Data-driven and Brain-inspired Autonomous Systems

Fabio Pasqualetti

September 27, 2024

Department of Mechanical Engineering
University of California at Riverside

Engineering autonomous systems

(credit Weinstein, Kimbrell, Bohg, Frontiers Neurorobotics, Yellina)

Engineering autonomous systems

(credit Weinstein, Kimbrell, Bohg, Frontiers Neurorobotics, Yellina)

- operate in unstructured, unknown, dynamic, contested environments
- learn from data and experience
- collaborate with systems and humans
- fail gracefully against tampering

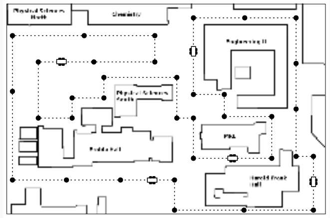
Engineering autonomous systems

(credit Weinstein, Kimbrell, Bohg, Frontiers Neurorobotics, Yellina)

- operate in unstructured, unknown, dynamic, contested environments
- learn from data and experience
- collaborate with systems and humans
- fail gracefully against tampering
- precision agriculture, transportation, search and monitoring, healthcare

My contributions towards autonomy

Coordinated Robotics

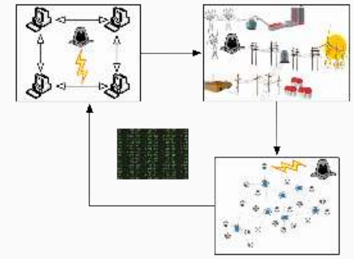


Distributed coordination and cooperation algorithms for teams of robots:

- persistent surveillance, search
- algorithm design, analysis, testing
- autonomous exploration, detection

My contributions towards autonomy

Coordinated Robotics | **Security Robustness**



Expose and remedy vulnerabilities of (networked) cyber-physical systems:

- attack analysis, detection, design
- model-based and data-driven settings
- power networks, multi-agent systems

My contributions towards autonomy

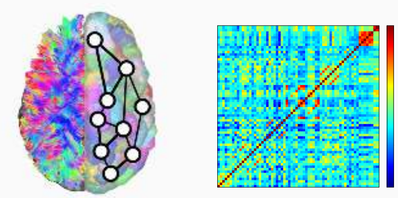
Coordinated Robotics | **Security Robustness** | **Networks Learning**

Analyze, predict, and control dynamic processes over large networks:

- (data-driven) control, resilience
- performance limitations vs topology
- natural, social, tech. systems

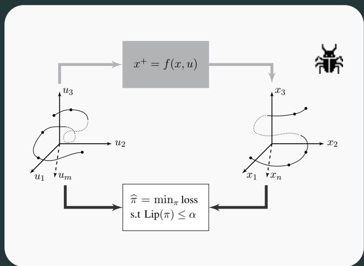
My contributions towards autonomy

Coordinated Robotics | **Security Robustness** | **Networks Learning** | **Brain Dynamics**

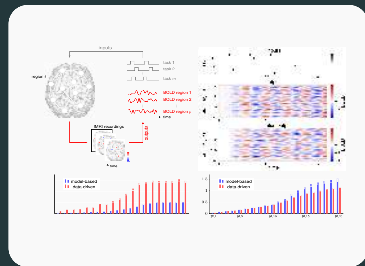


How does the anatomy of the brain constrain and enable its functions?

- models, analysis of neural processes
- better algorithms and architectures
- treatments, cognitive enhancement

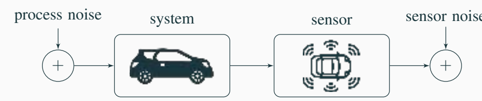


Data-driven control



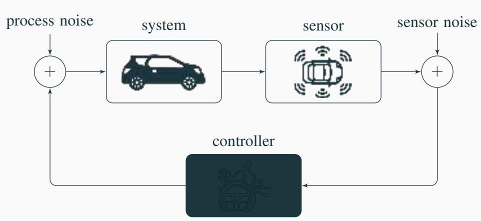
Brain dynamics

Model-based LQG control



$$x(t+1) = Ax(t) + Bu(t) + w(t)$$
$$y(t) = Cx(t) + v(t)$$

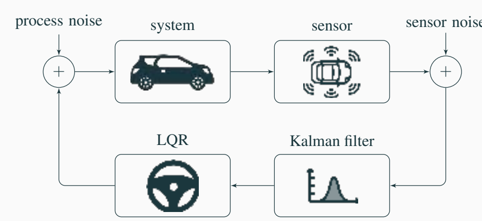
Model-based LQG control



$$x(t+1) = Ax(t) + Bu(t) + w(t)$$
$$y(t) = Cx(t) + v(t)$$

$$\lim_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=0}^{T-1} x(t)^T Q x(t) + u(t)^T R u(t)\right]$$

Model-based LQG control

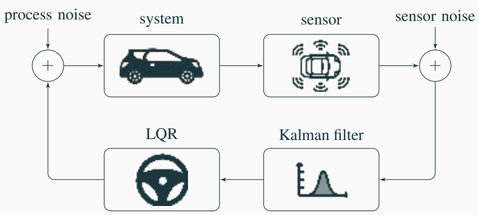


$$x(t+1) = Ax(t) + Bu(t) + w(t)$$
$$y(t) = Cx(t) + v(t)$$

$$\lim_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=0}^{T-1} x(t)^T Q x(t) + u(t)^T R u(t)\right]$$

$$u(t) = \underbrace{K_{\text{lqr}}}_{\text{LQR gain}} \overbrace{\hat{x}(t)}^{\text{Kalman estimate}}$$

Model-based LQG control



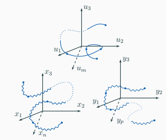
$$x(t+1) = Ax(t) + Bu(t) + w(t)$$
$$y(t) = Cx(t) + v(t)$$

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} x(t)^T Q x(t) + u(t)^T R u(t)\right]$$

$$u(t) = \underbrace{K_{\text{lqr}}}_{\text{LQR gain}}\overbrace{\hat{x}(t)}^{\text{Kalman estimate}}$$

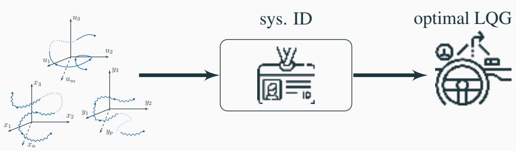
- known dynamical model and noise statistics
- recursive Riccati equations for KF/LQR
- separation principle: LQG = LQR + KF

Data-driven LQG control



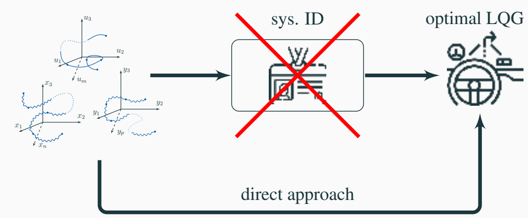
- input (U), state (X), output (Y) data
- multiple noisy, open-loop trajectories
- (linear) unknown dynamics and noise

Data-driven LQG control



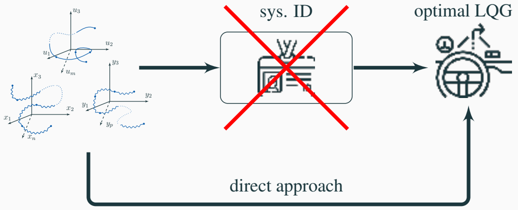
- input (U), state (X), output (Y) data
- multiple noisy, open-loop trajectories
- (linear) unknown dynamics and noise

Data-driven LQG control



- input (U), state (X), output (Y) data
- multiple noisy, open-loop trajectories
- (linear) unknown dynamics and noise

Data-driven LQG control



- input (U), state (X), output (Y) data
- multiple noisy, open-loop trajectories
- (linear) unknown dynamics and noise

Direct data-driven formula for LQR gain:

- $\begin{bmatrix} u_v \\ x_v \end{bmatrix} = \begin{bmatrix} H \\ M \end{bmatrix} P^{-1/2} \left(\begin{bmatrix} I & 0 \end{bmatrix} P^{-1/2} \right)^{\dagger} x_0$
- $K_{\mathrm{LQR}}^{\mathrm{D}} = u_m x_m^{\dagger}$

[Al Makdah *et al.*, 2023 CDC]

Data-driven LQG control

- input (U), state (X), output (Y) data
- multiple noisy, open-loop trajectories
- (linear) unknown dynamics and noise

Notation:

$$\begin{bmatrix} u_v \\ x_v \end{bmatrix} = \begin{bmatrix} H \\ M \end{bmatrix} P^{-1/2} \left(\begin{bmatrix} I & 0 \end{bmatrix} P^{-1/2} \right)^{\dagger} x_0$$

input/state trajectories

initial state

$$H = \begin{bmatrix} 0 & I \end{bmatrix} \qquad M = X \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \qquad P = M^{\mathsf{T}} Q M + R$$

data matrices

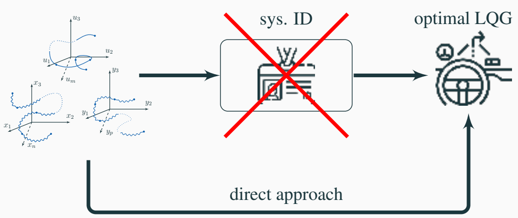
cost matrices

Direct data-driven formula for LQR gain:

- $\begin{bmatrix} u_v \\ x_v \end{bmatrix} = \begin{bmatrix} H \\ M \end{bmatrix} P^{-1/2} \left(\begin{bmatrix} I & 0 \end{bmatrix} P^{-1/2} \right)^{\dagger} x_0$
- $K_{\mathrm{LQR}}^{\mathrm{D}} = u_m x_m^{\dagger}$

[Al Makdah *et al.*, 2023 CDC]

Data-driven LQG control

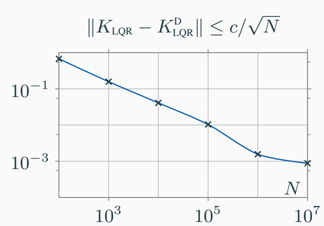


- input (U), state (X), output (Y) data
- multiple noisy, open-loop trajectories
- (linear) unknown dynamics and noise

Direct data-driven formula for LQR gain:

- $\begin{bmatrix} u_v \\ x_v \end{bmatrix} = \begin{bmatrix} H \\ M \end{bmatrix} P^{-1/2} \left(\begin{bmatrix} I & 0 \end{bmatrix} P^{-1/2} \right)^{\dagger} x_0$
- $K_{\mathrm{LQR}}^{\mathrm{D}} = u_m x_m^{\dagger}$

[Al Makdah *et al.*, 2023 CDC]

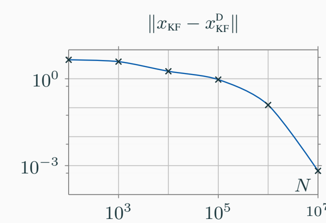


Data-driven LQG control

Direct data-driven formula for Kalman filter:

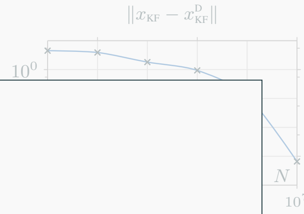
- $\hat{x}(t) = X_t \begin{bmatrix} U_{t-1} \\ Y \end{bmatrix}^{\dagger} \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix}$
- $\|L_{\mathrm{KF}} - L_{\mathrm{KF}}^{\mathrm{D}}\| \leq c/\sqrt{N}$

[Al Makdah *et al.*, 2023 CDC]



Data-driven LQG control

Direct data-driven formula for Kalman filter:



Notation:

$$\hat{x}(t) = X_t \begin{bmatrix} U_{t-1} \\ Y \end{bmatrix}^{\dagger} \begin{bmatrix} u_0^{t-1} \\ y_0^{t} \end{bmatrix}$$

state estimate

data matrices

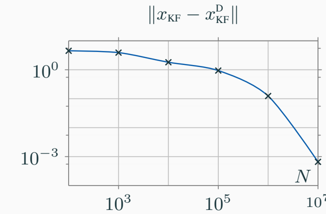
current input/output

Data-driven LQG control

Direct data-driven formula for Kalman filter:

- $\hat{x}(t) = X_t \begin{bmatrix} U_{t-1} \\ Y \end{bmatrix}^{\dagger} \begin{bmatrix} u_0^{t-1} \\ y_0^{t} \end{bmatrix}$
- $\|L_{\mathsf{KF}} - L_{\mathsf{KF}}^{\mathsf{D}}\| \leq c/\sqrt{N}$

[Al Makdah *et al.*, 2023 CDC]

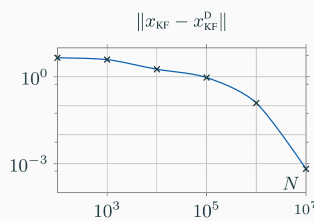


Data-driven LQG control

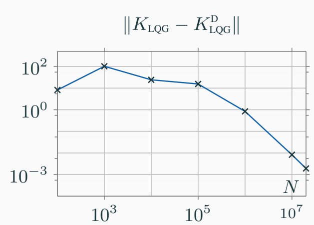
Direct data-driven formula for Kalman filter:

- $\hat{x}(t) = X_t \begin{bmatrix} U_{t-1} \\ Y \end{bmatrix}^{\dagger} \begin{bmatrix} u_0^{t-1} \\ y_0^{t} \end{bmatrix}$
- $\|L_{\mathsf{KF}} - L_{\mathsf{KF}}^{\mathsf{D}}\| \leq c/\sqrt{N}$

[Al Makdah *et al.*, 2023 CDC]



- LQR + KF leads to data-driven LQG formulas
- data overcomes lack of model/noise knowledge

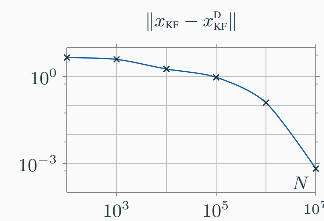


Data-driven LQG control

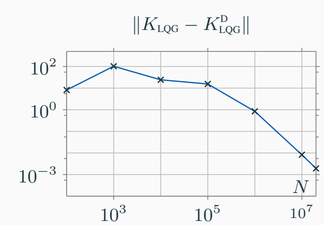
Direct data-driven formula for Kalman filter:

- $\hat{x}(t) = X_t \begin{bmatrix} U_{t-1} \\ Y \end{bmatrix}^{\dagger} \begin{bmatrix} u_0^{t-1} \\ y_0^{t} \end{bmatrix}$
- $\|L_{\mathsf{KF}} - L_{\mathsf{KF}}^{\mathsf{D}}\| \leq c/\sqrt{N}$

[Al Makdah *et al.*, 2023 CDC]



- LQR + KF leads to data-driven LQG formulas
- data overcomes lack of model/noise knowledge
- direct vs indirect $\Rightarrow$ variance vs bias tradeoff



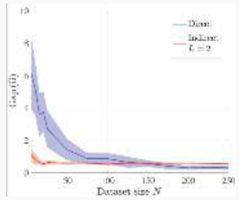
Data-driven LQG control

Direct data-driven formula for Kalman filter:

- $\hat{x}(t) = X_t \begin{bmatrix} U_{t-1} \\ Y \end{bmatrix}^{\dagger} \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix}$
- $\|L_{\mathrm{KF}} - L_{\mathrm{KF}}^{\mathrm{D}}\| \leq c/\sqrt{N}$

[Al Makdah *et al.*, 2023 CDC]

- LQR + KF leads to data-driven LQG formulas
- data overcomes lack of model/noise knowledge
- direct vs indirect ⇒ variance vs bias tradeoff

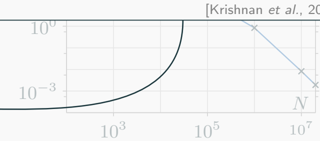


Direct: no asymptotic bias, large variance

Indirect: asymptotic bias, small variance

Two distinct data regimes!

[Krishnan *et al.*, 2021 CDC]

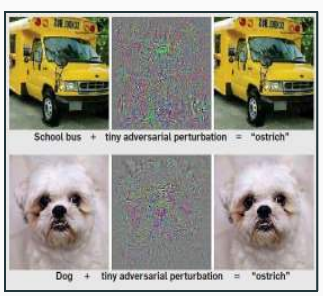


Sensitivity to data perturbations and uncertainties

- data-driven and ML algorithms perform well in nominal conditions

Sensitivity to data perturbations and uncertainties

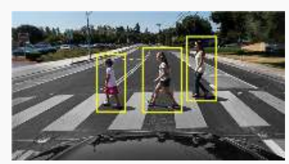
- data-driven and ML algorithms perform well in nominal conditions
- yet, brittle performance against perturbations and non-ideal conditions
- lack of robustness guarantees limits deployment in physical environments



Sensitivity to data perturbations and uncertainties

- data-driven and ML algorithms perform well in nominal conditions

Insufficient data:

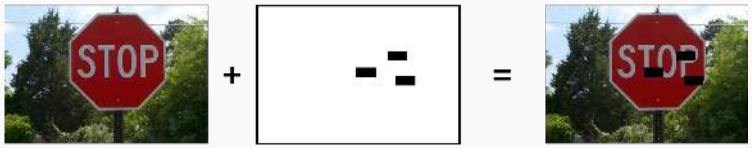


- actual data not in training dataset
- different operating conditions (weather, lighting)
- increasing training data mitigates this problem

Sensitivity to data perturbations and uncertainties

- data-driven and ML algorithms perform well in nominal conditions

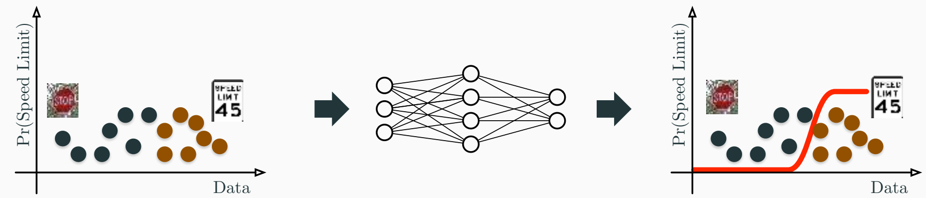
An issue of algorithm sensitivity:

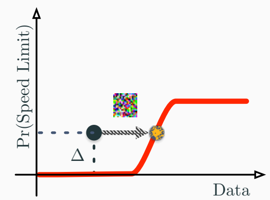


stop sign perturbation 45 mph speed limit sign

- small (adversarial) perturbations cause large decision errors
- perturbations "meaningless" to human eye
- more data not a solution, need better (less sensitive) algorithms

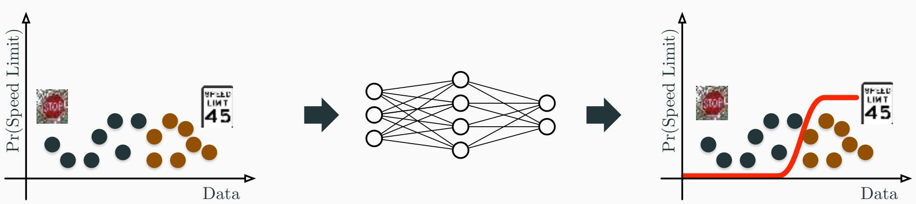
Sensitivity to data perturbations and Lipschitz constant

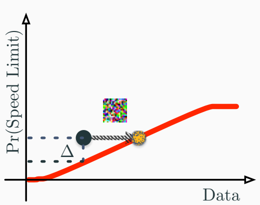




- training optimizes prediction accuracy
- data perturbations reduce accuracy

Sensitivity to data perturbations and Lipschitz constant





- training optimizes prediction accuracy
- data perturbations reduce accuracy
- robustness depends on "slope" of trained model

Sensitivity to data perturbations and Lipschitz constant

Literature on regulating Lipschitz constant (very incomplete and a bit outdated):

- Lipschitz constant *after* training [Weng et al., 2018][Fazlyab et al., 2019]
  (computationally intensive, loose bounds, not useful for algorithm design)
- Lipschitz-regularized training [Gouk et al., 2018][Finlay et al., 2018]
  (no guarantees on neither nominal nor adversarial performance, not interpretable)
- distributionally robust training [Wong et al., 2018][Pauli et al., 2020]
  (not interpretable, no prescribed design of robustness, lack of adversarial guarantees)

Sensitivity to data perturbations and Lipschitz constant

Literature on regulating Lipschitz constant (very incomplete and a bit outdated):

- Lipschitz constant *after* training [Weng et al., 2018][Fazlyab et al., 2019]
  (computationally intensive, loose bounds, not useful for algorithm design)
- Lipschitz-regularized training [Gouk et al., 2018][Finlay et al., 2018]
  (no guarantees on neither nominal nor adversarial performance, not interpretable)
- distributionally robust training [Wong et al., 2018][Pauli et al., 2020]
  (not interpretable, no prescribed design of robustness, lack of adversarial guarantees)

Safety certificates for open-loop data-driven algorithms

$$\min_{f \in \mathrm{Lip}} \; L(f)$$
$$\text{subject to} \; \mathrm{lip}(f) \leq \alpha$$

Safety certificates for open-loop data-driven algorithms

$$\min_{f \in \mathrm{Lip}} \; L(f)$$
$$\text{subject to} \; \mathrm{lip}(f) \leq \alpha$$

loss function

trained model

Lipschitz constant

Safety certificates for open-loop data-driven algorithms

$$\min_{f \in \mathrm{Lip}} \; L(f)$$
$$\text{subject to} \; \mathrm{lip}(f) \leq \alpha$$

loss function

trained model

Lipschitz constant

A unique saddle point exists and satisfies:

$$\nabla \cdot (\lambda \nabla f) + \partial_f L = 0$$

[Krishnan *et al.*, 2020 Neurips]

Safety certificates for open-loop data-driven algorithms

$$\min_{f \in \mathrm{Lip}} L(f)$$
subject to $\mathrm{lip}(f) \leq \alpha$

loss function
trained model
Lipschitz constant

A unique saddle point exists and satisfies:

$$\nabla \cdot (\lambda \nabla f) + \partial_f L = 0$$

Laplace operator weighted by Lagrange multiplier λ

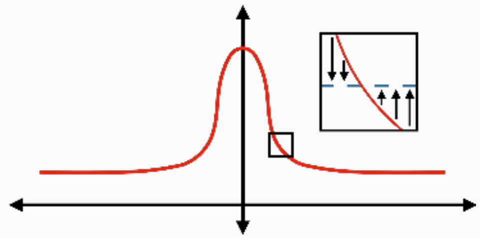
derivative of loss

[Krishnan *et al.*, 2020 Neurips]

Safety certificates for open-loop data-driven algorithms

$$\min_{f \in \mathrm{Lip}} L(f)$$
subject to $\mathrm{lip}(f) \leq \alpha$

loss function
trained model
Lipschitz constant



A unique saddle point exists and satisfies:

$$\nabla \cdot (\lambda \nabla f) + \partial_f L = 0$$

Laplace operator weighted by Lagrange multiplier λ

derivative of loss

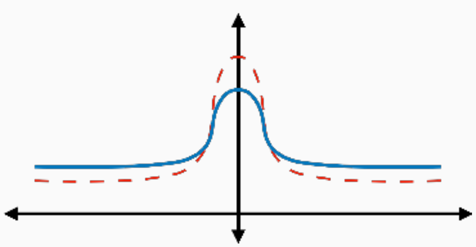
smoothing of f while ensuring Lipshitz bound α

[Krishnan *et al.*, 2020 Neurips]

Safety certificates for open-loop data-driven algorithms

$$\min_{f \in \mathrm{Lip}} L(f)$$
subject to $\mathrm{lip}(f) \leq \alpha$

loss function
trained model
Lipschitz constant



A unique saddle point exists and satisfies:

$$\nabla \cdot (\lambda \nabla f) + \partial_f L = 0$$

Laplace operator weighted by Lagrange multiplier λ

derivative of loss

smoothing of f while ensuring Lipshitz bound α

[Krishnan *et al.*, 2020 Neurips]

Safety certificates for open-loop data-driven algorithms

Analogy with heat flow:

$$\nabla \cdot (\lambda \nabla f) + \partial_f L = 0$$

- solution to robust learning = steady state solution of heat flow
- data-driven map = steady state temperature profile
- Lagrange multiplier λ = thermal conductivity of medium
- Laplace operator = heat conduction driven by constraint activation
- derivative of loss function = heat source

Safety certificates for open-loop data-driven algorithms

$$\min_{f \in \mathrm{Lip}} L(f)$$

subject to $\mathrm{lip}(f) \leq \alpha$

loss function

trained model

Lipschitz constant

- hard Lipschitz constraint $\Rightarrow$ certificate
- link adv. robustness to PDE theory
- $f^* =$ optimal temperature profile
- leads to PDE and multi-agent algorithms
- fund. performance vs robustness tradeoff

A unique saddle point exists and satisfies:

$$\nabla \cdot (\lambda \nabla f) + \partial_f L = 0$$

Laplace operator weighted by Lagrange multiplier λ

derivative of loss

smoothing of f while ensuring Lipschitz bound α

[Krishnan *et al.*, 2020 Neurips]

Safety certificates for open-loop data-driven algorithms

$$\min_{f \in \mathrm{Lip}} L(f)$$

subject to $\mathrm{lip}(f) \leq \alpha$

loss function

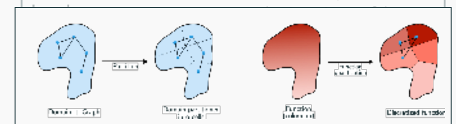
trained model

Lipschitz constant

- hard Lipschitz constraint $\Rightarrow$ certificate
- link adv. robustness to PDE theory
- $f^* =$ optimal temperature profile
- leads to PDE and multi-agent algorithms
- fund. performance vs robustness tradeoff

A unique saddle point exists and satisfies:

$$\nabla \cdot (\lambda \nabla f) + \partial_f L = 0$$

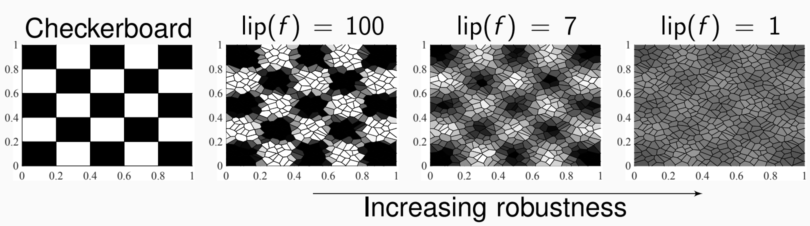


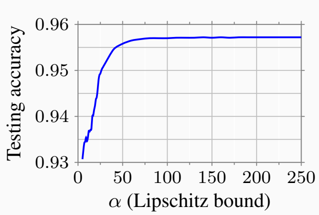
smoothing of f while ensuring Lipschitz bound α

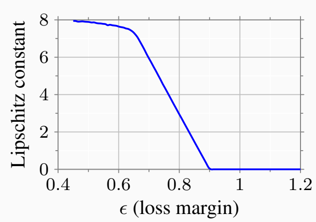
[Krishnan *et al.*, 2020 Neurips]

8/25

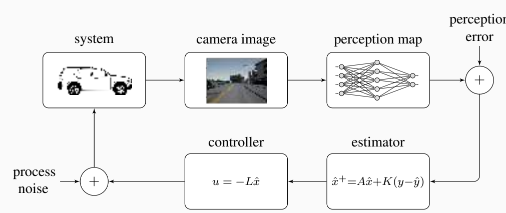
Safety certificates for open-loop data-driven algorithms

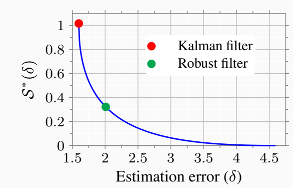






Safety certificates for closed-loop data-driven algorithms





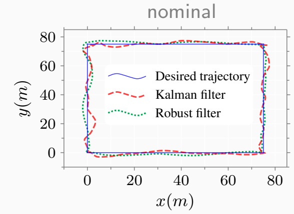
- Pareto frontier of estimators:

$$K(\lambda) = XC^\top (CXC^\top + I + \lambda R)$$

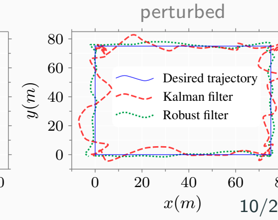
- strict perf. vs rob. tradeoff

[Al Makdah *et al.*, 2021 O. Hugo Schuck Best Paper Award]

nominal



perturbed



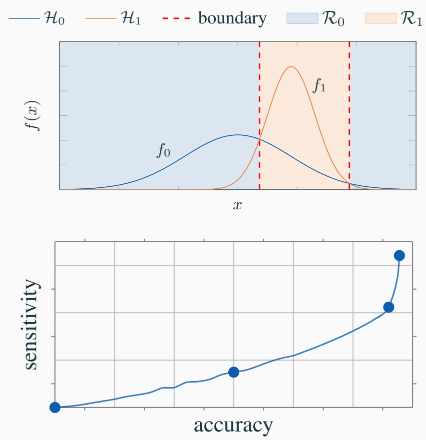
Robustness vs performance tradeoffs for control and decision-making

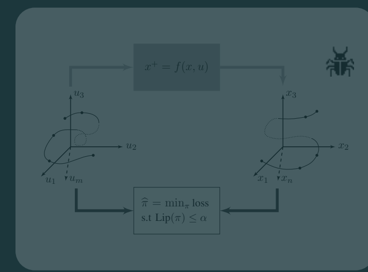
Classification as hypothesis testing problem

$$\mathcal{H}_0 : x \sim f_0 \text{ and } \mathcal{H}_1 : x \sim f_1$$

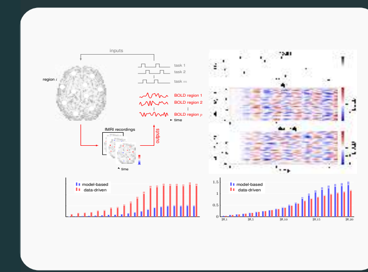
- fundamental accuracy vs robustness tradeoff; independent of classification algorithm; depends on data dist. only
- simpler algorithms often more robust
- abstaining can also improve robustness

[Al Makdah *et al.*, 2019 LCSS and 2021 ACC]





Data-driven control

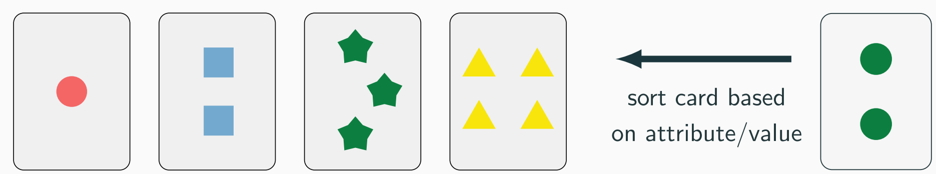


Brain dynamics

Reverse engineering the human brain is one of the greatest challenges of modern science, and the key to advancing health and artificial intelligence.

(credit Ellen Weinstein)

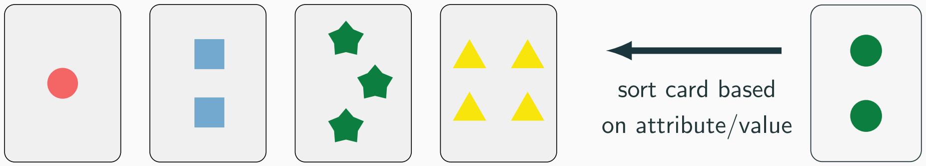
Reverse engineer the human brain to advance AI



Wisconsin card sorting task

- each card has 3 attributes (number, shape, color)
- each attribute has 4 values (blue, green, yellow, orange)
- task: classify cards based on value of one attribute
- reward: after each classification player knows if correct

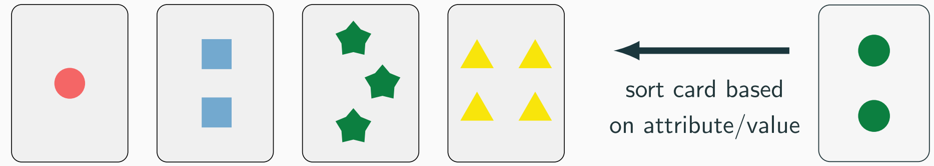
Reverse engineer the human brain to advance AI



Wisconsin card sorting task

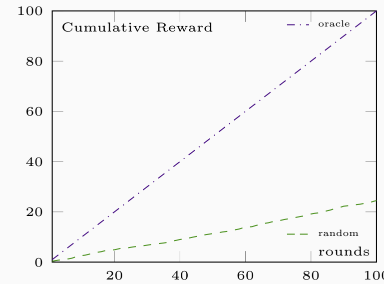
- each card has 3 attributes (number, shape, color)
- each attribute has 4 values (blue, green, yellow, orange)
- task: classify cards based on value of one attribute
- reward: after each classification player knows if correct
- **catch**: classification rule changes and unknown to player

Reverse engineer the human brain to advance AI

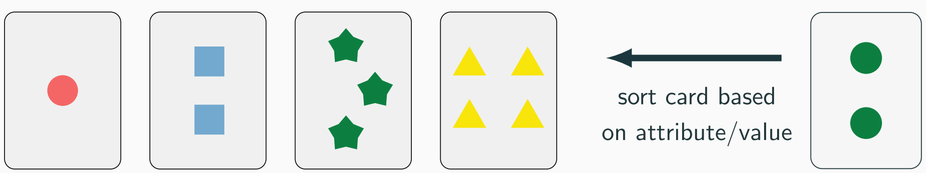


Wisconsin card sorting task

- each card has 3 attributes (number, shape, color)
- each attribute has 4 values (blue, green, yellow, orange)
- task: classify cards based on value of one attribute
- reward: after each classification player knows if correct
- **catch**: classification rule changes and unknown to player

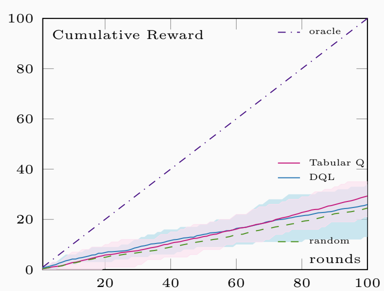


Reverse engineer the human brain to advance AI

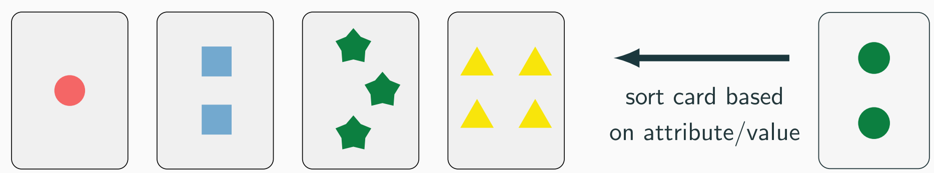


Wisconsin card sorting task

- each card has 3 attributes (number, shape, color)
- each attribute has 4 values (blue, green, yellow, orange)
- task: classify cards based on value of one attribute
- reward: after each classification player knows if correct
- **catch**: classification rule changes and unknown to player

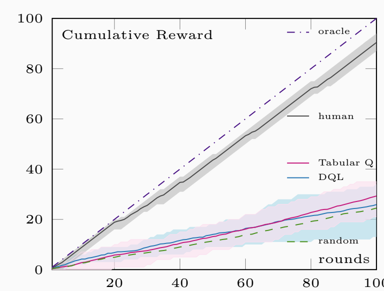


Reverse engineer the human brain to advance AI



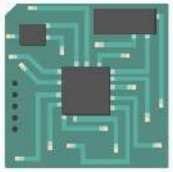
Wisconsin card sorting task

- each card has 3 attributes (number, shape, color)
- each attribute has 4 values (blue, green, yellow, orange)
- task: classify cards based on value of one attribute
- reward: after each classification player knows if correct
- **catch**: classification rule changes and unknown to player



Reverse engineer the human brain for sustainable AI

brain ~ 10 W



laptop ~ 60 W

AI data center ~ 100 MW

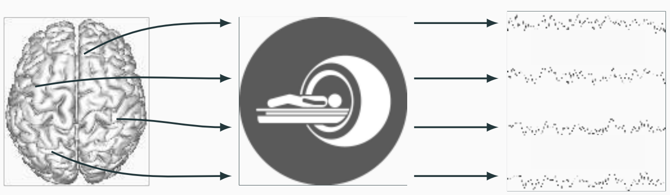
- training GPT-3 same energy as driving to the moon and back
- GPT query consumes 15x more energy than Google query
- increasingly more models, larger models, more sophisticated GPUs, more users ...

Reverse engineer the human brain for improved health

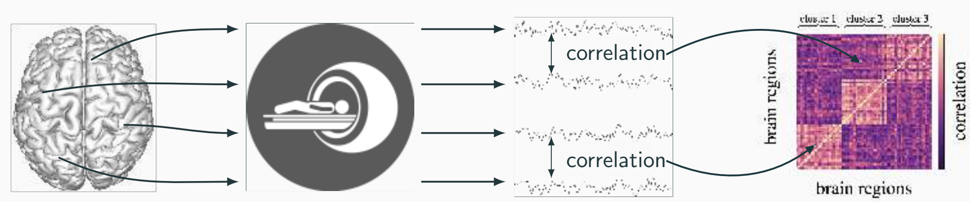
- personalized diagnostics therapies
- novel brain computer interfaces and implantable devices
- enhancement of brain functions and reversal of cognitive decline

[Medtronic]

Functional patterns of brain activity



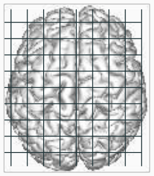
Functional patterns of brain activity



- synchrony between brain regions long known (EEG, fMRI, [Berger & Gray, 1929])
- rich repertoire of synchrony patters (transient, long-range, clustered)
- different patterns are biomarkers of health and disease (epilepsy, Parkinson's)

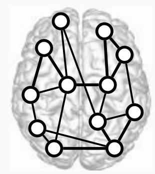
Modeling functional patterns

brain regions | white matter fibers | brain network



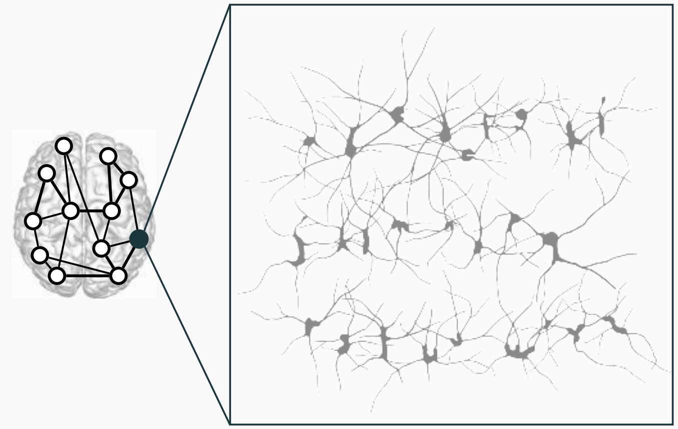
+

=



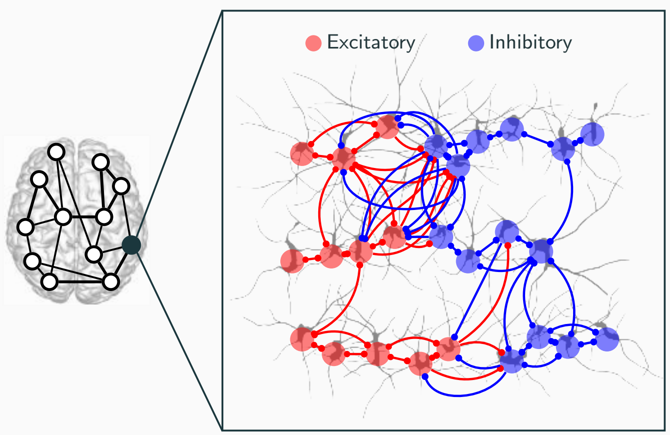
- nodes = brain regions; edges = bundles of white matter fibers
- static brain networks carry structural and statistical information
- *dynamic* brain networks are useful for the prediction & control of neural dynamics

Resting-state activity modeled by phase oscillators



each node of the brain network captures the dynamics of a population of neurons

Resting-state activity modeled by phase oscillators



excitatory and inhibitory communities are in a regime of self-sustained oscillations (weakly coupled Wilson-Cowan)

[Hoppensteadt and Izhikevich, 1997]

↓

neurons' firing rates describe a limit cycle

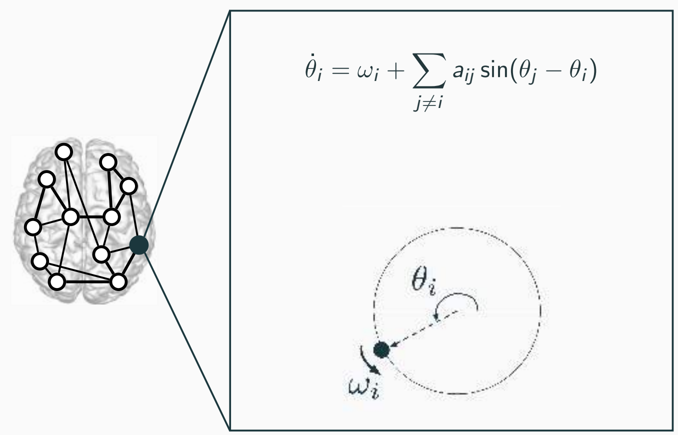
↓

dynamics approximated by a single phase variable

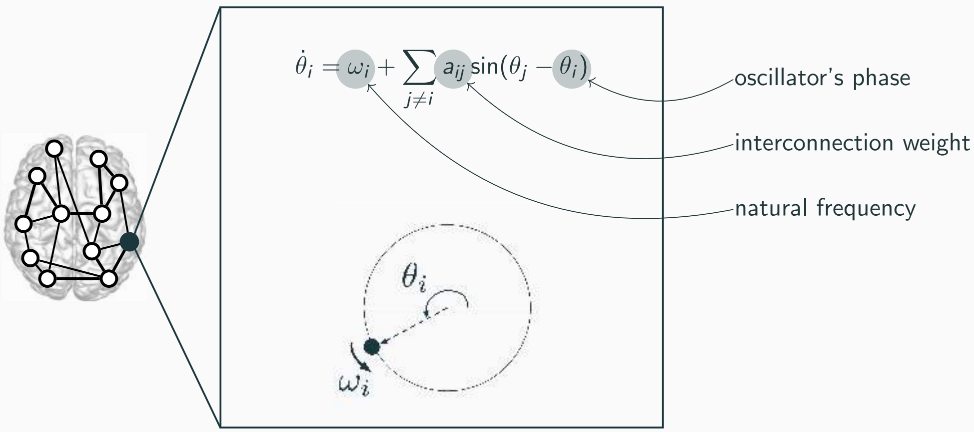
[Cabral *et al.*, 2011]

Resting-state activity modeled by phase oscillators

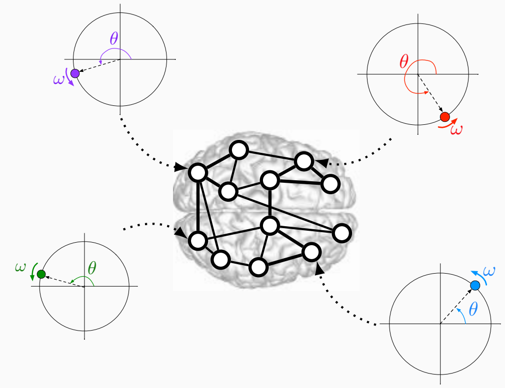
$$\dot{\theta}_i = \omega_i + \sum_{j \neq i} a_{ij} \sin(\theta_j - \theta_i)$$



Resting-state activity modeled by phase oscillators



Dynamical brain network to simulate neural activity



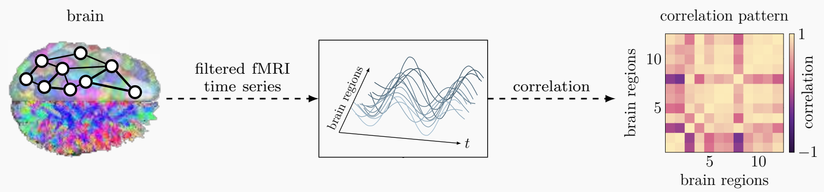
dynamical brain network with:

nodes = brain regions

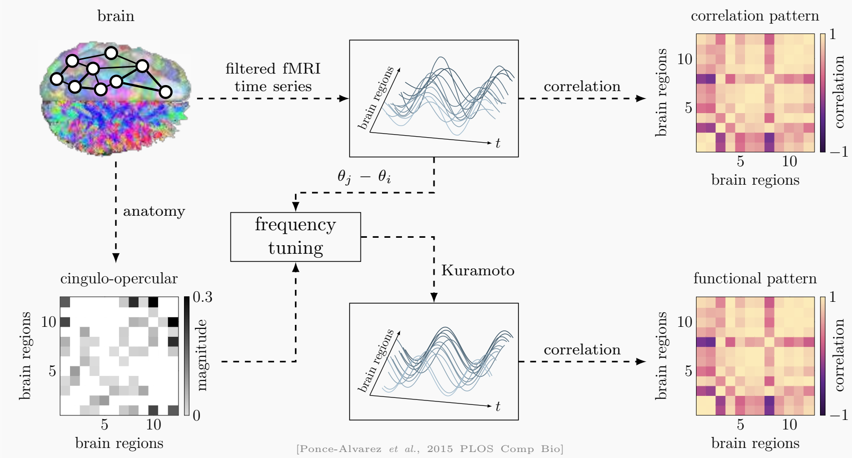
edges = white matter fibers

node dynamics = Kuramoto

Modeling of functional patterns

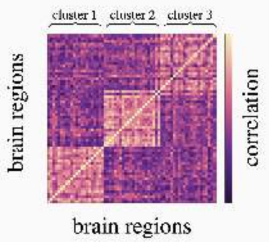


Modeling of functional patterns

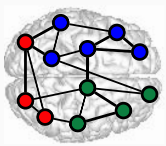


[Ponce-Alvarez *et al.*, 2015 PLOS Comp Bio]

Analysis of functional patterns #1



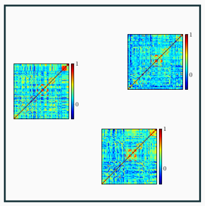
correlated brain regions as synchronized oscillators



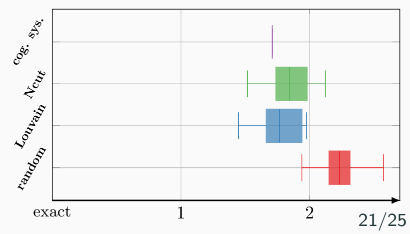
Cluster synchronization is possible if and only if:

- equal frequencies for all oscillators in the same cluster
- the network weights are *balanced*

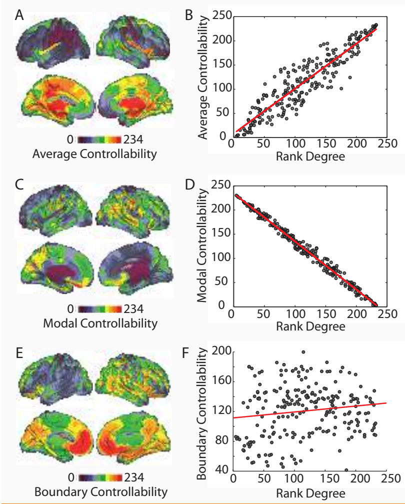
[Menara *et al.*, 2020 TCNS]



[Ponce-Alvarez *et al.*, 2015 PLOS Comp Bio]



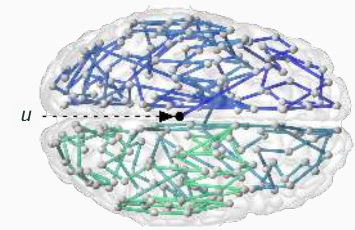
Analysis of functional patterns #2



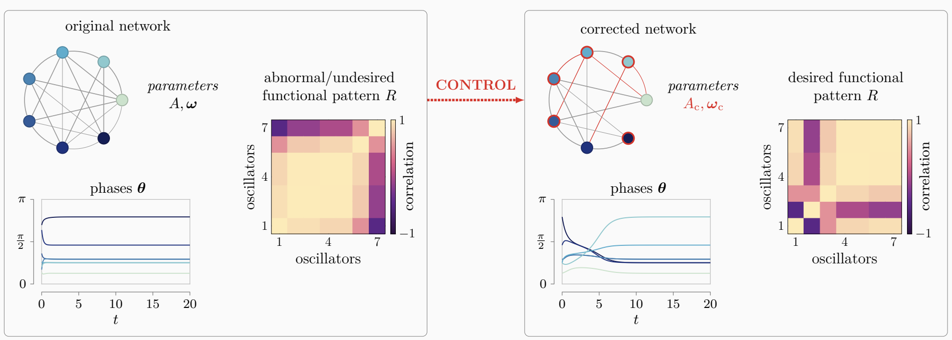
Controllability of the human brain (using fMRI data):

- the brain is *structurally* controllable
- brain *hubs* are optimal *average* control states
- sparse brain are optimal for *modal* control states

[Menara *et al.*, 2019 TAC] [Menara *et al.*, 2019 Neuroimage] [Gu *et al.*, 2015 Nature Comm.]



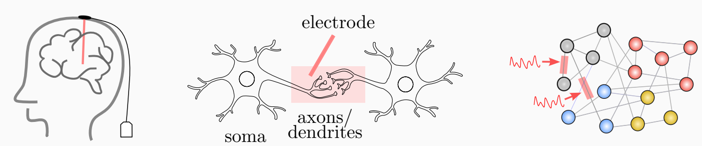
Control of functional patterns #1



[Menara *et al.*, 2020 Roberto Tempo Award] [Menara *et al.*, 2022 Nature Comm.]

- control knobs = network weights + oscillator frequencies
- biological constraints ⇒ positive weights, sparsity of interventions, magnitude

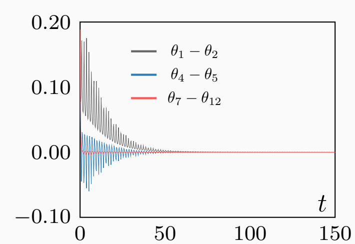
Vibrational control of brain networks #2



Deep Brain Stimulation

↕

vibrational control of network edges



Vibrational control for network systems:

- vibrational control explains DBS
- it allows for *functional* network modifications
- *open-loop* stabilization mechanisms

[Qin *et al.*, 2024 TCNS]

Summary

In this talk: autonomy as the convergence of control, learning, data, and neuroscience

Summary

In this talk: autonomy as the convergence of control, learning, data, and neuroscience

- data-driven LQG control
- sample complexity and asymptotic performance
- performance vs robustness tradeoffs for control/learning

Summary

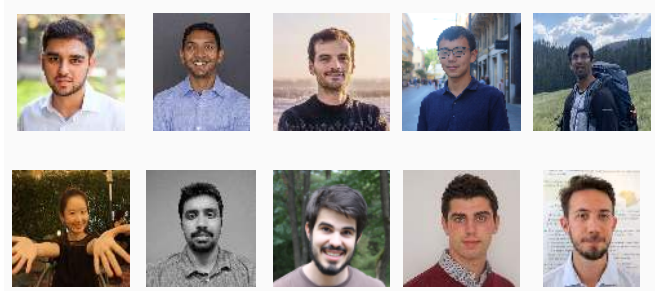
In this talk: autonomy as the convergence of control, learning, data, and neuroscience

- data-driven LQG control
- sample complexity and asymptotic performance
- performance vs robustness tradeoffs for control/learning

- brain-inspired decision making
- network control for brain analysis (anatomy & function)
- control-inspired methods for functional connectivity

Summary

My group



Collaborators