Towards a sheaf representation of distributed robot tasks

Bernardo Hummes Flores

11 June 2024

I: Motivation

Objectives of this talk

I want to convince you of a few things:

1. Sheaves are a nice way to talk about robots;
2. The Laplacian operator is a nice way to find agreement;
3. I am really talking about robots.

Objectives of this talk

I want to convince you of a few things:

1. Sheaves are a nice way to talk about robots;
2. The Laplacian operator is a nice way to find agreement;
3. I am really talking about robots.

I also want to share with you some of my hopes:

- Agreement problems are a nice way to talk about any problem;
- Sheaves may not be a nice way to talk about tasks,
  but they allow to get an algorithm for free (and this is nice).

For now, just keep in mind that:

A **distributed** mission means that only **local** information is available.

While at the same time:

A **solution** to a mission is a **global** state of the system.

For now, just keep in mind that:

A **distributed** mission means that only **local** information is available.

While at the same time:

A **solution** to a mission is a **global** state of the system.

Assessing global properties from local data is main motivation behind **sheaf theory**.

II: Sheaves

Intuition

Sheaves appeared as a way to talk about **properties** of some **data** defined on a **space** by looking only at **pieces** of the space.

Sheaves appeared as a way to talk about **properties** of some **data** defined on a **space** by looking only at **pieces** of the space.

Space	$\mathbb{R}$
Pieces	Intervals in $\mathbb{R}$
Data	$\mathbb{R}$-valued functions
Property	Is continuous

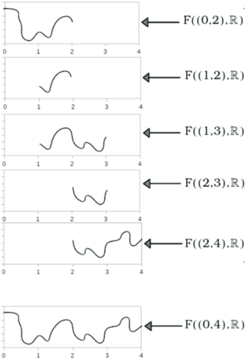


Figure: Continuous $\mathbb{R}$-valued functions defined on $\mathbb{R}$, image from Rosiak 2022.

Intuition

Sheaves appeared as a way to talk about **properties** of some **data** defined on a **space** by looking only at **pieces** of the space.

Space	$\mathbb{R}$
Pieces	Intervals in $\mathbb{R}$
Data	$\mathbb{R}$-valued functions
Property	Is continuous

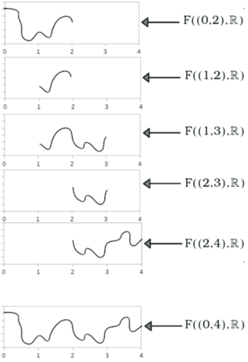


Figure: Continuous $\mathbb{R}$-valued functions defined on $\mathbb{R}$, image from Rosiak 2022.

Sheaf condition

The sheaf structure describes how to to *tie* the data parametrized a space such that it is *coherent*.

Intuition

Sheaves appeared as a way to talk about **properties** of some **data** defined on a **space** by looking only at **pieces** of the space.

Space	$\mathbb{R}$	Set of robots
Pieces	Intervals in $\mathbb{R}$	Subsets of robots
Data	$\mathbb{R}$-valued functions	Gathering positions
Property	Is continuous	Is the same

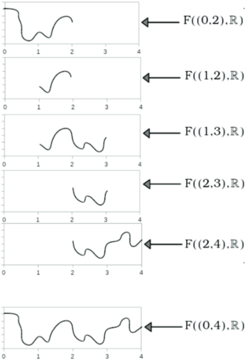


Figure: Continuous $\mathbb{R}$-valued functions defined on $\mathbb{R}$, image from Rosiak 2022.

Sheaf condition

The sheaf structure describes how to to *tie* the data parametrized a space such that it is *coherent*.

Idea

To talk about properties of algorithms executed by groups of robots in terms of *coherence* among interacting robots.

Sheaves appeared as a way to talk about **properties** of some **data** defined on a **space** by looking only at **pieces** of the space.

Space	$\mathbb{R}$	Set of robots	Set of robots
Pieces	Intervals in $\mathbb{R}$	Subsets of robots	Subsets of robots
Data	$\mathbb{R}$-valued functions	Gathering positions	*Task data*
Property	Is continuous	Is the same	*Is coherent*

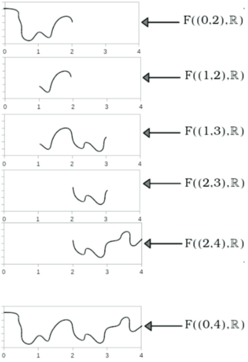


Figure: Continuous $\mathbb{R}$-valued functions defined on $\mathbb{R}$, image from Rosiak 2022.

Sheaf condition

The sheaf structure describes how to to *tie* the data parametrized a space such that it is *coherent*.

Idea

To talk about properties of algorithms executed by groups of robots in terms of *coherence* among interacting robots.

The first requirement is a description of **space** that correctly identifies the pieces and describes how they are connected.

The first requirement is a description of **space** that correctly identifies the pieces and describes how they are connected.

Idea

The goal is to model coherence whenever two robots interact.

- A piece of the space should be delimited by what is local to a robot.
- Intersections should correspond to links allowing for interaction.

The structure that represents this is a **graph**.

Interaction (and graphs)

The first requirement is a description of **space** that correctly identifies the pieces and describes how they are connected.

Idea

The goal is to model coherence whenever two robots interact.

- A piece of the space should be delimited by what is local to a robot.
- Intersections should correspond to links allowing for interaction.

The structure that represents this is a **graph**.

Definition (Graph)

A **graph** $G = (V, E)$ is a set of vertices V and a set of edges$E$. An **edge** is an unordered pairs of vertices.

An edge is said to be incident of the vertices it contains, and it defines an **incidence relation** $v \trianglelefteq e$ whenever $v \in e$.

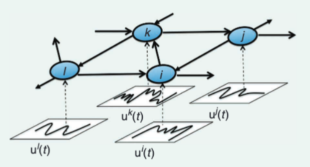


Figure: Depiction by Kia et al. 2019 of a group of communicating agents, together with a reference signal. Also, a depiction of a graph being the base space of some local signal.

This is where it often ends

Oftentimes the analysis of distributed control stops at the graph theoretic analysis, e.g.

- Resilience of multi-agent stems is measured via *robustness of communication graphs*: Dibaji et al. 2018; Luo et al. 2023; Pirani et al. 2023; Shang 2023b; Wen et al. 2023
- The above results are made usable by methods to *preserve the connectivity of communication graphs*: Cortes et al. 2006; Yi et al. 2021; Zavlanos et al. 2011
- Gathering is solved using *averaging algorithms on graphs*: Dutta et al. 2023; Iqbal et al. 2022; Kia et al. 2019; Romero et al. 2024
- Higher order interactions are modeled via *higher-order graphs*: Majhi et al. 2022; Shang 2023a

This is where it often ends

Oftentimes the analysis of distributed control stops at the graph theoretic analysis, e.g.

- Resilience of multi-agent stems is measured via *robustness of communication graphs*: Dibaji et al. 2018; Luo et al. 2023; Pirani et al. 2023; Shang 2023b; Wen et al. 2023
- The above results are made usable by methods to *preserve the connectivity of communication graphs*: Cortes et al. 2006; Yi et al. 2021; Zavlanos et al. 2011
- Gathering is solved using *averaging algorithms on graphs*: Dutta et al. 2023; Iqbal et al. 2022; Kia et al. 2019; Romero et al. 2024
- Higher order interactions are modeled via *higher-order graphs*: Majhi et al. 2022; Shang 2023a

But often is not always

Recent efforts use sheaf theory as the single point of view:

- Opinion dynamics: Ghrist 2022; Hansen and Ghrist 2021; Riess and Ghrist 2022
- Topological data analysis: Curry 2014
- Sensor fusion: Robinson 2017; Robinson 2020
- Dynamical systems: Schultz and Spivak 2019; Schultz, Spivak, and Vasilakopoulou 2020

We need to define how the data attached to the graph should be **coherent**.

We need to define how the data attached to the graph should be **coherent**.

Definition (Sheaf over a graph)

A **sheaf** $\mathcal{F}$ over a graph $G = (V, E)$ is defined by attaching some data to its vertices and edges:

- (*stalk over v*) For each vertex $v \in V$, a set $\mathcal{F}(v)$
- (*stalk over e*) For each edge $e \in E$, a set $\mathcal{F}(e)$
- (*restriction map*) For each pair of vertex and incident edge $v \trianglelefteq e$, a map $\mathcal{F}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$

The restriction map should respect identities, $\mathcal{F}_{v \trianglelefteq v} = \mathrm{id}$

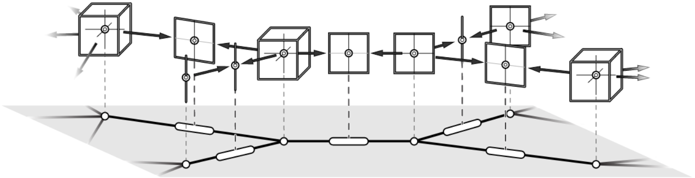


Figure: Cartoon of a sheaf of vector spaces over a graph, from Hansen and Ghrist 2021

Choosing the *data on the sheaf* is choosing the *definition of agreement*.

Global sections

The "task data type" is that assigned to the space through the sheaf.

The structure of the data induces the constraints.

- Agreeing on a value in $\mathbb{R}^n$ without update constraints can be modeled as a sheaf of vector spaces $\mathbb{R}^n$[1].
- Opinion dynamics has been modeled with sheaf of lattices, where agents need to find a lower upper bound to their individual opinions[2].

[1] Hansen, Jakob. “Laplacians Of Cellular Sheaves: Theory And Applications”. 2020.
[2] Riess, Hans. “Lattice Theory in Multi-Agent Systems”. 2023.

The robots that interact should reach the same conclusions.

[2] Cross-section of a sheaf figure from Rosiak 2022

The robots that interact should reach the same conclusions.

The position in space to meet

The position in space of every robot or a target

The assignment of objectives to each robot

The move action to perform

[2] Cross-section of a sheaf figure from Rosiak 2022

The robots that interact should reach the same conclusions.

- **Gathering** The position in space to meet
- **Localization** The position in space of every robot or a target
- **Exploration** The assignment of objectives to each robot
- **Motion planning** The move action to perform

Each task can be seen as a *local* agreement problem. A solution to this agreement problem is a solution to the task.

[2] Cross-section of a sheaf figure from Rosiak 2022

Solutions (and global sections)

The robots that interact should reach the same conclusions.

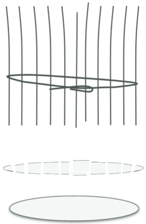
Gathering	The position in space to meet
Localization	The position in space of every robot or a target
Exploration	The assignment of objectives to each robot
Motion planning	The move action to perform

Each task can be seen as a *local* agreement problem. A solution to this agreement problem is a solution to the task.

Local agreement in sheaves has a very specific meaning, built-in from its creation: *sections*!

Definition (Global section)

A **global section** of a sheaf $\mathcal{F}$ on a graph $G = (V, E)$ is a choice of values $x_v \in \mathcal{F}(v)$ for all $v \in V$ such that $\mathcal{F}_{v_1 \trianglelefteq e}(x_{v_1}) = \mathcal{F}_{v_2 \trianglelefteq e}(x_{v_2})$ whenever $(v_1, v_2) \in E$.



[2] Cross-section of a sheaf figure from Rosiak 2022

A *Laplacian operation* is first found as differential operator in the Euclidean space.
It is defined as the divergence ($\nabla\cdot$) of the gradient of a function (∇f). Note that they are adjoint maps.

$$\Delta f = \nabla \cdot \nabla f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

[3] Hansen, Jakob and Ghrist, Robert. “Toward a Spectral Theory of Cellular Sheaves”. 2019.
[4] Ghrist, Robert and Riess, Hans. “Cellular Sheaves of Lattices and the Tarski Laplacian”. 2022.

A *Laplacian operation* is first found as differential operator in the Euclidean space.
It is defined as the divergence ($\nabla \cdot$) of the gradient of a function (∇f). Note that they are adjoint maps.

$$\Delta f = \nabla \cdot \nabla f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

In other settings:

- In sheaves of *vector spaces* we have the the Hodge-Laplacian[3]

$$L_{\mathcal{F}} = \delta^* \delta + \delta \delta^*$$

where δ is a specific restriction map and δ^* the Hermitian adjoint.

[3] Hansen, Jakob and Ghrist, Robert. "Toward a Spectral Theory of Cellular Sheaves". 2019.
[4] Ghrist, Robert and Riess, Hans. "Cellular Sheaves of Lattices and the Tarski Laplacian". 2022.

A *Laplacian operation* is first found as differential operator in the Euclidean space. It is defined as the divergence $(\nabla\cdot)$ of the gradient of a function (∇f). Note that they are adjoint maps.

$$\Delta f = \nabla \cdot \nabla f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

In other settings:

- In sheaves of *vector spaces* we have the the Hodge-Laplacian[3]

$$L_{\mathcal{F}} = \delta^*\delta + \delta\delta^*$$

  where δ is a specific restriction map and δ^* the Hermitian adjoint.

- In sheaves of *lattices* we have the Tarski-Laplacian[4]

$$L\mathbf{x} = \bigwedge_{j \in \mathcal{N}_i} \mathcal{F}(ij)^{+}_{i \trianglelefteq ij} \mathcal{F}(ij)_{j \trianglelefteq ij}(x_j)$$

  where $\mathcal{N}_i$ is the set of neighbours of vertex i and the pair $\mathcal{F}(ij)^{+}_{i \trianglelefteq ij} \mathcal{F}(ij)_{j \trianglelefteq ij}(x_j)$ forms a Galois connection.

[3] Hansen, Jakob and Ghrist, Robert. “Toward a Spectral Theory of Cellular Sheaves”. 2019.

[4] Ghrist, Robert and Riess, Hans. “Cellular Sheaves of Lattices and the Tarski Laplacian”. 2022.

A *Laplacian operation* is first found as differential operator in the Euclidean space. It is defined as the divergence ($\nabla \cdot$) of the gradient of a function (∇f). Note that they are adjoint maps.

$$\Delta f = \nabla \cdot \nabla f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

In other settings:

- In sheaves of *vector spaces* we have the the Hodge-Laplacian[3]

  $$L_{\mathcal{F}} = \delta^* \delta + \delta \delta^*$$

  where δ is a specific restriction map and δ^* the Hermitian adjoint.
- In sheaves of *lattices* we have the Tarski-Laplacian[4]

  $$L\mathbf{x} = \bigwedge_{j \in \mathcal{N}_i} \mathcal{F}(ij)^{+}_{i \trianglelefteq ij} \mathcal{F}(ij)_{j \trianglelefteq ij}(x_j)$$

  where $\mathcal{N}_i$ is the set of neighbours of vertex i and the pair $\mathcal{F}(ij)^{+}_{i \trianglelefteq ij} \mathcal{F}(ij)_{j \trianglelefteq ij}(x_j)$ forms a Galois connection.

A pattern on adjoint maps

In settings where inverses do not necessarily exist, one finds a pair of adjoint maps that *best approximate the local differences*, and then minimizes the differences.

[3] Hansen, Jakob and Ghrist, Robert. "Toward a Spectral Theory of Cellular Sheaves". 2019.

[4] Ghrist, Robert and Riess, Hans. "Cellular Sheaves of Lattices and the Tarski Laplacian". 2022.

The sheaf Laplacian provides a way to evolve an arbitrary initial assignment of data towards the best approximation of agreement.

The sheaf Laplacian provides a way to evolve an arbitrary initial assignment of data towards the best approximation of agreement.

Definition (Sheaf Laplacian)

The **sheaf Laplacian** of a sheaf $\mathcal{F}$ over a graph $G = (V, E)$ is computed in a vertex v for an assignment of data x.

$$L_{\mathcal{F}}(x) = \sum_{v,u \trianglelefteq} \mathcal{F}^{\dagger}_{v_i \trianglelefteq e}(\mathcal{F}_{v_i \trianglelefteq e}(x_i) - \mathcal{F}_{v_j \trianglelefteq e}(x_j)) \quad (1)$$

Where $\mathcal{F}_{v_i \trianglelefteq e}$ is the restriction map of $\mathcal{F}$, and $\mathcal{F}^{\dagger}_{v_i \trianglelefteq e}$ is its adjoint.

III: Robots

I want to distinguish the *physical* problems from the *algorithmic* problems.

[5] Adapted from L Jaulin (Nov. 2023). *Guaranteed Numerical Methods to Secure a Zone with Autonomous Robots*.

I want to distinguish the *physical* problems from the *algorithmic* problems.

Definition (Robot with memory[5])

A **robot with memory** has sensors, actuators, an intelligence and a memory. It has an ontic state $\mathbf{x}$ and an epistemic state μ. It is capable of making observations and evolving its states.

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \qquad \text{(ontic evolution)} \qquad (2)$$

$$\mathbf{y}(t) = g(\mathbf{x}(t)) \qquad \text{(observation)} \qquad (3)$$

$$\dot{\mu}(t) = \varphi(\mu(t), \mathbf{y}(t)) \qquad \text{(epistemic evolution)} \qquad (4)$$

$$\mathbf{u}(t) = h(\mu(t), t) \qquad \text{(control)} \qquad (5)$$

Note that the control steers the ontic state, while observations steer the epistemic state.

This definition displays the case of a single robot as a dynamical system, with state $Z = (\mathbf{x}, \mu)$.

[5] Adapted from L Jaulin (Nov. 2023). *Guaranteed Numerical Methods to Secure a Zone with Autonomous Robots*.

Multiple robots with memory

Interaction requires a notion of *perception* and *communication*.

Interaction requires a notion of *perception* and *communication*.

Definition (Communication and perception of robots)

A robot i obtains an estimation of the ontic state of robot j through **perception** and of its epistemic state through **communication**.

$$\widehat{\mathbf{x}}_{ij}(t) = \eta(\mathbf{x}_j(t)) \qquad \text{(perception)} \qquad (6)$$

$$\widehat{\mu}_{ij}(t) = \lambda(\mu_j(t)) \qquad \text{(communication)} \qquad (7)$$

Communication and perception add to Eq. 4 the information about other robots.

$$\dot{\mu}_i(t) = \varphi(\overbrace{\mu_i(t)}^{\text{current epistemic state}}, \underbrace{\mathbf{y}_i(t)}_{\text{estimation of own ontic state}}, \overbrace{\{\widehat{\mathbf{x}}_{ij}(t)\}_{j\neq i}}^{\text{estimation of ontic state of others}}, \underbrace{\{\widehat{\mu}_{ij}(t)\}_{j\neq i}}_{\text{estimation of epistemic state of others}}) \qquad (8)$$

Interaction requires a notion of *perception* and *communication*.

Definition (Communication and perception of robots)

A robot i obtains an estimation of the ontic state of robot j through **perception** and of its epistemic state through **communication**.

$$\widehat{\mathbf{x}}_{ij}(t) = \eta(\mathbf{x}_j(t)) \qquad \text{(perception)} \qquad (6)$$

$$\widehat{\mu}_{ij}(t) = \lambda(\mu_j(t)) \qquad \text{(communication)} \qquad (7)$$

Communication and perception add to Eq. 4 the information about other robots.

$$\dot{\mu}_i(t) = \varphi(\overbrace{\mu_i(t)}^{\text{current epistemic state}}, \underbrace{\mathbf{y}_i(t)}_{\text{estimation of own ontic state}}, \overbrace{\{\widehat{\mathbf{x}}_{ij}(t)\}_{j\neq i}}^{\text{estimation of ontic state of others}}, \underbrace{\{\widehat{\mu}_{ij}(t)\}_{j\neq i}}_{\text{estimation of epistemic state of others}}) \qquad (8)$$

Remark

To have a range for communication or perception is a special case, where such thresholds are added to Eq. 6 and 7.

IV: Sheaves in robotics

Distributed robot tasks combine problems of *distributed control* with those of *distributed computing*.

Distributed robot tasks combine problems of *distributed control* with those of *distributed computing*.

The separation of a robot's state in *ontic* and *epistemic* allows for the individual treatment of each.

- A *distributed computing system* can be recovered by setting the ontic evolution to obey a simplistic point-mass dynamics, where it is essentially controlled by the memory.
- A traditional *feedback controller* can be obtained from the epistemic evolution, leading to the typical depiction of robots in control theory.

Distributed robot tasks combine problems of *distributed control* with those of *distributed computing*.

The separation of a robot's state in *ontic* and *epistemic* allows for the individual treatment of each.

- A *distributed computing system* can be recovered by setting the ontic evolution to obey a simplistic point-mass dynamics, where it is essentially controlled by the memory.
- A traditional *feedback controller* can be obtained from the epistemic evolution, leading to the typical depiction of robots in control theory.
- A *hybrid dynamic system* can be used to describe the different nature of the time domains of the physical and computational aspects of the robots.
- A *switched dynamic system* could be used to represent the asynchronicity of updates, capturing the uncertainty in the communication network.

Distributed robot tasks combine problems of *distributed control* with those of *distributed computing*.

The separation of a robot's state in *ontic* and *epistemic* allows for the individual treatment of each.

- A *distributed computing system* can be recovered by setting the ontic evolution to obey a simplistic point-mass dynamics, where it is essentially controlled by the memory.
- A traditional *feedback controller* can be obtained from the epistemic evolution, leading to the typical depiction of robots in control theory.
- A *hybrid dynamic system* can be used to describe the different nature of the time domains of the physical and computational aspects of the robots.
- A *switched dynamic system* could be used to represent the asynchronicity of updates, capturing the uncertainty in the communication network.

Remark

Overall, this distinction is semantic and serves mostly to provide an intuition on what changes can be done in a robot model.

The *Laplacian operator* on a task sheaf provides an *approximate solution to the task*.

[6] Fauconnier, Hugues et al. "Non-Negotiating Distributed Computing". 2023.

The *Laplacian operator* on a task sheaf provides an *approximate solution to the task*.

The idea: use the Laplacian as the epistemic evolution

Each robot i is associated to a vertex v_i and its epistemic state μ_i is its assignment of data on the sheaf

$$\mathcal{F}(v_i) = \mu_i$$

Its *dynamics is given by the sheaf Laplacian*, a dynamical system representing the memory.

$$\dot{\mu}(t) = -L_{\mathcal{F}}(v_i)$$

[6] Fauconnier, Hugues et al. "Non-Negotiating Distributed Computing". 2023.

A Laplacian as the algorithm

The *Laplacian operator* on a task sheaf provides an *approximate solution to the task*.

The idea: use the Laplacian as the epistemic evolution

Each robot i is associated to a vertex v_i and its epistemic state μ_i is its assignment of data on the sheaf

$$\mathcal{F}(v_i) = \mu_i$$

Its *dynamics is given by the sheaf Laplacian*, a dynamical system representing the memory.

$$\dot{\mu}(t) = -L_{\mathcal{F}}(v_i)$$

Theorem (Th. 4.1 in Hansen and Ghrist 2021)

Solutions $x(t)$ to the heat equation $\frac{\mathrm{d}x}{\mathrm{d}(}t) = -\alpha L_{\mathcal{F}x}$, $\alpha > 0$ on $x \in C^0(G; \mathcal{F})$ converge as $t \to \infty$ to the orthogonal projection of $x(0)$ onto $H^0(G; \mathcal{F})$.

[6] Fauconnier, Hugues et al. "Non-Negotiating Distributed Computing". 2023.

The *Laplacian operator* on a task sheaf provides an *approximate solution to the task*.

The idea: use the Laplacian as the epistemic evolution

Each robot i is associated to a vertex v_i and its epistemic state μ_i is its assignment of data on the sheaf

$$\mathcal{F}(v_i) = \mu_i$$

Its *dynamics is given by the sheaf Laplacian*, a dynamical system representing the memory.

$$\dot{\mu}(t) = -L_{\mathcal{F}}(v_i)$$

Theorem (Th. 4.1 in Hansen and Ghrist 2021)

Solutions $x(t)$ to the heat equation $\frac{\mathrm{d}x}{\mathrm{d}(}t) = -\alpha L_{\mathcal{F}x}$, $\alpha > 0$ on $x \in C^0(G; \mathcal{F})$ converge as $t \to \infty$ to the orthogonal projection of $x(0)$ onto $H^0(G; \mathcal{F})$.

Universality of approximate agreement

This usage of the sheaf Laplacian hints at the notion of the *universality of approximation algorithms*, folklore in distributed computing and stated for full-information protocols[6] (roughly computing models with memory).

[6] Fauconnier, Hugues et al. "Non-Negotiating Distributed Computing". 2023.

Assumption (Point-mass dynamics)

Robots have point-mass dynamics, as in Definition 8.

Definition (Point-mass dynamics)

A **point-mass dynamical system** is entirely described by the force applied to it. That is, the input $\mathbf{u}(t)$ controls the system directly.

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{u}(t)$$

Assumption (Accurate self-estimation)

Robots have access to their own ontic states, i.e. $g(\mathbf{x}_i) = \mathbf{x}_i$.

Assumption (Accurate perception)

Robots can accurately measure the ontic state of others through perception. For robot i perceiving robot j, $\widehat{\mathbf{x}}_{ij} = \eta(\mathbf{x}_j) = \mathbf{x}_j$

Assumption (Accurate communication)

Robots can accurately obtain the epistemic state of others through communication. For robot i receiving a communication from robot j, $\widehat{\mu}_{ij} = \lambda(\mu_j) = \mu_j$

Example: full-information approximate gathering (1)

Consider a group of N robots with memory modeled as follows.

Example: full-information approximate gathering (1)

Consider a group of N robots with memory modeled as follows.

- Their ontic state $\mathbf{x} \in \mathbb{R}^2$ represents their physical coordinates.
- Their epistemic state $\mu \in \mathbb{R}^2$ represents their current estimation of a gathering point.

Example: full-information approximate gathering (1)

Consider a group of N robots with memory modeled as follows.

- Their ontic state $\mathbf{x} \in \mathbb{R}^2$ represents their physical coordinates.
- Their epistemic state $\mu \in \mathbb{R}^2$ represents their current estimation of a gathering point.
- The epistemic state is initialized with the current coordinates, available through Assumption 2.

$$\mu_i(0) = \mathbf{x}_i(0)$$

Example: full-information approximate gathering (1)

Consider a group of N robots with memory modeled as follows.

- Their ontic state $\mathbf{x} \in \mathbb{R}^2$ represents their physical coordinates.
- Their epistemic state $\mu \in \mathbb{R}^2$ represents their current estimation of a gathering point.
- The epistemic state is initialized with the current coordinates, available through Assumption 2.

$$\mu_i(0) = \mathbf{x}_i(0)$$

- The control is defined to follow that of the epistemic state.

$$\mathbf{u}_i(t) = \dot{\mu}_i(t)$$

Example: full-information approximate gathering (1)

Consider a group of N robots with memory modeled as follows.

- Their ontic state $\mathbf{x} \in \mathbb{R}^2$ represents their physical coordinates.
- Their epistemic state $\mu \in \mathbb{R}^2$ represents their current estimation of a gathering point.
- The epistemic state is initialized with the current coordinates, available through Assumption 2.

$$\mu_i(0) = \mathbf{x}_i(0)$$

- The control is defined to follow that of the epistemic state.

$$\mathbf{u}_i(t) = \dot{\mu}_i(t)$$

- The ontic evolution follows a point-mass dynamics as by Assumption 1.

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t)$$

Example: full-information approximate gathering (1)

Consider a group of N robots with memory modeled as follows.

- Their ontic state $\mathbf{x} \in \mathbb{R}^2$ represents their physical coordinates.
- Their epistemic state $\mu \in \mathbb{R}^2$ represents their current estimation of a gathering point.
- The epistemic state is initialized with the current coordinates, available through Assumption 2.

$$\mu_i(0) = \mathbf{x}_i(0)$$

- The control is defined to follow that of the epistemic state.

$$\mathbf{u}_i(t) = \dot{\mu}_i(t)$$

- The ontic evolution follows a point-mass dynamics as by Assumption 1.

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t)$$

- The estimation of other robots is exchanged accurately due to Assumption 4.

$$\widehat{\mu}_{ij}(t) = \mu_j(t)$$

Example: full-information approximate gathering (1)

Consider a group of N robots with memory modeled as follows.

- Their ontic state $\mathbf{x} \in \mathbb{R}^2$ represents their physical coordinates.
- Their epistemic state $\mu \in \mathbb{R}^2$ represents their current estimation of a gathering point.
- The epistemic state is initialized with the current coordinates, available through Assumption 2.

$$\mu_i(0) = \mathbf{x}_i(0)$$

- The control is defined to follow that of the epistemic state.

$$\mathbf{u}_i(t) = \dot{\mu}_i(t)$$

- The ontic evolution follows a point-mass dynamics as by Assumption 1.

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t)$$

- The estimation of other robots is exchanged accurately due to Assumption 4.

$$\widehat{\mu}_{ij}(t) = \mu_j(t)$$

- The epistemic evolution is defined as the Laplacian of the associated sheaf $\mathcal{F}$.

$$\dot{\mu}_i(t) = L_{\mathcal{F}}(v_i)$$

Example: full-information approximate gathering (1)

Consider a group of N robots with memory modeled as follows.

- Their ontic state $\mathbf{x} \in \mathbb{R}^2$ represents their physical coordinates.
- Their epistemic state $\mu \in \mathbb{R}^2$ represents their current estimation of a gathering point.
- The epistemic state is initialized with the current coordinates, available through Assumption 2.

$$\mu_i(0) = \mathbf{x}_i(0)$$

- The control is defined to follow that of the epistemic state.

$$\mathbf{u}_i(t) = \dot{\mu}_i(t)$$

- The ontic evolution follows a point-mass dynamics as by Assumption 1.

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t)$$

- The estimation of other robots is exchanged accurately due to Assumption 4.

$$\widehat{\mu}_{ij}(t) = \mu_j(t)$$

- The epistemic evolution is defined as the Laplacian of the associated sheaf $\mathcal{F}$.

$$\dot{\mu}_i(t) = L_{\mathcal{F}}(v_i)$$

The robot model is rewritten as

$$\begin{aligned} \dot{\mathbf{x}}_i(t) &= & \mathbf{u}_i(t) &= & \dot{\mu}_i(t) \\ \dot{\mu}_i(t) &= \varphi(\mu_i(t), \mathbf{x}_i(t), \{\mathbf{x}_j(t)\}_{i \neq j}, \{\mu_j(t)\}_{i \neq j}) &&= & -L_{\mathcal{F}}(v_i) \end{aligned}$$

Example: full-information approximate gathering (2)

The sheaf $\mathcal{F}$ is defined over a communication graph $G = (V, E)$ as $\underline{\mathbb{R}^2}$, a constant sheaf of vector spaces $\mathbb{R}^2$.

[7] Hansen and Ghrist 2021, Theorem 4.1

The sheaf $\mathcal{F}$ is defined over a communication graph $G = (V, E)$ as $\underline{\mathbb{R}^2}$, a constant sheaf of vector spaces $\mathbb{R}^2$.

- To each vertex $v \in V$ is associated the space of possible estimations of the gathering point of the corresponding robot.
- To each edge $e \in E$ between vertices v_i and v_j is associated the compatibility space between two interacting robots.

[7] Hansen and Ghrist 2021, Theorem 4.1

The sheaf $\mathcal{F}$ is defined over a communication graph $G = (V, E)$ as $\underline{\mathbb{R}^2}$, a constant sheaf of vector spaces $\mathbb{R}^2$.

- To each vertex $v \in V$ is associated the space of possible estimations of the gathering point of the corresponding robot.
- To each edge $e \in E$ between vertices v_i and v_j is associated the compatibility space between two interacting robots.

The sheaf Laplacian (Eq. 1) is rewritten in this context as

$$L_{\mathcal{F}}(\mu_i) = -\sum_{v_i, v_j \trianglelefteq e} \overbrace{\mathrm{id}}^{\mathcal{F}^{\dagger}_{v_i \trianglelefteq e}}(\overbrace{\mathrm{id}}^{\mathcal{F}_{v_i \trianglelefteq e}}(\underbrace{\mu_i}_{\text{robot } i \text{ has access to own memory}}) - \overbrace{\mathrm{id}}^{\mathcal{F}_{v_j \trianglelefteq e}}(\underbrace{\mu_j}_{\lambda = \mathrm{id}, \text{ by Assumption 4}}))$$

The system described by $\dot{\mathbf{x}}(t) = -\alpha L_{\mathcal{F}}(x)$ converges to the global sections of $\mathcal{F}$ [7].

[7] Hansen and Ghrist 2021, Theorem 4.1

The sheaf $\mathcal{F}$ is defined over a communication graph $G = (V, E)$ as $\underline{\mathbb{R}^2}$, a constant sheaf of vector spaces $\mathbb{R}^2$.

- To each vertex $v \in V$ is associated the space of possible estimations of the gathering point of the corresponding robot.
- To each edge $e \in E$ between vertices v_i and v_j is associated the compatibility space between two interacting robots.

The sheaf Laplacian (Eq. 1) is rewritten in this context as

$$L_{\mathcal{F}}(\mu_i) = -\sum_{v_i, v_j \trianglelefteq e} \overbrace{\mathrm{id}}^{\mathcal{F}^{\dagger}_{v_i \trianglelefteq e}} \big(\overbrace{\mathrm{id}}^{\mathcal{F}_{v_i \trianglelefteq e}} (\underbrace{\mu_i}_{\text{robot } i \text{ has access to own memory}}) - \overbrace{\mathrm{id}}^{\mathcal{F}_{v_j \trianglelefteq e}} (\underbrace{\mu_j}_{\lambda = \mathrm{id}, \text{ by Assumption 4}}) \big)$$

The system described by $\dot{\mathbf{x}}(t) = -\alpha L_{\mathcal{F}}(x)$ converges to the global sections of $\mathcal{F}$ [7].

Remark

Assumption 3 is never used, as the only comparisons are made through the accurately communicated estimations.

[7] Hansen and Ghrist 2021, Theorem 4.1

A very simple one :)

V: Final considerations

Each aspect of a robot task is reflected in a different aspect of the sheaf model.

- Changes in *observation, communication and perception* usually reflect to changes in the *restriction maps*;
- Changes in the *interaction model* correspond to changes in the *base space* (now graphs);
- Changes in the *definition of agreement* correspond to a proper choice of *task data*.

Other types of tasks

Each aspect of a robot task is reflected in a different aspect of the sheaf model.

- Changes in *observation, communication and perception* usually reflect to changes in the *restriction maps*;
- Changes in the *interaction model* correspond to changes in the *base space* (now graphs);
- Changes in the *definition of agreement* correspond to a proper choice of *task data*.

Guesses on possible modeling choices

Problem variation	Modeling change
Vision based sensing	*Restriction maps* adapted to projection matrices
Malicious communication or unreliable sensors	*Restriction maps* and task data adapted to uncertainty
Obstacles	*Task data* with global sections that cannot be distinguished according to intuition of equivalent paths
Different local coordinates	*Task data* or restriction maps, mainly interpretation
Exploration	*Task data*, such as preference lattices to agree on a coherent partition
Tracking of a target	*Task data*, re-interpreting vector spaces for keeping track of all information to be agreed on
Dynamic constraints to behavior	*Task data* with more structure, e.g. vector fields
Mapping	*Task data* to represent and merge non-conflicting maps

Many paths come from the sheaf theoretic point of view:

- *Exploration*: agreement of some preference structure could offer a unified view for both converging and diverging tasks[8].
- *Robot dynamics*: it may be possible to pack all dynamical information within the sheaf with vector fields (or similar).
- *Interaction dynamics*: sheaf morphisms could shed a light in the consequences of time-varying visibility constraints and graphs[9].
- *Asynchronicity*: a Laplacian with firing sequences could be related to the switched systems theory used to model asynchronous behavior[10].
- *Hybrid systems*: the description of algorithms as differential equations is a bit sketchy and more convincing arguments should be found[11].
- *Other tasks* from before, as a lot of justification is still needed.

---

[8] Alcántara, Manuel et al. "The Topology of Look-Compute-Move Robot Wait-Free Algorithms with Hard Termination". 2019.

[9] Hansen, Jakob and Ghrist, Robert. "Toward a Spectral Theory of Cellular Sheaves". 2019.

[10] Lee, Kooktae. *Asynchronous Distributed Averaging: A Switched System Framework for Average Error Analysis*. 2020.

[11] Graça, Daniel S. et al. "Computability with Polynomial Differential Equations". 2008.

- Sheaf theory is a language for distributed information
- The Laplacian operator generalizes computing approximate agreement
- The robot with memory formalism connects the sheaf language to the dynamical systems view of robots
- And some insight can come from seeing problems as a lack of agreement

Thank you for your attention!

At the center of this work is the characterization of **robot tasks**.

Definition (Robot task)

A robot task $(\mathcal{I}, \mathcal{O}, \Delta)$ represents a mission assigned to multiple robots, here assumed identical.
They are expressed via a collection of *initial configurations* $\mathcal{I}$, a set of acceptable *final configurations* $\mathcal{O}$ and a relationship Δ between $\mathcal{I}$ and $\mathcal{O}$.

Those are called *input-output tasks* in the distributed computing community when referring to processes in a network.

Example (Gathering)

The gathering task may allow robots to start in any configuration, but it restricts the final configurations to only the ones where all robots are present in the same position. The Δ relation associates to each initial configuration one or more final gathering points that are acceptable, according to the mission restrictions.

In order to study robot tasks we think about the related algorithms, which I as define follows.

Definition (Robot algorithm)

An algorithm $\mathcal{A}$ that solves a desired robot task $\mathcal{T}$ must satisfy three properties:

1. `correctness`: The algorithm only proposes correct outputs.
2. `validity`: The algorithm only proposes valid outputs.
3. `termination`: The algorithms terminates.

A *correct* output is a configuration that satisfies the objective of the mission.
A *valid* output is a configuration that does not violate restrictions placed upon the possible solutions.

The `correctness` alongside `termination` guarantee that the mission will be satisfied in finite time, while `validity` assures that it will not present any undesired behavior.

Example (Gathering with termination)

In the gathering task, we have **correctness** respected only in configurations that have **all robots in the same position**.
Validity requires that if robots are **already gathered** in a starting configuration, they **should not** terminate gathered in any other configuration.
In the **input-output description**, correctness affects the collection of final configurations $\mathcal{O}$, and validity the Δ relation.

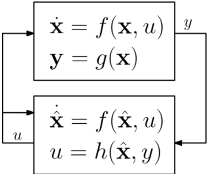
The robot with memory model generalizes the traditional dynamical system model of a robot with a controller.
Let a robot be a dynamical system with observer.

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = g(\mathbf{x}) \end{cases}$$

An output feedback controller simulates of the system $\mathbf{x}$ in order to obtain an estimation of the state $\widehat{\mathbf{x}}$ and regulate the input $\mathbf{u}$ of the original system.

$$\begin{cases} \dot{\widehat{\mathbf{x}}} = f(\widehat{\mathbf{x}}, \mathbf{u}) \\ \mathbf{u} = h(\widehat{\mathbf{x}}, \mathbf{y}) \end{cases}$$

It corresponds to setting the epistemic state to be an estimation of its ontic state, $\mu(t) = \widehat{\mathbf{x}}(t)$, by following the same evolution, $\varphi = f$.
The epistemic evolution has acces to the previous input $\mathbf{u}$ produced. The control accesses the observation $\mathbf{y}$, in order to compare with its own, i.e. $\mathbf{u} = h(\widehat{\mathbf{x}}, \mathbf{y}) = \mathbf{y} - g(\widehat{\mathbf{x}})$.
Both changes do not pose problems, as they are present in the same unit.



An important and unrealistic requirement for defining a Laplacian is that the graph is static. Our communication graph is induced by the positions of the robots, which are constantly changing.

For a given graph homeomorphism $f : X \to Y$, we can define the following (Hansen and Ghrist 2019, Def. 2.10, 2.11).

Definition (Pullback)

The pullback $f^*\mathcal{F}$ of a sheaf over Y is a sheaf over X with

- $f^*\mathcal{F}(v) = \mathcal{F}(f(v))$ and $f^*\mathcal{F}(e) = \mathcal{F}(f(e))$;
- $(f^*\mathcal{F})_{v \trianglelefteq e} = (\mathcal{F})_{f(v) \trianglelefteq f(e)}$.

Definition (Pushforward)

The pushforward $f_*\mathcal{F}$ of a sheaf over X is a sheaf over Y with

- $f_*\mathcal{F}(v)$ is the limit $lim_{v \trianglelefteq f(e)} \mathcal{F}(e)$;
- $(f_*\mathcal{F})_{v \trianglelefteq e}$ is induced by $\mathcal{F}_{v \trianglelefteq e}$ with the restriction above.

Note that the Laplacian is *invariant* under pushforwards of locally injective cell morphisms, when each point x has a neighborhood that is mapped injectively, as in ibid., Prop. 5.10 . This means that if we can restrict changes of the graph with a certain class of maps, we *can preserve convergence properties*.

References I

Alcántara, Manuel et al. (June 2019). “The Topology of Look-Compute-Move Robot Wait-Free Algorithms with Hard Termination”. In: *Distributed Computing* 32.3. doi: 10.1007/s00446-018-0345-3.

Cortes, J., S. Martinez, and F. Bullo (Aug. 2006). “Robust Rendezvous for Mobile Autonomous Agents via Proximity Graphs in Arbitrary Dimensions”. In: *IEEE Transactions on Automatic Control* 51.8. doi: 10.1109/TAC.2006.878713.

Curry, Justin M (2014). “Sheaves, Cosheaves and Applications”. PhD thesis. University of Pennsilvania.

Dibaji, Seyed Mehran, Hideaki Ishii, and Roberto Tempo (Aug. 2018). “Resilient Randomized Quantized Consensus”. In: *IEEE Transactions on Automatic Control* 63.8. doi: 10.1109/TAC.2017.2771363.

Dutta, Maitreyee et al. (2023). “A δ-Persistently-Exciting Formation Controller for Non-Holonomic Systems Over Directed Graphs”. In: *IEEE Control Systems Letters* 7. doi: 10.1109/LCSYS.2023.3287951.

Fauconnier, Hugues et al. (2023). "Non-Negotiating Distributed Computing". In.

Ghrist, Robert (Dec. 2022). "Network SheafModels for Social Information Dynamics". In: *2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC)*. doi: 10.1109/CIC56439.2022.00015.

Ghrist, Robert and Hans Riess (2022). "Cellular Sheaves of Lattices and the Tarski Laplacian". In: *Homology, Homotopy and Applications* 24.1. doi: 10.4310/HHA.2022.v24.n1.a16.

Graça, Daniel S., Manuel L. Campagnolo, and Jorge Buescu (Mar. 2008). "Computability with Polynomial Differential Equations". In: *Advances in Applied Mathematics* 40.3. doi: 10.1016/j.aam.2007.02.003.

Hansen, Jakob (2020). "Laplacians Of Cellular Sheaves: Theory And Applications". PhD thesis. University of Pennsilvania.

Hansen, Jakob and Robert Ghrist (Dec. 2019). "Toward a Spectral Theory of Cellular Sheaves". In: *Journal of Applied and Computational Topology* 3.4. doi: 10.1007/s41468-019-00038-7.

— (Jan. 2021). "Opinion Dynamics on Discourse Sheaves". In: *SIAM Journal on Applied Mathematics* 81.2. doi: 10.1137/20M1341088.

Iqbal, Muhammad, Zhihua Qu, and Azwirman Gusrialdi (2022). "Resilient Dynamic Average-Consensus of Multiagent Systems". In: *IEEE Control Systems Letters* 6. doi: 10.1109/LCSYS.2022.3185800.

Jaulin, L (Nov. 2023). *Guaranteed Numerical Methods to Secure a Zone with Autonomous Robots*.

Kia, Solmaz S. et al. (June 2019). "Tutorial on Dynamic Average Consensus: The Problem, Its Applications, and the Algorithms". In: *IEEE Control Systems* 39.3. doi: 10.1109/MCS.2019.2900783.

Lee, Kooktae (June 2020). *Asynchronous Distributed Averaging: A Switched System Framework for Average Error Analysis*. arXiv: 2006.01925 [cs, eess].

Luo, Xiaoyuan et al. (Nov. 2023). "Resilient Coordination of Nonlinear Uncertain Lagrangian Systems With Adversarial Agents: A Norm-Based Approach". In: *IEEE Transactions on Industrial Informatics* 19.11. doi: 10.1109/TII.2023.3245194.

Majhi, Soumen, Matjaž Perc, and Dibakar Ghosh (Mar. 2022). "Dynamics on Higher-Order Networks: A Review". In: *Journal of The Royal Society Interface* 19.188. doi: 10.1098/rsif.2022.0043.

Pirani, Mohammad, Aritra Mitra, and Shreyas Sundaram (Nov. 2023). "Graph-Theoretic Approaches for Analyzing the Resilience of Distributed Control Systems: A Tutorial and Survey". In: *Automatica* 157. doi: 10.1016/j.automatica.2023.111264.

Riess, Hans (Apr. 2023). "Lattice Theory in Multi-Agent Systems". PhD thesis. arXiv. arXiv: 2304.02568 [cs, eess, math].

- Riess, Hans and Robert Ghrist (Dec. 2022). "Diffusion of Information on Networked Lattices by Gossip". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. doi: 10.1109/CDC51059.2022.9992539.

- Robinson, Michael (July 2017). "Sheaves Are the Canonical Data Structure for Sensor Integration". In: *Information Fusion* 36. doi: 10.1016/j.inffus.2016.12.002.

- — (June 2020). "Assignments to Sheaves of Pseudometric Spaces". In: *Compositionality* 2. doi: 10.32408/compositionality-2-2.

- Romero, Jose Guadalupe et al. (Jan. 2024). "Global Consensus-Based Formation Control of Nonholonomic Mobile Robots With Time-Varying Delays and Without Velocity Measurements". In: *IEEE Transactions on Automatic Control* 69.1. doi: 10.1109/TAC.2023.3264744.

- Rosiak, Daniel (2022). *Sheaf Theory through Examples*. Cambridge,MA London, UK: The MIT Press. isbn: 978-0-262-54215-9.

Schultz, Patrick and David I. Spivak (2019). "Temporal Type Theory: A Topos-Theoretic Approach to Systems and Behavior". In: Progress in Computer Science and Applied Logic 29. doi: 10.1007/978-3-030-00704-1.

Schultz, Patrick, David I. Spivak, and Christina Vasilakopoulou (Feb. 2020). "Dynamical Systems and Sheaves". In: *Applied Categorical Structures* 28.1. doi: 10.1007/s10485-019-09565-x.

Shang, Yilun (Feb. 2023a). "Non-Linear Consensus Dynamics on Temporal Hypergraphs with Random Noisy Higher-Order Interactions". In: *Journal of Complex Networks* 11.2. doi: 10.1093/comnet/cnad009.

— (May 2023b). "Resilient Consensus in Continuous-Time Networks with l-Hop Communication and Time Delay". In: *Systems & Control Letters* 175. doi: 10.1016/j.sysconle.2023.105509.

Wen, Guanghui et al. (Nov. 2023). "Joint Robustness of Time-Varying Networks and Its Applications to Resilient Consensus". In: *IEEE Transactions on Automatic Control* 68.11. doi: 10.1109/TAC.2023.3237493.

Yi, Sha, Wenhao Luo, and Katia Sycara (May 2021). "Distributed Topology Correction for Flexible Connectivity Maintenance in Multi-Robot Systems". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE. isbn: 978-1-72819-077-8. doi: 10.1109/ICRA48506.2021.9561721.

Zavlanos, M. M., M. B. Egerstedt, and G. J. Pappas (Sept. 2011). "Graph-Theoretic Connectivity Control of Mobile Robot Networks". In: *Proceedings of the IEEE* 99.9. doi: 10.1109/JPROC.2011.2157884.