Chapter 7

Localization in Two-Dimensional Space

In this chapter, we introduce a distributed localization problem of multi-agent systems in two-dimensional (2D) space. This problem has found numerous important applications in (wireless) sensor networks, including environmental information collection, wildlife monitoring, target tracking, and intrusion detection. In these applications, it is essential that the individual sensor nodes know their positions in a common (global) coordinate frame. For example, it would be ideal to have a GPS onboard each sensor. In practical sensor networks, however, there are typically a large number of sensor nodes each with limited hardware/software capacities. Thus it is costly and implementationally difficult to install a device like GPS on every sensor, not to mention that there are situations where GPS is at best inaccurate and at worst denied.

Therefore it is desirable to have a distributed scheme to determine the global positions of individual sensor nodes based on low-cost, easily implementable onboard devices. A typical such scheme is to compose a sensor network with a minority of *anchor* nodes that do know their positions in the global coordinate frame, and the rest majority of *free* nodes that need to determine their global positions based on their local frames and locally sensed information (e.g. distances and bearing angles with respect to neighboring nodes). Those anchor nodes play the role of *leaders* or *landmarks*, while the free nodes are *followers*. We adopt this distributed scheme, and focus in this chapter on solving a localization problem in 2D, while 3D localization will be covered in Chapter 9.

To solve the 2D distributed localization problem, we present an approach based on complex Laplacian matrices. Modeling the interacting sensor nodes by digraphs, we show that a necessary graphical condition to achieve 2D localization is that the digraph contains a *spanning* 2-*tree* whose two roots are anchor nodes. This condition is similar to that for achieving 2D similar formations in the preceding chapter. However, the two anchor nodes (i.e. two roots) who already know their global positions should not, and will not, change their positions; hence they do not have, nor do they need, any neighbors (i.e. incoming edges). In this way, the *exact* global positions of the free nodes may be determined (without the flexibility of translation, rotation, and scaling as in the similar formation problem). Under the above graphical condition, we present a distributed algorithm for the free nodes to achieve localization in 2D.

7.1 Problem Statement

Consider a network of n (> 1) agents that are stationary in a plane (i.e. their two-dimensional positions are fixed), and a global coordinate frame Σ which is unknown to the agents. The agents labeled $1, 2$ (renumbering if necessary) are the *anchor agents*, whose positions $\xi_1, \xi_2 \in \mathbb{C}$ in Σ are known. Here $\mathrm{Re}(\xi_i)$ and $\mathrm{Im}(\xi_i)$ are the positions of agents $i \in [1, 2]$ on the real and imaginary axes, respectively. The rest agents labeled $3, \ldots, n$ are the *free agents*, whose positions $\xi_3, \ldots, \xi_n \in \mathbb{C}$ in Σ are unknown and need to be determined by these individual free agents. Let

$$\xi_a := \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \in \mathbb{C}^2, \quad \xi_f := \begin{bmatrix} \xi_3 \\ \vdots \\ \xi_n \end{bmatrix} \in \mathbb{C}^{n-2}$$

be the aggregated position vectors of the anchor and free agents, respectively. Write

$$\xi := \begin{bmatrix} \xi_a \\ \xi_f \end{bmatrix} \in \mathbb{C}^n$$

and call ξ the *configuration* of the agents.

To determine its own position, each free agent i $(\in [3, n])$ is equipped with a *state* variable $x_i(k) \in \mathbb{C}$, which denotes the *estimate* of agent i's position ξ_i under the global frame Σ. The time $k \geq 0$ is a nonnegative integer and denotes the *discrete* time. Let

$$x_f(k) := \begin{bmatrix} x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} \in \mathbb{C}^{n-2}$$

be the aggregated state vector of the free agents at time k. It is desired that

$$x_f(k) \to \xi_f \text{ as } k \to \infty.$$

For convenience, also let $x_a(k) := [x_1(k)\ x_2(k)]^\top \in \mathbb{C}^2$ be the aggregated state vector of the two anchor agents, such that $x_a(k) = \xi_a$ for all $k \geq 0$ (i.e. the anchor agents know their positions in the global frame Σ from the initial time $k = 0$ and never update their estimates). Write $x(k) := [x_a(k)^\top\ x_f(k)^\top]^\top \in \mathbb{C}^n$. Hence the purpose of localization is to achieve

$$\lim_{k \to \infty} x(k) = \xi.$$

We model the interconnection structure of the networked agents by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Each *node* in $\mathcal{V} = \{1, ..., n\}$ stands for an agent, and each directed *edge* (j, i) in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes that agent i can obtain the relative state information from agent j. The *neighbor set* of agent i is $\mathcal{N}_i := \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$. For the two anchor nodes (numbered 1 and 2 without loss of generality), since they do not update their states, even if they had neighbors, the corresponding incoming edges would be associated with weight 0. This is equivalent to considering that the anchor nodes do not have neighbors. For this reason, henceforth in this chapter we consider that $\mathcal{N}_i = \emptyset$ $(i = 1, 2)$.

Moreover, consider that digraph $\mathcal{G}$ is weighted: each edge $(j, i) \in \mathcal{V}$ is associated with a complex weight $a_{ij} \in \mathbb{C}$. Hence the adjacency matrix $A = (a_{ij})$, degree matrix $D = \mathrm{diag}(A\mathbf{1})$, and Laplacian matrix $L = D - A$ are all complex. Since $\mathcal{N}_i = \emptyset$ for the anchor nodes $i = 1, 2$, the Laplacian matrix L has the following structure:

$$L = \begin{bmatrix} L_{aa} & L_{af} \\ L_{fa} & L_{ff} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix}. \tag{7.1}$$

Here $L_{fa} \in \mathbb{C}^{(n-2)\times 2}$ and $L_{ff} \in \mathbb{C}^{(n-2)\times(n-2)}$.

To achieve localization, consider the distributed control

$$u_i(k) = \sum_{j \in \mathcal{N}_i} w_{ij}(x_j(k) - x_i(k)), \quad i \in [1, n]. \tag{7.2}$$

Here the control gain w_{ij} satisfies

$$\text{(i)} \sum_{j \in \mathcal{N}_i} w_{ij}(\xi_j - \xi_i) = 0 \tag{7.3}$$

$$\text{(ii)}\ w_{ij} = \epsilon_i a_{ij}, \quad \epsilon_i \in \mathbb{C} \setminus \{0\}. \tag{7.4}$$

This control u_i in (7.2) is in the same form as that for similar formation control: the gains w_{ij} are not simply the edge weights a_{ij}, but are complex (nonzero) multiples of a_{ij} (7.4) and satisfy linear constraints with respect to the configuration ξ (7.3).

Substituting (7.4) into (7.3) and removing the common multiple ϵ_i yield

$$\sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0. \tag{7.5}$$

This in vector form is $L\xi = 0$. In view of (7.1) we have

$$\begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix} \begin{bmatrix} \xi_a \\ \xi_f \end{bmatrix} = 0.$$

Hence the following equation ensues:

$$L_{ff}\xi_f = -L_{fa}\xi_a \tag{7.6}$$

which relates the configuration of the free agents to that of the anchor agents through appropriate multiplications of submatrices of the complex Laplacian matrix.

Two-Dimensional Localization Problem:

Consider a network of agents (stationary in a 2D space) interconnected through a digraph and a configuration $\xi := [\xi_a^\top \ \xi_f^\top]^\top \in \mathbb{C}^n$, which represents the fixed positions of the agents under the global coordinate frame Σ. Here $\xi_a \in \mathbb{C}^2$ is known but $\xi_f \in \mathbb{C}^{n-2}$ is unknown. Design a distributed algorithm using the control u_i in (7.2) such that

(i) $\mathrm{rank}(L) = n - 2$

(ii) $(\forall x_f(0) \in \mathbb{C}^{n-2}) \lim_{k\to\infty} x_f(k) = \xi_f$.

The first requirement (i) implies $\mathrm{rank}(L_{ff}) = n - 2$; namely L_{ff} is invertible. Then it follows from (7.6) that $\xi_f = -L_{ff}^{-1}L_{fa}\xi_a$. Hence the second requirement (ii) becomes:

$$(\forall x_f(0) \in \mathbb{C}^{n-2}) \lim_{k\to\infty} x_f(k) = -L_{ff}^{-1}L_{fa}\xi_a.$$

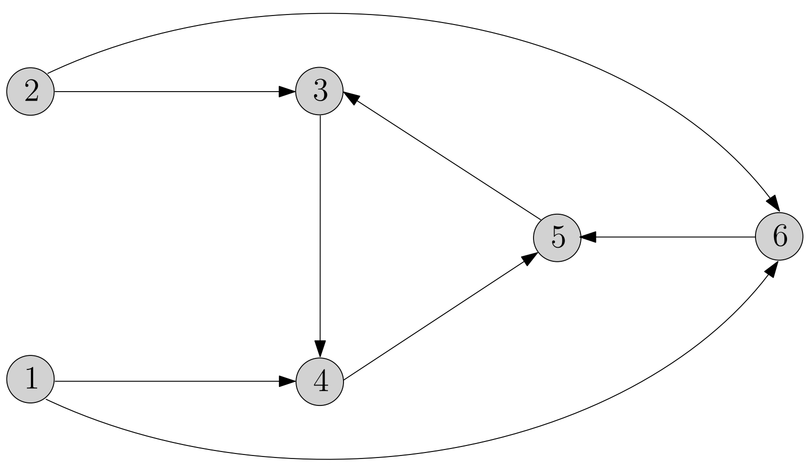

Figure 7.1: Illustrating example of six agents

Example 7.1 *We provide an example to illustrate the localization problem in 2D. As displayed in Fig. 7.1, six agents are interconnected through a digraph; agents 1 and 2 are anchor agents while the rest four are free agents. The neighbor sets of the agents are* $\mathcal{N}_1 = \mathcal{N}_2 = \emptyset$, $\mathcal{N}_3 = \{2, 5\}$, $\mathcal{N}_4 = \{1, 3\}$, $\mathcal{N}_5 = \{4, 6\}$, *and* $\mathcal{N}_6 = \{1, 2\}$.
Let the configuration of the agents be $\xi = [1\ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{2\pi}{3}\mathrm{j}}\ \mathrm{e}^{\pi\mathrm{j}}\ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$, *i.e. a regular hexagon. The position vector of the anchor agents* $\xi_a = [1\ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}}]^\top$ *is known, and that of the free nodes* $\xi_f = [\mathrm{e}^{\frac{2\pi}{3}\mathrm{j}}\ \mathrm{e}^{\pi\mathrm{j}}\ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$ *is unknown and needs to be determined.*
The localization problem is to design a distributed algorithm using the control u_i *in (7.2) such that the rank of the complex Laplacian matrix* L *is* $n-2$, *and moreover the free agents' state vector asymptotically converges to* ξ_f.

A necessary graphical condition for solving the two-dimensional localization problem is given below.

Proposition 7.1 *Suppose that there exists a distributed control* u_i *in (7.2) that solves the two-dimensional localization problem. Then the digraph contains a spanning* 2*-tree whose two roots are the two anchor agents.*

Proof. Suppose that there exists a distributed control in (7.2) that solves the two-dimensional localization problem, but that the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ does *not* contain a spanning 2-tree whose two roots are the two anchor agents. We will derive a contradiction that $\text{rank}(L) < n - 2$, thereby proving that after all $\mathcal{G}$ must contain a spanning 2-tree whose two roots are the two anchor agents.

There are two cases that need to be considered separately. First, the digraph contains a spanning 2-tree but at least one of the two roots is a free agent. In this case, the subdigraph of free agents contains either a spanning tree or a spanning 2-tree. Hence $\text{rank}(L_{ff}) < n - 2$. Since the anchor agents do not have neighbors, $\text{rank}(L) < n - 2$.

The second case is that the digraph does not contain a spanning 2-tree. Then it follows similarly to the proof of Proposition 6.1 that $\text{rank}(L) < n - 2$.

Therefore in both cases above, a contradiction is derived to the solvability of the two-dimensional localization problem. The proof is now complete. □

Owing to Proposition 7.1, we shall henceforth assume the following graphical condition.

Assumption 7.1 *The digraph* $\mathcal{G}$ *modeling the interconnection structure of the networked agents contains a spanning* 2*-tree whose two roots are the two anchor agents.*

Even if Assumption 7.1 holds, not every configuration ξ may be determined by a distributed control u_i in (7.2). Similar to Example 6.2, if ξ is not generic, it is possible that $\text{rank}(L) < n - 2$ for all complex Laplacian matrices L satisfying $L\xi = 0$. This means that the two-dimensional localization problem is not solvable. For this reason, and also the fact that the set of all non-generic configurations has Lebesgue measure zero after all, we assume that the configuration ξ is generic.

Assumption 7.2 *The configuration $\xi := [\xi_a^\top \ \xi_f^\top]^\top \in \mathbb{C}^n$ is generic.*

7.2 Distributed Algorithm

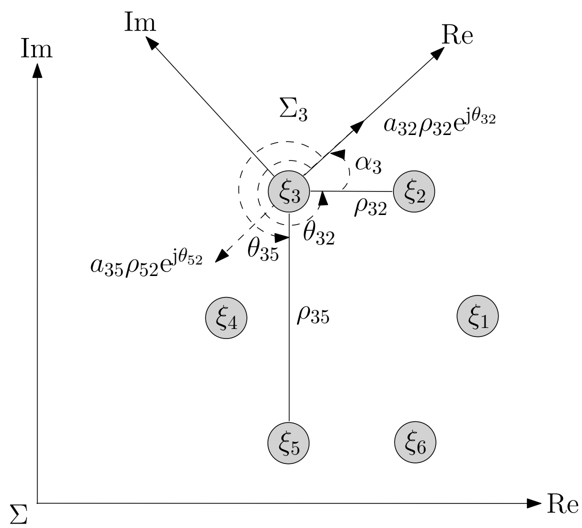

Figure 7.2: Illustration of design of complex weights

Example 7.2 *Consider again Example 7.1, where the configuration is the regular hexagon $\xi = [1 \ e^{\frac{\pi}{3}j} \ e^{\frac{2\pi}{3}j} \ e^{\pi j} \ e^{\frac{4\pi}{3}j} \ e^{\frac{5\pi}{3}j}]^\top$. This ξ is generic.*
The anchor agents' configuration $\xi_a = [1 \ e^{\frac{\pi}{3}j}]^\top$ is known, and the free agents' configuration $\xi_f = [e^{\frac{2\pi}{3}j} \ e^{\pi j} \ e^{\frac{4\pi}{3}j} \ e^{\frac{5\pi}{3}j}]^\top$ is to be determined. To this end, we consider using the simplest form of distributed control (7.2) by setting all $\epsilon_i = 1$:

$$x_i(k+1) = x_i(k) + \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)), \quad i \in [1,6] \tag{7.7}$$

where $a_{ij} \in \mathbb{C}$ are complex weights of edges (j,i) to be designed to satisfy (7.5):

$$\sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0, \quad i \in [1,6].$$

In the following we illustrate how the complex weights may be designed locally to satisfy the *above linear constraints. Each free agent $i \in [3,6]$ has a local coordinate frame Σ_i, whose origin is the (stationary) position of agent i. The orientation of Σ_i is fixed, but the offset angle θ_i with respect to the global coordinate frame Σ is unknown. For each neighbor (free or anchor) $j \in \mathcal{N}_i$, we assume that agent i can sense the relative position by measuring the relative distance and relative bearing angle in Σ_i. That is, if agent j is a neighbor of agent i, then the distance ρ_{ij} between j and i, as well as the bearing angle θ_{ij} of j in Σ_i are measured by i. Thus the relative position in Σ_i is*

$$y_{ij} := \rho_{ij} \mathrm{e}^{\mathrm{j}\theta_{ij}}. \tag{7.8}$$

Note that $y_{ij}\mathrm{e}^{\mathrm{j}\theta_i} = \xi_j - \xi_i$; since θ_i is unknown, even though the relative position y_{ij} in Σ_i is known, $\xi_j - \xi_i$ in Σ is unknown. Substituting $\xi_j - \xi_i = y_{ij}\mathrm{e}^{\mathrm{j}\theta_i}$ into (7.5) and removing the common factor $\mathrm{e}^{\mathrm{j}\theta_i}$, we derive

$$\sum_{j \in \mathcal{N}_i} a_{ij} y_{ij} = 0. \tag{7.9}$$

Hence the weights a_{ij} may be designed based on the relative position y_{ij} in (7.8) under the local coordinate frame Σ_i.
For example, Fig. 7.2 provides an illustrative example. For agent 3, it has two neighbors $2, 5$. Thus we must find weights a_{32}, a_{52} such that $a_{32}y_{32} + a_{35}y_{35} = 0$. In the local coordinate frame Σ_3, $y_{32} = \rho_{32}\mathrm{e}^{\mathrm{j}\theta_{32}}$ and $y_{35} = \rho_{35}\mathrm{e}^{\mathrm{j}\theta_{35}}$. Thus we want to find a_{32}, a_{35} such that

$$a_{32}\rho_{32}\mathrm{e}^{\mathrm{j}\theta_{32}} + a_{35}\rho_{35}\mathrm{e}^{\mathrm{j}\theta_{35}} = 0.$$

There are infinitely many choices; a simple one is $a_{32} = \frac{\mathrm{e}^{-\mathrm{j}\theta_{32}}}{\rho_{32}}$ and $a_{35} = -\frac{\mathrm{e}^{-\mathrm{j}\theta_{35}}}{\rho_{35}}$. Concretely, $\rho_{32} = 1$, $\rho_{35} = \sqrt{3}$, and let $\theta_{32} = \frac{7\pi}{4}$, $\theta_{35} = \frac{5\pi}{4}$; then $a_{32} = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}\mathrm{j}$, $a_{35} = \frac{\sqrt{6}}{6} - \frac{\sqrt{6}}{6}\mathrm{j}$. Similarly we design other complex weights to satisfy (7.9), and write (7.7) in vector form: $x(k+1) = (I - L)x(k)$ where

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}\mathrm{j} & \frac{3\sqrt{2}+\sqrt{6}}{6} + \frac{3\sqrt{2}-\sqrt{6}}{6}\mathrm{j} & 0 & -\frac{\sqrt{6}}{6} + \frac{\sqrt{6}}{6}\mathrm{j} & 0 \\ -\frac{\sqrt{3}}{4} - \frac{1}{4}\mathrm{j} & 0 & \frac{\sqrt{3}}{2} - \frac{1}{2}\mathrm{j} & -\frac{\sqrt{3}}{4} + \frac{3}{4}\mathrm{j} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} + \frac{\sqrt{3}}{2}\mathrm{j} & -\sqrt{3}\mathrm{j} & \frac{1}{2} + \frac{\sqrt{3}}{2}\mathrm{j} \\ -\frac{\sqrt{3}}{2} + \frac{1}{2}\mathrm{j} & \frac{\sqrt{3}}{6} - \frac{1}{2}\mathrm{j} & 0 & 0 & 0 & \frac{\sqrt{3}}{3} \end{bmatrix}.$$

It is verified that the complex Laplacian matrix L has zero row sums and satisfies $L\xi = 0$.

Moreover, partition the matrix L according to anchor agents and free agents:

$$L = \begin{bmatrix} L_{aa} & L_{af} \\ L_{fa} & L_{ff} \end{bmatrix}.$$

Thus $L_{aa} = L_{af} = 0$; $L_{fa} \in \mathbb{C}^{4\times 2}$ and $L_{ff} \in \mathbb{C}^{4\times 4}$. It is checked that rank$(L_{ff}) = 4$, and thus L_{ff} is invertible. Therefore the first condition of the two-dimensional localization problem is satisfied.

It is left to verify the second condition that the state vector of the free agents $x_f(k)$ converges to $-L_{ff}^{-1}L_{fa}\xi_a$ (when $x_a(k) = \xi_a$ for all $k \geq 0$). Fix $\xi_a \in \mathbb{C}^2$. First note that

$$\bar{x} = \begin{bmatrix} \bar{x}_a \\ \bar{x}_f \end{bmatrix} = \begin{bmatrix} \xi_a \\ -L_{ff}^{-1}L_{fa}\xi_a \end{bmatrix}$$

is the unique fixed point of (7.7). To see this, substituting $\bar{x}$ into (7.7) yields $\bar{x}$, which means that $\bar{x}$ is a fixed point of (7.7). Moreover, let

$$\bar{x}' = \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}$$

be another fixed point of (7.7), namely

$$\begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix} = \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix} \right) \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix} = \begin{bmatrix} I & 0 \\ -L_{fa} & I - L_{ff} \end{bmatrix} \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}.$$

From the above we derive

$$\bar{x}'_f = -L_{ff}^{-1}L_{fa}\xi_a = \bar{x}_f.$$

This shows that $\bar{x}$ is the unique fixed point of (7.7), which in turn implies that starting from an arbitrary initial condition $x(0) = [\xi_a^\top \; x_f^\top(0)]^\top \in \mathbb{C}^n$, $x_f(k)$ converges to $-L_{ff}^{-1}L_{fa}\xi_a$ if and only if all the eigenvalues of $I - L_{ff}$ lie inside the unit circle.

Unfortunately, the eigenvalues of matrix $I - L_{ff}$ are

$$-0.5774, 0.3041 - 0.6475\text{j}, -0.9368 - 0.3062\text{j}, -0.0497 + 1.637\text{j}.$$

The last eigenvalue lies outside the unit circle. Hence (7.7) is unstable and $x_f(k)$ diverges. To stabilize $x_f(k)$ to the desired fixed point $-L_{ff}^{-1}L_{fa}\xi_a$ (to satisfy the second requirement *of the two-dimensional localization problem), the unstable eigenvalues of $I - L_{ff}$ must be moved inside the unit circle. This shows that simply setting all $\epsilon_i = 1$ in (7.2) does not work in general. In fact, ϵ_i need to be properly chosen in order to stabilize $I - L_{ff}$.*

In the following we describe a distributed algorithm using (7.2) in vector form, and will analyze its stability in relation to the values of ϵ_i in the next section.

Two-Dimensional Localization Algorithm (TDLA):

Each anchor agent $i \in [1, 2]$ has a state variable $x_i(k) \in \mathbb{C}$ whose initial value is set to be $x_i(0) = \xi_i$ (which is known). Each free agent $i \in [3, \ldots, n]$ also has a state variable $x_i(k) \in \mathbb{C}$ whose initial value is an arbitrary complex number. Offline, each free agent i computes weights $a_{ij} \in \mathbb{C}$ based on the measured relative positions $y_{ij} = \rho_{ij} \mathrm{e}^{\theta_{ij}}$ in (7.8) by solving

$$\sum_{j \in \mathcal{N}_i} a_{ij} y_{ij} = 0.$$

Then online, at each time $k \geq 0$, while each anchor agent stays put, i.e.

$$x_i(k+1) = x_i(k), \quad i \in [1, 2]$$

each free agent i updates its $x_i(k)$ using the following local update protocol:

$$x_i(k+1) = x_i(k) + \epsilon_i \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)), \quad i \in [3, n] \tag{7.10}$$

where $\epsilon_i \in \mathbb{C} \setminus \{0\}$ is a (nonzero) complex control gain.

Let $x := [x_1 \cdots x_n]^\top \in \mathbb{C}^n$ be the aggregated state vector of the networked agents, and

$$E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n) \in \mathbb{C}^{c \times n}$$

the (diagonal and invertible) control gain matrix. Then the n equations (7.10) become

$$x(k+1) = x(k) - ELx(k) = (I - EL)x(k). \tag{7.11}$$

Remark 7.1 *The above TDLA requires that the following information be available for each free agent $i \in [3, n]$:*

- y_{ij} *for all* $j \in \mathcal{N}_i$ *(offline computation of weights)*
- $x_j - x_i$ *for all* $j \in \mathcal{N}_i$ *(online state update).*

7.3 Convergence Result

The following is the main result of this section.

Theorem 7.1 *Suppose that Assumptions 7.1 and 7.2 hold. There exists a (diagonal and invertible) control gain matrix* $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ *such that TDLA solves the two-dimensional localization problem.*

To prove Theorem 7.1, we analyze the eigenvalues of the matrix $I - EL$ in (7.11). For this, the following fact is useful (which is the discrete counterpart of Lemma 6.1).

Lemma 7.1 *Consider an arbitrary square complex matrix* $M \in \mathbb{C}^{n\times n}$*. If all the principal minors of* M *are nonzero, then there exists an invertible diagonal matrix* $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n) \in \mathbb{C}^{n\times n}$ *such that all the eigenvalues of* $I - EM$ *lie inside the unit circle.*

Proof: The proof is based on induction on n. For the base case $n = 1$, $M = m_{11}$ is a nonzero scalar (as the principal minor of M is nonzero). Write $m_{11} = \rho_1 \mathrm{e}^{\mathrm{j}\theta_1}$, and let $\epsilon_1 := \gamma_1 \mathrm{e}^{\mathrm{j}\phi_1}$ where $\gamma_1 \in (0, \frac{1}{\rho_1})$ and $\phi_1 = -\theta_1$. Then $EM = \epsilon_1 m_{11} = \rho_1\gamma_1 \in (0, 1)$. Hence $1 - EM \in (0, 1)$ which lies inside the unit circle.

For the induction step, suppose that the conclusion holds for $M \in \mathbb{C}^{(n-1)\times(n-1)}$. Now consider $M \in \mathbb{C}^{n\times n}$, with all of its principal minors nonzero. Let M_1 be the submatrix of M with the last row and last column removed. Then all the principal minors of M_1 are nonzero, and by the hypothesis there exists an invertible diagonal matrix $E_1 = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_{n-1})$ such that all the eigenvalues $1 - \lambda_1, \ldots, 1 - \lambda_{n-1}$ of $I - E_1M_1$ lie inside the unit circle. Now write

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix}$$

where m_{nn} is a nonzero scalar (since all the principal minors of M are nonzero). Also let

$$E = \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix}$$

for some complex ϵ_n. Thus

$$I - EM = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix} = \begin{bmatrix} I - E_1M_1 & -E_1M_2 \\ -\epsilon_n M_3 & 1 - \epsilon_n m_{nn} \end{bmatrix}.$$