Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria dell'Automazione

Tesi di laurea magistrale
8 Marzo 2016

$SO\left(2\right)$ Rotation Estimation in Camera Networks

Laureanda
Alessia Cocco

Relatore
Prof. Angelo Cenedese

Correlatore
Dott. Giulia Michieletto

Anno Accademico 2015-2016

Contents

List of Figures

Abstract

This thesis focuses on the study of the localization problem in planar camera networks. Two distributed models for detecting the orientation of cameras are discussed and analyzed.

The performance of these models is evaluated with respect to modifications in the network, such as the use of different communication protocols among agents and the change in the topology of the network.

The theoretical findings are assessed via numerical simulations based on synthetic and experimental scenarios.

Chapter 1

Introduction

Nowadays systems involving camera networks are a well-known technology.
Camera networks have a very large spectrum of application fields. They play an important role in solving several problems such as environmental monitoring [27],[11], exploration in critical conditions, intrusion detection [8], [23] and surveillance [4] tasks, management and employment of robots [12] for different purposes and tracking [24] of objects.

The digital information collected from a camera system, as images and videos, has to be set in a certain spatial collocation to allow a consistent scene reconstruction and hence have some significance. This is the context where the localization problem arises.
From this point of view it is essential that a camera network can be localized in the space where it operates. This means that each device has to be identified by a unique orientation and position (pose) with respect to a fixed reference frame.

The simplest possible solution is a manual setting of the poses of the cameras, though this strategy is not much reliable due to the high probability of errors.
A self-localizing camera network finds all the poses from certain initial noisy measurements exploiting the communication among the agents of the network. Indeed this is a more common and suitable method since it can be easily adjusted to changes of the network topology.
This autonomous localization issue can be solved either in a centralized manner or in a distributed one. In the first strategy the information is gathered by a unique and powerful control unit which have to process all the data. This approach is certainly optimal to perform algorithms, still it limits the flexibility and scalability of the network, moreover it is computationally expensive and communications costs are high.
Conversely these kind of problems do not emerge taking up a distributed paradigm which is, consequently, preferable in many situations: it exploits the capability of communication among the cameras and the locality of information such that each device provides an estimates of its own pose. On the other hand, a distributed solution

may lead to sub-optimal result and only approximate its centralized counterpart.

1.1 State of the art

Concerning the localization problem of camera networks, literature has ranged in many different directions depending on the particular assumptions on the framework and choices about the method to be implemented.

Some works address this issue using traditional localization methods [15] for a generic standard WSN exploiting the signal power strength, angle of arrival, time of arrival or time difference of arrival to obtain positions.

Otherwise beacons, which are fixed nodes of known positions, can be deployed within the network. In this way the positions of the nodes are estimated from the measurements of distances between these beacons and the nodes themselves.

An interesting distributed algorithm DALT, which stands for Distributed Alternating Localization-Triangulation, was built by Mantzel et al.[17]: each camera locally locates itself using some triangulated points and then, communicating with the other agents nearby, triangulates more common points.

Other researches lay the foundations on computer vision. For example Meingast et al.[18] proposed a procedure to recover the pose of the cameras from the tracking of a moving object. The algorithm is developed in two steps: an intra-camera step provides the track of the moving object on the image plane of a single camera, while the inter-camera step creates the correspondence between cameras using object images tracks. Fixing a certain camera coordinate frame as a reference, each absolute camera pose is achieved from the relative positions and orientations.

Anjum [2] planned an algorithm for camera localization using trajectory estimation (CLUTE) which is consistent only if the views of the cameras do not overlap. The trajectories caught by each camera are completely reconstructed with the aid of a Kalman filter and linear regression for the unobserved areas. These estimations are then filtered to obtain the relative poses comparing the exit and entry points of an object in each field of view of a camera.

Considering the problem of localization in a two dimensional space and only about the rotational part of the pose, conditions under which this work is developed, there were built different algorithms to solve this task. Borra et al.[5] proposed a method based on a non-convex optimization problem. In a first step a convex region where the global minimum should be located is identified through a set of integers. In a second phase the optimization problem is carried out using classical algorithms.
Another distributed approach in similar conditions is given in Piovan et al.[21]'s paper. The scenario consists of a ring network made up of sensing nodes capable of detecting the presence of other agents and to provide relative noisy measurements of the poses. The worked out algorithm reduces the effect of the noise and then the estimation is computed through a least square procedure with an exponential rate of convergence.

1.2 Aim and structure of the thesis

The subject of this thesis is the study of particular features of the localization problem in a camera network in a specific scenario.
This work is based on the research that has been conducted by A. Cenedese, M. Michielan and G. Michieletto [6].

The starting point is the Tron-Vidal [26] approach to the localization problem, which provides the estimates of the poses of cameras in a connected network by the minimization of a suitable functional cost on $SE\left(3\right)^N$ taking inspiration from the consensus procedure among agents in a network. The authors use a computer vision technique to obtain the relative measurements between pairs of cameras, which are noisy and inconsistent. Then they improve these measurements deriving a distributed algorithm and they give a condition for convergence that depends on the step-size of the consensus procedure and the degree of the network.

In the context of this work each camera is assumed to know nothing about its physical position in the space, but it can only compute and share its self-estimate from noisy measurements about the relative poses among neighbouring cameras.
In a real set-up these noisy measures can be obtained through some calibration techniques such as the eight point algorithm or using the Bouguet camera toolbox provided by MATLAB®.

The analysis is simplified by the introduction of certain restrictions.
First of all the translational contribution of the functional cost is neglected in order to focus on the estimate of rotations.
Moreover the 3D space is reduced to a plane where the rotational part of the pose of a camera can be identified with only one parameter.
Then, according to [6], two distributed methods to obtain the estimates of the angles are proposed that are different as far as the dynamical convergence properties are involved but generally converge to the same solution.
These types of procedure allows to understand how the topology of the network has to be to achieve convergence to the centralized solution and how to improve the performance analyzing the consensus matrix.
In particular, the contribution of this dissertation consists in studying how these distributed models behave while changing some aspects of the network, which are listed below, in the summary of the structure of the thesis.

Following this introductory chapter, some basic mathematical tools regarding the overall approach used are unveiled in Chapter 2. Chapter 3 presents the localization problem in a detailed way and leads to the formulation of the distributed models for solving it. In Chapter 4 communication protocols are discussed: an evaluation in terms of effectiveness of the proposed methods and speed of convergence is made and possible problems are tackled. Then a comparison among them is made. Actually a similar analysis has been carried out by G.Baggio, M. Michielan and S. Patron [3] in $SE\left(3\right)$ referring to the Tron-Vidal algorithm. In Chapter 5 the camera network is modified in its topology: the structure of the graph is changed by adding links and analyzing particular configurations of the network. Then some methods of edge selection are discussed in order to detect the best set of links that give the best performance. The last Chapter 6 illustrates an application of the studied strategies on a real physical scenario.

Each chapter integrates the theoretical notions and argumentations with simulations and experimental tests to verify the validity of the results and also to observe particular emergent trend.

Chapter 2

Mathematical background

2.1 Riemannian geometry of $SO(3)$

The group of rotations in $\mathbb{R}^3$ is called special orthogonal group and it is defined as

$$SO(3) = \left\{R \in \mathbb{R}^{3\times 3} : R^T R = I, det(R) = +1\right\}. \tag{2.1}$$

The Lie algebra[1] associated to this Lie group[2] $SO(3)$ is the space of skew-symmetric matrices

$$\mathfrak{so}(3) = \left\{\hat{\mathbf{v}} \in \mathbb{R}^{3\times 3} : \hat{\mathbf{v}} = -\hat{\mathbf{v}}^T\right\} \tag{2.2}$$

where the matrix $\hat{\mathbf{v}}$ generates the cross product by $\mathbf{v} \in \mathbb{R}^3$, *i.e.* $\hat{\mathbf{v}}\mathbf{u} = \mathbf{v} \times \mathbf{u}, \forall \mathbf{u} \in \mathbb{R}^3$. Given a rotational matrix $R \in SO(3)$ the tangent space at R is defined as

$$T_R(SO(3)) = \{R\hat{\mathbf{v}} : \hat{\mathbf{v}} \in \mathfrak{so}(3)\}. \tag{2.3}$$

At this point a relation between the rotational space and the tangent one can be found through the concepts of *exponential map* and *logarithm map*.
An exponential map at a point $R \in SO(3)$ is a diffeomorphism which associates to each point Δ in a neighbourhood of the origin of the tangent space $T_R(SO(3))$ a point S on the (unique) geodesic passing through R in the direction Δ:

$$\exp_R(\Delta) : T_R(SO(3)) \rightarrowtail SO(3). \tag{2.4}$$

On the other hand the logarithm map $\log_R(S) : SO(3) \rightarrowtail T_R(SO(3))$ is the inverse function of the previous one.
These mappings at a generic point of the rotational space can be computed using the parallel transport method:

$$\exp_R(\Delta) = R \exp\left(R^T \Delta\right) \tag{2.5}$$

[1] A *Lie algebra*, associated to a Lie group G, is a pair $(\mathfrak{g}, [\cdot,\cdot])$ with $\mathfrak{g}$, vector space over G and $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ closed under the Lie bracket operation and satisfying the Jacobi identity.

[2] A *Lie group* G is a smooth manifold whose group operations are smooth.

$$\log_R(S) = R\log\left(R^T S\right) \tag{2.6}$$

where $\exp(\cdot)$ and $\log(\cdot)$ are the exponential and the logarithm of a matrix, respectively. An intuitive representation of these mappings is illustrated in Figure 2.1.

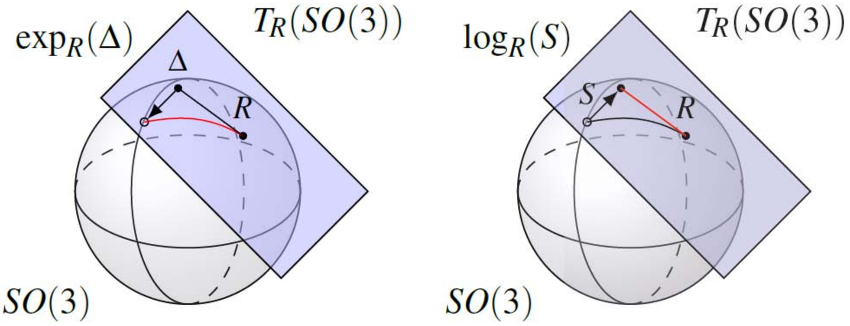


Figure 2.1: Representation of the exponential and logarithm maps on $SO(3)$.

In order to solve the calibration task it is useful to introduce a definition of distance in the $SO(3)$ space. There are different types of metrics [13], [19] that can be adopted. For instance the *Riemannian metric* is a family of varying inner products on the tangent space

$$\langle R_i, R_j\rangle = tr\left(R_i^T R_j\right), \quad \forall R_i, R_j \in T_R(SO(3)) \tag{2.7}$$

and it can be used to measure the length of curves between two given points on the manifold $SO(3)$. A curve with minimum length is known as *geodesic* and the length between the two points takes the name of *Riemannian or geodesic distance*:

$$d_R^2(R_i, R_j) = \frac{1}{2}\left\|\log\left(R_i^T R_j\right)\right\|_F^2, \quad R_i, R_j \in SO(3). \tag{2.8}$$

Alternatively using the *Frobenius metric* the *Frobenius or chordal distance* between two points on $SO(3)$ is calculated as

$$d_F^2(R_i, R_j) = \|R_i - R_j\|_F^2, \quad R_i, R_j \in SO(3) \tag{2.9}$$

where $\|\cdot\|_F$ is the Frobenius norm.

The group of 2D rotations $SO(2) = \left\{R \in \mathbb{R}^{2\times 2} : R^T R = I, det(R) = +1\right\}$ is isomorphic to the circle group $\mathbb{S}^1 = \{z \in \mathbb{C} : |z| = 1\}$. In this situation the concept of Riemannian exponential map coincides with the exponential map of the Lie group and with the general complex exponential: the tangent space to 1 of the unit circle can be recognized as the imaginary axis jt, with $t \in \mathbb{R}$, in the complex plane

$$jt \mapsto \exp(jt) = e^{jt} = \cos(t) + j\sin(t).$$

The two distances defined in (2.8) and (2.9) in $SO(3)$, on the 2D space, are represented in Figure 2.2.

The Riemannian distance between two points R_{β_1}, R_{β_2} on $SO\,(2)$ is the length of the arc joining the corresponding two points on the unit circle $z_1 = e^{j\beta_1}$, $z_2 = e^{j\beta_2} \in \mathbb{S}^1$, on the other hand the Frobenius distance is the segment bounded by the end points z_1, z_2, *i.e.* the chordal distance.

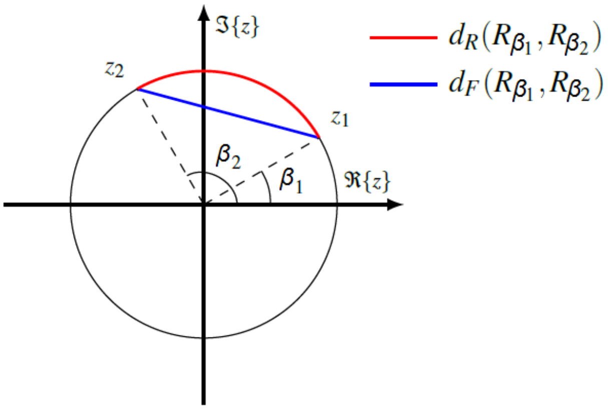

$\Im\{z\}$
$\Re\{z\}$
z_2
z_1
β_2
β_1
$d_R(R_{\beta_1}, R_{\beta_2})$
$d_F(R_{\beta_1}, R_{\beta_2})$

Figure 2.2: Graphical representation of the Riemmanian and Frobenius distances on the 2D space.

Finally, it is worth noting that if $R_i \simeq R_j$ the two definitions of distance are similar, for example this happens when R_i is an exact rotational matrix, while $R_j = \widehat{R}_i$ is an estimate of it.

2.2 Elements of graph theory

Each device in a camera network can be seen as an element in a communication graph. A node of a graph has its own state and can update it through the exchange of information with the other nodes.
As a mathematical tool a graph can be defined as follows:

Definition: A **graph** $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is a set of nodes $\mathscr{V} = (v_1, ..., v_n)$ and a set of edges $\mathscr{E} = \mathscr{V} \times \mathscr{V}$ which connect the nodes.

Some notions and properties concerning a graph are here reported.

- Two nodes of a graph v_i and $v_j \in \mathscr{V}$ are **adjacent** if they are connected by an edge, *i.e.*: $(v_i, v_j) \in \mathscr{E}$.
- The **neighbourhood** of a node v_i is the set of its neighbours defined as:

$$\mathscr{N}_i = \{v_j \in \mathscr{V} : (v_i, v_j) \in \mathscr{E}\}\,.$$

- The **path** between v_i and v_j is an ordered sequence of nodes whose the first one is v_i while the last is v_j and they are connected by edges belonging to $\mathscr{E}$.

- A **cycle** is a path where the first node is also the last one.
- A graph is said to be **connected** if for all couples of nodes (v_i, v_j) it exists a path which connects them.
- A graph is said to be **direct** if the edges have a direction associated with them, contrariwise it is **undirect**.
- A graph is **strongly connected** if it exists an oriented path which connects v_i and v_j, $\forall\ i, j$.
- A graph is said to be **complete** if it is simple and undirected and each couple of vertices (v_i, v_j), $\forall\ i, j$ is connected by an unique edge.
- A **regular** graph of degree k is a graph where each vertex has the same number of neighbours k.
- A **bipartite** graph is a graph which can be partitioned in two disjoint and independent sets U e V such that every edge connects a vertex in U to one in V.
- A **tree** is a connected graph without cycles.

An intuitive mathematical way to represent a graph is through matrices.
In particular, in a graph with N nodes, the *degree matrix* $\Delta_{\mathscr{G}} \in \mathbb{R}^{N\times N}$ is a diagonal matrix whose elements are the degrees of each node v_i, *i.e.* the number of edges outgoing the *i-th* node:

$$\Delta_{\mathscr{G}} = \begin{bmatrix} d\,(v_1) & & \\ & \ddots & \\ & & d\,(v_n) \end{bmatrix} \tag{2.10}$$

On the other hand the *adjacency matrix* gives information about the links among the nodes in the network, in fact $A_{\mathscr{G}} \in \mathbb{R}^{N\times N}$ is:

$$A_{\mathscr{G}}\,(i,j) = \begin{cases} 1 & \text{if}\,(v_i, v_j) \in \mathscr{E} \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

The presence of the value 1 in the diagonal points out the fact that the nodes have a self-loop, *i.e.* the vertex knows its own information.
Moreover, if the matrix A is symmetric, the graph is undirect, otherwise it is direct.

2.3 Consensus problem

In multiagent system the consensus problem aims to convergence to a same common value for all the agents. In detail, every agent of the system has a local information state, that is modified by the communication with the other agents till the reaching of an agreement, namely the consensus value.

Formally, the consensus problem is described as follows.
The task of a system made up of N agents is the distributed estimate of a certain quantity, starting from the local measurements q_i of each agent i. The communication network that involves the agents can be represented by a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where nodes belonging to $\mathscr{V}$ correspond to the agents and the presence of an edge e_{ij} indicates the communication from the *i-th* node to the *j-th*. The communication network is supposed to be fixed in time.
The algorithm of consensus can be modelled with the equation:

$$x_i\left(t+1\right) = \sum_{j \in \mathscr{N}_i} P_{ij} x_j\left(t\right), \quad x_i\left(0\right) = q_i$$

where $x_i\left(t\right)$ is the information carried by the *i-th* agent at instant t and $\mathscr{N}_i$ is the neighbourhood associated to the same node. P is a suitable time-invariant matrix such that $P_{ij} = 0$ if e_{ij} does not belong to $\mathscr{E}$, otherwise $P_{ij} > 0$. Moreover if an agent is a neighbour of itself, *i.e.* $\mathscr{G}$ contains all the self-loops (i, i), it must hold the following condition:

$$\sum_{j \in \mathscr{N}_i} P_{ij} = 1.$$

This means that P has to be a row stochastic matrix.

The expression of the system using a compact form is

$$x\left(t+1\right) = Px\left(t\right), \quad x\left(0\right) = q \tag{2.12}$$

with $x, q \in \mathbb{R}^N$. In this context the consensus is obtained, starting from the system (2.12), if $\forall x\left(0\right) \in \mathbb{R}^N$ the limit

$$\lim_{t\to\infty} x\left(t\right) = \alpha \mathbf{1}$$

exists, where α is the consensus value and $\mathbf{1} = (1, \ldots, 1)^T$.
It remains to demonstrate that $\lim\limits_{t\to\infty} P^t x\left(0\right)$ exists and converge to a finite value.
For this purpose, it can be applied the theorem of Perron-Frobenius, under the assumption of a primitive P, thanks to which P has an eigenvalue in 1 with single multiplicity, while all the other eigenvalues have an inferior modulus. Being w_0^T the left eigenvector associated to the eigenvalue 1 it holds $P^t \to \mathbf{1} w_0^T$, therefore

$$\lim_{t\to\infty} x\left(t\right) = \mathbf{1} w_0^T x\left(0\right) = \mathbf{1}\alpha.$$

Moreover, if P is doubly stochastic, the consensus value coincides with the average of the initial conditions, in formula:

$$\lim_{t\to\infty} x(t) = \frac{1}{N}\mathbf{1}\mathbf{1}^T x(0).$$

2.4 Camera model

The *absolute* pose of a camera is completely defined by the pair $g = (R, T)$, where R is a rotation matrix and T a translational vector. This pose belongs to the *special Euclidean group*

$$SE(3) = \left\{ g = (R, T) : R \in SO(3), T \in \mathbb{R}^3 \right\}$$

which is the group of rigid body transformations.
As regards the rotational part of the pose, in a camera network the orientation of a device is represented by a rotational matrix in $SO(3)$. As Figure 2.3 illustrates, if two cameras are identified with c_i and c_j, their corresponding *absolute* rotational matrices are R_i and R_j, because they are referring to a fixed reference frame taken as *world* frame centered in O. With these premises, the *relative* orientation between these two cameras is computed as $R_{ij} = R_i^T R_j$.

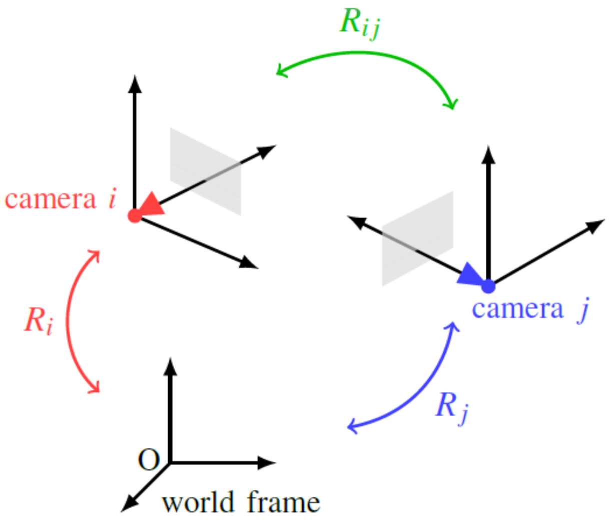


Figure 2.3: Graphical representation about rotational matrices in a camera system.

Considering also the translational contribution of the pose, the relation of neighbouring cameras c_i and c_j can be computed from the composition of the associated absolute poses $g_i = (R_i, T_i)$ and $g_j = (R_j, T_j)$ as

$$g_i \circ g_j = (R_i R_j, R_i T_j + T_i).$$

The relative poses expressions are

$$\begin{cases} R_{ij} = R_i^T R_j \\ T_{ij} = R_i^T \left(T_j - T_i\right) \end{cases}$$

As a typical example of use of a camera network, the problem of perimeter monitoring is considered. The cameras surround the scene of interest and are connected via a communication network. In such a case the topology of the network can be modelled as a circulant graph, whose representation is given in Figure 2.4.

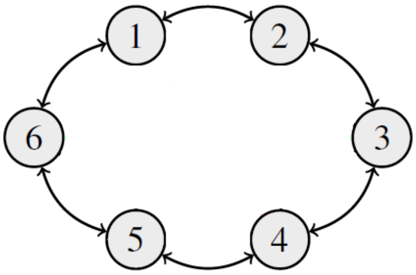


Figure 2.4: Representation of a circulant graph with 6 cameras.

2.5 Pinhole camera view

A camera is commonly represented by a pinhole model (Figure 2.5).
According to this geometric interpretation, the point **C** is the *optical center* of the camera whose corresponding coordinate reference system is individuated by the axis $(\mathsf{x}, \mathsf{y}, \mathsf{z})$.
The *world coordinate system* is the reference system centered in **O**.
Figure 2.5 also shows the *image plane* π, plane parallel to the xy axis along the coordinate z, placed at distance f, named *focal length*, from the optical center.

- Relation between $\mathbf{Q_C}$ and $\mathbf{Q_0}$

  A generic point $\mathbf{Q}$ in the $\mathbb{R}^3$ space is described as $\mathbf{Q_0} = (X_0, Y_0, Z_0)$ in the world coordinate system. This point $\mathbf{Q_C}$ seen from the camera perspective is connected to $\mathbf{Q_0}$ by the expression

$$\mathbf{Q_C} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + T$$

  where (R, T), respectively the rotational matrix and the translational vector, are the euclidean relations between the world coordinate and the camera systems.

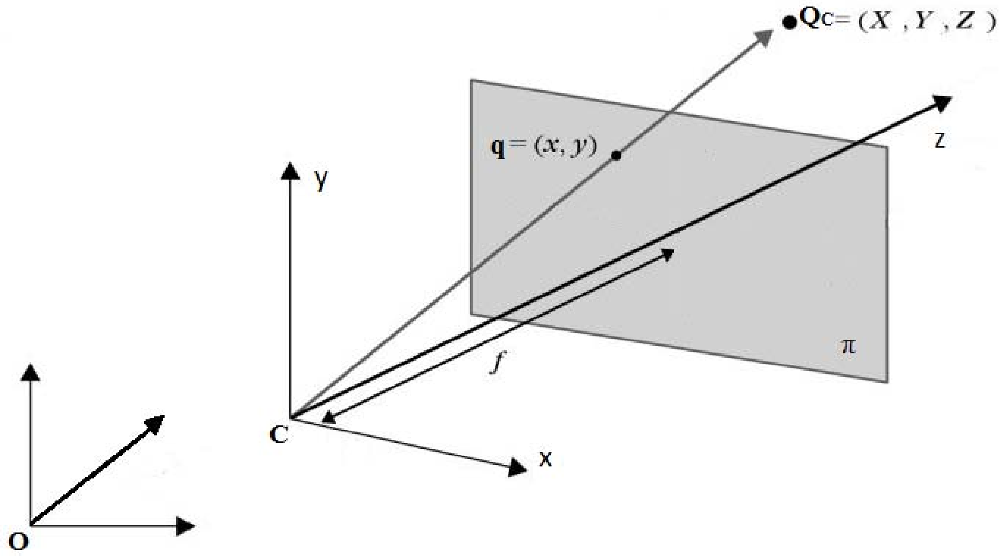


Figure 2.5: Representation of a pinhole camera.

- Relation between $\mathbf{Q_0}$ and $\mathbf{q}$

  The point $\mathbf{Q_C}$ in $\mathbb{R}^3$ has its own projection $\mathbf{q}$ onto the image plane π through the optical center. The coordinates of this two-dimensional point are denoted as $\mathbf{q} = (x, y)$.
  Taking advantage of the relations of projective geometry:

$$\begin{cases} x = f\dfrac{X}{Z} \\ \\ y = f\dfrac{Y}{Z} \end{cases}$$

  and working with the homogeneous coordinates (operation of *embedding*, $\mathbb{R}^2 \to \mathbb{R}^3$), up to a scaling factor $\lambda \neq 0$, the projection point $\mathbf{q}$ can be expressed referring to $\mathbf{Q}_C$ as

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} =$$

$$= \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K\Pi_0\mathbf{Q_C}$$

The matrix K, called the *calibration matrix*, contains the intrinsic parameters of the camera, while Π_0 is the *standard projection matrix.* Indeed the matrix K is given by the product

$$K = K_s K_f = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where are highlighted the five intrinsic parameters:

- s_x and s_y are *scaling factors* along the respective directions and represent the resolution of a pixel on the image plane;
- o_x and o_y are *offsets* in the respective directions which would be ideally the center of the image plane;
- finally s_θ is the *skew factor*, which is an indicator of the shape of a generic pixel on the plane.

It is immediate to conclude that the relation between $\mathbf{q}$ and $\mathbf{Q_0}$ is

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{P} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

calling $P := K\Pi_0 g$, where g includes the six extrinsic parameters in (R, T). Overall, the number of degrees of freedom of a camera is 11, since it is only defined up to a scaling factor.

2.6 Epipolar geometry

Considering the scenario of a multiview geometry where two cameras i and j are able to view the same point $\mathbf{Q}$, the projections of this point on the two image planes are $\mathbf{q_i}$ and $\mathbf{q_j}$. These two planar points are linked by

$$\mathbf{q_j} T_{ij}^{\perp} R_{ij} \mathbf{q_i} = 0$$

which is the *epipolar constraint* between the two cameras [16].

In the image 2.6 some elements of interest are marked to understand the meaning of the epipolar constraint. The line connecting the centers of the two cameras $\mathbf{C_i}$ and $\mathbf{C_j}$ is called *baseline.* The intersection points between the baseline and the image planes π_i and π_j are named *epipoles* e_i, e_j. *Epipolar lines* l_i and l_j are the lines given by the intersection of the epipolar plane Π, generated from the points $\mathbf{Q}$ and the centers $\mathbf{C_i}$ and $\mathbf{C_j}$, and the image planes (these lines depend on the position of $\mathbf{Q}$).

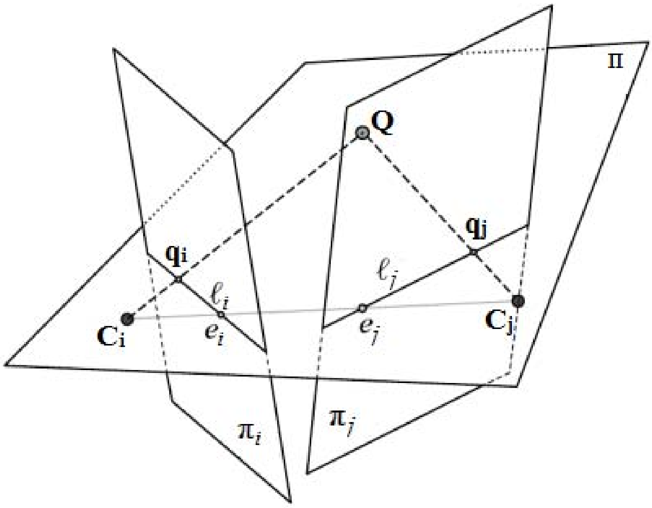


Figure 2.6: Representation of multiview geometry.

Therefore because of the epipolar constraint the projection $\mathbf{q_j}$ of a point $\mathbf{Q}$ onto the image plane of the *j-th* camera must belong to the epipolar line l_j.

The matrix $E := T^{\perp}R$ is the *essential matrix*, whose importance is due to the fact that it codifies the relative position between the cameras.

2.7 Algorithms for calibration

A calibration algorithm is a computer vision procedure through which the essential matrix E is acquired from a set of points belonging to the 2D space and then the pose of the camera in $SE(3)$ can be reconstructed. In this work this kind of algorithm becomes useful to find the initial data of the localization problem, which are relative noisy rotations $\widetilde{R}_{ij}$ between pairs of cameras.

2.7.1 Eight-point algorithm

The eight-point algorithm was created by Christopher Longuet-Higgins in 1981 to compute the three dimensional structure of a scene. It starts from the correspondence between image points $\mathbf{q}$ to estimate the essential matrix E. The number of points taken in consideration is at least eight, to which the method owes its name.
The algorithm consists in three phases. Firstly a linear homogeneous problem is built up which can be solved with the help of SVD, then the solution is an estimated matrix $\widetilde{E}$. $\widetilde{E}$ is projected onto the essential matrices space $\mathcal{E}$ obtaining E. It is noticeable

that performing this operation the epipolar constraint on E may be lost. Finally the correct pose is chosen by inspection, trying a point in the four possible configurations. The criticality of this approach comes up when the extracted points produce particular degenerate configurations. In this case the feature points are derived through the use of chessboards which, being planar surfaces in a 3D environment, fall within the critical sets just cited above (*critical surfaces*).

A more suitable technique to overcome the problem of critical surfaces and specifically designed for working with chessboards images is the Bouguet camera toolbox provided by MATLAB®.
Briefly, this tool allows the estimation of the intrinsic parameters of each device (*calibration matrix*), taking multiple pairs of images of the same chessboard from different angles by a chosen camera. The pose is identified with respect to the reference frame of the chessboard. In this way, the initial relative noisy rotations and consequently the associated angles can be derived as inputs of the localization procedure.

Chapter 3

Localization problem

In this work the localization problem of the network cuts down on finding the orientation of cameras, while the translational part of the pose is assumed to be known or determined otherwise. This means that it exists a set of relative poses $\{g_{ij}\}$ such that all the other nodes are univocally determined. In $SO\left(3\right)$ this task is translated in determining three parameters which are sufficient to describe an orientation in the space.

In fact in robotics there are different ways to express a triplet of angles (Euler angles) which stands for a rotation, such as ZYZ angles or RPY angles.
Alternatively a possible representation of rotations is through quaternions.
According to ZYZ representation, a rotation R is obtained by composition of elementary rotations with respect to the current frame. In particular, it is computed as the post-multiplication $R = R_z\left(\varphi\right) R_{y'}\left(\theta\right) R_{z''}\left(\psi\right)$, where the angles $\left(\varphi, \theta, \psi\right)$ represent the rotations performed in the current reference frame about the new axis $\left(z, y', z''\right)$.
If a rotation R is represented by RPY (Roll-Pitch-Yaw) angles, it is computed as premultiplication of matrices $R = R_z\left(\gamma\right) R_y\left(\beta\right) R_x\left(\alpha\right)$, where R is expressed with respect to a fixed reference frame. In fact each angle of the triplet $[\alpha, \beta, \gamma]$ (pitch, yaw, roll) refers to the amount of rotation that has to be performed in the corresponding axis of the reference frame to obtain the whole rotation R.

Considering a camera network made up of PTZ cameras, angles $[\alpha, \beta, \gamma]$ stand for tilt, pan and zoom movements respectively.
In the planar framework here considered the degree of freedom required to identify a rotation is only one angle, *i.e* the *pan* angle β.
Under these assumptions cameras lie in a plane at the same height and they can view the scene in front of them rotating with respect to a perpendicular axis passing through their center of mass.

The estimates of the rotations are obtained as the solution of a minimization problem over a functional cost that will be discussed in the following. First of all it is necessary to have some initial conditions about the poses of the cameras. These piece of information is derived from the application of a computer vision algorithm based on the knowledge of the noisy relative measurements $\widetilde{R}_{ij}$. Adopting one of the metrics presented in section 2.1, the functional cost is defined as the sum of the distance between the true unknown relative poses and the noisy measurements of communicating nodes:

$$\varphi_* \left(R_1, \ldots, R_N\right) = \frac{1}{2} \sum_{e_{ij} \in \mathscr{E}} d_*^2 \left(R_{ij}, \widetilde{R}_{ij}\right) = \frac{1}{2} \sum_{e_{ij} \in \mathscr{E}} d_*^2 \left(R_i^T R_j, \widetilde{R}_{ij}\right) \tag{3.1}$$

The definition of the functional just exposed yields to the set of estimated rotations (under the condition that $\widehat{R}_i \in SO\left(3\right), \quad \forall i = 1, \ldots, N$):

$$\left\{\widehat{R}_1, \ldots, \widehat{R}_N\right\} = \underset{R_1, \ldots, R_N}{\operatorname{argmin}} \varphi_* \left(R_1, \ldots, R_N\right) = \underset{R_i, R_j}{\operatorname{argmin}} \frac{1}{2} \sum_{e_{ij} \in \mathscr{E}} d_*^2 \left(R_i^T R_j, \widetilde{R}_{ij}\right) \tag{3.2}$$

3.1 Derivation in the 2D space

In a two dimensional space the minimization of the functional cost can be solved analytically. In this context the rotation of the *i-th* camera is univocally determined by its pan angle β_i. Then the rotation matrix associated is

$$R_i = \left[\begin{array}{ccc} \cos\beta_i & 0 & \sin\beta_i \\ 0 & 1 & 0 \\ -\sin\beta_i & 0 & \cos\beta_i \end{array} \right]. \tag{3.3}$$

Similarly, the relative rotation matrix, which describes the relative orientation between the *i-th* and *j-th* camera, is

$$R_{ij} = \left[\begin{array}{ccc} \cos\left(\beta_j - \beta_i\right) & 0 & \sin\left(\beta_j - \beta_i\right) \\ 0 & 1 & 0 \\ -\sin\left(\beta_j - \beta_i\right) & 0 & \cos\left(\beta_j - \beta_i\right) \end{array} \right] \tag{3.4}$$

with $\beta_{ij} = \beta_j - \beta_i$. Since only the pan angle is considered, the calculations can be simplified restricting the study from $SO\left(3\right)$ to the $SO\left(2\right)$ group.

In Chapter 2 it has already been highlighted that the group of rotations in $SO\left(2\right)$ is isomorphic to $\mathbb{S}^1$, *i.e.* $SO\left(2\right) \cong \mathbb{S}^1$:

$$\mathbb{S}^1 \ni z = e^{j\beta_i} \longleftrightarrow \left[\begin{array}{cc} \cos\beta_i & \sin\beta_i \\ -\sin\beta_i & \cos\beta_i \end{array} \right] = R_i \in SO\left(2\right), \tag{3.5}$$

where the angle of rotation β_i belongs to the interval $[-\pi, \pi)$.

The expression of the functional (3.1), choosing the geodesic distance, becomes the following:

$$\begin{aligned}\varphi_R\left(R_1, \ldots R_N\right) &= \frac{1}{2}\sum_{i,j\in\mathscr{E}} d_R^2\left(R_{ij}, \tilde{R}_{ij}\right)\\ &= \frac{1}{2}\sum_{c_i\in\mathscr{V}}\sum_{c_j\in\mathscr{N}_i} d_R^2\left(R_i^T R_j, \tilde{R}_{ij}\right) \qquad (3.6)\\ &= \frac{1}{2}\sum_{c_i\in\mathscr{V}}\sum_{c_j\in\mathscr{N}_i}\frac{1}{2}\left\|\log\left(R_j^T R_i\right)^T \tilde{R}_{ij}\right\|_F^2\end{aligned}$$

From the definition of Frobenius norm:

$$\left\|\log\left(R_j^T R_i \tilde{R}_{ij}\right)\right\|_F^2 = tr\left(\log\left(R_j^T R_i \tilde{R}_{ij}\right)\right)\left(\log^T\left(R_j^T R_i \tilde{R}_{ij}^T\right)\right). \qquad (3.7)$$

Recalling (3.4), the noisy relative rotation is the matrix

$$R_j^T R_i = \left[\begin{array}{cc}\cos\left(\beta_i - \beta_j\right) & \sin\left(\beta_i - \beta_j\right)\\ -\sin\left(\beta_i - \beta_j\right) & \cos\left(\beta_i - \beta_j\right)\end{array}\right] \qquad (3.8)$$

and similarly the product in (3.7) can be written as

$$R_j^T R_i \tilde{R}_{ij} = \left[\begin{array}{cc}\cos\left(\beta_i - \beta_j + \tilde{\beta}_{ij}\right) & \sin\left(\beta_i - \beta_j + \tilde{\beta}_{ij}\right)\\ -\sin\left(\beta_i - \beta_j + \tilde{\beta}_{ij}\right) & \cos\left(\beta_i - \beta_j + \tilde{\beta}_{ij}\right)\end{array}\right]. \qquad (3.9)$$

The principal logarithm [19] of a generic matrix $R \in SO\left(2\right)$ is given by

$$\log\left(R\right) = \begin{cases} 0 & \text{if } \theta = 0\\ \dfrac{\theta}{2\sin\theta}\left(R - R^T\right) & \text{if } \theta \neq 0\end{cases} \qquad (3.10)$$

where $\theta \in \left(-\pi, \pi\right)$ satisfies $tr\left(R\right) = 1 + 2\cos\theta$ and in the case of R_i it actually coincides with the angle rotation $\theta = \beta_i$.
This expression leads to the following calculations matching the product of the rotation matrices with their respective rotation angles

$$\begin{aligned}&\log\left(R_j^T R_i \tilde{R}_{ij}\right) =\\ &= \frac{\beta_i - \beta_j + \tilde{\beta}_{ij}}{2\sin\left(\beta_i - \beta_j + \tilde{\beta}_{ij}\right)}\left[\begin{array}{cc}0 & 2\sin\left(\beta_i - \beta_j + \tilde{\beta}_{ij}\right)\\ -2\sin\left(\beta_i - \beta_j + \tilde{\beta}_{ij}\right) & 0\end{array}\right] \qquad (3.11)\\ &= \left(\beta_i - \beta_j + \tilde{\beta}_{ij}\right)\left[\begin{array}{cc}0 & 1\\ -1 & 0\end{array}\right]\end{aligned}$$

Thus the Frobenius norm is simply equal to

$$tr\left(\left(\beta_i-\beta_j+\tilde{\beta}_{ij}\right)^2\begin{bmatrix}1&0\\0&1\end{bmatrix}\right)=2\left(\beta_i-\beta_j+\tilde{\beta}_{ij}\right)^2 \tag{3.12}$$

Finally, substituting (3.12) in (3.6) the functional cost reduces to

$$\varphi_R\left(R_1,\ldots,R_N\right)=\frac{1}{2}\sum_{c_i\in\mathscr{V}}\sum_{c_j\in\mathscr{N}_i}\left(\beta_i-\beta_j+\tilde{\beta}_{ij}\right)^2. \tag{3.13}$$

The expressions of the gradient and the Hessian of φ_R are

$$\begin{aligned}\left[\nabla\varphi_R\right]_i &= \frac{1}{2}\left[2\sum_{j\in\mathscr{N}_i}\left(\beta_i-\beta_j+\tilde{\beta}_{ij}\right)^2-2\sum_{k\in\mathscr{N}_i}\left(\beta_i-\beta_k+\tilde{\beta}_{ik}\right)\right]\\ &= 2\mathrm{deg}\left(c_i\right)\beta_i-2\sum_{j\in\mathscr{N}_i}\beta_j+\sum_{j\in\mathscr{N}_i}\left(\tilde{\beta}_{ij}-\tilde{\beta}_{ji}\right)\end{aligned} \tag{3.14}$$

$$\left[H\varphi_R\right]_{ij}=\frac{\partial\varphi_R}{\partial\beta_i\partial\beta_j}=\begin{cases}2\mathrm{deg}\left(c_i\right) & \text{if } i=j\\ -2 & \text{if } i\neq j \quad j\in\mathscr{N}_i\\ 0 & \text{if } i\neq j \quad j\notin\mathscr{N}_i\end{cases} \tag{3.15}$$

The Hessian matrix (3.15) is also represented by the formula: $H\varphi_R=2\mathscr{L}_{\mathscr{G}}$, where $\mathscr{L}_{\mathscr{G}}$ is the Laplacian matrix associated with the graph defined as the difference between the degree matrix and the adjacency matrix: $\mathscr{L}_{\mathscr{G}}=\Delta_{\mathscr{G}}-A_{\mathscr{G}}$.

In [26] the estimation of the rotational part of the pose is derived exploiting consensus algorithms, but with a Riemannian gradient descent on the space of rotations. The algorithm proposed by the authors can be divided in two phases.
The functional that has to be minimized is expressed as

$$\varphi_R\left(\{R_i\}\right)=\frac{1}{2}\sum_{(i,j)\in\mathcal{E}}d_R^2\left(R_i^TR_j,\tilde{R}_{ij}\right).$$

At the first step each node k of the network computes the Riemannian gradient of φ_R as

$$\mathrm{grad}_{R_k}\varphi_R=-R_k\sum_{(k,i)\in\mathcal{E}}\left(\log\left(R_k^TR_i\tilde{R}_{ki}^T\right)+\log\left(R_k^TR_i\tilde{R}_{ki}\right)\right).$$

This quantity can be calculated in a distributed fashion, since the computation of the gradient φ_R involves a sum over nodes i, neighbours of the *k-th* node.
Secondly, the estimate $R_k\left(l\right)$ at the *l-th* iteration is updated moving along the geodesic in the direction $-\mathrm{grad}_{R_k(l)}$ with a suitable step-size of ε, namely

$$R_k\left(l+1\right)=\exp_{R_k(l)}\left(-\varepsilon\mathrm{grad}_{R_k(l)}\varphi_R\right).$$

These two steps are repeated until convergence is reached and the number of estimation has to be fixed at the beginning.
The main problem in this approach is the fact that the functional cost φ_R has multiple local minima, indeed the initial set of rotations $\{R_i\}$ has to be close to the optimal solution in order to perform a correct localization.

3.2 Centralized approach

The minimization of the functional cost (3.13) is now performed to get the estimates of the pan angles.
Since the functional φ_R is a convex function and its second derivative is positive semi-definite within the set $\{\beta_i \in [-\pi, \pi)\}$, it is possible to reach the minimum, which is not only local, but also global.
Hence the solution of the optimization problem is achieved by nullifying the first derivative $\nabla\varphi_R = 0$:

$$2\mathscr{L}_{\mathscr{G}} \underbrace{\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}}_{\boldsymbol{\beta}} = \underbrace{\begin{bmatrix} \sum_{j\in\mathscr{N}_1} \left(\tilde{\beta}_{j1} - \tilde{\beta}_{1j}\right) \\ \vdots \\ \sum_{j\in\mathscr{N}_N} \left(\tilde{\beta}_{jN} - \tilde{\beta}_{Nj}\right) \end{bmatrix}}_{\widetilde{\boldsymbol{\Delta\beta}}} \tag{3.16}$$

In a shorthand notation: $\nabla\varphi_R = 2\mathscr{L}_{\mathscr{G}}\boldsymbol{\beta} - \widetilde{\boldsymbol{\Delta\beta}} = 0$.

Now the equation (3.16) can be solved with the standard least squares method, if $\widetilde{\boldsymbol{\Delta\beta}} \notin \ker\mathscr{L}_{\mathscr{G}}$. Actually there are infinite solutions of the form $\boldsymbol{\beta} = \frac{1}{2}\mathscr{L}_{\mathscr{G}}^{\dagger}\widetilde{\boldsymbol{\Delta\beta}} + \frac{1}{2}\left(I - \mathscr{L}_{\mathscr{G}}^{\dagger}\mathscr{L}_{\mathscr{G}}\right)\boldsymbol{\chi}, \boldsymbol{\chi} \in \mathbb{R}^N$ and $\boldsymbol{\chi} \in \ker\, \mathscr{L}_{\mathscr{G}}$. This is due to the fact that matrix $\mathscr{L}_{\mathscr{G}}$, since the graph is connected, has at least one 0 eigenvalue, hence $\left(I - \mathscr{L}_{\mathscr{G}}^{\dagger}\mathscr{L}_{\mathscr{G}}\right)\boldsymbol{\chi}$ belongs to ker $\mathscr{L}_{\mathscr{G}}$, which has unitary dimension. The minimum norm solution is given by the former contribution of the formula just written:

$$\boldsymbol{\beta}^* = \frac{1}{2}\mathscr{L}_{\mathscr{G}}^{\dagger}\widetilde{\boldsymbol{\Delta\beta}} \tag{3.17}$$

This expression is the *centralized* solution to the rotation localization problem, where the matrix $\mathscr{L}_{\mathscr{G}}^{\dagger}$ is the pseudo-inverse[1] of the Laplacian matrix.
The relation (3.17) points out the fact that the information required to have the estimates of the pan angles is the knowledge of the topology of the network and the initial noisy relative measurements that can be derived, for example, by employing the eight-point algorithm (see 2.7) .

[1] The *pseudo-inverse* of a matrix M is computed as $M^\dagger = \left(M^T M\right)^{-1} M^T$.

3.3 Distributed approach

Formula (3.14) suggests an iterative *distributed* technique to solve the same problem of localization. Indeed a local expression for the estimate for each single pan angle can be derived from it:

$$\beta_i = \frac{\sum_{j \in \mathscr{N}_i} \beta_j}{\deg(c_i)} - \frac{\widetilde{\Delta\beta}_i}{2\deg(c_i)} \tag{3.18}$$

The meaning of this expression is that each camera updates its estimate from the data received by its neighbours and from the relative noisy measurements regarding it.

From (3.18) it is straightforward to draw out a *first discrete state-space model* in a vector form

$$\boldsymbol{\beta}(k+1) = F\boldsymbol{\beta}(k) + \boldsymbol{u} \tag{3.19}$$

where the state matrix F is equal to the adjacency matrix normalized by the degree matrix and it is given by

$$F = \Delta_{\mathscr{G}}^{-1} A_{\mathscr{G}} \tag{3.20}$$

while the vector $\boldsymbol{u}$, input of the system, is

$$\boldsymbol{u} = \frac{1}{2}\Delta_{\mathscr{G}}^{-1}\widetilde{\boldsymbol{\Delta\beta}}. \tag{3.21}$$

In this model each camera does not take into account its self-estimate at the *k-th* iteration to compute the next estimate, but only the information coming from its neighborhood $\mathscr{N}_i$.

It can be demonstrated that the solution of this first model converges to the centralized one (3.17):

$$\begin{aligned}
\boldsymbol{\beta}^* &= F\boldsymbol{\beta}^* + \boldsymbol{u} \\
&= \Delta^{-1}A\boldsymbol{\beta}^* + \frac{1}{2}\Delta^{-1}\widetilde{\boldsymbol{\Delta\beta}} \\
(\Delta - A)\boldsymbol{\beta}^* &= \frac{1}{2}\widetilde{\boldsymbol{\Delta\beta}} \\
\boldsymbol{\beta}^* &= \frac{1}{2}\mathscr{L}_{\mathscr{G}}^{\dagger}\widetilde{\boldsymbol{\Delta\beta}}.
\end{aligned}$$

The procedure is managed by the eigenvalues of the matrix of consensus F and, since F is row stochastic, its N real eigenvalues are distributed in the interval $[-1, 1]$[14]. If the theorem of Perron-Frobenius holds (see 2.3), the dominant eigenvalue, always present, is $\lambda_0 = 1$ with single multiplicity. The speed of convergence of the angles to the centralized solution is thus regulated by the second largest eigenvalue of the matrix F.

A first observation is that eigenvalue $\lambda = -1$ with unitary multiplicity has not to be part of the set of the eigenvalues of F in fact its presence cause the instability of the system and the estimates oscillate around the equilibria without converging. This algebraic observation reflects on the structure of the network, indeed the presence of $\lambda = -1$ corresponds to the characterization[2] of a **bipartite** graph (see section 2.2) [14].

The non-convergent behaviour of the system can be solved adopting some strategies, in particular, changes at different levels of the network are pointed out below and developed here and in the next chapters.

1. As a first solution, instead of working directly on the structure of the network, a second state-space model is proposed. By means of this approach the amount of information increases because of the introduction of self-loops and the dynamics of the system is controlled in its performance tuning a specific parameter in an appropriate way.

2. The communication protocol of the network is changed from a synchronous one to a different method for exchanging data.

3. The topology, *i.e.* the configuration of the network, is modified by the addition of a link in order to obtain a non-bipartite graph.

As far as the first point is concerned, a *second state-space model* [6] is introduced to change the range of the eigenvalues of the matrix F, in particular to avoid the presence of the eigenvalue $\lambda = -1$ which is the cause of an oscillator behaviour of the angles. Such a model ...:

$$\begin{aligned} \boldsymbol{\beta}(k+1) &= \eta\boldsymbol{\beta}(k) + (1-\eta)\left(F\boldsymbol{\beta}(k) + \boldsymbol{u}\right) \\ &= \underbrace{(\eta I_3 + (1-\eta)F)}_{F_1}\boldsymbol{\beta}(k) + (1-\eta)\,\boldsymbol{u} \end{aligned} \tag{3.22}$$

The new matrix F_1, this time, considers also the self-estimate of each node being the identity matrix an additive term of the sum. The parameter η represents the weight attributed to the previous estimate of the camera updating its state and it permits to regulate the speed of convergence to the centralized solution.

[2] A graph is said to be bipartite if it does not contain cycles whose length is odd, *i.e.* all cycles in the graph have an even number of nodes.

The equilibria of this second model are the same as the centralized method:

$$\begin{aligned}\boldsymbol{\beta}^* &= \eta\boldsymbol{\beta}^* + (1-\eta)\left(F\boldsymbol{\beta}^* + \boldsymbol{u}\right)\\ &= \eta\boldsymbol{\beta}^* + (1-\eta)\,\Delta^{-1}A\boldsymbol{\beta}^* + \frac{1}{2}(1-\eta)\,\Delta^{-1}\widetilde{\boldsymbol{\Delta\beta}}\\ (\Delta - A)\,\boldsymbol{\beta}^* &= \frac{1}{2}\widetilde{\boldsymbol{\Delta\beta}}\\ \boldsymbol{\beta}^* &= \frac{1}{2}\mathscr{L}_{\mathscr{G}}^{\dagger}\widetilde{\boldsymbol{\Delta\beta}}.\end{aligned}$$

Differently from (3.20), the region where the eigenvalues of the new stochastic matrix F_1 lie is the circle centred in $(\eta, 0)$ having radius $\rho = 1-\eta$ ($\eta \in (0,1) \implies \rho \in (0,1)$), the domain can also be restricted to the range $[-1+2\eta, 1]$, since the eigenvalues are real. This leads to the exclusion of the unstable mode caused by the presence of the eigenvalue $\lambda = -1$.
Moreover the control parameter η allows the regulation of the speed of convergence because its value influences directly the eigenvalues of F_1, in particular, the second largest eigenvalue corresponding to the dominant mode.

The sign of the second largest eigenvalue of the state matrix F_1 determines the behaviour of the response of the system: if it is negative, the convergence is reached with a transient oscillatory period, while, on the other hand, if this value is positive, the estimated angles are represented by monotonic and smooth curves all along the procedure.

In order to give evidence of these observations, a circulant camera network with a unique cycle, denoted by C_N^1, is considered. Generally the eigenvalues of the matrix F_1 for a network of such a type are calculated as the set:

$$\sigma_F = \left\{\lambda_h = \eta + (1-\eta)\cos\left(\frac{2\pi}{N}h\right), \quad h = 0, \ldots, N-1\right\}$$

Taking for granted that the largest positive eigenvalue is $\lambda_0 = 1$ and the next largest positive eigenvalue is λ_1, a distinction between the case of a network constituted by an even or an odd number of cameras has to be made to understand which is the largest negative eigenvalue. The aim of these analysis is the detection of the sign of the second largest eigenvalue that rules the convergence of the estimated angles.

If N is even, the largest negative eigenvalue is $\lambda_{N/2} = -1 + 2\eta$. This one leads the convergence only if the condition $\left|\lambda_{N/2}\right| \geq |\lambda_1|$ holds. Solving the inequality

$$\begin{aligned}1 - 2\eta &\geq \eta + (1-\eta)\cos\left(\frac{2\pi}{N}\right)\\ \eta\left(3 - \cos\left(\frac{2\pi}{N}\right)\right) &\leq 1 - \cos\left(\frac{2\pi}{N}\right)\end{aligned}$$

$$\eta \leq \frac{1-\cos\left(\dfrac{2\pi}{N}\right)}{3-\cos\left(\dfrac{2\pi}{N}\right)}$$

and exploiting the trigonometric property $\cos(2\alpha) = 2\cos^2(\alpha) - 1$, the following threshold is obtained:

$$\eta \leq \frac{\left(1-\cos\left(\frac{\pi}{N}\right)\right)\left(1+\cos\left(\frac{\pi}{N}\right)\right)}{1+\left[\left(1-\cos\left(\frac{\pi}{N}\right)\right)\left(1+\cos\left(\frac{\pi}{N}\right)\right)\right]} = \eta_e^{cr} \tag{3.23}$$

In the opposite case of an odd number of devices in the network, the largest negative eigenvalue is $\lambda_{(N-1)/2} = -\left(\eta + (1-\eta)\cos\left(\frac{2\pi}{N}\frac{N-1}{2}\right)\right) = \eta - (1-\eta)\cos\left(\frac{\pi}{N}\right)$. The critical value of the control parameter η is found by imposing $\left|\lambda_{(N-1)/2}\right| \geq |\lambda_1|$:

$$\begin{aligned} -\eta + (1-\eta)\cos\left(\frac{\pi}{N}\right) &\geq \eta + (1-\eta)\cos\left(\frac{2\pi}{N}\right) \\ 2\eta - \eta\left[\cos\left(\frac{2\pi}{N}\right) - \cos\left(\frac{\pi}{N}\right)\right] &\leq -\left[\cos\left(\frac{2\pi}{N}\right) - \cos\left(\frac{\pi}{N}\right)\right] \end{aligned}$$

$$\eta \leq \frac{-\left[\cos\left(\dfrac{2\pi}{N}\right) - \cos\left(\dfrac{\pi}{N}\right)\right]}{2-\left[\cos\left(\dfrac{2\pi}{N}\right) - \cos\left(\dfrac{\pi}{N}\right)\right]}$$

Thus leading to the expression:

$$\eta \leq \frac{\left(1-\cos\left(\frac{\pi}{N}\right)\right)\left(\frac{1}{2}+\cos\left(\frac{\pi}{N}\right)\right)}{1+\left[\left(1-\cos\left(\frac{\pi}{N}\right)\right)\left(\frac{1}{2}+\cos\left(\frac{\pi}{N}\right)\right)\right]} = \eta_o^{cr} \tag{3.24}$$

If η satisfies these inequalities, an approximate value of the centralized solution can be inferred by averaging, otherwise the iterations may condition the final steady result, giving an incorrect value.

3.4 Simulations

In this first analysis the simulations are evaluated in a fixed environment made up of a circulant camera network C_N^1 where each node can share its information with the previous and next neighbours.

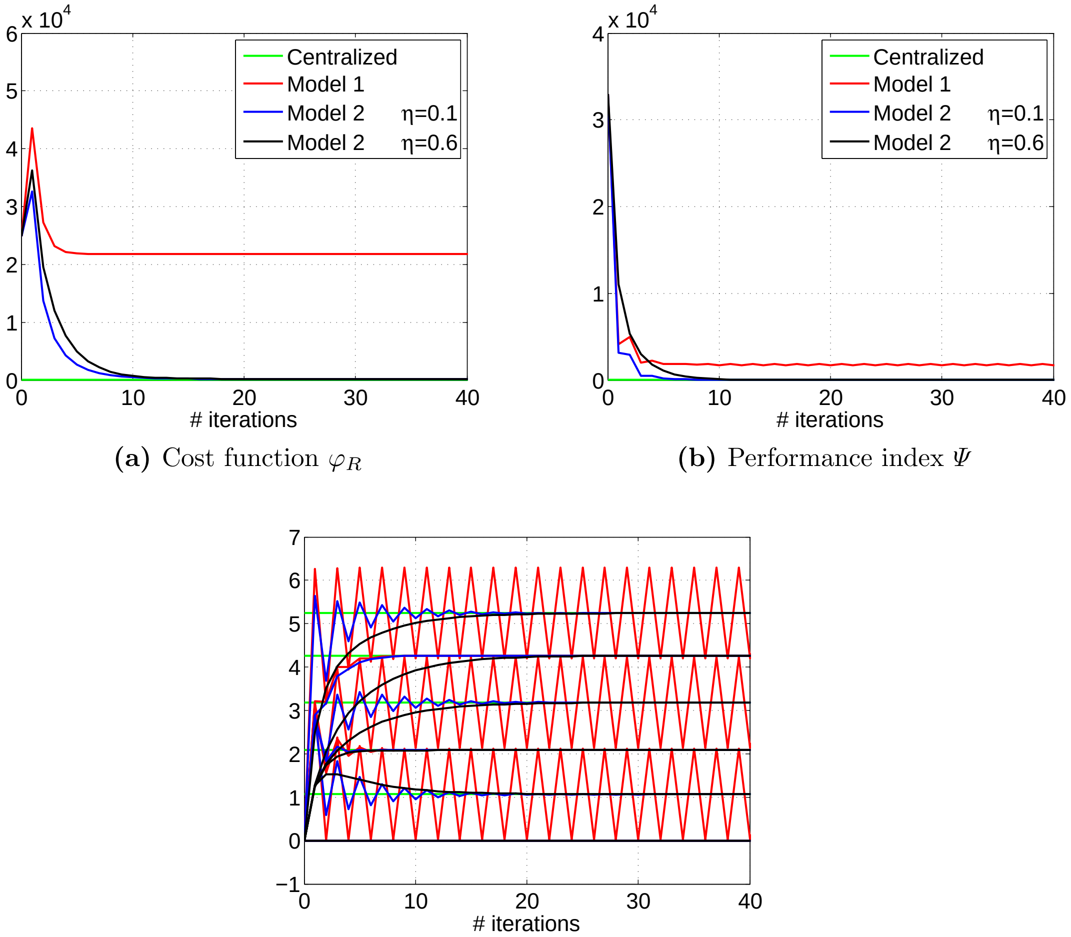
In these simulations the number of agents N is chosen equals to 6 but the achievements can be easily extended to networks with a great amount of devices.

The estimation procedure of the pan angles begins from the initial condition $\boldsymbol{\beta}(0) = \mathbf{0}$ and it is repeated for 40 iterations. To simulate the relative angle measurements between the cameras a noise uniformly distributed in the interval $\left[-\frac{\pi}{18}, \frac{\pi}{18}\right]$ is added to the difference of the angles.

The evaluation of the performance of the estimation process is made considering the centralized solution (3.17) as the ground truth. In addition to that, the cost function φ_R is an index of the goodness of the convergence as well as the *mean square error* defined as

$$\Psi\left(\widehat{\beta}_1, \ldots, \widehat{\beta}_N\right) = \frac{1}{N} \sum_{i=1}^{N} \left(\widehat{\beta}_i - \beta_i\right)^2 \tag{3.25}$$

i.e. the squared error between the estimated angles $\widehat{\beta}_i$ and the real values β_i.



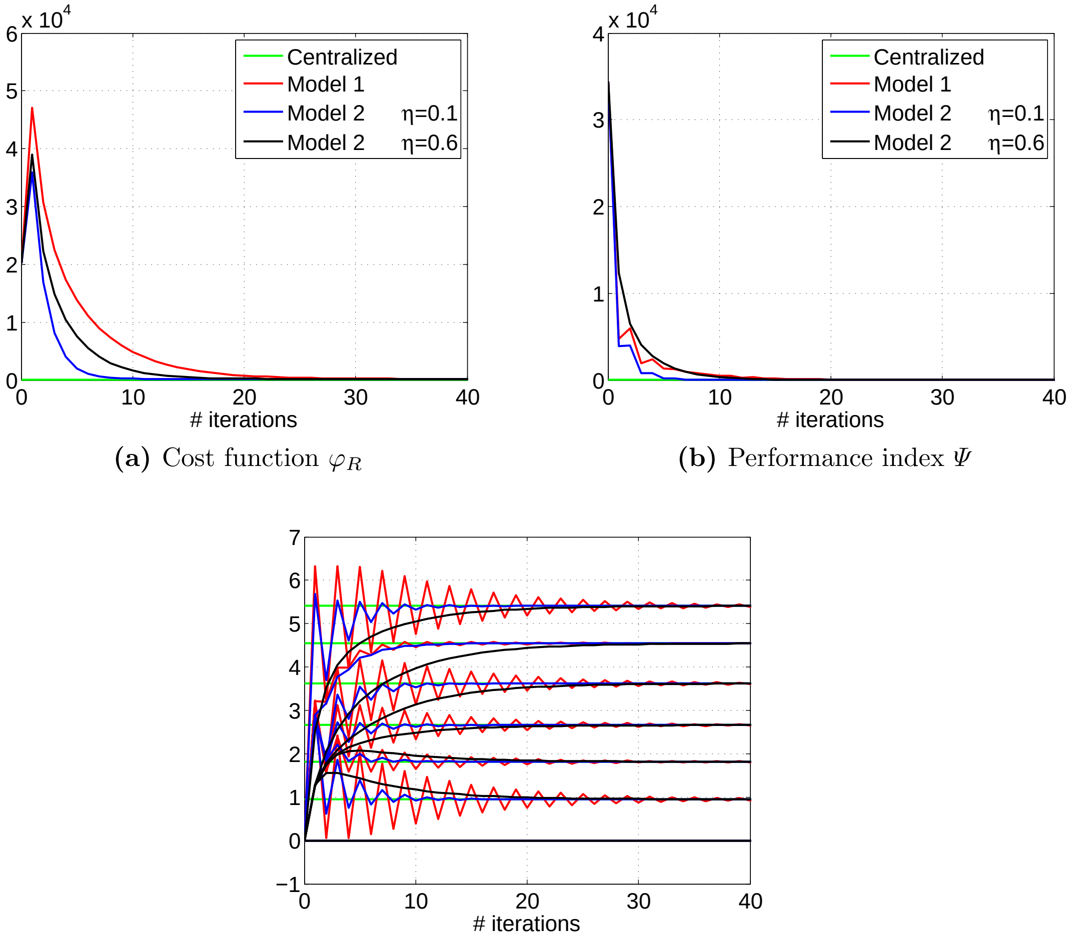
(a) Cost function φ_R

(b) Performance index Ψ

(c) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 3.1: Comparison between models (3.19) and (3.22): convergence analysis for a cyclic network C_6^1.

Figure 3.1 and Figure 3.2 show the result of the estimation process for a network respectively with an even ($N = 6$) and an odd ($N = 7$) number of devices. Focusing on the first model, in Figure 3.1a the cost function does not reach the 0 value and also the mean squared error (Figure 3.1b) does not decrease to zero. Some of the estimated angles of Figure 3.1c oscillate around the centralized solution. As already observed, this is due to the presence of the eigenvalue $\lambda = -1$, which is the cause of the instability of the system. On the contrary, for C_7^1 and applying the model (3.22), since $\lambda = -1$ is not within the spectrum of the matrix F, the stability of the system and the convergence of the angles to the equilibrium value are assured.



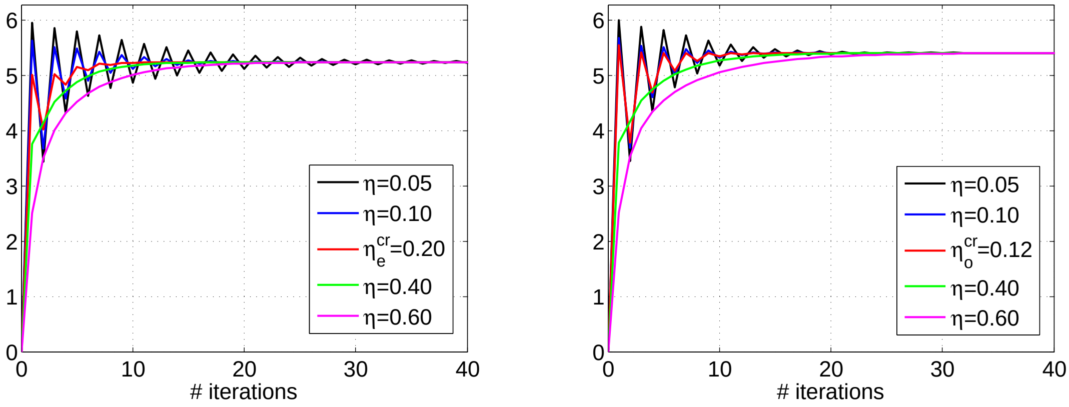
(a) Cost function φ_R

(b) Performance index Ψ

(c) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 3.2: Comparison between models (3.19) and (3.22): convergence analysis for a cyclic network C_7^1.

The following Figure 3.3 illustrates how the parameter η influences directly the performance of the convergence towards the equilibrium. It confirms the results previously obtained from the theory: over the threshold constituted by the critical values η^{cr} the estimated angle reaches its true value asymptotically, while if (3.23) and (3.24) are not verified oscillations arise.



(a) Estimated angle $\widehat{\beta}_6$ (rad)

(b) Estimated angle $\widehat{\beta}_7$ (rad)

Figure 3.3: Model (3.22): scaling of the control parameter η with respect to the critical value in the C_6^1 (left) and C_7^1 (right) networks.

Chapter 4

Communication protocols

This chapter deals with the change in the communication among the cameras of the network.
The localization problem described in the previous chapter is studied with respect to different communication protocols. The types of protocols analyzed are:

1. *Synchronous* - in this protocol at the same instant of time t all cameras in the network send their information to their neighbours and update their state (4.1a). It has already been discussed in the previous simulations.

2. *Broadcast* - at the instant t only one node communicates its state to its neighbours and only these latter update their state. At the same time all the other agents are turned off (4.1b).

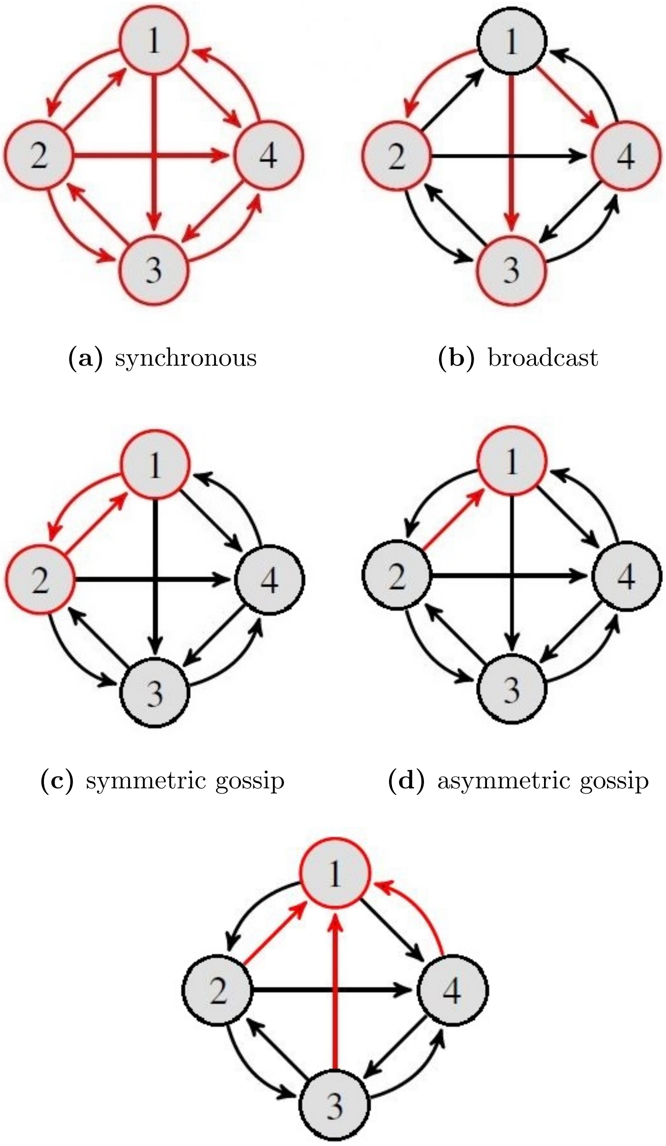
3. *Asynchronous symmetric gossip* - at the instant t two adjacent nodes exchange their information and both update their state (4.1c).

4. *Asynchronous asymmetric gossip* - the exchange of information is unilateral that is a single camera sends information to the other one, which updates its estimate (4.1d).

5. *Coordinated broadcast* - it is a transmission of information where at the instant t a node of the network receives the data from all its neighbours and then computes its state on the basis of them (4.1e).

These protocols are better explained by Figure 4.1.

The study of the communication protocols is significant because if the localization solution is similar to the one achieved with a synchronous protocol, where all the communication links are active, it might be possible to reduce the consumption of energy and communications in the network, otherwise a trade off is posed among complexity, convergence properties, resources needed.



(a) synchronous

(b) broadcast

(c) symmetric gossip

(d) asymmetric gossip

(e) coordinated broadcast

Figure 4.1: Schematic representation of the protocols for a generic network of 4 devices. Arrows point the direction in which the communication works. Nodes circled in red update their state, while the red links are the active edges at a precise instant.

The main point in changing the protocol with respect to a synchronous one is the transition from a matrix of consensus P that is time-invariant (see section 2.3) to a matrix $P(t)$ that is dependent on time. At every instant t_i a dynamic graph is associated to the matrix $P(t_i)$ which has the same nodes $\mathscr{V}$ of the original graph but the set of edges $\mathscr{E}_i$ is a subset of all the possible links among the agents. Consequently the convergence is obtained by studying by the product of all the varying $P\left(t_i\right), t_i = 0, \ldots T$:

$$x\left(T\right) = P\left(T\right) P\left(T-1\right) \ldots P\left(1\right) P\left(0\right) x\left(0\right)$$

The difficulty of these *switching systems* is that a general theory for studying the stability does not exist, but only some propositions under specific assumptions are valid.

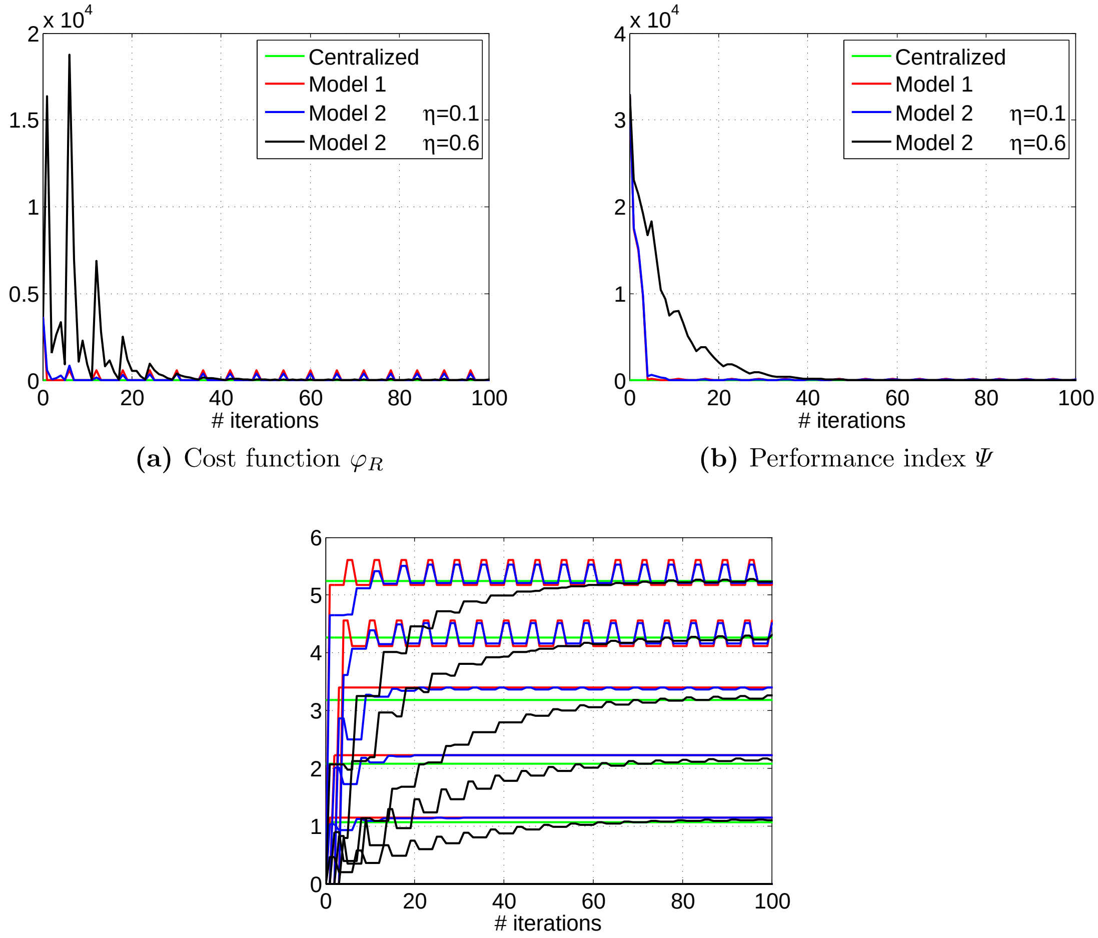
4.1 Convergence analysis

In this section the functioning of the estimation process with the protocols just mentioned is tested through simulations.
For a generic protocol, at a instant t_i an angle estimate relating to an agent is computed according to the state of the neighbouring nodes and to the relative noisy measurement and, without deriving again the measurement, the estimate is repeated for some iterations t. For simplicity it is assumed to take only an iteration per estimate, *i.e.* $t = 1$ and therefore there is a switch in the communication at every iteration.
The number of iterations of the process is increased to 100, since active communication links are less than in the synchronous case per iteration.
The noise that influences the measurements is always in the interval $\left[\frac{-\pi}{18}, \frac{\pi}{18}\right]$.
The reported simulations refer to a C_6^1 camera network. Generally the results obtained for this structure can be extended to all the other C_N^1 networks, even if the number of agents is odd, because communication does not involve all the cameras in a time instant.
At first the examined protocols are deterministic which means that the communication takes place in a specific pre-arranged order starting with the estimate of the first node and keep going clockwise.

In Figure 4.2 the results for the deterministic **broadcast** protocol are represented. The state-space model 1 (3.19) (red line) is compared with the second approach (3.22) for two possible values of the control parameter η ($\eta = 0.1$, blue line and $\eta = 0.6$, black line).

In all the three cases the convergence is not obtained exactly due to the presence of noise. Model 1 is the fastest to settle down to a constant value near the optimal solution. Concerning the second model, the increase of the parameter η causes the convergence to be reached, in an approximately way, at a lower speed but with a smoother behaviour.

Indeed, there is a shift from the pan angles computed with the centralized method which is caused by the presence of the noise. In the same way oscillations are related to the noise: if the measurements were not corrupted by it, these fluctuations would disappear. Doing manually the computations about the estimation process it results that the maximum amplitude value for the oscillations a_{max} is the maximum absolute value of the noise multiplied l times, where l stands for the number of links in the camera network. In formulas: $a_{max} = la$, if the noise is model with a uniform distribution in the interval $[-a, a]$.



(a) Cost function φ_R

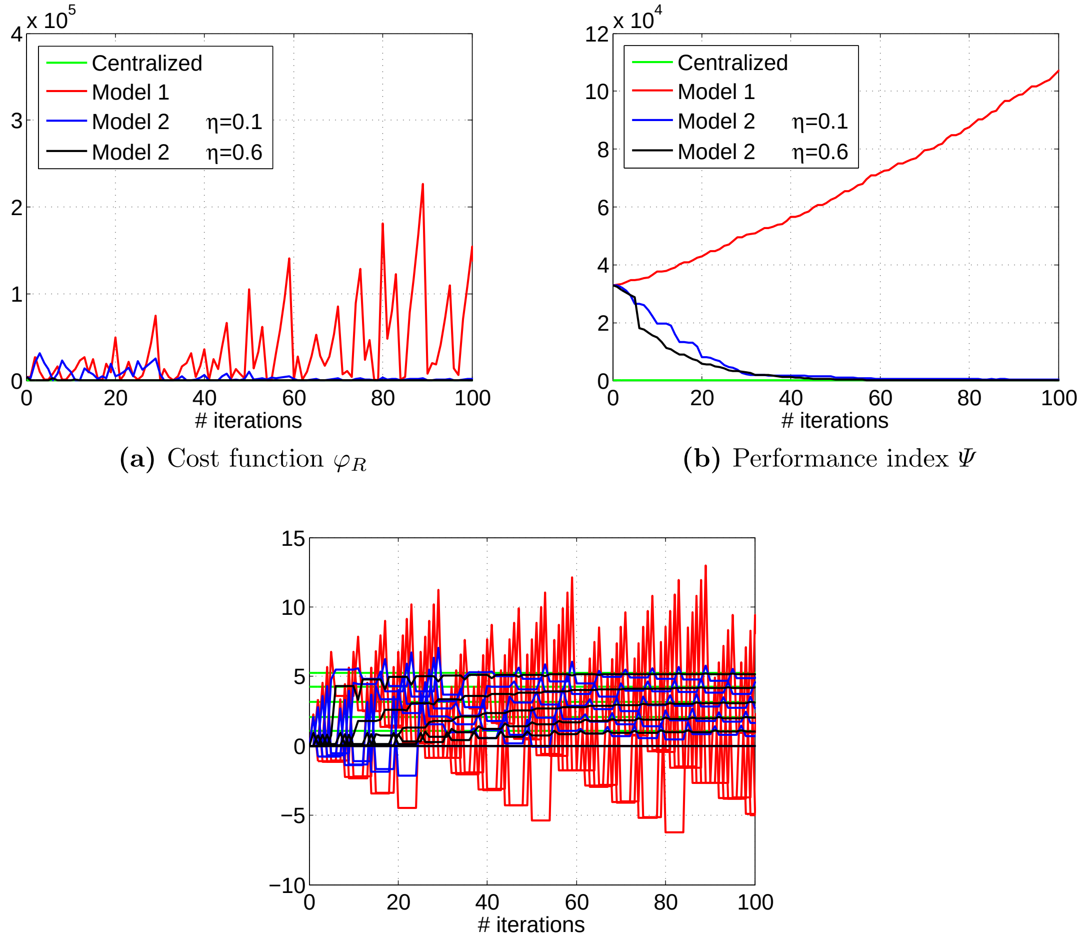
(b) Performance index Ψ

(c) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.2: Broadcast communication protocol: convergence analysis for a C_6^1 network.

The estimated angles have to be normalized with respect to a chosen reference frame, in this case the one referring to the first camera, consequently with respect to $\beta_1 = \widehat{\beta}_1 = 0$. Thus, while in a synchronous network the normalization is applied to all the angles taking part in the communication, in a generic protocol only cameras which updates their state at an instant have to be normalized. The problem of normalization becomes a relevant issue, in particular, with the symmetric gossip protocol, as it is reported in the next paragraph.

The **symmetric gossip** protocol does not work properly in the estimation of the pan angles with model 1. Figure 4.3 illustrates how the functional cost does not converge to zero and the increase of the error through the process.



(a) Cost function φ_R

(b) Performance index $\varPsi$

(c) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.3: Symmetric gossip communication protocol: convergence analysis for a C_6^1 network.

This behaviour can be in part justified by the nature of the protocol itself. In fact if the first couple of cameras exchange their state, at the next step a node of the successive pair forgets the estimate computed at the preceding instant in favour of the new acquired state. Because of this and due to the noise there is a growth of the drift through the process. Even if the initial conditions coincide with the true pan angles, *i.e.* $\boldsymbol{\beta}(0) = \boldsymbol{\beta}$, the estimation with this protocol does not give satisfying results (see Figure 4.4).

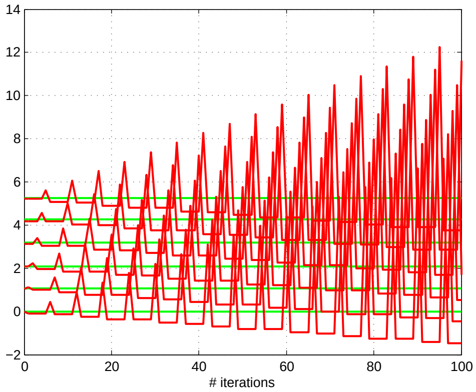


Figure 4.4: Estimated angles $\widehat{\beta}_i$ (rad) starting from initial condition $\boldsymbol{\beta}(0) = \boldsymbol{\beta}$: non-convergence of the estimated pan angles.

A better performance can be achieved using the second distributed model with $\eta = 0.6$, where the information of a camera itself is considered and it has a greater weight (Figure 4.5). In this way during the process a loss of information does not occur.

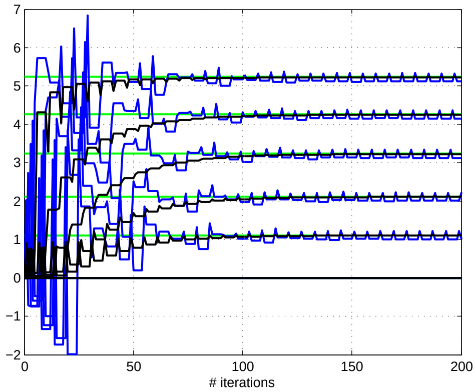
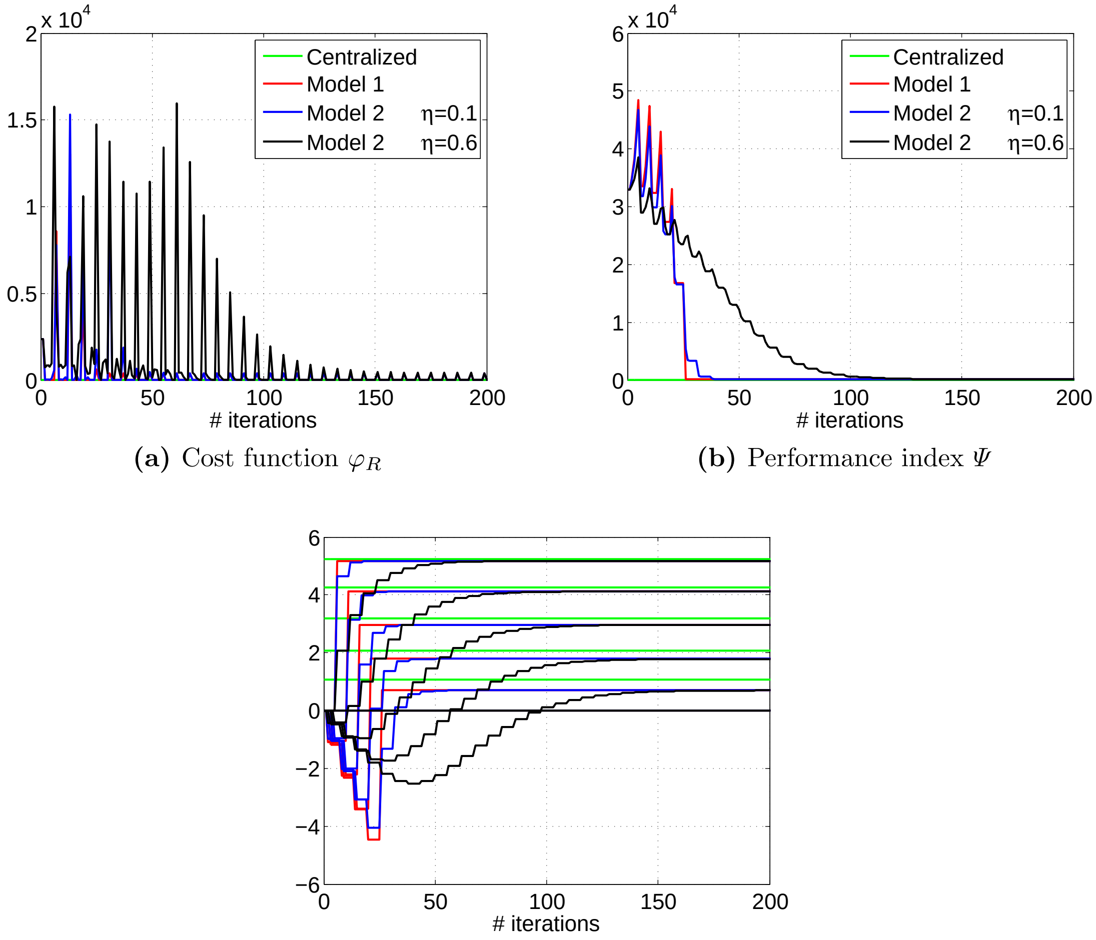


Figure 4.5: Estimated angles $\widehat{\beta}_i$ (rad): Convergence with the symmetric gossip protocol, model 2 $\eta = 0.1$ (blue line) - number of iterations increased to 200, model 2 $\eta = 0.6$ (black line).

Normalization is calculated as in the case of the broadcast protocol (**1**), though the structure of this communication suggests possible alternative methods to normalize the estimated angles in order to aim to convergence (subsection 4.1.1).

Similar considerations to the broadcast can be done for the **asymmetric gossip** protocol (Figure 4.6). During an initial phase the angles are underestimated and the values of the estimated angles settle at a lower rate than the broadcast protocol because every camera effects its update at an iteration exploiting the data of a single neighbour.
The idea behind the deterministic structure applied is that of performing a communication cycle among the cameras and to repeat this scheme through the process.



(a) Cost function φ_R

(b) Performance index Ψ

(c) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.6: Asymmetric gossip communication protocol: convergence analysis for a C_6^1 network.

Since the communication between every pair of nodes can take place in two directions, it is interesting that this kind of protocol performs differently according to the order of the communication (see Figure 4.7). This observation suggests that the estimation gives results that depends on which link is active and on the particular order of communication.

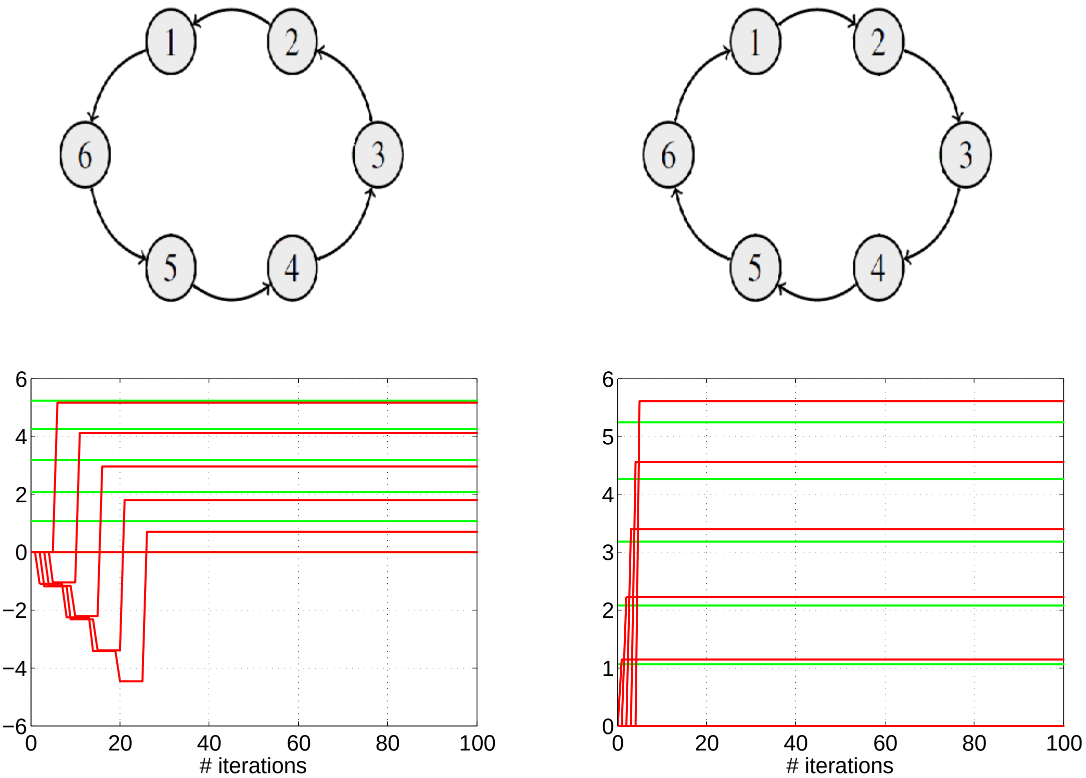
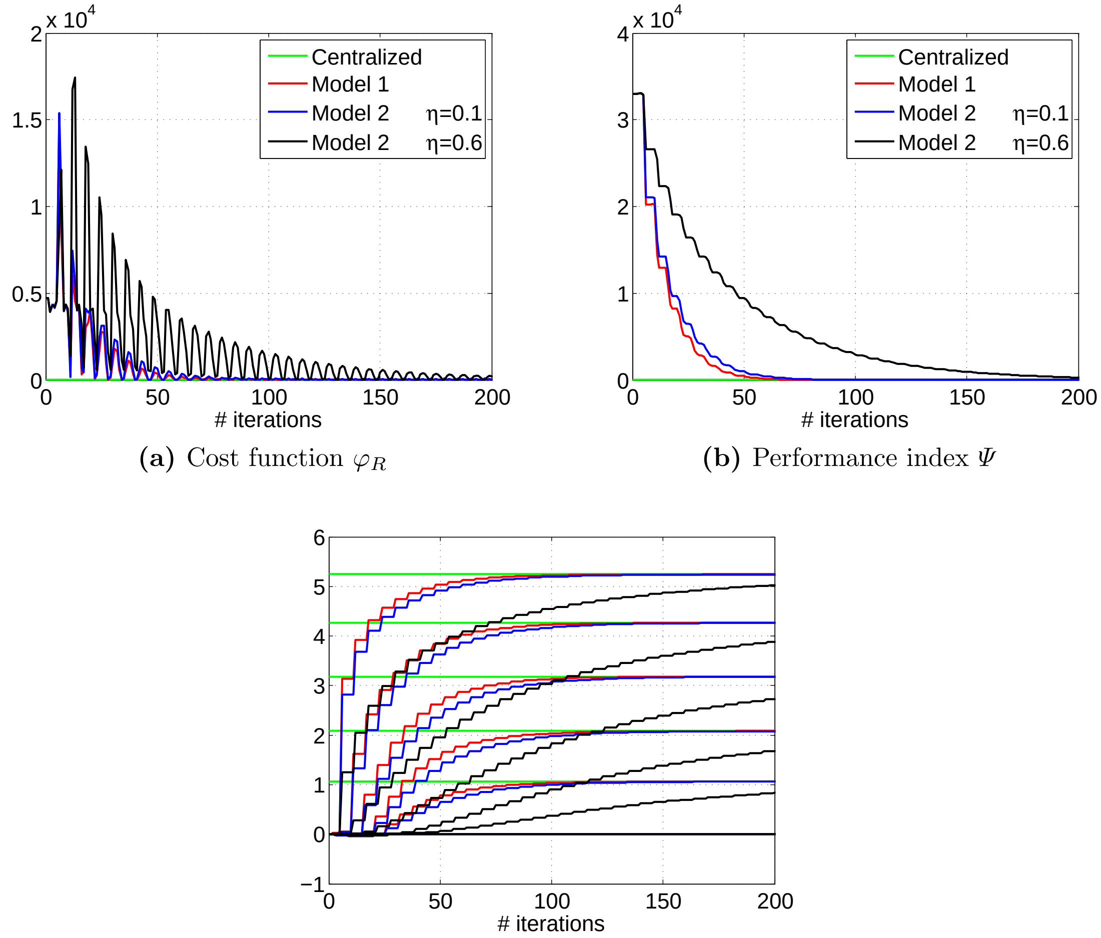


Figure 4.7: Estimated angles $\widehat{\beta}_i$ (rad): two possible deterministic versions of the asymmetric protocol (model 1).

As regards the **coordinated broadcast**, Figure 4.8 shows the comparison among the different distributed strategies.

This protocol differs from the other because every camera has information from all its neighbourhood and the combination of this, even if noisy, permits to reach the centralized value correctly, to the detriment of a lower speed. For the sake of completeness, estimation computed with model (3.22) is graphically represented, even though it is not very efficient.

Ending up this part of analysis of the protocols it is remarkable that the second distributed model is generally slower in converging than the first one, but it reduces the presence of oscillations.



(a) Cost function φ_R

(b) Performance index Ψ

(c) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.8: Coordinated broadcast communication protocol: convergence analysis for a C_6^1 network.

4.1.1 Methods of normalization

In this section it is considered how the normalization may affect the estimation, in the case that the initial estimate for all the cameras is the same. In the case of a symmetric gossip protocol the normalization adopted does not lead to the convergence to the centralized solution. This fact allows to think about other possible types of normalization which may resolve the problem.

A simpler configuration of the network, made up of $N = 4$ devices, is considered to develop the calculations of normalization (Figure 4.9).

Each communication edge is marked by a weight $w_i(k)$ belonging to the set $\mathcal{W} = \{\pm a, \pm b, \pm c, \pm d\}$ which represents the components of the input $\boldsymbol{u}(k) = \frac{1}{2}\Delta_{\mathscr{G}}^{-1}(k)\,\widetilde{\boldsymbol{\Delta\beta}}(k)$ (3.21).

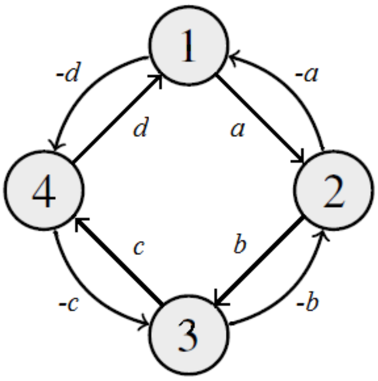


Figure 4.9: Cyclic weighted graph with $N = 4$.

The updating rule of the pan angles for the symmetric deterministic protocol at each instant is expressed as

$$\begin{cases} \beta_i\,(k+1) & = & \beta_j(k) + w_i(k) \\ \beta_j\,(k+1) & = & \beta_i(k) + w_j(k) \end{cases} \qquad \forall i, j = 1, ..., 4.$$

where j is the next camera with respect to the *i-th* node and $w_i(k) = -w_j(k)$.
The algorithm starts from the same random initial conditions, for instance $\boldsymbol{\beta}\,(0) = \mathbf{0}$ and at a certain iteration of the procedure the estimated angles are expected to be equal to the centralized solution.
This value, according to this specific representation, is

$$\boldsymbol{\beta}^* = [-a - b - c - d \quad -a \quad -a - b \quad -a - b - c]^T$$

where $-\,(a + b + c + d) = 0$. Indeed, since the measurements taken are noisy, this last expression in the simulations becomes

$$-\,(a + b + c + d) \simeq 0 \tag{4.1}$$

Based on these assumptions, besides of normalizing with respect to the first camera (**1**), the normalization can be applied with respect to the neighbouring device (**2**). In this way the computed estimates should be consistent with the orientation of the previous camera.

it	0	1	2	3	4	5	6	7	8	9	10	11	12	...
$\widehat{\beta}_1$	0	0	0	0	0	0	0	0	0	0	0	0	0	
$\widehat{\beta}_2$	0	$-2a$	b	b	b	$-2a-b$	$b+c$	$b+c$	$b+c$	$-2a-b-c$	$b+c$	$b+c$	$b+c$	
$\widehat{\beta}_3$	0	0	$-2\,(a+b)$	c	c	c	$-2a-3b-c$	c	c	c	$-2\,(a+c)-3b$	$b+2c$	$b+2c$	
$\widehat{\beta}_4$	0	0	0	$-2\,(a+b+c)$	0	0	0	$-2a-3\,(b+c)$	$b+c$	$b+c$	$b+c$	$-2a-4b-5c$	$2b+3c$	
		$1 \leftrightarrows 2$	$2 \leftrightarrows 3$	$3 \leftrightarrows 4$	$4 \leftrightarrows 1$	$1 \leftrightarrows 2$	$2 \leftrightarrows 3$	$3 \leftrightarrows 4$	$4 \leftrightarrows 1$	$1 \leftrightarrows 2$	$2 \leftrightarrows 3$	$3 \leftrightarrows 4$	$4 \leftrightarrows 1$	...

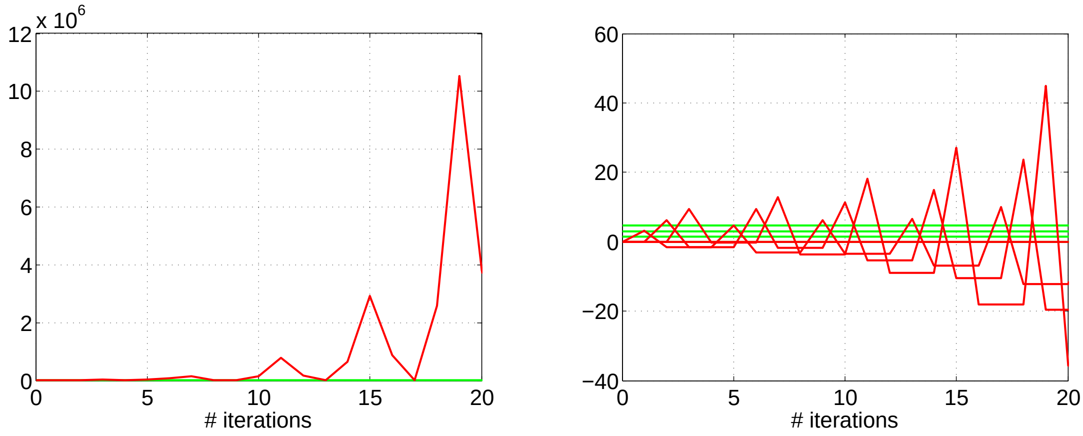
Table 4.1: Development of the estimation procedure with normalization (**2**).

In the following normalization (**3**) the idea is quite similar to (**2**): a camera is normalized with respect to the other one in the communicating couple but this time the normalization about the first angle is not carried out.

it	0	1	2	3	4	5	6	7	8	9	10	11	12	...
$\widehat{\beta}_1$	0	a	a	a	$-a$	$a+b$	$a+b$	$a+b$	$-2a-b$	$a+b+c$	$a+b+c$	$a+b+c$	$-3a-2b-c$	
$\widehat{\beta}_2$	0	$-2a$	b	b	b	$-3a-b$	$b+c$	$b+c$	$b+c$	$-4a-2b-c$	0	0	0	
$\widehat{\beta}_3$	0	0	$-2(a+b)$	c	c	c	$-3(a+b)-c$	$a+c+d$	$a+c+d$	$a+c+d$	$-4a-3b-c$	0	0	
$\widehat{\beta}_4$	0	0	0	$-2(a+b+c)$	$a+d$	$a+d$	0	$-3a-2(b+c)$	$a+b+d$	$a+b+d$	$a+b+d$	$-4a-3b-2c$	0	
		$1 \leftrightarrows 2$	$2 \leftrightarrows 3$	$3 \leftrightarrows 4$	$4 \leftrightarrows 1$	$1 \leftrightarrows 2$	$2 \leftrightarrows 3$	$3 \leftrightarrows 4$	$4 \leftrightarrows 1$	$1 \leftrightarrows 2$	$2 \leftrightarrows 3$	$3 \leftrightarrows 4$	$4 \leftrightarrows 1$	...

Table 4.2: Development of the estimation procedure with normalization (**3**).

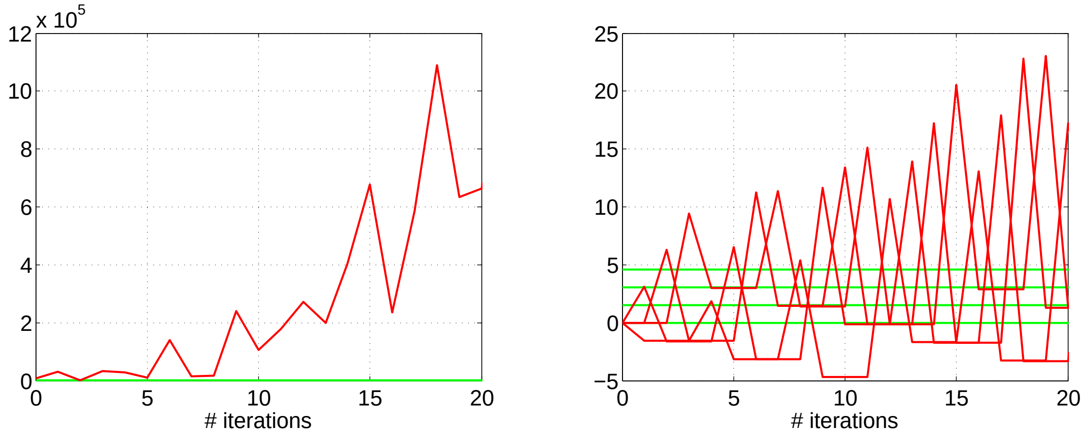
Both these types of normalization are quite reasonable from a theoretical point of view yet they do not improve the the performance of the distributed model 1 and the functional cost diverges (Figure 4.10, 4.11).



(a) Cost function φ_R

(b) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.10: $N = 4$. Normalization strategy (**2**), zoom on the first 20 iterations.



(a) Cost function φ_R

(b) Estimated angles $\widehat{\beta}_i$ (rad)

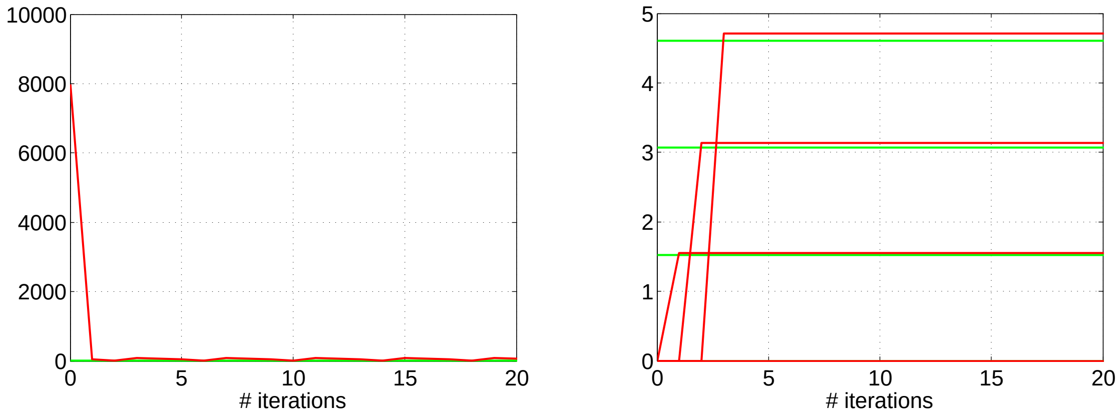
Figure 4.11: $N = 4$. Normalization strategy (**3**), zoom on the first 20 iterations.

Another strategy proposed (**4**) is a sort of normalization with respect to the estimation for a node computed at the preceding step of the process. More in detail when two cameras update their state the first camera of the couple in the order of the cycle keeps the estimate calculated at the previous step as shown in Table 4.3. Since effectively only one update of a node is relevant, the nature of the symmetric gossip is compromised: the protocol has an asymmetric structure. Moreover, since 4.1 holds, when the first device is involved, it is necessary to apply a further normalization to maintain its associated angle constantly equals to 0.

it	0	1		2		3	
$\widehat{\beta}_1$	0	a	0	0	0	0	0
$\widehat{\beta}_2$	0	$-a$	$-a$	b	$-a$	$-a$	$-a$
$\widehat{\beta}_3$	0	0	0	$-a-b$	$-a-b$	c	$-a-b$
$\widehat{\beta}_4$	0	0	0	0	0	$-a-b-c$	$-a-b-c$
		$1 \leftrightarrows 2$		$2 \leftrightarrows 3$		$3 \leftrightarrows 4$	

it	4		5		
$\widehat{\beta}_1$	$-a-b-c-d=0$	0	$-a+a=0$	0	$= \boldsymbol{\beta}^*$
$\widehat{\beta}_2$	$-a$	$-a$	$-a$	$-a$	
$\widehat{\beta}_3$	$-a-b$	$-a-b$	$-a-b$	$-a-b$	
$\widehat{\beta}_4$	d	$-a-b-c$	$-a-b-c$	$-a-b-c$	
	$4 \leftrightarrows 1$		$1 \leftrightarrows 2$		

Table 4.3: Development of the estimation procedure with normalization (**4**).



(a) Cost function φ_R

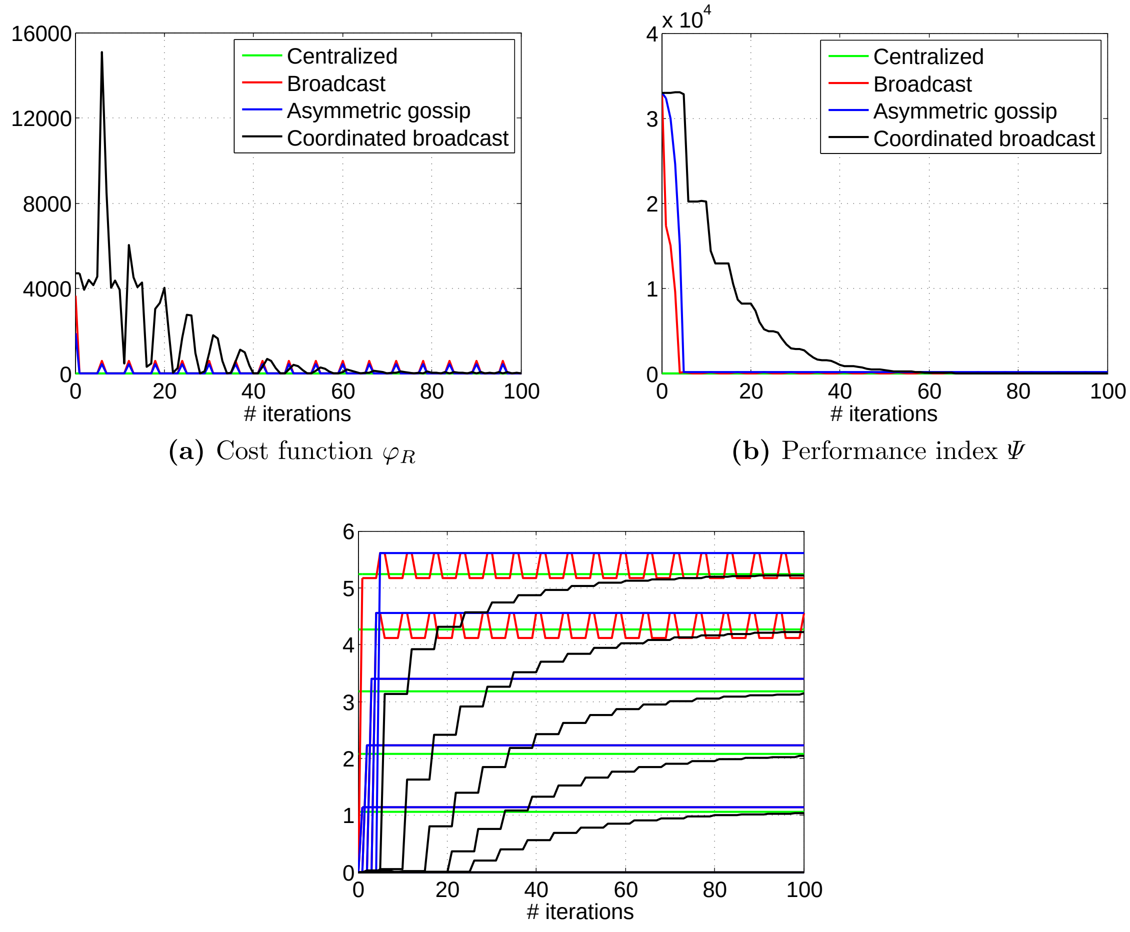
(b) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.12: $N = 4$. Normalization strategy (**4**), zoom on the first 20 iterations.

To sum up, it is difficult to deal with a symmetric gossip communication protocol because several problems arise due to the nature of the protocol itself. The owned noisy measurements are relative and this implies the necessity to fix a reference framework. Regardless of the type of normalization, the reference frame cannot be determined. Probably all these matters can be faced off adapting the state-space model (3.19) from a synchronous protocol to the symmetric gossip in a different way.

4.2 Comparison among protocols

As a conclusion for this section, a comparison among all the analyzed protocols is provided with the goal to find the pros and cons of each of them and eventually to establish a best communication solution for a camera network. The symmetric gossip protocol is neglected because of the problems described before.



(a) Cost function φ_R

(b) Performance index Ψ

(c) Estimated angles $\widehat{\beta}_i$ (rad)

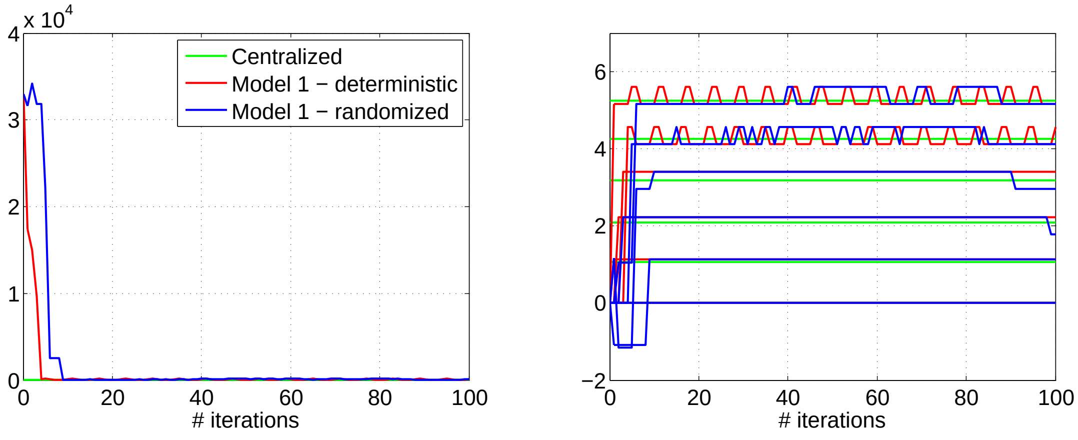
Figure 4.13: State-space model 1: comparison among protocols for a C_6^1 network.

Figure 4.13 highlights some interesting facts. Though broadcast and asymmetric gossip are relatively fast to achieve a neighbourhood of the centralized value they do not reach it precisely. Moreover estimated pan angles either oscillates or however have a significant error from the ground-truth.

On the other hand the coordinated broadcast protocol requires several iterations to converge but can do it exactly.
If an estimate of the error measurements can be found and an averaging operation can be performed then the first two protocols are preferable, while if the accuracy on the orientation of the cameras is not to be set aside, the coordinated broadcast is the best solution. This latter one prevails above the others also when a network with a large number of cameras have to be managed, case where, using the first two protocols, the estimated pan angles may misunderstand the real rotations of the cameras due to non-exact convergence.

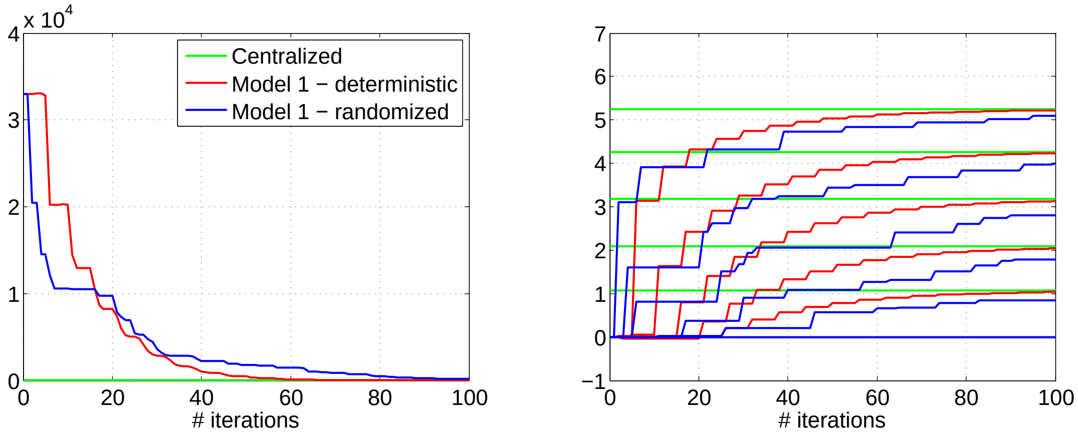
4.3 Deterministic and randomized versions

Up to now the simulations were made taking into consideration the deterministic version of the protocols. The following study examines the case where communication among the devices is randomized.
Intuitively, if the exchange of information is not arranged, in order to get close progressively to the centralized value, it is required that all the communication links between two nodes of a cyclic network have to be operative at least one time in the randomized procedure. Then it is straightforward to imagine that the speed of convergence in the randomized situation may be reduced with respect to the deterministic case (Figure 4.14 and 4.15). Similarly the mean error should be higher.



(a) Performance index Ψ **(b)** Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.14: Deterministic and randomized versions of the broadcast protocol.



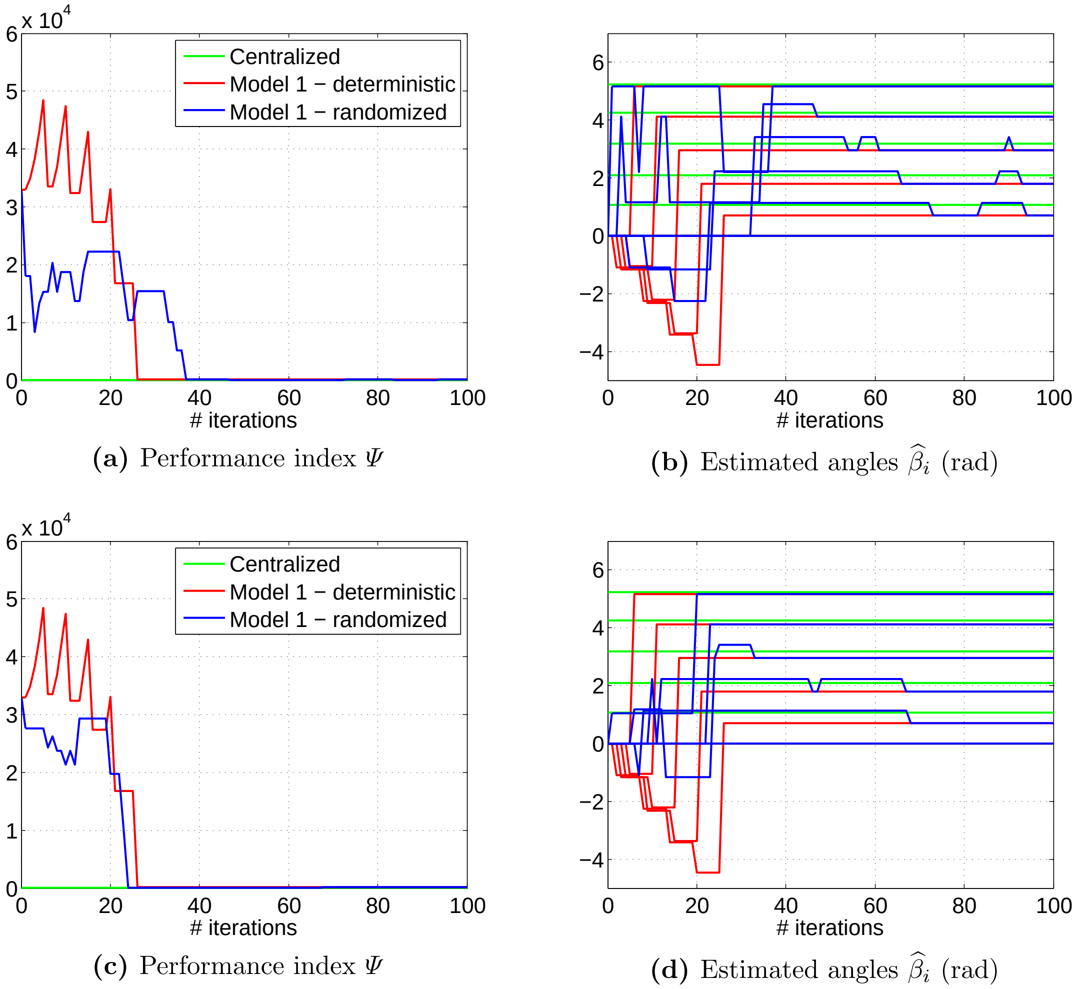
(a) Performance index Ψ

(b) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.15: Deterministic and randomized versions of the coordinated broadcast protocol.

For the asymmetric gossip the counter-clockwise deterministic case is considered as reference.

In Figure 4.16b some pan angles are underrated or overrated because of the lack of information of some nodes or maybe it can happen that a communication between two cameras is repeated into subsequent instants adding no new upgrade of the estimate. Figure 4.16d illustrates a less frequent situation where the randomized version of the protocol converges more quickly. This can be justified partially by the fact that in the randomized version a link between two cameras can become active in two possible directions of communication while in the deterministic case the set up has a fixed direction. In some sense the randomized process covers a wider communication network.



(a) Performance index Ψ

(b) Estimated angles $\widehat{\beta}_i$ (rad)

(c) Performance index Ψ

(d) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 4.16: Comparison between the deterministic and randomized versions of the asymmetric gossip protocol.

Chapter 5

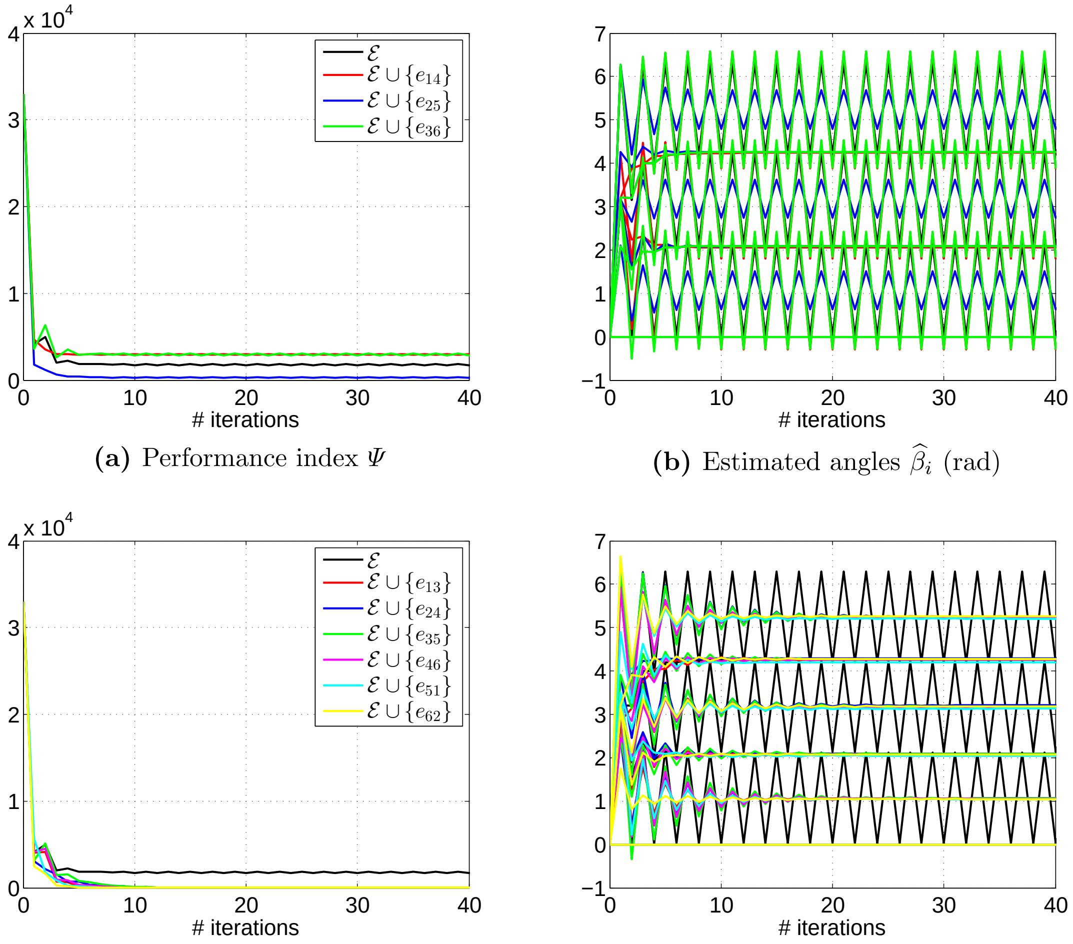
Network Topology

This chapter is concerned with the modification of the network topology. The term *topology* refers to a schematic representation of network nodes arrangement including the physical and communication links among them.
Topology variations may entail advantages or difficulties in completing the task of localization efficiently. For example if the network is fully connected some information may be redundant especially if the number of devices is high. On the other hand if the examined topology is that of a tree, loss of information may occur if a node does not work properly.
For that reason, in following sections the distributed models proposed are tested adding edges or changing the configuration of the network.

5.1 Addition of an edge

In the previous chapter the camera network had a fixed cyclic configuration. As already mentioned, this is a common structure for a camera network since each device communicates with its two nearest neighbours, which possibly share overlapping views with.
When the synchronous protocol is considered, the application of the first state-space model to a bipartite graph involves the presence of the eigenvalue $\lambda = -1$, which provokes the oscillations of the estimated angles around the equilibrium. In order to avoid this undesired behaviour the configuration of the network is now alterated: the addition of a specific link makes the graph to become non-bipartite thus leading to convergence. In the following simulations the network C_6^1 represented in Figure 2.4 is taken as reference for the addition of links.

The theoretical findings are demonstrated by simulations in Figure 5.1: adding any link such that the graph remains bipartite, some of the estimated pan angles oscillate around the equilibrium state, without converging. On the other hand if the additional connection between two cameras generates at least one cycle with an odd number of nodes, *i.e.* the graph becomes non-bipartite, the convergence is now ensured.



(a) Performance index Ψ

(b) Estimated angles $\widehat{\beta}_i$ (rad)

(c) Performance index Ψ

(d) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 5.1: Model (3.19): convergence analysis for a cyclic network C_6^1 with an extra link. Top row: bipartite graph. Bottom row: non-bipartite graph.

Figure 5.1 hints at the fact that the addition to the network of any edge belonging to the same set $\mathcal{E}_1$ or $\mathcal{E}_2$, *i.e.*

$$\begin{aligned}\mathcal{E}_1 &= \{e_{14}, e_{25}, e_{36}\}\\ \mathcal{E}_2 &= \{e_{13}, e_{24}, e_{35}, e_{46}, e_{51}, e_{62}\}\end{aligned}$$

may give the same convergence properties in the angle estimation. This latter statement can be proved checking the eigenvalues of the matrix F.

In particular, being F a consensus matrix, the convergence speed is regulated by the absolute value of its second largest eigenvalue λ_d, which stands for the dominant mode of the system, namely the slowest one. Thus it can be drawn a distinction among the

added links for which λ_d results the smallest. Indeed the more λ_d is close to 0 the faster the convergence is reached and contrariwise if it tends to 1.
Clearly this kind of reasoning is pointless for the first set of links, as there is not convergence. Referring to $\mathcal{E}_2$, for each edge it holds $\lambda_d = -0,8259$ (see Figure 5.2).

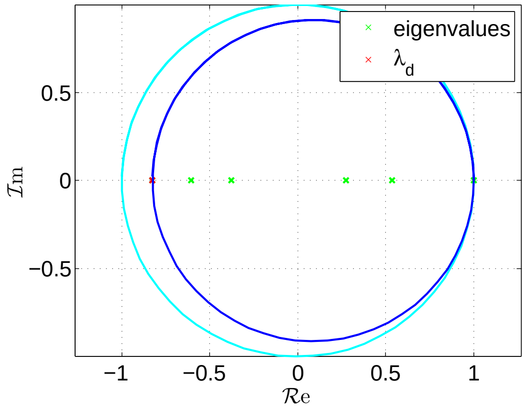


Figure 5.2: Eigenvalues of F with an extra link belonging to $\mathcal{E}_2$. Blue circle delimits the region where eigenvalues smaller than the second largest one lie.

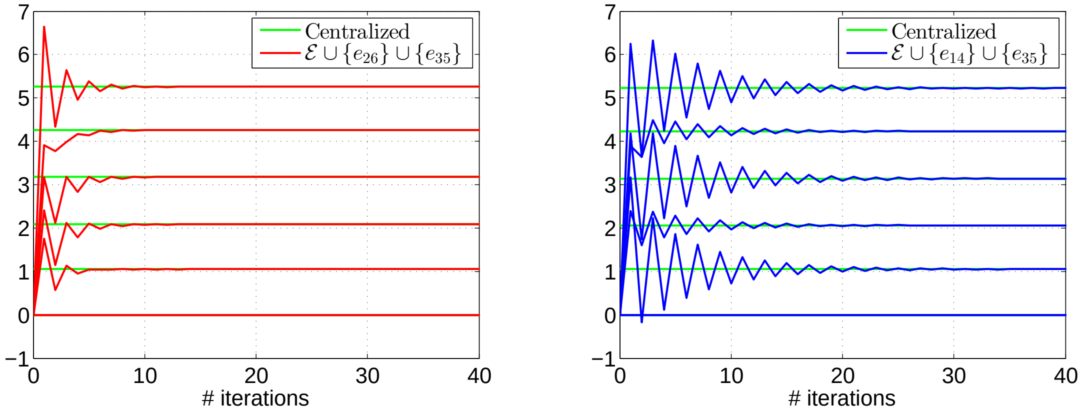
The difference between those edges which divide the network in a half keeping the graph bipartite (set $\mathcal{E}_1$) and the remaining ones which make the graph non-bipartite (set $\mathcal{E}_2$) comes to light also in the application of the second distributed model (3.22). The presence of an additional edge in a cyclic graph implies the modification of the spectrum of the state-space matrix: a closed-form formula is not known. This assertion means that the control parameter η_{cr}, which determines the performance of convergence, cannot be computed directly but it is only possible to approximate it through experimental trials.
The objective is to find this threshold in order to make the convergence be led by the second largest eigenvalue which has to be negative, otherwise the estimate may overestimate or underestimate the asymptotic solution.
Observing values of λ_d in simulations with different η, $\eta_{cr,1}$ in the case of $\mathcal{E}_1$ is found to be $\simeq 0.2$, while for $\mathcal{E}_2$, $\eta_{cr,2}$ belongs to the interval $(0.126, 0.127)$.

5.2 Addition of two edges

The following analysis regards the addition of two links to the cyclic graph.
The evaluation of the performance of the two distributed models is made focusing the second largest eigenvalue of the matrix F and F_1.
The table below 5.1 collects the values of λ_d for the model (3.19) depending on the particular extra pair of edges.

Edges	$(1,4)$	$(1,5)$	$(2,4)$	$(2,5)$	$(2,6)$	$(3,5)$	$(3,6)$	$(4,6)$
$(1,3)$	-0,8356	-0,7287	-0,7287	-0,839	-0,7287	-0,7287	-0,8356	-0,6667
$(1,4)$		-0,8356	-0,8356	x	-0,839	$\underline{-0,839}$	x	-0,8356
$(1,5)$			-0,6667	-0,8356	-0,7287	-0,7287	-0,839	-0,7287
$(2,4)$				-0,8356	-0,7287	-0,7287	-0,839	-0,7287
$(2,5)$					-0,8356	-0,8356	x	-0,839
$(2,6)$						$\underline{-0,6667}$	-0,8356	-0,7287
$(3,5)$							-0,8356	-0,7287
$(3,6)$								-0,8356

Table 5.1: C_6^1+2 edges: values of the second largest eigenvalue for model (3.19). Underlined eigenvalues relate to the example reported in Figure 5.3.



(a) **(b)**

Figure 5.3: Speed of convergence of model 1 with two couples of added links to C_6^1: in this example the difference is clear, tough it is not appropriate to point out a time instant of convergence, since there are always deviations from the true value.

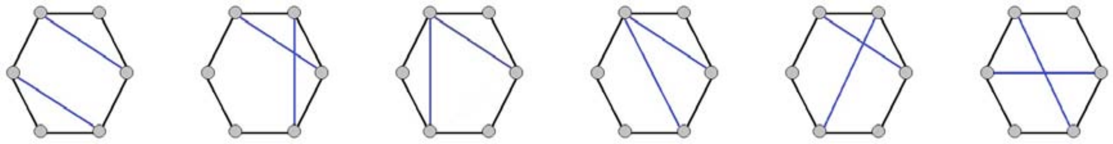
Obviously, if the additional couple is made up of two links belonging to the set $\mathcal{E}_1$, the graph remains bipartite and the localization issue cannot be fulfilled. Further considerations can be drawn from this table but they are postponed to the next paragraph.

Tables (from 5.3 to 5.7) report data about the second largest eigenvalue when the model (3.22) is applied, varying the tuning parameter η. Several interesting observations can be made.

- Couples of two links for which the second largest eigenvalue is the same can be grouped in sets, as shown in (5.1). Generally a fixed set of "fastest" links cannot be established since the modification of η influences the computation of the spectrum of F_1.

$$\begin{aligned}
\mathcal{E}_1 &= \{(e_{13}, e_{46}), (e_{15}, e_{24}), (e_{26}, e_{35})\} \\
\mathcal{E}_2 &= \{(e_{13}, e_{24}), (e_{13}, e_{26}), (e_{15}, e_{26}), (e_{15}, e_{46}), (e_{24}, e_{35}), (e_{35}, e_{46})\} \\
\mathcal{E}_3 &= \{(e_{13}, e_{15}), (e_{13}, e_{35}), (e_{15}, e_{35}), (e_{24}, e_{26}), (e_{24}, e_{46}), (e_{26}, e_{46})\} \\
\mathcal{E}_4 &= \{(e_{13}, e_{14}), (e_{13}, e_{36}), (e_{14}, e_{15}), (e_{14}, e_{24}), (e_{14}, e_{46}), (e_{15}, e_{25}), (e_{24}, e_{25}), \\
&\quad (e_{25}, e_{26}), (e_{25}, e_{35}), (e_{26}, e_{36}), (e_{35}, e_{36}), (e_{36}, e_{46})\} \\
\mathcal{E}_5 &= \{(e_{13}, e_{25}), (e_{14}, e_{26}), (e_{14}, e_{35}), (e_{15}, e_{36}), (e_{24}, e_{36}), (e_{25}, e_{46})\} \\
\mathcal{E}_6 &= \{(e_{14}, e_{25}), (e_{14}, e_{36}), (e_{25}, e_{36})\}
\end{aligned} \tag{5.1}$$

- Giving a graphical representation to the partitions $\mathcal{E}_i$ with $i = 1, \ldots, 6$ characterising configurations emerge, regardless of the cyclic numeration of cameras in the network (Figure 5.4).



(a) $\mathcal{E}_1$ (b) $\mathcal{E}_2$ (c) $\mathcal{E}_3$ (d) $\mathcal{E}_4$ (e) $\mathcal{E}_5$ (f) $\mathcal{E}_6$

Figure 5.4: Configurations for the identified sets of two edges.

Thanks to simulations, a range of possible values for η_{cr} is computed when an edge belonging to a specific set is added to the network. These intervals are ad scripted in Table 5.2.

Edge Set	η_{cr}
$\mathcal{E}_1$	$(0.0425, 0.043)$
$\mathcal{E}_2$	$(0.115, 0.13)$
$\mathcal{E}_3$	$(0.13, 0.14)$
$\mathcal{E}_4$	$(0.14, 0.15)$
$\mathcal{E}_5$	$(0.2, 0.225)$
$\mathcal{E}_6$	$(0.25, 0.26)$

Table 5.2: Belonging range of η_{cr} for each set of edges.

This type of experimental analysis is envisaged about the addition of a *k-th* set of edges till the saturation of the graph structure. Furthermore, all these results may be the starting point for a future analytical analysis.

- The simulations performed changing the value of the control parameter η, besides finding the belonging range of the critical value for each set of edges, allow to

identify a common global value of the $\eta_c r$. for any added couple of links. This value is found to be approximately $\eta_{cr} \simeq 0.04$. Then, if $\eta <= \eta_{cr}$ the second largest eigenvalue will be negative implying the presence of an oscillatory transient period towards the equilibrium in the estimation of the angles.
This critical value may be inferred analytically finding a bound on the second largest positive eigenvalue of the system.

Edges	$(1,4)$	$(1,5)$	$(2,4)$	$(2,5)$	$(2,6)$	$(3,5)$	$(3,6)$	$(4,6)$
$(1,3)$	-0.8173	-0.7114	-0.7114	-0.8206	-0.7114	-0.7114	-0.8173	-0.65
$(1,4)$		-0.8173	-0.8173	-0.98	-0.8206	-0.8206	-0.98	-0.8173
$(1,5)$			-0.65	-0.8173	-0.7114	-0.7114	-0.8206	-0.7114
$(2,4)$				-0.8173	-0.7114	-0.7114	-0.8206	-0.7114
$(2,5)$					-0.8173	-0.8173	-0.98	-0.8206
$(2,6)$						-0.65	-0.8173	-0.7114
$(3,5)$							-0.8173	-0.7114
$(3,6)$								-0.8173

Table 5.3: $C_6^1 + 2$ links: values of the second largest eigenvalue for $\eta = 0.01$.

Edges	$(1,4)$	$(1,5)$	$(2,4)$	$(2,5)$	$(2,6)$	$(3,5)$	$(3,6)$	$(4,6)$
$(1,3)$	-0.7622	-0.6596	-0.6596	-0.7654	-0.6596	-0.6596	-0.7622	-0.6
$(1,4)$		-0.7622	-0.7622	-0.92	-0.7654	-0.7654	-0.92	-0.7622
$(1,5)$			-0.6	-0.7622	-0.6596	-0.6596	-0.7654	-0.6596
$(2,4)$				-0.7622	-0.6596	-0.6596	-0.7654	-0.6596
$(2,5)$					-0.7622	-0.7622	-0.92	-0.7654
$(2,6)$						-0.6	-0.7622	-0.6596
$(3,5)$							-0.7622	-0.6596
$(3,6)$								-0.7622

Table 5.4: $C_6^1 + 2$ links: values of the second largest eigenvalue for $\eta = 0.04$.

Edges	$(1,4)$	$(1,5)$	$(2,4)$	$(2,5)$	$(2,6)$	$(3,5)$	$(3,6)$	$(4,6)$
$(1,3)$	-0.6521	-0.5558	-0.5558	-0.6551	-0.5558	-0.5558	-0.6521	0.6196
$(1,4)$		-0.6521	-0.6521	-0.8	-0.6551	-0.6551	-0.8	-0.6521
$(1,5)$			0.6196	-0.6521	-0.5558	-0.5558	-0.6551	-0.5558
$(2,4)$				-0.6521	-0.5558	-0.5558	-0.6551	-0.5558
$(2,5)$					-0.6521	-0.6521	-0.8	-0.6551
$(2,6)$						0.6196	-0.6521	-0.5558
$(3,5)$							-0.6521	-0.5558
$(3,6)$								-0.6521

Table 5.5: $C_6^1 + 2$ links: values of the second largest eigenvalue for $\eta = 0.1$.

Edges	$(1,4)$	$(1,5)$	$(2,4)$	$(2,5)$	$(2,6)$	$(3,5)$	$(3,6)$	$(4,6)$
$(1,3)$	0.5901	0.5266	0.5602	-0.4712	0.5602	0.5266	0.5901	0.6619
$(1,4)$		0.5901	0.5901	-0.6	-0.4712	-0.4712	-0.6	0.5901
$(1,5)$			0.6619	0.5901	0.5602	0.5266	-0.4712	0.5602
$(2,4)$				0.5901	0.5266	0.5602	-0.4712	0.5266
$(2,5)$					0.5901	0.5901	-0.6	-0.4712
$(2,6)$						0.6619	0.5901	0.5266
$(3,5)$							0.5901	0.5602
$(3,6)$								0.5901

Table 5.6: $C_6^1 + 2$ links: values of the second largest eigenvalue for $\eta = 0.2$.

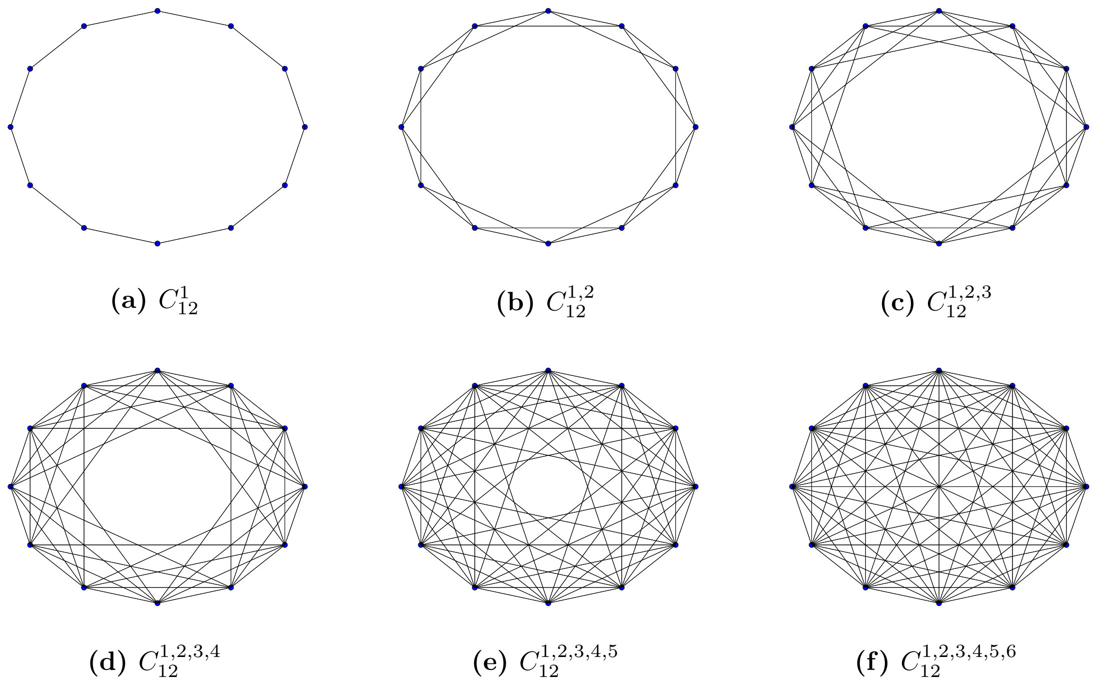
Edges	$(1,4)$	$(1,5)$	$(2,4)$	$(2,5)$	$(2,6)$	$(3,5)$	$(3,6)$	$(4,6)$
$(1,3)$	0.795	0.7633	0.7801	0.7097	0.7801	0.7633	0.795	0.8309
$(1,4)$		0.795	0.795	0.7333	0.7097	0.7097	0.7333	0.795
$(1,5)$			0.8309	0.795	0.7801	0.7633	0.7097	0.7801
$(2,4)$				0.795	0.7633	0.7801	0.7097	0.7633
$(2,5)$					0.795	0.795	0.7333	0.7097
$(2,6)$						0.8309	0.795	0.7633
$(3,5)$							0.795	0.7801
$(3,6)$								0.795

Table 5.7: $C_6^1 + 2$ links: values of the second largest eigenvalue for $\eta = 0.6$.

5.3 Regular graphs

This section is devoted to the evaluation of some regular graphs (defined in section 2.2).
The goal is to understand if the addition of several edges to the network can imply a deterioration of the estimates: additional links correspond to an increment of information within the estimation procedure, but also enhance noisy measurements.

A cyclic graph of $N = 12$ devices is considered. From this configuration, the topology is progressively enriched adding to each node communication links with the next two nearest neighbours. In this way the degree of each agent increases of 2 for every graph generated till the completion of the network, where each camera shares information with all the remaining ones.
These graphs are called *circulant graphs*: they are undirected graphs which have a cyclic group of symmetries that includes a symmetry taking any vertex to any other vertex. Equivalently a graph is said to be circulant if it is described by a circulant matrix.
A useful writing to identify a circulant graph is $C_N^{s_1,\dots,s_k}$ with hops $s_1, \dots, s_k$ (*i.e.* the distance between a node and a neighbour), where N is the cardinality of the graph and each node i is adjacent to $2k$ vertices; in other words neighbours of i belong to the set $\{i \pm s_1, \dots, i \pm s_k \mod N\}$.
Figure 5.5 describes the k regular graphs just mentioned.



(a) C_{12}^{1}

(b) $C_{12}^{1,2}$

(c) $C_{12}^{1,2,3}$

(d) $C_{12}^{1,2,3,4}$

(e) $C_{12}^{1,2,3,4,5}$

(f) $C_{12}^{1,2,3,4,5,6}$

Figure 5.5: Graphical representation of the k regular graphs under consideration.

To check the potential degradation of the pan angles $\hat{\beta}_i$ an indicative measure is the mean square error (3.25) computed at the end of the process. In this case the number of iterations is set to $it = 1000$ (Table 6.1).

	C_{12}^{1}	$C_{12}^{1,2}$	$C_{12}^{1,2,3}$	$C_{12}^{1,2,3,4}$	$C_{12}^{1,2,3,4,5}$	$C_{12}^{1,2,3,4,5,6}$
$\Psi_1\,(it)$	439.4565	1.5788	0.8248	0.305	0.2186	0.1969
$\Psi_2\,(it)$	494.7419	0.8194	0.6853	0.2763	0.2704	0.5519
$\Psi_3\,(it)$	354.9625	7.0564	2.5715	0.2848	0.2426	0.3104
$\Psi_4\,(it)$	419.9193	2.8131	0.8558	0.9264	0.75	1.2884
$\Psi_5\,(it)$	444.2805	3.5327	1.6325	0.3675	0.524	0.3691

Table 5.8: Values of $\Psi\,(it)$ for five simulations with different initial noisy measurements.

In these simulations the noise is limited to the interval $[-5, 5]$ degrees. Naturally, the first column contains high values because there is not convergence.
From the above Table 6.1 it can be deduced that expanding the communication range among nodes, estimations generally improve their accuracy with respect to the real pan angles. An evidence of improvement of the estimate is shown in Figure 5.6. In some cases the addition of edges can lead up to a greater error than the one relating to a network with a smaller number of links, as it was supposed. The non-monotonic behaviour of the error, as in $\Psi_4\,(it)$, is due to the randomic distribution of the noise on the measurements.

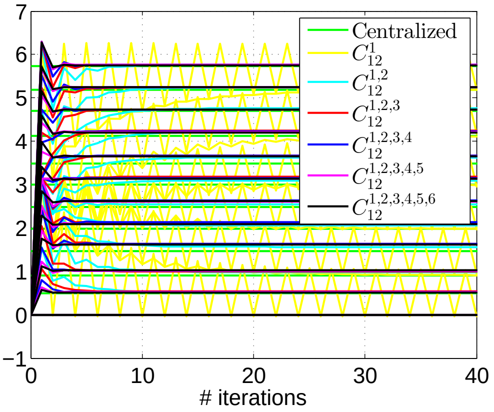
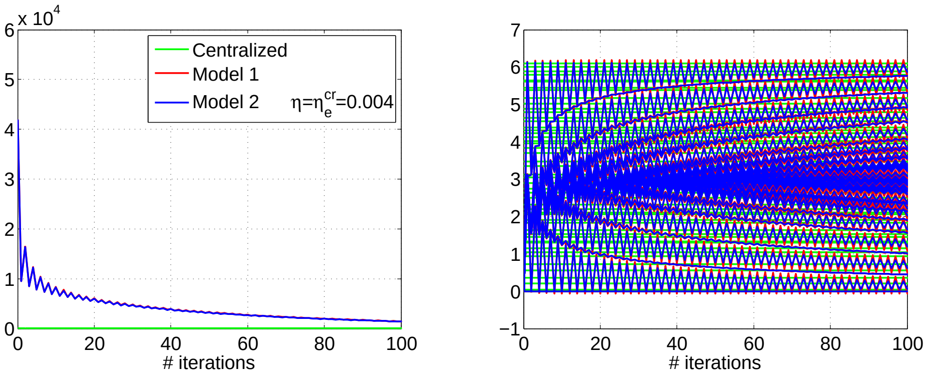


Figure 5.6: Estimated angles $\widehat{\beta}_i$ (rad) for the circulant graphs analyzed.

5.4 Large scale networks

In this section a large scale network made up of many devices is considered. The results provided are tested and validated in a synthetic framework.

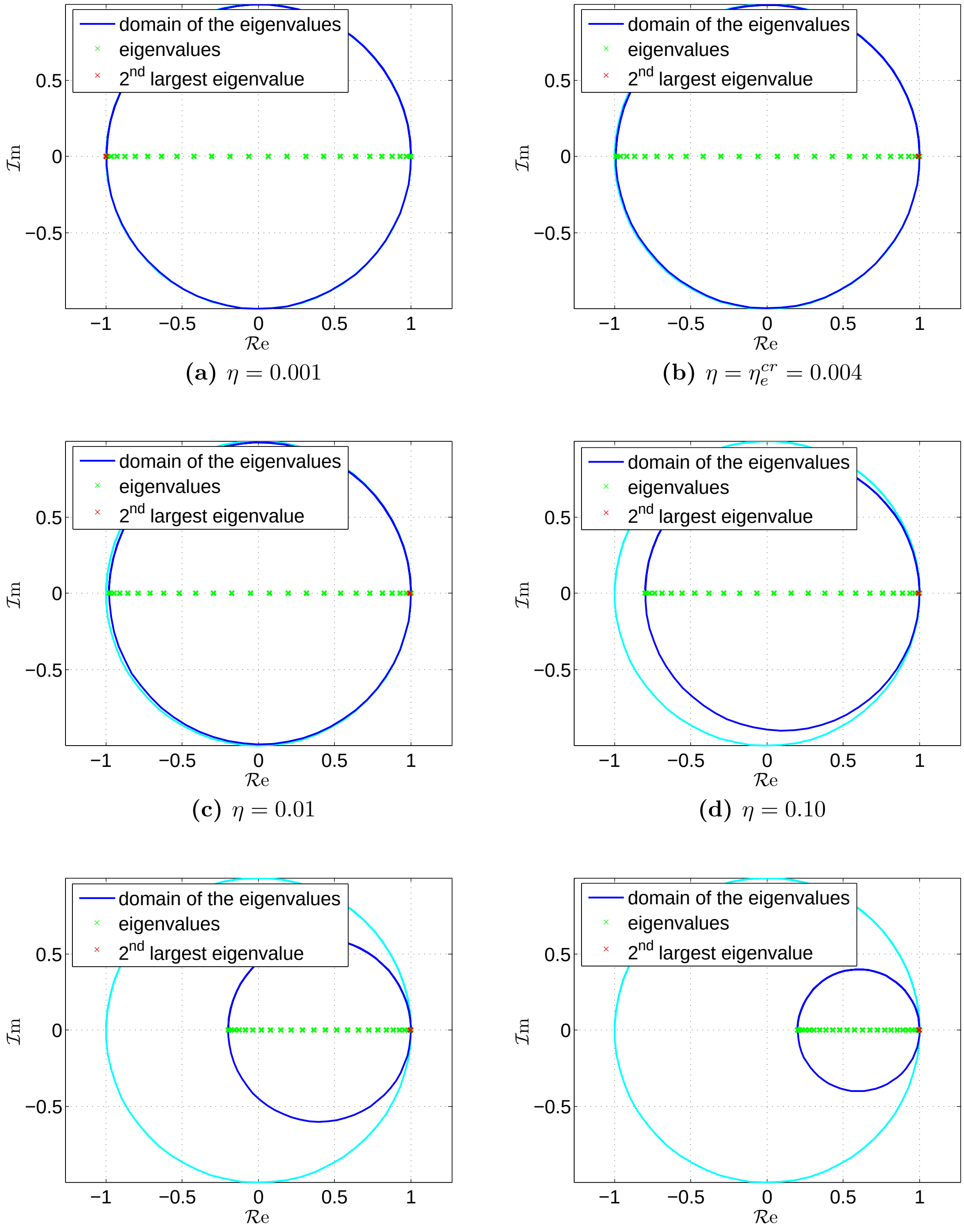
The main issue emerging is related to the use of the state-space model (3.22): indeed since N is big, the critical value η_e^{cr} becomes close to 0. This means that the state-space model (3.22) reduces itself to an approximation of the model (3.19) (see Figure 5.7).
Figure 5.8 report the domain of the eigenvalues and the spectre of F_1 with $N = 50$ varying the tuning parameter η. As η is increased, the domain shrinks and when $\eta = \eta_e^{cr} = 0.004$, the second largest eigenvalue is positive (Figure 5.8b). In this situation eigenvalues are near to the point -1, which causes instability of the system. From this point of view, the critical value becomes less significant as an index of the performance of the convergence (Figure 5.9) because damped oscillations linger for several iterations of the process before reaching the steady value. Indeed, applying the second distributed model with $\eta \leq \eta_{cr}$ is not convenient, then, in this case, it is preferable to choose for example $\eta = 0.1$, which is a little more higher than η_{cr}, to avoid oscillations.



(a) Performance index Ψ

(b) Estimated angles $\widehat{\beta}_i$ (rad)

Figure 5.7: Comparison between models (3.19) and (3.22) with $\eta = \eta_{cr}$: convergence analysis for a cyclic network C_{50}^1.



(a) $\eta = 0.001$

(b) $\eta = \eta_e^{cr} = 0.004$

(c) $\eta = 0.01$

(d) $\eta = 0.10$

(e) $\eta = 0.40$

(f) $\eta = 0.60$

Figure 5.8: N=50: Eigenvalues on the imaginary plane with respect to the variation of the control parameter η.

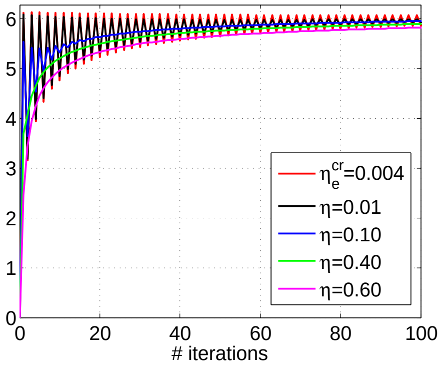


Figure 5.9: Model (3.22): scaling of the control parameter η with respect to the critical value in the C^1_{50} network (estimated angle $\hat{\beta}_{50}$).

5.5 Edge Selection

Taking inspiration from the analysis about the addition of links to the camera network C^1_N, the next step is the selection of a subset of k additional edges which allow the estimation system to perform in the best possible way. The choice of these links depends on the part of the estimation that has to be improved: for instance the speed of convergence, the reduction of oscillatory phenomena or the robustness to the noise. Then a challenging question would be the choice of a set of optimal links without the basis of the cyclic network, only starting from the number of available nodes, *i.e.* the construction of an *ad hoc* network for solving the localization issue.

In literature the generic problem of choosing edges to add to a graph in order to optimize some property of a network is known as **edge selection** and can be applied in various contexts.

From this perspective, recent studies have focused on the application of edge selection in the consensus procedure for multi-agent systems. Indeed, the topology of a communication graph is strictly related to the agreement process among nodes. Moreover the structure of a graph plays a key role also in the problem of network synchronization. Main researches tackle with the problem of edge selection in order to make the network more reliable with respect to white noise or external disturbances. For that purpose *network coherence* is the introduced quantity to measure robustness, defined in terms of the H_2 norm of the system. A sort of selection strategy is proposed in [20] where the objective is to find a group of *leader* nodes that maximize the network coherence. The authors suggest two algorithms to solve the related optimization problem and compare them. This is the topic also of the study conducted by Lin et al.[9]. Fitch and Leonard [10] devote their work in seeking systematic solutions to this subject proved by theoretical means.

Alternatively to network coherence, effective graph resistance is a graph measure which can be an index of robustness in a network [7]. In this cited paper, the question of adding an edge to optimize the effective graph resistance is a still open question. Keeping in mind the target of optimization, the problem of edge selection can be interpreted as the more generic issue of partitioning a graph. An interesting criterion is the *normalized cut* [22], which gives information about the degree of similarity and dissimilarity among nodes. This technique requires a pre-discrimination among links, that is an assignment of weights, which, in the localization task, is not a priori feasible.

Goal of this section is the attempt to apply the algorithm planned in [25] to the localization system. Firstly an overview of the functioning of the method for selecting edges is described, endowed with the mathematical tools needed for it. Then it is tested and some important facts are highlighted.

5.5.1 Greedy algorithm

The design of topology for optimizing network coherence is based on the definition of network coherence itself.
A weighted graph $\mathscr{G} = (\mathcal{V}, \mathcal{E}, w)$ is identified by the set of nodes, the set of edges and the corresponding non-negative weights associated (which, in this case, have the same value). The consensus dynamics of the network is modelled by the equation

$$dx\left(t\right) = -Lx\left(t\right)dt + dW$$

where $x(t)$ is a scalar state variable relating to a node, L the Laplacian matrix and dW is a vector representing the noise of the system.

Formally, *network coherence* is defined as the steady-state deviation from the average value:

$$\mathcal{C} = \lim_{t\to\infty}\sum_{t=1}^{n}\mathbf{E}\left[\left(x_i\left(t\right) - \frac{1}{n}\sum_{j=1}^{n}x_j\right)^2\right]$$

The expression is linked with the H_2 norm and to the eigenvalues $\lambda_i,\ i = 1, \ldots, n$ of the Laplacian matrix

$$\mathcal{C} = \frac{1}{2}\mathbf{trace}\left(L^\dagger\right) = \frac{1}{2}\sum_{i=2}^{n}\frac{1}{\lambda_i\left(L\right)}$$

Then, paper [25] formulates the problem from the perspective of a set function optimization and solves it taking advantage of the definition of *submodularity.*
Indeed, for a set $V = \{1, \ldots M\}$, a set function $f : 2^V \to \mathbb{R}$ matches a real number to each subset of V. The optimization is made by choosing from all the possible subsets of k elements of V, the one that maximizes the function f, in formula:

$$\underset{S\subseteq V, |S|=k}{\text{maximize}}\ f\left(S\right).$$

The property of *submodularity* of a set function f is that for all $A \subseteq B \subseteq V$ and all elements $s \notin B$ it holds

$$f\left(A \cup \{s\}\right) - f\left(A\right) \geq f\left(B \cup \{s\}\right) - f\left(B\right).$$

In addition, a function f is said to be *monotone increasing* if, for all $A, B \subseteq V$, it holds the relation

$$A \subseteq B \Rightarrow f\left(A\right) \leq f\left(B\right),$$

conversely f is called monotone decreasing if

$$A \subseteq B \Rightarrow f\left(A\right) \geq f\left(B\right).$$

A characterization of submodular functions is given by the following theorem:
Theorem: the function $f : 2^V \to \mathbb{R}$ is submodular if and only if the derived set functions
$f_a : 2^{V \setminus \{a\}} \to \mathbb{R}$

$$f_a\left(X\right) = f\left(X \cup \{a\}\right) - f\left(X\right)$$

are monotone decreasing $\forall a \in V$.
This theorem is crucial in assuring that function

$$f\left(\mathcal{E}\right) = -\textbf{trace}\left(L_{\mathcal{E}}^{\dagger}\right),$$

where $L_{\mathcal{E}}$ is the Laplacian associated to the initial edge set joined with the additional edges, is submodular (for a complete proof see [25]).
Then the problem is reduced to

$$\underset{\mathcal{E} \subset V \times V \setminus E}{\text{minimize}} \; \textbf{trace}\left(L_{\mathcal{E}}^{\dagger}\right).$$

Maximization of a submodular function is generally a NP-hard problem, but it can be found an approximated solution with a greedy heuristic. The steps of implementation of the algorithm are reported in the pseudo-code above:

```
$S_0 \leftarrow \emptyset$
$\triangleright$ $S_0$: initial set of added edges
**for** $i \leftarrow 1$ **to** $k$
$\triangleright$ $k$: number of added edges
        **do** $\forall a \in V \setminus S_i$
            $\Delta\left(a|S_i\right) = f\left(S_i \cup \{a\}\right) - f\left(S_i\right)$
            $\triangleright$ $\Delta\left(a|S_i\right)$: gain computed for every possible extra edge
            $S_{i+1} \leftarrow S_i \cup \left\{\underset{a}{\text{argmax}}\Delta\left(a|S_i\right)|a \in V \setminus S_i\right\}$
            $\triangleright$ addition to $S_{i+1}$ of any element with the highest gain
$\triangleright$ $S_k$: final set of added edges
```

5.5.2 Test of the algorithm

The proposed technique is validated on the standard set-up of a C_6^1 network. Returned edges are represented in Figure 5.10.

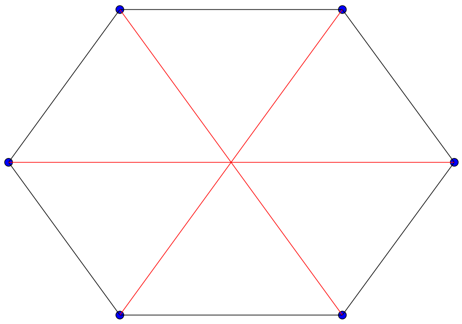
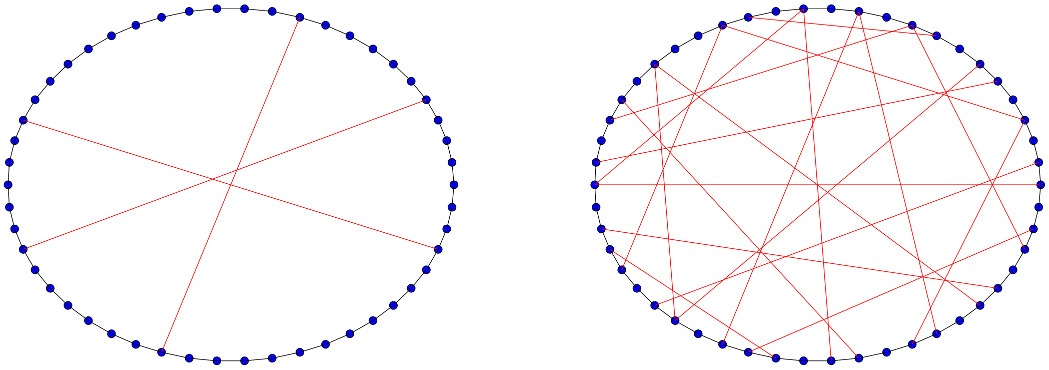


Figure 5.10: Subset of $k = 3$ added edges to the C_6^1 network.

It is noticeable that the added links are really the ones for which the graph remains bipartite. In fact this response makes sense since these are long distance links which intuitively reflect the idea of robustness. For a large network the application of the algorithm gives the results shown in Figure 5.11.



(a) $k = 3$ **(b)** $k = 20$

Figure 5.11: Added edges to the C_{50}^1 network with different cardinality of the subset k.

5.5.3 Observations on the algorithm

Actually, the analyzed distributed model (3.19) certainly implies a consensus dynamic, but it has been drastically adapted for the specific task of localization. Even tough the consensus is regulated by the matrix F which is connected, in some way, to the Laplacian L, here the noise is composed by the noisy measurements, which are directly linked to the state of the system. From this point of view one could imagine to separate the real noise from the state and to build a modified system.

From (3.18), the estimate of measurements is added and subtracted to both the parts, neglecting the dependence on time k:

$$\hat{\beta}_i = \frac{\sum_{j\in\mathscr{N}_i}\hat{\beta}_j}{\deg\left(c_i\right)} + \frac{\sum_{j\in\mathscr{N}_i}\left(\tilde{\beta}_{ji}-\tilde{\beta}_{ij}\right)}{2\deg\left(c_i\right)} + \frac{\sum_{j\in\mathscr{N}_i}\left(\hat{\beta}_{ji}-\hat{\beta}_{ij}\right)}{2\deg\left(c_i\right)} - \frac{\sum_{j\in\mathscr{N}_i}\left(\hat{\beta}_{ji}-\hat{\beta}_{ij}\right)}{2\deg\left(c_i\right)} \tag{5.2}$$

Remembering that

$$\tilde{\beta}_{ji} = \hat{\beta}_{ji} + \hat{w}_{ji} = \hat{\beta}_i - \hat{\beta}_j + \hat{w}_{ji}$$

the equation 5.2 becomes

$$\hat{\beta}_i = \underbrace{\frac{\sum_{j\in\mathscr{N}_i}\hat{\beta}_j}{\deg\left(c_i\right)} + \frac{\sum_{j\in\mathscr{N}_i}\left(\hat{\beta}_{ji}-\hat{\beta}_{ij}\right)}{2\deg\left(c_i\right)}}_{F'\hat{\boldsymbol{\beta}}} + \underbrace{\frac{\sum_{j\in\mathscr{N}_i}\left(\tilde{\beta}_{ji}-\tilde{\beta}_{ij}\right)}{2\deg\left(c_i\right)} - \frac{\sum_{j\in\mathscr{N}_i}\left(\hat{\beta}_{ji}-\hat{\beta}_{ij}\right)}{2\deg\left(c_i\right)}}_{\hat{\boldsymbol{w}}} \tag{5.3}$$

leading to the system

$$\rightarrow \hat{\boldsymbol{\beta}}\left(k+1\right) = F'\hat{\boldsymbol{\beta}}\left(k\right) + \hat{\boldsymbol{w}}. \tag{5.4}$$

With some algebraic manipulations, the state is expressed as

$$\hat{\beta}_i = \frac{\sum_{j\in\mathscr{N}_i}\hat{\beta}_j + \left(\hat{\beta}_i - \hat{\beta}_j\right)}{\deg\left(c_i\right)} + \hat{w}_i = \frac{\sum_{j\in\mathscr{N}_i}\hat{\beta}_i}{\deg\left(c_i\right)} + \hat{w}_i = \hat{\beta}_i + \hat{w}_i \longrightarrow F' = I. \tag{5.5}$$

Being the state-space matrix of the modified model (5.4) the identical matrix, the application of the greedy algorithm has no significance.

For a cyclic network C_N^1 the connection between the eigenvalues of L and of F can be calculated analytically since eigenvalues of F are known:

$$\Lambda\left(L\right) = \Lambda\left(\Delta - A\right) = 2 - 2\cos\left(\frac{2\pi}{N}h\right) \qquad h = 0,\ldots,N-1.$$

The Fiedler eigenvalue λ_2, which is the smallest eigenvalue of L excluding 0, is an index of the speed of convergence of the system, in particular more it is high, more the system is faster. This value is generally related to the second largest eigenvalue of F. For this reason, at the beginning, it was presumed that the greedy algorithm could have worked properly, because the set function f is defined in terms of the eigenvalues of the Laplacian matrix.

Probably, extracting a subset of links to optimize the speed of convergence is only a matter within the structure of the matrix of consensus F and strictly related to the computation of the eigenvalues. Regarding the localization task, edge selection is an unresolved problem.

Chapter 6

Experiments with a laboratory setup

In this chapter the models proposed for the localization of planar camera networks are tested on an experimental setup installed in the NAVLAB laboratory at DEI. The environment is represented in Figure 6.1. This type of configuration portrays a possible real scenario, in fact it may reproduce a situation where a task of perimeter video-surveillance is required. The camera network has the structure of a circulant graph, as already implemented in the previous simulations, and it is heterogeneous, since it consists of both PTZ devices and webcams.

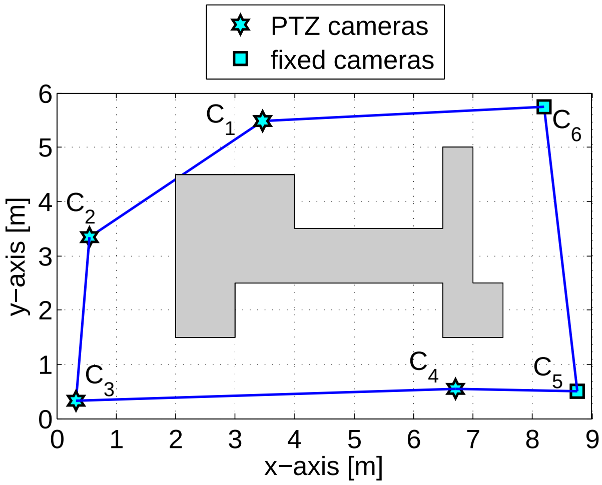


Figure 6.1: Plan of the experimental setup.

Concerning the implementation of the distributed models, the initial noisy measurements are introduced starting from pan-tilt relative rotations collected experimentally: the variance of the noise is estimated according to the different types of cameras considered in the rotations. In other words, if the noisy relative pan angle concerns PTZ cameras, a first variance is estimated, similarly other variances are evaluated for webcams, and pairs of PTZ and fixed cameras. This noise is then used to produce a synthetic setup. This choice is made because the intention is to make the analysis uniform and consistent also if an additional edge is added, having available experimental data only relating to the initial camera network.

6.1 Application of distributed models

Considering a synchronous protocol communication among agents, the first distributed strategy (3.19) is applied to find the orientations of the cameras. Estimated angles are normalized with respect to the first camera C_1, whose coordinate frame is taken as the reference, at the end of the process. It is supposed that the translational part of the pose is known. Figure 6.2 shows the results, where outlined rotations are computed on the basis of the last executed estimation.

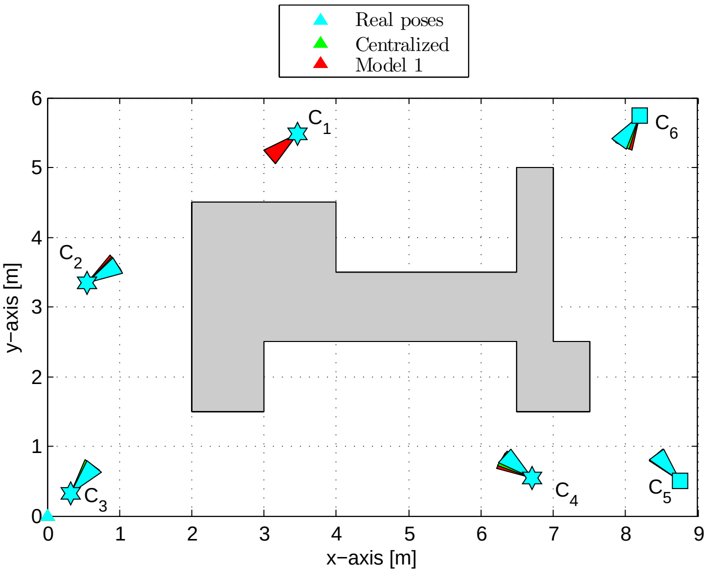


Figure 6.2: Test of the first distributed model on the scenario.

A substantial difference between this simulation and the one performed in Chapter 3 is that oscillations relating to some of the estimated angles are not perceived so clearly (Figure 6.3). Instead, even though convergence is not assured, in this case these estimates can be evaluated as a decent approximation of the real orientations. However, this does not depend on the particular distribution of the error on the measurements, as it could be thought, but, rather, on the combination of the initial noisy measurements $\tilde{\beta}_{ij}$ that compose the input of the system u.

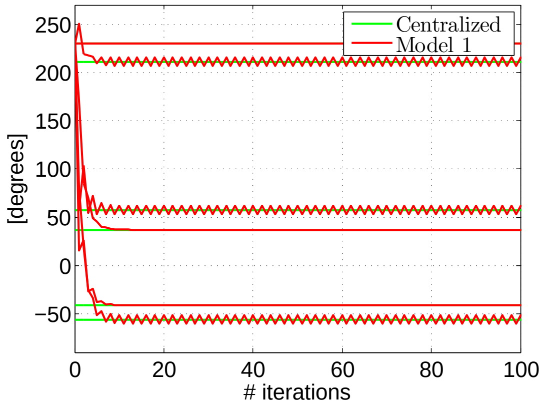


Figure 6.3: Estimated angles $\widehat{\beta}_i$ (degrees).

The second model (3.22) is applied with the tuning parameter fixed to $\eta = 0.1$ and $\eta = 0.6$ respectively (Figure 6.4 and 6.5). As it is expected, in both cases at the final instant of the process convergence to the centralized solution is nearly reached, since rotations coincide (see Figure 6.6). An evidence of this observation is also given by values of the mean squared error (Table 6.1).

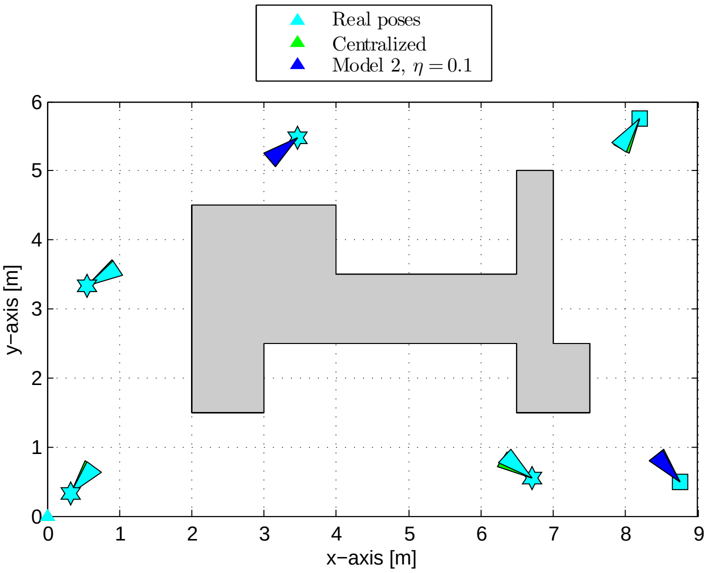


Figure 6.4: Test of the second distributed model ($\eta = 0.1$) on the scenario.

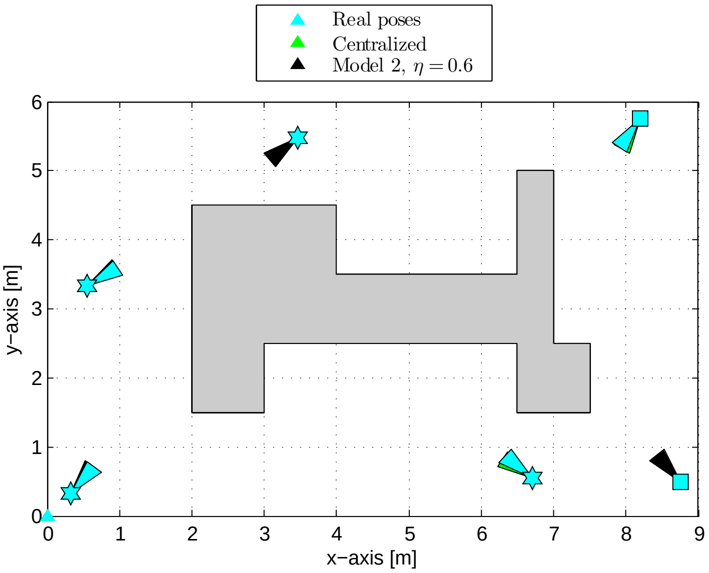
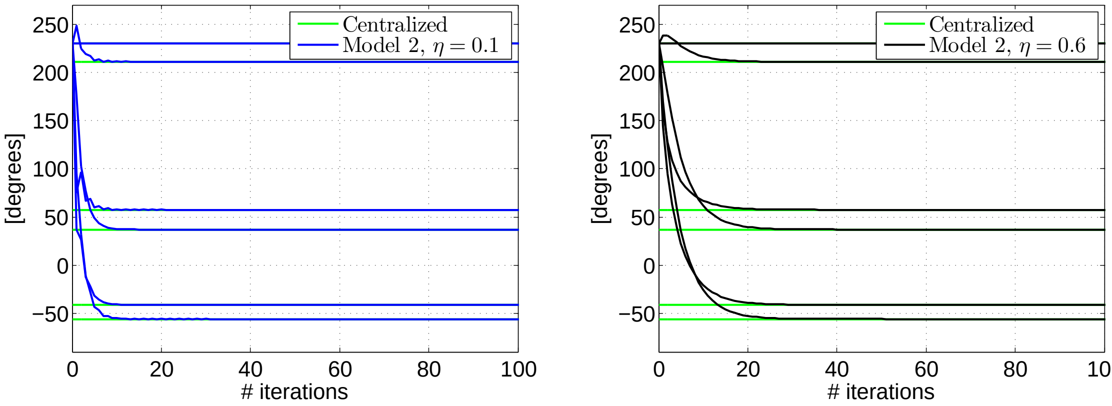


Figure 6.5: Test of the second distributed model ($\eta = 0.6$) on the scenario.



(a) Model 2, $\eta = 0.1$

(b) Model 2, $\eta = 0.6$

Figure 6.6: Estimated angles $\widehat{\beta}_i$ (degrees).

In Table 6.1 the mean squared error is reported comparing the values found with these experimental tests, that are based on measurements collected in laboratory, and by simulations, where data are purely synthetic. Regarding the first model, as already observed, the estimation becomes closer to the asymptotic solution. The difference between values of experimental and simulated test is due to the distribution of the noise, which in the experimental setup is characterized by higher variances.

Ψ (100) [°]	Experimental	Simulated
Centralized	11.1737	4.5695
Model 1	39.334	1881.9
Model 2, $\eta = 0.1$	11.1737	4.5695
Model 2, $\eta = 0.6$	11.1737	4.5695

Table 6.1: Values of mean squared error at the end of convergence process (*100-th* iteration) in the tests carried out.

6.2 Addition of a link

In this section the aim is to check the performance of model 1 (3.19) if an extra communication link is added to the camera network. In particular, it is supposed that it is possible to create an open path in the compact structure in order to make the views of the considered cameras overlap. On a real environment this might correspond to the presence of a passage or a decrease in the height of the building so that the devices can communicate and share the view of the scene without occlusions.

In Figure 6.7 is reported the camera network where cameras C_1 and C_4 are neighbours.

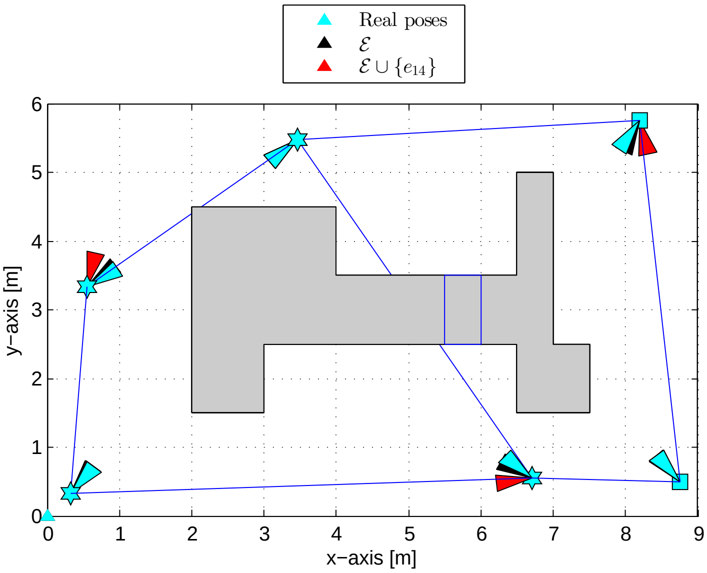


Figure 6.7: Test of the first distributed model with the communication between cameras C_1 and C_4.

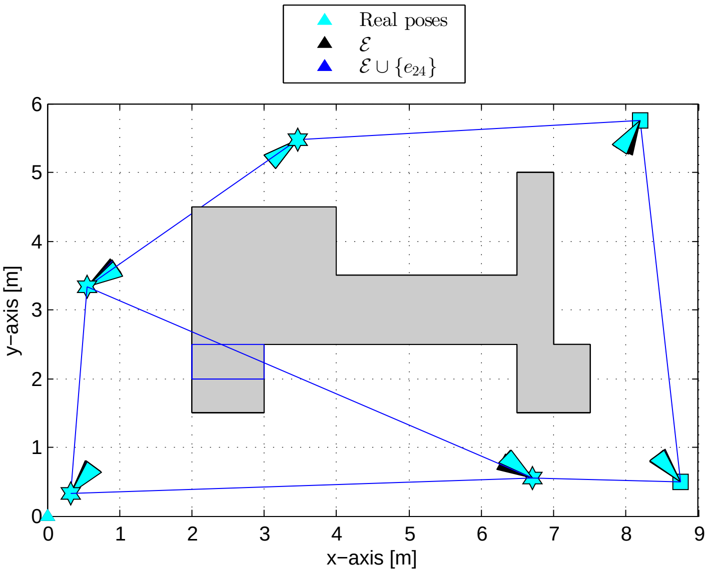
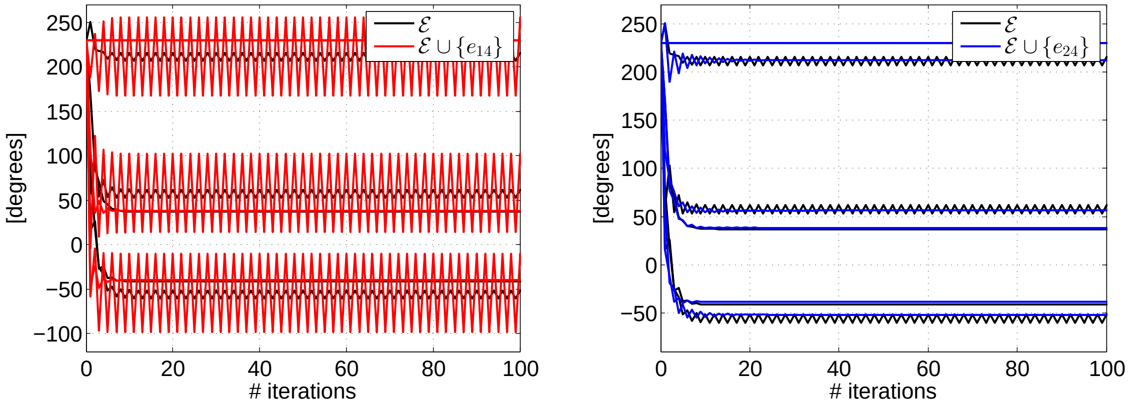


Figure 6.8: Test of the first distributed model with the communication between cameras C_2 and C_4.



(a) $\mathcal{E} \cup \{e_{14}\}$

(b) $\mathcal{E} \cup \{e_{24}\}$

Figure 6.9: Estimated angles $\widehat{\beta}_i$ (degrees).

The presence of this further edge, which maintains the layout of the graph bipartite, causes the incorrect estimation of rotations of cameras C_2, C_4 and C_6. The amplitude of oscillations of the pan angles is due to the specific combination of the noisy measurements which compose the input of the system.
Conversely, adding the edge which connects C_2 and C_4 (Figure 6.8), the graph associated to the camera network becomes non-bipartite. Consistent with this observation,

estimated rotations are close to their true value and the mean squared error is lower than the one of C_6^1 network (see Table 6.2). Figure 6.9 reports the trend of the estimated angles with the addition of the specified links.

$\Psi\,(100)$ [°]	$\mathcal{E}$	$\mathcal{E} \cup \{e_{14}\}$	$\mathcal{E} \cup \{e_{24}\}$
Model 1	39.334	1116.4	5.1147

Table 6.2: Values of mean squared error at the end of convergence process (*100-th* iteration) with an additional link.

6.3 Enlargement of the circulant network

In this section a new synthetic setup is presented. With reference to the standard experimental configuration the camera network is expanded including new cameras, creating the network N_{12}. These are placed in an intermediate position with respect to a pair of devices already arranged, to ensure the covering of some common fields and therefore to allow the communication. In this way each new node has two neighbours and the remaining nodes gain two additional neighbours. Added cameras are both PTZ and webcams.

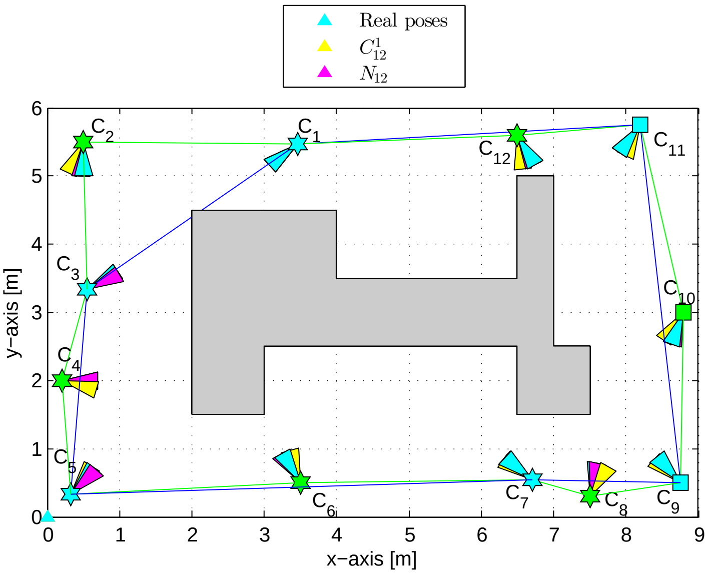


Figure 6.10: Test on the enlargement of the network: additional cameras are coloured in green, while, as before, the others are light blue. Similarly, green lines point out the additional communications, whereas the light blue links belong to the primal layout.

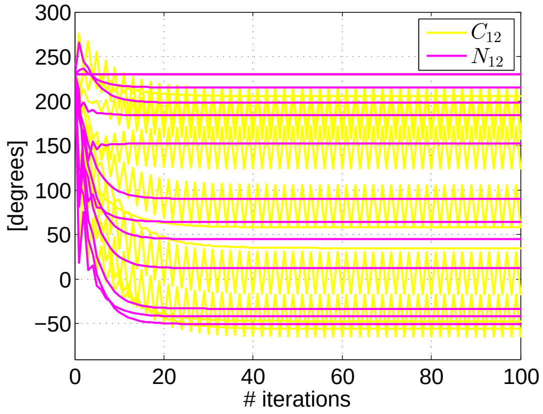


Figure 6.11: Estimated angles $\widehat{\beta}_i$ (degrees).

$\Psi\,(100)$ [°]	C^1_{12}	$C^1_{12} \cup \{e_{24}\}$	N_{12}
Model 1	191.6509	22.5987	7.276

Table 6.3: Values of mean squared error at the end of convergence process (*100-th* iteration) for the cyclic network and its enlargement.

A graphical representation of the network and of the estimation outcomes is provided in Figure 6.10.

A comparison between the performance of the model 1 on the cyclic network C^1_{12} and on the network N_{12} is made, also by means of the data gathered in Table 6.3 and observing the estimated angles in Figure 6.11. Analyzing the mean squared error, the configuration N_{12} gives the closest approximation to the true poses and it can be inferred that generally the addition of links allows an improvement of the estimates, as it was mentioned in section 5.3.

Chapter 7

Conclusions

The subject discussed in this thesis is the problem of localization in a camera system. In detail, only the rotational part of the pose is considered and the case study is restricted to a planar setup, where a rotation can be identified with a single angle.
An analytical solution is derived by minimizing a suitable functional cost. Alongside to a centralized approach to the problem, two distributed strategies are presented and examined.
These models are evaluated changing the communication protocol and varying the network topology.

Since the performance of the second distributed system can be regulated tuning the parameter η, a challenge is to overcome the non-convergent trend of the first model in a network represented by a bipartite graph. There are several possibilities to deal with that. Apart from the addition of an edge which makes the network non-bipartite, a protocol different from the synchronous one can be adopted. While the symmetric gossip entails some intrinsic problems, probably the best protocol to be used is the coordinated broadcast which assures convergence to the asymptotic value. The distinction between a small or a large scale network has to be taken into consideration to establish the preferable solution.

Concerning the second distributed approach, features about the behaviour to reach the centralized value, relating to the network topology, are studied. For instance, it emerges that groups of additional edges can be partitioned in sets corresponding to determined network configurations according to the speed of convergence of the estimation process.
Other significant points of the analysis are that the increase of information within the camera network, in the majority of the cases, does not deteriorate the estimates and that the localization process is heavily influenced by the distribution of the error on the measurements.

On the basis of this work, future projects can be developed.
First of all, bounds on the eigenvalues can be derived in order to find the critical value

of η for networks where a closed formula for the eigenvalues of the state-space matrix is not known. Secondly, a further study can be conducted to look for an algorithm for edge selection.

A research on the extension of the analytical rotation estimation from the $SO\left(2\right)$ to the $SO\left(3\right)$ space can be done. In parallel with this, an analytical formulation for the translational part of the pose can be worked out in the same planar scenario.

Bibliography

[1] H. Aghajan and A. Cavallaro. *Multi-Camera Networks: Principles and Applications*. Ed. by Academic Press. 2009.

[2] N. Anjum. “Camera localization in distributed networks using trajectory estimation”. In: *Journal of Electrical and Computer Engineering* 2011 (2011).

[3] G. Baggio, M. Michielan, and S. Patron. “Distributed image-based localization of camera networks: a comparative analysis of different communication protocols”. 2012-2013.

[4] J. Black, T. J. Ellis, and D. Makris. “Wide area surveillance with a multi camera network”. In: *Intelligent Distributed Surveillance Systems, Conference on.* 2004.

[5] D. Borra et al. “Path planning for mobile robots using a video camera network”. In: *Automatica [submitted]* (2012).

[6] A. Cenedese, M. Michielan, and G. Michieletto. “Distributed strategies for the estimation of rotations in camera networks”. In: *IEEE Robotics and Automation Letters and ICRA [submitted]* (2015).

[7] W. Ellens et al. “Effective graph resistance”. In: *Linear Algebra and its Applications* 435.10 (2011), 2491–2506.

[8] A. O. Ercan et al. “Optimal placement and selection of camera network nodes for target localization”. In: ed. by Springer. Chap. Distributed Computing in Sensor Systems, pp. 389–404.

[9] M. Fardad F. Lin and M. R. Jovanović. “Algorithms for leader selection in large dynamical networks: noise-corrupted leaders”. In: *Decision and Control, 50th IEEE Conference on.* 2011.

[10] K. Fitch and N. E. Leonard. “Information centrality and optimal leader selection in noisy networks”. In: *Decision and Control, 52th IEEE Conference on.* 2013.

[11] R. Holman, J. Stanley, and T. Ozkan-Haller. “Applying video sensor networks to nearshore environment monitoring”. In: *Pervasive Computing, IEEE Conference on.* Vol. 2. 4, pp. 14–21.

[12] A. Hoover and B. D. Olsen. “Path planning for mobile robots using a video camera network”. In: *Advanced Intelligent Mechatronics, Proceedings of the IEEE/ASME International Conference on.* 1999.

[13] D. Huynh. “Metrics for 3d rotations: Comparison and analisys”. In: *Journal of Mathematical Imaging and Vision* 35.2 (2009), pp. 155–164. URL: `http://dx.doi.org/10.1007/s10851-009-0161-2`.

[14] H. Landau and A. Odlyzko. “Bounds for eigenvalues of certain stochastic matrices”. In: *Linear Algebra and its Applications* 38.0 (1981), pp. 5–15. URL: `http://www.sciencedirect.com/science/article/pii/0024379581900033`.

[15] K. Langendoen and N. Reijers. “Distributed localization in wireless sensor networks: a quantitative comparison”. In: *Computer Networks: The International Journal of Computer and Telecommunications Networking* 43.4 (2003), pp. 499–518.

[16] Y. Ma et al. *An Invitation to 3-D Vision.* Ed. by Springer. 2001.

[17] W. E. Mantzel, H. Choi, and R. G. Baraniuk. “Distributed camera network localization”. In: *Signals, Systems and Computers, Conference Record of the Thirty-Eighth Asilomar Conference on.* Vol. 2. 2004, pp. 1381–1386.

[18] M. Meingast, S. Oh, and S. Sastry. “Automatic camera network localization using object image tracks”. In: *Computer Vision, IEEE 11th International Conference on.* 2007.

[19] M. Moakher. “Means and averaging in the group of rotations”. In: *SIAM Journal on Matrix Analysis and Applications* 24.1 (2002), pp. 1–16. URL: `http://dx.doi.org/10.1137/S0895479801383877`.

[20] S. Patterson and B. Bamieh. “Leader selection for optimal network coherence”. In: *Decision and Control, 49th IEEE Conference on.* 2010.

[21] G. Piovan et al. “On frame and orientation localization for relative sensing networks”. In: *Decision and Control, 47th IEEE Conference on.* 2008.

[22] J. Shi and J. Malik. “Normalized cuts and image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905.

[23] P. Skraba and L. Guibas. “Energy efficient intrusion detection in camera sensor networks”. In: ed. by Springer. Chap. Distributed Computing in Sensor Systems, pp. 309–323.

[24] B. Song et al. “Tracking and activity recognition through consensus in distributed camera networks”. In: *Image Processing, IEEE Transactions on.* Vol. 19. 10, pp. 2564 –2579.

[25] T. Summers et al. “Topology design for optimal network coherence”. In: *Control Conference (ECC).* 2015.

[26] R. Tron and R. Vidal. “Distributed image-based 3d localization of camera sensor networks”. In: *Decision and Control, Proceedings of the 48th IEEE Conference on.* 2009, pp. 901–908. URL: `http://www.cis.jhu.edu/~rvidal/publications/cdc09-localization.pdf`.

[27] A. Williams, D. Ganesan, and A. Hanson. “Aging in place: fall detection and localization in a distributed smart camera network”. In: *Multimedia, Proceedings of the 15th International Conference on.* 2007, pp. 892–901.