Estimation in Sensor Networks

Sharad C. Shankar

University of California, Santa Barbara

July 24, 2019

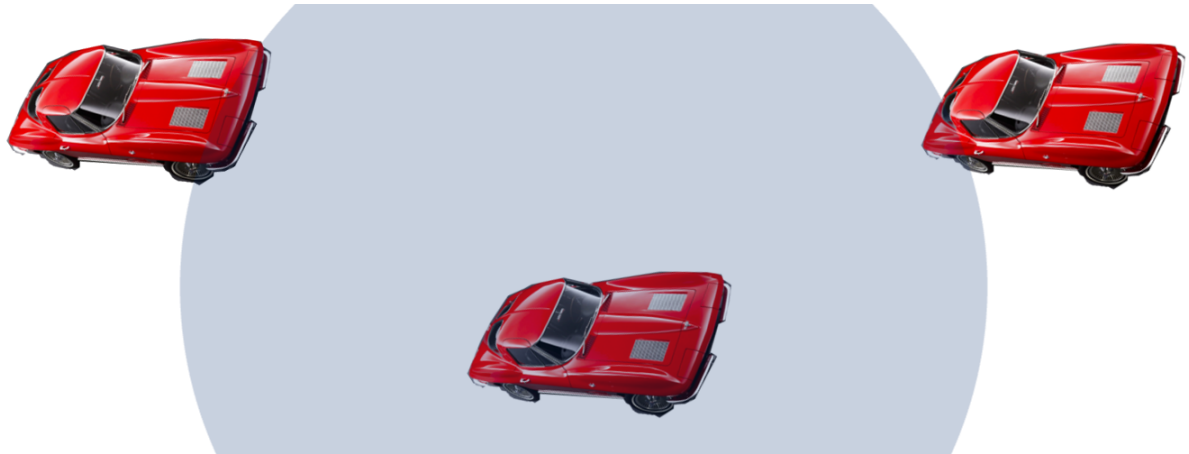
About me

I am a Bay Area native (Homestead High School). I got my Bachelor's in Electrical Engineering with a minor in Mathematics from UCLA and I am currently pursuing a PhD at UCSB focusing on novel ways of estimating the states of dynamical systems (our topic today).

When I was young I was interested in how things work, cars and planes, biology and ecosystems, the weather, cities, the economy, etc. I couldn't decide what to pursue, and so I guess I've ended up studying the common abstractions of these problems, the general rules which govern each of them.

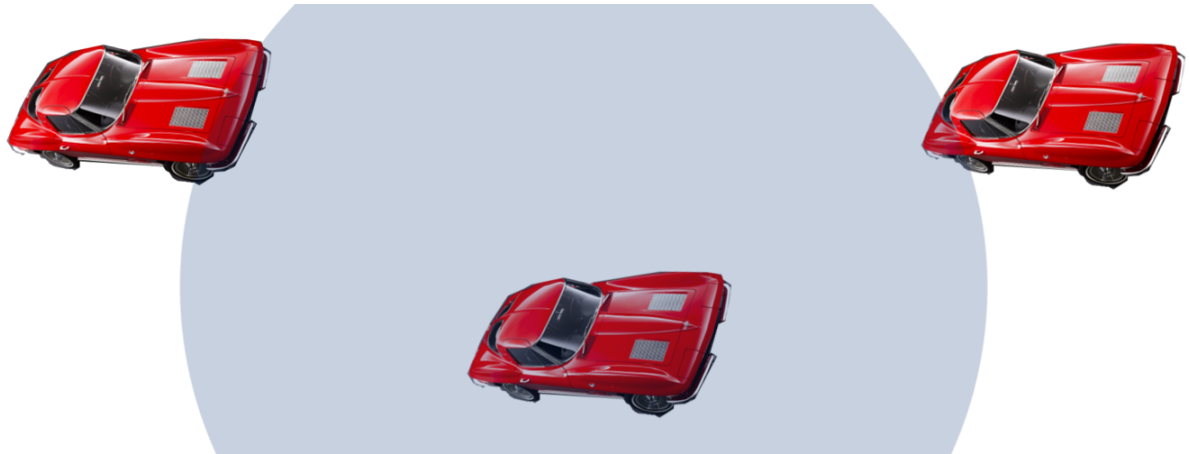
Motivation

Consider a platoon of cars that can communicate with each within a 30 meters and want to maintain a constant velocity and spacing.



Motivation

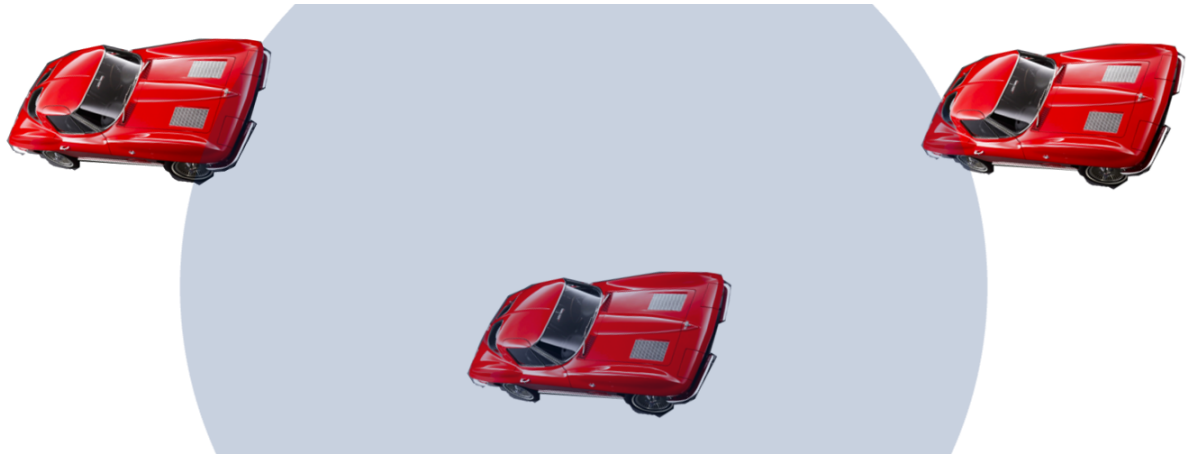
Consider a platoon of cars that can communicate with each within a 30 meters and want to maintain a constant velocity and spacing.



- How do vehicles keep track of their and the other cars locations (need sensors)

Motivation

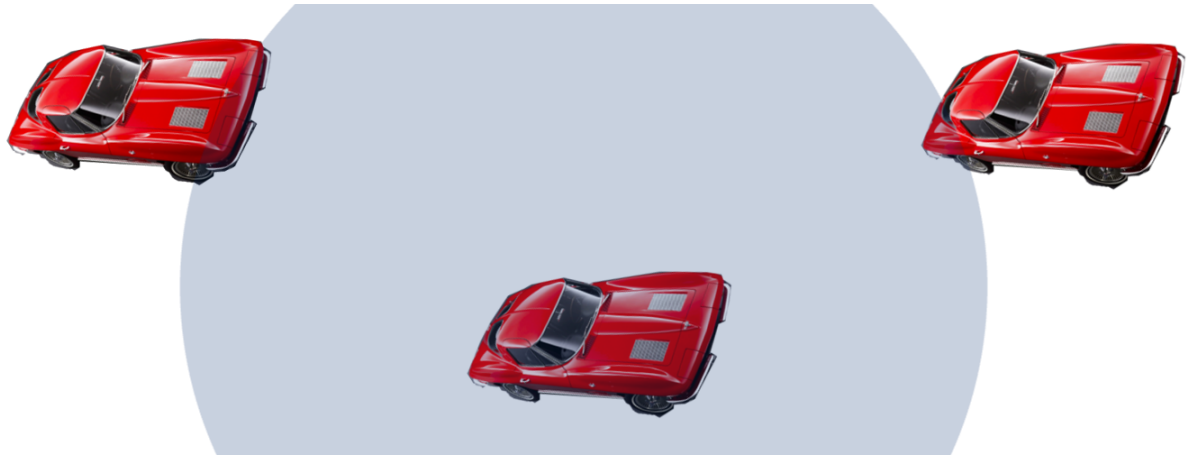
Consider a platoon of cars that can communicate with each within a 30 meters and want to maintain a constant velocity and spacing.



- How do vehicles keep track of their and the other cars locations (need sensors)
- Should the vehicles communicate with each other? How much?

Motivation

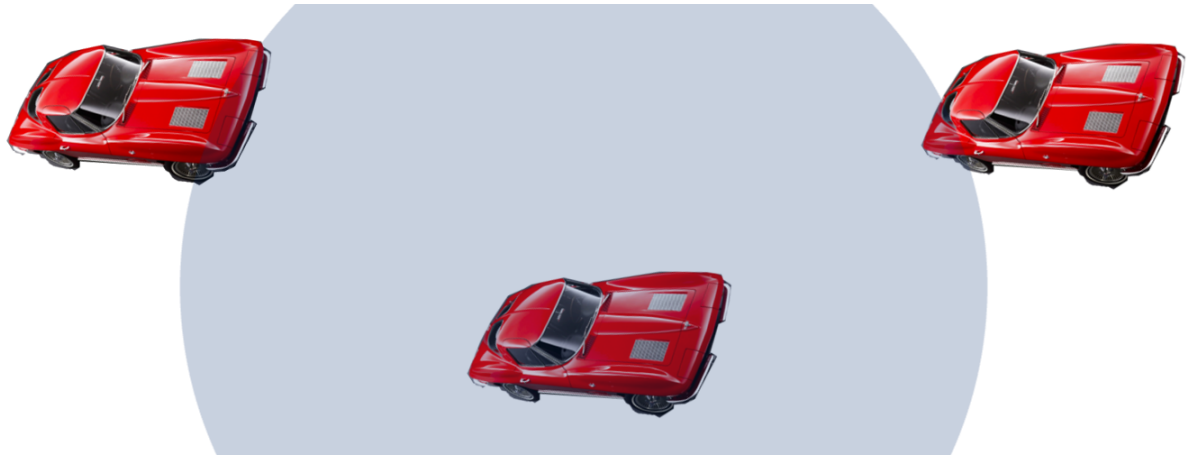
Consider a platoon of cars that can communicate with each within a 30 meters and want to maintain a constant velocity and spacing.



- How do vehicles keep track of their and the other cars locations (need sensors)
- Should the vehicles communicate with each other? How much?
- Is there a trade-off between accuracy and speed due to communication costs?

Motivation

Consider a platoon of cars that can communicate with each within a 30 meters and want to maintain a constant velocity and spacing.



- How do vehicles keep track of their and the other cars locations (need sensors)
- Should the vehicles communicate with each other? How much?
- Is there a trade-off between accuracy and speed due to communication costs?
- Should nodes agree (consensus) on a single estimate? If so, how?

Outline

Dynamical Systems

A dynamical system is any process that evolves according to some rules which determine future states based on the current ones. Part of this evolution can be random (stochastic).

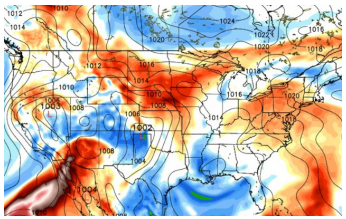
Examples:

Dynamical Systems

A dynamical system is any process that evolves according to some rules which determine future states based on the current ones. Part of this evolution can be random (stochastic).

Examples:

- weather

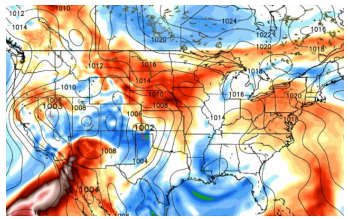


Dynamical Systems

A dynamical system is any process that evolves according to some rules which determine future states based on the current ones. Part of this evolution can be random (stochastic).

Examples:

- weather



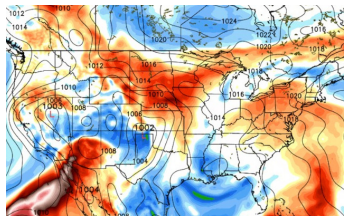
- chemical reactions

Dynamical Systems

A dynamical system is any process that evolves according to some rules which determine future states based on the current ones. Part of this evolution can be random (stochastic).

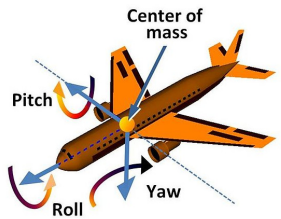
Examples:

- weather



- chemical reactions

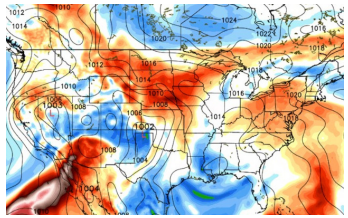
- aircraft position/orientation



Dynamical Systems

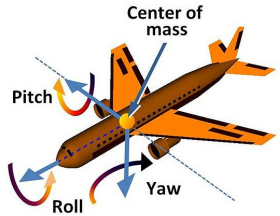
A dynamical system is any process that evolves according to some rules which determine future states based on the current ones. Part of this evolution can be random (stochastic).

Examples:

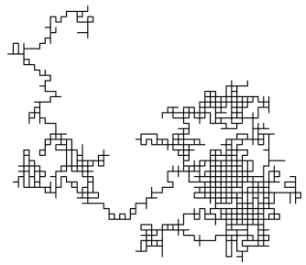


- weather

- chemical reactions



- aircraft position/orientation



- Markov processes (e.g. random walk)

Dynamical Systems

In simple language, a system is dynamical if knowing about the present, tells us something about the future!

If we have a model of our dynamical system, we can:

Dynamical Systems

In simple language, a system is dynamical if knowing about the present, tells us something about the future!

If we have a model of our dynamical system, we can:

- Predict future states based on the current one, e.g. the weather report, catching a ball, or intercepting a missile.

Dynamical Systems

In simple language, a system is dynamical if knowing about the present, tells us something about the future!

If we have a model of our dynamical system, we can:

- Predict future states based on the current one, e.g. the weather report, catching a ball, or intercepting a missile.
- Filter a noisy signal, e.g. de-noising of audio or video signals

Dynamical Systems

In simple language, a system is dynamical if knowing about the present, tells us something about the future!

If we have a model of our dynamical system, we can:

- Predict future states based on the current one, e.g. the weather report, catching a ball, or intercepting a missile.
- Filter a noisy signal, e.g. de-noising of audio or video signals

Both prediction and filtering are examples of Estimation

Estimation

Our goal is to noisy sensor measurements and produce good estimates of the evolving quantity of interest. This has two parts:

- How do we merge data from many sensors (sensor fusion)?
- How do we use knowledge of the past to refine our present estimates?

A simple estimation algorithm

The following is a simple recursive method for estimating the state of a dynamical system:

A simple estimation algorithm

The following is a simple recursive method for estimating the state of a dynamical system:

1. Given your estimate of the previous state, use a model to predict the current state

A simple estimation algorithm

The following is a simple recursive method for estimating the state of a dynamical system:

1. Given your estimate of the previous state, use a model to predict the current state
2. Use your measurements at the current time to refine (update) your prediction

A simple estimation algorithm

The following is a simple recursive method for estimating the state of a dynamical system:

1. Given your estimate of the previous state, use a model to predict the current state
2. Use your measurements at the current time to refine (update) your prediction

The famous Kalman Filter is an example of this method

A simple estimation algorithm

Let's see an example:

A simple estimation algorithm

Let's see an example:

Suppose our process is a 1-dimensional Gaussian random walk:

$$x(t+1) = x(t) + d(t)$$

where $d(t)$ is drawn from a Gaussian (bell curve) distribution with mean 0 and variance 1.

A simple estimation algorithm

Let's see an example:

Suppose our process is a 1-dimensional Gaussian random walk:

$$x(t+1) = x(t) + d(t)$$

where $d(t)$ is drawn from a Gaussian (bell curve) distribution with mean 0 and variance 1.

We also have a measurement of the state

$$y(t) = x(t) + n(t)$$

where $n(t)$ is drawn from a Gaussian distribution with mean 0 and variance 2.

A simple estimation algorithm

Algorithm:

A simple estimation algorithm

Algorithm:

1. (Prediction) Suppose we have an initial estimate $\hat{x}(0) = y(0) = 0$ with uncertainty of variance 2. Then we predict that $\bar{x}(1) = \hat{x}(0)$. Our uncertainty in this prediction will be with variance 2+1=3.

A simple estimation algorithm

Algorithm:

1. (Prediction) Suppose we have an initial estimate $\hat{x}(0) = y(0) = 0$ with uncertainty of variance 2. Then we predict that $\bar{x}(1) = \hat{x}(0)$. Our uncertainty in this prediction will be with variance 2+1=3.

2. (Update) We receive a measurement $y(1) = 0.5$. Our uncertainty in this measurement has variance 2.

A simple estimation algorithm

Algorithm:

1. (Prediction) Suppose we have an initial estimate $\hat{x}(0) = y(0) = 0$ with uncertainty of variance 2. Then we predict that $\bar{x}(1) = \hat{x}(0)$. Our uncertainty in this prediction will be with variance 2+1=3.

2. (Update) We receive a measurement $y(1) = 0.5$. Our uncertainty in this measurement has variance 2.

   To obtain our estimate at time 1, we merge the prediction and measurement through a weighted average:

A simple estimation algorithm

Algorithm:

1. (Prediction) Suppose we have an initial estimate $\hat{x}(0) = y(0) = 0$ with uncertainty of variance 2. Then we predict that $\bar{x}(1) = \hat{x}(0)$. Our uncertainty in this prediction will be with variance 2+1=3.

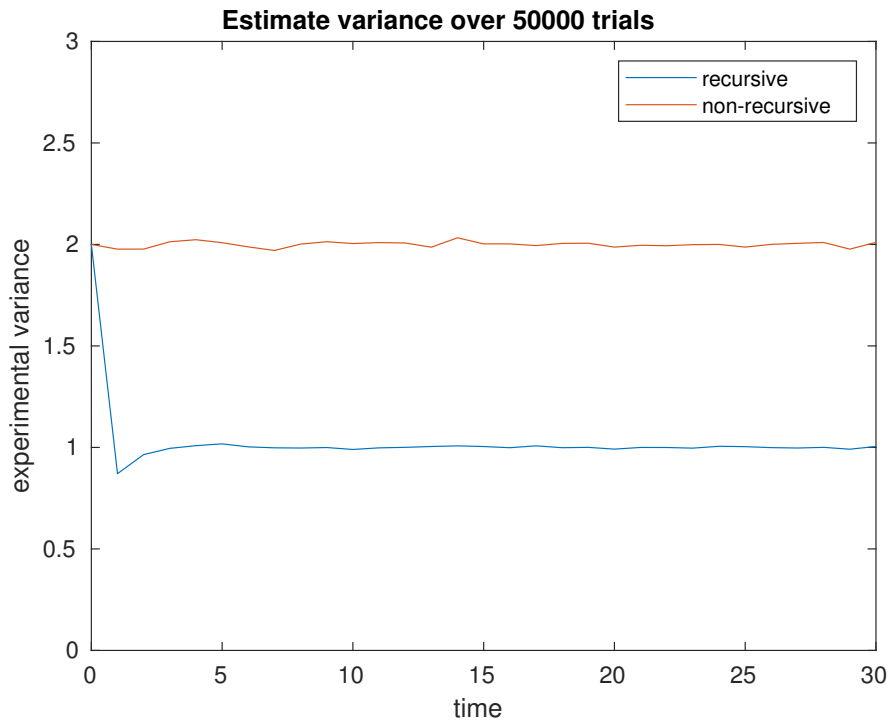
2. (Update) We receive a measurement $y(1) = 0.5$. Our uncertainty in this measurement has variance 2.

   To obtain our estimate at time 1, we merge the prediction and measurement through a weighted average:

$$\hat{x}(1) = \frac{2}{2+3}\bar{x}(1) + \frac{3}{2+3}y = \frac{2}{5} \cdot 0 + \frac{3}{5} \cdot 0.5 = 0.3$$

Results

Without the recursive algorithm, just using $\hat{x} = y$, each estimate would have a variance of 2. The following compares the experimental variance of the recursive algorithm over many trials.



More measurements = better

One important fact to keep in mind:

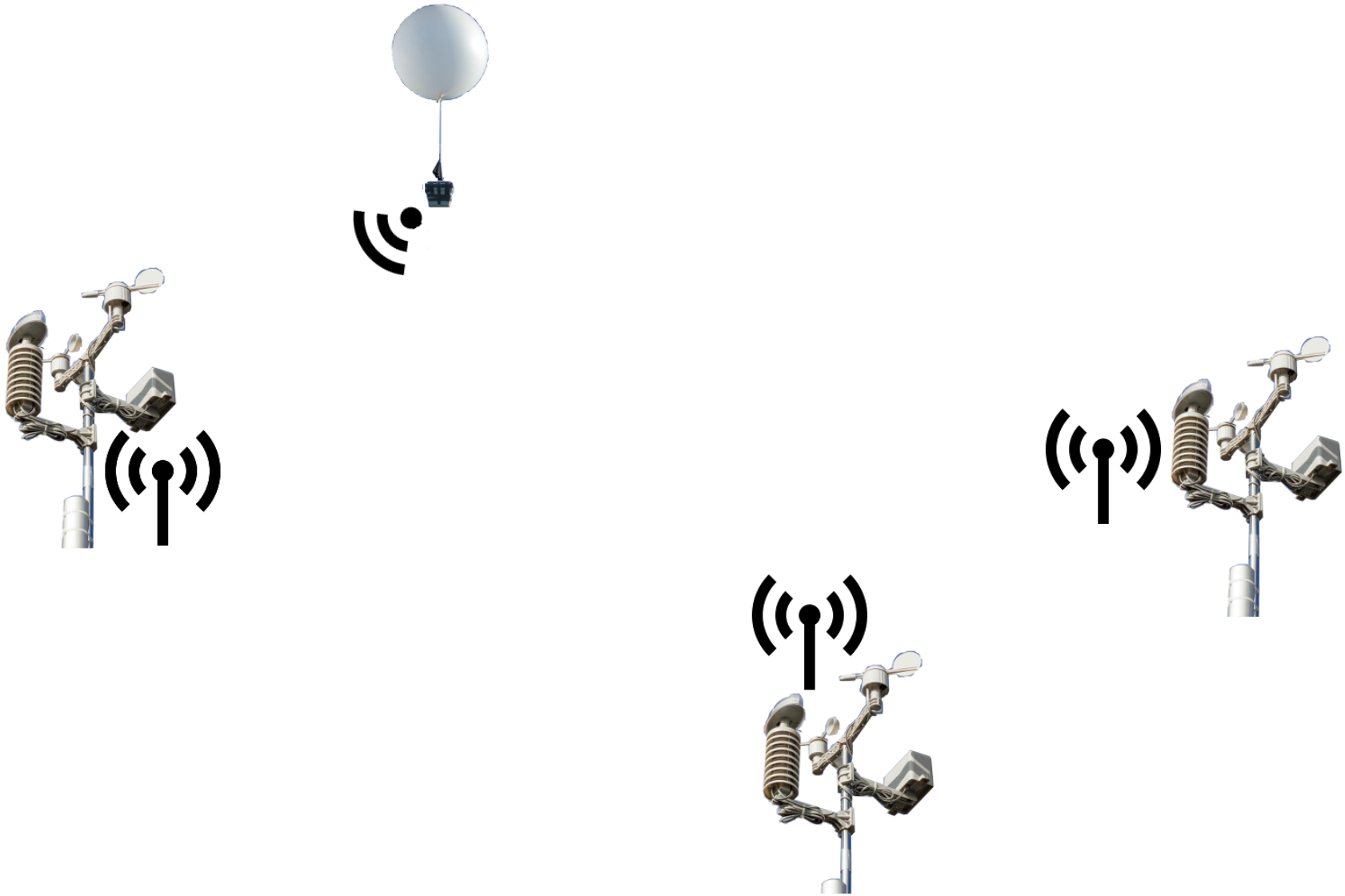
It is almost always better to have more measurements, this results in more accuracy in your estimates.

Sensor Networks

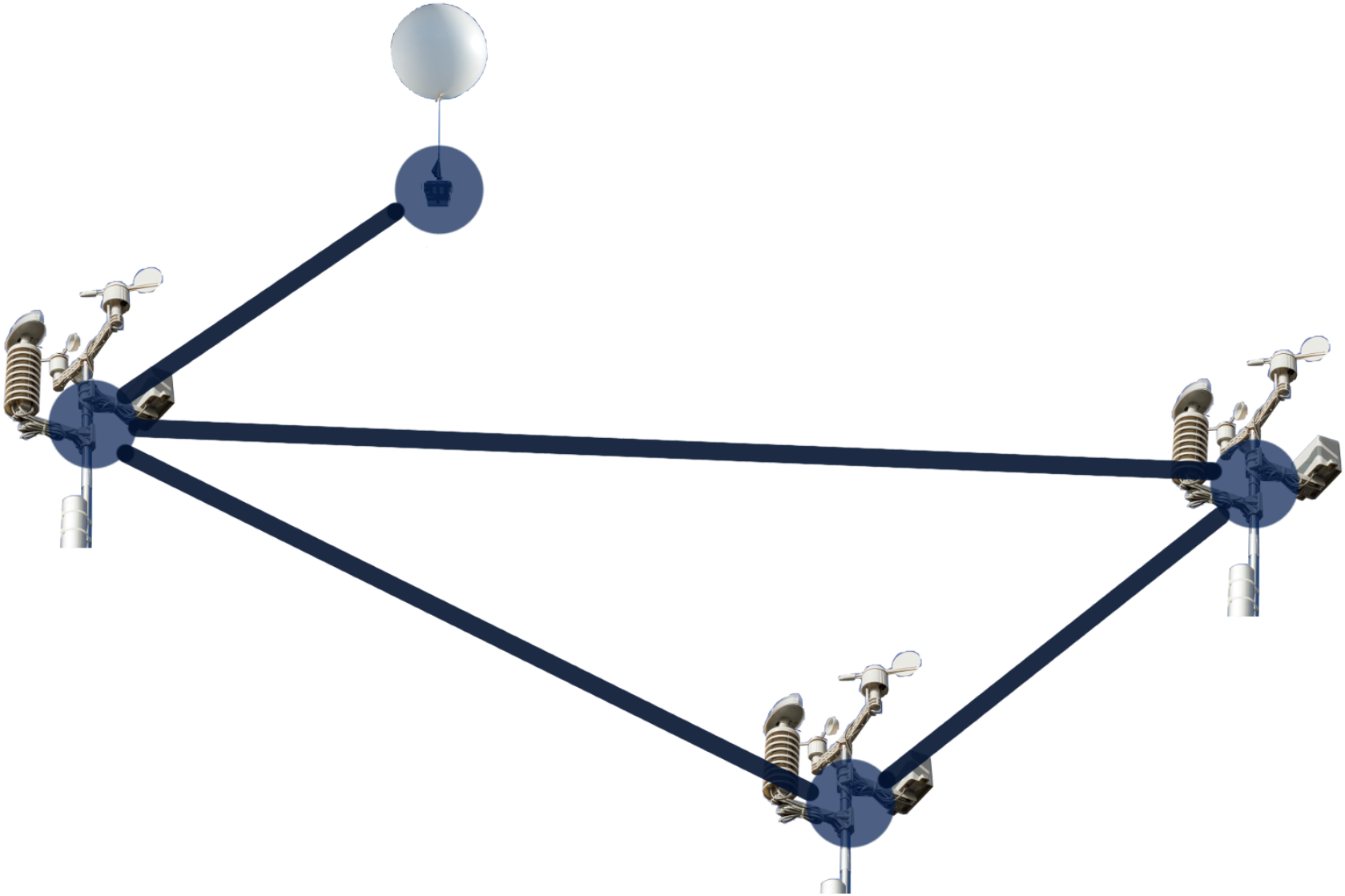
Let's consider cases where a dynamical process is monitored by disparate yet communicating sensors. Examples include monitoring of:

- weather (through balloons, stations, etc.)
- nuclear reactor cores
- multi-robot tasks
- large power grids
- inter-vehicle dynamics (platoons, collision avoidance)

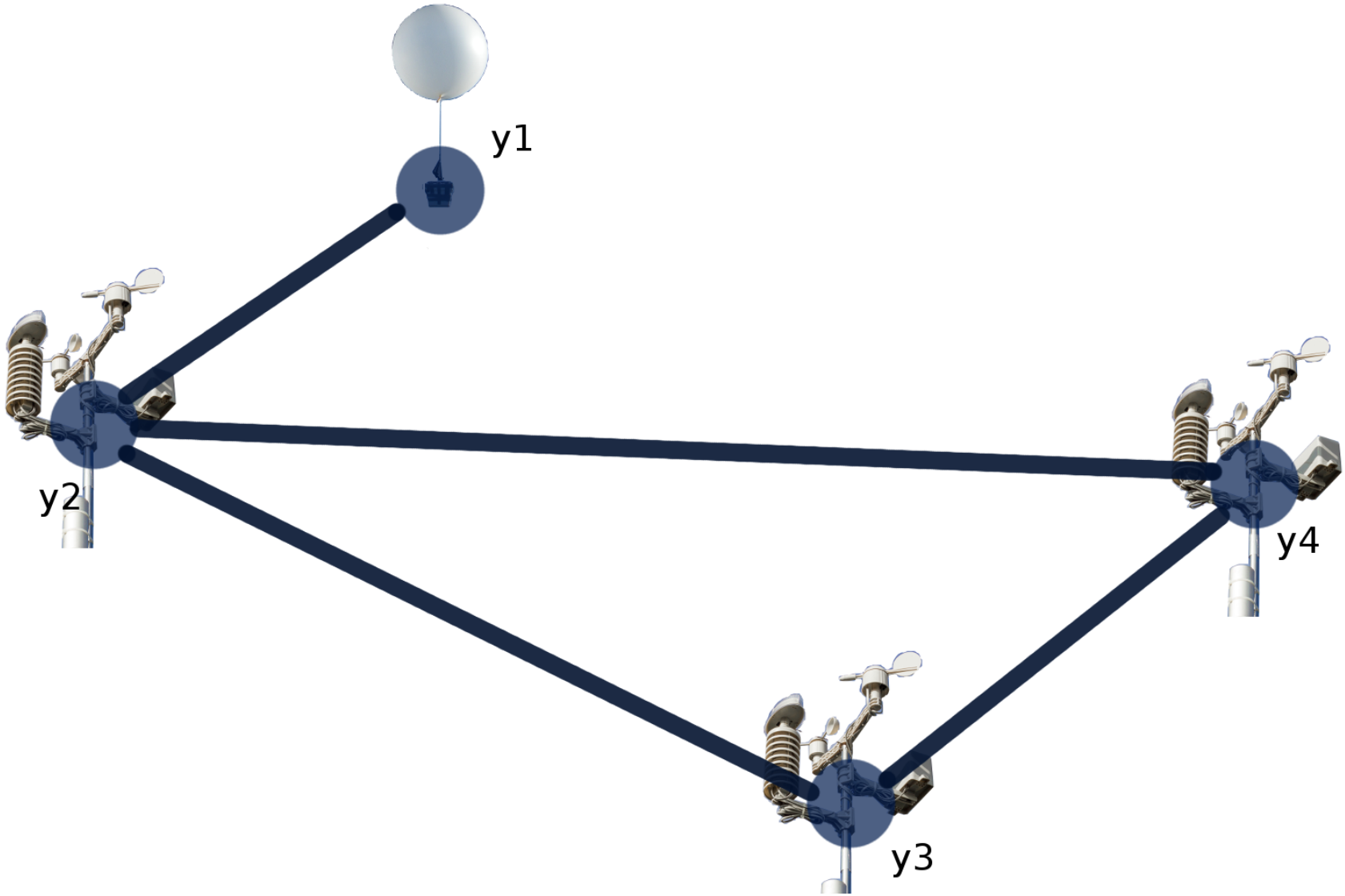
Sensor Networks: Central Estimation



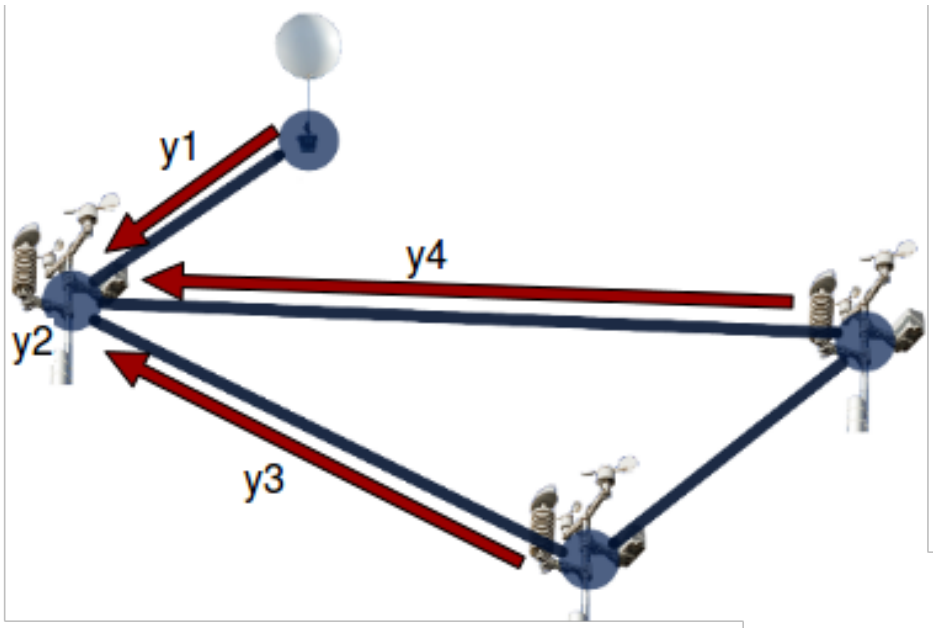
Sensor Networks: Central Estimation

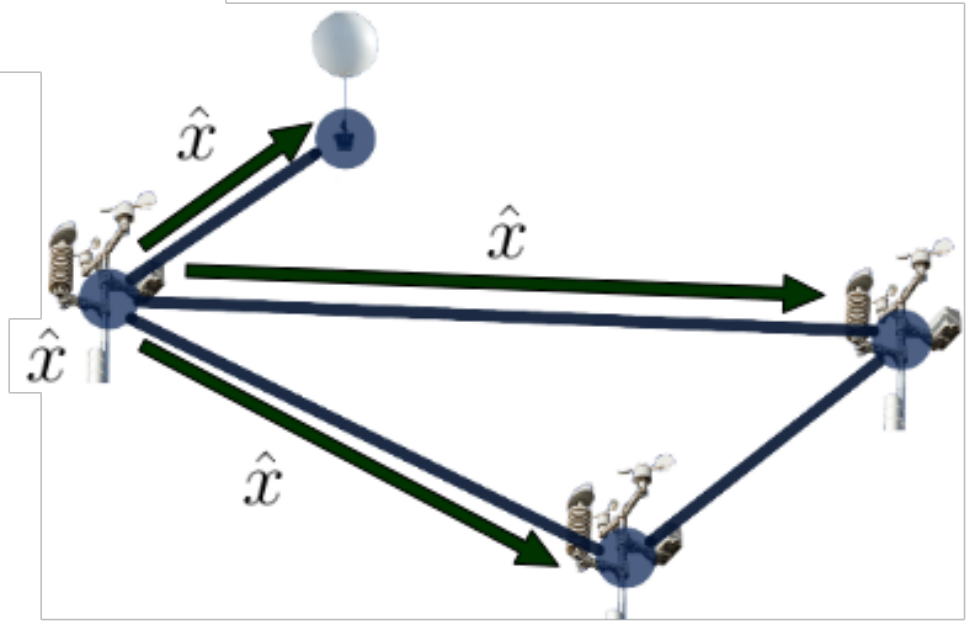


Sensor Networks: Central Estimation



Sensor Networks: Central Estimation





Sensor Networks: Central Estimation

Computing estimates centrally guarantees us

Sensor Networks: Central Estimation

Computing estimates centrally guarantees us

- accuracy ✓

Sensor Networks: Central Estimation

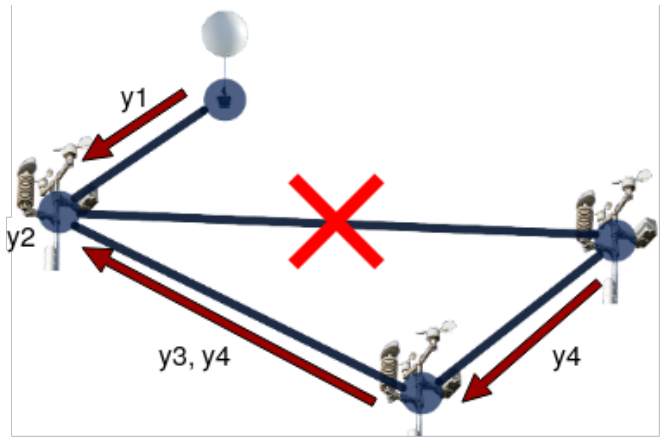
Computing estimates centrally guarantees us

- accuracy ✓
- consensus ✓

Sensor Networks, cont.

Computing estimates based on *all* measurements in the network may be inconvenient for various reasons:

- a central node is infeasible due to distances
- communication cost/delay for computation is unacceptable



So a decentralized scheme may sacrifice performance for speed

Decentralized Scheme - No Communication

If nodes keep track of their own estimates, rather than a single global estimate, we call this a decentralized scheme.

Decentralized Scheme - No Communication

If nodes keep track of their own estimates, rather than a single global estimate, we call this a decentralized scheme.

One approach could be to use no communication, Nodes simply compute their estimates with their own measurements, independent of each other.

Decentralized Scheme - No Communication

If nodes keep track of their own estimates, rather than a single global estimate, we call this a decentralized scheme.

One approach could be to use no communication, Nodes simply compute their estimates with their own measurements, independent of each other.

With this approach do we have

Decentralized Scheme - No Communication

If nodes keep track of their own estimates, rather than a single global estimate, we call this a decentralized scheme.

One approach could be to use no communication, Nodes simply compute their estimates with their own measurements, independent of each other.

With this approach do we have

- accuracy?

Decentralized Scheme - No Communication

If nodes keep track of their own estimates, rather than a single global estimate, we call this a decentralized scheme.

One approach could be to use no communication, Nodes simply compute their estimates with their own measurements, independent of each other.

With this approach do we have

- accuracy? ✗

Decentralized Scheme - No Communication

If nodes keep track of their own estimates, rather than a single global estimate, we call this a decentralized scheme.

One approach could be to use no communication, Nodes simply compute their estimates with their own measurements, independent of each other.

With this approach do we have

- accuracy? ✗
- consensus?

Decentralized Scheme - No Communication

If nodes keep track of their own estimates, rather than a single global estimate, we call this a decentralized scheme.

One approach could be to use no communication, Nodes simply compute their estimates with their own measurements, independent of each other.

With this approach do we have

- accuracy? ✗
- consensus? ✗

Decentralized Schemes - Limited Communication

The best approach then, may be to use limited communication within a decentralized scheme. Each node will have their own estimate, but using some information from neighbors.

Decentralized Schemes - Limited Communication

The best approach then, may be to use limited communication within a decentralized scheme. Each node will have their own estimate, but using some information from neighbors.

There are two possible pieces of information the nodes could share

Decentralized Schemes - Limited Communication

The best approach then, may be to use limited communication within a decentralized scheme. Each node will have their own estimate, but using some information from neighbors.

There are two possible pieces of information the nodes could share

- their raw measurements, or

Decentralized Schemes - Limited Communication

The best approach then, may be to use limited communication within a decentralized scheme. Each node will have their own estimate, but using some information from neighbors.

There are two possible pieces of information the nodes could share

- their raw measurements, or
- their estimates

Decentralized Schemes - Limited Communication

The best approach then, may be to use limited communication within a decentralized scheme. Each node will have their own estimate, but using some information from neighbors.

There are two possible pieces of information the nodes could share

- their raw measurements, or
- their estimates

With either approach we hope to approximate some properties of the centralized scheme, especially accuracy and consensus.

Measurement Sharing

Nodes share their raw measurements with neighbors. So each node gets additional measurements, but importantly many of those are shared and so we expect some local consistency between their estimates.

Measurement Sharing

Nodes share their raw measurements with neighbors. So each node gets additional measurements, but importantly many of those are shared and so we expect some local consistency between their estimates.

Does this help us achieve:

Measurement Sharing

Nodes share their raw measurements with neighbors. So each node gets additional measurements, but importantly many of those are shared and so we expect some local consistency between their estimates.

Does this help us achieve:

- accuracy?

Measurement Sharing

Nodes share their raw measurements with neighbors. So each node gets additional measurements, but importantly many of those are shared and so we expect some local consistency between their estimates.

Does this help us achieve:

- accuracy? $\approx$ more than no communication case
- consensus?

Measurement Sharing

Nodes share their raw measurements with neighbors. So each node gets additional measurements, but importantly many of those are shared and so we expect some local consistency between their estimates.

Does this help us achieve:

- accuracy? $\approx$ more than no communication case
- consensus? ✗ not really

Estimate Sharing

If nodes share estimates, we could consider a couple of possibilities, either we run a few steps of consensus, or we treat their estimates a additional measurements.

Estimate Sharing

If nodes share estimates, we could consider a couple of possibilities, either we run a few steps of consensus, or we treat their estimates a additional measurements.

Does this help us achieve:

Estimate Sharing

If nodes share estimates, we could consider a couple of possibilities, either we run a few steps of consensus, or we treat their estimates a additional measurements.

Does this help us achieve:

- accuracy?

Estimate Sharing

If nodes share estimates, we could consider a couple of possibilities, either we run a few steps of consensus, or we treat their estimates a additional measurements.

Does this help us achieve:

- accuracy? $\approx$
- consensus?

Estimate Sharing

If nodes share estimates, we could consider a couple of possibilities, either we run a few steps of consensus, or we treat their estimates a additional measurements.

Does this help us achieve:

- accuracy? $\approx$
- consensus? $\approx$ estimates propagate through the network

Hmmm, is that all...

So what are we missing?

Hmmm, is that all...

So what are we missing?

The answer is that we considered only simple process, measurement, and noise models. Many of the applications we talked about have much more complicated behavior not captured by these models.

Furthermore, the communication graph structure may change in time! How do we adjust our algorithm?

Where do we go from here?

Once we have achieved estimation in sensor networks, we can then accomplish various distributed control tasks. For example:

Where do we go from here?

Once we have achieved estimation in sensor networks, we can then accomplish various distributed control tasks. For example:

- robot swarms

Where do we go from here?

Once we have achieved estimation in sensor networks, we can then accomplish various distributed control tasks. For example:

- robot swarms

- vehicle formation control

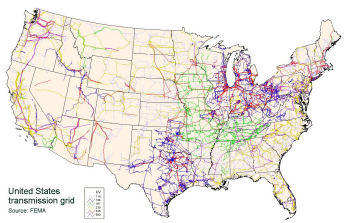
Where do we go from here?

Once we have achieved estimation in sensor networks, we can then accomplish various distributed control tasks. For example:

- robot swarms

- vehicle formation control

- power grid synchronization



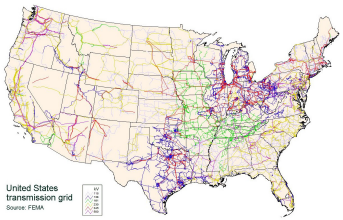
Where do we go from here?

Once we have achieved estimation in sensor networks, we can then accomplish various distributed control tasks. For example:

- robot swarms

- vehicle formation control

- power grid synchronization



- traffic control