Inference Techniques in Low-Cost Sensor Networks

BY

Georgios Apostolakis

A THESIS
SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Master of Science
IN
Technical University of Crete
School of Electrical and Computer Engineering

Committee in Charge:

Professor Aggelos Bletsas (Supervisor)
Professor Michail G. Lagoudakis
Professor Antonios Deligiannakis

April 2023

Abstract

Distributed execution of algorithms across resource-constrained terminals has become increasingly popular, especially when fault tolerance is required. Asynchronous operation is brought to light in such scenarios, and in particular, *probabilistic asynchronous* operation, which models the failure probability of each terminal. The focus of this work is on the affine update model, which is applicable to a wide range of distributed inference algorithms. Applications include estimation of the average, solving linear systems, linear minimum mean square error estimation and spectral clustering, presented in detail in this thesis. Multimodal inference is also investigated, where two or more alternate sources of data are exploited for increased prediction accuracy. In that context, two variations of linear regression are presented, with uniform or Gaussian prior, which are equivalent to iterative affine updates. Furthermore, this work offers an asymptotic analysis for the arithmetic mean of the state vector, across a finite number of experiments, for the discovery of fixed points. It is shown that there are cases where the arithmetic mean behaves differently than the expected mean, and a sufficient condition is provided for convergence of the arithmetic mean to a fixed point. The lack of necessity for this condition is explained and subcases where the arithmetic mean converges, diverges, or has unpredictable behaviour are highlighted. Additionally, cases where the individual iterations never converge but their arithmetic mean does and offers fixed point are offered. Simulations corroborate the theoretical findings for various affine model setups. Finally, implementation details of a distributed low-cost sensor network are presented; the low-cost sensor network estimates the arithmetic mean of temperature across the network, by executing average consensus. The implementation is divided into hardware, network and software layers, and each layer is presented separately.

Acknowledgements

Firstly, I would like to express my sincere gratitude to Professor Aggelos Bletsas, my supervisor, for his invaluable guidance, unwavering support, and insightful ideas. Without his involvement, this work would not have been possible.

I am also grateful to Maria Manolikaki for her constant encouragement, patience, and support.

Furthermore, I would like to express my gratitude to my friends and colleagues from the lab, particularly George Vougioukas, George Perakis, Iosif Vardakis, Roza Chatzigeorgiou, and Vaggelis Giannelos, who offered suggestions, assistance and lots of unforgettable moments.

Moreover, I am deeply indebted to my family and close friends, for being there every time I needed them.

Finally, the Hellenic Foundation for Research and Innovation (HFRI) and SPACE HELLAS S.A. generously provided financial support for this research, for which I am deeply grateful. Their support was instrumental in addressing the problems that I aimed to investigate. The formal acknowledgement follows:

The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the "First Call for HFRI Research Projects to support Faculty members and Researchers and the Procurement of High-cost research equipment" (Project Number: 2846), as well as by SPACE HELLAS S.A. under the 'Dimitris Manolopoulos' postgraduate scholarship program.

Contents

Chapter 1

Introduction

Low-cost sensor networks have been gaining attention in various fields due to their importance and usefulness in monitoring and collecting data from the environment. These networks consist of small, low-power sensors that can be deployed in large numbers to gather data from different locations. They are cost-effective, easy to install, and can operate continuously for long periods. However, low-cost sensors have some limitations in terms of accuracy, precision, and reliability due to their low-quality components and lack of calibration.

Low-cost sensors have several advantages over traditional sensors, including their low cost, small size, and low power consumption. These sensors can be easily deployed in large numbers, providing dense spatial coverage for environmental monitoring and other applications. Low-cost sensors can also be integrated with wireless communication technologies, making it possible to transmit data remotely and in real-time.

However, low-cost sensors have some limitations that affect their accuracy and reliability. These sensors are typically less accurate and less precise than traditional sensors due to the lower quality of their components and lack of calibration. They are also more prone to drift and noise, which can affect the quality of the collected data. In addition, low-cost sensors may have limited durability, making them unsuitable for long-term monitoring applications. Finally, these sensors are prone to failures and may interrupt their operation temporarily, due to power or communication outage.

Inference techniques can be used to eliminate the limitations of low-cost sensors and provide accurate estimates of the underlying phenomenon. These techniques can compensate for the inaccuracies and noise in the sensor data and provide more reliable measurements. Inference techniques can also be used to reduce the power consumption of low-cost sensors by processing the data locally and transmitting only the essential information.

1.1 The problem of cost reduction

The problem of cost reduction is a critical issue when it comes to low-cost sensor networks. While these networks offer numerous benefits, including their ease of deployment, low power consumption, and real-time data transmission, the cost of building and maintaining them can be a significant barrier to their adoption.

One of the primary reasons why low-cost sensor networks can be expensive is the cost of the individual sensors. While the cost of sensors has been decreasing in recent years, the cost of high-quality sensors can still be prohibitive, especially when large numbers of sensors are needed. Additionally, the cost of the communication

infrastructure required to transmit data from the sensors to a central node can also be significant.

Another cost associated with low-cost sensor networks is the cost of maintenance. The sensors and communication infrastructure may require periodic maintenance, calibration, and replacement, which can add to the overall cost of the network. Additionally, the cost of storing and processing the data collected by the sensors can also be significant, especially if the data needs to be analyzed in real-time.

Reducing the cost of low-cost sensor networks is crucial for their widespread adoption and utilization. One way to reduce the cost is to use lower-cost sensors, which may be less accurate but can still provide valuable data. Additionally, developing low-cost communication infrastructure, such as mesh networks [1] or low-power wide-area networks (LPWANs) [2], can help to reduce the cost of data transmission.

Another approach to reducing the cost of low-cost sensor networks is to exploit the potential for distributed inference algorithms. These algorithms can be run locally on the sensors, reducing the amount of data that needs to be transmitted and processed centrally. This approach can help to reduce the cost of both data transmission and processing.

Finally, the use of free ambient energy sources, such as solar or wind power, can significantly reduce the cost of powering low-cost sensor networks [3]. These sources of energy can be used to power the sensors and communication infrastructure, reducing the need for expensive batteries or external power sources.

1.2 Data collection & processing techniques

Data collection in low-cost sensor networks typically involves the deployment of a large number of sensors in the target environment. These sensors can be distributed over a wide area and can collect data on various parameters, such as temperature, humidity, and air quality. The collected data is then either transmitted to a central node for processing and analysis, or processed by the network itself distributively [4].

There are several data collection techniques used in low-cost sensor networks, including time-based, event-based, and hybrid approaches [4], [5]. Time-based approaches involve the sensors collecting data at regular intervals, such as every minute or every hour. This approach is suitable for applications where data needs to be collected regularly, such as environmental monitoring. Event-based approaches, on the other hand, involve the sensors collecting data when specific events occur, such as when a threshold value is exceeded. This approach is suitable for applications such as intrusion detection. Finally, hybrid approaches combine both time-based and event-based approaches, providing greater flexibility in data collection. This approach is suitable for applications where both regular data collection and event-based data collection are needed, such as in agriculture.

Once the data has been collected, it needs to be processed and analysed. The processing and analysis of data can be performed using various techniques, including statistical analysis and machine learning. Statistical analysis involves the use of statistical models to analyse the collected data. This approach is suitable for appli-

cations where the data is relatively simple and well-understood. Machine learning techniques, on the other hand, involve the use of algorithms to learn from the collected data and make predictions or decisions based on that learning. This approach is suitable for applications where the data is complex and difficult to understand, such as in predictive maintenance or medical diagnosis.

1.3 Applications

Low-cost sensor networks have numerous potential applications across a wide range of industries and domains, such as:

- *Environmental monitoring* [6]: such networks can be used to collect real-time data on various environmental factors such as air quality, temperature, humidity, and noise levels. This data can be used to assess the impact of human activity on the environment and to develop strategies for mitigating these impacts.
- *Smart homes and buildings* [7]: these systems can monitor energy consumption, temperature, humidity, and occupancy levels. This data can be used to optimize energy usage and to improve the comfort and safety of occupants.
- *Healthcare* [8]: sensor networks can be used to monitor the health status of patients and to detect early warning signs of disease or medical conditions. This data can be used to provide timely medical interventions.
- *Agriculture* [9], [10]: these networks can monitor soil moisture levels, temperature, and nutrient levels. This data can be used to optimize crop growth and to reduce water and fertilizer usage.
- *Industrial process control* [11]: sensor networks can be used to monitor various parameters such as temperature, pressure, and flow rate. This data can be used to optimize production processes and to improve product quality.
- *Smart cities* [12]: such systems can be used to monitor traffic congestion, air quality, and noise levels. This data can be used to improve urban planning and to develop strategies for reducing the environmental impact of urbanization.

1.4 Thesis outline

This work provides an overview of the most popular affine inference algorithms that can be run by a distributed sensor network, analyses the asymptotic behaviour of the affine iterations under both synchronous and asynchronous operation (with emphasis on probabilistic asynchronous operation), and presents details of an actual implementation. Chapter 2 presents some inference algorithms which are equivalent to affine iterations and, thus, can be run at any distributed system. Chapter 3 discusses some extra affine inference algorithms (variations of linear regression) which combine

multiple sources of data, for increased accuracy and reliability. Also, chapter 4 analyses the asymptotic properties of the affine iterations, under both synchronous and asynchronous operation. Special emphasis is placed on the probabilistic asynchronous framework, and the differences on the convergence behaviour of the arithmetic and the expected mean are highlighted. Furthermore, chapter 5 presents the implementation of a low-cost sensor network which executes the average consensus algorithm. Finally, chapter 6 gives some conclusions and possible directions for future work.

1.5 Notation

Matrices and vectors are denoted with bold uppercase and lowercase letters, respectively. A_{ij} denotes the element in the i-th row and j-th column of matrix $\mathbf{A}$, while x_i denotes the element in the i-th position of vector $\mathbf{x}$. $\text{diag}\{\mathbf{x}\}$ implies a square matrix with all its elements equal to 0, except from its diagonal ones which are the elements of $\mathbf{x}$. For any matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\text{vec}\{\mathbf{A}\}$ is a linear transformation which converts $\mathbf{A}$ into a $mn \times 1$ column vector, obtained by stacking the columns of $\mathbf{A}$ on top of one another [13, p. 34, 2.4]. $\text{sgn}\{\mathbf{x}\}$ denotes a vector of same dimensions with $\mathbf{x}$, where its entries equal to $-1, 0$ or 1, depending on the sign of the respective entries of $\mathbf{x}$. Also, $\mathbf{0}$ implies the zero vector, $\mathbf{O}$ denotes the 2-dimensional matrix whose all entries are zero, and $\mathbf{I}$ implies the identity matrix. $\mathbf{A}^\dagger$ denotes the Moore-Penrose pseudoinverse [14, p. 221, 4.5.20] of matrix $\mathbf{A}$, while $|\mathbf{A}|$ stands for its determinant. Furthermore, $\mathbb{R}$ denotes the set of real numbers, while $\mathbb{R}^+$ denotes the set of non-negative real numbers. $\mathcal{U}(\mathbf{x}; -\mathbf{u}, \mathbf{u})$ implies that a real random vector $\mathbf{x}$ follows the uniform distribution bounded by vectors $-\mathbf{u}$ and $\mathbf{u}$; $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C})$ implies that a real random vector $\mathbf{x}$ follows the normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$. Also, i.i.d. stands for independent, identically distributed random variables, and pdf stands for the probability density function of a random variable. Additionally, $\mathbf{A} \otimes \mathbf{B}$ denotes the Kronecker product between $\mathbf{A}$ and $\mathbf{B}$, and $\mathcal{N}(\mathbf{A})$ stands for the nullspace [14, p. 174] of matrix $\mathbf{A}$. $\rho(\mathbf{A})$ implies the spectral radius of matrix $\mathbf{A}$, namely $\rho(\mathbf{A}) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$, where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $\mathbf{A}$. Additionally, $\lim_{t\to\infty} \mathbf{C}^{(t)} < \infty$ notation denotes that for every entry $c_{ij}^{(t)}$ of $\mathbf{C}^{(t)}$, there exists a (finite) constant k_i s.t. $\left|c_{ij}^{(t)}\right| \leq k_i$ for any t. Finally, $\lim_{t\to\infty} \mathbf{C}^{(t)} \to_{1..} \infty$ notation implies that *at least* one entry of $\mathbf{C}^{(t)}$ is required to increase to infinity, but not necessarily all of them.

Chapter 2

Affine Inference Techniques

Let's assume a vector $\mathbf{x}^{(t)} \in \mathbb{R}^n$ which represents the state of a system (i.e., the state of the i-th component at time t is represented by $x_i^{(t)}$). Moreover, let's assume that for any t, the current state of the system is an affine transformation of its previous state, i.e.,

$$\mathbf{x}^{(t)} = \mathbf{A}\mathbf{x}^{(t-1)} + \mathbf{b}. \tag{2.1}$$

This model has the benefits of both simplicity and applicability in a wide range of inference algorithms, with some of them further analysed in sections 2.1 - 2.3. Moreover, it can be easily mapped to a (possibly wireless) distributed network of terminals, which will perform their execution. Fig. 2.1 depicts an example of mapping a factor graph [15] onto a network with 3 terminals. Factors $\{g_i : \mathbb{R}^n \to 1\}$ are affine functions of the system's state vector. Notice that the allocation is not unique, but can be performed in a variety of alternative ways. Finally it is obvious that, depending on $\mathbf{A}$, message exchange between the various terminals may be required.

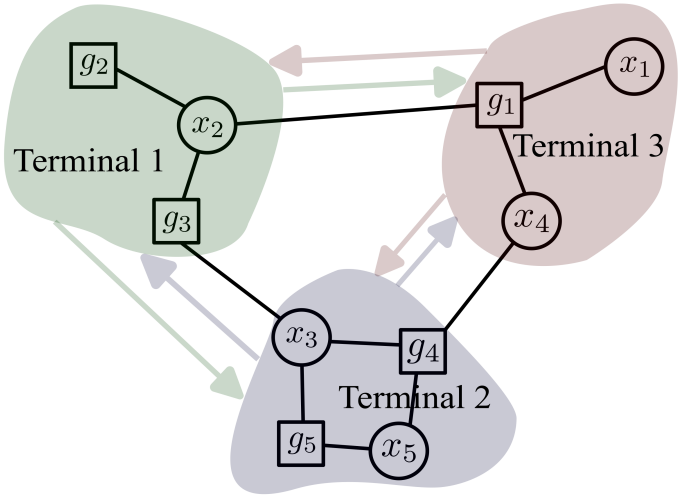


Figure 2.1: An example of mapping a factor graph onto a distributed network.

2.1 Average Consensus

Distributed consensus algorithms involve agents who communicate with each other across a network in a decentralized way, as described in [16], [17], until they reach on a common agreement. One of these algorithms is known as the average consensus [18],

which enables a group of agents to reach a consensus on the average of their initial values. The purpose of average consensus is to allow multiple agents to collaborate and coordinate their values in a distributed manner, without relying on a centralized authority or communication.

The average consensus algorithm works by having each agent update its value based on the values of its neighbors. At each time step, each agent sends its current value to its neighbors, and then updates its value based on the average of its neighbors' values. This process continues until all the agents converge to a common average value. If the values from all agents are stacked on a state vector $\mathbf{x}$, the matrix representation of the algorithm's (affine) iterations is the following:

$$\mathbf{x}^{(t)} = \mathbf{W}\mathbf{x}^{(t-1)}, \tag{2.2}$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$. Notice that the aforementioned convergence is guaranteed [18] if and only if $\mathbf{W}$ satisfies the following conditions:

$$\mathbf{1}^\top \mathbf{W} = \mathbf{1}^\top \tag{2.3}$$

$$\mathbf{W}^\top \mathbf{1} = \mathbf{1} \tag{2.4}$$

$$\rho\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^\top}{n}\right) < 1 \tag{2.5}$$

Let's denote with $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the graph which stems from the topology of the network (namely, two agents are connected with an edge if and only if they can exchange messages). Then, one (but not the only) family of matrices $\mathbf{W}$ which satisfy conditions (2.3) - (2.5) and guarantee convergence of average consensus to the desired fixed point [19] is:

$$W_{ij} = \begin{cases} \frac{1}{d+1}, & i \neq j, \{i,j\} \in \mathcal{E}, \\ 1 - \frac{d_i}{d+1}, & i = j \\ 0, & i \neq j, \{i,j\}, \notin \mathcal{E}, \end{cases} \tag{2.6}$$

where d_i is the degree of vertex i and d is the maximum degree of $\mathcal{G}$ [20].

Average consensus has the advantage of in-network inference, without the need for external mechanisms to aggregate data. It is particularly useful in sensor networks where it can be employed to estimate the average value of a physical parameter, such as temperature or humidity. Once the calculation is complete, all nodes within the network are aware about the estimated average. Additionally, it is often for the sensors to produce noisy measurements, especially when a decrease to their cost is attempted. Averaging of multiple noisy measurements (if taken under the same conditions) performs denoising and increases the sensing accuracy of the system.

2.2 Gaussian Belief Propagation

2.2.1 Theory

Belief propagation [21, p. 707, ch. 20] is a message-passing algorithm that is used to perform probabilistic inference in graphical models, such as Bayesian networks or

Markov random fields. The purpose of belief propagation is to efficiently calculate the marginal probabilities of the variables in the graphical model, given some evidence or observations. It works by passing messages between neighboring variables in the graph, which allows it to take advantage of the structure of the graph to make inference more efficient.

Gaussian Belief Propagation (GBP) [22], [21, p. 710, 20.2.3] is a special case of continuous Belief Propagation, where the variables of the graphical model follow a jointly Gaussian distribution. The exchanged messages between the nodes of the graph are valid Gaussian pdfs, thus every message consists of two numbers, its mean and its inverse variance (precision).

To use a matrix representation of the algorithm, assume that the variables of the model are stacked into a vector $\mathbf{x} \in \mathbb{R}^n$, which is random Gaussian with $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C})$. Its pdf in covariance form is

$$p(\mathbf{x}) \propto \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}, \tag{2.7}$$

while the same pdf in information form [21, p. 115, 4.3.3] is

$$p(\mathbf{x}) \propto \exp\left\{-\frac{1}{2}\mathbf{x}^\top \mathbf{J}\mathbf{x} + \mathbf{x}^\top \mathbf{h}\right\} \tag{2.8}$$

where $\mathbf{J} = \mathbf{C}^{-1}$ is the precision matrix and $\mathbf{h} = \mathbf{J}\boldsymbol{\mu}$ is the potential vector. As shown in [23], $\mathbf{J}$ and $\mathbf{h}$ can be factorized as

$$\mathbf{J} = \boldsymbol{\Lambda} + \boldsymbol{\Xi}^\top \boldsymbol{\Sigma} \boldsymbol{\Xi}, \tag{2.9}$$

$$\mathbf{h} = \boldsymbol{\Lambda}\boldsymbol{\xi} + \boldsymbol{\Xi}^\top \boldsymbol{\Sigma}\mathbf{u}, \tag{2.10}$$

where $\boldsymbol{\Lambda} \triangleq \operatorname{diag}\{\eta_1, \ldots, \eta_n\}$ with $\eta_i \geq 0 \, \forall i$, $\boldsymbol{\Sigma} \triangleq \operatorname{diag}\{\zeta_1, \ldots, \zeta_m\}$ with $\zeta_j > 0 \, \forall j$, $\boldsymbol{\Xi} \in \mathbb{R}^{m \times n}$, $\boldsymbol{\xi} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$.

According to [23], the expressions for Gaussian Belief Propagation under high-order factorization only require the factor-to-variable messages $m^{(t)}_{g_j \to x_i}(x_i)$ in each iteration. As mentioned above, these messages consist of two values: mean $\mu^{(t)}_{g_j \to x_i}(x_i)$ and precision $v^{(t)}_{g_j \to x_i}(x_i)$. However, [23] proves that the calculation of a precision only requires other precision parameters $v^{(t)}_{g_p \to x_q}(x_q)$. What is more, if they are initialized carefully, then the precisions are guaranteed to converge. Thus, convergence of GBP depends on convergence of the means $\mu^{(t)}_{g_j \to x_i}(x_i)$, whose computation involves both other precisions and means $\left\{\mu^{(t)}_{g_p \to x_q}(x_q), v^{(t)}_{g_p \to x_q}(x_q)\right\}$.

Assuming that the precisions $v^{(t)}_{g_j \to x_i}(x_i)$ are guaranteed to converge (due to a proper initialization), let's denote with $\boldsymbol{\mu}^{(t)}$ the vector which contains stacked all means $\mu^{(t)}_{g_j \to x_i}(x_i)$ from iteration t. Then, [23] proves that GBP is equivalent to the execution of (and converges simultaneously with) the following affine iterations:

$$\boldsymbol{\mu}^{(t)} = \mathbf{A}\boldsymbol{\mu}^{(t-1)} + \mathbf{c}, \tag{2.11}$$

where $\mathbf{A}$ and $\mathbf{c}$ are extracted from the converged values of the precisions $v_{g_j \to x_i}^{(t)}(x_i)$. Finally, the marginal beliefs $b^{(t)}(x_i)$ at time t are also Gaussian, such that $b^{(t)}(x_i) \sim \mathcal{N}\left(\epsilon_i^{(t)}, \sigma_i^{(t)}\right)$.

2.2.2 Applications

Solving Linear Systems

Consider the linear system

$$\mathbf{L}\mathbf{x} = \mathbf{g}, \tag{2.12}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{L} \in \mathbb{R}^{n \times n}$ and $\mathbf{g} \in \mathbb{R}^n$.

When matrix $\mathbf{L}$ is square, a solution can be reached by directly applying the affine updates. If in Eq. (2.1) it is set $\mathbf{A} = \mathbf{I} - \mathbf{L}$, $\mathbf{b} = \mathbf{g}$ and the iterations converge, then the fixed point is a solution of system (2.12).

For more generic systems where $\mathbf{L} \in \mathbb{R}^{m \times n}$ and $\mathbf{g} \in \mathbb{R}^m$ (provided that $m \geq n$ and $\mathbf{L}$ is a full rank matrix, for the system to be well determined), the affine updates cannot be applied directly, due to inconsistency at the assigned dimensions. In this case GBP can be exploited to get a solution. If its high-order factorization is considered as described above, and set $\mathbf{\Lambda} = \mathbf{O}_{n \times n}$, $\mathbf{\Xi} = \mathbf{L}$, $\mathbf{\Sigma} = \mathbf{I}$, $\xi = \mathbf{0}$ and $\mathbf{u} = \mathbf{g}$, then the algorithm will be applied on the following distribution:

$$\begin{aligned} \mathbf{x} \sim \mathcal{N}\left(\mathbf{x}; \left(\mathbf{L}^\top \mathbf{L}\right)^{-1} \mathbf{L}^\top \mathbf{g}, \left(\mathbf{L}^\top \mathbf{L}\right)^{-1}\right) \Leftrightarrow \\ \Leftrightarrow p(\mathbf{x}) \propto \exp\left\{-\frac{1}{2}\mathbf{x}^\top \mathbf{L}^\top \mathbf{L}\mathbf{x} + \mathbf{g}^\top \mathbf{L}\mathbf{x}\right\}. \end{aligned} \tag{2.13}$$

If GBP converges, then it will return the mean value of $\mathbf{x}$, i.e., $\left(\mathbf{L}^\top \mathbf{L}\right)^{-1} \mathbf{L}^\top \mathbf{g}$, which is also a solution of the linear system (2.12).

Linear Minimum Mean Square Error (LMMSE) Estimator

Consider a real Gaussian vector $\mathbf{x} \in \mathbb{R}^n$, such that $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Sigma}_\mathbf{x})$, where $\mathbf{\Sigma}_\mathbf{x}$ a positive definite matrix. Moreover, assume a real matrix $\mathbf{L} \in \mathbb{R}^{m \times n}$ and a Gaussian (independent of $\mathbf{x}$) vector $\mathbf{g} \in \mathbb{R}^m$, such that $\mathbf{g} \sim \mathcal{N}(\mathbf{g}; \mathbf{0}, \mathbf{\Sigma}_\mathbf{g})$, where again $\mathbf{\Sigma}_\mathbf{g}$ is a positive definite matrix. Finally, also consider the following system:

$$\mathbf{y} = \mathbf{L}\mathbf{x} + \mathbf{g}. \tag{2.14}$$

Now assume that the noisy measurements of Eq. (2.14) are given, and the linear minimum mean square error (LMMSE) estimator of $\mathbf{x}$ is asked for, let it be denoted by $\hat{\mathbf{x}}$. Then, as stated by [21, 24, 25], it is given from:

$$\hat{\mathbf{x}} = \left(\mathbf{\Sigma}_\mathbf{x}^{-1} + \mathbf{L}^\top \mathbf{\Sigma}_\mathbf{g}^{-1} \mathbf{L}\right)^{-1} \mathbf{L}^\top \mathbf{\Sigma}_\mathbf{g}^{-1} \mathbf{y}. \tag{2.15}$$

GBP can be employed to estimate $\hat{\mathbf{x}}$ under high order factorization, as described in [23]. For this problem $\mathbf{\Lambda} = \mathbf{\Sigma}_{\mathbf{x}}^{-1}$, $\mathbf{\Xi} = \mathbf{L}$, $\mathbf{\Sigma} = \mathbf{\Sigma}_{\mathbf{g}}^{-1}$, $\boldsymbol{\xi} = \mathbf{0}$ and $\mathbf{u} = \mathbf{y}$, and the algorithm will be applied on the following distribution:

$$\begin{aligned} &\mathbf{x} \sim \mathcal{N}\Big(\mathbf{x}; \left(\mathbf{\Sigma}_{\mathbf{x}}^{-1} + \mathbf{L}^{\top}\mathbf{\Sigma}_{\mathbf{g}}^{-1}\mathbf{L}\right)^{-1}\mathbf{L}^{\top}\mathbf{\Sigma}_{\mathbf{g}}^{-1}\mathbf{y}, \left(\mathbf{\Sigma}_{\mathbf{x}}^{-1} + \mathbf{L}^{\top}\mathbf{\Sigma}_{\mathbf{g}}^{-1}\mathbf{L}\right)^{-1}\Big) \Leftrightarrow \\ \Leftrightarrow\ & p(\mathbf{x}) \propto \exp\Big\{-\frac{1}{2}\mathbf{x}^{\top}\left(\mathbf{\Sigma}_{\mathbf{x}}^{-1} + \mathbf{L}^{\top}\mathbf{\Sigma}_{\mathbf{g}}^{-1}\mathbf{L}\right)\mathbf{x} + \mathbf{y}^{\top}\left(\mathbf{\Sigma}_{\mathbf{g}}^{-1}\right)^{\top}\mathbf{L}\mathbf{x}\Big\}. \end{aligned} \tag{2.16}$$

Notice that the inference of the expected value of pdf $p(\mathbf{x})$ yields the desired estimator $\hat{\mathbf{x}}$. As mentioned previously, it has been proved in [23] that the execution of GBP is equivalent to running the affine iterations from Eq. (2.1). Consequently, the LMMSE estimator $\hat{\mathbf{x}}$ can be identified as the fixed point of these affine iterations.

2.3 Spectral clustering on graphs

Spectral clustering [26] is the process of partitioning a graph into some clusters, based on the connectivity of the nodes to each other. Of all the cases, the connected graph (i.e., when the dimension of the nullspace of its Laplacian matrix [27] is 1) partitioned into 2 clusters is the most intriguing. Let λ_2 be the second smallest eigenvalue of the Laplacian matrix. Then, the respective eigenvector $\mathbf{u}_2$ is called the 'Fiedler' vector and separates the graph into 2 partitions, with $\text{sgn}(\mathbf{u}_2)$ denoting the cluster into which each node belongs.

Assume the following affine model, where with $\mathbf{L}$ is denoted the Laplacian matrix of a connected graph:

$$\mathbf{x}^{(t+1)} = \mathbf{L}\mathbf{x}^{(t)}. \tag{2.17}$$

Also, assume that $\mathbf{L}$ has one or more eigenvalues equal to 1, and the rest of them have a magnitude strictly less than 1. Moreover, let's denote with $\mathbf{R}$ the eigenspace of that eigenvalue in matrix $\mathbf{L}$ which is equal to 1. Then, it has been proved [28] that the mean value of the squared distance between $\mathbf{x}^{(t)}$ and $\mathbf{R}$ is bounded from $\psi^t \left\|(\mathbf{I} - \mathbf{R})\mathbf{x}^{(0)}\right\|_2^2$, where $\psi < 1$. Consequently, $\psi^t \to 0$ while $t \to \infty$ and $\mathbf{x}^{(t)}$ of Eq. (2.17) converges to the eigenspace of the eigenvalue $\lambda_{\mathbf{L}} = 1$ of the Laplacian matrix.

Spectral clustering is based on the idea that the state vector $\mathbf{x}^{(t)}$ will converge to the 'Fiedler' vector $\mathbf{u}_2$. However, the setup described above converges to it only when $\lambda_2 = 1$, which is a very special and rare case. For that reason, a scalar polynomial h can be constructed such that:

$$h(\lambda_i) \begin{cases} = 1, & \text{if } i = 2 \\ < 1 \text{ and } > -1, & \text{if } i \neq 2 \end{cases} \tag{2.18}$$

It is true that h is a scalar polynomial, but it can be easily turned into a matrix polynomial by substituting the scalar argument with the (square) matrix $\mathbf{L}$. That way we get $h(\mathbf{L})$ with the property that it has a unique eigenvalue equal to 1 and the

respective eigenvector is the Fiedler vector $\mathbf{u}_2$ of the matrix $\mathbf{L}$. Moreover, the other eigenvalues have a magnitude strictly less than 1. Consequently, the following affine iterations:

$$\mathbf{x}^{(t+1)} = h(\mathbf{L})\mathbf{x}^{(t)}, \tag{2.19}$$

will converge to the Fiedler vector $\mathbf{u}_2$ of the original graph. Finally, it's worth noting that the polynomial h is not unique.

Spectral clustering is a powerful technique in machine learning and data analysis that is used to group similar data points together based on their spectral properties. Some of its applications include:

- Image segmentation: Grouping of similar pixels based on their color and texture.
- Document clustering: Clustering of similar documents based on the similarity of their word frequency distributions.
- Market segmentation: Grouping of the customers based on their buying patterns and preferences.
- Genetics: Clustering of genes together based on their expression patterns, allowing for the identification of potential gene functions.
- Natural Language Processing: Grouping of text documents based on semantic similarity or topic modeling.

Chapter 3

Multimodal Affine Inference

Multimodal inference techniques can be a valuable approach for analyzing data from low-cost sensor networks. These techniques combine information from multiple sources or modes of data, such as data from different types of sensors or data collected at different times or locations. By combining information in this way, multimodal inference can improve the accuracy and reliability of data analysis in low-cost sensor networks. For example, combining data from a temperature sensor and a humidity sensor can provide more accurate estimates of environmental conditions than using either sensor alone. Machine learning algorithms, such as neural networks, can be used to model the relationships between different modes of data and make predictions or classifications based on the combined information. Multimodal inference techniques can help to overcome some of the limitations of low-cost sensors, such as incomplete or noisy data, and can enable more sophisticated analysis and decision-making based on the available data.

There are several machine learning algorithms that can be used for multimodal inference in low-cost sensor networks, including:

- **Neural Networks** [29], [30], [21]: They are a powerful class of machine learning algorithms that are particularly well-suited for handling complex and nonlinear relationships between different modes of data. They can be used to model the relationships between different types of sensors and predict missing values or classify data based on the combined information.

- **Random Forests** [31], [32]: They are an ensemble learning method that combines the predictions of multiple decision trees to make more accurate predictions. They can be used for multimodal inference by combining information from different types of sensors or data sources to make predictions or classifications.

- **Support Vector Machines (SVMs)** [29], [33]: They are a supervised learning method that can be used for classification and regression tasks. They are particularly well-suited for handling high-dimensional data and can be used for multimodal inference by combining information from different types of sensors or data sources.

- **Gradient Boosting** [34], [21]: It is also used for building predictive models and constitutes an improvement of a previous technique named 'AdaBoost'. It is an iterative algorithm that combines weak learners (often decision trees) to create a strong learner that can accurately predict the target variable. The idea behind gradient boosting is to fit each new tree to the negative gradient of the

loss function of the previous tree. In other words, the algorithm tries to correct the errors of the previous trees by adding new trees that focus on the areas where the previous trees performed poorly. One of the most popular variations of gradient boosting is eXtreme Gradient Boosting (XGBoost) [35].

- **Linear Regression** [36], [21]: It is a popular statistical method used for modeling the relationship between a dependent variable and one or more independent variables. Its goal is to find a relationship between them, which can then be used to make predictions about the dependent variable based on the values of the independent variables.

In the subsequent sections, an in-depth analysis of Linear Regression will be performed. Moreover, it will be proved that it is equivalent to an LMMSE estimator, and GBP can be employed to make predictions for the faulty sensors.

3.1 Model

In low-cost sensor networks, multimodal inference can be employed to establish a model that characterizes the relationship between one or more dependent variables, such as air quality, and multiple independent variables, such as temperature, humidity, and wind speed. These variables are measured using different types of sensors. By utilizing data from other sensors, missing or noisy data from one or more sensors can be compensated for, leading to an improvement in the accuracy and reliability of data analysis.

To be specific, consider a network comprising of N nodes, out of which m are operational, and M are faulty, where $m + M = N$. The measurements from the active sensors are denoted by $y_{(i)}$, $i = 1, \dots, m$, while the measurements from the faulty sensors are denoted by $y_{*(j)} = \text{null}$, $j = 1, \dots, M$. The vectors $\bar{\mathbf{y}} \in \mathbb{R}^m$ and $\bar{\mathbf{y}}_* \in \mathbb{R}^M$ are obtained by stacking all $y_{(i)}$ and $y_{*(i)}$, respectively. The dependent variable for this model is represented by the vector $\mathbf{y} = \begin{bmatrix} \bar{\mathbf{y}} \\ \bar{\mathbf{y}}_* \end{bmatrix} \in \mathbb{R}^N$.

Furthermore, assume that there exist g variables whose values are known for all nodes in the network. These variables may be measured in various ways, such as using aerial or ground vehicles, humans, etc. and may remain constant for extended periods. For instance, these variables could be the RGB colors of an aerial image of the field where the network has been installed, wind speed, solar radiation intensity, or geographical coordinates of the network's nodes. Let $\mathbf{x}_{(i)} \in \mathbb{R}^g$ denote the values of these variables at the i-th node of the network, whose sensor is operating normally ($i = 1, \dots, m$). Also, let $\mathbf{x}_{*(j)} \in \mathbb{R}^g$ denote the values of these variables at the j-th node of the network, whose sensor is faulty ($j = 1, \dots, M$). Matrices $\bar{\mathbf{X}} \in \mathbb{R}^{m \times g}$ and $\bar{\mathbf{X}}_* \in \mathbb{R}^{M \times g}$ are obtained by stacking $\mathbf{x}_{(i)}^\top$ and $\mathbf{x}_{*(j)}^\top$, respectively. Thus, matrix $\mathbf{X} = \begin{bmatrix} \bar{\mathbf{X}} \\ \bar{\mathbf{X}}_* \end{bmatrix} \in \mathbb{R}^{N \times g}$ represents the independent variables for all nodes of the system.

Finally, let's denote with $\phi : \mathbb{R}^g \to \mathbb{R}^d$, $d \geq g$ the transformation of a g-sized vector into an equal or higher-dimensional space. If ϕ is applied to every row of

$\mathbf{X}$, $\Phi(\mathbf{X})$ occurs which is a $N \times d$ matrix. $\Phi(\mathbf{X})$ is called a feature transformation function, and maps the original features $\mathbf{X}$ into a new feature space, where the non-linear relationship between $\mathbf{X}$ and the target variable $\mathbf{y}$ can be better captured. The choice of ϕ depends on the specific problem at hand, and can be based on domain knowledge, or can be learned from the data. However, a hard constraint on the choice of the transformation is that the d columns of $\Phi(\mathbf{X})$ have to be linearly independent, as discussed in Appendix A.

Some examples of feature transformation functions that are commonly used in linear regression include polynomial functions and radial basis functions. Other more complex feature transformation functions can also be used, such as neural networks or kernel methods, which can learn a non-linear mapping from the original feature space to the new feature space based on the data.

Overall, the purpose of this chapter is to provide techniques for predicting the value of missing sensor measurements, denoted by $\mathbf{y}_*$, based on the retrieved measurements $\mathbf{y}$ and independent variables $\mathbf{X}$.

3.2 Linear regression with uniform prior belief

Linear regression assumes the following relation between the dependent and independent variables:

$$\mathbf{y} = \Phi(\mathbf{X})\mathbf{w} + \mathbf{q}, \tag{3.1}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{q} \in \mathbb{R}^N$. $\mathbf{w}$ is assumed to follow the uniform distribution $\mathcal{U}(\mathbf{w}; -\mathbf{u}, \mathbf{u})$, where $\mathbf{u} \in \mathbb{R}^d$. Then, its pdf is $p_{\mathbf{w}}(\mathbf{w}) = \frac{1}{z}$, where $z \in \mathbb{R}^+$. Also, $\mathbf{q}$ is a Gaussian noise vector, and follows the normal distribution $\mathcal{N}(\mathbf{q}; \mathbf{0}, \sigma^2\mathbf{I})$, where $\sigma \geq 0$.

From Eq. (3.1) it can be inferred that

$$\bar{\mathbf{y}}|\mathbf{w} \sim \mathcal{N}\left(\bar{\mathbf{y}}; \Phi(\bar{\mathbf{X}})\mathbf{w}, \sigma^2\mathbf{I}\right). \tag{3.2}$$

The posterior pdf of $\mathbf{w}$ can be computed by the Bayes rule [21]:

$$p_{\mathbf{w}|\bar{\mathbf{y}}}(\mathbf{w}|\bar{\mathbf{y}}) = \frac{p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w})p_{\mathbf{w}}(\mathbf{w})}{p_{\bar{\mathbf{y}}}(\bar{\mathbf{y}})} \propto_{\mathbf{w}} p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w})p_{\mathbf{w}}(\mathbf{w}) = p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w})\frac{1}{z} \propto_{\mathbf{w}} p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w}) \tag{3.3}$$

From Eq. (3.3) arises that the Maximum A Posteriori (MAP) estimator $\mathbf{w}_{MAP} = \operatorname{argmax}_{\mathbf{w}} p_{\mathbf{w}|\bar{\mathbf{y}}}(\mathbf{w}|\bar{\mathbf{y}})$ is equivalent to its Maximum Likelihood (ML) estimator $\mathbf{w}_{ML} = \operatorname{argmax}_{\mathbf{w}} p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w})$ [21]:

$$\mathbf{w}_{MAP} = \mathbf{w}_{ML} = \operatorname*{argmax}_{\mathbf{w}} p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w}) =^{1} \left(\Phi(\bar{\mathbf{X}})^\top \Phi(\bar{\mathbf{X}})\right)^{-1} \Phi(\bar{\mathbf{X}})^\top \bar{\mathbf{y}} \tag{3.4}$$

[1] Proof is given in appendix A.

The missing sensor readings $\bar{\mathbf{y}}_*$ can be predicted from the following distribution:

$$\begin{aligned} p_{\bar{\mathbf{y}}_*|\bar{\mathbf{y}}}\left(\bar{\mathbf{y}}_*|\bar{\mathbf{y}}\right) &= \int_{-\infty}^{+\infty} p_{\bar{\mathbf{y}}_*|\bar{\mathbf{y}}}\left(\bar{\mathbf{y}}_*|\bar{\mathbf{y}}, \mathbf{w}\right) p_{\mathbf{w}|\bar{\mathbf{y}}}\left(\mathbf{w}|\bar{\mathbf{y}}\right) \mathrm{d}\mathbf{w} = \\ &\overset{\bar{\mathbf{y}} \perp \bar{\mathbf{y}}_*|\mathbf{w}}{=\!=\!=} \int_{-\infty}^{+\infty} p_{\bar{\mathbf{y}}_*|\mathbf{w}}\left(\bar{\mathbf{y}}_*|\mathbf{w}\right) p_{\mathbf{w}|\bar{\mathbf{y}}}\left(\mathbf{w}|\bar{\mathbf{y}}\right) \mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*} \\ &\overset{(3.3)}{\propto}_{\bar{\mathbf{y}}_*} \int_{-\infty}^{+\infty} p_{\bar{\mathbf{y}}_*|\mathbf{w}}\left(\bar{\mathbf{y}}_*|\mathbf{w}\right) p_{\bar{\mathbf{y}}|\mathbf{w}}\left(\bar{\mathbf{y}}|\mathbf{w}\right) \mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*} {}^{2} \\ &\propto_{\bar{\mathbf{y}}_*} \exp\left(-\frac{1}{2}(\bar{\mathbf{y}}_* - \mathbf{t})^\top \mathbf{P}^{-1}(\bar{\mathbf{y}}_* - \mathbf{t})\right), \end{aligned} \tag{3.5}$$

where

$$\mathbf{t} = \Phi(\bar{\mathbf{X}}_*)\mathbf{w}_{MAP} = \Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top \Phi(\bar{\mathbf{X}})\right)^{-1} \Phi(\bar{\mathbf{X}})^T \bar{\mathbf{y}} \tag{3.6}$$

$$\mathbf{P} = \sigma^2\mathbf{I} + \sigma^2\Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top \Phi(\bar{\mathbf{X}})\right)^{-1} \Phi(\bar{\mathbf{X}}_*)^\top \tag{3.7}$$

In conclusion, the predictive distribution is Gaussian, such that:

$$\begin{aligned} \bar{\mathbf{y}}_*|\bar{\mathbf{y}} \sim \mathcal{N}\Big(&\bar{\mathbf{y}}_*; \Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top \Phi(\bar{\mathbf{X}})\right)^{-1} \Phi(\bar{\mathbf{X}})^\top \bar{\mathbf{y}}, \\ &\sigma^2\mathbf{I} + \sigma^2\Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top \Phi(\bar{\mathbf{X}})\right)^{-1} \Phi(\bar{\mathbf{X}}_*)^\top\Big). \end{aligned} \tag{3.8}$$

As a result, the optimal value to predict for the missing measurements $\bar{\mathbf{y}}_*$ is the expected mean of the aforementioned distribution. Moreover notice that, if $\mathbf{L}$, $\mathbf{g}$ of Eq. (2.13) are replaced with $\Phi(\bar{\mathbf{X}})$, $\bar{\mathbf{y}}$, prediction of $\bar{\mathbf{y}}_*$ turns into solving a linear system. Thus, GBP can be employed, which is equivalent to running affine iterations, as explained in section 2.2.

Finally, if there is interest on the covariance of the prediction, hyperparameter σ^2 has to be estimated too (since it is required for the computation of the covariance matrix in Eq. (3.8)). From Eq. (3.1) it arises that

$$\mathbf{y}|\mathbf{w} \sim \mathcal{N}\left(\mathbf{y}; \Phi(\mathbf{X})\mathbf{w}, \sigma^2\mathbf{I}\right). \tag{3.9}$$

In Eq. (3.8) the MAP estimator of $\mathbf{w}$ is utilized, and σ^2 quantifies the variance of the partially known dependent variable $\mathbf{y}$ around the regression line $L(\Phi(\bar{\mathbf{X}}_*)) = \Phi(\bar{\mathbf{X}}_*)\mathbf{w}_{MAP}$. This variance can be estimated from the known elements of $\mathbf{y}$ as follows [36]:

$$\sigma^2 = \frac{\sum_{i=1}^{m}\left(y_{(i)} - \phi\left(\mathbf{x}_{(i)}^\top\right)\mathbf{w}_{MAP}\right)^2}{m - d} \tag{3.10}$$

Notice that σ is often called *regression standard error*.

[2] Proof is given in appendix B.

3.3 Linear regression with Gaussian prior belief

Again, the dependent variables are connected with the independent via the same model:

$$\mathbf{y} = \Phi(\mathbf{X})\mathbf{w} + \mathbf{q}, \tag{3.11}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{q} \in \mathbb{R}^N$. However, this model assumes that $\mathbf{w}$ follows the Gaussian distribution $\mathcal{N}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2\mathbf{I})$, where $\sigma_{\mathbf{w}} > 0$. Also, $\mathbf{q}$ is a Gaussian noise vector, and follows the normal distribution $\mathcal{N}(\mathbf{q}; \mathbf{0}, \sigma^2\mathbf{I})$, where $\sigma > 0$.

From Eq. (3.11) it can be inferred that

$$\bar{\mathbf{y}}|\mathbf{w} \sim \mathcal{N}\left(\bar{\mathbf{y}}; \Phi(\bar{\mathbf{X}})\mathbf{w}, \sigma^2\mathbf{I}\right). \tag{3.12}$$

The posterior pdf of $\mathbf{w}$ can be computed by the Bayes rule [21]:

$$\begin{aligned} p_{\mathbf{w}|\bar{\mathbf{y}}}(\mathbf{w}|\bar{\mathbf{y}}) = \frac{p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w})p_{\mathbf{w}}(\mathbf{w})}{p_{\bar{\mathbf{y}}}(\bar{\mathbf{y}})} &\propto_{\mathbf{w}} p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w})p_{\mathbf{w}}(\mathbf{w}) \propto_{\mathbf{w}} {}^{3} \\ &\propto_{\mathbf{w}} \exp\left(-\frac{1}{2}(\bar{\mathbf{y}} - \boldsymbol{\mu})^\top \boldsymbol{\Lambda}^{-1}(\bar{\mathbf{y}} - \boldsymbol{\mu})\right) \end{aligned} \tag{3.13}$$

where

$$\boldsymbol{\mu} = \left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Phi(\bar{\mathbf{X}})^\top\bar{\mathbf{y}} \tag{3.14}$$

$$\boldsymbol{\Lambda} = \left(\frac{1}{\sigma^2}\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1} \tag{3.15}$$

Thus, $\mathbf{w}|\bar{\mathbf{y}}$ follows a Gaussian distribution such that

$$\begin{aligned} \mathbf{w}|\bar{\mathbf{y}} \sim \mathcal{N}\Bigg(\mathbf{w}; &\left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Phi(\bar{\mathbf{X}})^\top\bar{\mathbf{y}}, \\ &\left(\frac{1}{\sigma^2}\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Bigg), \end{aligned} \tag{3.16}$$

which makes obvious that the Maximum A Posteriori (MAP) estimator $\mathbf{w}_{MAP}$ is:

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p_{\mathbf{w}|\bar{\mathbf{y}}}(\mathbf{w}|\bar{\mathbf{y}}) = \left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Phi(\bar{\mathbf{X}})^\top\bar{\mathbf{y}}. \tag{3.17}$$

The missing sensor readings $\bar{\mathbf{y}}_*$ can be predicted from the following distribution:

$$\begin{aligned} p_{\bar{\mathbf{y}}_*|\bar{\mathbf{y}}}(\bar{\mathbf{y}}_*|\bar{\mathbf{y}}) &= \int_{-\infty}^{+\infty} p_{\bar{\mathbf{y}}_*|\bar{\mathbf{y}}}(\bar{\mathbf{y}}_*|\bar{\mathbf{y}}, \mathbf{w})\, p_{\mathbf{w}|\bar{\mathbf{y}}}(\mathbf{w}|\bar{\mathbf{y}})\, \mathrm{d}\mathbf{w} = \\ &\overset{\bar{\mathbf{y}}\perp\bar{\mathbf{y}}_*|\mathbf{w}}{=\!=} \int_{-\infty}^{+\infty} p_{\bar{\mathbf{y}}_*|\mathbf{w}}(\bar{\mathbf{y}}_*|\mathbf{w})\, p_{\mathbf{w}|\bar{\mathbf{y}}}(\mathbf{w}|\bar{\mathbf{y}})\, \mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*} {}^{4} \\ &\propto_{\bar{\mathbf{y}}_*} \exp\left(-\frac{1}{2}(\bar{\mathbf{y}}_* - \mathbf{t})^\top\mathbf{P}^{-1}(\bar{\mathbf{y}}_* - \mathbf{t})\right), \end{aligned} \tag{3.18}$$

[3] Proof is given in appendix C.

[4] Proof is given in appendix D.

where

$$\mathbf{t} = \Phi(\bar{\mathbf{X}}_*)\mathbf{w}_{MAP} = \Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Phi(\bar{\mathbf{X}})^T\bar{\mathbf{y}} \tag{3.19}$$

$$\mathbf{P} = \sigma^2\mathbf{I} + \sigma^2\Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Phi(\bar{\mathbf{X}}_*)^\top \tag{3.20}$$

In conclusion, the predictive distribution is Gaussian, such that:

$$\begin{aligned}\bar{\mathbf{y}}_*|\bar{\mathbf{y}} \sim \mathcal{N}\Bigg(&\bar{\mathbf{y}}_*; \Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Phi(\bar{\mathbf{X}})^\top\bar{\mathbf{y}},\\ &\sigma^2\mathbf{I} + \sigma^2\Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Phi(\bar{\mathbf{X}}_*)^\top\Bigg).\end{aligned} \tag{3.21}$$

As a result, the optimal value to predict for the missing measurements $\bar{\mathbf{y}}_*$ is the expected mean of the aforementioned distribution. Moreover notice that, if $\boldsymbol{\Sigma}_{\mathbf{x}}$, $\boldsymbol{\Sigma}_{\mathbf{g}}$, $\mathbf{L}$, $\mathbf{y}$ of Eq. (2.15) are replaced with $\frac{\sigma_{\mathbf{w}}^2}{\sigma^2}\mathbf{I}$, $\mathbf{I}$, $\Phi(\bar{\mathbf{X}})$, $\bar{\mathbf{y}}$, prediction of $\bar{\mathbf{y}}_*$ turns into LMMSE estimation. Thus, GBP can be employed, which is equivalent to running affine iterations, as explained in section 2.2.

Hyperparameter σ (regression standard error) can be estimated as shown in Eq. (3.10).

Also, $\lambda = \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}$ is called *regularization hyperparameter*, and controls the balance between model complexity and predictive accuracy. By selecting an appropriate value for λ, performance of the model can be improved and overfitting on the training data $\bar{\mathbf{y}}$ is prevented. The optimal value is typically determined through *cross-validation* [29, p. 119, 11.2.4], [21, p. 22, 1.4.8], where the model is trained on a subset of the data and evaluated on the remaining data. A higher value of the regularization parameter will result in a simpler model with lower variance but potentially higher bias, while a lower value will result in a more complex model with higher variance but potentially lower bias.

Chapter 4

Asymptotic Behaviour of Affine Iterations

This chapter analyzes the asymptotic behavior of the iterative affine updates of Eq. (2.1). Convergence to a fixed point is a prerequisite for any algorithm utilizing them to terminate and return a result. However, it cannot be guaranteed for any affine model, and conditions have to be extracted which allow the distinction of the convergent from the divergent setups.

4.1 Synchronous model

From Eq. (2.1) it occurs that:

$$x_i^{(t)} = \sum_{j=1}^{n} a_{ij} x_j^{(t-1)} + b_i,\ \forall i \in \{1, \ldots, n\}. \tag{4.1}$$

Consequently, in order to update the value of any component x_i during iteration t, it is necessary to have all variables x_j (where $j \neq i$ and $a_{ij} \neq 0$) from the previous iteration $t-1$ readily available. Due to this constraint, *synchronous operation* arises, as no output update can take place for any element of $\mathbf{x}$ during a particular iteration until all necessary input is available for all variables.

The existence of a fixed point for the synchronous operation is of special importance, since it indicates whether an algorithm that implements the AFP iterations can reach to a solution or not. Specifically, $\mathbf{x}_*$ is a fixed point if and only if:

$$\mathbf{x}_* = \mathbf{A}\mathbf{x}_* + \mathbf{b} \Leftrightarrow (\mathbf{I} - \mathbf{A})\,\mathbf{x}_* = \mathbf{b}. \tag{4.2}$$

Notice that there may exist 0, 1 or infinitely many fixed points, depending on matrix $\mathbf{A}$. Let's denote with m the geometric multiplicity of eigenvalue $\lambda = 1$ of $\mathbf{A}$. Thus, the nullspace of $(\mathbf{I} - \mathbf{A})$ has dimension m. If $m = 0$ then Eq. (4.2) will have a unique solution for any $\mathbf{b} \in \mathbb{R}^n$. On the other hand, if $m \geq 1$ then Eq. (4.2) may have either 0 or infinitely many solutions.

The convergence conditions of the synchronous operation of Eq. (2.1) are known [3], [37], and are presented below:

Theorem 1. *The synchronous affine iterations of Eq.* (2.1) *converge to a fixed point if and only if:*

- $\mathbf{b} \neq \mathbf{0}$ *and* $\rho(\mathbf{A}) < 1$, *or*

- $\mathbf{b} = \mathbf{0}$ *and* $\rho(\mathbf{A}) \leq 1$, *where* $\lambda = 1$ *is the only eigenvalue of* $\mathbf{A}$ *on the unit circle, as well as semisimple [14, p. 510].*

Proof. Eq. (2.1) can be rewritten as:

$$\mathbf{x}^{(t)} = \mathbf{A}\mathbf{x}^{(t-1)} + \mathbf{b} = \mathbf{A}^t\mathbf{x}^{(0)} + \left(\sum_{j=0}^{t-1} \mathbf{A}^t\right)\mathbf{b} \tag{4.3}$$

Thus, a fixed point $\mathbf{x}_*$ exists if and only if:

$$\mathbf{x}_* = \lim_{t\to\infty} \mathbf{x}^{(t)} = \lim_{t\to\infty} \mathbf{A}^t\mathbf{x}^{(0)} + \lim_{t\to\infty}\left(\sum_{j=0}^{t-1} \mathbf{A}^t\right)\mathbf{b} \in \mathbb{R}^n \tag{4.4}$$

According to [14], the following statements hold:

Lemma 1. *For* $\mathbf{A} \in \mathbb{R}^{n\times n}$, *the following statements are equivalent:*

- $\sum_{j=0}^{t-1} \mathbf{A}^\infty$ *converges.*
- $\rho(\mathbf{A}) < 1$.
- $\lim_{t\to\infty} \mathbf{A}^t = \mathbf{0}$.

In which case, $(\mathbf{I} - \mathbf{A})^{-1}$ *exists and* $\sum_{t=0}^{\infty} \mathbf{A}^t = (\mathbf{I} - \mathbf{A})^{-1}$.

Lemma 2. *For* $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\lim_{t\to\infty} \mathbf{A}^t$ *exists if and only if:*

- $\rho(\mathbf{A}) < 1$, *or*
- $\rho(\mathbf{A}) = 1$, *where* $\lambda = 1$ *is the only eigenvalue on the unit circle, as well as semisimple [14, p. 510].*

Finally, if Lemmas 1, 2 are combined with the condition stated by Eq. (4.4), the result of Theorem 1 arises and proof is complete. ■

4.2 Deterministic asynchronous model

A key feature required in modern (wired or wireless) networks is the tolerance to failures; some terminals may interrupt their operation temporarily, due to power or communication outage. This requirement brings asynchronous operation to the light; when some terminals cannot operate, the rest of them continue exchanging messages by assuming that the incoming messages from the unavailable terminals remain the same (and thus, the corresponding variables retain their current values). For example, work in [3] demonstrates distributed and asynchronous execution of inference algorithms in an embedded wireless network, ambiently powered by solar panels, where terminals could suspend their communication depending on the available energy budget.

Asynchronous operation relies on the assumption that every variable $x_i^{(t)}$ may either update its value or keep the same for the next iteration. Consequently, only a subset of the entries of $\mathbf{x}$ is updated at each iteration. More specifically, assume the following variable, for any $i \in \{1, \ldots, n\}$:

$$\psi_i^{(t)} = \begin{cases} 1, & \text{if } x_i \text{ is updated at iteration } t, \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

Let $\boldsymbol{\psi}^{(t)}$ the vector obtained when $\left\{\psi_i^{(t)}\right\}$ (for all $i \in \{1, \ldots n\}$) are stacked. Moreover, let $\boldsymbol{\Psi}^{(t)} = \text{diag}\{\boldsymbol{\psi}^{(t)}\}$ the diagonal matrix with $\boldsymbol{\psi}^{(t)}$ at its diagonal. In that case, the update of Eq. (2.1) becomes:

$$\begin{aligned} \mathbf{x}^{(t)} &= \boldsymbol{\Psi}^{(t)} \left(\mathbf{A}\mathbf{x}^{(t-1)} + \mathbf{b}\right) + \left(\mathbf{I} - \boldsymbol{\Psi}^{(t)}\right) \mathbf{x}^{(t-1)} = \\ &= \left(\boldsymbol{\Psi}^{(t)}\mathbf{A} + \mathbf{I} - \boldsymbol{\Psi}^{(t)}\right) \mathbf{x}^{(t-1)} + \boldsymbol{\Psi}^{(t)}\mathbf{b}. \end{aligned} \tag{4.6}$$

A vector $\mathbf{x}_*$ is a fixed point in asynchronous operation, if and only if, for any possible value of $\boldsymbol{\Psi}^{(t)}$, the following equation holds:

$$\mathbf{x}_* = \left(\boldsymbol{\Psi}^{(t)}\mathbf{A} + \mathbf{I} - \boldsymbol{\Psi}^{(t)}\right) \mathbf{x}_* + \boldsymbol{\Psi}^{(t)}\mathbf{b} \Leftrightarrow \boldsymbol{\Psi}^{(t)} \left(\mathbf{I} - \mathbf{A}\right) \mathbf{x}_* = \boldsymbol{\Psi}^{(t)}\mathbf{b}. \tag{4.7}$$

Lemma 3. *Any fixed point in the synchronous case is a fixed point in the asynchronous operation too and vice versa.*

Proof. Notice that if $\boldsymbol{\Psi}^{(t)} = \mathbf{I}$ for any t, then Eq. (4.6) is simplified to the synchronous case of Eq. (2.1). When Eq. (4.7) holds for $\boldsymbol{\Psi}^{(t)} = \mathbf{I}$, it also holds for any $\boldsymbol{\Psi}^{(t)} \neq \mathbf{I}$. As a result, Eq. (4.7) holds for any value of $\boldsymbol{\Psi}^{(t)}$ if and only if Eq. (4.2) holds. This leads to the desired conclusion. ■

To ensure convergence of the asynchronous updates of Eq. (4.6), it must be ensured that there exists *no* possible sequence $\left\{\boldsymbol{\Psi}^{(1)}, \boldsymbol{\Psi}^{(2)}, \ldots\right\}$ which will make Eq. (4.6) diverge. If at least one such sequence exists, then the model is assumed to be a divergent one.

[37] contains a remarkable study on the asynchronous affine model. Moreover it proves Theorem 2, which provides a criterion to check whether the aforementioned definition of convergence is satisfied or not.

Theorem 2. *If $\rho(|\mathbf{A}|) < 1$, then the iterative updates of Eq. (4.6) will converge for any chosen sequence of matrices $\boldsymbol{\Psi}^{(t)}$. Otherwise, there exists at least one sequence $\left\{\boldsymbol{\Psi}^{(1)}, \boldsymbol{\Psi}^{(2)}, \ldots\right\}$ for which the iterations will diverge.*

4.3 Probabilistic asynchronous model

Probabilistic asynchronous model makes the same assumptions with the deterministic asynchronous setup, and additionally that the $\psi_i^{(t)}$ variables are random and

follow the Bernoulli distribution with parameters $p_i > 0$[1]:

$$\psi_i^{(t)} = \begin{cases} 1, & \text{with probability } p_i, \\ 0, & \text{with probability } 1 - p_i. \end{cases} \tag{4.8}$$

Intuitively, this means that each terminal of the network is a random variable with a predetermined probability of failure at every iteration. Moreover, notice that $\psi_i^{(t)}$ are independent across both t and i, thus Lemma 3 also holds on this case (since the probability of $\mathbf{\Psi}^{(t)} = \mathbf{I} = \prod_i p_i > 0$). As a result, $\boldsymbol{\psi}^{(t)}$ is a n-dimensional Bernoulli process and the following can be defined:

$$\mathbf{P} = \mathbb{E}\left[\mathbf{\Psi}^{(t)}\right] = \operatorname{diag}\left\{[p_1, \ldots, p_n]^\top\right\} \in \mathbb{R}^{n \times n}. \tag{4.9}$$

This model is also described by Eqs. (4.6), (4.7). However, the criterion of Theorem 2 cannot be applied here because it does not consider the probability of appearance of each sequence, which might be zero for the divergent ones. In other words, the probabilistic asynchronous model may converge even in cases where the deterministic asynchronous setup diverges.

Most recent studies have focused on the aforementioned probabilistic model, with studying the convergence of the affine iterations in the mean square sense. The following definition [38, 7.2.6] is utilized in all of them:

Definition 1. *Let a sequence X_n and a constant $X_* \in \mathbb{R}$. Then, X_n converges to X_* (while $n \to \infty$) in the mean square sense if and only if:*

$$\mathbb{E}\left[(X_n - X_*)^2\right] \to 0. \tag{4.10}$$

On the randomized asynchronous affine model the sequence X_n is replaced by the state vector $\mathbf{x}^{(t)}$ and the fixed point X_* by $\lim_{t\to\infty} \mathbb{E}\left[\mathbf{x}^{(t)}\right]$. Moreover, it has to be ensured that $\lim_{t\to\infty} \mathbb{E}\left[\mathbf{x}^{(t)}\right]$ converges to a fixed point, in order for the definition to be satisfied. Consequently, the asynchronous affine updates converge in the mean square sense if and only if:

$$\begin{aligned} \lim_{t\to\infty} \mathbb{E}\left[\mathbf{x}^{(t)}\right] &\in \mathbb{R}^n \\ \operatorname{Cov}\left[\mathbf{x}^{(t)}\right] &\to \mathbf{0} \end{aligned} \tag{4.11}$$

Significant conclusions have been extracted by [23], which studies the convergence in the mean square sense and provides some necessary and sufficient conditions for its achievement. Moreover, [39], [40] study the asymptotic properties of some generalizations of the affine model, in the mean square sense again. It's worth noticing that, both [23] and [39] reach to the same conditions for mean square convergence (however the equivalence is not obvious and the proof is given in Appendix E).

[1] When $p_i = 0$ then $\operatorname{Prob}\left(\psi_i^{(t)} = 1\right) = 0$ and variable $x_i^{(t)}$ will be never updated, so it can be set equal to $x_i^{(0)}$ and be omitted from $\mathbf{x}^{(t)}$.

4.3.1 Preliminaries

In order to study the probabilistic asynchronous affine updates, some notation from seminal work in [40] is adopted. First of all, $\mathbf{V} \in \mathbb{R}^{n\times m}$ denotes an orthonormal basis for the nullspace of $(\mathbf{I}-\mathbf{A})$, i.e., $\mathcal{N}(\mathbf{I}-\mathbf{A})$. Then, $\mathbf{P_v} = \mathbf{V}\left(\mathbf{V}^T\mathbf{V}\right)^{-1}\mathbf{V}^T$ is the projection matrix onto $\mathcal{N}(\mathbf{I}-\mathbf{A})$, and thus, for any vector $\mathbf{u}\in\mathbb{R}^n$ it holds that $\mathbf{P_v u}\in\mathcal{N}(\mathbf{I}-\mathbf{A})$. Notice that $\mathbf{P_v}$ can be also rewritten as $\mathbf{P_v} = \mathbf{V}\mathbf{V}^\dagger$. Moreover, let's define

$$\mathbf{Q} = \mathbf{I} - \mathbf{P_v}, \tag{4.12}$$

the projection matrix onto the orthogonal complement of $\mathcal{N}(\mathbf{I}-\mathbf{A})$. Notice that when $(\mathbf{I}-\mathbf{A})$ is full-rank (namely Eq. (4.2) has a unique solution), then $\mathbf{P_v} = \mathbf{O}$ and $\mathbf{Q} = \mathbf{I}$.

From now on, it is assumed that at least one fixed point exists. Also, let's consider the fixed point with the minimum Euclidean norm [14, p. 423]:

$$\mathbf{x}_{**} = (\mathbf{I}-\mathbf{A})^\dagger\mathbf{b} = \left\{\underset{\mathbf{x}}{\operatorname{argmin}}\,\|\mathbf{x}\|_2,\ \text{s.t.}\ (\mathbf{I}-\mathbf{A})\mathbf{x} = \mathbf{b}\right\}.$$

Then,

$$\begin{aligned}\mathbf{Q}\mathbf{x}_{**} &= (\mathbf{I}-\mathbf{A})^\dagger\mathbf{b} - \mathbf{P_v}(\mathbf{I}-\mathbf{A})^\dagger\mathbf{b} = \\ &= (\mathbf{I}-\mathbf{A})^\dagger\mathbf{b} - \mathbf{V}\left((\mathbf{I}-\mathbf{A})\mathbf{V}\right)^\dagger\mathbf{b} \overset{\mathbf{V}\in\mathcal{N}(\mathbf{I}-\mathbf{A})}{=\!=\!=} \mathbf{x}_{**}.\end{aligned} \tag{4.13}$$

Moreover, notice that any point $\mathbf{x}_*$ is a fixed point if and only if it can be written as

$$\mathbf{x}_* = \mathbf{x}_{**} + \mathbf{c}, \quad \mathbf{c}\in\mathcal{N}(\mathbf{I}-\mathbf{A}), \tag{4.14}$$

or equivalently, if and only if

$$\begin{aligned}\mathbf{Q}\mathbf{x}_* &= \mathbf{Q}(\mathbf{x}_{**}+\mathbf{c}) \overset{(4.13)}{=\!=} \mathbf{x}_{**} + \mathbf{Q}\mathbf{c} \overset{\mathbf{c}=\mathbf{P_v}\mathbf{c}}{=\!=\!=} \\ &= \mathbf{x}_{**} + (\mathbf{I}-\mathbf{P_v})\mathbf{P_v}\mathbf{c} \overset{\mathbf{P_v^2}=\mathbf{P_v}}{=\!=\!=} \mathbf{x}_{**}.\end{aligned} \tag{4.15}$$

Therefore, the set of fixed points is denoted as follows:

$$\mathcal{X}_* = \left\{\mathbf{x}_* : \mathbf{x}_* = \mathbf{x}_{**} + \mathbf{c},\quad \mathbf{c}\in\mathcal{N}(\mathbf{I}-\mathbf{A})\right\}. \tag{4.16}$$

Finally, vector $\mathbf{r}^{(t)}$ is defined as the residual of $\mathbf{x}^{(t)}$ from the nearest point in $\mathcal{X}_*$:

$$\mathbf{r}^{(t)} = \mathbf{Q}(\mathbf{x}^{(t)} - \mathbf{x}_{**}). \tag{4.17}$$

The utility of such definition will become obvious immediately below.

4.3.2 Convergence properties

Initially Lemmas 4, 5 are given with useful results that will be utilized to extract the convergence properties of the probabilistic affine iterations.

Lemma 4. $\lim_{t\to\infty}\mathbf{x}^{(t)}$ *exists (i.e., it gets real values from the set of fixed points* $\mathcal{X}_*$*) if and only if* $\lim_{t\to\infty}\mathbf{r}^{(t)}=\mathbf{0}$.

Proof.

$$\begin{aligned}&\lim_{t\to\infty}\mathbf{r}^{(t)}=\mathbf{0}\Leftrightarrow\lim_{t\to\infty}\mathbf{Q}\mathbf{x}^{(t)}=\mathbf{Q}\mathbf{x}_{**}\overset{(*),\ (4.13)}{\Longleftrightarrow}\mathbf{Q}\lim_{t\to\infty}\mathbf{x}^{(t)}=\mathbf{x}_{**}\overset{(4.15)}{\Longleftrightarrow}\\&\Leftrightarrow\lim_{t\to\infty}\mathbf{x}^{(t)}\in\mathcal{X}_*\overset{(4.14),(4.16)}{\Longleftrightarrow}\lim_{t\to\infty}\mathbf{x}^{(t)}\in\mathbb{R}^n.\end{aligned}\tag{4.18}$$

Note: At point $(*)$ the existence of $\lim_{t\to\infty}\mathbf{x}^{(t)}$ is guaranteed. In a case where that limit did not exist, the state vector should constantly change its value, but that value should always belong to set $\mathcal{X}_*$ (for efficiently large t). However, the set $\mathcal{X}_*$ contains the fixed points of the iterations, i.e. the points where the state vector remains unchanged through the asynchronous updates. This fact makes the existence of $\lim_{t\to\infty}\mathbf{x}^{(t)}$ equivalent to $\lim_{t\to\infty}\mathbf{r}^{(t)}=\mathbf{0}$. ■

Notice that when $\mathbf{Q}=\mathbf{I}$, then convergence of $\mathbf{r}^{(t)}$ to zero is equivalent with convergence of $\mathbf{x}^{(t)}$ to $\mathbf{x}_{**}$. What is more, when $\mathbf{Q}\neq\mathbf{I}$ and infinite fixed points exist, then convergence of $\mathbf{r}^{(t)}$ to zero is equivalent with convergence of $\mathbf{x}^{(t)}$ to any one of them and that's why it has to be projected on the subspace that they define. On the contrary, when $\mathbf{Q}\neq\mathbf{I}$ and no fixed point exists, then $\mathbf{r}^{(t)}$ cannot converge to $\mathbf{0}$.

Lemma 5. *The expectation of the asynchronous affine updates* $\mathbb{E}\left[\mathbf{x}^{(t)}\right]$ *converges to a fixed point if and only if one of the following criteria holds:*

- $\mathbf{b}\neq\mathbf{0}$ *and* $\rho\left(\mathbf{PA}+\mathbf{I}-\mathbf{P}\right)<1$, *or* (4.19)
- $\mathbf{b}=\mathbf{0}$ *and* $\rho\left(\mathbf{PA}+\mathbf{I}-\mathbf{P}\right)\leq 1$, *where* $\lambda=1$ *is the only eigenvalue of* $\left(\mathbf{PA}+\mathbf{I}-\mathbf{P}\right)$ *on the unit circle, as well as semisimple.* (4.20)

Proof.

$$\begin{aligned}&\lim_{t\to\infty}\mathbb{E}\left[\mathbf{x}^{(t)}\right]\in\mathbb{R}^n\Leftrightarrow\\&\Leftrightarrow\lim_{t\to\infty}\mathbb{E}\left[\left(\mathbf{\Psi}^{(t)}\mathbf{A}+\mathbf{I}-\mathbf{\Psi}^{(t)}\right)\mathbf{x}^{(t-1)}+\mathbf{\Psi}^{(t)}\mathbf{b}\right]\in\mathbb{R}^n\Leftrightarrow\\&\Leftrightarrow\lim_{t\to\infty}\mathbb{E}\left[\prod_{i=1}^{t}\left(\mathbf{\Psi}^{(i)}\mathbf{A}+\mathbf{I}-\mathbf{\Psi}^{(i)}\right)\right]\mathbf{x}^{(0)}+\\&\quad+\lim_{t\to\infty}\sum_{i=1}^{t}\mathbb{E}\left[\left(\prod_{j=i+1}^{t}\left(\mathbf{\Psi}^{(j)}\mathbf{A}+\mathbf{I}-\mathbf{\Psi}^{(j)}\right)\right)\mathbf{\Psi}^{(i)}\right]\mathbf{b}\in\mathbb{R}^n\overset{(*)}{\Leftrightarrow}\end{aligned}\tag{4.21}$$

$$\overset{(*)}{\Leftrightarrow} \lim_{t\to\infty} \left(\prod_{i=1}^{t} \mathbb{E}\left[\boldsymbol{\Psi}^{(i)}\mathbf{A} + \mathbf{I} - \boldsymbol{\Psi}^{(i)}\right] \right) \mathbf{x}^{(0)} + \\ + \lim_{t\to\infty} \sum_{i=1}^{t} \left(\prod_{j=i+1}^{t} \mathbb{E}\left[\boldsymbol{\Psi}^{(j)}\mathbf{A} + \mathbf{I} - \boldsymbol{\Psi}^{(j)}\right] \right) \mathbb{E}\left[\boldsymbol{\Psi}^{(i)}\right] \mathbf{b} \in \mathbb{R}^n \Leftrightarrow$$

$$\Leftrightarrow \lim_{t\to\infty} (\mathbf{PA} + \mathbf{I} - \mathbf{P})^t \mathbf{x}^{(0)} + \lim_{t\to\infty} \sum_{i=0}^{t-1} (\mathbf{PA} + \mathbf{I} - \mathbf{P})^i \mathbf{Pb} \in \mathbb{R}^n,$$

where the equivalence at $(*)$ holds due to the fact that $\boldsymbol{\Psi}^{(t)}$ are independent across t.

For a given matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, it occurs that $\lim_{t\to\infty} \sum_{i=0}^{t} \mathbf{B}^i$ converges if and only if $\rho(\mathbf{B}) < 1$ [14, p. 618, 7.10.11]. Moreover, $\lim_{t\to\infty} \mathbf{B}^t$ converges if and only if $\rho(\mathbf{B}) < 1$ or $\rho(\mathbf{B}) = 1$ with the prerequisite that the unitary-magnitude eigenvalues of $\mathbf{B}$ are all equal to 1 and semisimple [14, p. 630, 7.10.33].

As a result, in the general case where $\mathbf{b} \neq \mathbf{0}$ the above statement is true if and only if condition of Eq. (4.19) holds. However, in the special case where $\mathbf{b} = \mathbf{0}$, the statement holds when condition of Eq. (4.20) is satisfied. ■

At this point the main result of this section can be given, in the form of a theorem. Notice that $\text{avg}\left\{\mathbf{x}^{(t)}\right\}$ denotes the arithmetic mean of the state vector $\mathbf{x}^{(t)}$ after the execution of multiple experiments; specifically, for ℓ executed experiments, with $\mathbf{x}_i^{(t)}$ the state vector from the i-th experiment $(1 \leq i \leq \ell)$, the arithmetic mean follows:

$$\text{avg}\left\{\mathbf{x}^{(t)}\right\} = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{x}_i^{(t)}. \tag{4.22}$$

Also, the following matrices are defined:

$$\mathbf{J} \triangleq \text{diag}\left\{\text{vec}(\mathbf{I})\right\}, \tag{4.23}$$

$$\mathbf{S} \triangleq (\mathbf{Q}\otimes\mathbf{Q}) \Big((\mathbf{I} + \mathbf{P}(\mathbf{A}-\mathbf{I})) \otimes (\mathbf{I} + \mathbf{P}(\mathbf{A}-\mathbf{I})) + \\ + ((\mathbf{I}-\mathbf{P}) \otimes \mathbf{P})\ \mathbf{J}\ ((\mathbf{A}-\mathbf{I}) \otimes (\mathbf{A}-\mathbf{I})) \Big). \tag{4.24}$$

Theorem 3. *Let $\mathbf{x}^{(t)}$ a state vector updated according to the probabilistic asynchronous affine update rule of Eq. (4.6). Then, a sufficient condition for convergence of the arithmetic mean, i.e., $\text{avg}\left\{\mathbf{x}^{(t)}\right\}$ to a fixed point (after the execution of multiple experiments) follows:*

- *$\rho(\mathbf{S}) \leq 1$, with the prerequisite that the eigenvalues of $\mathbf{S}$ on the unit circle are all semisimple, and one of the following criteria holds too:*
  - *(i) $\mathbf{b} \neq \mathbf{0}$ and $\rho\left(\mathbf{P}\left(\mathbf{A}-\mathbf{I}\right)+\mathbf{I}\right) < 1$.*
  - *(ii) $\mathbf{b} = \mathbf{0}$ and $\rho\left(\mathbf{P}\left(\mathbf{A}-\mathbf{I}\right)+\mathbf{I}\right) \leq 1$, where $\lambda = 1$ is the only eigenvalue of $\left(\mathbf{P}\left(\mathbf{A}-\mathbf{I}\right)+\mathbf{I}\right)$ on the unit circle, as well as semisimple.*

Proof. If the vectors are replaced with their expected values in Lemma 4 it arises that $\mathbb{E}\left[\mathbf{r}^{(t)}\right]$ converges to $\mathbf{0}$ if and only if $\mathbb{E}\left[\mathbf{x}^{(t)}\right]$ converges to a fixed point. Moreover, due to Lemma 5, the following holds:

$$\mathbb{E}\left[\mathbf{r}^{(t)}\right] \to \mathbf{0} \Leftrightarrow \text{Eq. (4.19) or Eq. (4.20) holds.} \tag{4.25}$$

Condition (4.25) ensures that $\mathbb{E}\left[\mathbf{r}^{(t)}\right]$ converges to zero but gives no information about the covariance of the residual vector. If the covariance matrix of $\mathbf{r}^{(t)}$ or even some elements of it increase as a function of t, i.e., the matrix is unbounded, then its expected value has no natural sense since residual vectors from different experiments may diverge too. This means that their arithmetic mean may behave differently than their expected value. Inversely, only if $\mathrm{Cov}\left[\mathbf{r}^{(t)}\right]$ is guaranteed to be bounded then the interpretation of $\mathbb{E}\left[\mathbf{r}^{(t)}\right]$, as the average value of $\mathbf{r}^{(t)}$, is meaningful.

Intuitively, think of the following examples:

Example 1. *Assume the scalar binary variables z_i, i.i.d. through i, which assume value equal to 10 with probability 0.2 and value of 0 with probability 0.8. Then $\prod_{i=1}^{n} z_i \to 0$ with a probability $\to 1$, while $\mathbb{E}\left[\prod_{i=1}^{n} z_i\right] \to \infty$.*

Example 2. *Assume the scalar binary variables w_n, i.i.d. through n, which assume values of n and $-n$ with equal probability. Now $|w_n| \to +\infty$ for $n \to \infty$ (and so does $\mathrm{avg}\{w_n\}$, i.e., $\mathrm{avg}\{w_n\}$ is unbounded, for finite number of experiments), while $\mathbb{E}[w_n] = 0$.*

The reason for these phenomena is the fact that the variance of either $\prod_{i=1}^{n} z_i$ or w_n is unbounded. i.e., it grows to infinity (for $n \to \infty$); thus, the expected value has no natural sense.

Example 3. *Assume a sequence that oscillates between two specific bounds; in that case, the expected value is finite and the variance exists and is bounded, but not zero.*

Therefore in order to guarantee that $\mathrm{avg}\left\{\mathbf{r}^{(t)}\right\}$ will behave the same with $\mathbb{E}\left[\mathbf{r}^{(t)}\right]$, as well as that they will converge to zero (i.e., $\lim_{t\to\infty} \mathrm{avg}\left\{\mathbf{r}^{(t)}\right\} = \lim_{t\to\infty} \mathbb{E}\left[\mathbf{r}^{(t)}\right] = \mathbf{0}$), the following bound is required:

$$\begin{aligned} &\lim_{t\to\infty} \mathbf{C}_{\mathbf{r}}^{(t)} < \infty \Leftrightarrow \lim_{t\to\infty} \left(\mathbb{E}\left[\mathbf{r}^{(t)}\left(\mathbf{r}^{(t)}\right)^{\top}\right] - \mathbb{E}\left[\mathbf{r}^{(t)}\right]\mathbb{E}\left[\mathbf{r}^{(t)}\right]^{\top}\right) < \infty \Leftrightarrow \\ &\Leftrightarrow \lim_{t\to\infty} \mathbb{E}\left[\mathbf{r}^{(t)}\left(\mathbf{r}^{(t)}\right)^{\top}\right] < \infty. \end{aligned} \tag{4.26}$$

After denoting with $\mathbf{R}^{(t)} \triangleq \mathbb{E}\left[\mathbf{r}^{(t)}\left(\mathbf{r}^{(t)}\right)^{\top}\right]$, the following recursive equation (whose proof is given in Appendix F) arises:

$$\mathrm{vec}\left(\mathbf{R}^{(t)}\right) = \mathbf{S}\,\mathrm{vec}\left(\mathbf{R}^{(t-1)}\right) = \mathbf{S}^{t}\,\mathrm{vec}\left(\mathbf{R}^{(0)}\right), \tag{4.27}$$

where $\mathbf{S}$ is given by Eq. (4.24).

Distinction between the alternative cases follows:

(a) When $\rho(\mathbf{S}) < 1$, then $\mathbf{S}^t \to \mathbf{O}$ [14, p. 618, 7.10.10] and from Eqs. (4.26), (4.27) it occurs that $\lim_{t\to\infty} \mathbf{C}_{\mathbf{r}}^{(t)} = \mathbf{O}$. Thus, the covariance between different executions of the algorithm goes to zero and every sequence of updates for $\mathbf{r}^{(t)}$ (as well as $\text{avg}\left\{\mathbf{r}^{(t)}\right\}$) converges to its expected value $\mathbb{E}\left[\mathbf{r}^{(t)}\right]$. Thus, the conditions for such convergence are $\rho(\mathbf{S}) < 1$ and Eq. (4.19) or Eq. (4.20) (for the convergence of $\mathbb{E}\left[\mathbf{r}^{(t)}\right] \to \mathbf{0}$).

(b) When $\rho(\mathbf{S}) = 1$, let's denote with $\lambda_{u(j)}$, $j \in \{1, 2, \ldots\}$ the eigenvalues of $\mathbf{S}$ with unitary magnitude, i.e., $|\lambda_{u(j)}| = 1$, and let's assume that all of them are semisimple.

- If all $\{\lambda_{u(j)}\}$ are equal to 1, then $\mathbf{S}^t$ converges to a constant matrix $\mathbf{Z}$, not necessarily zero [14, p. 630, 7.10.33].
- If there exists at least one $\lambda_{u(j)} \neq 1$, then $\mathbf{S}^t$ oscillates indefinitely between some constant matrices for $t \to \infty$ [14, p. 629, first paragraph].

However, in both subcases $\mathbf{S}^t$ is bounded by a constant matrix $\mathbf{Z}_0$. Equivalently, due to Eqs. (4.26), (4.27), $\mathbf{C}_{\mathbf{r}}^{(t)}$ is also a bounded matrix. This means that we can exploit the law of large numbers and ensure that for an efficiently large number of experiments, the arithmetic mean of the iterations $\mathbf{r}^{(t)}$ will converge to its theoretical expected value $\mathbb{E}\left[\mathbf{r}^{(t)}\right]$. On the other hand, notice that the individual iterations $\mathbf{r}^{(t)}$ may never converge, since the elements of the covariance matrix do not necessarily approach zero. Thus, the conditions for such convergence are $\rho(\mathbf{S}) = 1$, unitary-magnitude eigenvalues of $\mathbf{S}$ are all semisimple and Eq. (4.19) or Eq. (4.20) (for the convergence of $\mathbb{E}\left[\mathbf{r}^{(t)}\right] \to \mathbf{0}$).

■

Discussion

When $\rho(\mathbf{S}) = 1$, let's denote with $\lambda_{u(j)}$, $j = \{1, 2 \ldots\}$ the eigenvalues of $\mathbf{S}$ for which $|\lambda_{u(j)}| = 1$, and let's assume that at least one of them is not semisimple. Then $\mathbf{S}^t$ increases and becomes unbounded [14, p.629, first paragraph] for $t \to \infty$ and so does $\mathbf{R}^{(t)}$. When $\rho(\mathbf{S}) > 1$, $\mathbf{S}^t$ increases for $t \to \infty$ and becomes unbounded [14, Eq. (7.10.7)]; as a result, $\mathbf{R}^{(t)}$ becomes unbounded too.

After all, it is concluded that when (a) or (b) holds then the arithmetic mean of the state vector $\mathbf{x}^{(t)}$ converges to a fixed point. This happens because the expected value of the residual vector is guaranteed to converge to $\mathbf{0}$ (and the covariance of the residuals does not grow to infinity). What is more, if condition (a) holds then every individual experiment converges to a fixed point. On the other hand, when condition (b) is true then the individual experiments may not converge, but their expected value does.

Lack of necessity for the conditions of Theorem 3

Notice that the conditions stated by Theorem 3 are not necessary for convergence of $\text{avg}\left\{\mathbf{x}^{(t)}\right\}$, which is explained by the following Lemma:

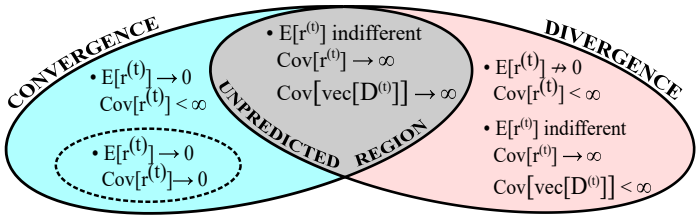


Figure 4.1: Behavior of the arithmetic mean avg $\left\{\mathbf{x}^{(t)}\right\}$ of the affine updates, after the execution of multiple experiments.

Lemma 6. *The conditions specified by Theorem 3 are sufficient but not necessary for convergence of the arithmetic mean, i.e.,* avg $\left\{\mathbf{x}^{(t)}\right\}$*, to a fixed point (after the execution of multiple experiments).*

Proof. Necessity of the conditions requires guarantee of divergence for the average value of the updates, every time these conditions are not satisfied. However, divergence of the arithmetic mean can be guaranteed only in the following sub-cases:

- If the vectors are replaced with their expected values in Lemma 4, $\mathbb{E}\left[\mathbf{r}^{(t)}\right]$ converges to $\mathbf{0}$ if and only if $\mathbb{E}\left[\mathbf{x}^{(t)}\right]$ converges to a fixed point. Consequently, $\mathbb{E}\left[\mathbf{x}^{(t)}\right]$ does not converge to a fixed point, i.e. diverges, when $\mathbb{E}\left[\mathbf{r}^{(t)}\right]$ does not converge to zero. Moreover, the covariance matrix $\mathbf{C}_{\mathbf{r}}^{(t)}$ is required to be bounded (in order for the expected value $\mathbb{E}\left[\mathbf{x}^{(t)}\right]$ to have the same behavior with avg $\left\{\mathbf{x}^{(t)}\right\}$). The former one happens when Eqs. (4.19), (4.20) are not satisfied, while the latter one requires $\mathrm{Cov}\left[\mathbf{r}^{(t)}\right] < \infty$.

- If one or more entries of the covariance matrix increase and become unbounded, i.e., $\lim_{t\to\infty} \mathbf{C}_{\mathbf{r}}^{(t)} \to_{1..} \infty$, then the arithmetic mean of the iterations will diverge too (even if $\mathbb{E}\left[\mathbf{x}^{(t)}\right]$ is predicted to converge, because the expected value in this case does not have a natural meaning, see Example 2). The aforementioned condition turns into:

$$\begin{aligned}
&\lim_{t\to\infty} \mathbf{C}_{\mathbf{r}}^{(t)} \to_{1..} \infty \Leftrightarrow \lim_{t\to\infty} \mathrm{vec}\left\{\mathbf{C}_{\mathbf{r}}^{(t)}\right\} \to_{1..} \infty \Leftrightarrow \\
&\Leftrightarrow \lim_{t\to\infty} \mathrm{vec}\left\{\mathbb{E}\left[\mathbf{r}^{(t)}\left(\mathbf{r}^{(t)}\right)^{\top}\right] - \mathbb{E}\left[\mathbf{r}^{(t)}\right]\mathbb{E}\left[\mathbf{r}^{(t)}\right]^{\top}\right\} \to_{1..} \infty \Leftrightarrow \\
&\Leftrightarrow \lim_{t\to\infty} \mathbb{E}\left[\mathrm{vec}\left\{\mathbf{r}^{(t)}\left(\mathbf{r}^{(t)}\right)^{\top} - \mathbb{E}\left[\mathbf{r}^{(t)}\right]\mathbb{E}\left[\mathbf{r}^{(t)}\right]^{\top}\right\}\right] \to_{1..} \infty \Leftrightarrow \\
&\Leftrightarrow \lim_{t\to\infty} \mathbb{E}\left[\mathrm{vec}\left\{\mathbf{D}^{(t)}\right\}\right] \to_{1..} \infty,
\end{aligned} \tag{4.28}$$

  where

$$\mathbf{D}^{(t)} = \mathbf{r}^{(t)}\left(\mathbf{r}^{(t)}\right)^{\top} - \mathbb{E}\left[\mathbf{r}^{(t)}\right]\mathbb{E}\left[\mathbf{r}^{(t)}\right]^{\top}. \tag{4.29}$$

  However it has also to be guaranteed that the expected value of Eq. (4.28) has a natural meaning. Thus, the following condition has to hold:

$$\mathrm{Cov}\left[\mathrm{vec}\left\{\mathbf{D}^{(t)}\right\}\right] < \infty \tag{4.30}$$

Note: If $\mathbb{E}\left[\mathbf{r}^{(t)}\right] \to \mathbf{0}$, then $\lim_{t\to\infty} \mathbf{C}_{\mathbf{r}}^{(t)} \to_{1..} \infty$ is equivalent with $\rho(\mathbf{S}) > 1$ (or with $\rho(\mathbf{S}) = 1$ and a unitary-magnitude eigenvalue of $\mathbf{S}$ not being semisimple).

Beyond the aforementioned cases the behaviour of the arithmetic mean cannot be predicted with certainty. As already explained, when both $\mathrm{Cov}\left[\mathbf{r}^{(t)}\right] \to_{1..} \infty$ and $\mathrm{Cov}\left[\mathrm{vec}\left\{\mathbf{D}^{(t)}\right\}\right] \to_{1..} \infty$ then the covariance matrix of the residual vector has no natural meaning and the arithmetic mean of the iterations may either diverge or converge. ■

Lemma 6 is also confirmed by the example presented in Fig. 4.3a. On that example $\mathbb{E}\left[\mathbf{r}^{(t)}\right] \to \mathbf{0}$, thus $\rho(\mathbf{S}) > 1$ indicates that $\mathbf{C}_{\mathbf{r}}^{(t)} \to_{1..} \infty$, due to Eqs. (4.26), (4.27). As a result, $\mathrm{avg}\left\{\mathbf{r}^{(t)}\right\}$ would be expected to diverge, if the conditions were both sufficient and necessary. However, it is observed that *all* 100 experiments converge which indicates that in the specific setup $\mathrm{Cov}\left[\mathbf{r}^{(t)}\right]$ has not a natural meaning.[2] Finally, a Venn diagram that summarizes the convergence/divergence regions of the arithmetic mean $\mathrm{avg}\left\{\mathbf{x}^{(t)}\right\}$ can be seen in Fig. 4.1. The case of Fig. 4.3a corresponds to the gray (middle) case of Fig. 4.1.

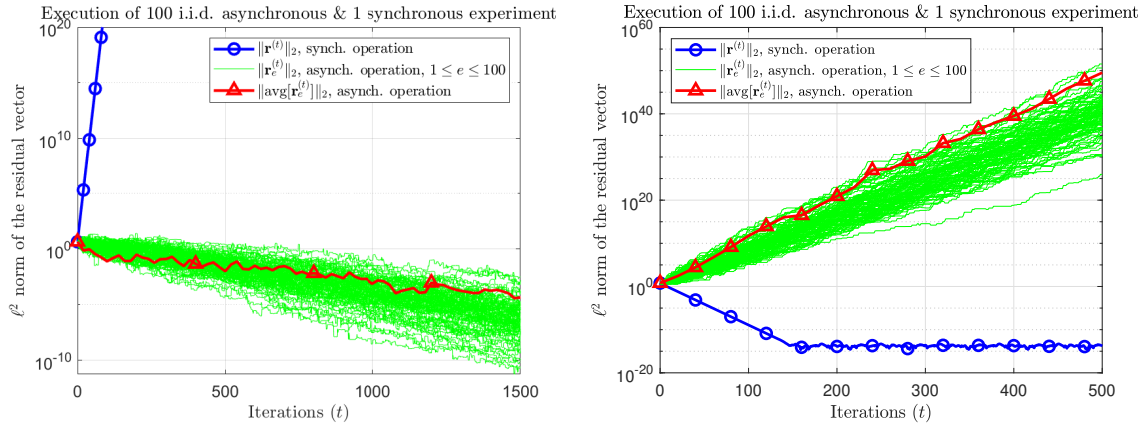
4.4 Numerical results

In this section the convergence properties of the affine setup will be tested, by implementing the (synchronous and probabilistic asynchronous) affine iterations for various matrices $\mathbf{A}$, $\mathbf{b}$.

Due to the high dimensionality of state vector $\mathbf{x}^{(t)}$, a graphical representation of each constituent term separately is not easy. For that reason the ℓ^2-norm metric will be utilized. However, the ℓ^2-norm of a vector is not a good way to ensure its convergence or divergence to a fixed point. The reason is the special case where oscillation phenomena may keep the norm constant (and give the impression of convergence), while the entries of the vector change repeatedly. This is avoided if the ℓ^2-norm of the residual vector is used instead, since in that case a zero ℓ^2-norm of the residual vector is equivalent to convergence to a fixed point. Moreover, if the vectors are replaced with their arithmetic mean in Lemma 4, it arises that $\mathrm{avg}\left\{\mathbf{r}^{(t)}\right\}$ converges to $\mathbf{0}$ if and only if $\mathrm{avg}\left\{\mathbf{x}^{(t)}\right\}$ converges to a fixed point. Thus, the affine experiments $\mathbf{x}^{(t)}$ or their arithmetic mean converge to a fixed point if and only if the residual vector $\mathbf{r}^{(t)}$ or its arithmetic mean (respectively) converges to zero, which happens if and only if its ℓ^2-norm converges to zero.

At every figure, the x axis represents the number of iterations t, while the y axis represents the ℓ^2-norm of the residual vector $\mathbf{r}^{(t)}$. Also, the thick blue (with circles) line represents the synchronous execution of the affine iterations, while the thick red (with triangles) line represents the arithmetic mean of the probabilistic asynchronous experiments. Finally, the thin green lines represent the individual asynchronous experiments.

[2]The reason is that both $\mathrm{Cov}\left[\mathbf{r}^{(t)}\right]$ and $\mathrm{Cov}\left[\mathrm{vec}\left\{\mathbf{D}^{(t)}\right\}\right]$ are unbounded in this setup.



(a) Convergence at the probabilistic asynchronous operation and divergence at the synchronous case.

(b) Convergence at the synchronous case and divergence at the probabilistic asynchronous operation.

Figure 4.2: Asymptotic behavior on synchronous vs asynchronous setups.

In Fig. 4.2a the execution of affine iterations is shown for

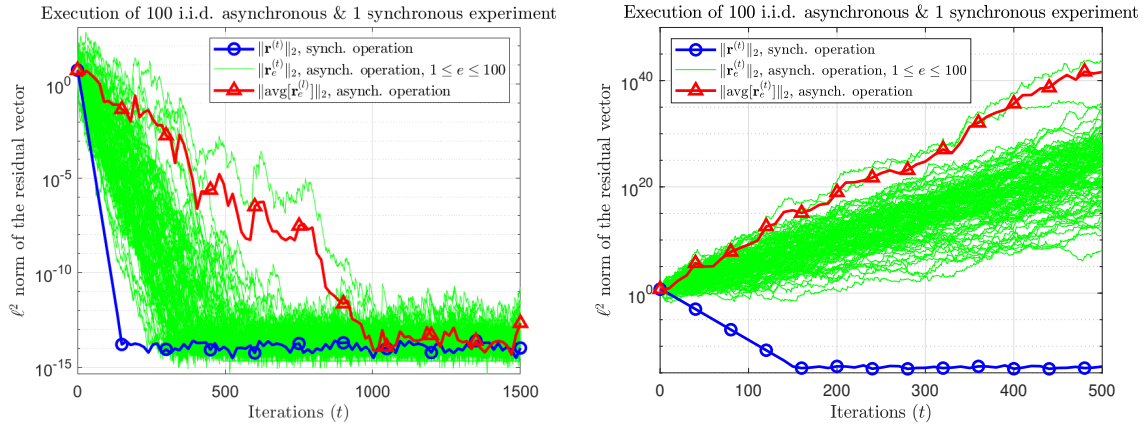
$$\mathbf{A} = \begin{bmatrix} -0.9900, & -0.0829, & 0.0699, & -0.0806 \\ -10.9498, & -0.5065, & -0.0278, & 0.7711 \\ -0.6192, & 0.1912, & -0.6085, & 0.2683 \\ -0.8535, & 0.2871, & -0.2580, & -0.9686 \end{bmatrix}, \tag{4.31}$$
$$\mathbf{b} = \begin{bmatrix} 0.6324, & 0.0975, & 0.2785, & 0.5469 \end{bmatrix}^\top, \quad \mathbf{P} = 0.15\mathbf{I}.$$

Then $\rho(\mathbf{A}) = 1.7$, while $\rho(\mathbf{PA} + \mathbf{I} - \mathbf{P}) = 0.89$ and $\rho(\mathbf{S}) = 0.99$. Notice that all 100 asynchronous experiments converge as expected, as well as their arithmetic mean. On the contrary the synchronous iterations diverge, since $\rho(\mathbf{A}) > 1$. This is a characteristic example of the utility of the asynchronous operation, since it can reach to a fixed point even when the synchronous case diverges.

On the other hand, Fig. 4.2b depicts the affine updates for

$$\mathbf{A} = \begin{bmatrix} 2.8130 & -1.7270 & -0.5310 & 5.2365 \\ -0.0625 & -0.3045 & 0.0785 & -0.4692 \\ 1.2487 & -1.0999 & -0.9897 & 3.2804 \\ -1.4574 & 0.7778 & 0.2362 & -2.8254 \end{bmatrix}, \tag{4.32}$$
$$\mathbf{b} = \begin{bmatrix} 0.3188, & -1.3077, & -0.4336, & 0.3426 \end{bmatrix}^\top, \quad \mathbf{P} = 0.1\mathbf{I}.$$

At this case $\rho(\mathbf{A}) = 0.8$, while $\rho(\mathbf{PA} + \mathbf{I} - \mathbf{P}) = 0.94$ and $\rho(\mathbf{S}) = 2.3$. As a result, the synchronous experiment converges, since $\rho(\mathbf{A}) < 1$, while all 100 asynchronous experiments diverge. This example is the opposite of Fig. 4.2a and proves the utility of the synchronous operation, which may reach to a fixed point when the probabilistic asynchronous fails to do so.



(a) Convergence at the probabilistic asynchronous case, while $\mathbf{P} = 0.95\mathbf{I}$.

(b) Divergence at the probabilistic asynchronous case, while $\mathbf{P} = 0.85\mathbf{I}$.

Figure 4.3: Experiments while $\rho(\mathbf{S}) > 1$, where the asymptotic behavior at the probabilistic asynchronous case changes with a slight change on the update probabilities.

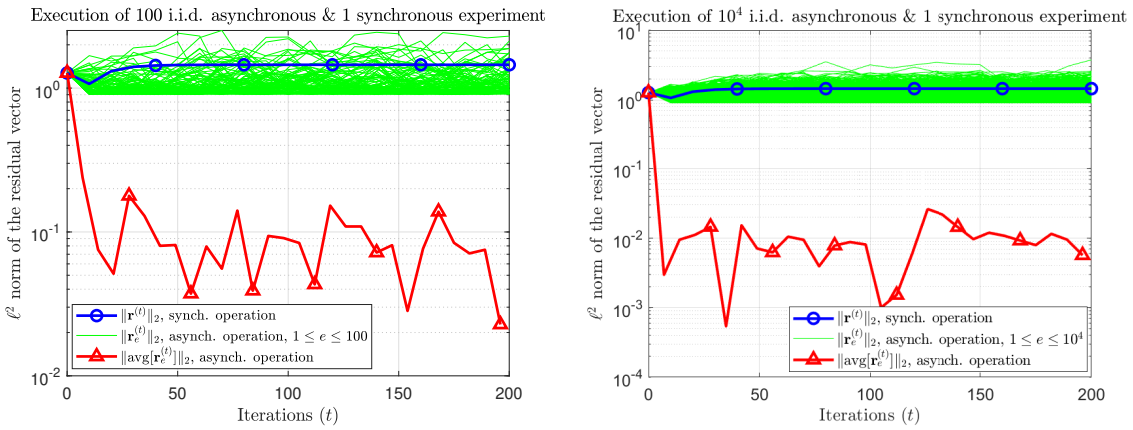
Fig. 4.3a shows the execution of AFP iterations for

$$\mathbf{A} = \begin{bmatrix} 2.8130, & -1.7270, & -0.5310, & 5.2365 \\ -0.0625, & -0.3045, & 0.0785, & -0.4692 \\ 1.2487, & -1.0999, & -0.9897, & 3.2804 \\ -1.4574, & 0.7778, & 0.2362, & -2.8254 \end{bmatrix}, \tag{4.33}$$

$$\mathbf{b} = \begin{bmatrix} 0.6324, & 0.0975, & 0.2785, & 0.5469 \end{bmatrix}^\top, \quad \mathbf{P} = 0.95\mathbf{I}.$$

In this example, the synchronous iterations converge as expected, since $\rho(\mathbf{A}) = 0.8$. Moreover, all 100 asynchronous executions of the algorithm also converge, as well as their arithmetic mean. This example is located inside the unpredicted region of Fig. 4.1 because $\rho(\mathbf{PA} + \mathbf{I} - \mathbf{P}) = 0.71$ and $\rho(\mathbf{S}) = 1.54 > 1$, which indicate that $\lim_{t\to\infty} \mathbf{E}\left[\mathbf{r}^{(t)}\right] \to \mathbf{0}$ and $\lim_{t\to\infty} \mathbf{C}_{\mathbf{r}}^{(t)} \to_{1..} \infty$, respectively.

Fig. 4.3b demonstrates the AFP iterations for the same parameters with Fig. 4.3a except from matrix $\mathbf{P}$ which is set equal with $0.85\mathbf{I}$. Then $\rho(\mathbf{A}) = 0.8$, so the synchronous case is expected to converge, something which is verified from the plot. However, the slight change on the update probability compared to the example presented in Fig. 4.3a, makes a significant change on the asymptotic properties, and now all probabilistic asynchronous updates diverge, as well as their arithmetic mean. On this example $\rho(\mathbf{PA} + \mathbf{I} - \mathbf{P}) = 0.53$ and $\rho(\mathbf{S}) = 2.87$. This is a setup where the expected value of the experiments cannot be used, even though it converges to a fixed point, since it has no natural meaning and their arithmetic mean diverges to infinity.



(a) Execution of 100 i.i.d. experiments. (b) Execution of 10^4 i.i.d. experiments.

Figure 4.4: Example where $\rho(\mathbf{S}) = 1$ and convergence of the arithmetic mean is achieved at the probabilistic asynchronous case, despite the fact that the individual asynchronous experiments do not converge. While the number of experiments increases, the arithmetic mean approaches closer to the fixed point.

Figs 4.4a, 4.4b demonstrate the affine iterations for

$$\mathbf{A} = \begin{bmatrix} -0.3230, & -0.0829, & 0.0699, & -0.0806 \\ 0, & -1, & 0, & 0 \\ -0.6192, & 0.1912, & -0.97, & 0.8683 \\ 0, & 0, & 0, & 1 \end{bmatrix}, \tag{4.34}$$
$$\mathbf{b} = \begin{bmatrix} 0, & 0, & 0, & 0 \end{bmatrix}^\top, \quad \mathbf{P} = 0.5\mathbf{I}.$$

In this case, $\rho(\mathbf{A}) = 1$, but one of the eigenvalues of $\mathbf{A}$ equals with -1. Therefore the synchronous iterations of $\mathbf{x}^{(t)}$ do not converge. This is verified by both figures, since the ℓ^2-norm of $\mathbf{r}^{(t)}$ does not converge to zero, in the synchronous case. In the asynchronous operation it is also true that $\rho(\mathbf{PA}+\mathbf{I}-\mathbf{P}) = 1$, but with the difference that all eigenvalues with unitary magnitude are equal to 1 and semisimple. Moreover $\rho(\mathbf{S}) = 1$ with all the unitary eigenvalues of $\mathbf{S}$ also being semisimple, which implies a bounded (but not zero) covariance matrix. Notice that this is the case where the individual vectors $\mathbf{r}^{(t)}$ do not converge to zero, but their arithmetic mean does. Naturally while the number of experiments grows, the arithmetic mean $\text{avg}\left\{\mathbf{r}^{(t)}\right\}$ approaches closer to the expected value $\mathbb{E}\left[\mathbf{r}^{(t)}\right] = \mathbf{0}$. This is also demonstrated by the results, with Fig. 4.4a showing the arithmetic mean of 100 experiments and Fig. 4.4b plotting the arithmetic mean of 10^4 experiments. As expected, the latter one is significantly closer to zero.

Finally, Fig. 4.5 demonstrates a case where neither the synchronous nor the probabilistic asynchronous AFP iterations (or their arithmetic mean) converge, because $\text{avg}\left\{\mathbf{r}^{(t)}\right\}$ converges to a fixed point different than zero. Specifically, the following

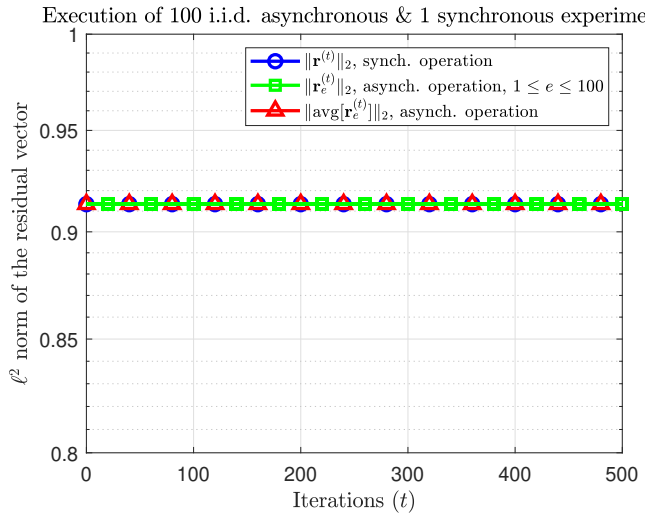


Figure 4.5: Divergence at both the asynchronous and the synchronous case, because the eigenvalues with unitary magnitude are not equal to 1.

matrices were utilized:

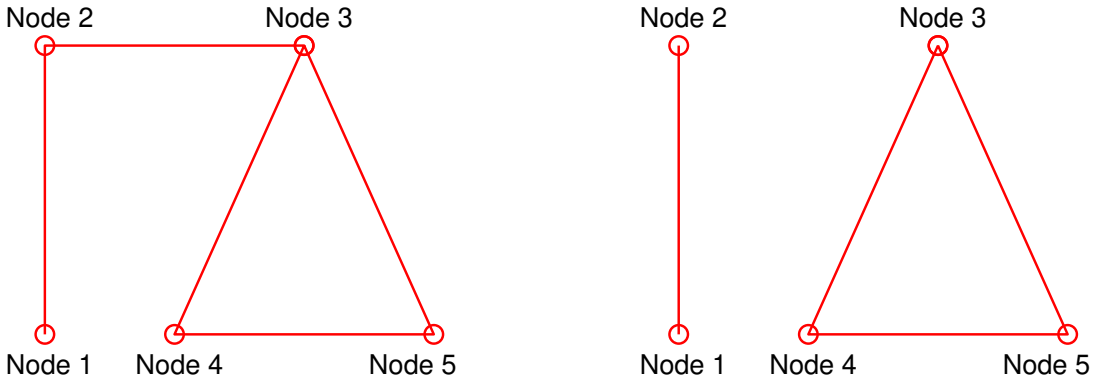
$$\mathbf{A} = \begin{bmatrix} 1, & 0, & 0, & 0 \\ 0, & 1, & 0, & 4 \\ 0, & 0, & 1, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 0, & 0, & 0, & 0 \end{bmatrix}^\top, \quad \mathbf{P} = 0.5\mathbf{I}. \tag{4.35}$$

In the synchronous case $\rho(\mathbf{A}) = 1$ but the eigenvalues of $\mathbf{A}$ with unitary magnitude are not semisimple, which explains the lack of convergence for the updates. On the asynchronous case $\rho(\mathbf{PA} + \mathbf{I} - \mathbf{P}) = 1$ but the unitary eigenvalues of $\mathbf{PA} + \mathbf{I} - \mathbf{P}$ are not semisimple either.

Chapter 5

Deployment: Embedded Distributed Averaging System

Embedded Distributed Averaging System (EDAS) has been designed to estimate the average of sensor measurements, collected by its components (nodes), which are spread in a large space (e.g., a block of flats). These nodes are capable of exchanging wireless messages, and they implement the Average Consensus algorithm to estimate the arithmetic mean of their values. However, for a correct estimation to be achieved, the communication graph (i.e., the graph with an edge between every pair of nodes which exchange messages, based on the geographical locations) has to be connected [20], i.e., a path has to exist from any node to any other node in the graph (an example can be found in Fig. 5.1).



(a) Connected graph.

(b) Disconnected graph.

Figure 5.1: Connected vs disconnected graph.

5.1 Hardware layer

As nodes of the system, Thunderboard Sense 2 EFR32MG24 devices [41] were selected. These devices cost about 20$, come with a 32-bit architecture and stand out from other options due to their wide range of sensors, low power consumption and powerful radio (up to 10.4 dBm) with multi-protocol capability, which make them ideal for embedded applications. A detailed schematic can be seen in Fig. 5.2.

The devices operate at a 3.3V voltage, supplied by either a coin CR2032 battery, or an external power supply. Moreover, the necessary power can be provided via the Mini Simplicity Connector, which is mostly used for debugging purposes. Finally,

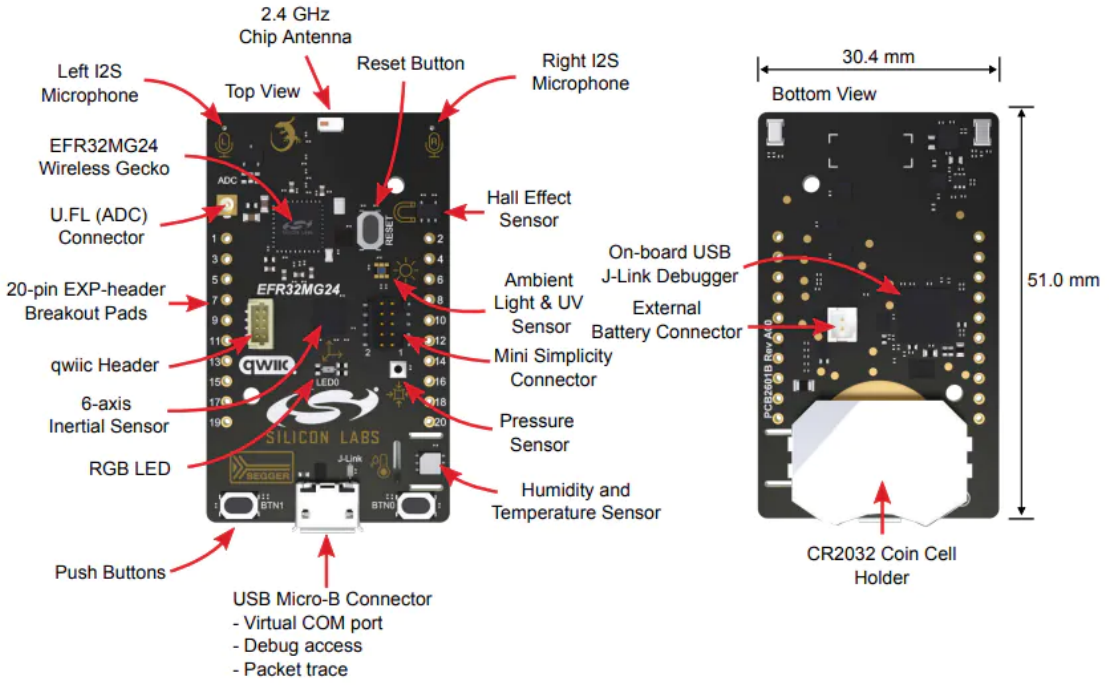


Figure 5.2: Schematic of a Thunderboard Sense 2 EFR32MG24 device.

there exists a Micro-USB port which can also power the devices (after the 5V voltage has been decreased to 3.3V, by an on-board regulator).

The power consumption of each Thunderboard device depends on its task. While it executes Average Consensus, its CPU is fully operational, as well as all required peripherals. At this state the electric current is about $30-35$ mA, which is equivalent to 100 mW of power consumption. For the rest of the time, the devices are on standby mode, waiting for the user to initiate a new execution of the algorithm (or read the last estimated average). As a result, the CPU is disabled but some of the other components (e.g., the antenna in receive-only mode and the high-frequency clock oscillator) are operational. The electric current at this state is about $5-10$ mA, which is equivalent to 25 mW of power.

Currently, EDAS estimates the average temperature of the environment. The temperature readings are obtained from the embedded Si7021 [42] relative humidity-temperature sensor, which utilizes a capacitive humidity sensing element and an energy-gap temperature sensor. These sensors provide precise and linear readings across a wide range of humidity and temperature values. The Si7021 communicates with the Thunderboard Sense 2 microcontroller via an I^2C interface, and it has a low power consumption, making it ideal for battery-powered devices. Additionally, the sensor has a fast response time, allowing for real-time monitoring of temperature and humidity changes. The Si7021 can measure temperature with an accuracy of ± 0.4∘C in a range of -10°C to 85°C. However its accuracy may decrease due the dissipated power on the board, and it is recommended to power the Thunderboard devices via a battery or the Mini Simplicity connector instead of a USB cable, to limit the effect

of influencing the readings of Si7021.

Notice that EDAS is not restricted to averaging only temperatures, but its operation can be expanded to any physical magnitude. The only change required is the establishment of a link with the respective sensors in order to retrieve their readings.

Finally, Simplicity Studio software [43] can be utilized for programming and flashing the nodes, in combination with GNU ARM compiler and Gecko Software Development Kit (SDK) [44].

5.2 Network layer

Network configuration of EDAS has been built upon a proprietary protocol, developed with the Radio Abstraction Interface Layer (RAIL) component of Gecko SDK [44]. Wireless communication between Thunderboard devices was implemented in the frequency domain, whereby each node listens only to its own dedicated channel (i.e., frequency) and transmits messages to different channels, depending on the destination node.

Binary Frequency Shift Keying (B-FSK) [45], [46] is a widely used modulation technique in digital communication, radio broadcasting, and remote control systems, due to its simplicity and reliability. The process of transmitting signals using B-FSK entails converting the input binary data into a sequence of tones, which are then modulated onto the carrier signal. As a result, the carrier signal alternates between two different frequencies, depending on whether the input signal is a binary 0 or 1.

The B-FSK receivers employed in this project are non-coherent, meaning that they do not require knowledge of the phase of the received signal. Moreover, the 2.4 GHz band is utilized, ranging from 2401 MHz to 2480 MHz. Despite the existence of many unlicensed frequency bands that could have been utilized, a 2.4-GHz system has the advantage of a longer range (for a line-of-sight wireless link) and typically requires a small antenna, enabling the construction of the network using only the embedded antennas on the Thunderboard devices. Also, a bit rate of 2.4 KBits/sec is set, resulting in a minimum distance of $Df = 1.2$ KHz between subcarriers (with a deviation of 0.6 KHz). Finally the payload of the exchanged messages is set to 16 bytes, which is more than enough for the needs of the EDAS system.

Notice that the Thunderboard devices used for this network are half-duplex, i.e., they cannot transmit and receive data at the same time. Consequently, a Media Access Control (MAC) protocol has to rule their behavior and ensure that concurrent transmission and reception operations will be never needed. EDAS implements a simplified version of the token ring protocol [47], where a token is passed from node to node in a circular order (according to the edges of the communication graph). Each node is granted permission to transmit only when it holds the token, which ensures that no more than one node can transmit at the same time. However, this also means that a sufficient power source is necessary for all nodes to prevent the loss of the token and the iterative restarting of the system. In addition, depending on the structure of the system's communication graph, some nodes may receive the token more than once before it completes a full loop. In such cases, the token is released without any

action taken by these nodes to ensure synchronous execution of the algorithm, where each node performs exactly one operation per full loop of the token.

5.3 Software layer

On the software layer, all nodes operate in standby mode with deactivated CPUs until a user connects via the serial port. Once the user connects, he may initiate the retrieval of measurements from the sensors and execution of the Average Consensus algorithm (see section 2.1). Then, through the first loop of the token (see section 5.2), all nodes wake up from the standby mode and read the temperature from their embedded Si7021 sensors.

After this, the iterations of Average Consensus are executed through the following loops of the token, until all nodes converge close enough to the arithmetic mean; convergence is determined by a constant threshold which is compared to the absolute difference of a node's 2 consecutive states. The moving token is crucial for the synchronization of the nodes and ensures concurrent start and ending of the algorithm's iterations, through the whole distributed network; thus, it is ensured that the synchronous version of Average Consensus is run. Moreover, it ensures that all nodes have access to some necessary information (such as when the termination criterion is satisfied, or when a restart is required due to a timeout), regardless of whether they are connected via an edge or not.

Furthermore, the design takes into account potential challenges that can arise during algorithm execution. For instance, the token may be lost due to packet losses, resulting in nodes being stuck in the execution loop. To mitigate this issue, the software layer incorporates a timeout mechanism that restarts the algorithm if the token does not move with at least a minimum a predefined speed across the nodes.

Finally, when all nodes simultaneously meet the convergence criterion, the token performs a last loop to place them back into standby mode. Subsequently, the estimated average temperature is returned to the user through the serial port of the node he was connected to.

The execution time depends on the shape of the communication graph, the size of the network, the efficiency of wireless communication, the initial temperature measurements, and the termination threshold of the algorithm. For a system with 6 nodes, the graph of Fig. 5.3, insignificant packet losses and a termination threshold equal to 0.1, about 10 to 20 iterations have to be run which last about 15 - 25 seconds.

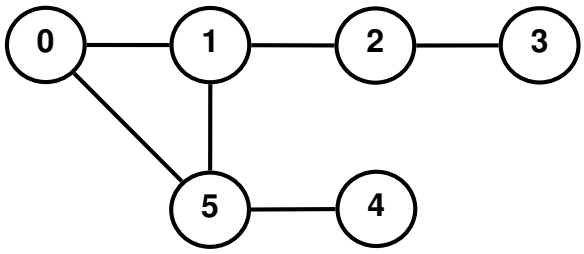


Figure 5.3: Example of a communication graph.

Chapter 6

Conclusions

This work provides an overview of the most popular inference algorithms that can be run by a distributed sensor network, and are equivalent to the execution of affine iterations. Specifically, the algorithms covered in this work include average consensus, Gaussian belief propagation, and spectral clustering. Furthermore, multimodal inference on sensor networks is discussed, with a particular focus on two variations of linear regression. These variations are equivalent to the execution of affine iterations and combine information from multiple sources of data. This way, the accuracy and reliability of data analysis can be improved.

In addition to discussing the algorithms themselves, this work also analyzes the asymptotic behavior of the affine model under both synchronous and asynchronous operation. Special emphasis is placed (and contributions are made, compared to the existing literature) on the probabilistic asynchronous framework, which is highly powerful and can be applied in a wide variety of implementations. More specifically, this work provides sufficient conditions for convergence and explains why the arithmetic mean is more appropriate than the expected mean for discovering fixed points. The reason is that the arithmetic mean is the only quantity that can be estimated experimentally when the asymptotic behaviours of these two quantities are different. Notice that cases exist where the asymptotic behaviour of the arithmetic mean is unpredictable. Finally, there are models where the arithmetic mean offers practical solutions with finite number of experiments, while the individual experiments never converge to a solution. Such findings are important when inference is run over distributed platforms, as in wireless sensor networks.

Finally, an implementation of a low-cost sensor network is presented. This distributed system executes the average consensus algorithm and estimates the average of the temperatures measured by its sensors.

Future work may expand the current study to non-linear models. Moreover the i.i.d. constant $\mathbf{P}$ assumption could be dismissed, giving emphasis to update probabilities that change and are correlated over time. Finally, further distinction of the unpredicted region into convergent and divergent models requires more research.

Appendix A

Maximum Likelihood Estimator of a Gaussian PDF

Problem: Given the Gaussian distribution $\mathcal{N}\left(\bar{\mathbf{y}}; \Phi(\bar{\mathbf{X}})\mathbf{w}, \sigma^2\mathbf{I}\right)$ of Eq. (3.2), compute the Maximum Likelihood estimator of $\mathbf{w}$, i.e.,

$$\mathbf{w}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p_{\bar{\mathbf{y}}|\mathbf{w}}\left(\bar{\mathbf{y}}|\mathbf{w}\right). \tag{A.1}$$

Solution: The pdf of the aforementioned distribution is:

$$p_{\bar{\mathbf{y}}|\mathbf{w}}\left(\bar{\mathbf{y}}|\mathbf{w}\right) \propto_{\mathbf{w}} \exp\left(-\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}} - \Phi(\bar{\mathbf{X}})\mathbf{w}\right)^{\top}\left(\bar{\mathbf{y}} - \Phi(\bar{\mathbf{X}})\mathbf{w}\right)\right) \tag{A.2}$$

Let's set $\mathbf{Z} = \Phi(\bar{\mathbf{X}})$. Also,

$$L = (\bar{\mathbf{y}} - \mathbf{Z}\mathbf{w})^{\top}(\mathbf{y} - \mathbf{Z}\mathbf{w}) \overset{\bar{\mathbf{y}}^{\top}\mathbf{Z}\mathbf{w} = \mathbf{w}^{\top}\mathbf{Z}^{\top}\bar{\mathbf{y}}}{=\!=\!=\!=\!=\!=\!=\!=} \bar{\mathbf{y}}^{\top}\bar{\mathbf{y}} - 2\bar{\mathbf{y}}^{\top}\mathbf{Z}\mathbf{w} + \mathbf{w}^{\top}\mathbf{Z}^{\top}\mathbf{Z}\mathbf{w}. \tag{A.3}$$

Maximization of $p_{\bar{\mathbf{y}}|\mathbf{w}}\left(\bar{\mathbf{y}}|\mathbf{w}\right)$ is equivalent to the minimization of L. The extremas of L (with respect to $\mathbf{w}$) can be found by setting $\nabla_{\mathbf{w}}L = 0$:

$$\nabla_{\mathbf{w}}L = 0 \Leftrightarrow -\mathbf{Z}^{\top}\bar{\mathbf{y}} + \mathbf{Z}^{\top}\mathbf{Z}\mathbf{w} = 0 \Leftrightarrow \mathbf{Z}^{\top}\mathbf{Z}\mathbf{w} = \mathbf{Z}^{\top}\bar{\mathbf{y}} \Leftrightarrow \mathbf{w} = (\mathbf{Z}^{\top}\mathbf{Z})^{-1}\mathbf{Z}^{\top}\bar{\mathbf{y}} \tag{A.4}$$

To ensure that this extrema is a minimum, the Hessian matrix ($\nabla^2_{\mathbf{w}}L$) has to be computed:

$$\nabla^2_{\mathbf{w}}L = \nabla_{\mathbf{w}}(-\mathbf{Z}^{\top}\bar{\mathbf{y}} + \mathbf{Z}^{\top}\mathbf{Z}\mathbf{w}) = \mathbf{Z}^{\top}\mathbf{Z} \tag{A.5}$$

which is positive semi-definite. Thus, $\mathbf{w} = (\mathbf{Z}^{\top}\mathbf{Z})^{-1}\mathbf{Z}^{\top}\bar{\mathbf{y}}$ is a global minimum of L and, if $\mathbf{Z}$ is replaced with its initial value it occurs that:

$$\mathbf{w}_{ML} = \left(\Phi(\bar{\mathbf{X}})^{\top}\Phi(\bar{\mathbf{X}})\right)^{-1}\Phi(\bar{\mathbf{X}})^{\top}\bar{\mathbf{y}} \tag{A.6}$$

Note: $\mathbf{Z}^{\top}\mathbf{Z}$ is invertible if and only if the columns of $\mathbf{Z}$ are linearly independent. This condition should be satisfied by any chosen function ϕ.

Appendix B

Proof of Eq. (3.5)

Let's denote with $\mathbf{Z}_* = \Phi(\bar{\mathbf{X}}_*)$ and $\mathbf{Z} = \Phi(\bar{\mathbf{X}})$. Then,

$$
\begin{aligned}
&\int_{-\infty}^{+\infty} p_{\bar{\mathbf{y}}_*|\mathbf{w}}\left(\bar{\mathbf{y}}_*|\mathbf{w}\right) p_{\bar{\mathbf{y}}|\mathbf{w}}\left(\bar{\mathbf{y}}|\mathbf{w}\right) \mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*} \\
\propto_{\bar{\mathbf{y}}_*} &\int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}}_* - \Phi(\bar{\mathbf{X}}_*)\mathbf{w}\right)^\top\left(\bar{\mathbf{y}}_* - \Phi(\bar{\mathbf{X}}_*)\mathbf{w}\right)\right) \\
&\qquad \exp\left(-\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}} - \Phi(\bar{\mathbf{X}})\mathbf{w}\right)^\top\left(\bar{\mathbf{y}} - \Phi(\bar{\mathbf{X}})\mathbf{w}\right)\right)\mathrm{d}\mathbf{w} = \\
= &\int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}}_* - \mathbf{Z}_*\mathbf{w}\right)^\top\left(\bar{\mathbf{y}}_* - \mathbf{Z}_*\mathbf{w}\right) + \left(\bar{\mathbf{y}} - \mathbf{Z}\mathbf{w}\right)^\top\left(\bar{\mathbf{y}} - \mathbf{Z}\mathbf{w}\right)\right)\mathrm{d}\mathbf{w} = \\
= &\int_{-\infty}^{+\infty} \exp\Bigg(-\frac{1}{2\sigma^2}\Big(\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_* - 2\mathbf{w}^\top\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \mathbf{w}^\top\mathbf{Z}_*^\top\mathbf{Z}_*\mathbf{w} + \\
&\qquad + \bar{\mathbf{y}}^\top\bar{\mathbf{y}} - 2\mathbf{w}^\top\mathbf{Z}^\top\bar{\mathbf{y}} + \mathbf{w}^\top\mathbf{Z}^\top\mathbf{Z}\mathbf{w}\Big)\Bigg)\mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*} \\
\propto_{\bar{\mathbf{y}}_*} &\int_{-\infty}^{+\infty} \exp\Bigg(-\frac{1}{2\sigma^2}\Big(\mathbf{w}^\top(\mathbf{Z}_*^\top\mathbf{Z}_* + \mathbf{Z}^\top\mathbf{Z})\mathbf{w} - \\
&\qquad - 2\mathbf{w}^\top(\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \mathbf{Z}^\top\bar{\mathbf{y}}) + \bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\Big)\Bigg)\mathrm{d}\mathbf{w}.
\end{aligned}
\tag{B.1}
$$

Assume $\mathbf{L} = \mathbf{Z}_*^\top\mathbf{Z}_* + \mathbf{Z}^\top\mathbf{Z}$. If the Sherman–Morrison–Woodbury identity [14, p. 124, 3.8.3] is utilized, the following arises:

$$
\begin{aligned}
\mathbf{L}^{-1} &= (\mathbf{Z}_*^\top\mathbf{Z}_* + \mathbf{Z}^\top\mathbf{Z})^{-1} \\
&= (\mathbf{Z}^\top\mathbf{Z})^{-1} - (\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}_*^\top\left(\mathbf{I} + \mathbf{Z}_*(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}_*^\top\right)^{-1}\mathbf{Z}_*(\mathbf{Z}^\top\mathbf{Z})^{-1}.
\end{aligned}
\tag{B.2}
$$

Existence of $(\mathbf{Z}^\top\mathbf{Z})^{-1}$ is guaranteed, as discussed in appendix A. A quadratic form such as $\mathbf{Z}_*(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}_*^\top$ is positive semi-definite, which makes $\mathbf{I}+\mathbf{Z}_*(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}_*^\top$ strictly positive definite (thus invertible). As a result, $\mathbf{L}^{-1}$ exists and $\mathbf{L}$ is an invertible matrix.

Furthermore, assume $\mathbf{m} = \mathbf{L}^{-1}(\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \mathbf{Z}^\top\bar{\mathbf{y}})$. Then,

$$
\begin{aligned}
\text{(B.1)} \propto_{\bar{\mathbf{y}}_*} &\int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{w}^\top\mathbf{L}\mathbf{w} - 2\mathbf{w}^\top\mathbf{L}\mathbf{m} + \bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\right)\right)\mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*} \\
\propto_{\bar{\mathbf{y}}_*} &\int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2\sigma^2}\left((\mathbf{w}-\mathbf{m})^\top\mathbf{L}(\mathbf{w}-\mathbf{m}) + \bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_* - \mathbf{m}^\top\mathbf{L}\mathbf{m}\right)\right)\mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*}
\end{aligned}
$$

$$\propto_{\bar{\mathbf{y}}_*} \exp\left(\frac{1}{2\sigma^2}\mathbf{m}^\top\mathbf{L}\mathbf{m} - \frac{1}{2\sigma^2}\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\right)\int_{-\infty}^{+\infty}\exp\left(-\frac{1}{2\sigma^2}(\mathbf{w}-\mathbf{m})^\top\mathbf{L}(\mathbf{w}-\mathbf{m})\right)\mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*}$$

$$\propto_{\bar{\mathbf{y}}_*} \exp\left(\frac{1}{2\sigma^2}\left(\mathbf{m}^\top\mathbf{L}\mathbf{m} - \bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\right)\right) =$$

$$= \exp\left(\frac{1}{2\sigma^2}\left(\left(\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \mathbf{Z}^\top\bar{\mathbf{y}}\right)^\top\left(\mathbf{L}^{-1}\right)^\top\mathbf{L}\mathbf{L}^{-1}\left(\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \mathbf{Z}^\top\bar{\mathbf{y}}\right) - \bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\right)\right) \overset{\mathbf{L}=\mathbf{L}^\top\Rightarrow\mathbf{L}^{-1}=\left(\mathbf{L}^{-1}\right)^\top}{=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=}$$

$$= \exp\left(\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}}_*^\top\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + 2\bar{\mathbf{y}}_*^\top\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}^\top\bar{\mathbf{y}} + \bar{\mathbf{y}}^\top\mathbf{Z}\mathbf{L}^{-1}\mathbf{Z}^\top\bar{\mathbf{y}} - \bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\right)\right) \propto_{\mathbf{y}_*}$$

$$\propto_{\mathbf{y}_*} \exp\left(\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}}_*^\top\left(\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}_*^\top - \mathbf{I}\right)\bar{\mathbf{y}}_* + 2\bar{\mathbf{y}}_*^\top\left(\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}^\top\bar{\mathbf{y}}\right)\right)\right). \tag{B.3}$$

Moreover, the definition of matrix $\mathbf{P}$ follows:

$$\mathbf{P} \triangleq \left(\frac{1}{\sigma^2}\left(\mathbf{I} - \mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}_*^\top\right)\right)^{-1} = \sigma^2\left(\mathbf{I} - \mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}_*^\top\right)^{-1} \overset{\text{Eq. (B.2)}}{=\!=\!=\!=}$$

$$= \sigma^2\left(\mathbf{I} - \mathbf{Z}_*\left(\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1} - \left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}_*^\top\left(\mathbf{I} + \mathbf{Z}_*(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}_*^\top\right)^{-1}\mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\right)\mathbf{Z}_*^\top\right)^{-1} =$$

$$\overset{\mathbf{K}=\mathbf{Z}_*(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}_*^\top}{=\!=\!=\!=\!=\!=\!=\!=\!=\!=} \sigma^2\left(\mathbf{I} - \mathbf{K} + \mathbf{K}(\mathbf{I}+\mathbf{K})^{-1}\mathbf{K}\right)^{-1} =$$

$$= \sigma^2\left(\mathbf{I} - (\mathbf{I}+\mathbf{K})^{-1}\mathbf{K}\right)^{-1} = \sigma^2\left((\mathbf{I}+\mathbf{K})^{-1}\right)^{-1} = \sigma^2(\mathbf{I}+\mathbf{K}). \tag{B.4}$$

By substituting the original values for $\mathbf{K}$, $\mathbf{Z}$, $\mathbf{Z}_*$, it arises that

$$\mathbf{P} = \sigma^2\mathbf{I} + \sigma^2\Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}})\right)^{-1}\Phi(\bar{\mathbf{X}}_*)^\top. \tag{B.5}$$

Also, assume $\mathbf{t} \triangleq \frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}^\top\bar{\mathbf{y}} \overset{(*)}{=} \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}^\top\bar{\mathbf{y}}$. Proof of the equality at point $(*)$ follows:

$$\frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}^\top\bar{\mathbf{y}} = \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}^\top\bar{\mathbf{y}} \Leftrightarrow$$

$$\frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_*\mathbf{L}^{-1} = \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1} \Leftrightarrow \frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_* = \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{L} \Leftrightarrow$$

$$\frac{1}{\sigma^2}\sigma^2\left(\mathbf{I} + \mathbf{Z}_*(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}_*^\top\right)\mathbf{Z}_* = \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\left(\mathbf{Z}_*^\top\mathbf{Z}_* + \mathbf{Z}^\top\mathbf{Z}\right) \Leftrightarrow$$

$$\mathbf{Z}_* + \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}_*^\top\mathbf{Z}_* = \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}_*^\top\mathbf{Z}_* + \mathbf{Z}_*. \tag{B.6}$$

Eq. (B.6) is valid, thus the initial statement is valid too. ■

Finally, Eq. (B.3) turns into:

$$\text{(B.3)} = \exp\left(-\frac{1}{2}\left(\bar{\mathbf{y}}_*^\top\bar{\mathbf{P}}\bar{\mathbf{y}}_* - 2\bar{\mathbf{y}}_*^\top\bar{\mathbf{P}}\mathbf{t}\right)\right) \propto_{\bar{\mathbf{y}}_*} \exp\left(-\frac{1}{2}\left(\bar{\mathbf{y}}_* - \mathbf{t}\right)^\top\bar{\mathbf{P}}(\bar{\mathbf{y}}_* - \mathbf{t})\right)\right)$$

$$\propto_{\bar{\mathbf{y}}_*} \exp\left(-\frac{1}{2}\left(\bar{\mathbf{y}}_* - \mathbf{t}\right)^\top\mathbf{P}^{-1}(\bar{\mathbf{y}}_* - \mathbf{t})\right)\right). \tag{B.7}$$

■

Appendix C

Proof of Eq. (3.13)

Let's denote with $\mathbf{Z}_* = \Phi(\bar{\mathbf{X}}_*)$ and $\mathbf{Z} = \Phi(\bar{\mathbf{X}})$. Then,

$$\begin{aligned}
&p_{\bar{\mathbf{y}}|\mathbf{w}}(\bar{\mathbf{y}}|\mathbf{w})p_{\mathbf{w}}(\mathbf{w}) \propto_{\mathbf{w}} \\
\propto_{\mathbf{w}} &\exp\left(-\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}} - \mathbf{Z}\mathbf{w}\right)^\top \left(\bar{\mathbf{y}} - \mathbf{Z}\mathbf{w}\right) - \frac{1}{2\sigma_{\mathbf{w}}^2}\mathbf{w}^\top\mathbf{w}\right) = \\
= &\exp\left(-\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}}^\top\bar{\mathbf{y}} - 2\mathbf{w}^\top\mathbf{Z}^\top\bar{\mathbf{y}} + \mathbf{w}^\top\mathbf{Z}^\top\mathbf{Z}\mathbf{w}\right) - \frac{1}{2\sigma_{\mathbf{w}}^2}\mathbf{w}^\top\mathbf{w}\right) = \\
= &\exp\left(-\frac{1}{2\sigma^2}\bar{\mathbf{y}}^\top\bar{\mathbf{y}} + \frac{1}{\sigma^2}\mathbf{w}^\top\mathbf{Z}^\top\bar{\mathbf{y}} - \mathbf{w}^\top\left(\frac{1}{2\sigma^2}\mathbf{Z}^\top\mathbf{Z} + \frac{1}{2\sigma_{\mathbf{w}}^2}\mathbf{I}\right)\mathbf{w}\right) \propto_{\mathbf{w}} \\
\propto_{\mathbf{w}} &\exp\left(-\frac{1}{2}\mathbf{w}^\top\left(\frac{1}{\sigma^2}\mathbf{Z}^\top\mathbf{Z} + \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)\mathbf{w} + \frac{1}{\sigma^2}\mathbf{w}^\top\mathbf{Z}^\top\bar{\mathbf{y}}\right). \qquad \text{(C.1)}
\end{aligned}$$

Matrix $\mathbf{\Lambda}$ is defined as

$$\mathbf{\Lambda} \triangleq \left(\frac{1}{\sigma^2}\mathbf{Z}^\top\mathbf{Z} + \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1} = \left(\frac{1}{\sigma^2}\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}. \qquad \text{(C.2)}$$

This is a well-defined matrix, which can be explained as follows. Let λ, $\mathbf{k}$ be an eigenvalue and an eigenvector respectively of $\left(\frac{1}{\sigma^2}\mathbf{Z}^\top\mathbf{Z}\right)$, i.e.,

$$\frac{1}{\sigma^2}\mathbf{Z}^\top\mathbf{Z}\mathbf{k} = \lambda\mathbf{k} \qquad \text{(C.3)}$$

Since $\left(\frac{1}{\sigma^2}\mathbf{Z}^T\mathbf{Z}\right)$ is positive semi-definite, it arises that $\lambda \geq 0$. Also, it holds that

$$\frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\mathbf{k} = \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{k} \qquad \text{(C.4)}$$

By adding the above equations, it occurs that

$$\left(\frac{1}{\sigma^2}\mathbf{Z}^\top\mathbf{Z} + \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)\mathbf{k} = \left(\lambda + \frac{1}{\sigma_{\mathbf{w}}^2}\right)\mathbf{k}, \qquad \text{(C.5)}$$

i.e., $\left(\lambda + \frac{1}{\sigma_{\mathbf{w}}^2}\right) > 0$ is an eigenvalue of $\left(\frac{1}{\sigma^2}\mathbf{Z}^\top\mathbf{Z} + \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)$. This is the proof that all eigenvalues of $\left(\frac{1}{\sigma^2}\mathbf{Z}^\top\mathbf{Z} + \frac{1}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)$ are positive, so its determinant (equal with the product of all of its eigenvalues) is positive too. As a result, it is an invertible matrix and $\mathbf{\Lambda}$ is well-defined.

Also, $\boldsymbol{\mu}$ is defined as

$$\boldsymbol{\mu} \triangleq \frac{1}{\sigma^2}\boldsymbol{\Lambda}\mathbf{Z}^\top\bar{\mathbf{y}} = \left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\mathbf{Z}^\top\bar{\mathbf{y}}. \tag{C.6}$$

Finally, Eq. (C.1) turns into:

$$\begin{aligned}(\text{C.1}) =& \exp\left(-\frac{1}{2}\mathbf{w}^\top\boldsymbol{\Lambda}^{-1}\mathbf{w} + \mathbf{w}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}\right) \propto_{\mathbf{w}} \\ \propto_{\mathbf{w}}& \exp\left(-\frac{1}{2}\mathbf{w}^\top\boldsymbol{\Lambda}^{-1}\mathbf{w} + \mathbf{w}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}\right) = \\ =& \exp\left(-\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^\top\boldsymbol{\Lambda}^{-1}(\mathbf{w}-\boldsymbol{\mu})\right)\end{aligned} \tag{C.7}$$

■

Appendix D

Proof of Eq. (3.18)

Let's denote with $\mathbf{Z}_* = \Phi(\bar{\mathbf{X}}_*)$ and $\mathbf{Z} = \Phi(\bar{\mathbf{X}})$. Then,

$$\begin{aligned}
&\int_{-\infty}^{+\infty} p_{\bar{\mathbf{y}}_*|\mathbf{w}}\left(\bar{\mathbf{y}}_*|\mathbf{w}\right) p_{\mathbf{w}|\bar{\mathbf{y}}}\left(\mathbf{w}|\bar{\mathbf{y}}\right) \mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*} \\
\propto_{\bar{\mathbf{y}}_*} &\int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2\sigma^2}\left(\bar{\mathbf{y}}_* - \mathbf{Z}_*\mathbf{w}\right)^\top\left(\bar{\mathbf{y}}_* - \mathbf{Z}_*\mathbf{w}\right)\right) \exp\left(-\frac{1}{2}\left(\mathbf{w}-\boldsymbol{\mu}\right)^\top \boldsymbol{\Lambda}^{-1}\left(\mathbf{w}-\boldsymbol{\mu}\right)\right) \mathrm{d}\mathbf{w} = \\
= &\int_{-\infty}^{+\infty} \exp\Bigg(-\frac{1}{2}\Bigg(\frac{1}{\sigma^2}\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_* - \frac{2}{\sigma^2}\mathbf{w}^\top\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \frac{1}{\sigma^2}\mathbf{w}^\top\mathbf{Z}_*^\top\mathbf{Z}_*\mathbf{w} + \\
&\qquad + \mathbf{w}^\top\boldsymbol{\Lambda}^{-1}\mathbf{w} - \mathbf{w}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} - \boldsymbol{\mu}^\top\boldsymbol{\Lambda}^{-1}\mathbf{w} + \boldsymbol{\mu}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}\Bigg)\Bigg) \mathrm{d}\mathbf{w} \overset{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^\top}{\propto_{\bar{\mathbf{y}}_*}} \\
\propto_{\bar{\mathbf{y}}_*} &\int_{-\infty}^{+\infty} \exp\Bigg(-\frac{1}{2}\Bigg(\mathbf{w}^\top\left(\frac{1}{\sigma^2}\mathbf{Z}_*^\top\mathbf{Z}_* + \boldsymbol{\Lambda}^{-1}\right)\mathbf{w} - \\
&\qquad 2\mathbf{w}^\top\left(\frac{1}{\sigma^2}\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}\right) + \frac{1}{\sigma^2}\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\Bigg)\Bigg) \mathrm{d}\mathbf{w}. \qquad \text{(D.1)}
\end{aligned}$$

Assume $\mathbf{L} = \frac{1}{\sigma^2}\mathbf{Z}_*^\top\mathbf{Z}_* + \boldsymbol{\Lambda}^{-1}$. If the Sherman–Morrison–Woodbury identity [14, p. 124, 3.8.3] is utilized, the following arises:

$$\mathbf{L}^{-1} = \boldsymbol{\Lambda} - \boldsymbol{\Lambda}\mathbf{Z}_*^\top\left(\sigma^2\mathbf{I} + \mathbf{Z}_*\boldsymbol{\Lambda}\mathbf{Z}_*^\top\right)^{-1}\mathbf{Z}_*\boldsymbol{\Lambda}. \qquad \text{(D.2)}$$

$\boldsymbol{\Lambda}$ is invertible, which also makes it a positive-definite matrix. Also, $a > 0$. Thus, a quadratic form such as $\mathbf{Z}_*\boldsymbol{\Lambda}\mathbf{Z}_*^\top$ is positive semi-definite, and $\sigma^2\mathbf{I} + \mathbf{Z}_*\boldsymbol{\Lambda}\mathbf{Z}_*^\top$ is strictly positive definite (and invertible). As a result, $\mathbf{L}^{-1}$ exists and $\mathbf{L}$ is an invertible matrix.

Furthermore, assume $\mathbf{m} = \mathbf{L}^{-1}(\frac{1}{\sigma^2}\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \boldsymbol{\Lambda}^{-1}\boldsymbol{\mu})$. Then, Eq. (D.1) turns into:

$$\begin{aligned}
\text{(D.1)} &= \int_{-\infty}^{+\infty} \exp\Bigg(-\frac{1}{2}\Bigg(\mathbf{w}^\top\mathbf{L}\mathbf{w} - 2\mathbf{w}^\top\mathbf{L}\mathbf{m} + \frac{1}{\sigma^2}\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\Bigg)\Bigg) \mathrm{d}\mathbf{w} = \\
&= \exp\left(\frac{1}{2}\mathbf{m}^\top\mathbf{L}\mathbf{m} - \frac{1}{2\sigma^2}\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\right) \int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2}(\mathbf{w}-\mathbf{m})^\top\mathbf{L}(\mathbf{w}-\mathbf{m})\right) \mathrm{d}\mathbf{w} \propto_{\bar{\mathbf{y}}_*} \\
&\propto_{\bar{\mathbf{y}}_*} \exp\left(\frac{1}{2}\left(\mathbf{m}^\top\mathbf{L}\mathbf{m} - \frac{1}{\sigma^2}\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_*\right)\right) =
\end{aligned}$$

$$= \exp\Bigg(-\frac{1}{2}\Bigg(\frac{1}{\sigma^2}\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_* - \\ -\left(\frac{1}{\sigma^2}\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}\right)^\top\left(\mathbf{L}^{-1}\right)^\top\mathbf{L}\mathbf{L}^{-1}\left(\frac{1}{\sigma^2}\mathbf{Z}_*^\top\bar{\mathbf{y}}_* + \boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}\right)\Bigg)\Bigg) =$$

$$\overset{\mathbf{L}=\mathbf{L}^\top}{=\!=} \exp\Bigg(-\frac{1}{2}\Bigg(\frac{1}{\sigma^2}\bar{\mathbf{y}}_*^\top\bar{\mathbf{y}}_* - \frac{1}{\sigma^4}\bar{\mathbf{y}}_*^\top\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}_*^\top\bar{\mathbf{y}}_* - \\ -2\frac{1}{\sigma^2}\bar{\mathbf{y}}_*^\top\mathbf{Z}_*\mathbf{L}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} - \boldsymbol{\mu}^\top\left(\boldsymbol{\Lambda}^\top\right)^{-1}\mathbf{L}^{-1}\boldsymbol{\Lambda}\boldsymbol{\mu}\Bigg)\Bigg) \propto_{\bar{\mathbf{y}}_*}$$

$$\propto_{\bar{\mathbf{y}}_*} \exp\left(-\frac{1}{2}\left(\bar{\mathbf{y}}_*^\top\left(\frac{1}{\sigma^2}\mathbf{I} - \frac{1}{\sigma^4}\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}_*^\top\right)\bar{\mathbf{y}}_* - 2\bar{\mathbf{y}}_*^\top\left(\frac{1}{\sigma^2}\mathbf{Z}_*\mathbf{L}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu}\right)\right)\right). \tag{D.3}$$

The definition of matrix $\mathbf{P}$ follows:

$$\mathbf{P} \triangleq \left(\frac{1}{\sigma^2}\mathbf{I} - \frac{1}{\sigma^4}\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}_*^\top\right)^{-1} = \sigma^2\left(\mathbf{I} - \frac{1}{\sigma^2}\mathbf{Z}_*\mathbf{L}^{-1}\mathbf{Z}_*^\top\right)^{-1} \overset{\text{Eq. (D.2)}}{=\!=\!=}$$

$$= \sigma^2\left(\mathbf{I} - \frac{1}{\sigma^2}\mathbf{Z}_*\left(\boldsymbol{\Lambda} - \boldsymbol{\Lambda}\mathbf{Z}_*^\top\left(\sigma^2\mathbf{I} + \mathbf{Z}_*\boldsymbol{\Lambda}\mathbf{Z}_*^\top\right)^{-1}\mathbf{Z}_*\boldsymbol{\Lambda}\right)\mathbf{Z}_*^\top\right)^{-1} \overset{\mathbf{K}=\mathbf{Z}_*\boldsymbol{\Lambda}\mathbf{Z}_*^\top}{=\!=\!=\!=}$$

$$= \sigma^2\left(\mathbf{I} - \frac{1}{\sigma^2}\left(\mathbf{K} - \mathbf{K}(\sigma^2\mathbf{I} + \mathbf{K})^{-1}\mathbf{K}\right)\right)^{-1} =$$

$$= \sigma^2\left(\mathbf{I} - \frac{1}{\sigma^2}\left(\mathbf{I} - \mathbf{K}(\sigma^2\mathbf{I} + \mathbf{K})^{-1}\right)\mathbf{K}\right)^{-1} =$$

$$= \sigma^2\left(\mathbf{I} - \frac{1}{\sigma^2}\sigma^2(\sigma^2\mathbf{I} + \mathbf{K})^{-1}\mathbf{K}\right)^{-1} = \sigma^2\left((\sigma^2\mathbf{I} + \mathbf{K})^{-1}\sigma^2\right)^{-1} = \sigma^2\mathbf{I} + \mathbf{K}. \tag{D.4}$$

After substituting the original values for $\mathbf{K}$, $\mathbf{Z}_*, \boldsymbol{\Lambda}$, it arises that

$$\mathbf{P} = \sigma^2\mathbf{I} + \sigma^2\Phi(\bar{\mathbf{X}}_*)\left(\Phi(\bar{\mathbf{X}})^\top\Phi(\bar{\mathbf{X}}) + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\Phi(\bar{\mathbf{X}}_*)^\top. \tag{D.5}$$

Also, vector $\mathbf{t}$ is defined as

$$\mathbf{t} \triangleq \frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_*\mathbf{L}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} \overset{(*)}{=} \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z} + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\mathbf{Z}^\top\bar{\mathbf{y}}. \tag{D.6}$$

Proof of the equality at point $(*)$*:*

$$\frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_*\mathbf{L}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} = \mathbf{Z}_*\left(\mathbf{Z}^\top\mathbf{Z} + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2}\mathbf{I}\right)^{-1}\mathbf{Z}^\top\bar{\mathbf{y}} \Leftrightarrow \frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_*\mathbf{L}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} = \mathbf{Z}_*\boldsymbol{\mu} \Leftrightarrow$$

$$\Leftrightarrow \frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_*\mathbf{L}^{-1}\boldsymbol{\Lambda}^{-1} = \mathbf{Z}_* \Leftrightarrow \frac{1}{\sigma^2}\mathbf{P}\mathbf{Z}_* = \mathbf{Z}_*\boldsymbol{\Lambda}\mathbf{L} \Leftrightarrow$$

$$\Leftrightarrow \frac{1}{\sigma^2}\left(\sigma^2\mathbf{I} + \mathbf{Z}_*\boldsymbol{\Lambda}\mathbf{Z}_*^\top\right)\mathbf{Z}_* = \mathbf{Z}_* \Leftrightarrow \boldsymbol{\Lambda}\left(\frac{1}{\sigma^2}\mathbf{Z}_*^\top\mathbf{Z}_* + \boldsymbol{\Lambda}^{-1}\right) \Leftrightarrow$$

$$\Leftrightarrow \mathbf{Z}_* + \frac{1}{\sigma^2}\mathbf{Z}_*\mathbf{\Lambda}\mathbf{Z}_*^\top\mathbf{Z}_* = \mathbf{Z}_*\mathbf{\Lambda}\frac{1}{\sigma^2}\mathbf{Z}_*^\top\mathbf{Z}_* + \mathbf{Z}_*. \tag{D.7}$$

Eq. (D.7) is valid, thus the initial statement is valid too. ■

Finally, Eq. (D.3) turns into

$$\begin{aligned}(\text{D.3}) &= \exp\left(-\frac{1}{2}\bar{\mathbf{y}}_*^\top\mathbf{P}^{-1}\bar{\mathbf{y}}_* + \bar{\mathbf{y}}_*^\top\mathbf{P}^{-1}\mathbf{t}\right) \propto_{\bar{\mathbf{y}}_*} \\ &\propto_{\bar{\mathbf{y}}_*} \exp\left(-\frac{1}{2}\bar{\mathbf{y}}_*^\top\mathbf{P}^{-1}\bar{\mathbf{y}}_* + \bar{\mathbf{y}}_*^\top\mathbf{P}^{-1}\mathbf{t} - \frac{1}{2}\mathbf{t}^\top\mathbf{P}^{-1}\mathbf{t}\right) = \\ &= \exp\left(-\frac{1}{2}(\bar{\mathbf{y}}_* - \mathbf{t})^\top\mathbf{P}^{-1}(\bar{\mathbf{y}}_* - \mathbf{t})\right). \end{aligned} \tag{D.8}$$

■

Appendix E

Proof of Equivalent Conditions

The condition for mean square convergence extracted in [39] is $\rho(\mathbf{S}) < 1$, with

$$\begin{aligned}\mathbf{S} =& \big(\mathbf{I} + \mathbf{P}\,(\mathbf{A} - \mathbf{I})\,\big) \otimes \big(\mathbf{I} + \mathbf{P}\,(\mathbf{A} - \mathbf{I})\,\big) + \\ &+ \big(\,(\mathbf{I} - \mathbf{P}) \otimes \mathbf{P}\big)\mathbf{J}\big(\,(\mathbf{A} - \mathbf{I}) \otimes (\mathbf{A} - \mathbf{I})\,\big),\end{aligned} \tag{E.1}$$

where $\mathbf{J} \triangleq \operatorname{diag}\{\operatorname{vec}(\mathbf{I})\}$.

Let's assume (as in [23]) that the update probability for each entry of the state vector is the same, i.e. $\mathbf{P} = p\mathbf{I}$. Then:

$$\begin{aligned}\mathbf{S} &= \mathbf{I} \otimes \mathbf{I} + p\,(\mathbf{A} - \mathbf{I}) \otimes \mathbf{I} + \mathbf{I} \otimes p\,(\mathbf{A} - \mathbf{I}) + (\mathbf{P} \otimes \mathbf{P})\,\big(\,(\mathbf{A} - \mathbf{I}) \otimes (\mathbf{A} - \mathbf{I})\,\big) + \\ &\quad + \big(\,(\mathbf{I} - \mathbf{P}) \otimes \mathbf{P}\big)\mathbf{J}\big(\,(\mathbf{A} - \mathbf{I}) \otimes (\mathbf{A} - \mathbf{I})\,\big) = \\ &= \mathbf{I} \otimes \mathbf{I} + p\,(\mathbf{A} - \mathbf{I}) \otimes \mathbf{I} + \mathbf{I} \otimes p\,(\mathbf{A} - \mathbf{I}) + \\ &\quad + \Big(\mathbf{P} \otimes \mathbf{P} + \big(\,(\mathbf{I} - \mathbf{P}) \otimes \mathbf{P}\big)\mathbf{J}\Big)\big(\,(\mathbf{A} - \mathbf{I}) \otimes (\mathbf{A} - \mathbf{I})\,\big) = \\ &= \mathbf{I} \otimes \mathbf{I} + p\,(\mathbf{A} - \mathbf{I}) \otimes \mathbf{I} + \mathbf{I} \otimes p\,(\mathbf{A} - \mathbf{I}) + \\ &\quad + \big(\,(\mathbf{P} \otimes \mathbf{P})\,(\mathbf{I}_{n^2 \times n^2} - \mathbf{J}) + (\mathbf{I} \otimes \mathbf{P})\,\mathbf{J}\big)\big(\,(\mathbf{A} - \mathbf{I}) \otimes (\mathbf{A} - \mathbf{I})\,\big) = \\ &= \mathbf{I} \otimes \mathbf{I} + p\,(\mathbf{A} - \mathbf{I}) \otimes \mathbf{I} + \mathbf{I} \otimes p\,(\mathbf{A} - \mathbf{I}) + \\ &\quad + \big(p^2\,(\mathbf{I}_{n^2 \times n^2} - \mathbf{J}) + p\mathbf{J}\big)\big(\,(\mathbf{A} - \mathbf{I}) \otimes (\mathbf{A} - \mathbf{I})\,\big).\end{aligned}$$

Setting $\mathbf{Y} \triangleq \big(p^2\,(\mathbf{I}_{n^2 \times n^2} - \mathbf{J}) + p\mathbf{J}\big)$ offers the respective matrix, described in [23]. ■

Appendix F

Proof of Eq. (4.27)

$$
\begin{aligned}
\mathbf{R}^{(t+1)} &\triangleq \mathbb{E}\left[\mathbf{r}^{(t+1)}\left(\mathbf{r}^{(t+1)}\right)^{\top}\right] = \\
&= \mathbb{E}\left[\mathbf{Q}\left(\mathbf{I}+\boldsymbol{\Psi}^{(t+1)}(\mathbf{A}-\mathbf{I})\right)\left(\mathbf{x}^{(t)}-\mathbf{x}_*\right)\left(\mathbf{x}^{(t)}-\mathbf{x}_*\right)^{\top}\left(\mathbf{I}+\boldsymbol{\Psi}^{(t+1)}(\mathbf{A}-\mathbf{I})\right)^{\top}\mathbf{Q}^{\top}\right] = \\
&= \mathbf{Q}\mathbb{E}\left[\left(\mathbf{I}+\boldsymbol{\Psi}^{(t+1)}(\mathbf{A}-\mathbf{I})\right)\left(\mathbf{x}^{(t)}-\mathbf{x}_*\right)\left(\mathbf{x}^{(t)}-\mathbf{x}_*\right)^{\top}\left(\mathbf{I}+(\mathbf{A}^{\top}-\mathbf{I})\boldsymbol{\Psi}^{(t+1)}\right)\right]\mathbf{Q}^{\top} = \\
&= \mathbf{Q}\mathbb{E}\Big[\mathbf{x}^{(t)}\left(\mathbf{x}^{(t)}\right)^{\top}+\boldsymbol{\Psi}^{(t+1)}(\mathbf{A}-\mathbf{I})\mathbf{x}^{(t)}\left(\mathbf{x}^{(t)}\right)^{\top}+\mathbf{x}^{(t)}\left(\mathbf{x}^{(t)}\right)^{\top}(\mathbf{A}^{\top}-\mathbf{I})\boldsymbol{\Psi}^{(t+1)}+ \\
&\qquad +\boldsymbol{\Psi}^{(t+1)}(\mathbf{A}-\mathbf{I})\mathbf{x}^{(t)}\left(\mathbf{x}^{(t)}\right)^{\top}(\mathbf{A}^{\top}-\mathbf{I})\boldsymbol{\Psi}^{(t+1)}\Big]\mathbf{Q}^{\top} = \\
&= \mathbf{Q}\Big(\mathbb{E}\Big[\boldsymbol{\Psi}^{(t+1)}(\mathbf{A}-\mathbf{I})\mathbf{x}^{(t)}\left(\mathbf{x}^{(t)}\right)^{\top}\Big]+\mathbb{E}\Big[\mathbf{x}^{(t)}\left(\mathbf{x}^{(t)}\right)^{\top}(\mathbf{A}^{\top}-\mathbf{I})\boldsymbol{\Psi}^{(t+1)}\Big]+ \\
&\qquad +\mathbb{E}\Big[\boldsymbol{\Psi}^{(t+1)}(\mathbf{A}-\mathbf{I})\mathbf{x}^{(t)}\left(\mathbf{x}^{(t)}\right)^{\top}(\mathbf{A}^{\top}-\mathbf{I})\boldsymbol{\Psi}^{(t+1)}\Big]+\mathbb{E}\Big[\mathbf{x}^{(t)}\left(\mathbf{x}^{(t)}\right)^{\top}\Big]\Big)\mathbf{Q}^{\top} = \\
&\overset{\mathbf{x}^{(t)}\perp\boldsymbol{\Psi}^{(t+1)}}{=\!=\!=} \mathbf{Q}\Big(\mathbf{R}^{(t)}+\mathbf{P}(\mathbf{A}-\mathbf{I})\mathbf{R}^{(t)}+\mathbf{R}^{(t)}(\mathbf{A}^{\top}-\mathbf{I})\mathbf{P}+\mathbf{P}(\mathbf{A}-\mathbf{I})\mathbf{R}^{(t)}(\mathbf{A}^{\top}-\mathbf{I})\mathbf{P}+ \\
&\qquad +\operatorname{diag}\left(\mathbf{P}(\mathbf{A}-\mathbf{I})\mathbf{R}^{(t)}(\mathbf{A}^{\top}-\mathbf{I})(\mathbf{I}-\mathbf{P})\right)\Big)\mathbf{Q}^{\top} = \\
&= \mathbf{Q}\big(\mathbf{I}+\mathbf{P}(\mathbf{A}-\mathbf{I})\big)\mathbf{R}^{(t)}\big(\mathbf{I}+\mathbf{P}(\mathbf{A}-\mathbf{I})\big)^{\top}\mathbf{Q}^{\top}+ \\
&\qquad +\mathbf{Q}\operatorname{diag}\left\{\mathbf{P}(\mathbf{A}-\mathbf{I})\mathbf{R}^{(t)}\left(\mathbf{A}^{\top}-\mathbf{I}\right)(\mathbf{I}-\mathbf{P})\right\}\mathbf{Q}^{\top}
\end{aligned}
$$

where $\mathbf{J}$ is given by Eq. (4.23).

If the above equation is vectorized [13, p. 34, 2.4], it turns into:

$$
\begin{aligned}
\operatorname{vec}\left\{\mathbf{R}^{(t+1)}\right\} = (\mathbf{Q}\otimes\mathbf{Q})\Big(&\big(\mathbf{I}+\mathbf{P}(\mathbf{A}-\mathbf{I})\big)\otimes\big(\mathbf{I}+\mathbf{P}(\mathbf{A}-\mathbf{I})\big)+ \\
&+\mathbf{J}\big((\mathbf{I}-\mathbf{P})\otimes\mathbf{P}\big)\big((\mathbf{A}-\mathbf{I})\otimes(\mathbf{A}-\mathbf{I})\big)\Big)\operatorname{vec}\left\{\mathbf{R}^{(t)}\right\} \overset{\text{Eq. (4.24)}}{\Longleftrightarrow} \\
\Leftrightarrow \operatorname{vec}\left\{\mathbf{R}^{(t+1)}\right\} = \mathbf{S}\operatorname{vec}\left\{\mathbf{R}^{(t)}\right\} &= \mathbf{S}^{t+1}\operatorname{vec}\left\{\mathbf{R}^{(0)}\right\}
\end{aligned}
$$

■

References

[1] K. C. Karthika, "Wireless mesh network: A survey," in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, 2016, pp. 1966–1970.

[2] M. Bembe, A. Abu-Mahfouz, M. Masonta, and T. Ngqondi, "A survey on low-power wide area networks for IoT applications," *Telecommunication Systems*, vol. 71, pp. 249–274, 2019.

[3] V. Papageorgiou, A. Nichoritis, P. Vasilakopoulos, G. Vougioukas, and A. Bletsas, "Design and implementation of ambiently powered internet-of-things-that-think with asynchronous principles," *IEEE Internet of Things Journal*, 2023, to appear.

[4] A. C. Djedouboum, A. A. Abba Ari, A. M. Gueroui, A. Mohamadou, and Z. Aliouat, "Big data collection in large-scale wireless sensor networks," *MDPI Sensors*, vol. 18, no. 4474, dec 2018.

[5] M. F. A. Salman and L. Farzinvash, "A hybrid algorithm for reliable and energy-efficient data gathering in wireless sensor networks," *International Journal of Communication Networks and Information Security*, vol. 11, no. 1, apr 2019.

[6] G. Parmar, S. Lakhani, and M. K. Chattopadhyay, "An IoT based low cost air pollution monitoring system," in *International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, Bhopal, India, 2017, pp. 524–528.

[7] M. Xu, L. Ma, F. Xia, T. Yuan, J. Qian, and M. Shao, "Design and implementation of a wireless sensor network for smart homes," in *7th IEEE International Conference on Ubiquitous Intelligence and Computing*, China, oct 2010.

[8] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, "Wireless sensor networks for healthcare," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947–1960, 2010.

[9] G. Vougioukas, N. Ntantidakis, E. Karatarakis, G. Apostolakis, and A. Bletsas, "Batteryless backscatter sensor networks—part II: Lessons from scalable deployment," *IEEE Communications Letters*, vol. 27, no. 3, pp. 768–772, 2023.

[10] E. Andrianakis, G. Vougioukas, E. Giannelos, O. Giannakopoulos, G. Apostolakis, K. Skyvalakis, and A. Bletsas, "Drone interrogation (and its low-cost

alternative) in backscatter environmental sensor networks," in *6th International Conference on Smart and Sustainable Technologies (SpliTech)*, Split, Croatia, 2021, pp. 1–6.

[11] J. Aponte-Luis, J. A. Gómez-Galán, F. Gómez-Bravo, M. Sánchez-Raya, J. Alcina-Espigado, and P. M. Teixido-Rovira, "An efficient wireless sensor network for industrial monitoring and control," *MDPI Sensors*, vol. 18, no. 182, jan 2018.

[12] J. Peixoto and D. Costa, "Wireless visual sensor networks for smart city applications: A relevance-based approach for multiple sinks mobility," *Future Generation Computer Systems*, vol. 76, pp. 51–62, 2017.

[13] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus With Applications in Statistics and Econometrics*, 3rd ed., ser. Wiley Series in Probability and Statistics. John Wiley & Sons Inc, 2007.

[14] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. USA: Society for Industrial and Applied Mathematics (SIAM), 2000.

[15] B. J. Frey, F. R. Kschischang, H.-A. Loeliger, and N. Wiberg, "Factor graphs and algorithms," in *Proc. 35th Allerton Conf. on Communications, Control, and Computing*, Illinois, USA, 1998, pp. 666–680.

[16] S. Kar and J. M. Moura, "Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems," *IEEE Signal Processing Mag.*, vol. 30, no. 3, pp. 99–109, Apr. 2013.

[17] U. A. Khan, "High-dimensional consensus in large-scale networks: Theory and applications," Ph.D. dissertation, ECE Department, Carnegie Mellon University, Aug. 2009, advisor: J. M. Moura.

[18] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[19] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, Jan. 2007.

[20] R. Diestel, *Graph Theory*, 3rd ed. New York, USA: Springer, 2005.

[21] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[22] D. Bickson, "Gaussian belief propagation: Theory and application," Ph.D. dissertation, Hebrew University of Jerusalem, Oct. 2008.

[23] B. Li and Y.-C. Wu, "Convergence analysis of gaussian belief propagation under high-order factorization and asynchronous scheduling," *IEEE Trans. Signal Processing*, vol. 67, no. 11, pp. 2884–2897, Jun. 2019.

[24] J. Du, S. Ma, Y.-C. Wu, S. Kar, and J. M. F. Moura, "Convergence analysis of distributed inference with vector-valued gaussian belief propagation," *Journal of Machine Learning Research*, vol. 18, no. 172, pp. 1–38, Apr. 2018.

[25] B. C. Levy, *Principles of Signal Detection and Parameter Estimation*. New York, USA: Springer, 2008.

[26] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, vol. 14. The MIT Press, 2001.

[27] F. Chung, *Spectral Graph Theory*, ser. Regional Conference Series in Mathematics, no. 92. USA: American Mathematical Society, 1997.

[28] O. Teke and P. P. Vaidyanathan, "Random node-asynchronous updates on graphs," *IEEE Trans. Signal Processing*, vol. 67, no. 11, pp. 2794–2809, Jun. 2019.

[29] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. New York, USA: Cambridge University Press, 2014.

[30] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Springer, 2018.

[31] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282.

[32] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[33] S. Abe, *Support Vector Machines for Pattern Classification*. Springer, 2010.

[34] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[35] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, USA: Association for Computing Machinery, 2016, p. 785–794.

[36] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, 6th ed. John Wiley & Sons Inc, 2021.

[37] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[38] P. N. Hossein, *Introduction to Probability, Statistics, and Random Processes*. Kappa Research, LLC, 2014.

[39] O. Teke and P. P. Vaidyanathan, "Randomized asynchronous recursions with a sinusoidal input," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2019, pp. 1491–1495.

[40] O. Teke and P. P. Vaidyanathan, "Node-asynchronous spectral clustering on directed graphs," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Spain, May 2020, pp. 5325–5329.

[41] Silicon Labs, "Thunderboard Sense 2 EFR32xG24 devices," https://www.silabs.com/development-tools/wireless/efr32xg24-dev-kit, [online; accessed 5-April-2023].

[42] Silicon Labs, "Si7021 Humidity & Temperature Sensor," https://www.silabs.com/sensors/humidity/si7006-13-20-21-34/device.si7021-a20-gm1, [online; accessed 5-April-2023].

[43] Silicon Labs, "Simplicity Studio 5," https://www.silabs.com/developers/simplicity-studio, [online; accessed 5-April-2023].

[44] Silicon Labs, "Gecko Software Development Kit," https://www.silabs.com/developers/gecko-software-development-kit, [online; accessed 5-April-2023].

[45] L. W. Couch, *Digital and Analog Communication Systems*, 8th ed. Pearson, 2012.

[46] M. K. Simon and M. S. Alouini, *Digital Communication over Fading Channels: A Unified Approach to Performance Analysis*. New York, USA: John Wiley & Sons, 2000.

[47] N. C. Strole, "The IBM token-ring network — a functional overview," *IEEE Network*, vol. 1, no. 1, pp. 23–30, jan 1987.