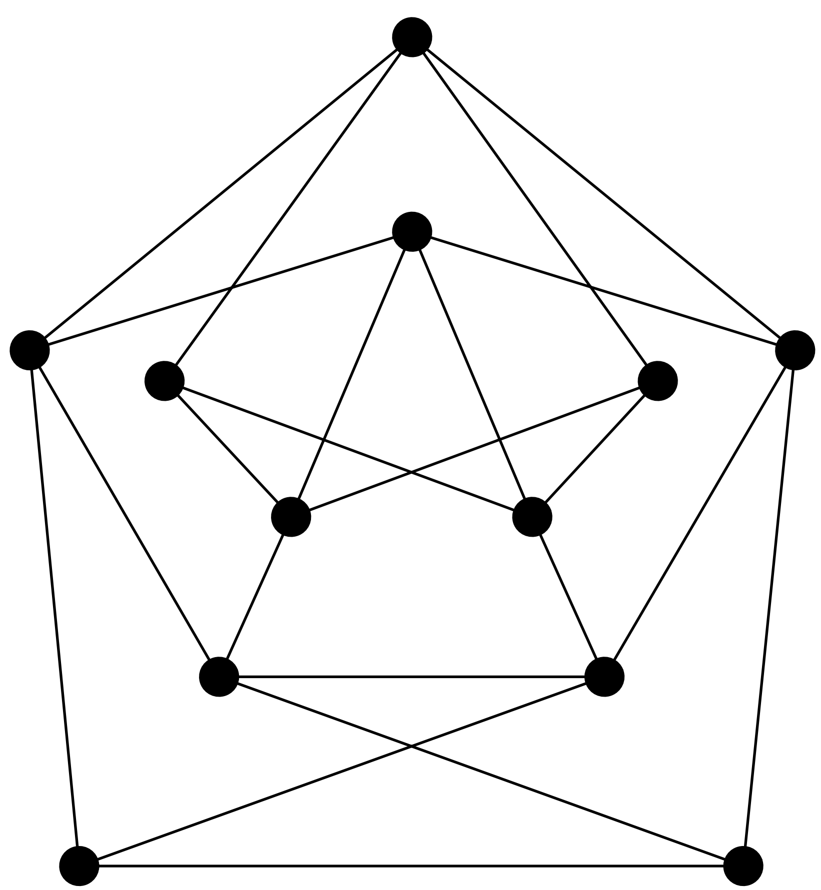
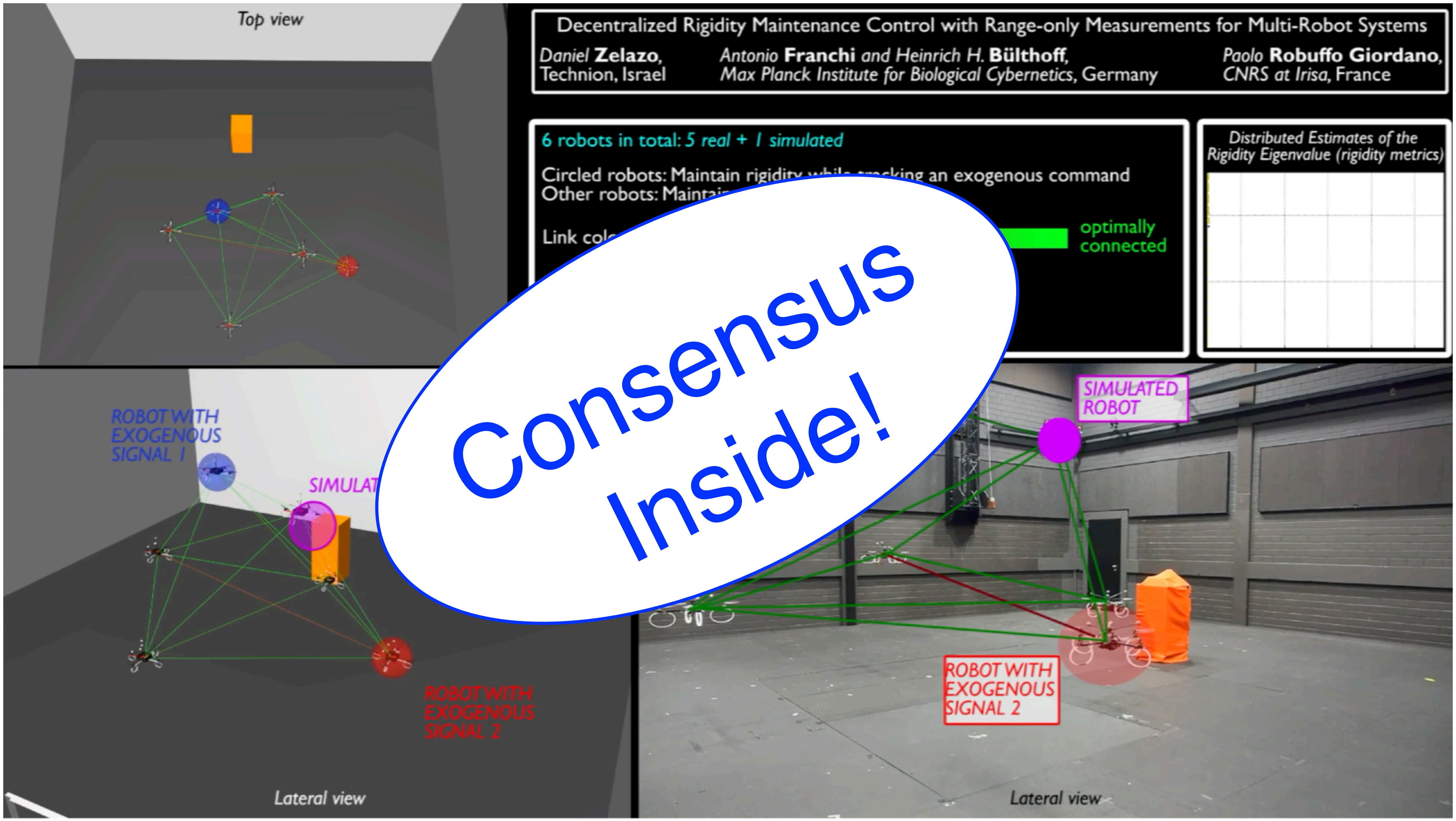
Cycles in Consensus Networks: Performance and Design

Daniel Zelazo
Faculty of Aerospace Engineering
Technion-Israel Institute of Technology

NCEPU September 2, 2013
Beijing, China



Networked Dynamic Systems



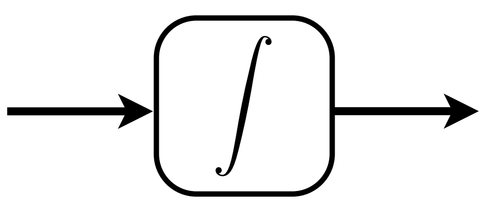
* this talk is not about robots…

The Consensus Protocol

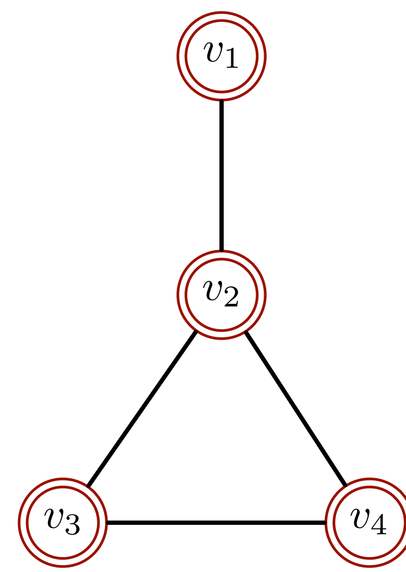
The consensus protocol is a *distributed and dynamic protocol* used for computing the average of a set of numbers.

Agent Dynamics

$$\dot{x}_i(t) = u_i(t)$$



Information Exchange Network



$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

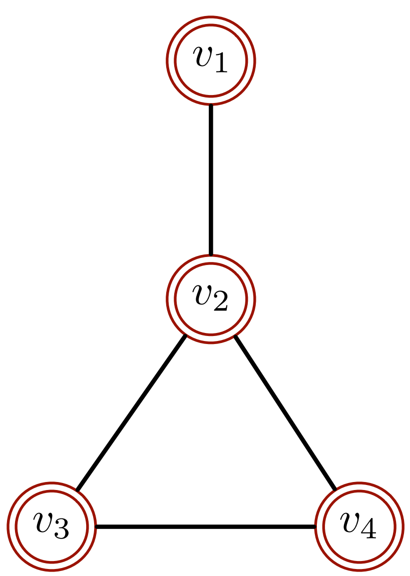
Incidence Matrix

$$E(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$$

$$E(\mathcal{G}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

The Consensus Protocol

The consensus protocol is a *distributed and dynamic protocol* used for computing the average of a set of numbers.



<u>Consensus Protocol</u>

$$u_i(t) = \sum_{i \sim j} (x_j(t) - x_i(t))$$

$$\dot{x}(t) = -L(\mathcal{G})x(t)$$

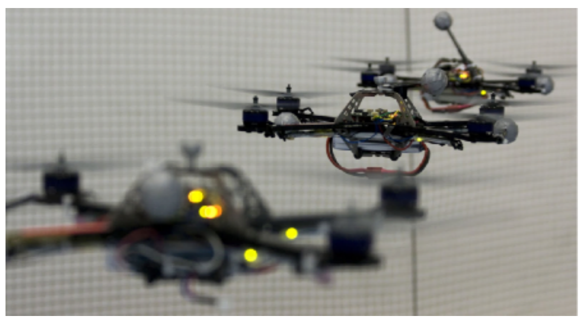
Laplacian Matrix

- $L(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$
- $L(\mathcal{G}) = E(\mathcal{G})E(\mathcal{G})^T$
- $L(\mathcal{G})\mathbf{1} = 0$

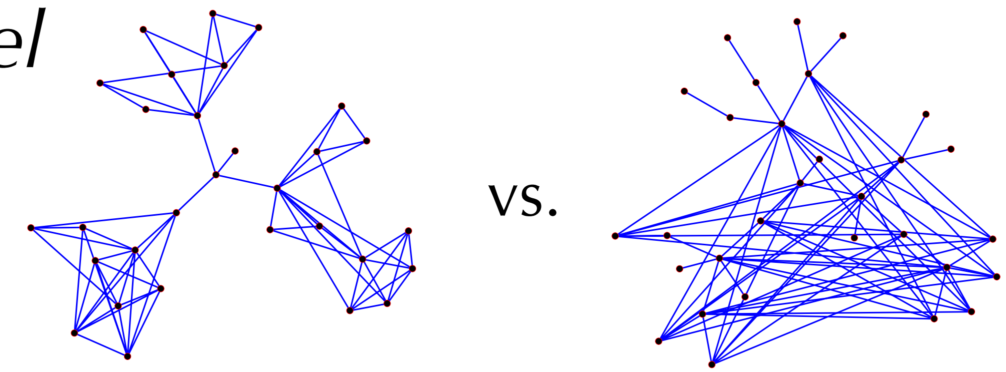
$$\lim_{t \to \infty} x(t) = \left(\frac{\mathbf{1}^T x(0)}{|\mathcal{V}|} \right) \mathbf{1}$$

Consensus-Seeking Networks

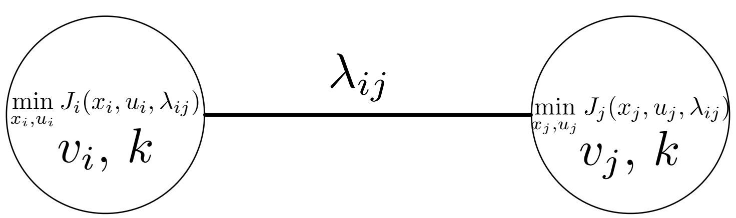
The consensus protocol is a *canonical model* for studying complex networked systems



formation control



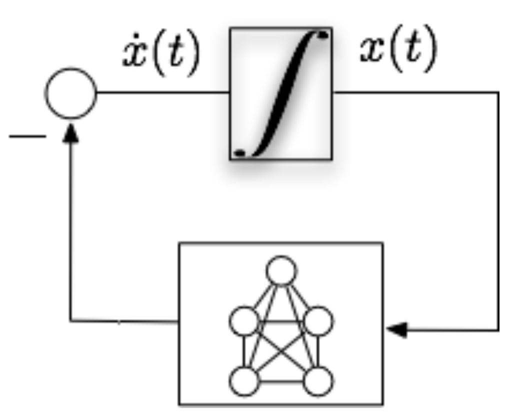
Are certain information structures more favorable to others?



distributed optimization

$\mathcal{H}_2$, $\mathcal{H}_\infty$, $\vdots$ $\quad \propto \quad$ cycle lengths, node degree, $\vdots$

Can notions of *dynamic system performance* be explained in terms of *properties of the graph?*



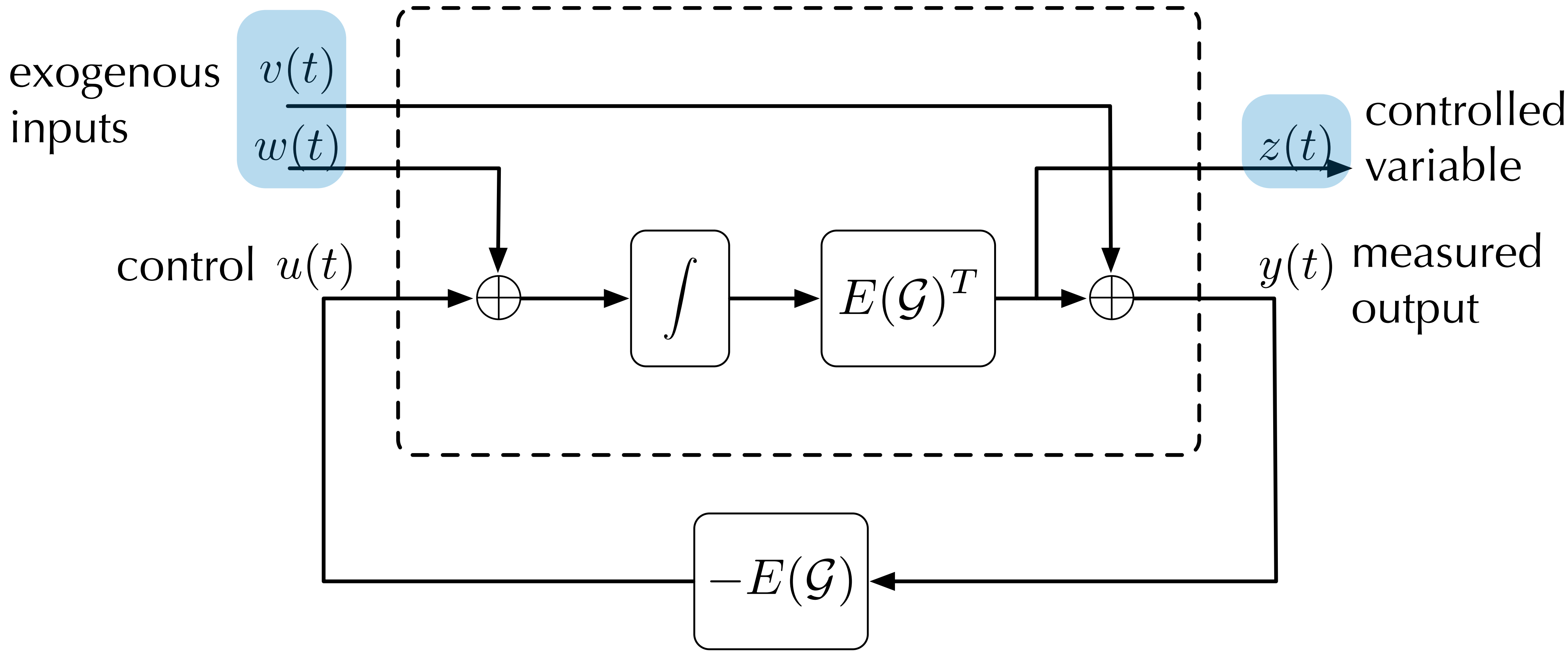
systems theory over graphs

$$\min_{\mathcal{G}} \|\Sigma(\mathcal{G})\|_p$$

How do we *synthesize* good information structures?

A Two-Port Consensus System

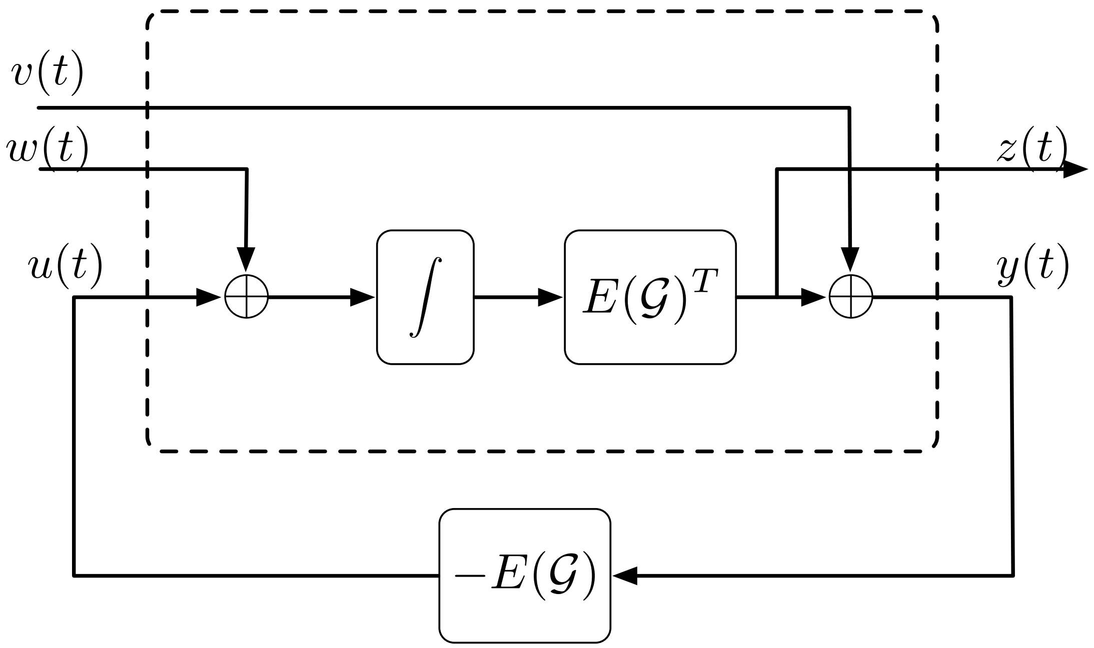
An ‘input-output’ consensus model



How do disturbances and noises affect the performance of the consensus protocol?

$\mathcal{H}_2$

A Two-Port Consensus System



$$\Sigma(\mathcal{G}) : \begin{cases} \dot{x}(t) = -L(\mathcal{G})x(t) + \begin{bmatrix} I & -E(\mathcal{G}) \end{bmatrix} \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \\ z(t) = E(\mathcal{G})^T x(t). \end{cases}$$

$$\overline{x}(t) = \frac{1}{n}\mathbf{1}^T x(t)$$

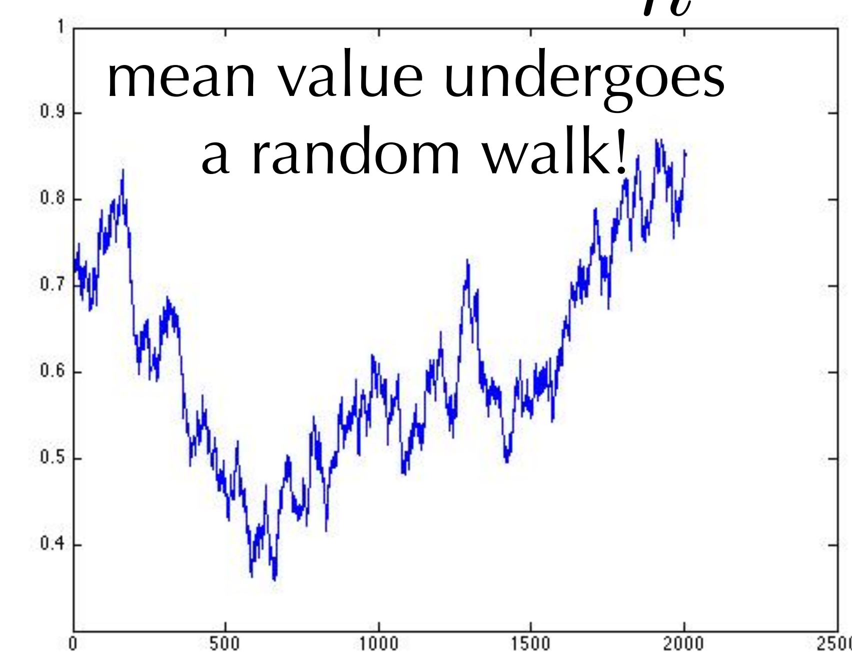
$$\dot{\overline{x}}(t) = \frac{1}{n}\mathbf{1}^T w(t)$$

$$\mathcal{E}(\overline{x}(t)^2) = \frac{\sigma_w^2}{n} t$$

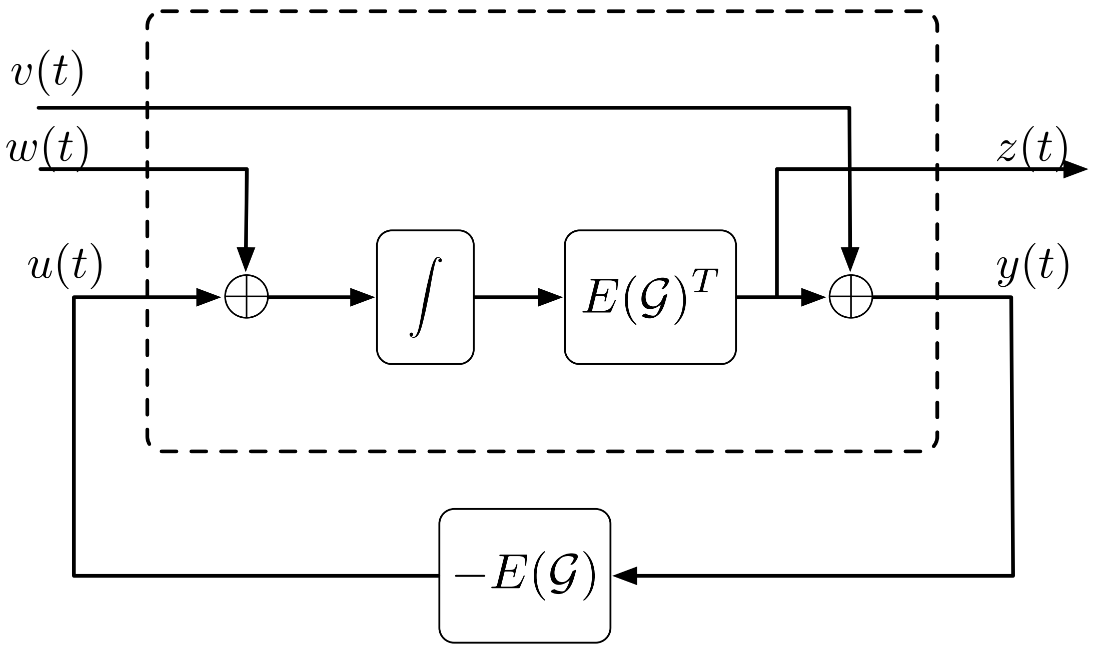
recall...

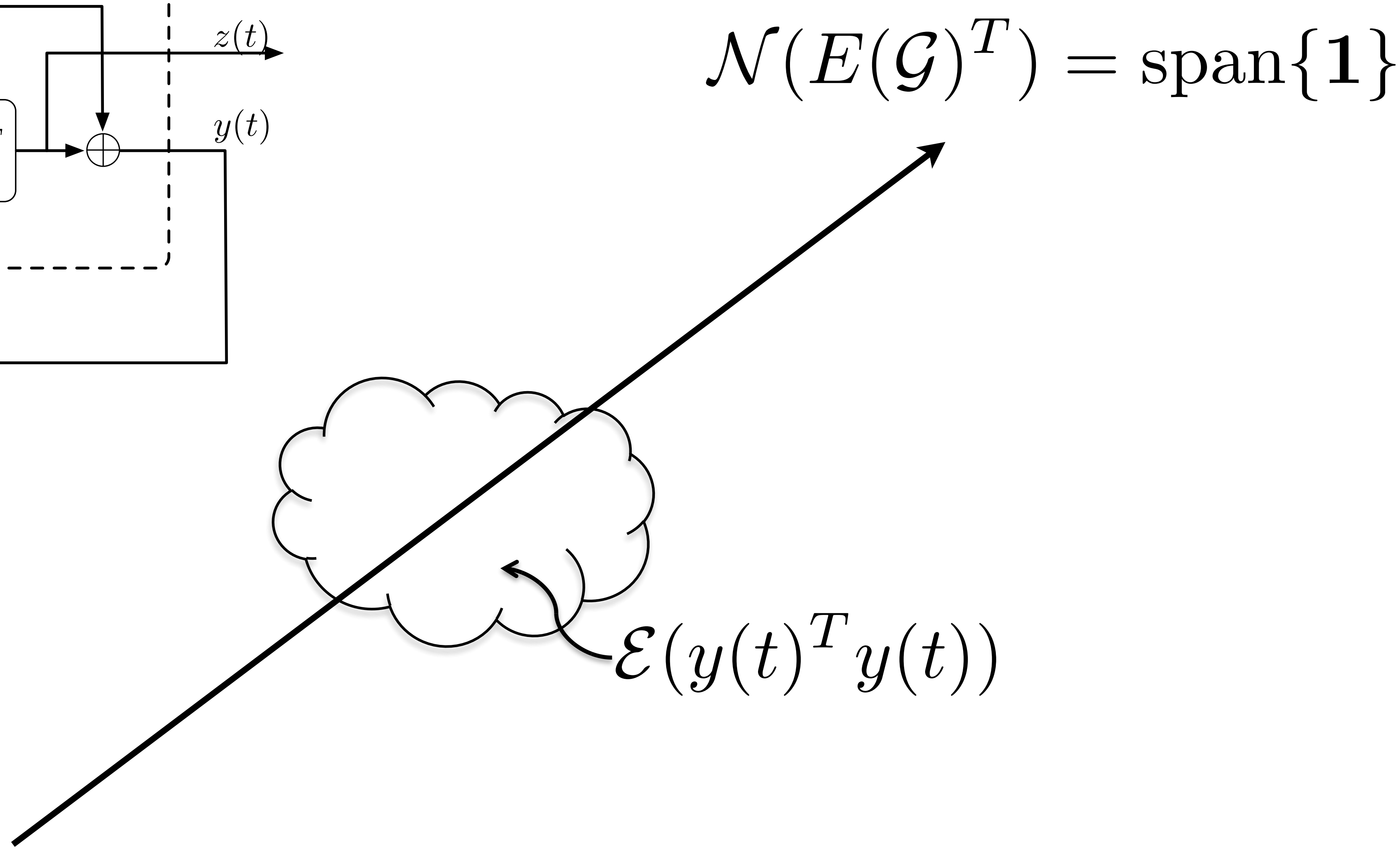
The $\mathcal{H}_2$ performance of a linear system characterizes how a WGN exogenous input propagates through the system and effects the variance of the output.

$$\|\Sigma(\mathcal{G})\|_2 \text{ is unbounded!}$$



Performance Interpretations





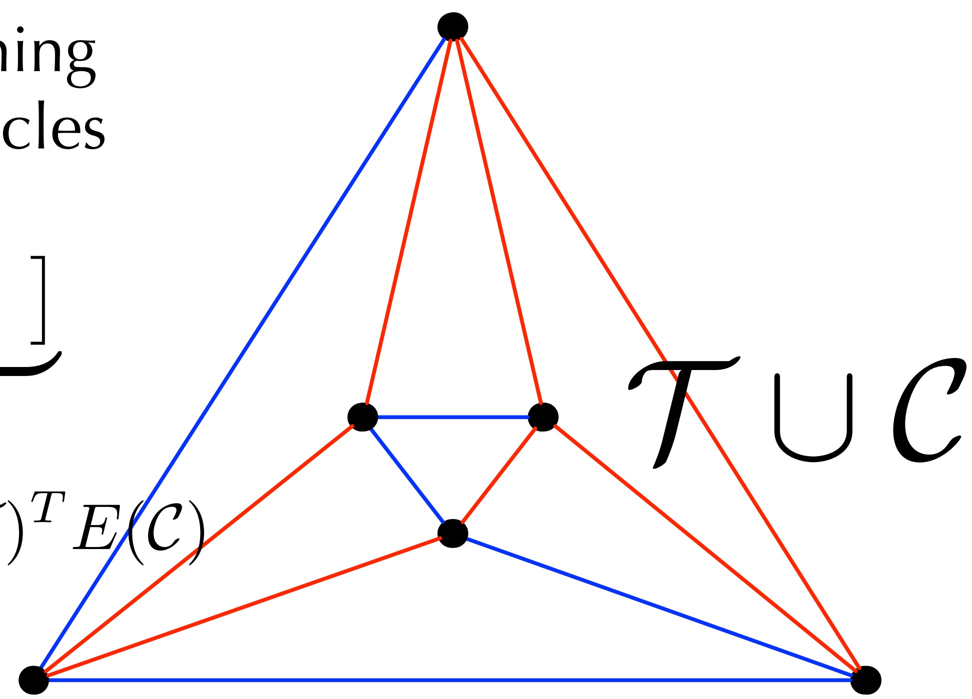
When driven by noise, it is meaningful to examine how noises effect the stead-state covariance of the *relative states*

Spanning Trees and Cycles

A graph as the union of a spanning tree and edges that complete cycles

$$E(\mathcal{G}) = E(\mathcal{T})\underbrace{\left[\begin{array}{cc} I & T_{(\mathcal{T},\mathcal{C})}\end{array}\right]}_{\mathcal{R}_{(\mathcal{T},\mathcal{C})}}$$

$$T_{(\mathcal{T},\mathcal{C})} = \left(E(\mathcal{T})^T E(\mathcal{T})\right)^{-1} E(\mathcal{T})^T E(\mathcal{C})$$



a spanning tree

remaining edges "complete cycles"

Edge Laplacian

$$L_e(\mathcal{G}) = E(\mathcal{G})^T E(\mathcal{G})$$

Essential Edge Laplacian

$$L_e(\mathcal{T})\mathcal{R}_{(\mathcal{T},\mathcal{C})}\mathcal{R}^T_{(\mathcal{T},\mathcal{C})}$$

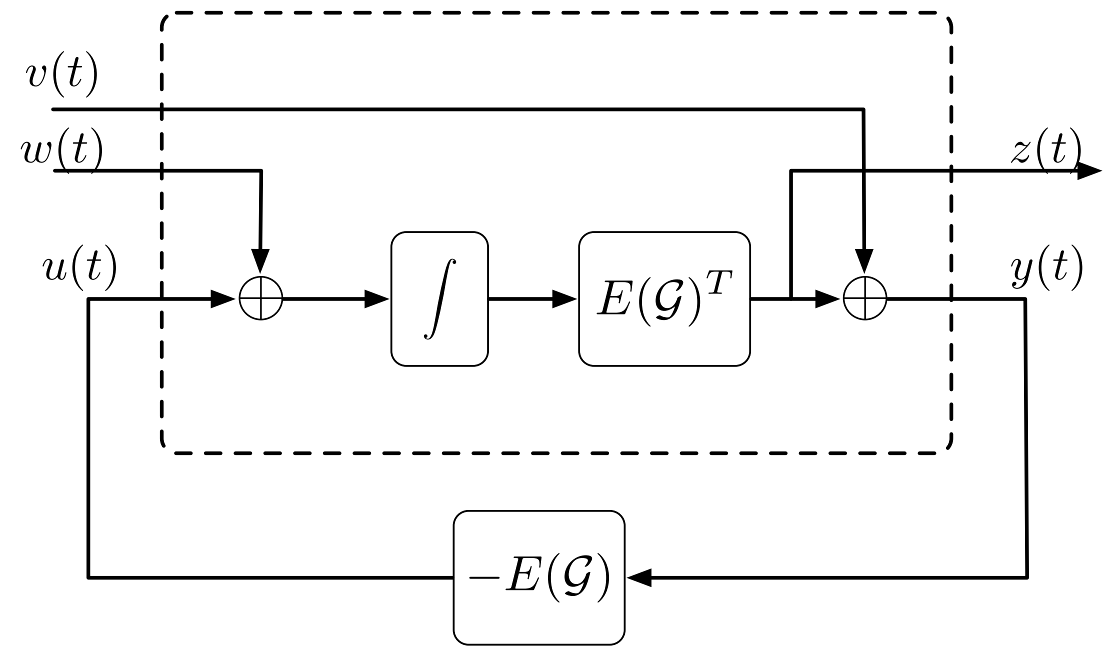
similarity between edge and graph Laplacians

$L(\mathcal{G}) \rightleftharpoons L_e(\mathcal{G})$

$\mathcal{R}_{(\mathcal{T},\mathcal{C})}$ rows form a basis for the *cut space* of the graph

The Edge Agreement Problem

$$\Sigma(\mathcal{G}) : \begin{cases} \dot{x}(t) &= -L(\mathcal{G})x(t) + \begin{bmatrix} I & -E(\mathcal{G}) \end{bmatrix} \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \\ z(t) &= E(\mathcal{G})^T x(t). \end{cases}$$

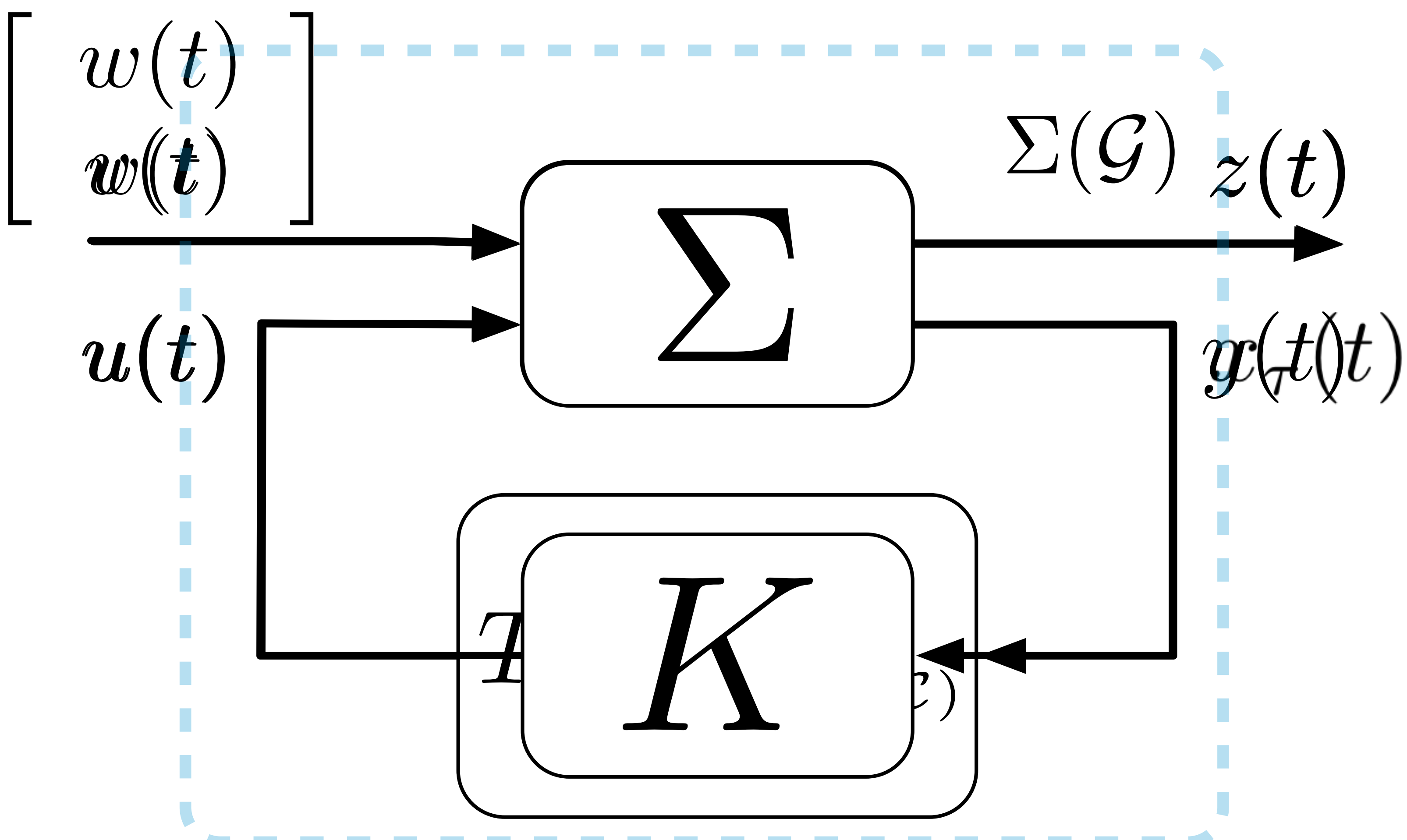


$$x_e(t) = \begin{bmatrix} E(\mathcal{T})^T \\ \frac{1}{n}\mathbf{1}^T \end{bmatrix} x(t)$$

$$\Sigma_e(\mathcal{G}) : \begin{cases} \dot{x}_\mathcal{T}(t) &= -L_e(\mathcal{T})R_{(\mathcal{T},\mathcal{C})}R^T_{(\mathcal{T},\mathcal{C})}x_\mathcal{T}(t) + \\ & \begin{bmatrix} E(\mathcal{T})^T & -L_e(\mathcal{T})R_{(\mathcal{T},\mathcal{C})} \end{bmatrix} \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \\ z(t) &= x_\mathcal{T}(t). \end{cases}$$

stable and minimal realization of consensus protocol

Cycles as Feedback



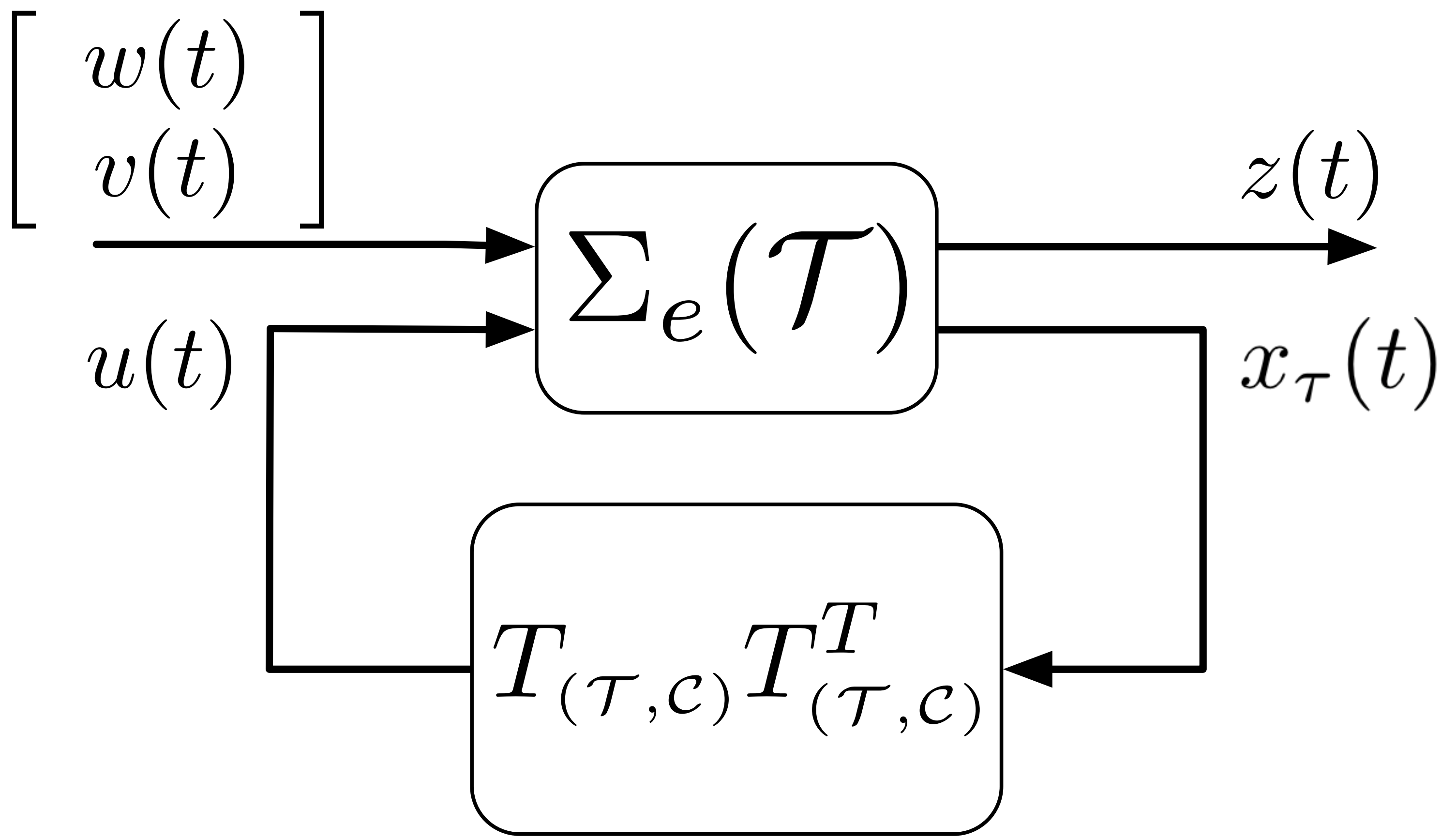
$$R_{(\mathcal{T},\mathcal{C})} = \left[\begin{array}{cc} I & T_{(\mathcal{T},\mathcal{C})} \end{array} \right]$$

$$E(\mathcal{T})T_{(\mathcal{T},\mathcal{C})} = E(\mathcal{C})$$

Design of consensus networks can be viewed as a *state-feedback problem*

$$L_e(\mathcal{T})R_{(\mathcal{T},\mathcal{C})}R^T_{(\mathcal{T},\mathcal{C})} = L_e(\mathcal{T}) + L_e(\mathcal{T})\underline{T_{(\mathcal{T},\mathcal{C})}T^T_{(\mathcal{T},\mathcal{C})}}$$

Cycles as Feedback



A synthesis problem

$$\min_{T_{(\mathcal{T},\mathcal{C})}\in\mathbb{R}^{|\mathcal{V}|\times k}} \|\Sigma_e(\mathcal{G})\|_2,$$

Performance of Consensus

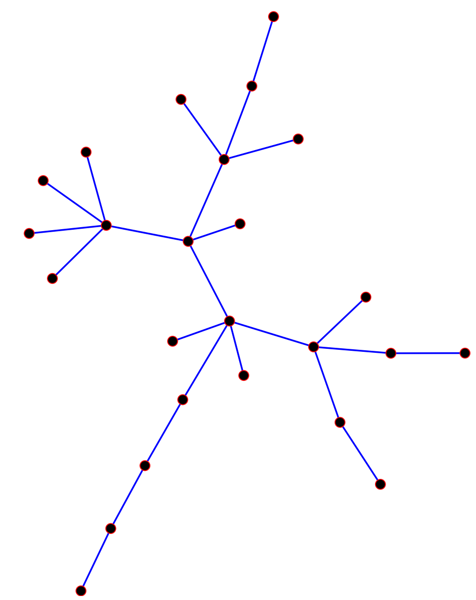
Theorem

$$\|\Sigma_e(\mathcal{G})\|_2^2 = \frac{1}{2}\mathbf{tr}\left[\left(R_{(\mathcal{T},\mathcal{C})}R_{(\mathcal{T},\mathcal{C})}^T\right)^{-1}\right] + (n-1)$$

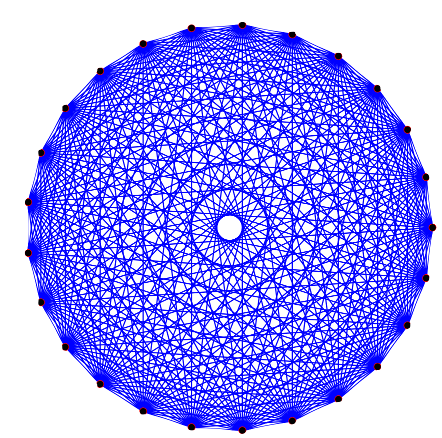
some immediate bounds...

$$\|\Sigma_e(\mathcal{G})\|_2^2 \leq \|\Sigma_e(\mathcal{T})\|_2^2 = \frac{3}{2}(n-1)$$

all trees are the same



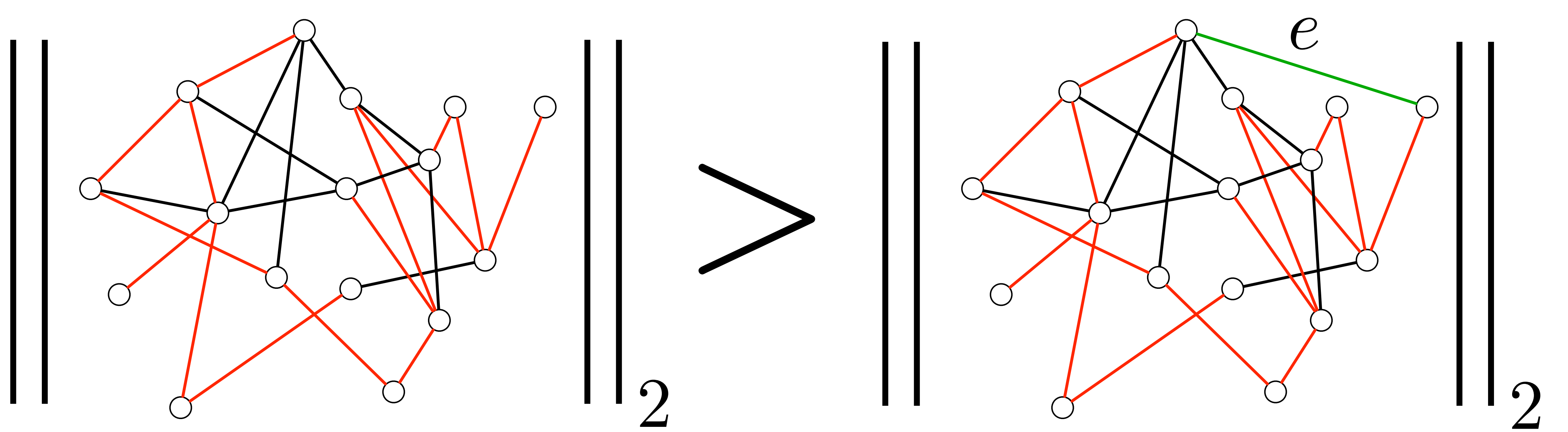
$$\|\Sigma_e(\mathcal{G})\|_2^2 \geq \|\Sigma_e(K_n)\|_2^2 = \frac{n^2-1}{n}$$



Performance and Cycles

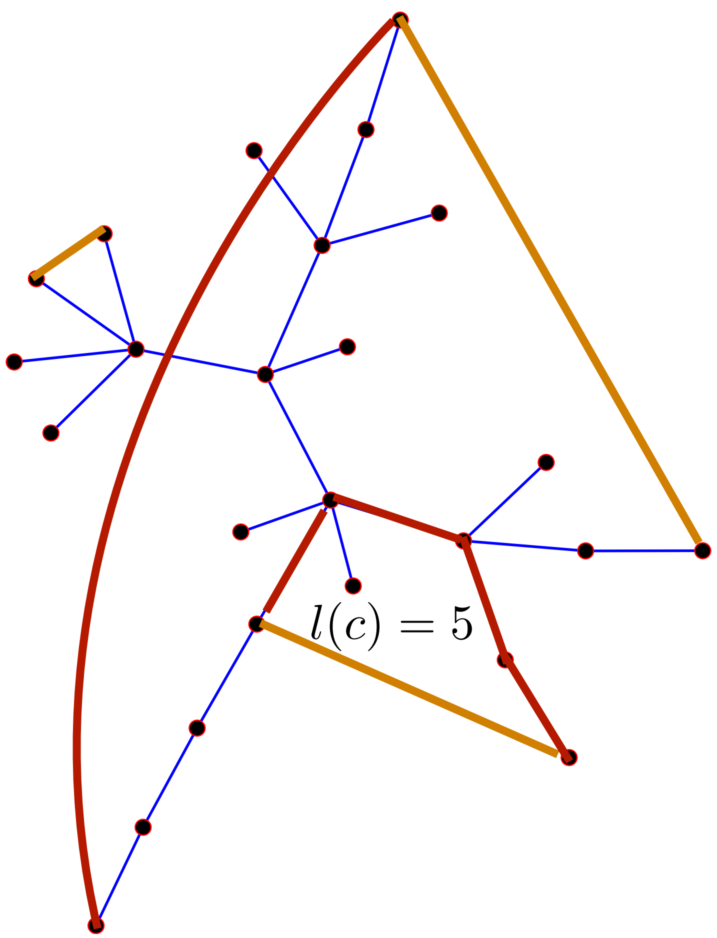
Theorem: Adding cycles always improves the performance.

$$\|\Sigma_e(\mathcal{G} \cup e)\|_2^2 \quad = \quad \|\Sigma_e(\mathcal{G})\|_2^2 - \frac{\left(R_{(\mathcal{T},\mathcal{C})} R^T_{(\mathcal{T},\mathcal{C})}\right)^{-1} cc^T \left(R_{(\mathcal{T},\mathcal{C})} R^T_{(\mathcal{T},\mathcal{C})}\right)^{-1}}{2(1 + c^T \left(R_{(\mathcal{T},\mathcal{C})} R^T_{(\mathcal{T},\mathcal{C})}\right)^{-1} c)}$$



Performance and Cycles

Is there a *combinatorial* feature
that affects the performance?



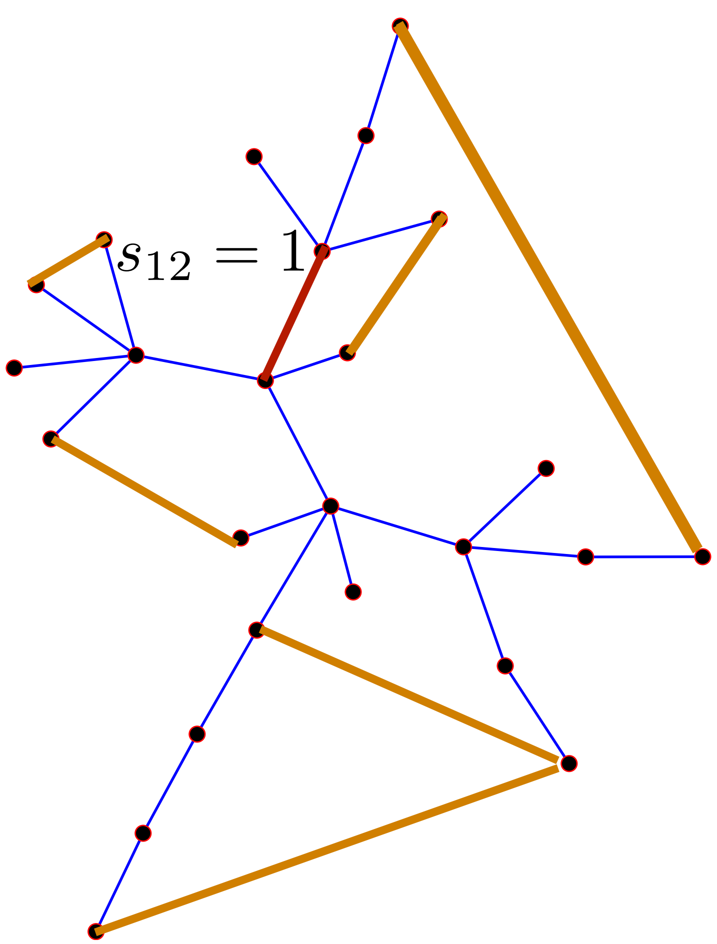
Corollary

$$\|\Sigma_e(\mathcal{T} \cup e)\|_2^2 = \|\Sigma_e(\mathcal{T})\|_2^2 - \frac{1}{2}(1 - l(c)^{-1})$$

long cycles are "better"

Performance and Cycles

Is there a *combinatorial* feature that affects the performance?

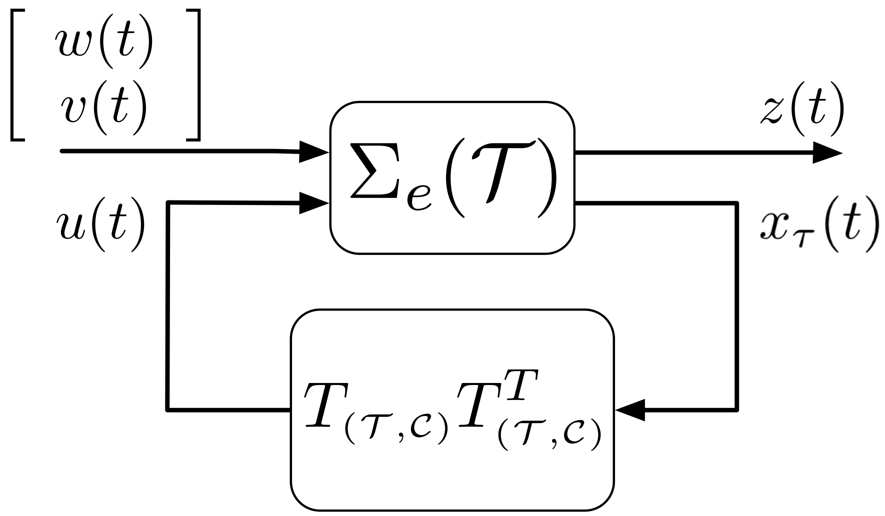


Corollary

$$\|\Sigma_e(\mathcal{T} \cup \{e_1, e_2\})\|_2^2 = \|\Sigma_e(\mathcal{T})\|_2^2 - \left(1 - \frac{l(c_1) + l(c_2)}{2(l(c_1)l(c_2) - s_{12}^2)}\right)$$

"edge disjoint" cycles are better

Design of Cycles



$$\min_{T_{(\mathcal{T},\mathcal{C})}\in\mathbb{R}^{|\mathcal{V}|\times k}} \|\Sigma_e(\mathcal{G})\|_2,$$

Given a spanning tree, add **k** edges that maximize the performance improvement

a mixed-integer SDP

$$\begin{aligned} \min_{M,w_i} \quad & \mathbf{trace}\,[M] \\ \text{s.t.} \quad & \begin{bmatrix} M & I \\ I & I + T_{(\mathcal{T},\overline{\mathcal{T}})} W T_{(\mathcal{T},\overline{\mathcal{T}})} \end{bmatrix} \geq 0 \\ & \sum_i w_i = k,\ w_i \in \{0,1\} \end{aligned}$$

Design of Cycles

a mixed-integer SDP

$$\begin{aligned} \min_{M, w_i} \quad & \textbf{trace}\,[M] \\ \text{s.t.} \quad & \begin{bmatrix} M & I \\ I & I + T_{(\mathcal{T},\overline{\mathcal{T}})} W T_{(\mathcal{T},\overline{\mathcal{T}})} \end{bmatrix} \geq 0 \\ & \sum_i w_i = k, \ \cancel{w_i \in \{0,1\}} \end{aligned}$$

$$w_i \in [0,1]$$

relaxation to *weighted* edges "misses the point"

$$\begin{aligned} \min_{M, w_i} \quad & \textbf{trace}[M] + \text{card}(w) \\ \text{s.t.} \quad & \begin{bmatrix} M & I \\ I & I + T_{(\mathcal{T},\overline{\mathcal{T}})} W T_{(\mathcal{T},\overline{\mathcal{T}})} \end{bmatrix} \geq 0 \\ & \sum_i w_i = k, \ \ w_i \in [0,1] \end{aligned}$$

attempt to minimize "# of non-zero elements"

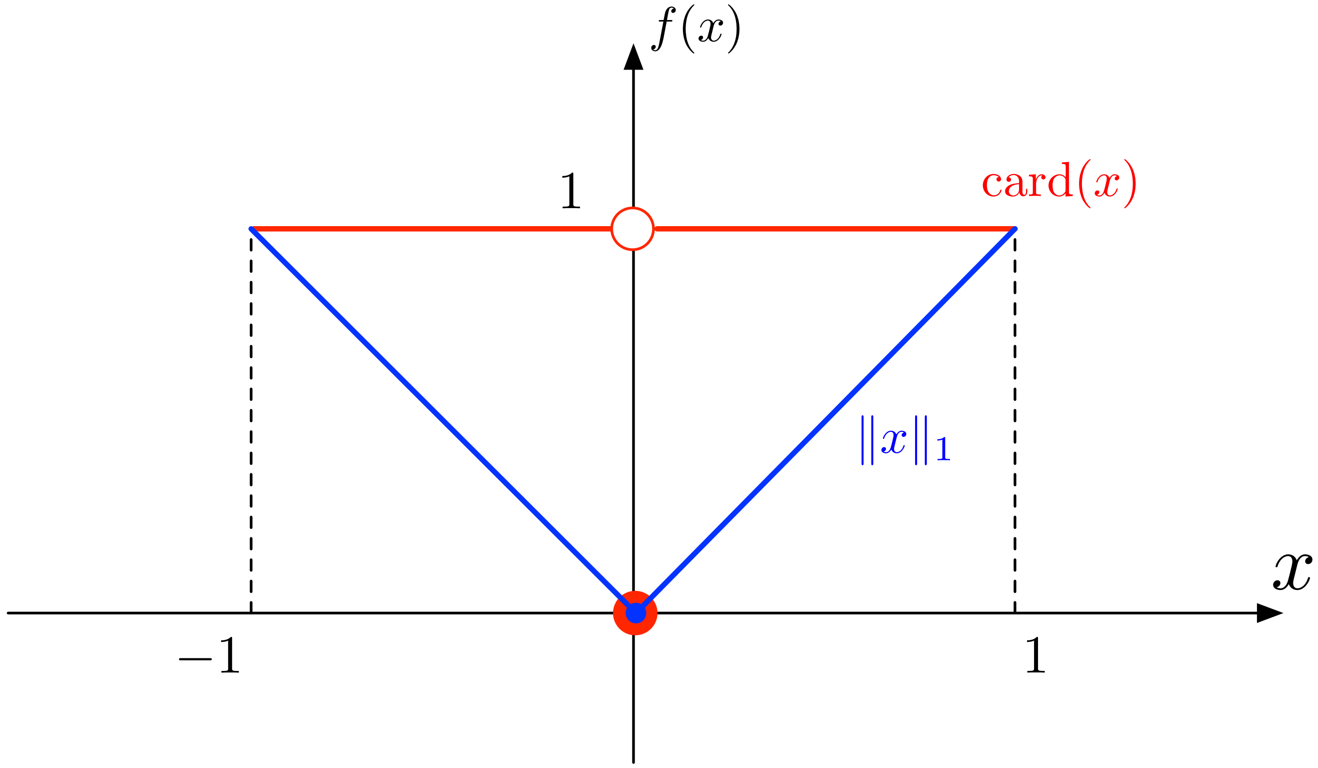
not a convex relaxation!

Convex Envelope of Cardinality

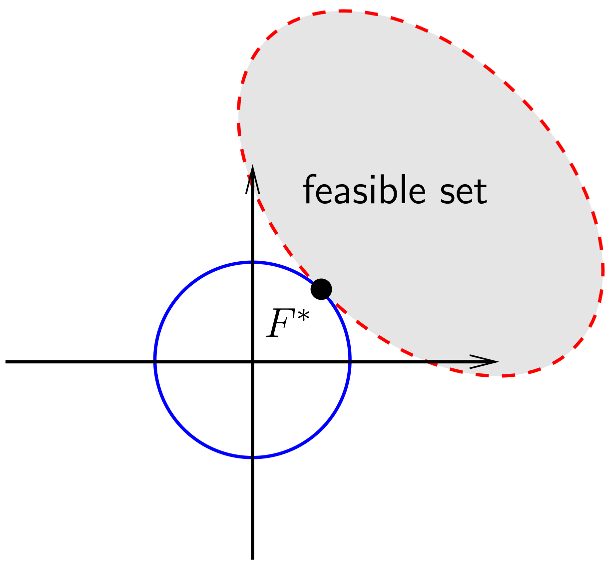
Definition. *The* convex envelope, f^{env}, *of a function* f *on a set* C *is the (point-wise) largest convex function that is an under estimator of* f *on* C.

example

$\|x\|_1 = \sum_i |x_i|$ is convex envelope of $\text{card}(x)$.

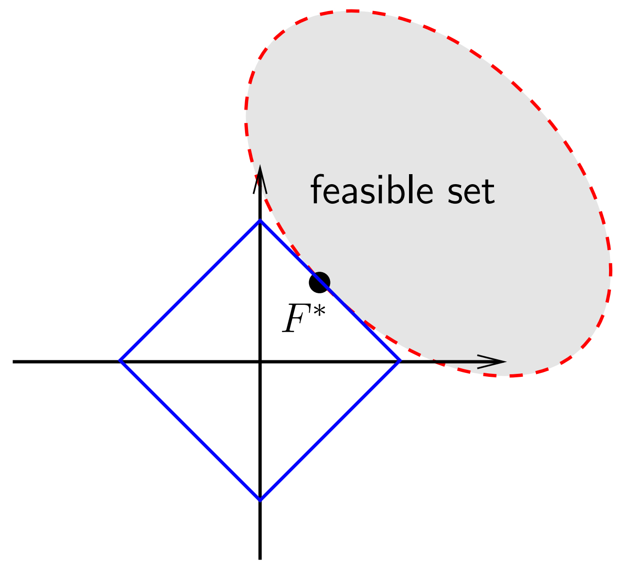


Sparsity Promoting Optimization



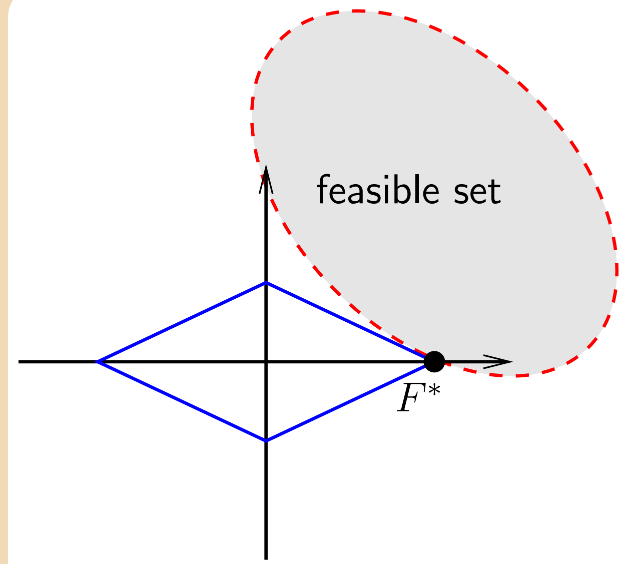
$$\min_{x\in\mathcal{X}} \|x\|_2$$

✳convex optimization
✳not sparse



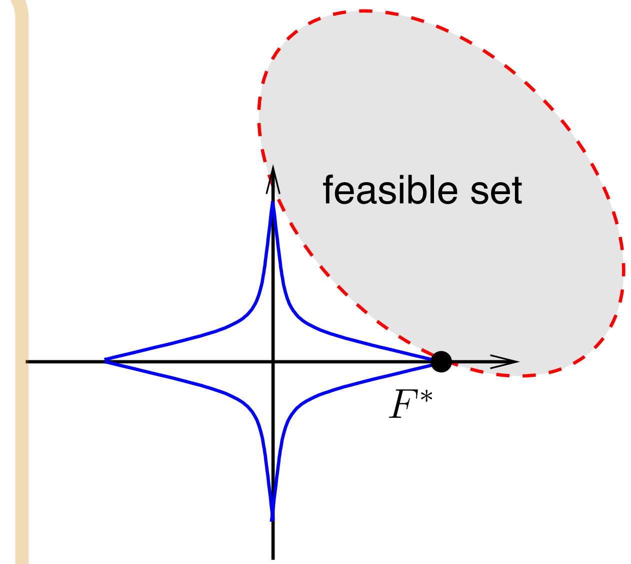
$$\min_{x\in\mathcal{X}} \|x\|_1$$

✳convex optimization
✳sparse for LP



$$\min_{x\in\mathcal{X}} \sum_i m_i|x_i|$$

✳convex optimization
✳sparse for SDP

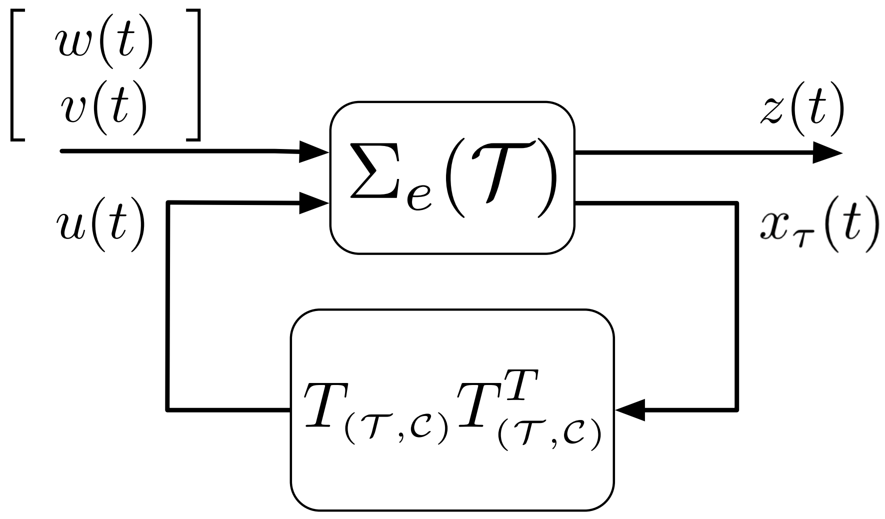


$$\min_{x\in\mathcal{X}} \|x\|_p$$

✳non-convex
✳sparse

re-weighted *l*-1 minimization algorithm
[Candes 2008]

Design of Cycles



$$\min_{T_{(\mathcal{T},\mathcal{C})}\in\mathbb{R}^{|\mathcal{V}|\times k}} \|\Sigma_e(\mathcal{G})\|_2,$$

Given a spanning tree, add **k** edges that maximize the performance improvement

$$\min_{M,w_i} \quad \alpha\,\mathbf{trace}\,[M] + (1-\alpha)\sum_i m_i w_i$$

$$\text{s.t.} \quad \begin{bmatrix} M & I \\ I & I + T_{(\mathcal{T},\overline{\mathcal{T}})} W T_{(\mathcal{T},\overline{\mathcal{T}})} \end{bmatrix} \geq 0$$

$$\sum_i w_i = k, \quad 0 \leq w_i \leq 1.$$

Design of Cycles

Re-weighted *l*-1 minimization algorithm

① set counter $h = 0$

choose initial weights for each edge $m_i^{(0)}$ ← combinatorial insights used here!

② solve convex program - obtain optimal weights $w_i^{(h)}$

$$\min_{M, w_i} \quad \alpha \mathbf{trace}\left[M\right] + (1-\alpha)\sum_i m_i^{(h)} w_i$$

$$\text{s.t.} \quad \begin{bmatrix} M & I \\ I & I + T_{(\mathcal{T},\overline{\mathcal{T}})} W T_{(\mathcal{T},\overline{\mathcal{T}})} \end{bmatrix} \geq 0$$

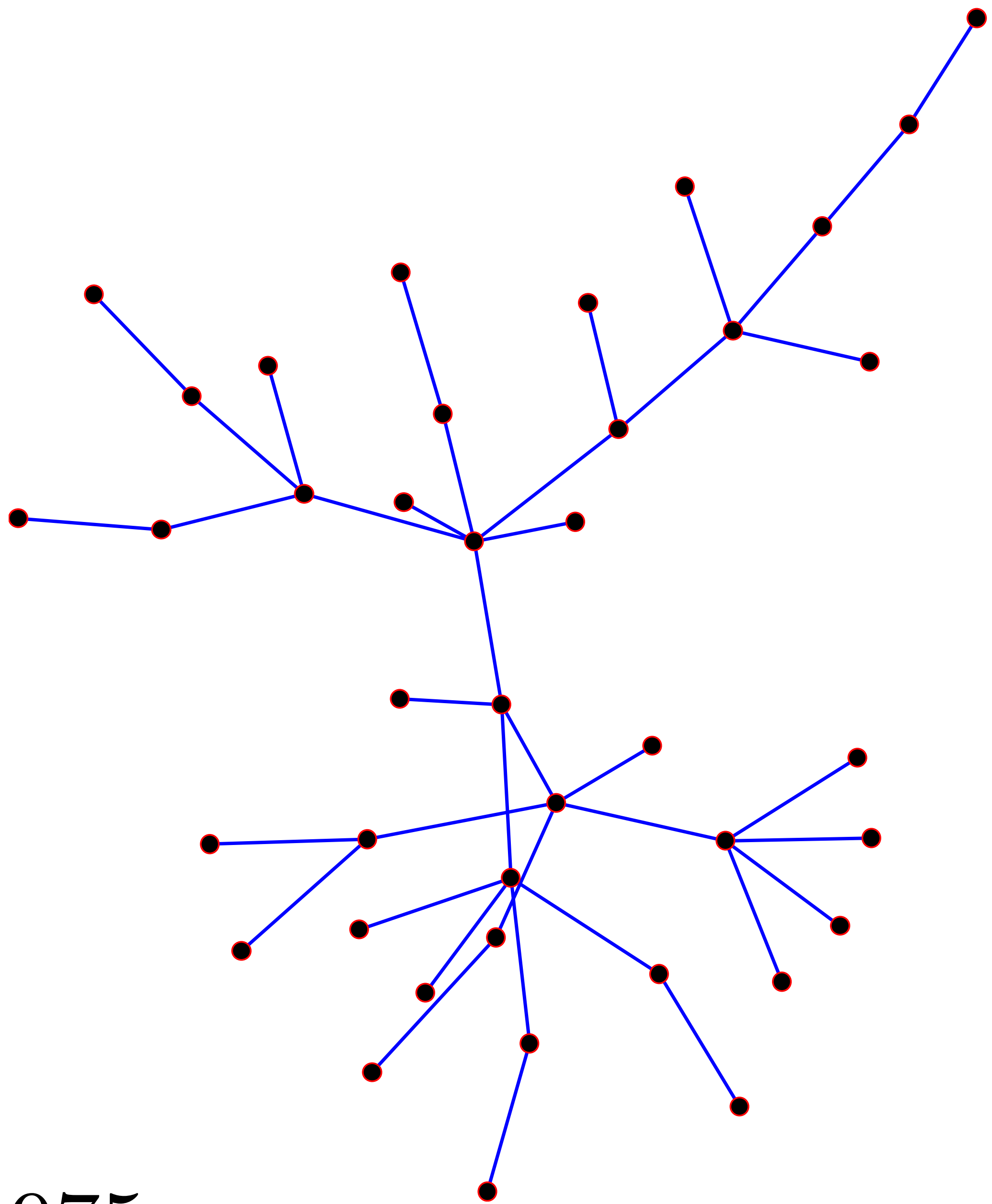
$$\sum_i w_i = k, \quad 0 \leq w_i \leq 1.$$

③ update weights $m_i^{(h+1)} = (w_i^{(h)} + \nu)^{-1}$

④ terminate on convergence, or
increment counter and go to step 2

[Candes 2008]

Simulation Examples



spanning tree
30 nodes

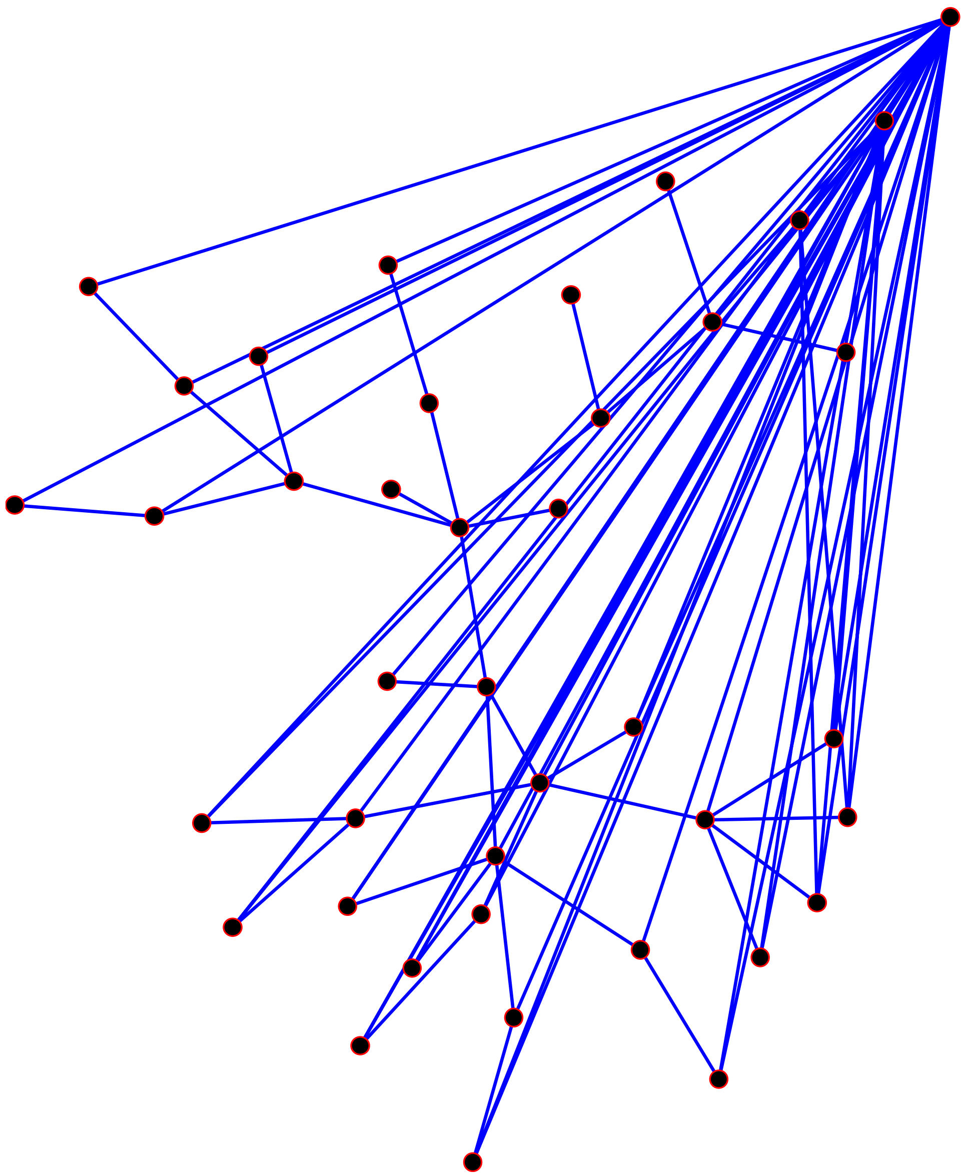
741 candidate edges

add 40 new edges

$$\|\Sigma(\mathcal{T})\|_2^2 = 58.5$$

$$\|\Sigma(K_n)\|_2^2 = 39.975$$

Simulation Examples



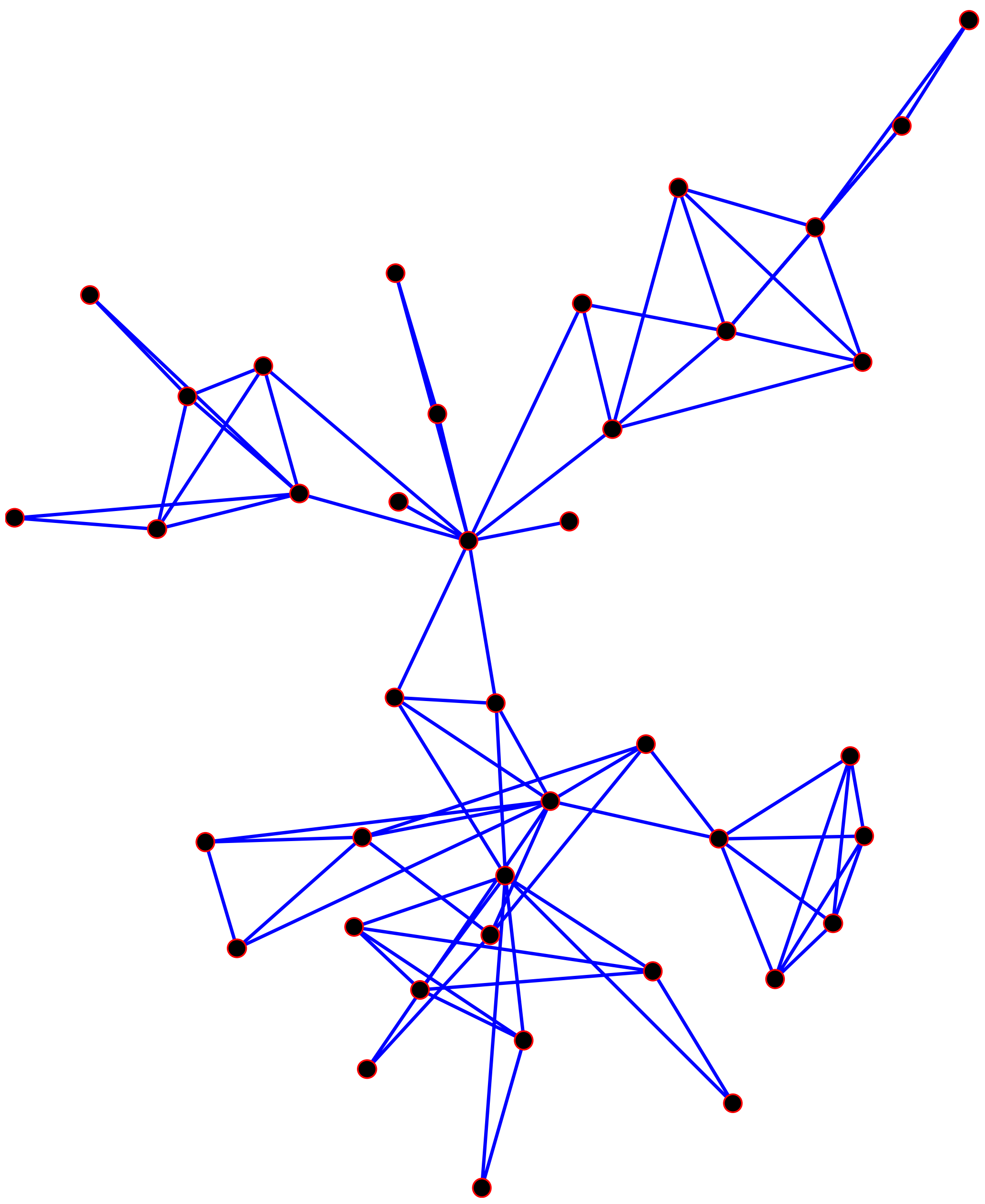
weights can be used to promote certain graph properties

"long cycle weights"

$$m_i = \mathbf{diam}(\mathcal{G}) - \|c_i\|_1 + 1$$

$$\|\Sigma(\mathcal{G})\|_2^2 = 50.233$$

Simulation Examples



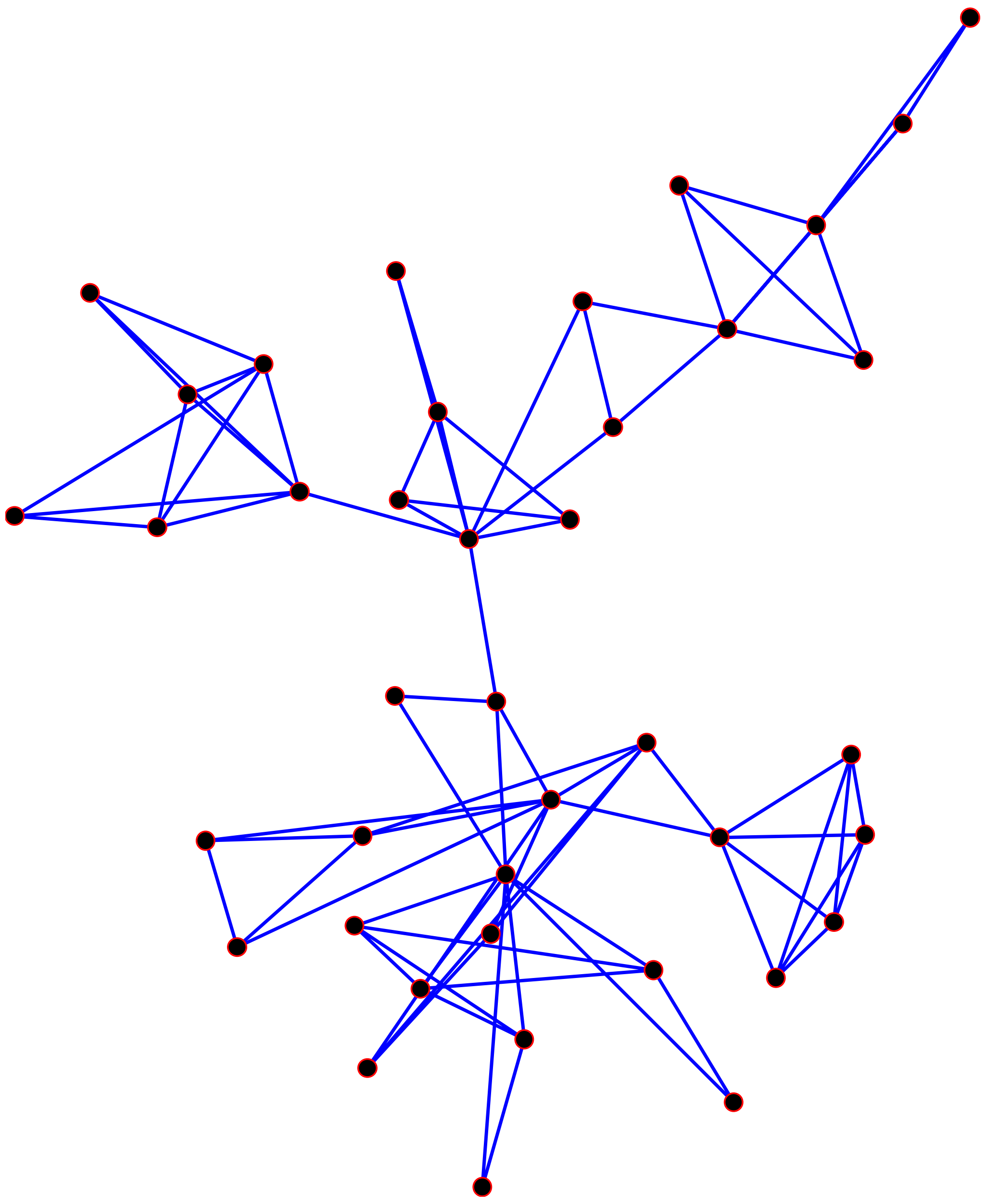
weights can be used to promote certain graph properties

"short cycle weights"

$$m_i = \|c_i\|_1$$

$$\|\Sigma(\mathcal{G})\|_2^2 = 48.704$$

Simulation Examples



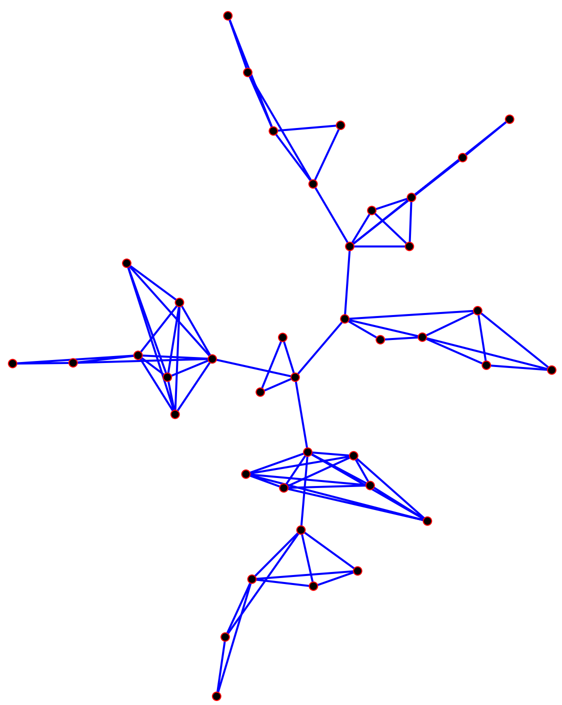
weights can be used to promote certain graph properties

"cycle correlation weights"

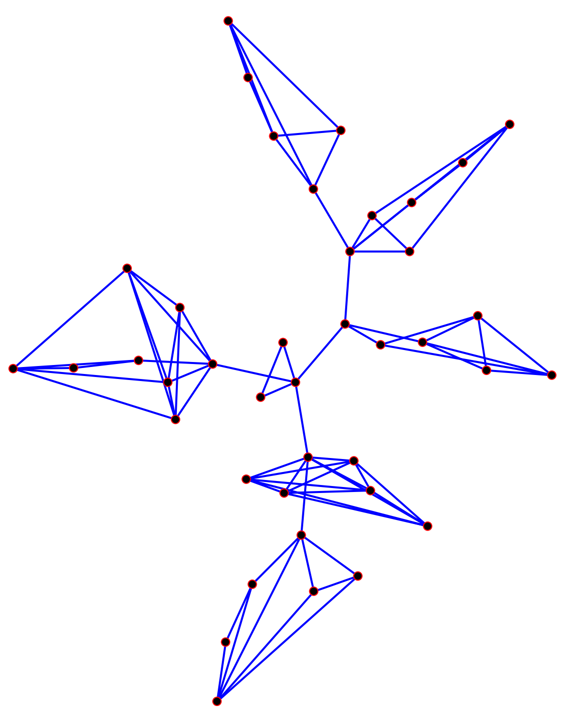
$$m_i = \frac{1}{|\mathcal{E}_c|} \sum_{j \neq i} \left| \left[T_{(\mathcal{T},\mathcal{C})} T_{(\mathcal{T},\mathcal{C})}^T \right]_{ij} \right|$$

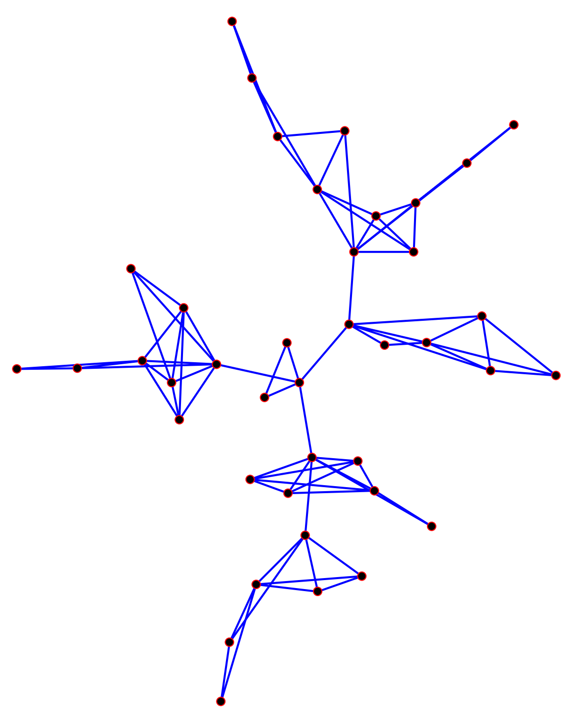
$$\|\Sigma(\mathcal{G})\|_2^2 = 48.939$$

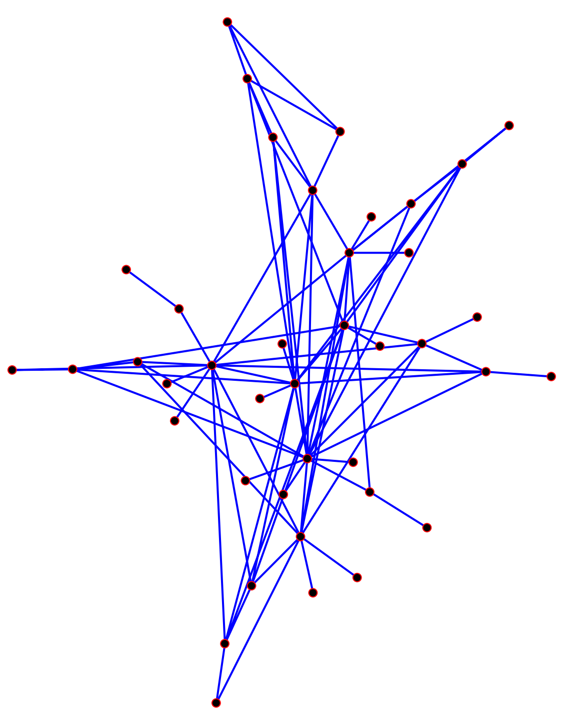
Simulation Examples



weights can be used to promote certain graph properties



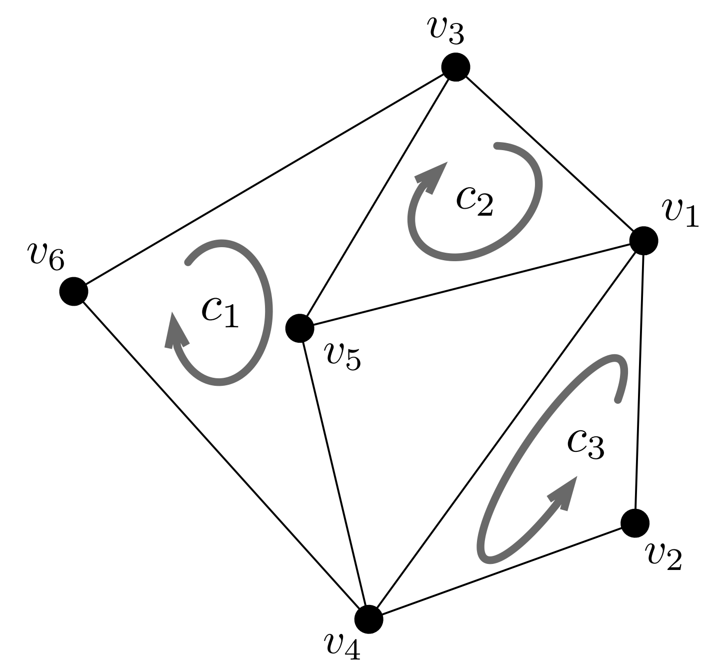




Concluding Remarks

role of cycles in consensus networks

- ✳ internal feedback
- ✳ performance



a tractable design procedure

- ✳ l1 optimization
- ✳ design of multi-agent systems

future works

- ✳ additional performance metrics
- ✳ push to large scale

*"Performance and design of cycles in consensus networks"
Systems & Control Letters 62(1) : 85-96, 2013.

*"Edge Agreement: Graph-theoretic Performance Bounds and Passivity Analysis"
IEEE Transactions on Automatic Control 56(3) : 554-555, 2011.

Concluding Remarks

謝謝

Prof. Dr. -Ing. Frank Allgöwer

Simone Schuler

Questions?