Ideas for manufacturing consensus in distributed robotics

Bernardo Hummes Flores

April 19, 2023

Outline

- Introduction
  context and motivation
- Dynamic average consensus*
  problem formulation, from static to dynamic, improvements and examples
- Manufacturing consensus
  higher order Laplacians and the Hodge theorem
- Conclusion

*Notes from 2019 - S. Kia et al - Tutorial on dynamic average consensus

Outline

- Introduction
  context and motivation
- Dynamic average consensus*
  problem formulation, from static to dynamic, improvements and examples
- Manufacturing consensus
  higher order Laplacians and the Hodge theorem
- Conclusion

*Notes from 2019 - S. Kia et al - Tutorial on dynamic average consensus

Context

The objective is for a decentralized group of cooperating agents capable of communication with neighbors to track a time varying locally available reference signal

Motivation

We desire certain properties for the algorithmic solution

- Scalability
- Robustness
- Correctness

That are lacking in naive extensions
of traditional methods, such as

- Flooding - all communications then all computations
- Static algorithms - consensus between samples

Outline

- Introduction
  context and motivation

- Dynamic average consensus*
  problem formulation, from static to dynamic,
  improvements and examples

- Manufacturing consensus
  higher order Laplacians and The Hodge theorem

- Conclusion

*Notes from 2019 - S. Kia et al - Tutorial on dynamic average consensus

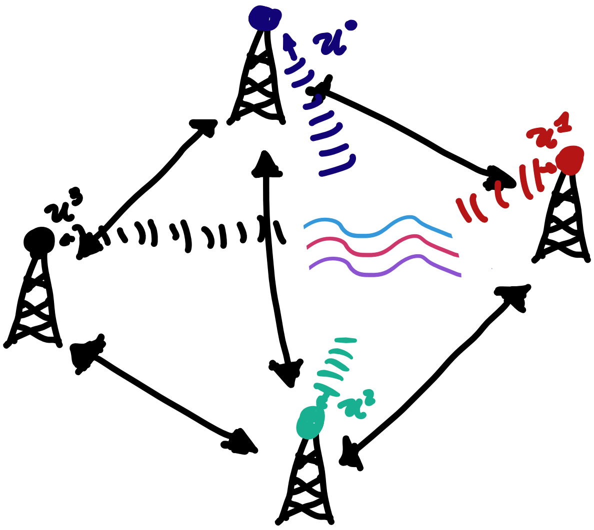
Dynamic average consensus

Problem formulation

- Each of the N sensors observes some reference signal

$$u^i(t): \mathbb{R}^+ \to \mathbb{R}, \quad i \in \{1, .., N\} \qquad (1)$$

- Where the communication is described by a graph G



adjacency matrix

$$A^G = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Dynamic average consensus

Problem formulation

- We want their estimation of the signal to converge to the average observation

$$x^i(t) \xrightarrow[t\to\infty]{} u^{avg}(t) = \frac{1}{N}\sum_{i=1}^{N} u^i(t) \qquad (2)$$

- We have to design a control to steer their state

$$\dot{x}^i = c^i\left(J^i(t), \{I^j(t)\}_{j\in N^i_{out}}\right) \qquad (3)$$

using only their local variables and of their neighbours

From static to dynamic:

Dynamic average consensus

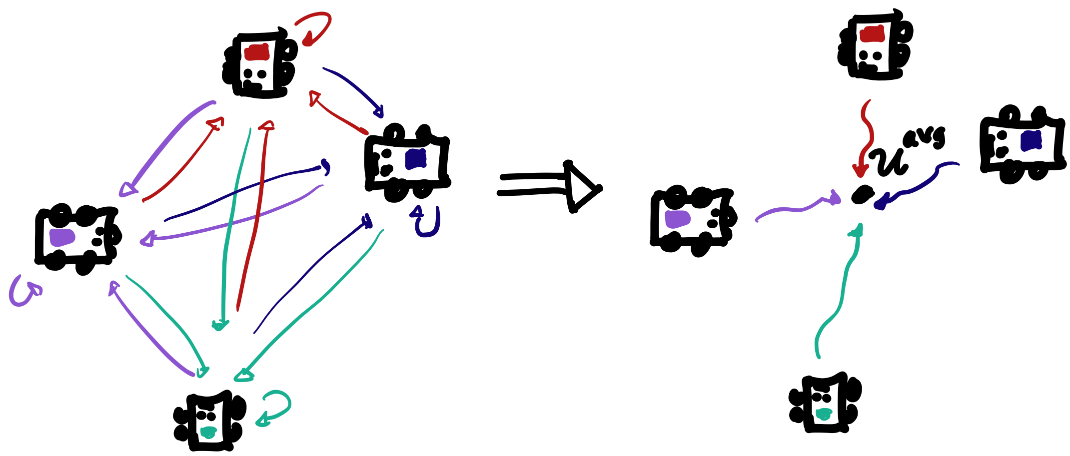
Static Laplacian average consensus

- We consider robots R^i capable of measuring each other's positions and communicating with neighbors

$$u^i(t) = \mathbb{R}^+ \longrightarrow \mathbb{R}^{N \times M}$$

- With the objective of estimating a point of gathering.

$$X^i(t) \in \mathbb{R}^M \longrightarrow u^{avg}(t)$$



From static to dynamic:

Dynamic average consensus

Static Laplacian average consensus

- This measurement only happens once at the start

$$x^i(0) = u^i(0)$$

- They will meet by averaging the neighboring estimates

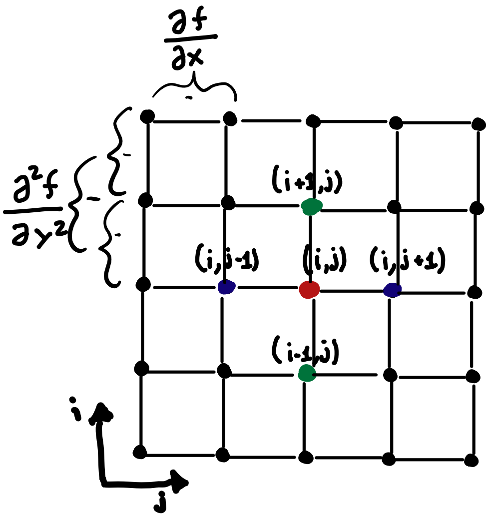
$$\dot{x}^i(t) = -\sum_{j=1}^{N} a_{ij} \left(x^i(t) - x^j(t) \right)$$

- This can be rewriten as the Laplacian of the communication graph G

$$\dot{x}^i(t) = -\Delta_G x(t) \,, \quad (\Delta_G)_{i,j} = \begin{cases} -1 & \text{if } a_{ij} = 1 \\ \mathrm{Deg}(G_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Example: graph Laplacian in a grid

The graph Laplacian applied to a grid graph can be seen similarly to the Euclidean plane discretized



- In first order we have

$$\frac{\partial f}{\partial x}(i,j) \sim f(i,j+1) - f(i,j)$$

- In second order we have

$$\frac{\partial^2 f}{\partial x^2}(i,j) \sim (f(i,j+1) - f(i,j)) - (f(i,j) - f(i,j-1))$$

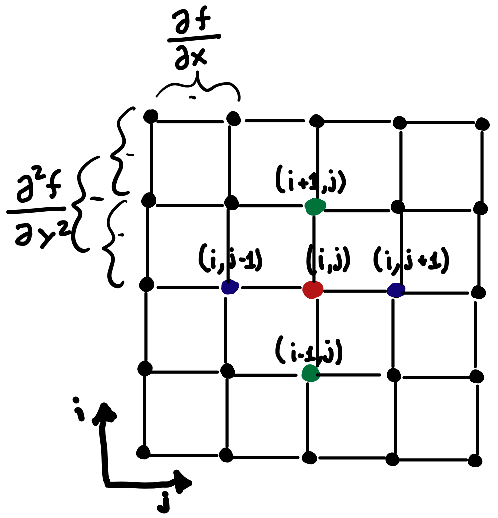
- And for the Laplacian

$$\Delta f(i,j) \sim f(i,j+1) + f(i,j-1) + f(i+1,j) + f(i-1,j) - 4f(i,j)$$

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \iff$$

Example: graph Laplacian in a grid

The graph Laplacian applied to a grid graph can be seen similarly to the Euclidean plane discretized



- In first order we have

$$\frac{\partial f}{\partial x}(i,j) \sim f(i,j+1) - f(i,j)$$

- In second order we have

$$\frac{\partial^2 f}{\partial x^2}(i,j) \sim (f(i,j+1) - f(i,j)) - (f(i,j) - f(i,j-1))$$

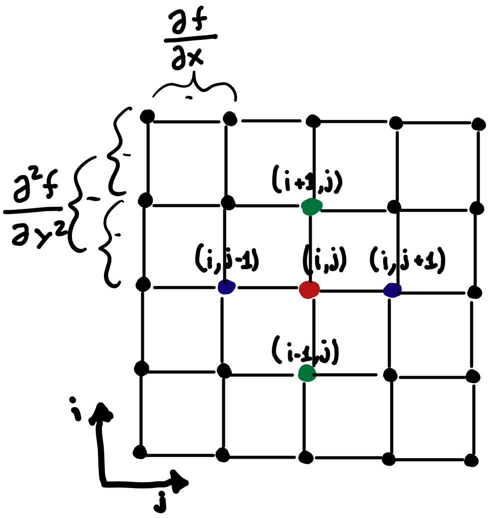
- And for the Laplacian

$$\Delta f(i,j) \sim f(i,j+1) + f(i,j-1) + f(i+1,j) + f(i-1,j) - 4f(i,j)$$

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \Leftrightarrow$$

Example: graph Laplacian in a grid

The graph Laplacian applied to a grid graph can be seen similarly to the Euclidean plane discretized



- In first order we have

$$\frac{\partial f}{\partial x}(i,j) \sim f(i,j+1) - f(i,j)$$

- In second order we have

$$\frac{\partial^2 f}{\partial x^2}(i,j) \sim (f(i,j+1) - f(i,j)) - (f(i,j) - f(i,j-1))$$

- And for the Laplacian

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \iff \Delta f(i,j) \sim f(i,j+1) + f(i,j-1) + f(i+1,j) + f(i-1,j) - 4f(i,j)$$

Dynamic average consensus

From static to dynamic:

(First) Dynamic Laplacian average consensus

- We can improve performance via repeated observations of the reference signal, feeding it back to the control.

$$\dot{x}^i(t) = -\sum_{j=1}^{N} a_{ij}\left(x^i(t) - x^j(t)\right) + u^i(t)$$

which can be rewriten in terms of the Laplacian

$$\dot{x}^i(t) = -\Delta x^i(t) + u^i(t)$$

where $x^i(0) = u^i(0)$.

Dynamic average consensus

Example: pattern formation

- We can describe the pattern formation task from the consensus problem by adding biases

$b^{i,j}$ = desired distance between robots R^i and R^j

where the control becomes

$$\dot{x}^i(t) = -\sum_{j=1}^{N} a_{ij}\left(x^i(t) - x^j(t) - b^{i,j}\right)$$

$$= -\Delta_G\, x^i(t) - \sum_{j=1}^{N} a_{ij}\, b_{ij}$$

Improvements

Multiple aspects of the initial solution can be improved:

- The requirement of the knowledge of the derivative of the signal can be mitigated
- The rate of convergence can be controlled.
- Modifications are possible to render it robust to initial conditions, with later appearing robots.
- More complex signals can be tracked using a priori knowledge.

Outline

- Introduction
  context and motivation

- Dynamic average consensus*
  problem formulation, from static to dynamic, improvements and examples

- Manufacturing consensus
  higher order Laplacians and the Hodge theorem

- Conclusion

*Notes from 2019 - S. Kia et al - Tutorial on dynamic average consensus

Manufacturing consensus

Higher order Laplacians

- Note the usage of the graph Laplacian as an operator for knowledge diffusion.

- We can consider more complex networks to model different levels of synchronicity and fault tolerance in communication.

- Finding consensus in distributed systems can be stated as finding the corresponding harmonic forms, where knowledge becomes stationary.

Manufacturing consensus

The Hodge Theorem

The Hodge theorem connects geometrical properties of those complex networks to the related dynamical system described by its Laplacian.

$$H^k(M) = \mathrm{Ker}(\Delta_k)$$

This point of view allows us to consider new arguments for the solvability of distributed robot tasks and for generating algorithms.

Example: connectivity argument

We can see the asymptotic convergence of a control basen on the Laplacian in terms of connectivity

We desire for the single convergence point x to respect $\Delta^G x = 0$, i.e.

$$x \in \mathrm{Ker}\,\Delta^G \quad \text{and} \quad \dim(\mathrm{Ker}\,\Delta^G) = 1$$

Via Hodge Theory, we can see that G must have a single generator, i.e. be connected

$$H^0(G) = \mathrm{Ker}\,\Delta^G$$

Outline

- Introduction
  context and motivation
- Dynamic average consensus*
  problem formulation, from static to dynamic,
  improvements and examples
- Manufacturing consensus
  higher order Laplacians and the Hodge theorem
- **Conclusion**

*Notes from 2019 - S. Kia et al - Tutorial on dynamic average consensus

Takeaways

- There are significant differences to consider when tracking time-varying signals.
- The Laplacian operator for Knowledge diffusion is an useful concept.
- It is possible to relate geometrical properties to the study of dynamic systems

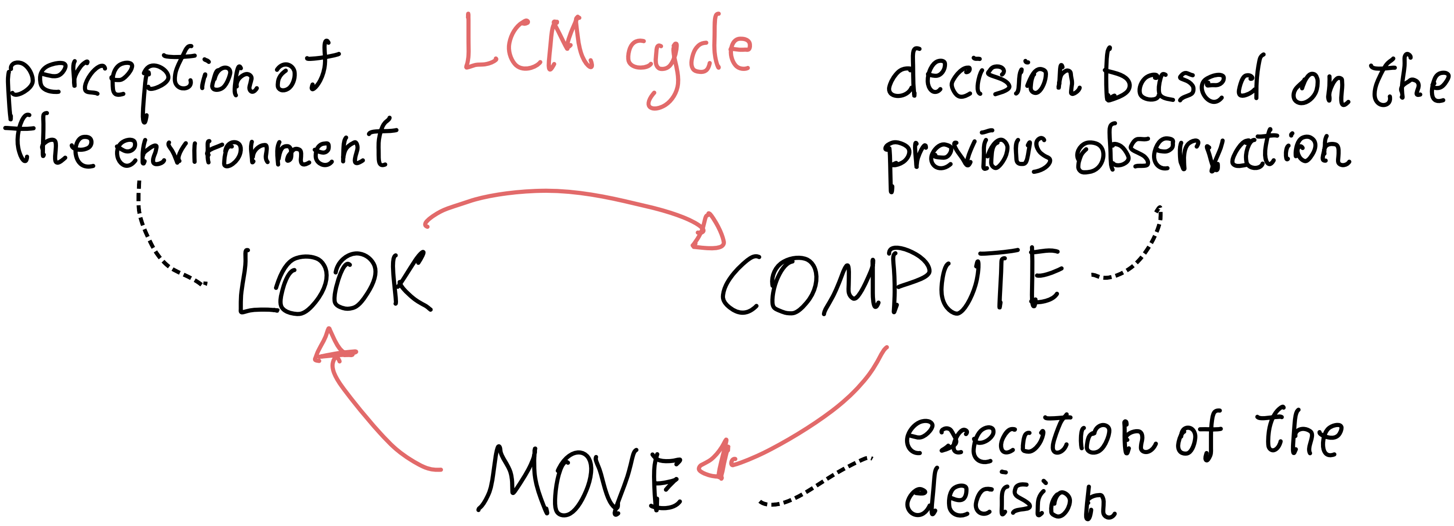
Further questions

- How to relate this general formulation of consensus to the studied discrete tasks?
- How should this work with dynamic graphs to represent the changing visibility during a mission?
- What practical arguments for solvability can we extract from the geometrical relations to the Laplacian?
- What further connections can be drawn to the logic point of view?

Thank you!

LOOK-COMPUTE-MOVE

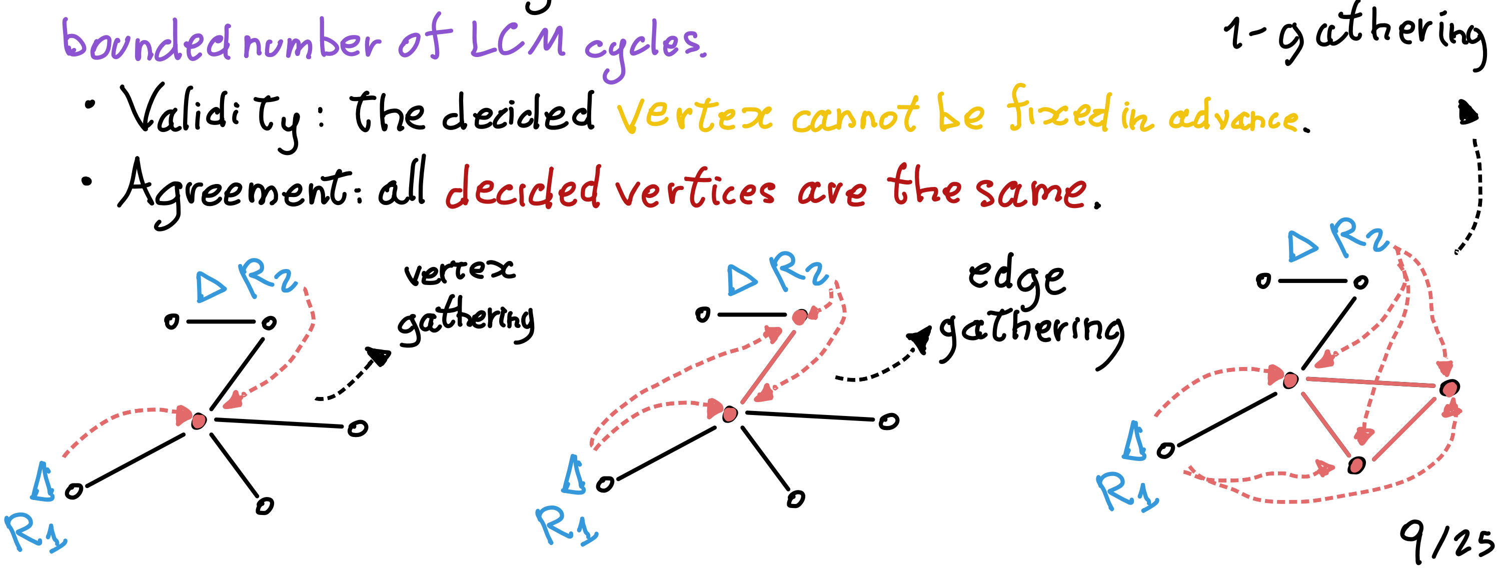
- Theoretical framework for studying different robotic scenarios in an unified view



GATHERING PROBLEM

We wish for $N \geq 2$ undistinguishable robots executing identical algorithms to decide on a vertex to move to, respecting the following properties:

- Termination: every robot decides a vertex in a bounded number of LCM cycles.
- Validity: the decided vertex cannot be fixed in advance.
- Agreement: all decided vertices are the same.



ROBOT TASKS

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "S"] \arrow[urr, "\delta"']
\end{tikzcd}$$

specification

solvability criteria

protocol complex

ROBOT TASKS
INPUT COMPLEX

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
& & \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[uu, "S"] \arrow[uurr, "\delta"'] & &
\end{tikzcd}$$

specification

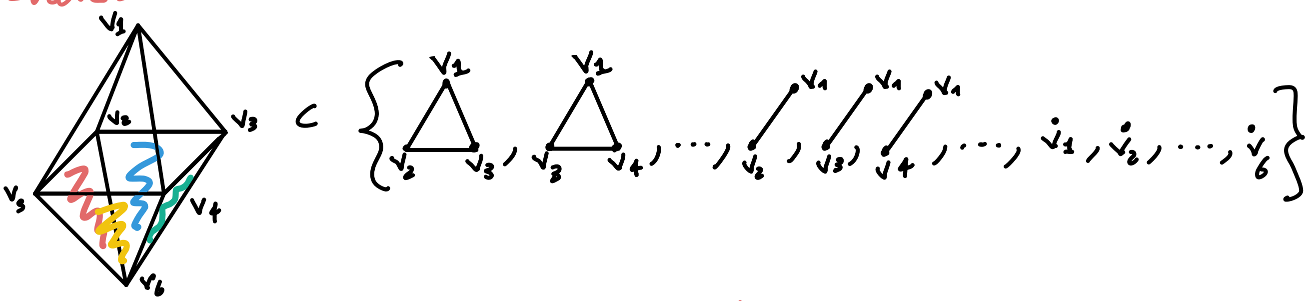
solvability criteria

protocol complex

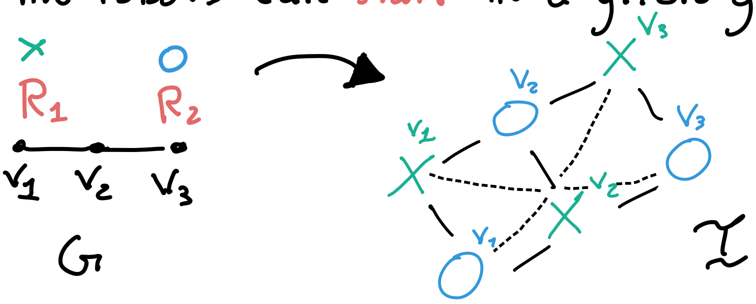
ROBOT TASKS
INPUT COMPLEX

$$
\begin{tikzcd}
\mathcal{I} & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[l, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq P(\mathcal{I}) \arrow[u, "S"] \arrow[ur, "\delta"'] &
\end{tikzcd}
$$

We use simplicial complexes to represent the combination of possible configurations of states



The input complex represents all possible configurations in which the robots can start in a given graph



ROBOT TASKS

TASK DEFINITION

Task definition

input complex

output complex

$$
\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq P(\mathcal{I}) \arrow[u, "S"] \arrow[urr, "\delta"'] & &
\end{tikzcd}
$$

specification

solvability criteria

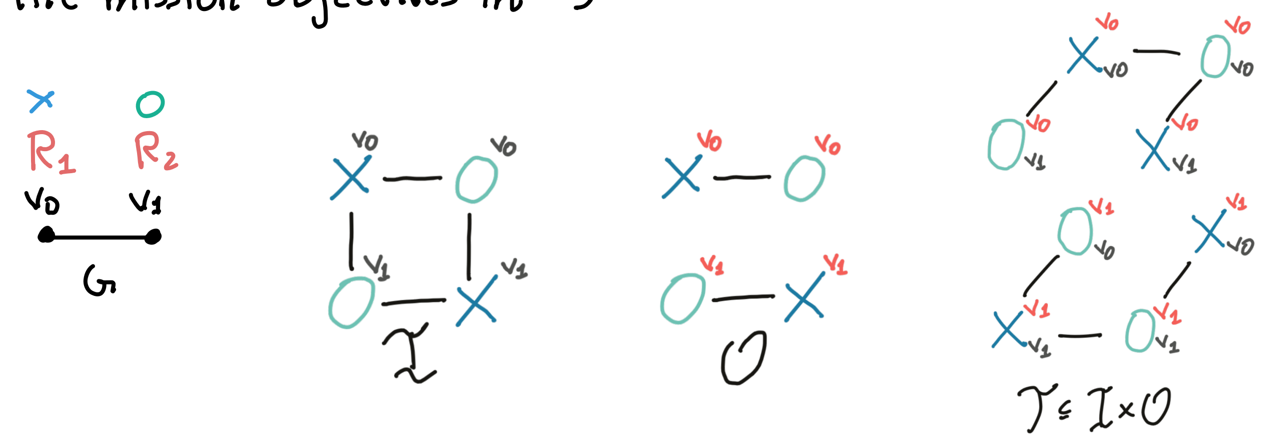
protocol complex

ROBOT TASKS

TASK DEFINITION

$$\begin{tikzcd} \mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\ Q \subseteq P(\mathcal{I}) \arrow[u, "S"] \arrow[urr, "\delta"'] & & \end{tikzcd}$$

A robot task is a triple $(\mathcal{I}, \mathcal{O}, \mathcal{T})$ that defines the pairs of initial and final states respecting the mission objectives in $\mathcal{T}$



Originally defined as the triple $(\mathcal{I}, O, \Delta)$

ROBOT TASKS

ROBOT SPECIFICATION

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
Q \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "S"] \arrow[urr, "\delta"'] & &
\end{tikzcd}$$

specification

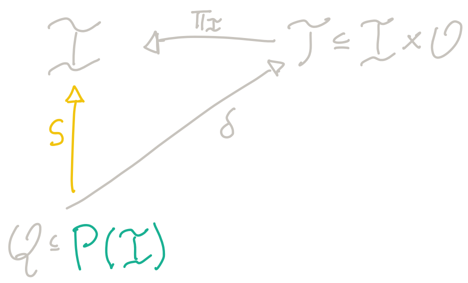
solvability criteria

protocol complex

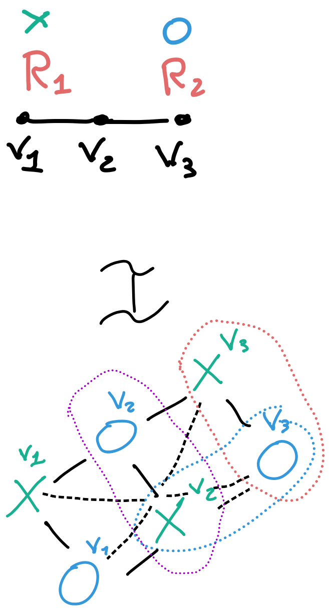
ROBOT TASKS

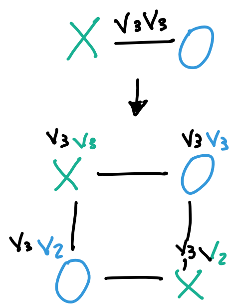
ROBOT SPECIFICATION

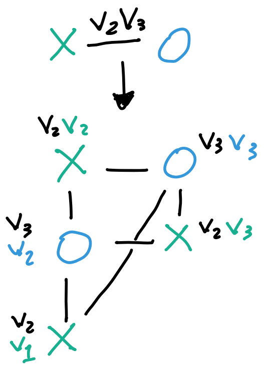
Robot tasks

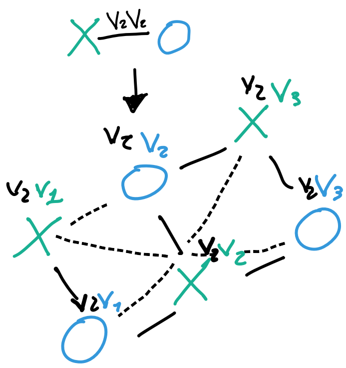


The protocol complex encodes all subdivisions of a state according to its possible variations in an execution









ROBOT TASKS

SOLVABILITY

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "s"] \arrow[urr, "\delta"'] & &
\end{tikzcd}$$

specification

solvability criteria

protocol complex

ROBOT TASKS
SOLVABILITY

Robot tasks

$$\begin{tikzcd} \mathcal{I} & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[l, "\pi_{\mathcal{I}}"'] \\ Q \subseteq P(\mathcal{I}) \arrow[u, "S"] \arrow[ur, "\delta"'] \end{tikzcd}$$

all possible evolutions of all robots

how all robots may start and end their missions

$$P(\mathcal{I}) \xleftarrow{S} Q \xrightarrow{\delta} \mathcal{T} \xrightarrow{\pi} \mathcal{I} \times \mathcal{O}$$

- has matching initial an final states, added to Q
- respects the robot's sensing limitations, filtered by δ