Distributed Systems & Control

Advanced Topics in Control 2022

Florian Dörfler

Vahid Mamduhi

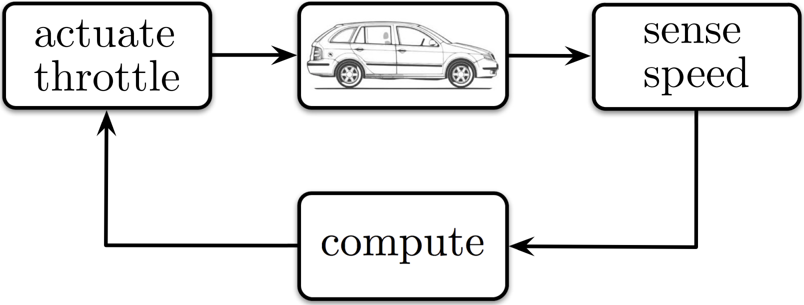
Mathias Hudoba de Badyn

ETH zürich

AUTOMATIC CONTROL LABORATORY

welcome!

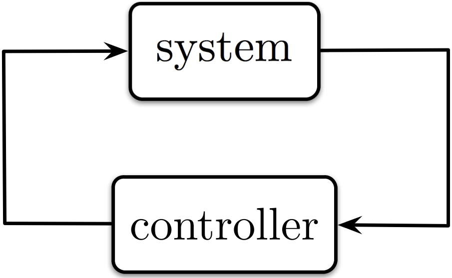
The baseline in this course



"simple" control systems are well understood

"complexity" can enter in many ways

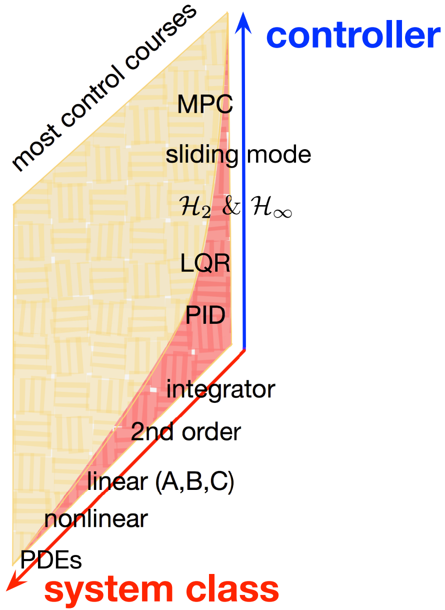
The centralized control loop



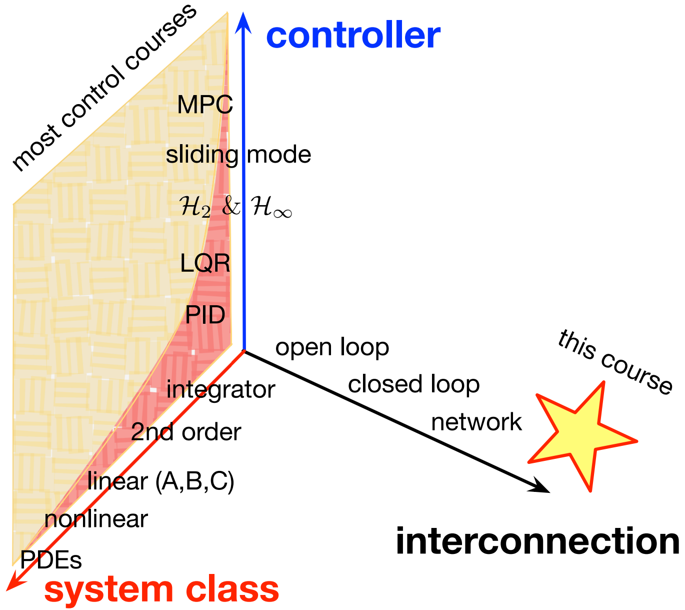
When deviating from a linear SISO setup the challenges can arise from

1. more complicated **system classes** (nonlinear, ∞-dimensional, etc.) &
2. most centralized approaches **do not scale** to large MIMO systems.

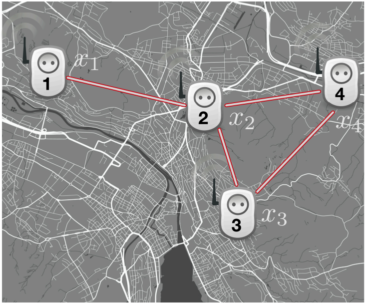
Complexities in control systems



Complexities in control systems — cont'd



A preliminary description of averaging algorithms



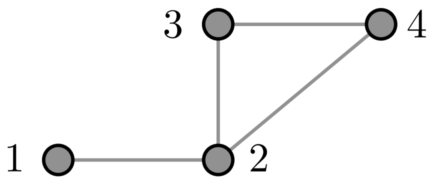
Setup: suppose that each node i in a wireless sensor network has a noisy measurement of a scalar environmental quantity x_i, e.g., aerosol concentration.

Task: construct the "best" aggregate measurement from the sensor data

Consider the following simple **distributed algorithm** based on the concepts of linear averaging: each node repeatedly executes

$$x_i^+ = \text{average}\big(x_i, \{x_j, \text{ for all neighbor nodes } j\}\big),$$

where x_i^+ denotes the new value of x_i.



For this example, we get

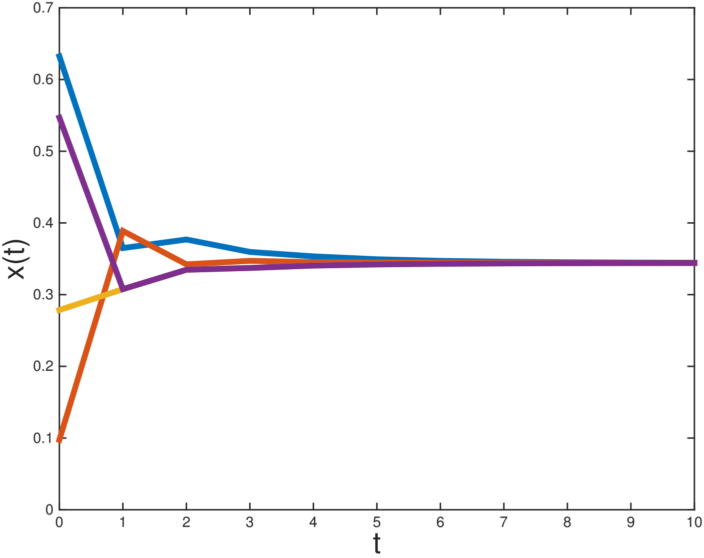
$$x_1^+ := (x_1 + x_2)/2,$$
$$x_2^+ := (x_1 + x_2 + x_3 + x_4)/4,$$

and so forth.

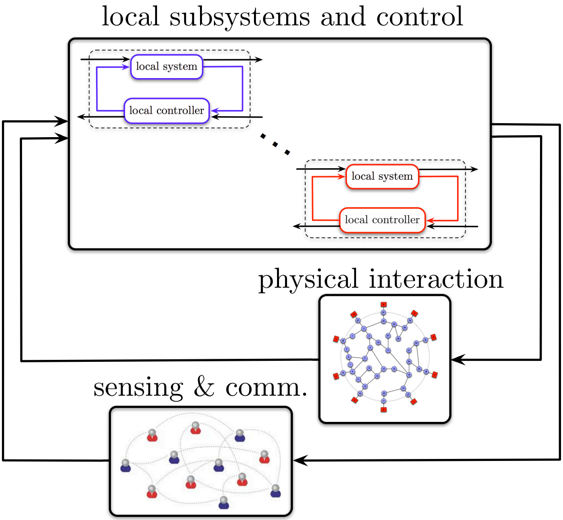
In summary, the algorithm's behavior is
$$x^+ = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 1/3 \end{bmatrix} x$$

Questions of interest are:

1. Does each node converge to a value?
2. Is this value equal to the average of the initial conditions?
3. What properties does the graph and the corresponding matrix need to have for the algorithm to converge? How quick is the convergence?


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
x(t)
0
1
2
3
4
5
6
7
8
9
10
t

A distributed control system



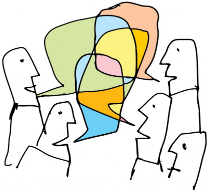
Distributed control systems include large-scale physical systems, engineered & social multi-agent systems, & interconnection in cyber-physical systems.

A few examples of distributed systems & control

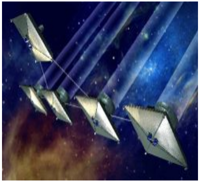
often the centralized perspective is simply not appropriate

robotic networks

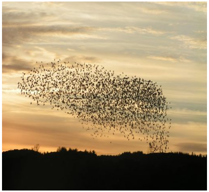
decision making



social networks



sensor networks



self-organization

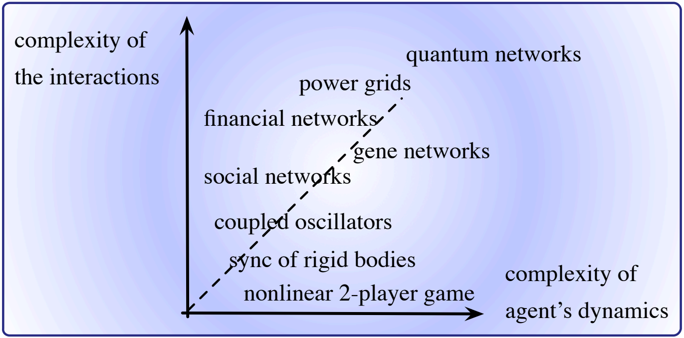
pervasive computing

traffic networks

smart power grids

Challenges in distributed systems & control

1. more complicated **system & controller classes** (as usual)
2. **interaction** through a physical/sensing/communication network
   - cyber-physical issues (sampled-data, channels, computation, etc.)
   - interaction through complex network (large, ad hoc, time-varying, etc.)
   - limited information, sensing, communication, & computation capabilities



Topics covered in this course

Within this universe of problems, we focus mostly on

- **simple agent dynamics** and **complex interactions**
- **matrix theory** as basic analysis tool (as in linear systems)

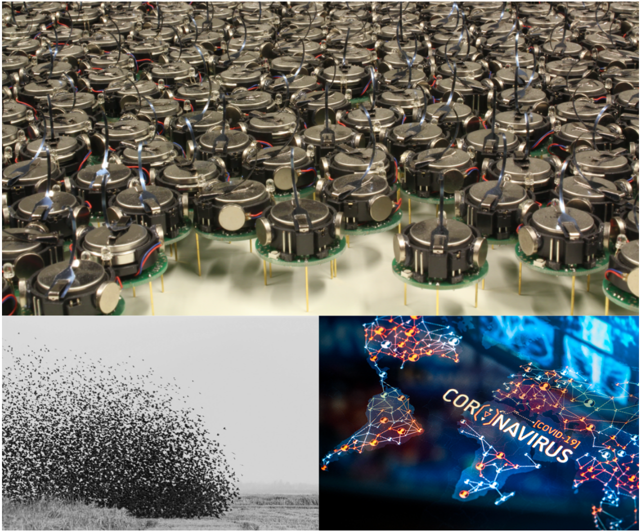
Once we get the gist, everything can be generalized to the nonlinear world.

We will cover the following topics:

1. **theory of graphs**: nonnegative matrices (Perron-Frobenius), algebraic & spectral graph theory, adjacency/Laplacian/incidence matrices
2. **basic network models** such as averaging dynamics, gradient/saddle flows, polynomial (SIR) dynamics, & Hamiltonian systems
3. **networked & triggered control** subject to resource constraints
4. **various application areas** including sensor/social/traffic networks, circuits, epidemics, robotic coordination, & distributed optimization

Time permitting, we can also venture into specialized topics of interests.

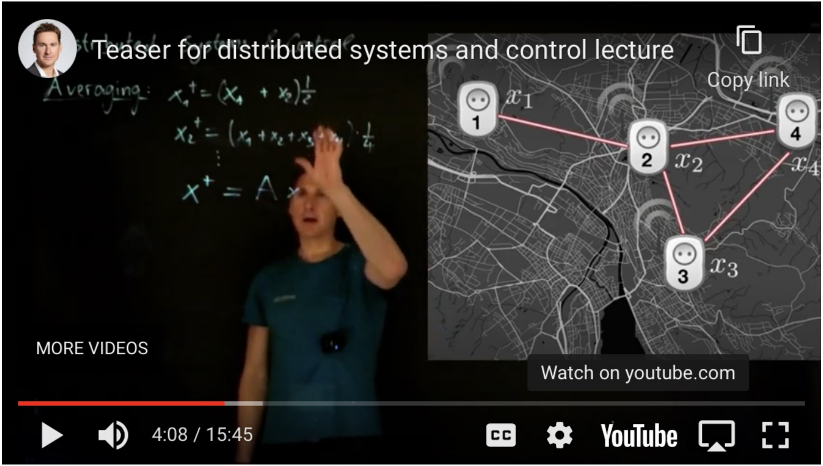
We made sure to include topics of contemporary interests



All theory will be illustrated with physical & socio-technical **applications**!

Did you watch the teaser ?

(link)



course organization

Lecturing team

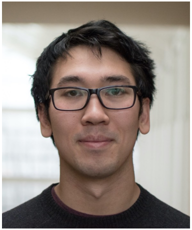
Florian Dörfler (www , email)

Mathias Hudoba de Badyn (www , email)

Vahid Mamduhi (www , email)

Carlo Cenedese (www , email)

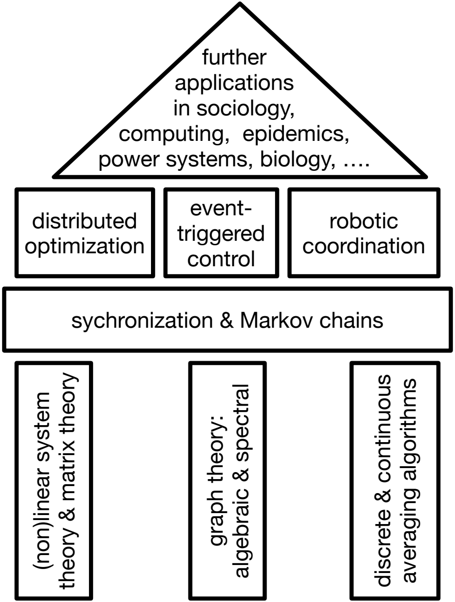
Alberto Padoan (www , email)



Dom Liao-McPherson (www , email)

Andrea Martinelli (www , email)

Building blocks of this course



Tentative course schedule

Part I: Linear Network Systems

Week	Tentative Contents	Lecturer
1	Motivating Examples	Florian Dörfler
2	Elements of Matrix Theory	Florian Dörfler
3	Elements of Graph Theory	Vahid Mamduhi
4	Discrete-Time Averaging	Vahid Mamduhi
5	Laplacian Matrix	Florian Dörfler
6	Continuous-Time Averaging	Florian Dörfler

Part II: Networked & Event-Triggered Control

Week	Tentative Contents	Lecturer
7	Stochastic Stability & Markov Chains	Vahid Mamduhi
8	Event-Triggered Control I	Vahid Mamduhi
9	Event-Triggered Control II	Vahid Mamduhi

Part III: Nonlinear Network Systems & Applications

Week	Tentative Contents	Lecturer
10	Nonlinear Network Systems	Mathias Hudoba de Badyn
11	Robotic Coordination I	Mathias Hudoba de Badyn
12	Robotic Coordination II	Mathias Hudoba de Badyn
13	Distributed Optimization I	Mathias Hudoba de Badyn
14	Distributed Optimization II	Mathias Hudoba de Badyn

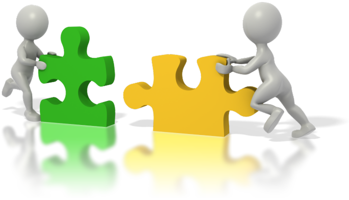
Learning objectives

This is the **advanced topics course**. So we will not venture into tedious exercises, lengthy calculations, coding, . . . , at least not during class.

Overall goal: by the end of this course you will have developed a sound and versatile toolkit to tackle any problem in distributed systems control.

This course won't be *monkey work* but rather all about problem solving **from the basics to the scientific frontier** with multiple applications

So we expect you to be motivated and enthusiastic about **puzzle solving**.



Course organization

227-0690-12L Advanced Topics in Control (Spring 2022) FS2022

Dashboard / Meine Kurse / 227-0690-12L Advanced Topics in Control (Spring 2022) FS2022

General Information

Lecturers: Prof. Florian Dörfler, Dr. Mathias Hudoba de Badyn, Dr. Vahid Mamduhi

Assistants: Andrea Martinelli, Dr. Dominic Liao-McPherson, Alberto Padoan, Carlo Cenedese

Student assistants: Joudi Hajar, Aristomenis Sfetsos

	When	**Where**	**Video link**
Lectures:	Mondays, 16:00-18:00	HG D1.1 (also streamed online)	link
Tutorials:	Fridays, 10:00-12:00	HG D1.1 (also streamed online)	link

Grading: based on 3 homework assignments (50%) and a final project (50%). Here is the information on grading, homework, and the final project.

Everything will be handled via **moodle**: up/downloads, forum, polls, etc.

Features of the moodle

227-0690-12L Advanced Topics in Control (Spring 2022) FS2022

Dashboard / Meine Kurse / 227-0690-12L Advanced Topics in Control (Spring 2022) FS2022

General Information

Lecturers: Prof. Florian Dörfler, Dr. Mathias Hudoba de Badyn, Dr. Vahid Mamduhi

Assistants: Andrea Martinelli, Dr. Dominic Liao-McPherson, Alberto Padoan, Carlo Cenedese

Student assistants: Joudi Hajar, Aristomenis Sfetsos

- **learning resources**: slides, videos, lecture notes, exercise sheets, ...
- **forum**: semi-supervised, i.e., you first try to help each and other and we will regularly step in for corrections or to provide missing answers
- **homework / project**: discussion, grading policies, project samples from last year
- **organization**: links, contact emails, announcements, ...
- ... more features will likely be added over time

Course organization – cont'd

Prerequisites:
- Control systems (227-0216-00L),
- Linear system theory (227-0225-00L),
- or equivalents, and basic Matlab skills.

Essentially, linear algebra & mathematical maturity suffice.

Lecture location & schedule

Lectures:	Mondays	16:00 to 18:00	HG D 1.1
Exercises:	Fridays	10:00 to 12:00	HG D 1.1
Live stream	accessible at	https://video.ethz.ch	

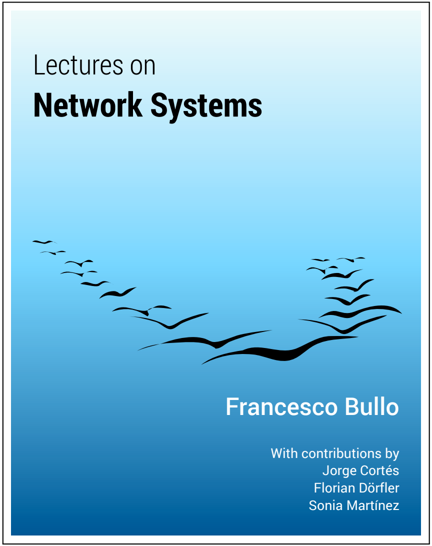
Special dates for make-up lectures due to holidays: Friday, 29.4.2021

Text: A set of self-contained notes are provided on the moodle.

Lecture Notes on Network Systems,
F. Bullo with J. Cortés, F. Dörfler, and S. Martínez, 2017.

Lecture notes

are available on **moodle** plus additional extra chapters on circuits & optimization



- **foundations**
  that you need to have (or acquire):
  - linear system theory
  - linear algebra + matrix theory
- **cornerstones**
  that we will develop in the lecture:
  - algebraic graph theory
  - (nonlinear) stability theory
- **icing on the cake**
  are real complex network systems:
  - robotic coordination
  - distributed optimization
  - epidemic spreading

Further reading, additional information and great insights

mostly freely available online via the authors' websites

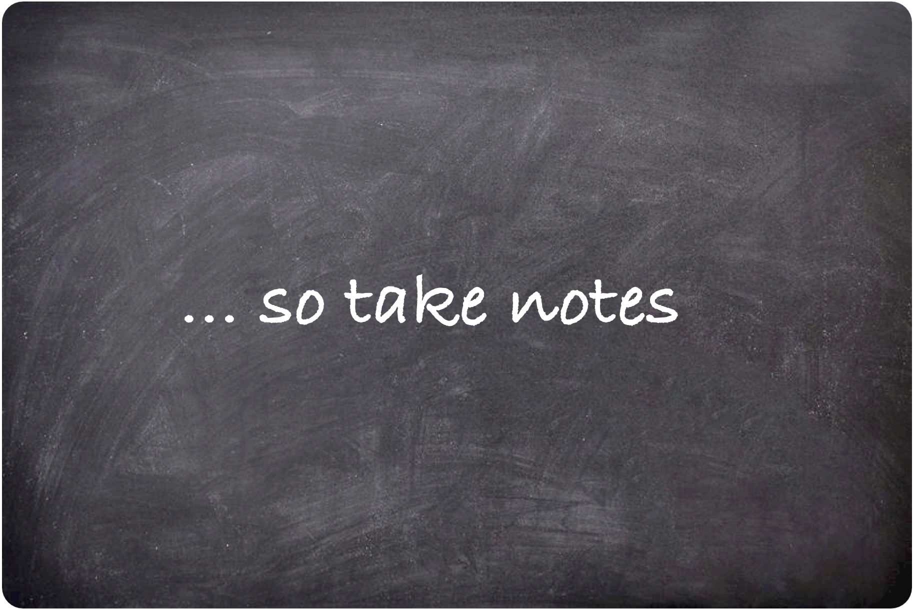
- *Distributed Control of Robotic Networks*, Francesco Bullo, Jorge Cortes and Sonia Martinez, Princeton University Press, 2009.
- *Graph Theoretic Methods in Multiagent Networks*, Mehran Mesbahi & Magnus Egerstedt, Princeton University Press, 2010.
- *Networked Control of Multi-Agent Systems*, Jan Lunze, Bookmundo, 2021.
- *A Course on Distributed Robotics*, Bruce Francis, Notes for CDC Bode Lecture, 2014.
- *Distributed Coordination of Multi-agent Networks*, Wei Ren & Yongcan Cao, Springer, 2011.
- *Cooperative Control Design*, H. Bai, M. Arcak, & J. Wen, Springer, 2011.
- *Introduction to Averaging Dynamics over Networks*, F. Fagnani & P. Frasca, Springer, 2017.
- *Parallel and Distributed Computation: Numerical Methods*, D. P. Bertsekas & J. N. Tsitsiklis, Athena Scientific, 1997.
- *Network Flows and Monotropic Optimization*, R. T. Rockafellar, Athena Scientific, 1998.
- *Network Optimization: Continuous and Discrete Models*, D. P. Bertsekas, Athena Scientific, 1998.

If you are interested beyond the course contents

(link)

IFAC Conference on Networked Systems (NecSys) 2022 @Zürich

We will use slides, white/blackboard, & EduApp (install !)



rules of the game:

please let us know of any conflicts within the next week; then the rules are set.

(see also the policies on moodle)

Homework policy

- The homework problems will constitute **50%** of your overall grade.
- **3/4 homework rule:** there will be four sets of graded homework problems. Only the three best ones count for your final grade.
- Problems are assigned on Monday (though Friday's exercise is relevant as well) & the homework is due **twelve days** afterwards in the exercise.
- **Due date:** return the homework at the latest by Friday 10:15am during via moodle. Late returns will not be accepted.
- **Form:** you can provide electronic or scanned hand-written solutions. If the presentation is unreadable, we subtract up to 20% of your score.
- **What to expect:** regular exercises, adaptation of theory to applications, short proofs, and occasional numerical simulations.

→ **Homework #1:** March 14 → return by March 25

→ **Homework #2:** April 5 → return by April 29 (after Sechseläuten)

Final project $\rightarrow$ see policy online

- The final project will constitute **50%** of your overall grade.
- **Task:** write a small conference paper on an interesting distributed control problem. We encourage you to identify such a problem yourselves, but we can also provide problem specifications if desired.
- We encourage you to **work in groups** of 2-3 people
- **Time line:** due until July 1, but we will provide feedback around May

When, what, & where?

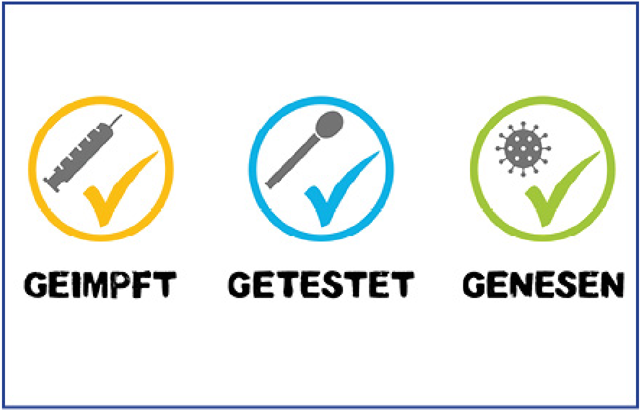
Lecture (Mondays, 16:00 to 18:00, HG D 1.1):

- will communicate the essential insights & intuition
- will cover the fundamental theoretic material & applications
- will be relevant for homework & final project
- live stream accessible at `https://video.ethz.ch/`

Exercises (Fridays, 10:00 to 12:00, HG D 1.1):

- will provide additional exercises and examples
- will be the place to ask questions regarding lecture
- will be very useful for homework & final project
- live stream accessible at `https://video.ethz.ch/`

COVID certificate: rules are in flux, please inform yourself



Official rules on BAG certificate (link)

ETH Zürich rules on COVID and up-to-date guidelines (link)

Let's start off with a EduApp quiz:

(I know that EduApp wasn't designed for such questions but anyways. . .)

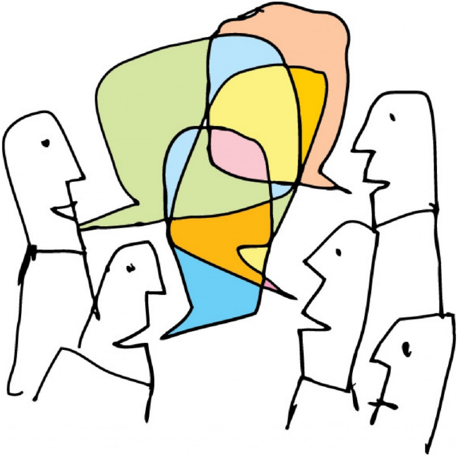
what is your background?

your interest in this course?

Chapter 1:

Motivating Problems and Systems

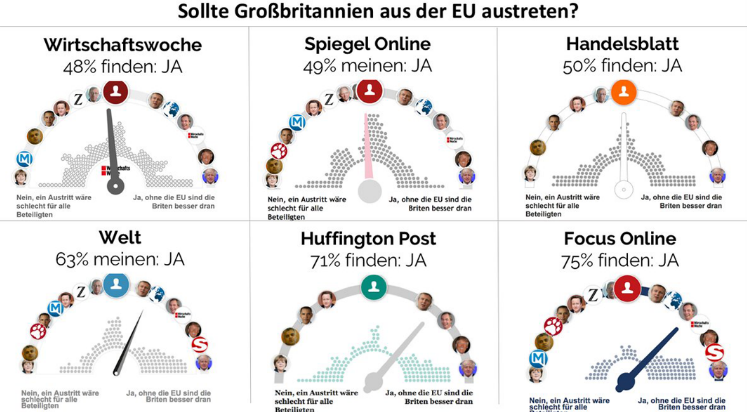
Social influence networks: opinion dynamics



on the board

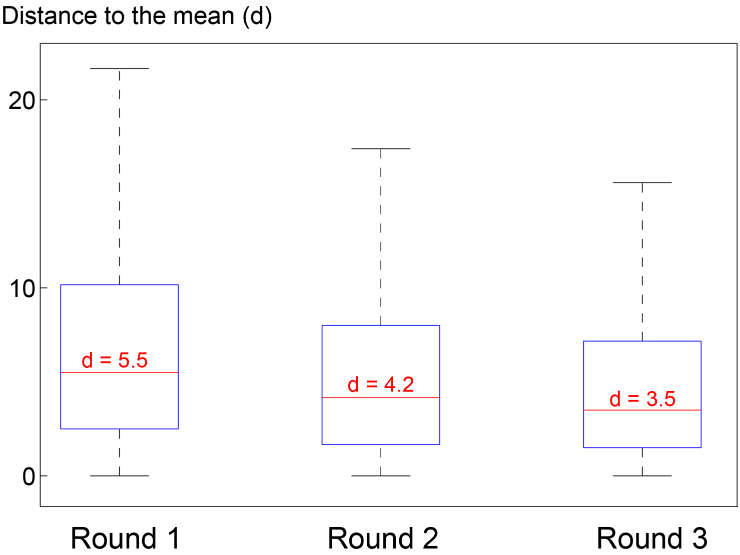
Opinion dynamics: collection of empirical data

Ever wondered about the role (one of many) of these **online polls** . . . ?



They estimate your **opinion averaging coefficients** a_{ij} (`link` to study I)!

Surprisingly accurate on a statistical level (link to study II)



Teaser: since opinion formation can be understood quantitatively (certainly on a statistical / population level), does opinion manipulation still sound whacky ???

Sensor networks: distributed parameter learning

Setup: we aim to estimate an unknown parameter $\theta \in \mathbb{R}^m$ via measurements by a sensor network. Each node $i \in \{1, \ldots, n\}$ measures

$$y_i = B_i \theta + v_i,$$

where $y_i \in \mathbb{R}^{m_i}$, B_i is a known matrix, & v_i is random measurement noise.

We **assume** the following:

(A1) **statistics:** noise vectors $v_1, \ldots, v_n$ are independent & jointly Gaussian variables with zero-mean $\mathbb{E}[v_i] = \mathbb{0}_{m_i}$ & covariance $\mathbb{E}[v_i v_i^T] = \Sigma_i = \Sigma_i^T > 0$, for $i \in \{1, \ldots, n\}$; and

(A2) **feasibility & redundancy:** the measurement parameters satisfy

$$\sum_i m_i \geq m \text{ and } \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} \text{ is full rank.}$$

Goal: given the measurements $y_1, \dots, y_n$, we want to compute the weighted least-square estimate of θ

$$\widehat{\theta}_{\mathrm{ML}} = \arg\min_{\widehat{\theta}} \ \sum_{i=1}^{n} \left(y_i - B_i\widehat{\theta}\right)^T \Sigma_i^{-1} \left(y_i - B_i\widehat{\theta}\right) \ .$$

Under assumptions (A1) and (A2), the **optimal solution** is

$$\widehat{\theta}_{\mathrm{ML}} = \left(\sum_{i=1}^{n} B_i^T \Sigma_i^{-1} B_i\right)^{-1} \sum_{i=1}^{n} B_i^T \Sigma_i^{-1} y_i.$$

This formula is easy to implement by a single processor with all the information about the problem, i.e., all parameters & all measurements.

This **centralized** solution strategy is not scalable and not very robust.

Distributed solution: the weighted least-square estimate is

$$\widehat{\theta}_{\mathrm{ML}} = \left(\sum_{i=1}^{n} B_i^T \Sigma_i^{-1} B_i \right)^{-1} \cdot \sum_{i=1}^{n} B_i^T \Sigma_i^{-1} y_i \,.$$

To compute $\widehat{\theta}_{\mathrm{ML}}$ in the sensor network, we perform two steps:

(i) We run two distributed algorithms in parallel to compute

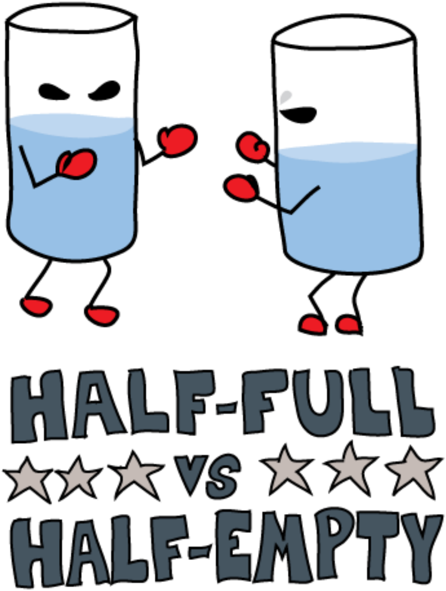
$$\text{average}(B_i^T \Sigma_i^{-1} B_i) \quad \text{and} \quad \text{average}(B_i^T \Sigma_i^{-1} y_i)\,.$$

(ii) Compute $\widehat{\theta}_{\mathrm{ML}} = \left(\text{average}(B_i^T \Sigma_i^{-1} B_i)\right)^{-1} \cdot \text{average}(B_i^T \Sigma_i^{-1} y_i)\,.$

Questions of interest are:

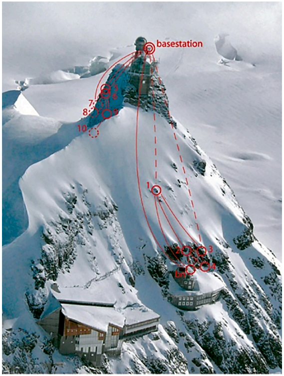
1. How do we design distributed algorithms to compute the averages?
2. What properties does the graph need to have in order for such an algorithm to converge? What about the convergence rate?

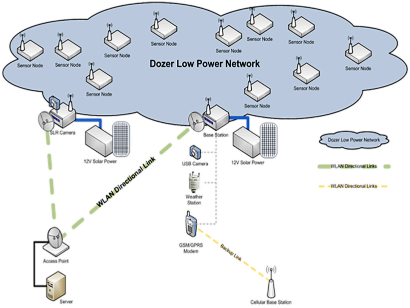
Sensor networks: distributed hypothesis testing

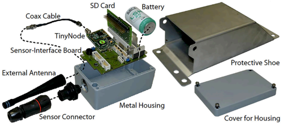


Example: permafrost melting $\rightarrow$ will mountain collapse?

source: sensor network by TIK (ETH Zürich) & Art of Technology

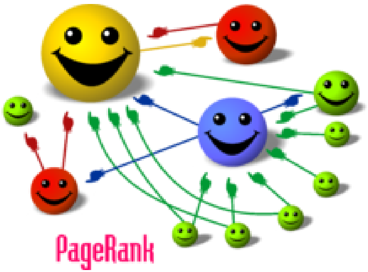






on the board

Google page rank



- sparse **world wide web** connection matrix: $a_{ij} = \frac{1}{\#\,\mathit{outgoing\ links}} > 0$ if page i links to j
- **importance** $x_i \in \mathbb{R}_{>0}$ of page i
- **page rank:** *"a page is important if other important pages point to it"*
- **formula:** $x_i = \sum_{j=1}^{n} a_{ji} x_j$ or $\boxed{x = A^T x}$

$\Rightarrow$ **iterative computation:** web is too large to calculate A^{-1}: $\boxed{x^+ = A^T x}$

- **issue:** iteration $x^+ = A^T x$ not convergent (due to poor connectivity)

$\Rightarrow$ addition of a **random surfer** that visits all pages with equal probability:

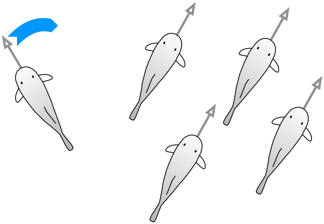
$$\boxed{x^+ = \left(\varepsilon \cdot A^T + (1-\varepsilon)/n \cdot \mathbb{1}\mathbb{1}^T\right) x} \quad \text{where } \varepsilon \in \,]0,1[$$

Flocking behavior for a group of animals

$$\dot{\theta}_i = \begin{cases} (\theta_j - \theta_i), & \text{if one neighbor} \\ \frac{1}{2}(\theta_{j_1} - \theta_i) + \frac{1}{2}(\theta_{j_2} - \theta_i), & \text{if two neighbors} \\ \frac{1}{m}(\theta_{j_1} - \theta_i) + \cdots + \frac{1}{m}(\theta_{j_m} - \theta_i), & \text{if } m \text{ neighbors} \end{cases}$$

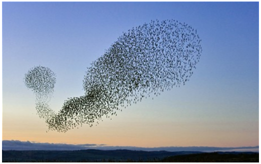
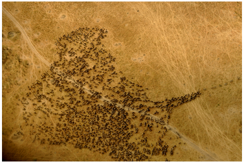
or

$$\dot{\theta}_i = \text{average}\left(\{\theta_j, j \in \mathcal{N}^{\text{out}}(i)\}\right) - \theta_i$$

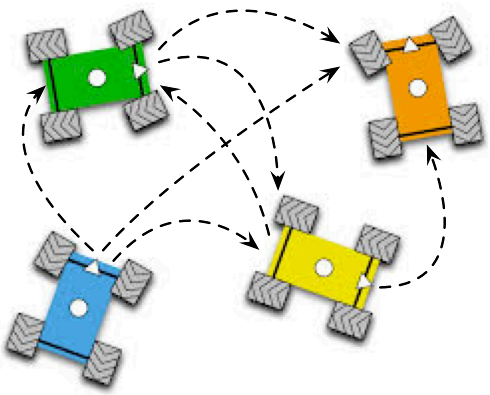


or

$$\dot{\theta} = -\,\texttt{Laplacian matrix} \cdot \theta$$

Robotic networks: cyclic pursuit and balancing



on the board

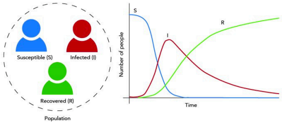
more examples in Chapter 1

Many further examples

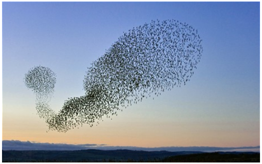
throughout the lecture

distributed computing

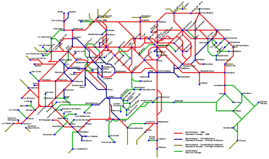
social networks



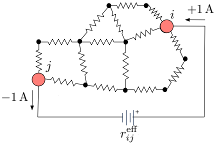
epidemic spreading



flocking of animals



traffic networks



circuits & grids

Reading assignment (lecture notes):

Chapter 1: Motivating problems and systems

Chapter 2.1: Linear systems & the Jordan normal form

Exercise session (Friday):

- led by Dominic Liao-Mc Pherson
- review of linear algebra
- further examples of distributed control systems

<u>Don't forget:</u>

- install ETH **EduApp**
- approach us with questions on project etc.
- use forum on moodle