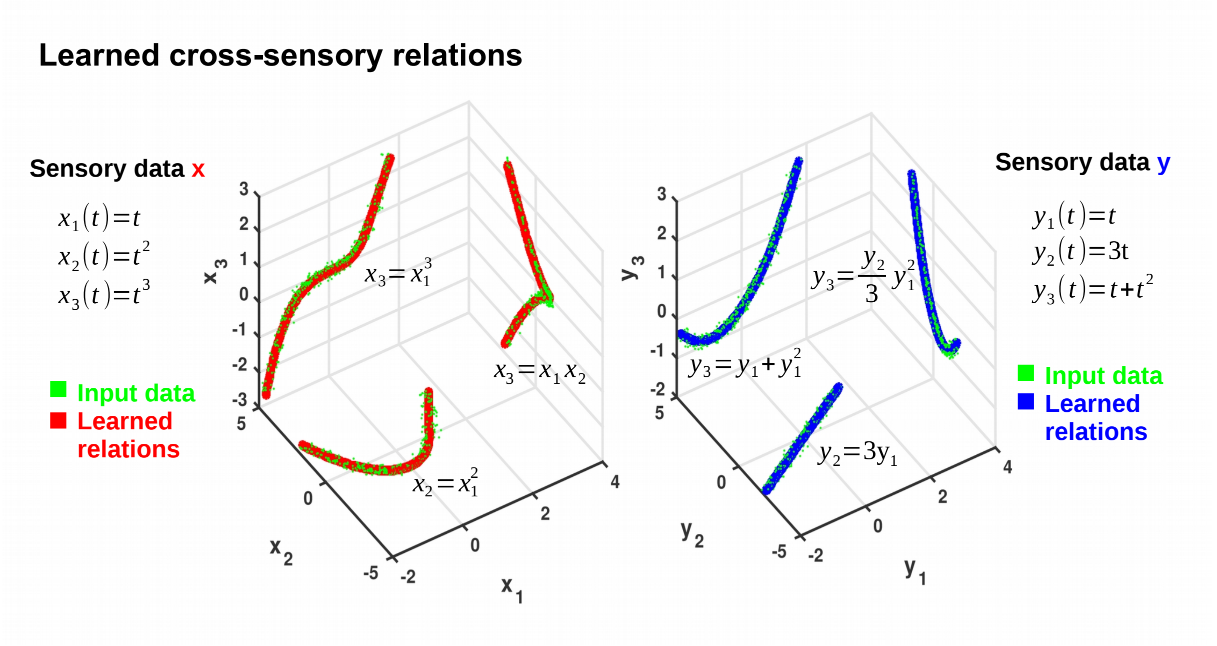


Fig. 5.11: Analysis of other approaches for learning sensory relations: [Hsieh, 2000]. Learned functional relations between the two datasets on a per dimension basis (N = 3, noisy input data - green, learned functional relation within x - red, learned functional relation within y - blue)

neural networks the NLCCA provides an interesting candidate for extracting nonlinear sensory data correlations. Although based on relatively precise correlation metrics and using optimisation to extract the best parameters to represent data statistics the model has some drawbacks which might prove to be unacceptable in real-time operation scenarios. In it's basic formulation, NLCCA cannot model curves which intersect themselves (e.g. a circle), it cannot model discontinuous functions (e.g. a step discontinuity can only be modelled by a continuous curve with a steep gradient at the step), and with noisy data, over-fitting (i.e. fitting to the noise in the data) can occur, resulting in wiggly solutions. Another problem is that with noisy data, the surface of the cost function may have many local minima, rendering most optimization searches to end at shallow local minima. Furthermore, the choice on the number of neurons should be minimal, as using excessive number of hidden neurons greatly aggravates the over-fitting problem.

Before switching to the more detailed description of the proposed model we summarize the most important features of the other models capable to extract sensory correlations. One initial aspect is the design and functionality. Either using distributed (neural) representations [Cook, Jug et al., 2010, Weber et al., 2007] or compact mathematical forms [Mandal et al., 2013, Hsieh, 2000], all methods encoded the input variables in a new representation to facilitate computation. At this level, employing neurally plausible dynamics [Cook, Jug et al., 2010, Weber et al., 2007, Hsieh, 2000] or pure mathematical multivariate optimisation [Mandal et al., 2013] the functionality was given by iterative processes converging to consistent representations of the sensory streams.

A second aspect refers to the amount of prior information set by the designer in the system. It is typical that, depending on the instantiation, a new set of parameters is needed, making the models less flexible. Although less intuitive, the pure mathematical approaches [Mandal et al., 2013] (i.e. using canonical correlation analysis) need less tuning effort as the parameters are the result of an optimisation procedure. On the other side, the neurally inspired approaches [Cook, Jug et al., 2010, Weber et al., 2007] or the hybrid approaches [Hsieh, 2000] (i.e. combining neural networks and correlation analysis) need a more judicious parameter tuning, as their dynamics are more sensitive, and can either reach instability (e.g. recurrent networks) or local minima. Except parametrisation, prior information about inputs is generally needed when instantiating the system for a certain scenario. Sensory values bounds and probability distributions must be explicitly encoded in the models through explicit tiling of tuning values over a population of neurons [Cook, Jug et al., 2010, Weber et al., 2007], linear coefficients in vector combinations [Mandal et al., 2013], or standardisation routines of input variables [Hsieh, 2000].

A third aspect relevant to the analysis is the stability and robustness of the obtained representation. The representation of the hidden relation can be encoded in a weight matrix [Cook, Jug et al., 2010, Weber et al., 2007] such that, after learning, given new input, the representation is continuously refined to accommodate new inputs; can be fixed in vector directions of random variables requiring a new iterative algorithm run from initial conditions to accommodate new input [Mandal et al., 2013]; or can be obtained as an optimisation process given the new available input signals [Hsieh, 2000]. Given initial conditions, prior knowledge and an optimisation criteria [Mandal et al., 2013, Hsieh, 2000] or a recurrent relaxation process towards a point attractor [Cook, Jug et al., 2010, Weber et al., 2007], the obtained representations are stable, optimising a cost function or reaching a desired tolerance.

The capability to handle noisy data, is an important aspect concerning the applicability in real-world scenarios. Using either computational mechanisms for de-noising [Cook, Jug et al., 2010, Weber et al., 2007], iterative updates to minimise a distance metric [Mandal et al., 2013], or optimisation [Hsieh, 2000], each method is capable to cope with moderate amounts of noise and becomes unusable when the signal-to-noise ratio is too low. Despite this, some methods have intrinsic methods to cope with noisy data intrinsicly, through their dynamics, by recurrently propagating correct estimates and balancing new samples [Cook, Jug et al., 2010].

Another relevant feature is *the capability to infer (i.e. predict / anticipate) missing quantities once the relation is learned.* The capability to use the learned functional relations to determine missing quantities is not available in all presented models like [Mandal et al., 2013] due to the fact that the divergence and correlation coefficient expressions might be non-invertible functions, to support a simple pass through of available values to extract missing ones. On the other side, using either the learned co-activation weight matrix [Cook, Jug et al., 2010, Weber et al., 2007], or the known standard deviations of the canonical variants [Hsieh, 2000] the model is able to predict missing quantities.

Finally, due to the fact that all methods re-encode the real-world values in new representation, it is important to study the capability to decode the learned representation and subsequently measure the precision of the learned representation. Although not explicitly treated in the presented models, decoding the extracted representations is not trivial. Using a tiled mapping of the input values along the neural representations [Cook, Jug et al., 2010] decoded the encoded value in activity patterns by simply computing the distribution of the input space over the neural population units, while [Weber et al., 2007] used a simple winner-take-all readout given that the representation was constrained to have a uniquely defined mapping (i.e. in the scenario the assumption is made that the object to be tracked by the robot is always at the same elevation from the floor). Given that the model learns the relations in data space through optimisation processes [Hsieh, 2000] can use learned curves to simply project available sensory values through the learned function to get the second value, as the scale is preserved. Albeit its capability to precisely extract nonlinear relations from high-dimensional random datasets [Mandal et al., 2013] cannot provide any readout mechanisms to support a proper decoded representation of the extracted relations. This is due to the fact that the method cannot recover the sign and scale of the relations.

5.2.2 Analysis of the basic model

In the following section we introduce the features of the basic sensory relation learning model. The model acquires samples from the two sensory streams, encodes them in distributed populations of neurons (i.e. activation pattern), and then learns the correlation patterns between the two distributed representations (i.e. co-activation).

For the basic scenario we consider a bimodal relation learning problem. Each input sensory stream is encoded by a SOM composed of 100 neurons distributed in a one-dimensional lattice. We use a one-dimensional representation to encode single subsequent samples from the input stream and provides a sufficient substrate to extract and represent the input data distribution (i.e. through the shape and density of neurons' tuning curves). Each input sample elicits a distributed activation pattern across the network such that each neuron responds proportionally to the distance between his preferred value and the input sample value. If the activation patterns in each of the input SOMs are correlated, the Hebbian linkage between the two networks will enhance the links between highly activated neurons. Subsequent samples will determine the enhancement for correlated structure in the two input signals and depression for un-correlated modes.

As previously mentioned the correlation learning rule enhances correlated neural activities by strengthening synaptic weights following the original Hebbian postulate. This formulation only allows for an increase in synaptic weight between synchronously firing neurons. To prevent unlimited growth, it is necessary to extend the Hebb's rule to allow for weight decreases when neurons fire asynchronously using a covariance learning rule. In our experiments we used two rules for extracting the sensory relation, namely covariance learning and Oja's local PCA learning [Chen et al., 2007], both providing relatively similar results, with insignificant differences in computational implementations, but similar impact on the precision of the representation. In the case of the covariance learning rule, the synaptic strength between neurons i and j in populations p and q, respectively, is given by

$$\Delta w^{p}_{cross,i,j}(k) = \eta(k)(a^{p}_{i}(k) - \overline{a}^{p}_{i}(k))(a^{q}_{j}(k) - \overline{a}^{q}_{j}(k)), \tag{5.12}$$

where if we take a time average of the change in synaptic weight,

$$\overline{w}^{p}_{cross,i,j}(k) = \eta(k)(\overline{a^{p}_{i}(k)a^{q}_{j}(k)} - \overline{a}^{p}_{i}(k)\overline{a}^{q}_{j}(k)), \tag{5.13}$$

the first term on the right-hand side denotes the Hebbian synapse and the second term may be viewed as an activity-dependent threshold that changes with the product of time-averaged pre- and postsynaptic activity levels. If, on average, the presynaptic activity $a_i^p(k)$ is independent on the postsynaptic activity $a_j^q(k)$, namely $\overline{a_i^p(k)a_j^q(k)} - \overline{a}_i^p(k)\overline{a}_j^q(k)$, then no change in synaptic weight should occur. As a special case of the covariance learning rule, Oja's local PCA learning, is a local and computationally efficient learning rule, keeping the Euclidean norm of a neuron's incoming synaptic weight vector at unity. The online version of Oja's rule used in our work assumes the weight update is given by

$$w^p_{cross,i,j}(k+1) = \frac{w^p_{cross,i,j}(k) + \eta(k)a_i^p(k)a_j^p(k)}{\sqrt{\sum_{l=1}^{N}(w^p_{cross,l,j}(k) + \eta(k)a_l^p(k)a_j^p(k))^2}}, \tag{5.14}$$

In order to test the functionality of the basic model we fed the network with correlated artificial sensory datasets. Each sensory dataset contained 1500 samples and followed different data distributions. As the proposed model comprises multiple learning and adaptation processes we often varied the input data distribution, such that we were able to analyse the behaviour and performance of the network by feeding data with uniform, nonuniform, or mixed probability distributions. An overview of some notable experiments is given in Figure 5.12. In the first experiment we feed sensory data with a hidden linear sensory relation with nonuniform data distribution (e.g. convex probability distribution), Figure 5.12 a left panel. The network extracts the relation and encodes it in the strength of the Hebbian links and in the tuning curves of each input SOM neuron. Higher density areas in the input space are characterised by narrower tuning curves and wider areas by broader ones. Consistent with the learned sensory data distribution, the network allocates more neurons to represent areas with a higher density (i.e. narrow tuning curve), and less neurons for coarser represented areas in the input space. The capability to encode the density of the data distribution can be used to define reliability maps of the sensors, and subsequently used in fault detection and accommodation.

In a second experiment we feed sensory data with a hidden nonlinear sensory relation (i.e. second order power-law) following a nonuniform data distribution (e.g. convex and powerlaw probability distributions), Figure 5.12 b left panel. Similar to the first scenario we observe that the network extracts the hidden relation, sensory data distribution, and judiciously allocates neurons for a consistent representation. Furthermore, we observe that the learned tuning curves' shapes and densities are uneven (heterogeneous), providing a non-equidistant tiling of the input space, and representing the irregularities and variability describing real-world data. In the current and all the other experiments we performed the representation method produced comparable results with [Ganguli et al., 2014]. We consider that the proposed approach in the thesis provides an alternative formulation of the efficient coding hypothesis for a neural population encoding a scalar stimulus variable drawn from an unknown prior distribution. In [Ganguli et al., 2014] the information-maximizing solution provided precise and intuitive predictions of the relationship between sensory prior, physiology, and perception: more frequently occurring stimuli should be encoded with a proportionally higher number of cells and a proportionally higher perceptual sensitivity for the frequently occurring stimuli. Our model was able to unsupervisedly obtain representations consistent with the predictions.

In the third third experiment we fed uniformly distributed data in the [-1, 1] interval implementing a nonlinear periodic function (i.e. sine wave). Tiling evenly the entire input

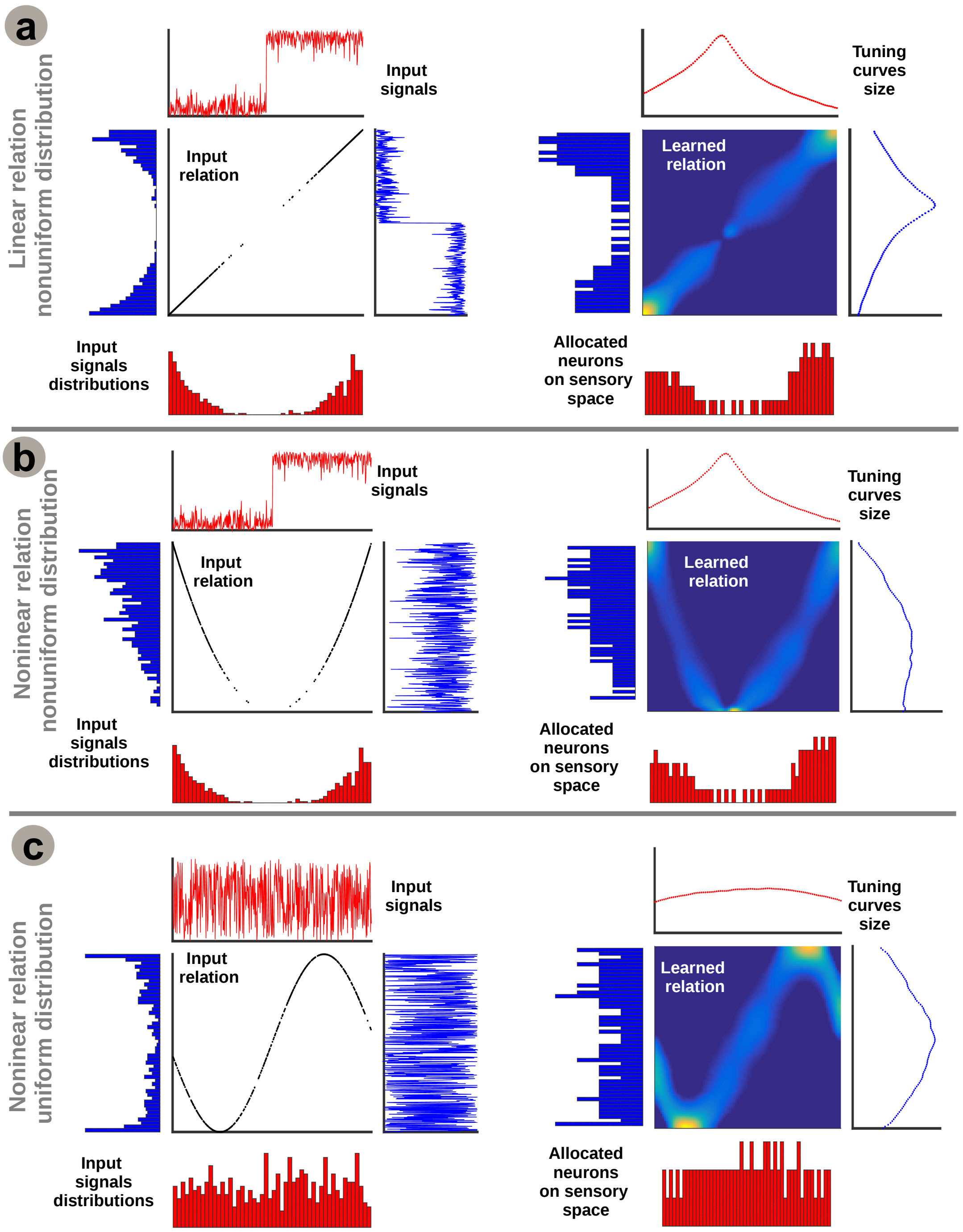


Fig. 5.12: Analysis of the basic model in a bimodal scenario. Different hidden relations and data distributions: input data and its probability distribution (left); learned relation and allocated resources (i.e. neurons) according to input distributions (right). a) Linear sensory relation with nonuniform data distribution; b) Nonlinear sensory relation with nonuniform data distribution; c) Nonlinear sensory relation with uniform data distribution.

space, the network allocates neurons uniformly, such that that each region of the input space is equally represented.

Given incoming streams of correlated sensory data, each input SOM uses cooperation, competition, and adaptation (plasticity) to learn and represent the input data statistics in a heterogeneous population code. The representation process is jointly evolving with the relation extraction process, such that, through Hebbian learning, the network learns the underlying relation between the data, given that the input are efficiently represented in the SOM. Subsequently, the network uses the stable state (the learned relation) for cue integration, such that the learned relation (weight matrix) imposes the constraint on the possible values a sensory stream can have. At this stage, during cue integration each sensory modality representation will do its best to keep consistency with all the relations it is involved in, subject to the constraint imposed by the relation encoded by the weight vector.

An important aspect is that the network models synaptogenesis, such that initially the SOM projection weights are 0, and it doesn't need any prior information about the span of the input data distribution. This aspect, as well as the fact that the two concurrent processes evolve simultaneously, is consistent with the processes known to explain development in cortical circuitry. Featuring biologically plausible mechanisms the network increases its robustness capabilities (i.e. adapt to incoming streams of sensory data by enhancing / penalizing contributions) as on the longer timescale the input representation process adapts the structure for the faster sensor fusion process. After relaxing in a stable state the network contains a fully informative representation of the input data and the learned sensory relation. In order to make use of the learned relation we developed a simple readout mechanism. Given the ordered representation of the input data space onto the SOM lattices, one can find the corresponding real-world values by finding the best (optimal) solution of a cost function of maximal sensory elicited activation given input patterns. Bounding the value of the cost function with learned preferred values, a simple optimization method decodes the corresponding sensor value.

5.2.3 Inference and fault tolerance capabilities

After the learning process, the network stores a stable representation of the hidden relation between the two sensory inputs considered during training. By considering only one input sensory source, the network can infer the corresponding quantity for the missing source by using the learned co-activation pattern stored in the Hebbian linkage.

Given one input sample from the input sensory stream, the network computes the elicited activity in the input SOM population (pre-synaptic neurons). The resulting activity pattern is projected through the Hebbian linkage to compute the post-synaptic activation pattern in the output SOM population. Due to the all-to-all connectivity pattern, the activity of a single neuron in the output population is given by the sum of (Hebbian) weighted activity values in the input population. The resulting output activation pattern will peak at the most active (post-synaptic) neuron given the pre-synaptic input pattern. The position in the SOM lattice and the corresponding activation value are subsequently used for decoding the population activation pattern and recover the real-world sensory value.

We developed two methods for decoding the activity pattern and extract the corresponding real-world value. The first method is a naïve decoder, which simply computes a term to finely tune the preferred value of the most active (winning) neuron towards a more precise estimate. Given samples from the input stream $x(t)$ the most active neuron neuron has index i in the output SOM, a preferred value $w^p_{in,i}$ and $\xi^p_i(k)$ tuning curve size, the corresponding increment term is given by

$$d^p_{fi}(k) = \sqrt{2\xi^p_i(k)^2 log(\sqrt{2\pi} a^p_i(k)\xi^p_i(k)^2)}. \tag{5.15}$$

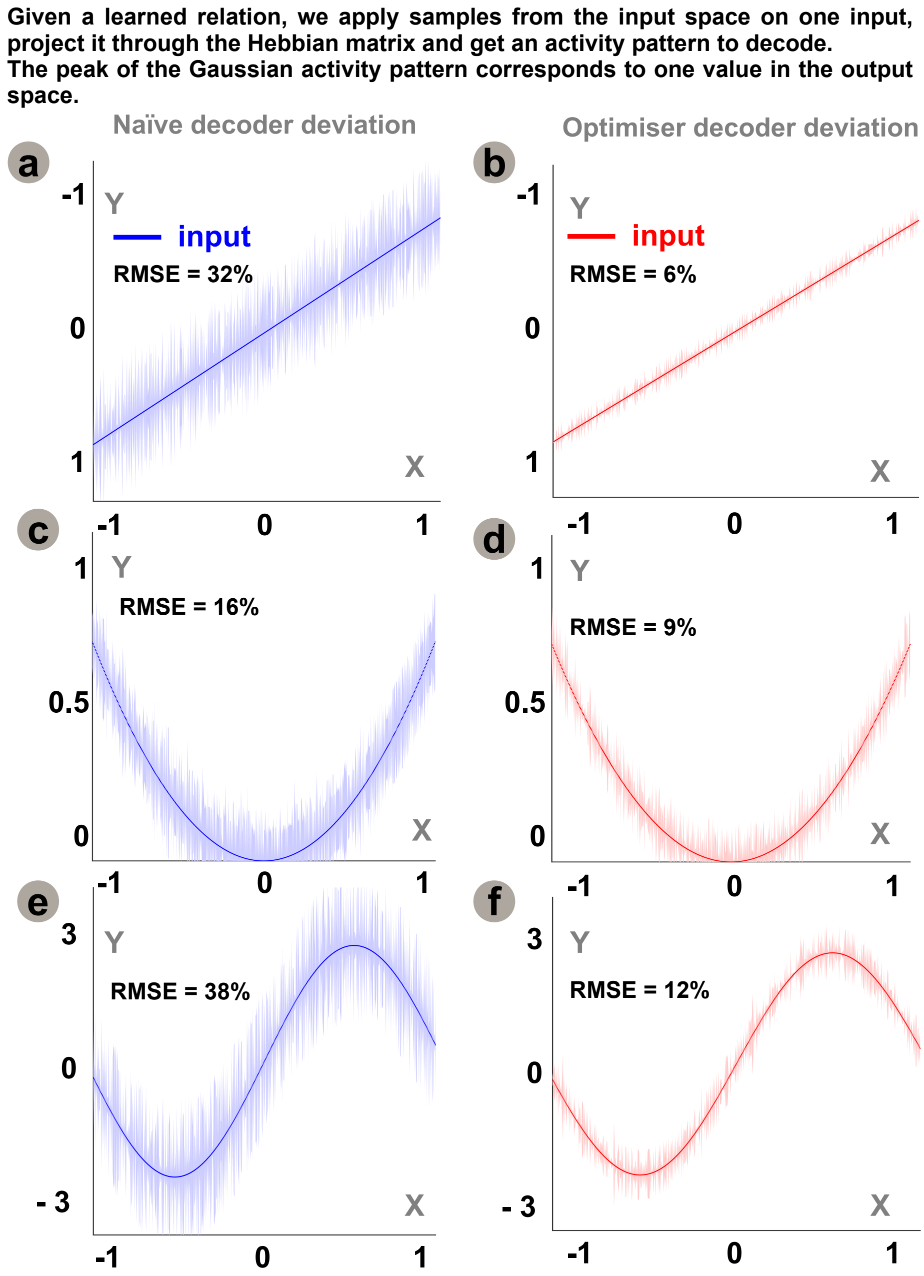
Depending on the position of the winning neuron in the N-dimensional lattice the recovered value $y(t)$ is computed as

$$y(t) = \begin{cases} w^p_{in,i} + d^p_{fi},\ if\ i \geqslant \frac{N}{2} \\ w^p_{in,i} - d^p_{fi},\ if\ i < \frac{N}{2} \end{cases}$$

A second, more precise, decoding mechanism is based on an optimisation method to recover the real-world value given known bounds in the input space. The bounds are obtained as minimum and maximum of a cost function of the distance between the current preferred value of the winner neuron and the input sample. The optimiser is based on Brent's method [Brent, 2013] which uses a recursive method to find the global optimum of a function for which the analytical form of its derivative is not available or too complex. Using this approach, after applying the input sensory stream and finding the winner in the input SOM population, the decoding decision is based on the position of the winner. Two bounds (i.e. left and right) are defined with respect to the winner's position such that the recovered value is obtained by running the algorithm between the preferred values of the neurons with indices given by the bounds. The method is not guaranteed to converge to global minima (of the cost function) and it's not immune to boundary effects, if winners are placed at the extremes of the SOM population.

In order to emphasize the capabilities of the two decoding mechanisms we provide in Figure 5.13 a brief analysis for some of the sensory learning scenarios previously used in the chapter. As one can see the decoding performance is satisfactory, yet the recovered values lie around the correct input pattern. By analysing the learned representation stored in the Hebbian matrix we noticed that, due to the asymmetric neighbourhood function in the input SOMs, the activity will saturate at the edges of the latent representation space. This behaviour is also visible in the co-activation pattern, such that the higher activity values characterise the bounds of the Hebbian representation towards the edges. Both decoding mechanisms assume that by applying one input to the network and projecting the sensory elicited activity pattern on the Hebbian matrix we can extract a plausible activation pattern for the missing sensory modality.

When decoding the activity pattern both approaches provided a relatively similar recovered probability distribution shape. This interesting behaviour relates the boundary effects in the SOM representation and Hebbian co-activation pattern with the extracted sensory data distribution learned from the data. Inspecting both decoders' probability distributions we observed that if the input data is uniformly distributed decoders' output is biased. The resulting distributions have a convex profile, concentrating a large number of



Given a learned relation, we apply samples from the input space on one input, project it through the Hebbian matrix and get an activity pattern to decode. The peak of the Gaussian activity pattern corresponds to one value in the output space.

Fig. 5.13: Analysis of decoder performance for various types of sensory data relations. Performance of the naïve decoder for: a) Linear input, b) Nonlinear input symmetric input, c) Nonlinear periodic input; Performance of the optimised decoder: a) Linear input, b) Nonlinear input symmetric input, c) Nonlinear periodic input.

samples towards the edges of the histogram with a large variance, while precisely decoded areas follow a relatively uniform distribution. We notice that the optimiser based decoder, although more complex, provides better recovery results (smaller RMSE is better), such that the deviation is relatively small for linear relations (RMSE: 0.0613) in comparison with the naïve approach, which provides really imprecise recovery values (RMSE: 0.3247). For nolinear relations the optimiser decoder is performing relatively well (RMSE: 0.0912), overtaking the naïve decoder, which surprisingly performs better than in the linear case, due to less prominent boundary effects (RMSE: 0.1671). Finally, for symmetric nonlinear relations, both decoders have a hard job to recover values due to the irregularities of the learned representation, such that the optimiser decoder lies in 12% from the mean of the input signal, while the naïve decoder is far off (38%).

5.2.4 Extensibility: from dual modality to multimodal processing

Following the analysis performed in previous sections, we now investigate the extensibility capabilities of the model for multimodal processing. When studying the scalability capabilities of the network we focused on two possible network architectures.

In the first approach we consider one sensory modality as providing an estimate of a desired quantity for which we need to have a precise estimate. All the other sensory modalities contribute to the network belief by being internally coupled within their own estimate of the desired feature. The coupling is reflected by the hidden relation in the data coming from individual sensory modalities. To exemplify, we propose a simple 4-dimensional scenario depicted in Figure 5.14. This scenario is not bound to a 4-dimensional architecture, rather it can accommodate an arbitrary number of modalities able to contribute to the estimation of the feature of interest. While analysing the learned representation in the network we observed that, locally, each Hebbian matrix encoding the representation is sharp and can be properly decoded, while the sensory modality encoding the initial cue to be estimated contains an interfering pattern of activations, Figure 5.14 b. As the 4 modalities are linked through a tree structure correlation there is no internal constraint explicitly

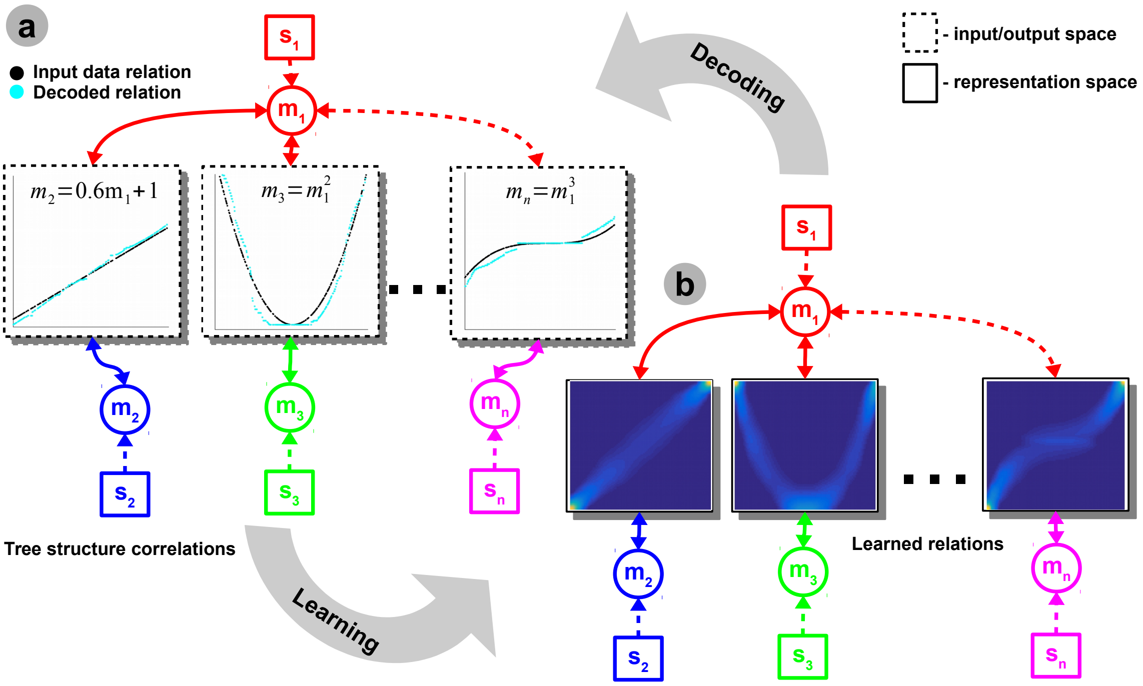


Fig. 5.14: Analysis of the extensibility capabilities of the network. Sample scenario with a 4-dimensional network with a tree shaped correlation structure. a) Input data and decoded learned representation; b) Learned relations.

defined in the network, such that the incoming contributions adapt on each branch the locally learned representation. We can see that the representations can be decoded easily, as there is no interference due to the open structure, Figure 5.14 a, such that each contribution is combined in the overall network belief.

In a second approach, we use a 3-dimensional network to investigate the use of explicit connectivity and representation in a fully-connected network. The difference between the first scenario and this one, is that the intrinsic correlation between modalities is now explicitly extracted in a dedicated (separate) Hebbian linkage. This approach imposes an additional constraint, such that the co-activation pattern in the Hebbian matrix back-projects an influence on the local representation, which subsequently propagates in the network representations. Interestingly enough, this mechanism supports the representations and dynamics introduced in Chapter 3, where all constraints narrowed the space of possible values a node (or population) can take. This mechanism ensures a more sharp representation, no interference, and a more precise decoding, as boundary effects are slowly compensated during the network operation.

To illustrate the proposed approach we considered a simple 3-dimensional network employing a mixture of linear and nonlinear relations and the representation of the intrinsic relation between the implicitly connected modalities, depicted in Figure 5.15. During our experiments we noticed that the network dynamics can get even more sharper representations of the underlying relations, Figure 5.15 b, in a circular structure due to the additional explicit constraint and back-projections of the Hebbian matrices. We also locally decoupled the network between m_2 and m_3 such that the network had learned the relations

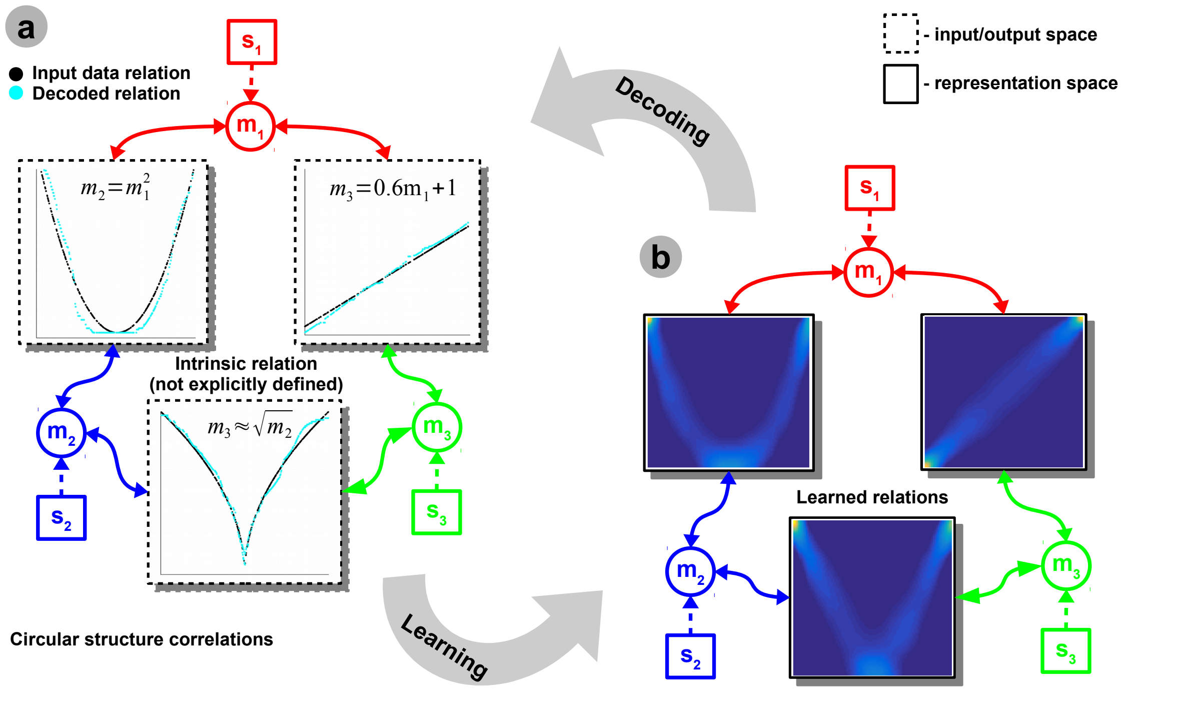


Fig. 5.15: Analysis of the extensibility capabilities of the network. Sample scenario with a 3-dimensional network with a circular correlation structure. a) Input data and decoded learned representation; b) Learned relations.

separately. After learning we fed sensory data through m_2 and m_3 and using an additional co-activation matrix we noticed the emergence of the intrinsic relation between the two, initially unconnected units.

5.3 Summary

In order to extract mappings between external stimuli and the internal state of the system, we investigated perceptual learning mechanisms and their capability to learn the dependencies between the different incoming streams of sensory data.

Typical engineering models provide just an approximation of the sensory models (i.e. simplified, constrained models) such that there is no explicit handling of uncertainty and real noise conditions. The intrinsic dynamics of the sensors are important to be extracted, such that the intimate structure of the data is exploited for more precise representation and computation. Extracting and making sense of the underlying relations in the sensory streams turns multisensory fusion more powerful and the outcome more precise.

Various tools to extract sensory correlations were developed [Mandal et al., 2013, Cook, Jug et al., 2010, Weber et al., 2007, Hsieh, 2000] employing different methodologies to extract the underlying relational structure, spanning from canonical correlation analysis, to biologically plausible networks and artificial neural networks. All these methods provide good results in dedicated scenarios, but lack the capability to be employed in novel contexts. We provide a thorough analysis of all these methods and how they compare with our approach with respect to: design challenges and functionality; the amount of prior information needed during design; the stability and robustness of the obtained representations; the capability to infer (i.e. predict / anticipate) missing quantities after extracting the relation; and the capability to decode the learned representation and subsequently measure the precision of the learned representation.

Turning towards biologically inspired mechanisms for models of representation and learning from sensory data, we propose a model which, rather than focusing on biologically precise descriptions of neural circuitry, employs simple computational blocks, known to be widespread in the brain, and which are well formalised and understood. Following models known to explain sensory processing in cortex, with respect to local processing and its influence upon the state of the formed features representations, [Cimponeriu et al., 2000, von der Malsburg, 1999, Michler et al., 2009, Quiton et al., 2011] provided different mechanisms to exploit sensory data structures for organising representation on various timescales and reference systems. Furthermore, one important aspect was the analysis of the reciprocal influence representations have on processing mechanisms, due to the mutual interaction of the communicating areas encoding a specific sensory modality.

Experience acquired through sensory exposure supports the learning mechanisms responsible to extract the correlational structures in the percept, and can be viewed as an outcome of a development process. Consistent with our goal, to keep the computational substrate simple and flexible enough, suitable real-time operation, the proposed model uses competition, cooperation, and correlation as mechanisms to unsupervisedly extract hidden relations between sensory streams.

Changing data representation and subsequently the computation paradigm, the model re-encodes single real-world values into a distributed activity pattern over a network of processing units (i.e. neurons). This representation allows the system to extract the underlying probability distribution of the sensory data, such that there is no need to explicitly embed it in the model at design time (and so, constraining the system). Competition and cooperation between the units ensure that the input space is faithfully represented: finer resolution representation to more relevant areas in the input space and coarser resolution to irrelevant areas and outliers.

Combining the timing and shape of activation patterns (i.e. the distributed response of the units), associated with different input streams allows the model to extract the co-activation, in fact their correlational structure. After learning, the underlying relation the model can be used to infer missing quantities or to detect anomalous or erroneous input signals, given that a correct relation was previously learned. Furthermore, the extracted relation can be decoded such that the real-world value can be recovered from the distributed activation pattern. This is useful when the systems is used in a real-world scenario, to provide feedback to a motor controller acting upon the perceived environment.

The proposed model relieves the system designer from the intense and cumbersome parametrisation routines, as the underlying learning processes take advantage of the intimate structure of the sensory data. This supports an efficient representation and subsequent fast computation for flexible and robust multisensory fusion sought in real-world technical systems.

6 Instantiating the model for perceptual learning in multisensory fusion

In order to develop and test the perceptual learning and sensory processing hypotheses introduced in Chapter 4, robotic systems provide a great experimentation and validation platform. Embedding principles of neural systems processing and development, is a key approach to leveraging robustness and adaptation capabilities in autonomous robotic systems.

Neuroscience lessons thought us that learning processes which take place during the development of a biological nervous system enable it to extract mappings between external stimuli and its internal state. Precise egomotion estimation is essential to keep these external and internal cues coherent given the rich multisensory environment. In this chapter we analyse sample instantiations of our learning model which, given various sensory inputs, converges to a state providing a coherent representation of the sensory space and the cross-sensory relations. Moreover, exploiting the intrinsic structure in the sensory streams, the system autonomously extracts cross-sensory regularities to form associations subsequently used for sensory fusion. Before analysing the specific instantiation, we provide some insight on the synthesis mechanisms of the learning multisensory fusion network. As mentioned in Chapter 5, given pairs of sensory stream the network is able to learn the underlying relation. The questions now, is how can the system learn itself which structure is providing an advantageous setting for combining sensory modalities?

6.1 Constructing a network for sensory representation and processing: from graph theory to developmental neurobiology

Before evaluating the capability of our model to learn sensory correlations for 3D motion estimation, we review some relevant concepts in networks theory for growth and development, spanning from graph theory to developmental neurobiology, relevant to the process of constructing the learning multisensory fusion network. In order to frame our work and motivate our model's characteristics, we analyse some relevant formal models of network growth processes at the base of topology and spatial patterning. This analysis is needed to emphasize that the network should intrinsicly reconcile the opposing demands of segregation and integration of functionally specialized sensory representations.

Deeply rooted in graph theory, the seminal work of Erdos [Erdos et al., 1960] focused on structural growth and evolution processes in random graphs. In this model, following only local rules, adding new connections determined the emergence of a patterned structure. Structural modifications influence the dynamics of local nodes following a reactive mechanism (i.e. force-spring growth process). In the context of sensory data combination and integration, the network could support realistic hypotheses or rules (e.g. physical constraints between nodes) and replace the initially equiprobable connections.

Extending the mathematical formalism [Albert et al., 2002] introduced the scale-free networks architectures, for which the probability distribution of the number of connections one node has to other nodes was a power-law. The growth process (i.e. adding nodes and connections) was given by a "preferential attachment", as a means of quantifying a correlation coefficient or cost function to optimise. Identifying a similar growth process known to describe cortical networks development, [Kaiser et al., 2004] proposed a model which started with a minimal number of nodes and added more nodes and connections with a probability that decreases exponentially with the Euclidian distance between the nodes. This principle was consistent with previous studies on minimal wiring theorem in cortical map formation processes [Mitchison, 1995], interpreting sensory representations (i.e. maps) as the solution of a minimisation problem, where the goal is to keep the "wiring" between neurons with similar receptive fields as short as possible. In a sensory learning context, using this kind of scale-free network growth as a solution of a minimization problem, could exploit the fact that the connection probability can be modulated by measuring coupling correlation provided by some metric (mimicking cellular and gene expression influence).

Theoretically examining the interdependence between structure and dynamics in the brain, [Rubinov et al., 2009] provided biophysical justification for the structural and functional dependencies in large networks of neurons. An important aspect in the study was the analysis of time scale dependent differences between structure and function, such that on fast time scales structure enables the emergence of complex dynamics, while on slow time scale structural connectivity is gradually adjusted towards the resulting functional patterns via an unsupervised, activity-dependent rewiring rule. Initially random, the structure converged towards asymptotic states characterized by globally invariant structural and functional clustering.

In-line with our idea of extracting sensory correlations in a network of distributed representations, [van Ooyen et al., 2003] proposed a functional model of the low-level interactions in developing neural circuitry. An important aspect is that the representations are tightly coupled through co-activation patterns of learning. Some interesting and highly relevant concepts were introduced and supported by experimental data. The first concept was the fact that the activity patterns generated by a developing neural network can modify the organization of the network and the functionality of its neurons, leading to altered activity patterns, which in turn can further modify structure and function.

The underling process shown that when the activity of a neuron is high, neuronal connectivity and excitability are modified by activity-dependent processes so as to decrease activity. Conversely, when the activity of a neuron is low, on the other hand, neuronal connectivity and excitability will be modified so as to increase activity. These phenomena emerge without assuming predetermined, time-scheduled mechanisms. The core idea is that each neuron attempts to keep a certain level of activity (i.e. homeostasis) and regulates its fan-in and fan-out connectivity pattern such as to maintain activity at a critical level. These concepts provide an interesting new insight in how a network able to extract correlations between different representations (of sensory streams) can be built. Yet, there is no precise information on where the critical homeostatic activity regulation threshold is, and how can this be used in a practical implementation scenarios to build a network able to shape its structure according to incoming sensory streams. Embedding the ”coevolutionary” impact on structure and function, [Westermann et al., 2007] designed a network combining topological changes with internal dynamics. Each computing node embedded complex dynamics and supported robust topological self-organization based on simple local rules. An important observation was that, for many applications it is not necessary to capture the exact topology of a given network in a model, rather the process of interest depends on certain topological properties. The proposed adaptive ”coevolutionary” network model proposed an interesting separation of dynamics, namely dynamics of network and dynamics on networks. In the first case, of the dynamics of the network, topology is regarded as a dynamical system itself, such that it changes in time according to specific rules. Dynamics on networks, focuses on the perspective that each node of the network represents a dynamical system and individual nodes are coupled according to the network topology which remains static while the states of the nodes change dynamically. An open question of the presented study was, which topological properties are affected by a given set of temporal changes in state or topology, so that they can act on topological degrees of freedom? This questions is highly relevant for multisensory integration, providing an understanding on how the combination rules can limit the overall capacity of the network to store a complex representation of the state or environment.

Providing a unifying view on development of sensory representations and processing, [Parise et al., 2012] framed experience dependent learning of internal representations as a trajectory emerging from the interplay of multiple constraints. The framework advocated that changes to the brain hardware change the nature of the representations and their processing (i.e. the algorithm) which leads to new experiences and further changes to the neural systems. Narrowing the generic perspective of the neuroconstructivist framework, we emphasize the contact points with our research. The focus falls now on experience dependent elaboration of small-scale canonical computing structures and the ”interactive specialization” view of cortical development, which stresses the role of interactions between different brain regions in functional development. Targeting real-world scenarios, the model used robotic systems to test the proposed hypotheses of the multiple interacting biological and environmental constraints, neural development, and the development of cognitive representations. Finally, inferring which signals have a common underlying cause, and hence should be integrated, represents a primary challenge for a perceptual system dealing with multiple sensory inputs [Parise et al., 2012]. The outcome of this study was that humans use the similarity in the temporal structure of multisensory signals to solve the correspondence problem, hence inferring causation from correlation. This principle is also at the core of our model, and was verified through the analysis carried on the results of our robotic experimentation scenario. Summing up, after reviewing this relatively wide range of mechanisms, emerging from pure formal mathematical theories to neurobiologically plausible processes to create a processing structure able to extract the underlying structure of incoming streams, we can extract some important design principles. Using distributed representations of sensory data yields for a distributed processing substrate. Using global and local dynamics, the model should be able to quantify the correlational structure in the input streams. As we found out in all the analysed studies, correlation extraction can be seen as an optimisation problem such that the model should find optimal parameters to represent the input space structure. Furthermore, given a stable representation of the input space, the model should be able to react to new input and adapt its structure to properly represent the new configuration of the input space. Of course, this process is bound to a different time scale than local representation dynamics, such that the model must keep consistency between the multiple scales to ensure consistency. Finally, another important aspect refers to the type and the dimensionality of the input streams, such that the model should solve a correspondence problem given multiple spatial and temporal sensory dimensions. All these concepts allowed us to build a model capable of extracting the underlying correlational structure in various input streams, as shown in Chapter 5. Despite its learning capabilities, our model, in its basic formulation, is not able to infer its own structure, rather uses a predefined configuration of sensory streams to learn the correlations. We extend our model in Section 5.3 such that it is able to build a network capable of performing multisensory fusion using information theoretic measures of correlational structure between all available sensory modalities.

6.2 Multisensory fusion for quadrotor 3D egomotion estimation

The initial scenario we consider is 3D egomotion estimation on a quadrotor, for which our model provides precise estimates for roll, pitch, and yaw angles, given available sensory data onboard. The data acquisition and control infrastructure was previously developed in our lab [Bergner et al., 2014]. The setup is depicted in Figure 6.1. For the basic testing scenario, the quadrotor hovers in an uncluttered environment, while an overhead camera system keeps track of its position and orientation. In this section we instantiate our multisensory fusion architecture for the quadrotor scenario. We provide an analysis of the sensory data, the network implementation, and finally a performance evaluation against ground truth (i.e. the camera tracking system) as well as the on-board EKF estimator.

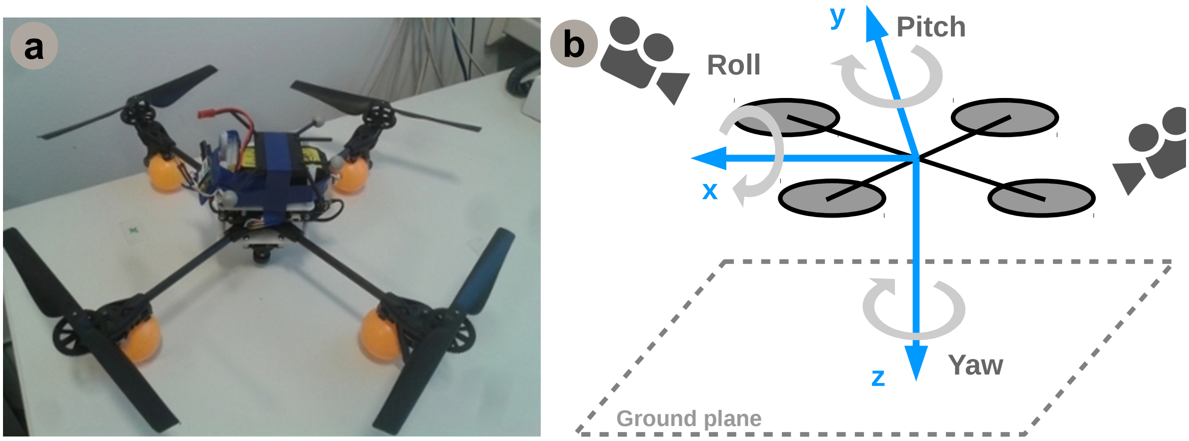


Fig. 6.1: Experimental setup: a) Quadrotor platform; b) Reference system alignment and ground truth camera tracking system.

Next, an analysis of the available sensory cues is provided to motivate the need for an adaptive learning and computational substrate for sensory fusion. The accelerometer on the quadrotor measures accelerations with respect to the quadrotor reference frame. As we generally cannot neglect linear accelerations, the accelerometer always measures a combination of linear accelerations and rotated gravitational accelerations, usually termed net linear acceleration. The net linear acceleration cannot be easily separated into its linear and rotational components, yet typical approaches in modelling and control of quadrotors impose a null linear contribution assumption which doesn't hold (while hovering). We sum up that, the accelerometer measures net linear accelerations, that linear accelerations cannot be neglected, that the accelerometer measures accelerations in x and y direction with sufficient accuracy and that the z component of the measured acceleration is erroneous. This is due to the fact that the z-axis of the accelerometer is always parallel to the thrust axis of the quadrotor so one might assume that thrust changes influence accelerometer measurements in that axis in a negative way. The accelerometer contributions are used in estimating roll and pitch, Figure 6.2b and Figure 6.3b. The gyroscope on-board the quadrotor measures angular velocities. We can integrate these angular velocities and get the roll, pitch and yaw (RPY) angles. The gyroscope is the best sensor to measure RPY angles as it measures angular velocities with high precision and reliability in a direct way. Nevertheless due to the RPY angles integration, the gyroscope tends to drift over time. The drift is partially caused by the properties of the gyroscope sensor and partially caused by the integration of noise and measurement errors. Gyroscope provides good contributions for all three degrees of freedom, Figure 6.2a, Figure 6.3a, and Figure 6.4a. Another sensory source on-board the quadrotor is the magnetometer. It measures the earth magnetic field which can be used to estimate yaw angles. High currents, characterising the rotors when manoeuvring the quadrotor, influence the magnetic field around the quadrotor such that the magnetic field measurements have errors. The magnetometer provides relatively stable estimates contributing to yaw angle estimation, Figure 6.4b. The quadrotor used in our experiments is also equipped with an optic flow sensor module. This sensor measures RPY compensated ground speeds and the height above ground, and uses a CMOS camera to recognize the flow of detected feature points on the ground and with that the ground speed of the quadrotor. A sonar is used to compute the height above ground. This optic flow sensor delivers reasonable measurements while being consistent to ground truth, but the data of the optic flow module is not reliable enough to use it without any additional feedback.

In our scenario (hover control) the quadrotor had fast and small amplitude changes in angles on the three axes, so that simple integration of the optic flow velocity output was not usable. In order to cope with this drawback and still use the flow contribution we used a multilayer perceptron to extract the nonlinear mapping from x and y direction velocities to RPY estimates [Requena-Witzig et al., 2015]. The optic flow contributes with estimates for full 3D egomotion estimation, Figure 6.2c, Figure 6.3c, and Figure 6.4c.

Using the available sensory streams, we instantiated our framework and implemented a model to extract motion components estimates in 3D space. Following similar relaxation dynamics as in the basic models in Chapter 3, as well as similar dynamics interpretation as in the 2D motion estimation (Chapter 4), we now introduce the architecture for the quadrotor scenario in Figure 6.5. We decouple the three motion components and consider different sensory contributions for estimating each degree of freedom. Given sensory data that mildly influences the activity in the network, gyroscope, optic flow, and accelerometer units, containing roll and pitch angle estimates, are mutually exchanging information

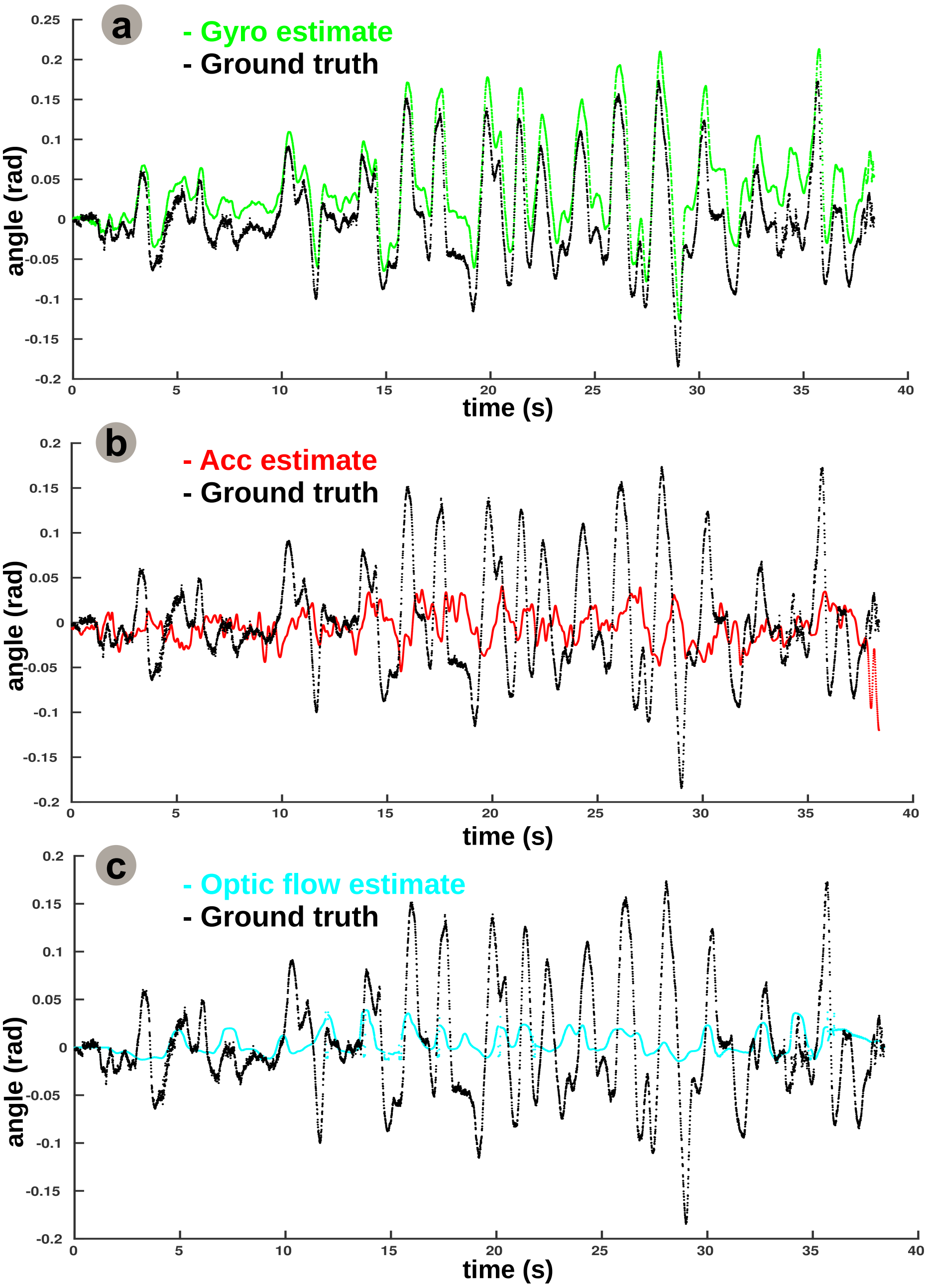


Fig. 6.2: Sensory data analysis for roll estimation: a) Gyroscope estimate; b) Accelerometer estimate; c) Optic flow estimate.

converging to a more precise estimate. This process is realised by taking steps towards minimising the mismatch among their local belief. Similarly, yaw estimates are continuously refined given new sensory samples (from gyroscope, optic flow, and magnetometer) and network's belief. In order to evaluate the performance of our network in terms of motion estimates precision, we compared it against the estimates provided by the ground truth system (3D camera tracking system), and against the on-board EKF attitude estimator. Using similar sensory contributions as the EKF on-board the drone, our model provides good results against ground truth. For roll estimation, the EKF tracks precisely the motion (RMSE: 1%) while our network underestimates on the negative angles due to

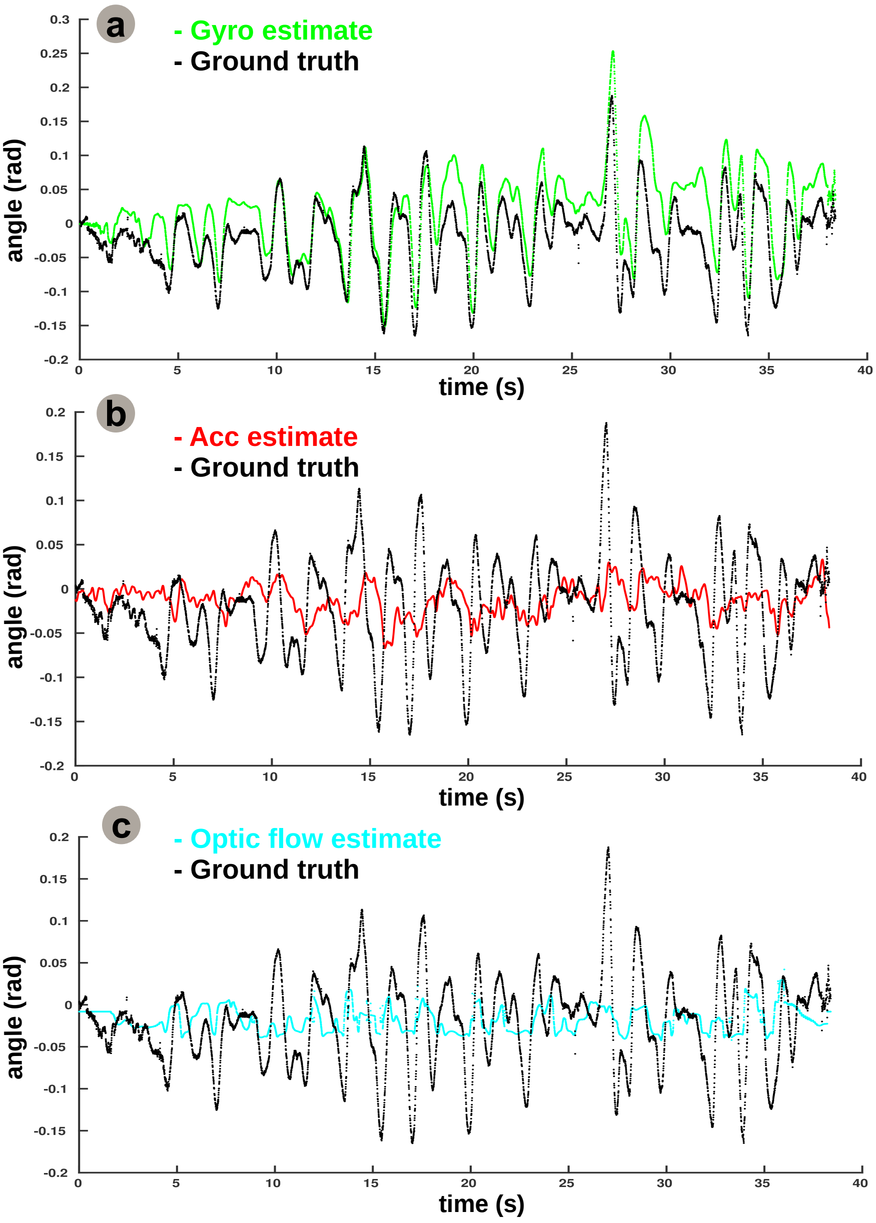


Fig. 6.3: Sensory data analysis for pitch estimation: a) Gyroscope estimate; b) Accelerometer estimate; c) Optic flow estimate.

the accelerometer and optic flow contributions (RMSE: 5%) which react with lower amplitude to the fast changes, Figure 6.6a. For pitch estimation the network overestimates (RMSE: 6%) on the positive angle values due to the gyroscope (drifting) contribution, yet balanced by a baseline provided by accelerometer and optic flow contributions, while EKF is relatively precise (RMSE: 2%) visible Figure 6.6b. Finally, for yaw estimation, optic flow information is noisy but provides a good trend, reacting to quadrotor's motion, Figure 6.6c. The network penalizes its contribution and enhances gyroscope's contribution, which is also supported by a stable magnetometer estimate (even if with an offset) such that overall the network performance (RMSE: 8%) is superior to the underestimating EKF (RMSE: 20%). The cause for the EKF performance penalty is given by the fact that, for yaw estimation, it heavily relies on the magnetometer. After demonstrating, through

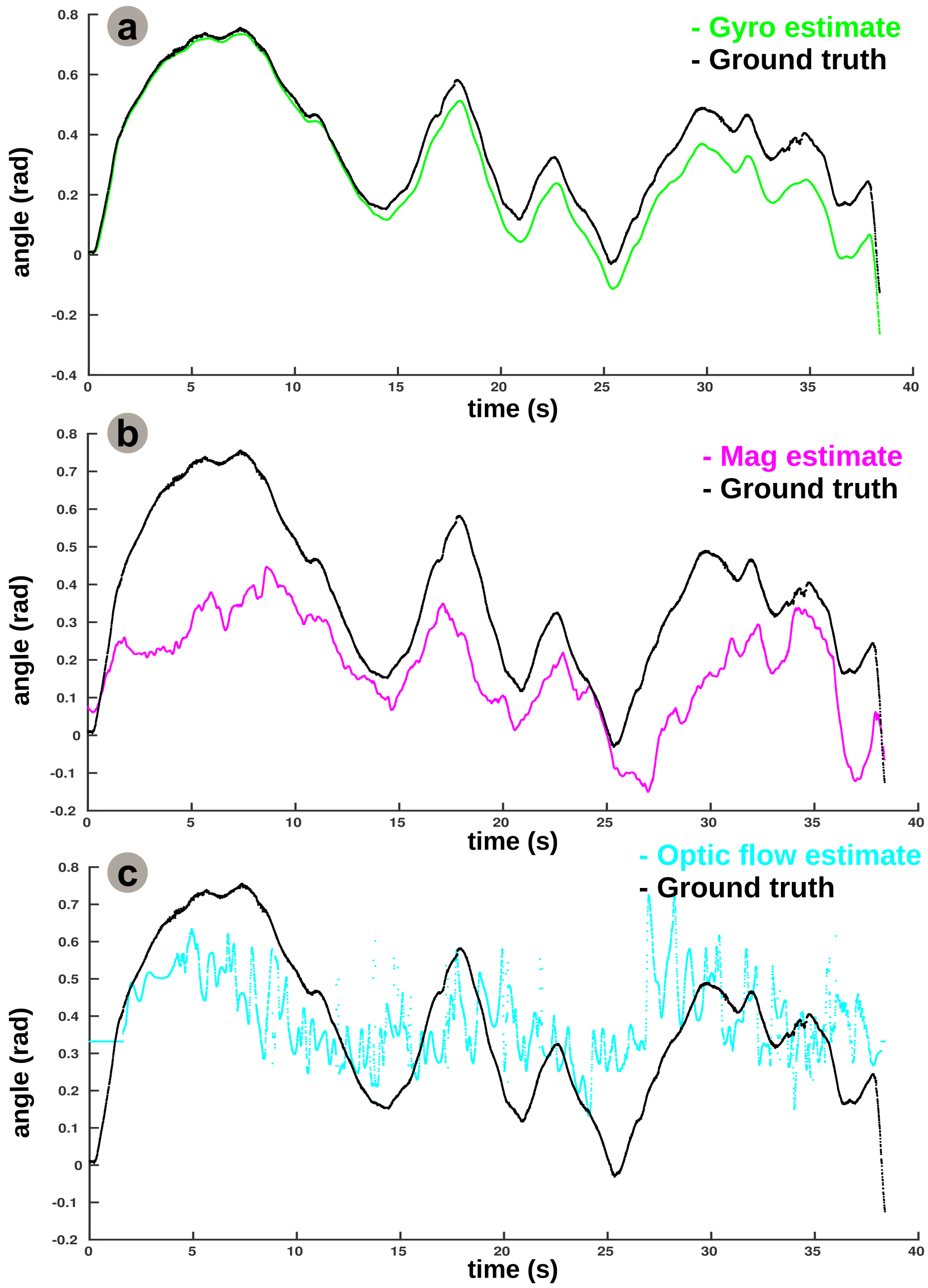


Fig. 6.4: Sensory data analysis for yaw estimation: a) Gyroscope estimate; b) Magnetometer estimate; c) Optic flow estimate.

the current instantiation of our framework, that our approach can provide precise state estimation, we next focus on probing learning mechanisms for extracting sensory correlations. Using the same scenario, we now do not consider embedding sensory relations in the network, rather we let the network learn them from the incoming streams of sensory data. We explore the capabilities of our approach and analyse its performance in the considered real-world scenario.

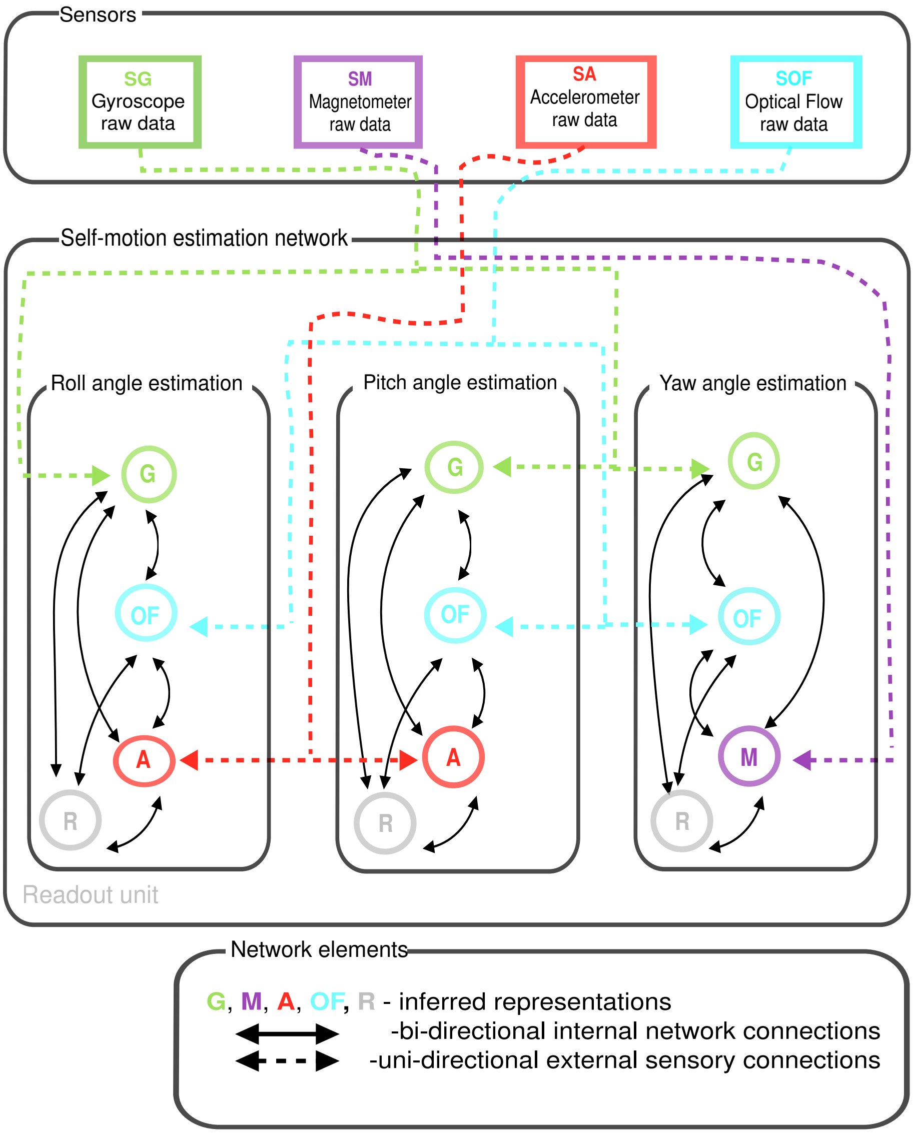


Fig. 6.5: Multisensory fusion network instantiation for quadrotor 3D egomotion estimation.

6.3 Learning sensory correlations for quadrotor 3D egomotion estimation

After analysing and providing insight on our framework capabilities to provide precise egomotion estimation in the case in which we complementary combine all available sensory data on-board, we now investigate the capability to learn the underlying sensory correlations. Learning the underlying correlations between the sensors alleviates the need for tedious modelling, parametrisation, and constraining assumptions. Due to its intrinsic learning capabilities, the network extracts sensory correlations which subsequently define multisensory fusion rules that the system uses to precisely represent its state and its environment.

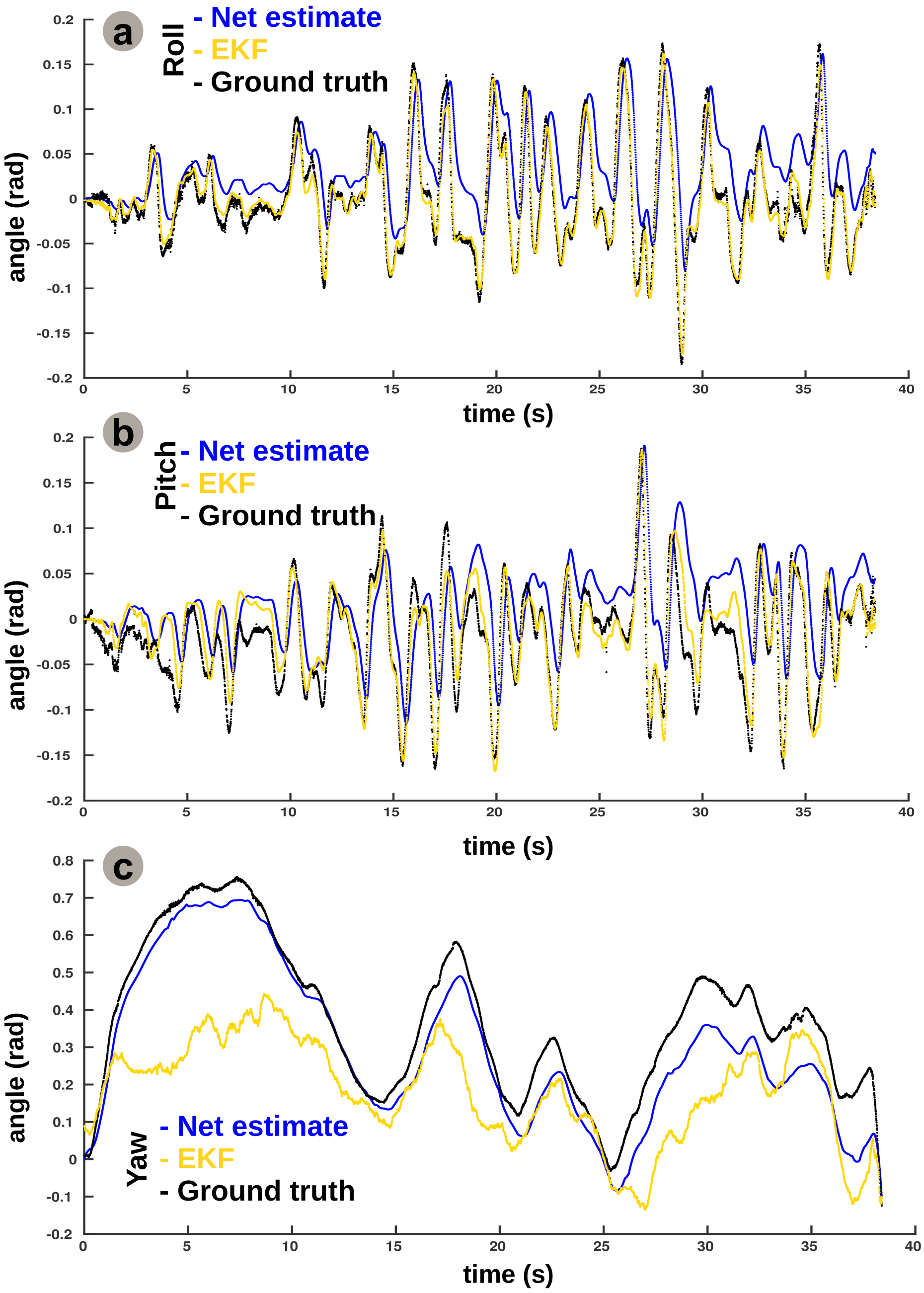


Fig. 6.6: Network performance analysis for 3D motion estimation: a) Roll angle estimate; b) Pitch angle estimate; c) Yaw angle estimate.

6.3.1 Network architecture and setup

After the quadrotor flight, preprocessed data from the available sensors (i.e. gyroscope, accelerometer, and a magnetic sensor, Figure 6.9 a) is fed to the model to extract the relations between the sensors for each of the three degrees of freedom (i.e. roll, pitch and yaw).

As initially discussed at the beginning of this chapter, and following principles of network creation and growth common in both graph theory and neurobiology, we extend our model, such that it is able to create its own structure given intrinsic structure of the input sensory streams. As a prior step to learning the sensor fusion rules, the system must learn which sensors can be associated for coherent estimates of each motion component. The basic idea is to determine which regularities in the different sensory streams are more informative to provide a good substrate for integration and enforce the connections between correlated sensors.

The model should make use of all available sensory information on-board the robot to build a model capable to learn and enforce sensory integration rules for precise egomotion estimation.

As previously postulated, physical systems are continuously and dynamically coupled to their environment. This coupling offers the system the capability to explicitly structure its sensory input and generate statistical regularities in it [Lungarella et al., 2005]. Such regularities in the structure of the incoming multisensory streams are crucial to enabling adaptation, learning, and development.

In our view, in order to use sensory correlations for integration, the system must extract the underlying regularities to determine an informative and valid combination. An interesting question is how to identify the origin of such regularities in the incoming sensory data streams. Self-generated motor activity brings an important contribution in shaping the informational structure and the quality of sensory information streams.

Sensory streams are not just optimised for efficient encoding and processing but are also well adapted to the structure of the environment or motion within it. Following this arguments, we identify the need for a quantitative characterization between the input sensory streams regularities and the processing mechanisms, such that the system can fully exploit available information.

Providing a practical approach to measure statistical regularities, dependencies, or relationships between sensory streams, information theoretic measures can be used to quantify statistical structure in real-world sensorimotor streams. This mathematical apparatus provides a generic and flexible analysis tool, as it can be applied to various levels (e.g. sensory signal, neural, behavioural) and at multiple time scales (e.g. learning, development, evolutionary). Taking advantage of its capability to provide a measure of uncertainty (or information), or in multivariate case, identify a nonlinear relation between multiple variables, entropy can be used to describe sensorimotor informational structure for our multimodal scenario.

Various methods for inferring the links among statistically coupled variables were developed, all founding their approach on statistical properties of observed variables: mutual information distance and entropy reduction [Villaverde et al., 2014], context likelihood of relatedness [Madar et al., 2010], or maximum relevance/minimum redundancy feature selectors [Meyer et al., 2014]. Providing a rigorous framework to address this issue, a large-number of the aforementioned methods use the information theoretic apparatus, but most of them focus on a particular type of problem, introducing various assumptions limiting their versatility.

In our approach we address the problem of recovering the structure of a network from available sensory data in its most general form, namely time-series streams of sensory data. No assumptions about the underlying structure of the sensory data are made and no prior knowledge about the system is taken into account. Furthermore, interactions between the various sensory streams are deduced from the statistical features of the data using information theory tools. This approach extends the generality of our framework for learning sensory correlations for multisensory fusion.

Let X denote a random sensory variable (e.g. sensory quantity) consisting of the set of possible samples $x_i, i = 1, ..., n$ with associated probability mass functions $p(x_i), i = 1, ..., n$. In order to transform sensory signals (given as time-series) into a set of discrete signals we partition the observation space into bins. The average amount of information gained from an observation that specifies X is defined by the entropy:

$$H(X) = -\sum_i p(x_i) log p(x_i). \tag{6.1}$$

Given sensory data as time-series we can estimate the probabilities, and hence the entropy, by binning the data. Unfortunately entropy estimates are dependent on the partitioning.

As we focus on extracting the structure from multiple streams of sensory information we can consider, in the simplest case, extracting the relative information between each two variables, without worrying about partitioning sensitivity. The joint entropy is a first measure which, based on Equation 6.1, can be defined for a pair of sensory variables (X, Y) as:

$$H(X, Y) = -\sum_i \sum_j p(x_i, y_j) log p(x_i, y_j). \tag{6.2}$$

Furthermore, given that $P(A|B) = \frac{P(A \cap B)}{P(B)}$, and X is measured and found to be x_i we can write the conditional entropy as,

$$H(Y|X = x_i) = -\sum_j \frac{p(x_i, y_j)}{p(x_i)} log \frac{p(x_i, y_j)}{p(x_i)}. \tag{6.3}$$

The average uncertainty of Y given x_i is provided by averaging $H(Y|X = x_i)$ from Equation 6.3 over x_i:

$$H(Y|X) = \sum_i p(x_i) H(Y|X = x_i) = -\sum_i \sum_j p(x_i, y_j) log p(y_j|x_i) = H(X, Y) - H(X). \tag{6.4}$$

As in our case we consider pairwise associations extraction, a good measure of the distance between two distributions, for example p and q, is the relative entropy (Kullback - Leibler divergence / information gain), defined as:

$$D(p||q) = \sum_i p(x_i) log \frac{p(x_i)}{q(x_i)}. \tag{6.5}$$

An important metric in our problem is the relative entropy (Equation 6.5) between the joint distribution $p(x_i, y_j)$ and the product distribution $p(x_i)p(y_j)$, which defines in fact the mutual information:

$$I(X, Y) = \sum_i \sum_j p(x_i, y_j) log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}. \tag{6.6}$$

Intuitively, mutual information is high if both sensory quantities have high variance (i.e. high entropy) and are highly correlated (i.e. high covariance). This metric provides the average amount by which a measurement of X reduces the uncertainty of Y, such that given Equation 6.4 we have:

$$I(X,Y) = H(H) - H(X|Y) = H(X) + H(Y) - H(X,Y) = I(Y,X). \tag{6.7}$$

Mutual information is symmetric and measures the amount of information one sensory variable contains about another. It does not assume any property of the dependence between variables, such that it is more general that linear measures (e.g. correlation coefficient) and is able to handle nonlinear interactions typically found in multisensory scenarios. Moreover, if two components of the network (i.e. sensory variables) interact closely (correlated statistical regularities) their mutual information will be large, whereas if they are not related their mutual information will be theoretically zero. The case in which the variables are statistically independent (i.e. mutual information is zero) is discarded in the considered scenario. This is due to the fact that all sensory streams on-board the quadrotor react to its 3-dimensional motion (readings are intrinsically coupled by motion).

In our framework, the main idea is to infer the network (of sensory variables) structure using a distance metric among variables. This metric is based on entropic measures of mutual information between time-series of sensory observations.

The core algorithm is relatively straightforward and is synthetically depicted in Figure 6.7. Initially, uni-dimensional, multi-dimensional (joint / conditional variables) entropies, and mutual information measures are estimated from sensory data, as shown in Figure 6.7 a. The estimates are subsequently used for calculating distances between variables and build a distance matrix.

In order to discriminate between direct and indirect (implicit) connections an entropy reduction (i.e. minimisation) step is applied [Samoilov et al., 2001], on conditional entropies, acting as a map refinement technique. The distance metric used for constructing the distance matrix is the Entropy Metric Construction (EMC) [Arkin et al., 1995, Samoilov et al., 1997], providing a minimum regardless of the possible time delays τ in the sensory data time-series:

$$d(X,Y)^{EMC} = min_{\tau} e^{-I(X(t+\tau),Y(t))}. \tag{6.8}$$

It is easy to see that high values of mutual information between variables determine a smaller distance value in the statistical relatedness space of variables' network, Figure 6.7 b, lowest-panel. Due to the fact that we need to infer network's structure from sensory data, knowledge about the underlying system cannot be used, so we need to estimate mutual information from the datasets instead of using the analytical form. Hence, taking advantage of the large number of sensory samples we binned the data in equally sized intervals and a function $\Theta_{i,j}$ counted the number of data points in each bin. Then, the needed probabilities are estimated from the relative frequencies of occurrence [Steuer et al., 1995],

$$\hat{p}(a_i, b_j) = \frac{1}{N} \sum \Theta_{i,j}(x_k, y_k). \tag{6.9}$$

As previously mentioned, we detect sensory variables interactions through an entropy reduction process, Figure 6.7 a. More precisely, we use an entropy minimisation mechanism, that seeks to determine variation in one sensory variable given variation in another sensory variable. The mechanism states that if a sensory variable X^* is connected to Y (which has

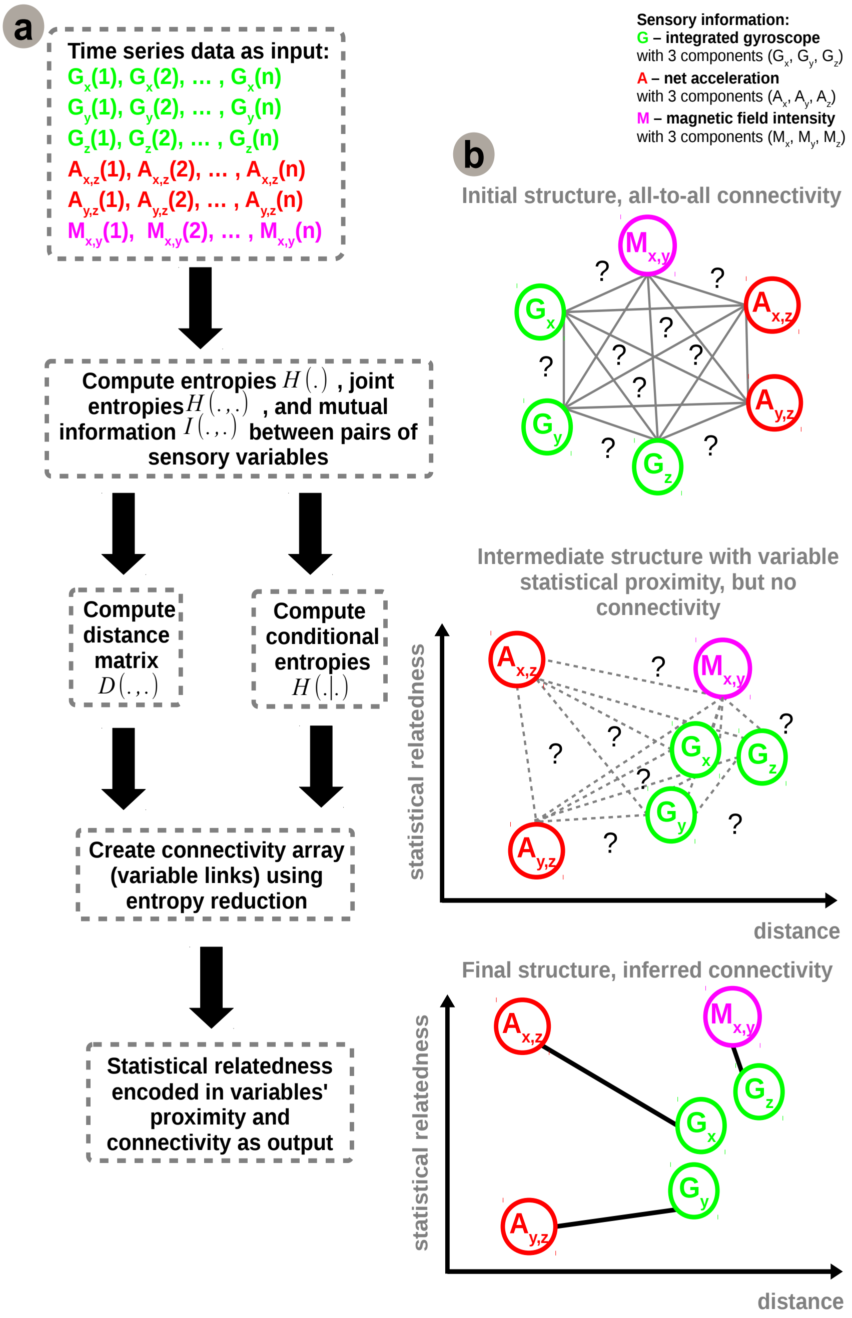

a
Time series data as input:
$G_x(1), G_x(2), \dots, G_x(n)$
$G_y(1), G_y(2), \dots, G_y(n)$
$G_z(1), G_z(2), \dots, G_z(n)$
$A_{x,z}(1), A_{x,z}(2), \dots, A_{x,z}(n)$
$A_{y,z}(1), A_{y,z}(2), \dots, A_{y,z}(n)$
$M_{x,y}(1), M_{x,y}(2), \dots, M_{x,y}(n)$
Compute entropies $H(.)$, joint entropies $H(.,.)$, and mutual information $I(.,.)$ between pairs of sensory variables
Compute distance matrix $D(.,.)$
Compute conditional entropies $H(.|.)$
Create connectivity array (variable links) using entropy reduction
Statistical relatedness encoded in variables' proximity and connectivity as output
b
Sensory information:
G – integrated gyroscope with 3 components (G_x, G_y, G_z)
A – net acceleration with 3 components (A_x, A_y, A_z)
M – magnetic field intensity with 3 components (M_x, M_y, M_z)
Initial structure, all-to-all connectivity
$M_{x,y}$
G_x
$A_{x,z}$
$A_{y,z}$
G_y
G_z
?
Intermediate structure with variable statistical proximity, but no connectivity
statistical relatedness
distance
Final structure, inferred connectivity
statistical relatedness
distance

Fig. 6.7: Network inference algorithm: a) Algorithm pipeline: feed time-series sensory input; compute statistics for individual and pairs of sensors (entropy and mutual information); compute statistical distance and conditional entropies to extract statistical relatedness; create connectivity array using entropy reduction (minimisation); b) Network structure evolution: Initial connectivity; Intermediate statistically clustered variables; Final structure and inferred connectivity.

already been predicted to be connected to a subset X_s^* of X^*), its inclusion in the network structure must reduce the entropy by a proportion at least equal to a threshold T. The threshold T is computed as a function of overall entropy values. Hence, a link between X^*

and Y is predicted if and only if the entropy reduction $E_R(Y, X^*)$,

$$E_R(Y, X^*) = \frac{H(Y|X_s^*) - H(Y|X_s^*, X^*)}{H(Y)} > T. \tag{6.10}$$

In order to obtain reliable estimates of joint entropies of the many sensory variables, the large amount of data observations provides an advantage. Furthermore, exploiting the rich input space, the proposed algorithm is able to exploit the intrinsic statistical regularities of the sensory data to generate a plausible network configuration, Figure 6.7 b. Analysing individual statistics, from the perspective of each variable with respect to all the others, we notice that the network configuration generated by the algorithm, Figure 6.8 a, is supported by estimates of mutual information, Figure 6.8 b. Although initially the network considers all sensory contributions for the estimation of all motion components, as shown in Figure 6.7 b, it will enforce only those connections providing a coherent correlation for each degree of freedom, as shown in Figure 6.8 b, based on the resulting configuration from the network inference algorithm. Using only the underlying statistical regularities and information content in incoming sensory streams, the algorithm is able to detect, to subsequently connect sensory contributions which are informative for estimating the same degree of freedom, and to, finally, combine them into motion estimates through our fusion mechanism, as depicted in Figure 6.9 c. For roll and pitch angles (i.e. rotation around the x and y reference frame axes), the network learns the relation between the roll and pitch angle estimates from integrated gyroscope data and rotational acceleration components (i.e. orthogonal x and y with respect to z reference frame axes). Similarly, the yaw angle is extracted by learning the relation between the yaw angle estimate from integrated gyroscope data (i.e. absolute angle) and aligned magnetic field components from the magnetic sensor (i.e. projected magnetic field vectors on orthogonal x and y

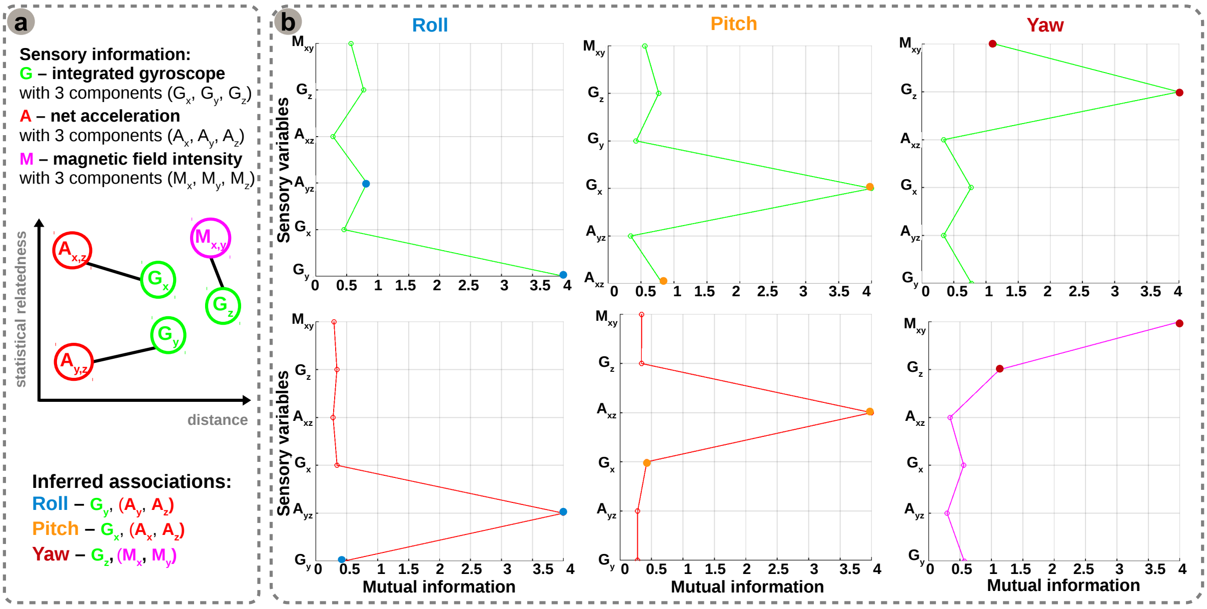


Fig. 6.8: Network inference analysis: a) Sensory data, inferred network structure, and associations for each motion component; b) Individual estimates of mutual information, on a per sensory variable basis, motivating the established network connections for sensory associations.

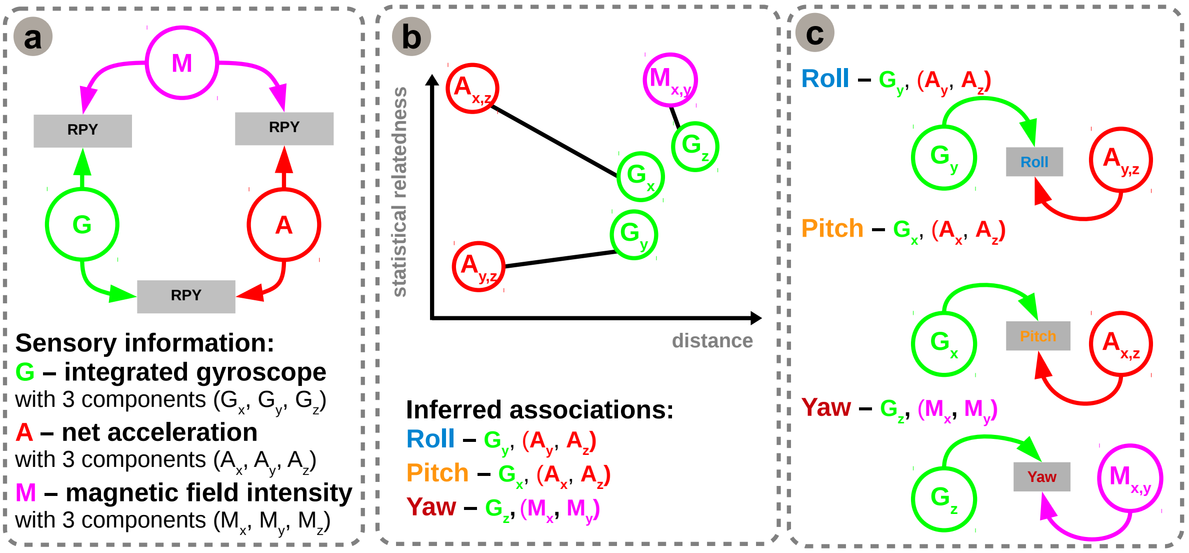

a
M
RPY
RPY
G
A
RPY
Sensory information:
G – integrated gyroscope
with 3 components (G_x, G_y, G_z)
A – net acceleration
with 3 components (A_x, A_y, A_z)
M – magnetic field intensity
with 3 components (M_x, M_y, M_z)
b
statistical relatedness
distance
$A_{x,z}$
$M_{x,y}$
G_z
G_x
G_y
$A_{y,z}$
Inferred associations:
Roll – G_y, (A_y, A_z)
Pitch – G_x, (A_x, A_z)
Yaw – G_z, (M_x, M_y)
c
Roll – G_y, (A_y, A_z)
G_y
Roll
$A_{y,z}$
Pitch – G_x, (A_x, A_z)
G_x
Pitch
$A_{x,z}$
Yaw – G_z, (M_x, M_y)
G_z
Yaw
$M_{x,y}$

Fig. 6.9: Network instantiation for 3D egomotion estimation: inferred network structure and sensory associations for learning. a) Sensory configuration of the robot; b) Inferred network connectivity; c) Sensory associations for learning.

reference frame axes). The learned sensory associations are not arbitrary, but rather represent the dynamics of the system and are consistent with recently developed modelling and control approaches for quadrotors [Hyon et al., 2012, Lee et al., 2012]. To make use of the learned relations we decode the Hebbian connectivity matrix using a relatively simple optimisation method [Brent, 2013]. After learning, we apply sensory data from one source and compute the sensory elicited activation in its corresponding (presynaptic) SOM neural population. Furthermore, using the learned cross-modal Hebbian weights and the presynaptic activation, we can compute the postsynaptic activation. Given that the neural populations encoding the sensory data are topologically organised (i.e. adjacent values coding for similar places in the input space), we can precisely extract (through optimisation) the sensory value for the second sensor given the postsynaptic activation pattern. Without using an explicit function to optimise, but rather the correlation in activation patterns in the input SOMs, the network can extract the relation between the sensors.

6.3.2 Experimental results

In order to validate the extracted relations, we use the aforementioned mechanism to extract the roll, pitch, and yaw estimates for the quadrotor scenario. Figure 6.10 presents a decoupled view for each degree of freedom, depicting the learned relations and estimation accuracy. We observe in Figure 6.10 a that the learned relations resemble the nonlinear functions (i.e. arctangent) used in typical modelling approaches, although preserving irregularities in the cross-sensory relations. The learned cross-sensory relations, encoded in the Hebbian matrix, provide the intrinsic constraints between the sensory cues contributing to the estimate of each degree of freedom.

For roll estimation, Figure 6.10 b upper panel, the network learns the relation between net rotational acceleration provided by the accelerometer and the absolute roll angle es-

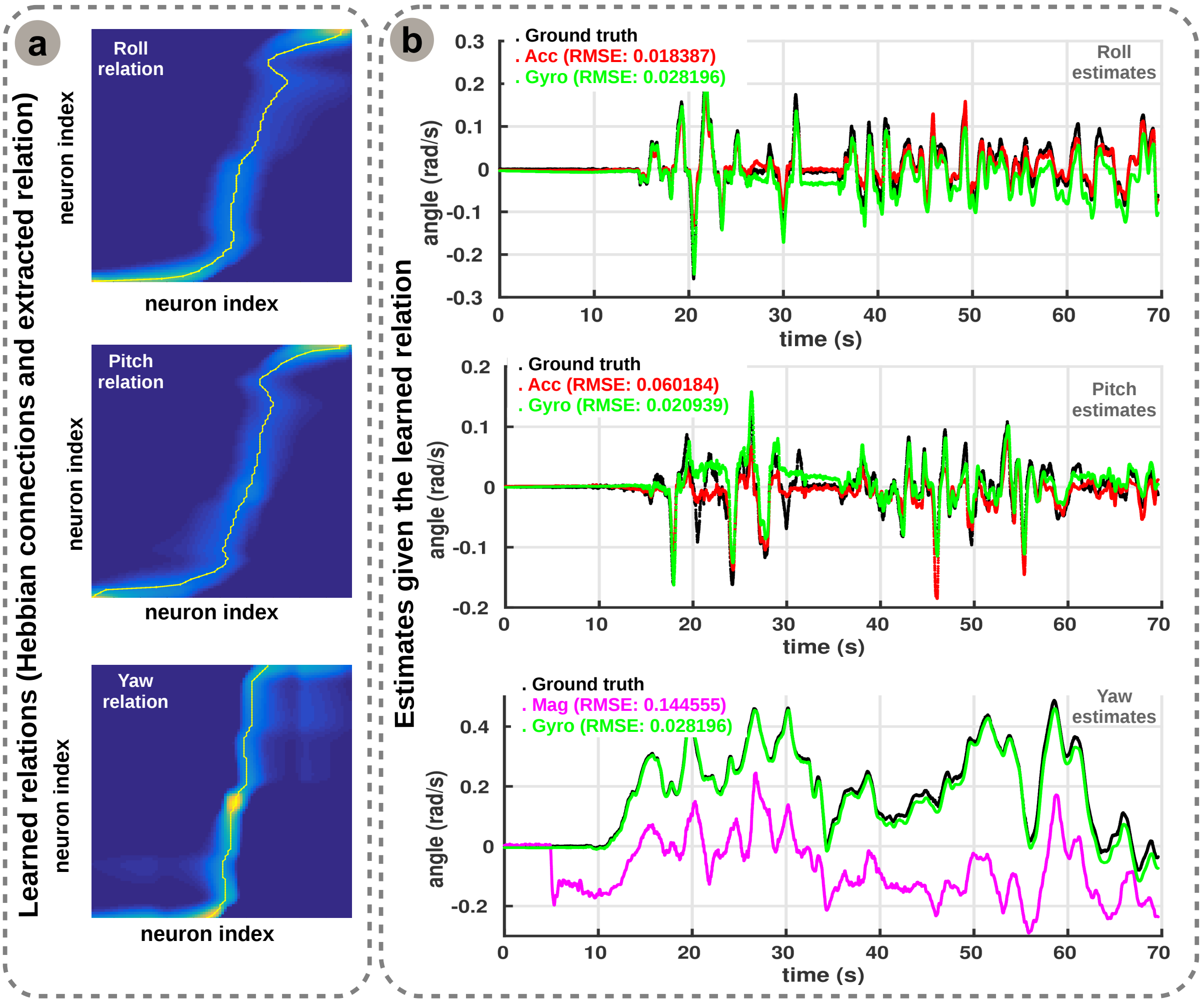


Fig. 6.10: Network instantiation for 3D egomotion estimation: a decoupled view analysis. a) Learned relations; b) Estimation quality using learned relations.

timate provided by the gyroscope. Given that accelerometer data is noisy and gyroscope data drifts, as a consequence of integration process, the network is able to ”pull” the values of the two cues towards the ”correct” value of the roll angle as given by ground truth (accelerometer RMSE: $< 2\%$, gyroscope RMSE:$< 3\%$).

For pitch estimation the network extracts the nonlinear dependency between the accelerometer data and the gyroscope data. Although both cues follow the trend of change in angle, as shown in Figure 6.10 b middle panel, the accelerometer is overestimating, due to the noisy signal and the overall limited motion of the drone on this axis. The gyroscope contribution was able to modulate the accelerometer contribution such that the overall estimates are acceptable (accelerometer RMSE: $< 7\%$, gyroscope RMSE: $< 3\%$).

Finally, for yaw estimation the network uses the gyroscope absolute angle and the magnetometer contribution, based on magnetic field readings on the other two axes. Interestingly, albeit the fact that the yaw estimate of the magnetometer follows the trend, Figure 6.10 b lower panel, there is an intrinsic offset (RMSE:$\sim$ 15%) visible from $t = 5s$. Investigating during many test flights, we noticed that the current change generated when arming the rotors introduced a significant modification in magnetic field distribution, subsequently reflected in the magnetometer readings. In the current setup, the inferred network is not able to explicitly compensate for the offset, as one can see in Figure 6.10 a lower panel, where co-activation pattern is not sharp like for roll and pitch.

As our results show, the model is able to extract the underlying data statistics without any prior information, as shown in Chapter 5, where the sensory data distribution was learned directly from the input data. Moreover, following the statistics of the data, the network allocates more neurons to represent areas in the sensory space with a higher density such that the cross-sensory relations are sharpened, visible in Figure 6.10a.

As also shown in Chapter 5, there is no specific parameter tuning routine to handle different kinds of input data for different scenarios. The generic processing elements (i.e. SOM, Hebbian learning) and their extensions (i.e. tuning curve adaptation, covariance update) ensure that the network first learns (in an unsupervised manner) the structure of the data, and then uses this representation to sharpen its correlational structure. Moreover, given the learned relations, the network is able to infer missing quantities in the case of sensor failures. As the relation is encoded as a synaptic weight, after learning, it is enough to provide samples from one sensor, encode them in the SOM, and project the activity pattern through the Hebbian matrix. The resulting activity pattern, subsequently decoded, will provide the missing real-world sensory value.

6.4 Summary

Given relatively complex and multimodal scenarios in which robotic systems operate, with noisy and partially observable environment features, the capability to precisely and rapidly extract estimates of egomotion critically influences the set of possible actions. Utilising simple and computationally effective mechanisms, the proposed model is able to learn the intrinsic correlational structure of sensory data and provide more precise estimates of egomotion. Initially, the model extracts the sensory associations from sensory streams, by exploiting the statistical regularities underlying time-series data, using information theoretic metrics. The learned associations determine a network structure connecting all sensory variables such that consistent associations between variables are realised for each motion component estimate. Furthermore, in order to combine sensory contributions, given extracted associations, the model uses competition, cooperation, and sensory data to extract correlations and encode them in a distributed pattern of neural activity. These correlations are subsequently decoded and provide the multisensory fusion constraints (rules), such that each sensor is pulled toward "plausible" values, ensuring that the network converges to consensus. Settled in a stable state, the network provides precise individual motion estimates, as perceived from each contribution sensory modality.

Being able to learn sensory data statistics and distribution, the model judiciously allocates resources for efficient representation and computation without any prior assumptions and simplifications. This ensures that all the individual components of the framework interact to increase its generality. Alleviating the need for tedious design and parametrisation, it provides a flexible and robust approach to multisensory fusion, making it a promising candidate for robust robotics applications.

7 Discussion and conclusions

Natural organisms and technical systems alike are continuously and dynamically coupled to their environments, with incoming sensory streams determining motor actions, and motor activity selecting and modulating the statistics of the sensory input. Despite environment's influence upon the system, the later itself "structures" sensorimotor information by coordinated and dynamic interaction with the environment.

By providing a global overview over the problem of multisensory fusion in Chapter 2, we are able to capture those relevant design aspects driving robust and flexible implementations in both natural and engineered systems. As our aim is to provide a generic framework in which a system can autonomously learn its sensorimotor capabilities and use them for precise interpretation of the environment, we identify those principles able to offer a representation and processing framework, simple enough to be generalised and robust enough to cope with real-world data.

It has been postulated that both biological and artificial systems refine their adaptation capabilities and are able to robustly represent, and interact with, their environment. In order to disambiguate their perception, they use a complex pattern of interactions to act upon the environment, which reciprocally influences their state. Furthermore, these interactions underline the need for an adaptive processing substrate to handle incoming perceptual streams, usually unfolding as a rich and noisy multisensory percept.

Maintaining a coherent internal representation of the environment and own state, given complementary percepts of the environment, is by far a non-trivial trivial task and certainly expects considerable adaptive capabilities from the system. Multisensory fusion defines the process responsible of combining information from the variety of sources of information available to the system, in order to provide a robust and complete description of the environment and/or own state.

Identified as a long sought goal in all engineering implementations aiming at autonomy, multisensory fusion techniques met various design approaches. Using various architectures, sensory data representations, mathematical apparatus, and aiming at different perceptual or decision outcome, these methods generated a wealth of design strategies and possibilities. Despite the broad range of methodologies and mechanisms, a generic recipe to identify, understand, and exploit available sensory streams is not yet defined.

Indeed, as we saw in Chapter 2, state-of-the-art methods employ different strategies to solve the problem of making sense of available sources of information to plan actions. More precisely, the two tasks that we address are state estimation and data association. In order to disambiguate the scene and perform precise state estimation, the system must combine all available sources of information in an advantageous way. Furthermore, in order to maximise the contribution of each source of sensory information, the system must capture the underlying sensory correlations. Using only informative contributions enhances estimates subsequently supporting reliable decision making.

In order to address the aforementioned aspects, current multisensory fusion systems approach the problem from different levels, so as to (potentially) provide a generic solution.

Considering different relations between input data sources, as shown in Figure 2.2, a multisensory fusion system is able to obtain more precise estimates by exploiting complementarity, redundancy, or cooperation between sensory contributions. Other approaches, exploiting sensory contributions at different abstraction levels, use low-level sensory features to build high-level inference used for fusing contributions, Figure 2.3. Given different type and nature of the input data, some state-of-the-art systems detach from the low-level signal noisy domain, build associations and high level representations to disambiguate the percept, and even infer missing or new quantities, as shown in Figure 2.4. Finally, all state-of-the-art build their approach on relatively different processing schemes, some to exploit data representation, whereas others to obey the physical (spatial and / or temporal) constraints of the system. Providing solutions for particular systems, centralised and decentralised processing architectures, Figure 2.5 a, b, are overtaken by more robust approaches using distributed schemes, Figure 2.5 c, capable of more robust, flexible, and still advantageous processing.

The in-depth analysis of state-of-the-art approaches to multisensory fusion as well as the known mechanisms in computation neuroscience, allow us to capture important design principles. Supported by real-world implementations and a thorough comparison with state-of-the-art approaches, we validate our approach as an alternative to existing methods for state estimation and data association.

Our perspective and motivation comes from analysing common approaches for multisensory fusion methods, their limitations and advantages on one side, and the superior nervous system's performance on the other side. We are interested in the capability of robustly combining available senses, given the noisy and uncertain environment. Both biological and technical systems need the capability to disambiguate perception by using different sources of sensory data. In our view this yields a coordinated interplay of available senses such that, given sensory observations, the system compensates for uncertainty and noise, and exploits the redundancy of sensory measurements.

Despite the capability to disambiguate their state, systems have to infer new quantities from existing sensory observations, given underlying causal relations. Another important aspect refers to the capability to optimise efficient processing of incoming sensory streams, such that the systems should constantly generate predictions about future events, or anticipate them. This allows the system to infer temporary degraded or missing information in sensory modalities.

Of major importance is the capability of the multisensory fusion scheme to handle inconsistencies and imperfections, assigning judicious confidence levels to contributing quantities. Usually, this is performed prior to system's design allowing it to provide good results for the considered (dedicated) scenario. If during operation system's parameters change, the scheme is not able to properly judge the validity of incoming streams. This yields a generic adaptive substrate which alleviates the need for considering constraining assumptions at the design stage, and turn the system overall more robust.

Although the goal is to exploit the multisensory structure of the environment and / or own state, the different available sensors bring heterogeneous streams, yielding the need for a system able to align multiple scales and representations to improve precision of the estimated features. From a computational point of view, most approaches aiming at real-time solutions approach multisensory fusion need to follow a distributed perspective. Distributing processing and representation provides a powerful paradigm for multisensory fusion, such that fragmented, locally coherent representations from different sensory contributions, enable a global consistent representation of the scene. Thus, combining local preprocessing, alignment, association, and estimation of individual modalities allows a distributed scheme to split global processing in relatively simple local processes which mutually interact to keep representations coherent, Figure 2.5 c.

Our model follows a similar structure, in which different sensory modalities are represented in a distributed network of interacting processing units capable of exchanging local information, such that a global, more precise, representation is obtained. Using relatively simple computation, given by the physics of the sensors (e.g. simple algebraic functions), our model combines individual sensory contributions, described by different reliabilities, noise patterns and uncertainty.

Finally, another core aspect assumes that, a multisensory fusion system needs to take into account and exploit the diversity in the sensory data, and extract spatio-temporal associations from it. Indeed, in order to exploit sensory contributions towards obtaining a precise representation, most informative sensors must be combined such that the result is a more precise estimate than individual sensors considered in isolation. Our model is capable of detecting regularities in available sensory streams, combine those which are highly correlated, extract their correlation pattern, and finally use the learned correlation to fuse them.

Proposing a new computational paradigm, our model finds its inspiration and advantages in (neuro-)biological substrate, following processing principles which differentiate it from traditional approaches to computation. We propose an alternative computational architecture, inspired by the high-level architecture of the mammalian cortex, where computation is performed in a widespread network of interconnected units, each representing a different type of sensory information measuring a feature of its environment or own state. In the basic model formulation, connectivity between the processing units implement known formalized relations (in fact equations) and computation takes place by each unit trying keep consistency with the other units it is connected to. This novel approach to computation ensures that the dynamics of the network follows the constraints, imposed by the sensors and / or the perceived environment, to reach consensus. A stable state is reached when the system settles in a solution providing a consistent representation of all the sensed quantities.

In our framework processing and storage are both local and intermeshed, such that each processing unit in the network has a local understanding of the perceived quantity. The local belief of a unit builds upon its corresponding sensory contribution and the constraining contributions from other units in the network. Each sensory modality available to the system is individually represented in the network, as either a point estimate (i.e. scalar) or a sparse representation (i.e. population coded), yet obeying same local dynamics.

As postulated by various studies ranging from computational neuroscience to neuropsychology, relational knowledge and representation, which describe associations amongst sensory signals, is a hallmark of human cognition. We base our design on a framework considering relations as the main driver for the dynamics and connectivity of a system, capable of providing a robust representation of the sensory space through the combination of all available sensory streams. Supported by knowledge from both computational and experimental neuroscience, we extract those principles known to explain sensory processing in cortex, and use them in our design.

The core formulation of our proposed style of processing starts with the idea that large-scale networks processing sensory information are based on parallel processing, and coherent representations are achieved through efficient coordination of information transactions. Locally, each sensory variable is represented through an area responsible to represent and process its incoming information streams. The input from other areas provides only a small fraction of the input to a target area. Furthermore, due to the highly interconnected networks and ongoing dynamic computation, a large number of competing constraints acting on their component areas must be solved rapidly. This process is described by a relaxation mechanism, which is able to avoid falling and settling in local minima, by reconciling competing constraints through increased relative coordination of the interacting areas.

Our work abstracts from neural models to a practical implementation. We propose a framework for multisensory fusion which assumes interactions between percepts in order to extract globally coherent representations given modalities' local interpretations. More precisely, we build a network of possibly conflicting local interpretations, which by using relaxation to solve the inherent constraints, ensures convergence to plausible and possible global interpretations. The model represents knowledge and constraints between percepts as a network of relations in which each one involves a given number variables representing sensory inputs, such that any overall relationship amongst the variables treated in the network is distributed across the network. This approach ensures that global knowledge representations can be extracted from local interpretations and interactions.

In order to test the capabilities of our model, we instantiate it for two simple scenarios, using two networks embedding simple algebraic relations, Figure 3.4, and more complex highly nonlinear trigonometric relations, Figure 3.8, respectively. Relaxing towards a stable state in which all relations are fulfilled, the network dynamics is able to rapidly compensate the initial mismatch (given by random initialisation) without external sensory contributions, Figure 3.5. Each of the network units encodes a 1-dimensional (scalar) representation of a real-world value. Using a relatively simple gradient update rule, each unit takes steps towards minimising the local mismatch between its estimate and the estimates of the units connected to it. In a slightly constrained scenario, shown in Figure 3.6, the network is coupled to external input such that each unit has an additional constraint, and the overall network has less degrees of freedom. Due to external inputs, the network balances the contributions and accommodates new data updating its internal belief. New values are propagated through the network which updates its state towards fulfilling the embedded relations. Although each unit receives an additional source of information, the update dynamics for each source obeys same rules.

Testing fully constrained scenarios, allows us to analyse the robustness of the network in the presence of conflicting external inputs with respect to the internal network belief, Figure 3.7. Using the same network structure as in Figure 3.4, we connected external inputs simultaneously to all units in the network. Due to the fully connected external inputs, the network balances the contributions against its own stable belief in a rather oscillatory pattern. As mentioned, the network follows a relaxation process (allowing each unit to update) which explains the visible oscillations each unit's estimate has with respect to the relations it is involved in. The oscillations amplitudes are proportional to the mismatch between subsequent updates from different sources.

The second testing scenario,depicted in Figure 3.8, brought more interesting insight in the network capabilities. Using a mixture of power-law and trigonometric functions, we explore network's robustness and stability given highly nonlinear dynamics imposed by the relations. Starting from initial random conditions, the complex network converges to a solution given the mathematically constrained functions in the relations. When all sensory connections are enabled, the network oscillates due to fully constrained space of values its units can take. We observe high jumps in the mismatches values mainly related to the nature of the functions, and the fact that units' values are updated continuously, effect visible in Figure 3.9. Once freely evolving driven by internal dynamics, the network settles again in a stable state.

Coming closer to real-world sensory data regularities, we investigate network's dynamics when we also have temporal relations embedded in the network, as shown in Figure 3.10. Typically encountered in sensory data, temporal integration provides the means to extract absolute changes of a quantity given raw sensory data. This process is not perfect, as integration propagates errors and leads to drift. Due to network's internal coupling and interactions, this behaviour is avoided. Moreover, we observe that in the presence of external input (e.g. rate of change) the integration unit accumulates incoming samples without drifting, as shown in Figure 3.10 a. This capability is provided by the other relations in the network, which constrain the possible values a unit can take, thus providing a baseline for the integration unit to cancel out drift.

In a another experiment we thoroughly analyse the underlying adaptation capabilities of the network to handle conflicting incoming streams of information. As a main feature of our model, at the unit level all incoming contribution (sensory / other units) are weighted, such that consistent contributions are enhanced, whereas inconsistent contributions are penalised. The weighting mechanism (i.e. confidence factor) is local, and it quantifies the mismatch of a source of information with respect to all the others.

Although we limited ourselves in providing rather small scale systems for analysis, we also investigated other features like scalability and fault tolerance. These features are really important in real-world applications. As shown in our initial instantiations, the network can take an arbitrarily large size, encoding arbitrarily complex relations, and arbitrary connectivity patterns between units. Using simple and general update rules, the network can be flexibly extended, as local dynamics ensure global consistency between the relations in the network. Accounting as a flexible constraint satisfaction framework, the network's unit level processing and storage ensure seamless extensibility capabilities.

A second powerful feature of the network is fault-tolerance. As mentioned earlier, the network employs an adaptive mechanism (confidence factor) to weight incoming contributions. This mechanism provides also a substrate for fault tolerance. As an outcome of the network dynamics, all units relax together, so that if a sensor provides conflicting data the collective computation (network belief) will locally analyse and weight its contribution such that its value is considered only when consistent with the rest of the data in the network.

In order to test and validate these principles in real-world scenarios, we instantiated our framework for various robotic scenarios, from 2-dimensional egomotion estimation for a omnidirectional robot, to 3-dimensional orientation and attitude estimation for quadrotors. Investigation was also extended with an analysis on the parallelisation capabilities of the model on traditional PCs, MCUs, to massively parallel neurmorphic hardware.

Probing high-level processing and organization principles of multisensory fusion known to take place in cortex, our work proposes a flexible framework validated in multiple instantiations targeting technical systems. Our approach develops a general solution for the problem, since each unit of the model is able to represent a different sensory modality, and extended networks can embed even more types of sensory information, for a rich environment representation. As shown in the in-depth analysis carried out in Chapter 3, the proposed framework provides a solution to multisensory fusion, employing a new style of information processing that is more robust to noise, sensory failures, and uncertainty. Using a distributed processing scheme based on localized intelligence that ensures asynchronous information exchange and adaptation based on external real-world sensory stimuli, the framework ensures the design of fast, robust, and scalable computational architectures appropriate for real-time real-world robotic applications.

An initial instantiation of our framework targeted the design of a multisensory fusion network for an omidirectional wheeled mobile robot egomotion estimation in 2D space. The system's goal was to provide precise estimates of mobile robot 2D egomotion, a combined rotational and translational displacement of the robot with respect to the environment. Our approach used a distributed network in which independent neural computing nodes obtained and represented sensory information, while processing and exchanging exclusively local data, to infer an estimate of robot orientation and position in 2D space. This was achieved by rapidly solving a large number of mutually imposed (physical) sensory constraints which led to globally coherent estimates. Sensory constraints define an internal model providing a prediction of possible sensor values. This prediction is subsequently integrated with acquired sensory observations within the network which is inferring a belief about the perceived motion components. In order to compute motion estimates, the model used all available sensors on-board the robot: an inertial measurement unit, consisting of 3-axis gyroscope, 3-axis magnetometer which acts as vestibular input; wheel encoders acting as proprioceptive input; motor driver providing an efferent copy of the motors' PWM signal; and a camera for visual input.

Raw sensory data was fed to the network, which updated its internal belief and inferred an estimate of robot's position and orientation locally, as seen from each sensor perspective, Figure 4.4. For inferring a heading estimate, the network was fed with data from gyroscope, magnetometer (compass), wheels encoders, and camera, whereas for position estimation it used data from wheels encoders, camera, and a motor PWM signal copy. Basic mathematical relations link different processing units representing different sensors, and enable feed-forward and feedback connections for information exchange. Given noisy input sensory data, the network kept all local units' estimates in agreement, as shown in Figure 4.8 e, f and Figure 4.13 b. In order to increase flexibility, sensory data preprocessing (e.g. integration, offset subtraction) was performed inside the network, such that sensory contributions are aligned to a common representation (i.e. absolute heading angle or Cartesian position). Mutual influence between units encoding an estimate of heading angle was modulated by the confidence factor such that each interaction pathway of an unit had an associated confidence factor adapting according to the level of trustworthiness of a source of information to which the unit was connected, as shown in Figure 4.9 b-e.

In order to assess the fault tolerance capabilities of our network we performed a set of experiments in which we tested the network in the presence of faulty sensors. As sensory data mildly influences ongoing network activity, in the absence of one sensory contribution the network can recover it based on the other modalities and its connectivity, as shown in Figure 4.16 b for magnetometer failure. This fault tolerance mechanism allowed the network to provide good estimates, Figure 4.16 b, c, given that temporarily the sensor didn't provide any measurements, such that its value was continuously inferred by the network given available other modalities.

In order to measure the performance of our model we compared it with two state-of-the-art methods: the Kalman filter and the Maximum Likelihood Estimator. Our model provided precise estimates of heading angle, position, and travelled distance comparable with state-of-the-art methods, Figure 4.14 c, d and Figure 4.15 c-f, but with less design assumptions and constraints. The RMSE was used as a metric to calculate the performance of our model against Kalman filter and MLE estimates with respect to ground truth data. The network was able to provide estimates for both heading angle (RMSE Heading: $\sim$ 10% , Figure 4.14) and position (RMSE Position: $\sim$ 1%, Figure 4.15) close to KF and MLE. Despite the fact that each source of information was affected by noise or systematic errors, the network was able to detect abnormal changes in sensory data, such that there was a small impact over its internal belief. Our analysis, using the mobile robot egomotion estimation scenario, was extended towards investigating network's parallelisation capabilities. In order to take advantage of the distributed processing scheme of the network, we explored the implementation on a series of computing platforms, from traditional PCs, to embedded MCUs, and finally, a massively-parallel computing platform, the SpiNNaker.

The overall results showed that due to its intrinsic parallelism, the network can take advantage of hardware parallelism, so that asynchronous exchange of local estimates between network units running on different cores is the most advantageous approach, as shown in Table 4.2. To quantify the performance, our experiments shown that the heading multisensory fusion (sub-network) takes around 6s for 5000 samples acquired at 25 Hz, while the multisensory fusion (sub-network) for position data estimation takes around 15s. Overall, the whole experiment took the robot approximately 198s, such that real-time multisensory fusion is possible. Furthermore, in order to evaluate the implementation of the proposed model, we also implemented a distributed version of a state-of-art method (i.e. DKF - distributed Kalman filter) combined with Covariance Intersection to infer heading and global position estimates from the different sensors available on-board the robot.

The parallel hardware implementation leveraged the capabilities of our network's architecture such that, in combination with the platform's event-based programming model, it provides a viable solution for real-time applications. Additionally, the low power consumption and form factor make it suitable for mobile applications. In another instantiation of our framework, we considered a more complex scenario, 3D egomotion estimation on a quadrotor, synthetically depicted in Figure 6.1. In this scenario, our model was able to provide precise estimates for roll, pitch, and yaw angles, given available sensory data on-board. Separating the three degrees of freedom, contributions from gyroscope, magnetometer, accelerometer, and optical flow were used to extract precise absolute angle estimates, using the network structure shown in Figure 6.5.

Using the process formally described in Chapter 3 and given sensory data that modulates the activity in the network, the gyroscope, optic flow, and accelerometer units, containing roll and pitch angle estimates, were mutually exchanging information to refine the local angle estimates. Similarly, yaw angle estimates were continuously refined given new sensory observations (from gyroscope, optic flow, and magnetometer) and current network's belief.

The performance of our network in terms of motion estimates precision, is quantified by the deviation from estimates provided by the ground truth system (3D camera tracking system), and evaluated against the on-board EKF attitude estimator. With good estimates for roll, pitch and yaw angles (RMSE Roll:∼ 5%, RMSE Pitch:∼ 6%, RMSE Yaw:∼ 8%) the network provided comparable performance with state-of-the-art methods given that no external source of absolute position was fed into the network, as results in Figure 6.6 show.

In order to extend the flexibility of our framework and alleviate the need for precise modelling and hand-crafting of the dynamics, we investigated learning processes which take place during the development of a biological nervous system. These processes enable it to extract mappings between external stimuli and its internal state. Employing such learning and development mechanisms can enhance adaptation and flexibility of our framework and its practical implementations.

Probing neural models of perceptual learning and development, we addressed the question of how can real-world sensory data be represented in a distributed neural substrate, such that its underlying structure and statistics can be exploited. Moreover, we were interested in how a system with relatively limited initial knowledge can learn and synthesize an appropriate processing infrastructure efficiently using the available sensory streams. This kind of system is able, by using relatively simple computational mechanisms, to learn efficient representations and make use of them for subsequent computation, aiming at coherently describing the environment and its own state given its sensory inputs.

As shown in previous implementations, sensory cues are correlated, and the underlying relations hidden in the data streams quantify their correlation level. Indeed, correlation is marked either by an explicit mathematical formulation or is just hidden in the data. In its more generic form, our framework is basically enforcing consensus by autonomously finding solutions to the constraints imposed by contributing sensors. Using a well studied and simple neural computation substrate, we extended the generic model in Chapter 3 by considering a distributed representation of the sensory space (instead of a point estimate) and replacing hard-coded relations through learned patterns of neural activity encoding the correlations.

In its basic formulation, our perceptual learning model extends the formulation introduced in Chapter 3 and instantiated in Chapter 4, as depicted in Figure 5.1. Samples from each input sensory modality are converted into a sparse representation (i.e. a SOM lattice of neurons) responsible for locally extracting the statistics of the incoming data and encoding sensory samples in a distributed activity pattern of component neurons, described in Figure 5.2. Each input SOM activity pattern is generated such that the closest preferred value of a neuron to the input sample will be strongly activated and will decay, proportional with distance, for neighbouring units.

To link the representations constructed in the input SOMs, we use a variant of Hebbian learning rule, such that our model is able to learn the underlying relation and encode it in a distributed pattern for easy readout. The correlation learning process is responsible for extracting the co-activation pattern between the input layers and eventually describe the hidden relation, as shown in Figure 5.2. Aiming at providing a solution for real-world implementations, self-organisation and correlation learning processes evolve simultaneously, such that both sensory representations and correlation pattern sharpening are continuously refined given incoming sensory observations.

Although our approach tries to extend the relational framework by adding learning capabilities, we address the general problem of extracting the underlying structure and correlations in various sensory streams. This problem is of high interest in real-world systems assuming robust environment interaction, as it is providing the means to obtain a more precise interaction with the environment. Various other methods for extracting sensory correlations were developed, spanning from probability theory to neurally plausible models. In order to evaluate our approach we provided a detailed investigation over: design and functionality; amount of prior information set by the designer in the system; stability and robustness of the obtained representation; capability to handle noisy data, capability to infer (i.e. predict / anticipate) missing quantities once the relation is learned; and capability to decode the learned representation and subsequently measure the precision of the learned representation. Our model is able to provide suitable solutions for all the considered aspects making it a good candidate for real-world implementations.

Initially focusing on the formal substrate and the integration within the computational framework introduced in Chapter 3, we thoroughly analysed the capabilities of our perceptual learning model using simulated data for various linear or nonlinear functions (relations) and input data distributions. Given incoming streams of correlated sensory data, each input SOM uses cooperation, competition, and adaptation to learn and represent input data statistics in a heterogeneous population code (visible in the number of allocated neurons and size of the tuning curves of neurons in Figure 5.12). Interestingly enough, after relaxing in a stable state, the network contains a fully informative representation of the input data and the learned sensory relation.

After learning has ended, given the ordered representation of the input data space onto the SOMs, one can find (decode) the corresponding real-world values given input patterns, comparatively described in Figure 5.13. Following this, obvious inference and fault tolerance capabilities are provided by the model without additional design considerations. After learning, the network stores a stable representation of the hidden relation between the sensory inputs considered during training. Furthermore, given one input sample from the input sensory stream, the network computes the elicited activity in the input SOM population. Finally, in order to extract the real-world value, a decoding mechanism based on an optimisation method is used to recover the corresponding value.

Another interesting feature is the capability of the mode to extend from dual modality to multimodal processing. This is highly relevant for real-world scenarios where more sensory cues can provide, through their combination, a more precise estimate than separate contributions. In a first approach we considered a 4-dimensional network with a tree shaped correlation structure. In this scenario our network was able to extract relatively sharp representations of the underlying relations between pairs of units in the network, as shown in Figure 5.14. In a second scenario we used a 3-dimensional network with a circular correlation structure, such that the network is fully constrained internally. Using similar dynamics with all our experiments performed in Chapter 5, the network was able to extract a sharp representation, with relatively no interference (even if a circular connection pattern was used) and, more interestingly, with the capability to compensate for boundary effects during network operation, as depicted in the sample network in Figure 5.15.

Our perceptual learning model for multisensory fusion is combining the timing and shape of activation patterns associated with different inputs in order to extract the correlational structure of the available sensory streams. After learning, the extracted relation is used to infer missing quantities or to detect anomalous or erroneous input signals given that the correct relation was previously learned. Finally, the learned relation can be decoded such that the real-world value can be recovered from the distributed activation pattern, to subsequently provide feedback to a motor controller.

In order to test the capabilities of our extended framework for perceptual learning for multisensory fusion introduced in Chapter 5, we instantiated it for a real-world 3D egomotion estimation on a quadrotor.

Given incoming streams of sensory data, our model extracts coherent sensory associations from provided time-series. Exploiting statistical regularities underlying sensory streams the model captured statistical relatedness such that sensory variables were coupled in a network structure offering a plausible interpretation. Using information theoretic approaches we propose an algorithm capable of inferring a network in which the distance among nodes indicates their statistical closeness and existing links are refined to distinguish between direct / indirect sensory interactions. Without using a priori knowledge about the underlying structure of the data, the network used a processing pipeline of information theoretic analysis, defined in Figure 6.7, to infer the most suitable network structure for the 3D egomotion estimation.

Each step of the algorithm, depicted in Figure 6.7 a, provided a more refined description of the network structure, from all-to-all connectivity to statistically determined, fully informative links between sensory variables,as shown in Figure 6.7 b. Using relatively basic metrics like entropy, mutual information, and relatively simple entropy reduction mechanisms, the system builds a map of distances which reflects informational content each sensory variable contains and determines the association affinity to other variables, depicted explicitly in Figure 6.8 b. Capturing a measure of the amount of information that one sensory variable contains about the others, plausible and consistent sensory associations are extracted as we can see in Figure 6.9 b.

After a preliminary analysis of the data and the instantiation of the basic relational model for 3D egomotion estimation on a quadrotor, we explored the capability to learn sensory correlations for the quadrotor scenario. In our experiments, preprocessed data from the available sensors (i.e. gyroscope, accelerometer and a magnetic sensor) was fed to the model in order to extract relations between the inferred network of sensors for each of the three degrees of freedom (i.e. roll, pitch and yaw), Figure 6.9 a. Initially, all-to-all connections between sensors were considered, but the system, Figure 6.7, inferred only the connection configuration encoding plausible relations (i.e. contributions to same degree of freedom estimate), considered for subsequent fusion, synthetically described in Figure 6.9 b. The underlying structure estimating the three degrees of freedom is consistent to the generic model we introduced in Chapter 3, as associations in Figure 6.9 c show.

In order to extract the relations for roll and pitch estimation, the network combined contributions from accelerometer and gyro, whereas for yaw estimation the network fused magnetometer and gyroscope observations. Experimental results shown that the network was able to infer relatively precise roll and pitch estimates (relative to ground truth) for individual sensors (accelerometer RMSE Roll:$<$ 2%, accelerometer RMSE Pitch: $<$ 7%, gyroscope RMSE Roll:$<$ 3%, gyroscope RMSE Pitch:$<$ 3%) despite the noisy accelerometer and drifting gyroscope, Figure 6.10 b. Although yaw estimates followed the motion trend, the error was considerably large due to the intrinsic offset that the network didn't explicitly compensate (magnetometer RMSE Yaw:$\sim$ 15%). Furthermore, the learned relations resemble the nonlinear functions (i.e. arctangent) used in typical modelling approaches, although preserving the inherent irregularities in cross-sensory relations, visible in Figure 6.10 a.

Using generic neurally inspired processing elements (i.e. SOM, Hebbian learning) ensures that the network first learns the structure of the data, and then uses this representation to sharpen its correlational structure. Approaching perceptual learning from a biological perspective by using a flexible computational substrate, our framework for perceptual learning for multisensory fusion provides superior learning capabilities, given noisy sensory contributions, useful in leveraging adaptation capabilities of today's technical systems.

Final remarks

Since there is no single "Cartesian theatre" where all sensory input meets together for simultaneous processing, human multisensory processing works "by synchronizing sets of neural activity in separate brain regions" involving "time binding of images" occurring in different places but "within approximately the same window of time". This requires "maintaining focused activity at different sites for as long as necessary for meaningful combinations to be made and for reasoning and decision making to take place" [Damasio, 2012].

The central focus of the proposed research agenda was to understand multisensory information processing in neural systems, to develop novel algorithms inspired by brain functionality, and to transfer these into technical systems. Approaching the problem of designing adaptive and robust multisensory fusion systems inspired by neural systems drove a translational approach. The crux of this approach focused on understanding the core processing principles and computational substrate in order to design artificial self-constructing systems capable of autonomously associate sensory streams, learn underlying sensory correlations, and subsequently integrate available streams into more precise representations of the perceived quantities.

The pillars the proposed work is built on follow a reductionist intuition. Advocating the use of a distributed paradigm, the proposed work proposes a computational framework employing a network of relatively simple units with limited local processing and storage capabilities. Given different acquired sensory streams the extracted representation is not global but rather fragmented among units which mutually interact obeying simple dynamics towards consensus. This stable state ensures that all local representations are consistent and the global network-wide representation is coherent.

Heterogeneous sensory data carries informative content, hidden in its statistics, which needs to be extracted in order to improve the outcome of the integration process. Albeit the inherent difficulty in extracting meaningful information, there is a greater challenge in detecting which sensory cues react to the same events in the percept. Following this line, we propose a mechanism exploiting the underlying informational content in sensory time series for synthesizing an interacting network of units, given only underlying data regularities and associations. These associations might not be obvious in real-world scenarios, but robust extraction of underlying sensory associations drive the autonomous learning and representation of inter-sensory correlations from incoming sensory streams.

Finally, we showed that our system learns correlations to integrate sensory contributions for more precise representations and subsequently decoded real-world estimates over an adaptive sensory association layer. This robust design avoids painstaking parameterization routines by dynamically adapting to changes in the perceived quantities. Moreover, this approach provides an integrated perspective over multisensory fusion in real-world scenarios making localized intelligence a true computational framework for such dynamical scenarios.

As outlook, we envision more challenging scenarios to instantiate the framework. A first direction will be hardware implementations, such that simple operations executed locally in hardware, allow real-time instantiation for efficient belief propagation. Our approach provides a technique to represent complex relations between maps as computationally simple distributed systems. Such maps can represent e.g. sensory readings or desired motor outputs of robotic systems. In some of our previous instantiations we performed computation on sequential digital hardware, which often resulted in long settling times of the network. Here, we are envisioning massively parallel hardware systems to compute equilibriums of large scale networks quickly (such as FPGAs) or even instantaneously (analog hardware systems). This direction might provide new insight in high speed sensorimotor control problems in robotics.

A second focus will be on mobile multisensory fusion, contributing to the current effort to achieve contextual awareness in embedded sensor technology. The technology enabling context awareness in mobile devices includes wireless, ambient intelligence, user interfaces, powerful search engine capabilities, power management, software, mobile computing, and myriad perceiving and data-collecting sensors. Added to this list are such human factor enablers as emotional state, biophysiological condition, goals and social interaction that, when combined with the technological factors, provide the potential for a meaningful and individualized experience. Multisensory fusion can enable context awareness, which has huge potential within the mobile devices community. This direction supports the idea that, given its learning and adaptation capabilities, our multisensory fusion framework, context awareness, and mobile computing combined, can support a viable practical approach where a large number of different and distributed sensors are used to predict a context description and subsequently guide intelligent environment interaction.

Bibliography

[Abolmaesumi et al., 2004] P. Abolmaesumi, M. R. Sirouspour, An Interacting Multiple Model Probabilistic Data Association Filter for Cavity Boundary Extraction From Ultrasound Images, IEEE Transactions on Medical Imaging, pp. 772-784, 2004.

[Albert et al., 2002] R. Albert, A-L. Barabasi, Statistical mechanics of complex networks, Rev. Mod. Phys. 74, 2002.

[Althaus et al., 2013] N. Althaus, D. Mareschal, Modeling Cross-Modal Interactions in Early Word Learning, IEEE Transactions on Autonomous Mental Development, pp. 288-297, 2013.

[Appriou, 2014] A. Appriou, Uncertainty Theories and Multisensor Data Fusion, Wiley, 2014.

[Arkin et al., 1995] A. Arkin, J. Ross, Statistical construction of chemical reaction mechanisms from measured time-series, J. Phys. Chem. 99, 1995.

[Arleo et al., 2007] A. Arleo, L. Rondi-Reig, Multimodal sensory integration and concurrent navigation strategies for spatial cognition in real and artificial organisms, J. Integr. Neurosci. 3, pp. 327-366, 2007.

[Arnal et al., 2012] L. H. Arnal, A. L. Giraud, Cortical oscillations and sensory predictions, Trends in Cogn. Sci. 16, pp. 390-398, 2012.

[Asnath et al., 2014] Y. Asnath Victy Phamila, R. Amutha, Discrete Cosine Transform based fusion of multi-focus images for visual sensor networks, Signal Processing, Volume 95, 2014.

[Axenie et al., 2013] C. Axenie, J. Conradt, Cortically Inspired Sensor Fusion Network for Mobile Robot Heading Estimation, Proc. of Intl. Conf. on Artificial Neural Networks, pp. 240-247, 2013.

[Axenie et al., 2014] C. Axenie, J. Conradt, Cortically inspired sensor fusion network for mobile robot egomotion estimation, Robotics and Autonomous Systems, 2014.

[Bagher et al., 2011] M. Bagher A. Haghighat, A. Aghagolzadeh, H. Seyedarabi, Multifocus image fusion for visual sensor networks in DCT domain. Comput. Electr. Eng. 37, 5, 2011.

[Barbas, 2015] H. Barbas, General Cortical and Special Prefrontal Connections: Principles from Structure to Function, Ann. Rev. of Neuroscience 38, pp. 269-289, 2015.

[Barczyk et al., 2015] M. Barczyk, S. Bonnabel, J. Deschaud, F. Goulette, Invariant EKF Design for Scan Matching-aided Localization, IEEE Transactions on Control Systems Technology, 2015.

[Bar-Shalom et al., 1975] Y. Bar-Shalom, E. Tse, Tracking in a cluttered environment with probabilistic data association, Automatica, vol. 11, no. 5, pp. 451-460, 1975.

[Bashi et al., 2003] A. Bashi , V. Jilkov , X. Rong Li , H. Chen, Distributed Implementations of Particle Filters, Proc. of Sixth International Conference of Information Fusion, pp. 1164-1171, 2003.

[Battistelli et al., 2014] G. Battistelli, L. Chisci, C. Fantacci, N. Forti, A. Farina, A. Graziano, Distributed peer-to-peer multitarget tracking with association-based track fusion, Proc. of 17th International Conference on Information Fusion (FUSION), pp.1-7, 2014.

[Bauer et al., 2012] J. Bauer, C. Weber, S. Wermter, A SOM-based model for multisensory integration in the superior colliculus, Proc. of the International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2012.

[Becker et al., 1996] S. Becker, Mutual information maximization: Models of cortical self-organization, Network: Computation in Neural Systems, pp. 7-31, 1996.

[Bergner et al., 2014] F. Bergner, C. Axenie, Cortically Inspired Sensor Fusion for Quadrotor Attitude Control, TUM Master Thesis Report, 2014.

[Besada-Portas et al., 2002] E. Besada-Portas, J. A. Lopez-Orozco, J.M. de la Cruz, Unified fusion system based on Bayesian networks for autonomous mobile robots, Proc. of 5th Information Fusion Conference, 2002.

[Bloesch et al., 2014] M. Bloesch, S. Omari, P. Fankhauser, H. Sommer, C. Gehring, J. Hwangbo, M. A. Hoepflinger, M. Hutter, R. Siegwart, Fusion of optical flow and inertial measurements for robust egomotion estimation, 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), pp.3102-3107, 2014.

[Brekke et al., 2015] E. Brekke, M Chitre, A multi-hypothesis solution to data association for the two-frame SLAM problem, Int. J. of Robotics Research, pp. 43-63, 2015.

[Brent, 2013] R. P. Brent, An Algorithm with Guaranteed Convergence for Finding a Zero of a Function. Algorithms for Minimization without Derivatives. Dover Books on Mathematics, pp. 47-58, 2013.

[Bressler, 1995] S. L. Bressler, Large-scale cortical networks and cognition, Brain Research, 1995.

[Bressler, 2002] S. L. Bressler, Understanding Cognition Through Large-Scale Cortical Networks, J. of Current Directions in Psychological Science 11, pp. 58-61, 2002.

[Bressler et al., 2001] S. L. Bressler, J. A. S. Kelso, Cortical coordination dynamics and cognition, Trends in Cogn. Sci. 5, pp. 26-36, 2001.

[Bressler et al., 2006] S. L. Bressler, E. Tognoli, Operational principles of neurocognitive networks, J. of Psychophysiology 60, pp. 139-148, 2006.

[Buneo et al., 2006] C. A. Buneo, R. A. Anderson, The posterior parietal cortex: sensorimotor interface for the planning and online control of visually guided movements, Neuropsychologia 44, pp. 2594-2606, 2006.

[Butler et al., 2010] J. S. Butler, S. Smith, J. Campos, H. Bulthoff, Bayesian integration of visual and vestibular signals for heading, J. of Vision 10, 2010.

[Calvert et al., 2004] G. A. Calvert, T. Thesen, Multisensory integration: methodological approaches and emerging principles in the human brain, J. of Physiol. 98, pp. 191-205, 2004.

[Campos et al., 2012] J. Campos, J. Butler, H. Bulthoff, Multisensory integration in the estimation of walked distances, Exp. Brain Res. 218(4), pp. 551-565, 2012.

[Carreira-Perpinan et al., 2005] M. A. Carreira-Perpinan, R.J. Lister, G.J. Goodhill, A Computational Model for the Development of Multiple Maps in Primary Visual Cortex, J. Cerebral Cortex 15, pp. 1222-1233, 2005.

[Castaldo et al., 2014] F. Castaldo, F. A. N. Palmieri, Image fusion for object tracking using Factor Graphs, Proc. of IEEE Aerospace Conference, pp. 1-8, 2014.

[Castanedo, 2013] F. Castanedo, A Review of Data Fusion Techniques, The Scientific World Journal, 2013.

[Chen et al., 2005] L. Chen, M. Cetin, A. S. Willsky, Distributed data association for multi-target tracking in sensor networks, Proc. of the 7th International Conference on Information Fusion, pp. 9-16, 2005.

[Chen et al., 2006] L. Chen, M. J. Wainwright, M. Cetin, A. S. Willsky, Data association based on optimization in graphical models with application to sensor networks, J. of Mathematical and Computer Modelling, vol. 43, pp. 1114-1113, 2006.

[Chen et al., 2007] Z. Chen, S. Haykin, J. J. Eggermont, S. Becker, Correlative Learning: A Basis for Brain and Adaptive Systems, Wiley, 2007.

[Chen et al., 2011] A. Chen, G. DeAngelis, D.E. Angelaki, Representation of Vestibular and Visual Cues to Self-Motion in Ventral Intraparietal Cortex, J. of Neuroscience 31, pp. 12036-12052, 2011.

[Chen et al., 2012] J. Chen, K. H. Low, C. K. Tan, A. Oran, P. Jaillet, J. M. Dolan, G. S. Sukhatme, Decentralized Data Fusion and Active Sensing with Mobile Sensors for Modeling and Predicting Spatiotemporal Traffic Phenomena, 28th Conference on Uncertainty in Artificial Intelligence, 2012.

[Chin et al., 2014] Y.J. Chin, T.S. Ong, A.B.J. Teoh, K.O.M. Goh, Integrated biometrics template protection technique based on fingerprint and palmprint feature-level fusion, Information Fusion, Volume 18, 2014.

[Chong et al., 2001] C. Chong, S. Mori, Convex combination and covariance intersection algorithms in distributed fusion, Proc. of the 4th International Conference on Information Fusion, 2001.

[Chong et al., 2014] C. Chong, S. Mori, F. Govaers, W. Koch, Comparison of tracklet fusion and distributed Kalman filter for track fusion, Proc. of 17th International Conference on Information Fusion (FUSION), pp. 1-8, 2014.

[Christie et al., 2010] S. Christie, D. Gentner, Where Hypotheses Come From: Learning New Relations by Structural Alignment, J. of Cognition and Development 11, pp. 356-373, 2010.

[Cifuentes et al., 2014] C. A. Cifuentes, C. Rodriguez, A. Frizera, T. Bastos,, Sensor fusion to control a robotic walker based on upper-limbs reaction forces and gait kinematics, Biomedical Robotics and Biomechatronics, pp.1098-1103, 2014.

[Cimponeriu et al., 2000] A. Cimponeriu, G. Goodhill, Dynamics of Cortical Map Development in the Elastic Net Model, Neurocomputing 32-33, pp. 83-90, 2000.

[Cook et al., 2004] M. Cook, J. Bruck, Networks of Relations for Representation, Learning, and Generalization, Proc. Fourth Intl. Conference on Intelligent Systems Design and Applications, 2004.

[Cook, Jug et al., 2010] M. Cook, F. Jug, C. Krautz, A. Steger, Unsupervised Learning of Relations. Proc. of Artificial Neural Networks Conference, pp. 162-173, 2010.

[Cook, Gugelmann et al., 2010] M. Cook, L. Gugelmann, F. Jug, C. Krautz, A. Steger, Interacting maps for fast visual interpretation, Proc. of Intl. Joint Conf. on Neural Networks,pp. 770-776, 2010.

[Cook et al., 2011] M. Cook, L. Gugelmann, F. Jug, C. Krautz, A. Steger, Interacting maps for fast visual interpretation, Neural Networks (IJCNN), The 2011 International Joint Conference on , pp.770 - 776, 2011.

[Coraluppi et al., 2011] S. Coraluppi, C. Carthel, Aggregate surveillance: a cardinality tracking approach, Proc. of the 14th International Conference on Information Fusion, 2011.

[da Costa et al., 2011] N. M. da Costa, K. A. C. Martin, How Thalamus Connects to Spiny Stellate Cells in the Cat's Visual Cortex, The Journal of Neuroscience 31 (8), pp. 2925-2937, 2011.

[Damasio, 2012] A. R. Damasio, H. Damasio, Neurobiology of Decision-Making, Springer, 2012.

[Dani et al., 2014] A. Dani, M. McCourt, J. W. Curtis, A. Mehta, Information fusion in human-robot collaboration using neural network representation, Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on, 2014.

[Dasarathy, 1997] B. V. Dasarathy, Sensor fusion potential exploitation-innovative architectures and illustrative applications, Proceedings of the IEEE, vol. 85, no. 1, pp. 24-38, 1997.

[DeAngelis et al., 2012] G. C. DeAngelis, D. E. Angelaki, Visual-Vestibular Integration for Self-Motion Perception, in M. M. Murray, M.T. Wallace (Eds.), The Neural Bases of Multisensory Processes, CRC Press, Boca Raton, pp. 629-652, 2012.

[Deneve et al., 2007] S. Deneve, J. Duhamel, Alexandre Pouget, Optimal Sensorimotor Integration in Recurrent Cortical Networks: A Neural Implementation of Kalman Filters, Journal of Neuroscience 27, pp. 5744-5756, 2007.

[Denoux et al., 2014] T. Denoux; N. El Zoghby, V. Cherfaoui, A. Jouglet, Optimal Object Association in the Dempster-Shafer Framework, Cybernetics, IEEE Transactions on, vol.44, no.12, pp.2521-2531, 2014.

[Doya et al., 2007] K. Doya et al., Bayesian brain: Probabilistic approaches to neural coding, MIT Press, 2007.

[Doyle et al., 2011] J C. Doyle, M. Csete, Architecture, constraints, and behavior, PNAS 108, pp. 15624-15630, 2011.

[Durrant-Whyte, 1988] H. F. Durrant-Whyte, Sensor models and multisensor integration, International Journal of Robotics Research, vol. 7, no. 6, pp. 97-113, 1988.

[Durrant-Whyte et al., 2008] H. Durrant-Whyte, T. C. Henderson, Multisensor Data Fusion, in B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer, Berlin, pp. 585-608, 2008.

[Edelman et al., 2013] G. M. Edelman, J A. Gally, Reentry: a key mechanism for integration of brain function, Front. Integr. Neurosci. 7:63, pp. 1-6, 2013.

[Engel et al., 2012] J. Engel, J. Sturm, D. Cremers, Camera-based navigation of a low-cost quadrocopter, Proc. of IEEE/RSJ International Conference Intelligent Robots and Systems (IROS), pp. 2815-2821, 2012.

[Engel et al., 2013] A. K. Engel, C. Gerloff, C. C. Hilgetag, G. Nolte, Intrinsic Coupling Modes: Multiscale Interactions in Ongoing Brain Activity, Neuron 80, pp. 867-886, 2013.

[Erdem et al., 2015] A. T. Erdem, A.O. Ercan, Fusing Inertial Sensor Data in an Extended Kalman Filter for 3D Camera Tracking, IEEE Transactions on Image Processing 24, pp. 538-548, 2015.

[Erdos et al., 1960] P. Erdos, A. Renyi, On the evolution of random graphs, Publ. Math. Inst. Hungary. Acad. Sci. 5, pp. 17-61, 1960.

[Ernst, 2012] M. O. Ernst, Optimal Multisensory Integration: Assumptions and limits from B. E. Stein (Ed.), The New Handbook of Multisensory Processing, MIT Press, pp. 527-543, 2012.

[Ernst et al., 2002] M. O. Ernst, M. S. Banks, Humans integrate visual and haptic information in a statistically optimal fashion, Nature 415, pp. 429-433, 2002.

[Ernst et al., 2004] M. O. Ernst , H. H. Bulthoff, Merging the senses into a robust percept, J. Trends in Cognitive Science 8, pp. 162-169, 2004.

[Ernst et al., 2012] M. O. Ernst, Optimal Multisensory Integration: Assumptions and Limits, from The New Handbook of Multisensory Processing, MIT Press, pp. 527-543, 2012.

[Erturk, 2002] S. Erturk, Real-Time Digital Image Stabilization Using Kalman Filters, Real-Time Imaging, Volume 8, Issue 4, 2002.

[Escamilla-Ambrosio et al., 2003] P. J. Escamilla-Ambrosio, N. Mort, Hybrid Kalman filter-fuzzy logic adaptive multisensor data fusion architectures, in: Proc. of the IEEE Conf. on Decision and Control, pp. 5215-5220, 2003.

[Fallon et al., 2012] M.F. Fallon, H. Johannsson, J. Brookshire, S. Teller, J. J. Leonard, Sensor fusion for flexible human-portable building-scale mapping, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4405-4412, 2012.

[Feldman, 2013] J. Feldman, Bayesian models of perceptual organization, in J. Wagemans (Ed.), Handbook of perceptual organization, Oxford University Press, Oxford, 2013.

[Feng et al., 2014] S. Feng, C. Wu, Y. Zhang, Z. Jia, Grid-Based Improved Maximum Likelihood Estimation for Dynamic Localization of Mobile Robots, International Journal of Distributed Sensor Networks, 2014.

[Ferreira et al., 2011] J. F. Ferreira, J. Lobo, J. Dias, Bayesian Real-Time Perception Algorithms on GPU - Real-Time Implementation of Bayesian Models for Multimodal Perception Using CUDA, J. of Real-Time Image Processing 6, pp. 171-186, 2011.

[Ferreira et al., 2012] J. F. Ferreira, M. Castelo-Branco, J. Dias, A hierarchical Bayesian framework for multimodal active perception, Adaptive Behavior 20, pp. 172-190, 2012.

[Ferreira et al., 2013] J. F. Ferreira, J. Lobo, P. Bessiere, M. Castelo-Branco, J. Dias, A Bayesian framework for active artificial perception, IEEE Transactions on Cybernetics 43, pp. 699-711, 2013.

[Ferreira et al., 2014] J. F. Ferreira, J. Dias, Probabilistic Approaches for Robotic Perception, Springer Tracts in Advanced Robotics (STAR) 91, 2014.

[Fetsch et al., 2009] C. Fetsch, A. Turner, G. DeAngelis, D. Angelaki, Dynamic reweighting of visual and vestibular cues during self-motion perception, J. of Neurosci., pp. 15601-15612, 2009.

[Furber et al., 2013] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, A. D. Brown, Overview of the SpiNNaker System Architecture, IEEE Transactions on Computers, vol.62, no.12, pp. 2454-2467, 2013.

[Ganguli et al., 2014] D. Ganguli, E. P. Simoncelli, Efficient Sensory Encoding and Bayesian Inference with Heterogeneous Neural Populations, Neural Computation 26, pp. 2103-2134, 2014.

[Gibson et al., 2003] E. J. Gibson, A. D. Pick, An Ecological Approach to Perceptual Learning and Development, Oxford University Press, 2003.

[Gil et al., 2006] A. Gil, O. Reinoso, O. M. Mozos, C. Stachniss, W. Burgard, Improving Data Association in Vision-based SLAM, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2076-2081, 2006.

[Gil et al., 2010] A. Gil, O. Reinoso , M. Ballesta , M. Juli, L. Paya, Estimation of Visual Maps with a Robot Network Equipped with Vision Sensors, Sensors 10, pp. 5209-5232, 2010.

[Gorji et al., 2007] A. Gorji, M. B. Menhaj, S. Shiry, Multiple Target Tracking for Mobile Robots Using the JPDAF Algorithm, Proc. of IEEE Int. Conf. on Tools with Artificial Intelligence, pp. 137-145., 2007.

[Graziano et al., 2004] M. S. A. Graziano, C. G. Gross, C. S. R. Taylor, T. Moore, A system of multimodal areas in the primate brain, in C. Spence, J. Driver (Eds.), Crossmodal space and crossmodal attention, Oxford University Press, New York, pp. 51-68, 2004.

[Grossberg, 2007] S. Grossberg, Towards a unified theory of neocortex: laminar cortical circuits for vision and cognition, Prog. Brain Res. 165, pp. 79-104, 2007.

[Guenthner et al., 2006] W. Guenthner et al., Biologically Inspired Multi-Sensor Fusion for Adaptive Camera Stabilization in Driver-Assistance Systems, Advanced Microsystems for Automotive Applications, Springer, Berlin, 2006.

[Gu et al., 2008] Y. Gu, D. E. Angelaki, G. C. DeAngelis, Neural Correlates of Multisensory Cue Integration in Macaque MSTd, Nature Neuroscience 11, pp. 1201-1210, 2008.

[Halford et al., 1998] G. S. Halford, W. H. Wilson, S. Phillips, Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology, Behav Brain Sci 21, pp. 803-831, 1998.

[Hall et al., 1997] D. L. Hall and J. Llinas, An introduction to multisensor data fusion, Proceedings of the IEEE, vol. 85, no. 1, pp. 6-23, 1997.

[Hall et al., 2004] D. L. Hall, S. A. H. McMullen, Mathematical Techniques in Multisensor Data Fusion, Artech House Inc, 2004.

[Han-Pang et al., 2014] C. Han-Pang, X. S. Zhou, L. Carlone, F. Dellaert, S. Samarasekera, R. Kumar, Constrained optimal selection for multi-sensor robot navigation using plug-and-play factor graphs, Proc. of IEEE International Conference on Robotics and Automation (ICRA), pp. 663-670, 2014.

[Hatano et al., 2005] Y. Hatano, M. Mesbahi, Agreement over random networks, IEEE Trans. on Automatic Control 50, pp. 1867-1872, 2005.

[Hawkins et al., 2006] J. Hawkins, D. George, Hierarchical Temporal Memory - Concepts, Theory, and Terminology, Numenta Inc., 2006.

[Heed et al., 2012] T. Heed, B. Roeder, The Body in a Multisensory World, in M. M. Murray, M.T. Wallace (Eds.), The Neural Bases of Multisensory Processes, CRC Press, Boca Raton, pp. 557-582, 2012.

[Hinton, 1976] G. E. Hinton, Using relaxation to find a puppet, Proc. of A.I.S.B. Summer Conference, University of Edinburgh, 1976.

[Hlinka et al., 2013] O. Hlinka, F. Hlawatsch, P. M. Djuric, Distributed particle filtering in agent networks: A survey, classification, and comparison, IEEE Signal Processing Magazine, pp. 61-81, 2013.

[Howard et al., 2002] A. Howard, M. J. Mataric, G. S. Sukhatme, Localization for Mobile Robot Teams Using Maximum Likelihood Estimation, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 434-459, 2002.

[Hsieh, 2000] W. W. Hsieh, Nonlinear canonical correlation analysis by neural networks, Neural Networks, pp. 1095-1105, 2000.

[Huerta et al., 2014] I. Huerta, G. Ferrer, F. Herrero, A. Prati, A. Sanfeliu, Multimodal feedback fusion of laser, image and temporal information. In Proceedings of the International Conference on Distributed Smart Cameras, 2014.

[Hyon et al., 2012] L. Hyon, J. Park, D. Lee, H. J. Kim, Build your own quadrotor, IEEE Robotics and Automation Magazine, pp. 33-45, 2012.

[Indelman et al., 2013] V. Indelman, S. Williams, M. Kaess, F. Dellaert, Information fusion in navigation systems via factor graph based incremental smoothing, Robotics and Autonomous Systems, pp. 721-738, 2013.

[Indelman et al., 2014] V. Indelman, E. Nelson, N. Michael, F. Dellaert, Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization, Proc. of IEEE International Conference on Robotics and Automation (ICRA), pp. 593-600, 2014.

[Jianqin et al., 2014] Y. Jianqin, T. Guohui, L. Guodong, Object localization and tracking based on multiple sensor fusion in intelligent home, Proc. of Control and Decision Conference, pp.5266-5270, 2014.

[Joo et al., 2007] S. W. Joo, R. Chellappa, A multiple-hypothesis approach for multiobject visual tracking, IEEE Transactions on Image Processing, vol. 16, no. 11, pp. 2849-2854, 2007.

[Julier et al., 1997] S. J. Julier, J. K. Uhlmann, A non-divergent estimation algorithm in the presence of unknown correlations, Proceedings of the American Control Conference, vol.4, pp. 2369-2373, 1997.

[Kaiser et al., 2004] M. Kaiser, C.C. Hilgetag, Spatial growth of real-world networks, Physical Review E 69, 2004.

[Kayser et al., 2015] C. Kayser, L. Shams, Multisensory Causal Inference in the Brain, PLoS Biol 13(2), 2015.

[Kelly et al., 2014] J. Kelly, G. S. Sukhatme, A General Framework for Temporal Calibration of Multiple Proprioceptive and Exteroceptive Sensors, J. of Experimental Robotics, vol. 79, 2014.

[Khalengi et al., 2013] B. Khaleghi, A. Khamis, F. O. Karray, S. N. Razavi, Multisensor data fusion: A review of the state-of-the-art, Information Fusion, vol. 14, pp. 28-44, 2013.

[Kirubarajan et al., 2004] T. Kirubarajan, Y. Bar-Shalom, Probabilistic Data Association Techniques for Target Tracking in Clutter, Proc. of the IEEE, pp. 536-557, 2004.

[Klatzky, 1998] R. Klatzky, Allocentric and Egocentric Spatial Representations: Definitions, Distinctions and Interconnections, Spatial Cognition, LNCS 1404, pp. 1-17, 1998.

[Knudsen et al., 1987] E. I. Knudsen, S. du Lac, S. D. Esterly, Computational maps in the brain, Annual Reviews Neuroscience, pp. 41-65, 1987.

[Kozma et al., 2008] B Kozma, A Barrat, Consensus formation on adaptive networks, Physical Review E, 2008.

[Kubelka et al., 2014] V. Kubelka, L. Oswald, F. Pomerleau, F. Colas, T. Svoboda, M. Reinstein, Robust Data Fusion of Multimodal Sensory Information for Mobile Robots, J. Field Robotics, 2014.

[Kumar et al., 2006] M. Kumar, D. P. Garg, R. A. Zachery, A generalized approach for inconsistency detection in data fusion from multiple sensors, American Control Conference, 2006.

[Lackner et al., 2004] J. R. Lackner, P. Dizio, Multisensory Influences on Orientation and Movement Control, in G. Calvert, C. Spence, B. E. Stein (Eds.), The Handbook of Multisensory Processes, MIT Press, Cambridge, Massachusetts, pp. 409-424, 2004.

[Lai et al., 1999] P. L. Lai, C. Fyfe, A neural network implementation of canonical correlation analysis, Neural Networks, pp. 1391-1397, 1999.

[Lai et al., 2000] P. L. Lai, C. Fyfe, Kernel and nonlinear canonical correlation analysis. Int. J. of Neural Systems, pp. 365-377, 2000.

[Landy et al., 2011] M. S. Landy, M. S. Banks, D. C. Knill, Ideal-Observer Models of Cue Integration, in J. Trommershauser, K. Koerding, M. S. Landy (Eds.), Sensory Cue Integration, Oxford University Press, pp.5-29, 2001.

[Landy et al., 2012] M. Landy, M. S. Banks, D. C. Knill, Ideal-Observer Model of Cue Integration, from Sensory Cue Integration, Oxford Press, pp. 5-29, 2012.

[Latham et al., 2003] P. E. Latham, S. Deneve, A. Pouget, Optimal computation with attractor networks, J. of Physiology, Elsevier, 2003.

[Law et al., 2008] C. Law, J. I. Gold, Neural correlates of perceptual learning in a sensory-motor, but not a sensory, cortical area, Nature Neuroscience 11, pp. 505-513, 2008.

[Lawson et al., 2015] L.S. Lawson, L. Pack Kaelbling, T. Lozano-Perez. Data Association for Semantic World Modeling from Partial Views, Int. J. of Robotics Research, In Press, 2015.

[Lee et al., 2012] J. K. Lee, E. J. Park, S. N. Robinovich, Estimation of attitude and external acceleration using inertial sensor measurement during various dynamic conditions, IEEE Transactions on Instrumentation and Measurement 61, pp. 2262-2273, 2012.

[Leivas et al., 2010] G. Leivas, S. Botelho, P. Drews, M. Figueiredo, C. Haeffele, Sensor fusion based on multi-self-organizing maps for SLAM, Proc. of IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 139-143, 2010.

[Liu et al., 2014] C. Liu, S. Dong, B. Lu, A. Hauptmann, C. Li, Study on Adaptive and Fuzzy Weighted Image Fusion Based on Wavelet Transform in Trinocular Vision of Picking Robot, J. of Inf. and Comp. Sc. 11 (6), 2014.

[Lungarella et al., 2005] M. Lungarella, O. Sporns, Information Self-Structuring: Key Principle for Learning and Development, Proc. of Intl. Conf. on Development and Learning, pp. 25-30, 2005.

[Luo et al., 2002] R. C. Luo, C.-C. Yih, K. L. Su, Multisensor fusion and integration: approaches, applications, and future research directions, IEEE Sensors Journal, vol. 2, no. 2, pp. 107-119, 2002.

[Luo et al., 2006] X. Luo, M. Dong, Y. Huang, On distributed fault-tolerant detection in wireless sensor networks, IEEE Transactions on Computers, vol.55, no.1, pp.58-70, 2006.

[Madar et al., 2010] A. Madar, A. Greenfield, E. Vanden-Eijnden, R. Bonneau, DREAM3: network inference using dynamic context likelihood of relatedness and the inferelator, PLoS One 5(3), 2010.

[Makarenko et al., 2009] A. Makarenko, A. Brooks, T. Kaupp, H. Durrant-Whyte, F. Dellaert, Decentralised data fusion: A graphical model approach, Proc. of 12th International Conference on Information Fusion, pp. 545-554, 2009.

[Mandal et al., 2013] A. Mandal, A. Cichocki, Non-linear canonical correlation analysis using alpha-beta divergence, Entropy, vol. 15, pp. 2788-2804, 2013.

[Mast et al., 2007] F. W. Mast, L. Jancke (Eds.), Spatial Processing in Navigation, Imagery and Perception, Springer, Berlin, 2007.

[Mayor et al., 2010] J. Mayor, K. Plunkett, A neurocomputational account of taxonomic responding and fast mapping in early word learning, Psychological Revues, pp. 1-31, 2010;

[Meredith et al., 2012] M. A. Meredith, K. J. Cios, A. R. MacQuistion, HK. Lim, L. P. Keniston, H. R. Clemo, Neuroanatomical Identification of Multisensory Convergence on Higher-level Cortical Neurons, from The New Handbook of Multisensory Processing, MIT Press, 2012.

[Mergner et al., 1990] T. Mergner, W. Becker, Perception of horizontal self-rotation: Multisensory and cognitive aspects, in R. Warren, A. H. Wertheim (Eds.), Perception & Control of Self-motion, Hillsdale, pp. 219-260, 1990.

[Meyer et al., 2014] P. E. Meyer, K. Kontos, F. Lafitte, G. Bontempi, Information-theoretic inference of large transcriptional regulatory networks, EURASIP J. Bioinform. Syst. Biol. (1), 2007.

[Michler et al., 2009] F. Michler, R. Eckhorn, T. Wachtler, Using Spatiotemporal Correlations to Learn Topographic Maps for Invariant Object Recognition, J. of Neurophysiology 102, pp. 954-964, 2009.

[Mitchel, 2010] R. W. Mitchel, Understanding the body: spatial perception and spatial cognition, in F. L. Dolins, R. W. Mitchell (Eds.), Spatial Cognition, Spatial Perception: Mapping the Self and Space, Cambridge University Press, Cambridge, pp. 341-364, 2010.

[Mitchison, 1995] G. Mitchison, A type of duality between self-organizing maps and minimal wiring, Neural Computation, 1995.

[Montemerlo et al., 2002] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, Proc. of the AAAI National Conference on Artificial Intelligence, pp. 593-598, 2002.

[Morefield, 1977] C. L. Morefield, Application of 0-1 integer programming to multitarget tracking problems, IEEE Transactions on Automatic Control, vol. 22, no. 3, pp. 302-312, 1977.

[Nakamura et al., 2005] E. F. Nakamura, C. M. Figueiredo, A. A. Loureiro, Information fusion for data dissemination in self-organizing wireless sensor networks, In Proc. of the 4th Intl. Conf. on Networking, 2005.

[Neagu, 2005] N. Neagu, Constraint Satisfaction Techniques for Agent-Based Reasoning, Whitestein Series in Software Agent Technologies and Autonomic Computing, Birkhäuser Basel, Springer, 2005.

[Novak et al., 2014] D. Novak, R. Riener, A survey of sensor fusion methods in wearable robotics, Robotics and Autonomous Systems, 2014.

[Ohshiro et al., 2011] T. Ohshiro, D.E. Angleaki, G.C. DeAngelis, A Normalization Model of Multisensory Integration, Nature Neuroscience 14, pp. 775-782, 2011.

[Olfati-Saber et al., 2011] R. Olfati-Saber, P. Jalalkamali, Collaborative target tracking using distributed Kalman filtering on mobile sensor networks, Proc. of American Control Conference (ACC), pp. 1100-1105, 2011.

[Oxenham, 2008] M. Oxenham, The effect of finite set representations on the evaluation of Dempster's rule of combination, in: Proc. of the Intl. Conf. on Information Fusion, pp. 1-8, 2008.

[Parise et al., 2012] C. V. Parise, C. Spence, M. O. Ernst, When Correlation Implies Causation in Multisensory Integration, Current Biology 22, pp. 46-49, 2012.

[Park et al., 2007] S. Park, K. Shin, A. Abraham, S. Han, Optimized Self Organized Sensor Networks, Sensors, pp. 730-742, 2007.

[Passingham et al., 2002] R. E. Passingham, E. K. Stephan, Rolf Koetter, The anatomical basis of functional localization in the cortex, Nature Reviews Neuroscience 3, pp. 606-616, 2002.

[Passino, 2005] K. M. Passino, Biomimicry for Optimization, Control, and Automation, Springer, 2005.

[Pennisi et al., 2014] A. Pennisi, F. Previtali, F. Ficarola, D.D. Bloisi, L. Iocchi, A. Vitaletti, Distributed Sensor Network for Multi-robot Surveillance, Procedia Computer Science, Volume 32, 2014.

[Pezeshki et al., 2003] A. Pezeshki, M. R. Azimi-Sadjadi, L. L. Scharf, A network for recursive extraction of canonical coordinates, Neural Networks, pp. 801-808, 2003.

[Phillips et al., 1995] S. Phillips, G. S. Halford, W. H. Wilson, The Processing of Associations versus the Processing of Relations and Symbols: A Systematic Comparison, Proc. of Seventh Annual Conf. Cog. Sci. Society, 1995.

[Polastre et al., 2004] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, SenSys'04, 2004.

[Pouget et al., 2004] A. Pouget, S. Deneve, J-R. Duhamel, A computational neural theory of multisensory spatial representations, in C. Spence, J. Driver (Eds.), Crossmodal space and crossmodal attention, Oxford University Press, New York, pp. 123-140, 2004.

[Pouget et al., 2013] A. Pouget, J. M. Beck, W. J. Ma, P. E. Latham, Probabilistic brains: knowns and unknowns, Nature Neuroscience 9, pp. 1170-1178, 2013.

[Quiton et al., 2011] J. Quiton, B. Girau, M. Lefort, Competition in high dimensional spaces using a sparse approximation of neural fields, From Brains to Systems Advances in Experimental Medicine and Biology Vol. 718, pp. 123-137, 2011.

[Rahnavard et al., 2013] G. Rahnavard, Y. S. Moon, L. McIver, E. F. Franzosa, L. Waldron, C Huttenhower, HAllA: Hierarchical All-against-All for Blocked Variable Selection and Association Discovery Among Large-Scale Heterogeneous Datasets, 2013.

[Rajesh et al., 2014] M. Rajesh, R. Joseph, T. S. B. Sudarshan, Fully distributed and decentralized map building for multi-robot exploration, Embedded Systems (ICES), 2014 International Conference on , pp.220-224, 2014.

[Rao et al., 1999] R. P. N. Rao, D. H. Ballard, Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects, Nature Neuroscience 2, pp. 79-87, 1999.

[Reggia et al., 2001] J. A. Reggia, Y. Shkuro, N. Shevtsova, Computational Investigation of Hemispheric Specialization and Interactions, Emergent Neural Computational Architectures Based on Neuroscience LNCS 2036, pp. 68-82, 2001.

[Reid, 1979] D. B. Reid, An algorithm for tracking multiple targets, IEEE Transactions on Automatic Control, vol. 24, no. 6, pp. 843-854, 1979.

[Reilly, 2001] R. G. Reilly, Collaborative Cell Assemblies: Building Blocks of Cortical Computation, Emergent Neural Computational Architectures Based on Neuroscience LNCS 2036, pp. 161-173, 2001.

[Reinhardt et al., 2012] M. Reinhardt, B. Noack, U. D. Hanebeck, The Hypothesizing Distributed Kalman Filter, Proc. of IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 305-312, 2012.

[Requena-Witzig et al., 2015] S. Requena-Witzig, C. Axenie, Cortically Inspired Quadrotor 3D Motion Estimation, TUM Bachelor Thesis Report, 2015.

[Ringach, 2007] D. L. Ringach, On the Origin of the Functional Architecture of the Cortex, PLoS ONE 2, 2007.

[Rosencrantz et al., 2003] M. Rosencrantz, G. Gordon, S. Thrun, Decentralized Sensor Fusion With Distributed Particle Filters, Proc. of the Nineteenth Conference on Uncertainty in Artificial Intelligence, 2003.

[Rowland, 2012] B. Rowland, Computational Models of Multisensory Integration, in B. Stein (Ed.), The New Handbook of Multisensory Processing, MIT Press, Cambridge, pp. 511-514, 2012.

[Rubinov et al., 2009] M. Rubinov, O. Sporns, C. van Leeuwen, M. Breakspear, Symbiotic relationship between brain structure and dynamics, BMC Neuroscience, 2009.

[Samoilov et al., 1997] M. Samoilov, Reconstruction and functional analysis of general chemical reactions and reaction networks, Ph.D. thesis, Stanford University, 1997.

[Samoilov et al., 2001] M. Samoilov, A. Arkin, J. Ross, On the deduction of chemical reaction pathways from measurements of time series of concentrations, Chaos 11, pp. 108-114, 2001.

[Santos et al., 2015] J. M. Santos, M. S. Couceiro, D. Portugal, R. P. Rocha, A Sensor Fusion Layer to Cope with Reduced Visibility in SLAM, J. of Intelligent and Robotic Systems, 2015.

[Saul et al., 2003] L. K. Saul, S. T. Roweis, Think globally, fit locally: unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research, pp. 119-155, 2003.

[Scherba et al., 2005] D. J. Scherba, P. Bajcsy, Depth map calibration by stereo and wireless sensor network fusion, Information Fusion, 8th International Conference on, vol.2, pp.25-28, 2005.

[Schultz et al., 2003] D. Schultz, W. Burgard, D. Fox, A. B. Cremers, People Tracking with Mobile Robots Using Sample-based Joint Probabilistic Data Association Filters, Int. J. of Robotics Research, pp. 99-116, 2003.

[Sequeira et al., 2009] J. Sequeira, A. Tsourdos, S. Lazarus. Robust covariance estimation in sensor data fusion, Proc. of IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR), 2009.

[Seung et al., 2000] H. S. Seung, D. D. Lee, The Manifold Ways of Perception, Science 22, pp. 2268-2269, 2000.

[Shastri et al., 1993] L. Shastri, V. Ajjanagadde, From Simple Associations to Systematic Reasoning: a Connectionist Representation of Rules, Variables and Dynamic Bindings Using Temporal Synchrony, Behavioral and Brain Sciences 16, pp. 417-494, 1993.

[Sheets-Johnstone, 2010] M. Sheets-Johnstone, Movement: the generative source of spatial perception and cognition, in F. L. Dolins, R. W. Mitchell (Eds.), Spatial Cognition, Spatial Perception: Mapping the Self and Space, Cambridge University Press, Cambridge, pp. 323-340, 2010.

[Sherman, 2012] S. Muray Sherman, Thalamocortical Interactions, Current Opinion in Neurobiology 22, pp. 575-579, 2012.

[Shindler et al., 2011] M. Shindler, A. Wong, A. Meyerson, Fast and accurate K-means for large datasets, Proc. of the 25th Annual Conference on Neural Information Processing Systems (NIPS), pp. 2375-2383, 2011.

[Siagian et al., 2014] C. Siagian, C. K. Chang, L. Itti, Autonomous Mobile Robot Localization and Navigation Using a Hierarchical Map Representation Primarily Guided by Vision, J. Field Robotics, 2014.

[Simlinger et al., 2015] B. Simlinger, S. Trendel, C. Axenie, Sensor fusion on SpiNNaker (Kalman filters vs. Neurally inspired models), TUM Interdisciplinary Project Report, 2015.

[Singh et al., 2006] A. Singh, R. Novak, P. Rmanathan, Active learning for adaptive mobile sensing networks, In Proc. of the 5th Intl. Conf. on Information Processing in Sensor Networks (IPSN'06), 2006.

[Smets, 2007] P. Smets, Analyzing the combination of conflicting belief functions, Information Fusion, Volume 8, Issue 4, 2007.

[Soumalya et al., 2014] S. Soumalya, S. Soumik, V. Nurali, R. Asok, Y. Murat, Sensor fusion for fault detection and classification in distributed physical processes, Frontiers in Robotics and AI, vol. 1, 2014.

[Spence, 2012] C. Spence, Multisensory perception, cognition and behaviour: Evaluating the factors modulating multisensory integration, The New Handbook of Multisensory Processing, B. Stein (Ed.), MIT Press, pp. 241-264, 2012.

[Spence et al., 2012] C. Spence, Y. Chen, Intramodal and cross-modal perceptual grouping, The New Handbook of Multisensory Processing, B. Stein (Ed.), MIT Press, pp. 265-282, 2012.

[Sporns, 2011] O. Sporns, Networks of the Brain, MIT Press, Cambridge, MA, 2011.

[Stein et al., 2004] B. E. Stein, T. R. Stanford, M. T. Wallace, J. W. Vaughan, W. Jiang, Crossmodal Spatial Interactions in Subcortical and Cortical Circuits, in C. Spence, J. Driver (Eds.), Crossmodal space and crossmodal attention, Oxford University Press, New York, pp. 25-50, 2004.

[Steuer et al., 1995] R. Steuer, J. Kurths, C. Daub, J. Weise, J. Selbig, The mutual information: detecting and evaluating dependencies between variables, Bioinformatics 18, pp. 231-240, 2002.

[Stone et al., 1995] J. Stone, A Bray, A Learning Rule for Extracting Spatio-Temporal Invariances, Network: Computation in Neural Systems, 1995.

[Swindale, 2005] N. V. Swindale, How different Feature Spaces may be Represented in Cortical Maps, Network: Computation in Neural Systems 5, pp. 217-242, 2005.

[Taylor et al., 2010] G W. Taylor, R. Fergus, Y. LeCun, C. Bregler, Convolutional Learning of Spatio-temporal Features, Proc. of Computer Vision Conference ECCV, pp. 140-153, 2010.

[Tchango et al., 2014] A.F. Tchango, V. Thomas, O. Buffet,A. Dutech, F. Flacher, Tracking multiple interacting targets using a joint probabilistic Data Association filter, Proc. of 17th International Conference on Information Fusion (FUSION), pp. 1-8, 2014.

[Thomas et al., 2004] P. J. Thomas, J. D. Cowan, Symmetry Induced Coupling of Cortical Feature Maps, Physical Reviews Letters 92, 2004.

[Thomas et al., 2007] U. Thomas, S. Molkenstruck, R. Iser, F. M. Wahl, Multi Sensor Fusion in Robot Assembly Using Particle Filters, Proc. of IEEE International Conference on Robotics and Automation, pp. 3837-3843, 2007.

[Thrun et al., 2005] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, Cambridge, MA, 2005.

[Tonia et al., 2001] I. Tonia, N. Ramnania, O Josephsa, J Ashburnera, R. E. Passingham, Learning Arbitrary Visuomotor Associations: Temporal Dynamic of Brain Activity, NeuroImage, pp. 1048-1057, 2001.

[Tsokas et al., 2012] N. A. Tsokas, K. J. Kyriakopoulos, Multi-robot multiple hypothesis tracking for pedestrian tracking, Autonomous Robots, Volume 32, pp 63-79, 2012.

[Uhlmann, 2003] J. Uhlmann, Covariance consistency methods for fault-tolerant distributed data fusion, Information Fusion 4, pp. 201-215, 2003.

[Vaccarella et al., 2013] A. Vaccarella, E. De Momi, A. Enquobahrie, G. Ferrigno, Unscented Kalman Filter Based Sensor Fusion for Robust Optical and Electromagnetic Tracking in Surgical Navigation, IEEE Transactions on Instrumentation and Measurement 62, pp. 2067-2081, 2013.

[van Atteveldt et al., 2014] N. van Atteveldt, M. M. Murray, G. Thut, C. E. Schroeder, Multisensory integration: flexible use of general operations, Neuron 19, pp. 1240-1253, 2014.

[van Ooyen et al., 2003] A. van Ooyen, J. van Pelt, M. A. Corner, S. B. Kater, Activity-dependent neurite outgrowth: Implications for network development and neuronal morphology, in Modeling Neural Development, Arjen van Ooyen (Ed.), pp. 111-132, 2003.

[Vermaak et al., 2005] J. Vermaak, S. J. Godsill, P. Perez, Monte Carlo Filtering for Multi-Target Tracking and Data Association, IEEE Transactions on Aerospace and Electronic Systems, pp. 309-332, 2005.

[Villaverde et al., 2014] A. F. Villaverde, J. Ross, F. Moran, J. R. Banga, MIDER: Network Inference with Mutual Information Distance and Entropy Reduction, PLoS One 9(5), 2014.

[von der Malsburg, 1999] C. von der Malsburg, The What and Why of Binding: The Modeler's Perspective, Neuron 24, pp. 95-104, 1999.

[Wan et al., 2000] W. Wan, D. Fraser, A Multiple Self-Organizing Map Scheme for Remote Sensing Classification, Lecture Notes in Computer Science: Multiple Classifier Systems, pp. 300-309, 2000.

[Wang et al., 2014] Z. Wang, Z. Dai, G. Gong, C. Zhou, Y. He, Understanding Structural-Functional Relationships in the Human Brain: A Large-Scale Network Perspective, Neuroscientist, 2014.

[Warren, 1990] T. Warren, Preliminary questions for the study of egomotion, in R. Warren, A. H. Wertheim (Eds.), Perception & Control of Self-motion, Hillsdale, pp. 3-33, 1990.

[Weber et al., 2007] C. Weber, S. Wermter, A self-organizing map of sigma-pi units, Neurocomputing, pp. 2552-2560, 2007.

[Weikersdorfer et al., 2012] D. Weikersdorfer, J. Conradt, Event-based Particle Filtering for Robot Self-Localization, Proc. of the IEEE International Conference on Robotics and Biomimetics (IEEE-ROBIO), pp. 866-870, 2012.

[Weiss et al., 2001] Y. Weiss, W. T. Freeman, On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs, IEEE Transactions on Information Theory, vol. 47, pp. 736-744, 2001.

[Wertheim, 1990] A. W. Wertheim, Visual, vestibular, and oculomotor interactions in the perception of object motion during egomotion, in R. Warren, A. H. Wertheim (Eds.), Perception & Control of Self-motion, Hillsdale, pp. 171-210, 1990.

[Westermann et al., 2007] G. Westermann, D. Mareschal, M. H. Johnson, S. Sirois, M. W. Spratling, M. S. Thomas, Neuroconstructivism, Dev. Sci. 10, pp. 75-83, 2007.

[White, 1991] F.E. White, Data Fusion Lexicon. Technical Panel For C3, San Diego, USA, Code 420, 1991.

[Wiener et al., 2011] J. M. Wiener, A. Berthoz, T. Wolbers, Dissociable cognitive mechanisms underlying human path integration, J. Exp. Brain Research 208, pp. 61-71, 2011.

[Wilson et al., 2009] R. Wilson, L. Finkel, A Neural Implementation of the Kalman Filter, Advances in Neural Information Processing Systems 22, pp. 2062-2070, 2009.

[Wiskott et al., 2002] L. Wiskott, T. J. Sejnowski, Slow Feature Analysis: Unsupervised Learning of Invariances, Neural Computation 14, pp. 715-770, 2002.

[Yangming et al., 2014] L. Yangming, L. Shuai, S. Quanjun, L. Hai, M.Q.-H. Meng, Fast and Robust Data Association Using Posterior Based Approximate Joint Compatibility Test, IEEE Transactions on Industrial Informatics, pp. 331-339, 2014.

[Zhang, 2001] J. Zhang, Dynamics and Formation of Self-Organizing Maps, in K. Obermayer, T. J. Sejnowski (Eds.), Self-Organizing Map Formation, Foundations of Neural Computation, MIT Press, Massachusetts, pp. 55-68, 2001.

[Zhang et al., 2008] T. Zhang et al., An FPGA implementation of insect-inspired motion detector for high-speed vision systems, Proc. of Intl. Conf. on Robotics and Automation, pp. 335-340, 2008.

[Zhou et al., 2014] G. Zhou, A. Liu, K. Yang, T. Wang, Z. Li, An Embedded Solution to Visual Mapping for Consumer Drones, Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, pp.670-675, 2014.

[Zhu et al., 2006] H. Zhu, O. Basir, A novel fuzzy evidential reasoning paradigm for data fusion with applications in image processing, J. of Soft Computing 10, pp. 1169-1180, 2006.

[Zulkifley et al., 2012] M. A. Zulkifley, B. Moran, Robust hierarchical multiple hypothesis tracker for multiple-object tracking, Expert Systems with Applications, Volume 39, Issue 16, pp. 12319-12331, 2012.