NCS Lecture 10
Cooperative Control

Richard M. Murray

19 March 2008

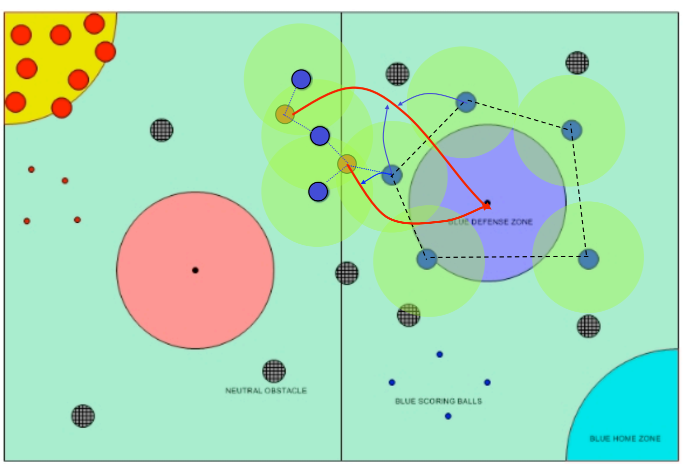
Goals:

- Information flow, stability, formations (Fax, Olfati-Saber, Jadbabaie, Jin)
- Distributed optimization (Dunbar)
- Distributed sensing (Gupta, Chung)

Reading:

- R. M. Murray, "Recent Research in Cooperative Control of Multi-Vehicle Systems", *J. Guidance, Control and Dynamics*, 2007.

RoboFlag Subproblems



1. Formation control
- Maintain positions to guard defense zone
- (Other spatio-temporal planning problems)

2. Distributed estimation
- Fuse sensor data to determine opponent location

3. Distributed assignment
- Assign individuals to tag incoming vehicles

Goal: develop systematic techniques for solving subproblems
- Distributed receding horizon control
- Packet-based, distributed estimation
- Verifiable protocols for consensus and control

Cooperative Control Framework

Agent dynamics

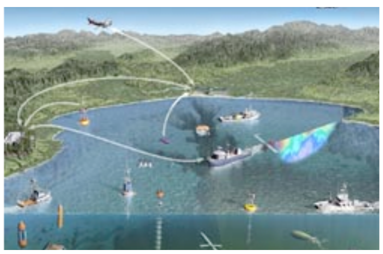
$$\dot{x}^i = f^i(x^i, u^i) \qquad x^i \in \mathbb{R}^n, u^i \in \mathbb{R}^m$$
$$\dot{y}^i = h^i(x^i) \qquad y^i \in \mathrm{SE}(3),$$

Vehicle "role"

- $\alpha \in \mathcal{A}$ encodes internal state + relationship to current task
- Transition $\alpha' = r(x, \alpha)$

Communications graph $\mathcal{G}$

- Encodes the system information flow
- Neighbor set $\mathcal{N}^i(x, \alpha)$



Task

- Encode as finite horizon optimal control

$$J = \int_0^T L(x, \alpha, u)\,dt + V(x(T), \alpha(T)),$$

- Assume task is *coupled*

Strategy

- Control action for individual agents

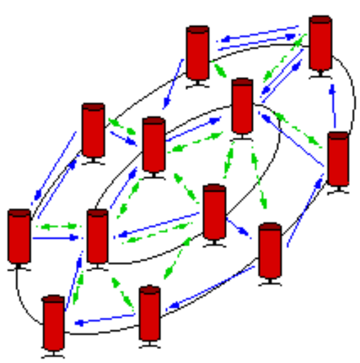
$$u^i = \gamma(x, \alpha) \qquad \{g^i_j(x, \alpha) : r^i_j(x, \alpha)\}$$
$$\alpha^{i\,\prime} = \begin{cases} r^i_j(x, \alpha) & g(x, \alpha) = \text{true} \\ \text{unchanged} & \text{otherwise.} \end{cases}$$

Decentralized strategy

$$u^i(x, \alpha) \quad = u^i(x^i, \alpha^i, x^{-i}, \alpha^{-i})$$
$$x^{-i} = \{x^{j_1}, \ldots, x^{j_{m_i}}\}$$
$$j_k \in \mathcal{N}^i \qquad m_i = |\mathcal{N}^i|$$

- Similar structure for role update

Information Flow in Vehicle Formations



Example: satellite formation

- Blue links represent *sensed* information
- Green links represent *communicated* information

Sensed information

- Local sensors can see some subset of nearby vehicles
- Assume small time delays, pos'n/vel info only

Communicated information

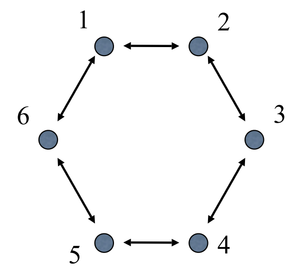
- Point to point communications (routing OK)
- Assume limited bandwidth, some time delay
- Advantage: can send more complex information

Topological features

- Information flow (sensed or communicated) represents a directed graph
- Cycles in graph ⇒ information feedback loops

Question: How does topological structure of information flow affect stability of the overall formation?

Sample Problem: Formation Stabilization



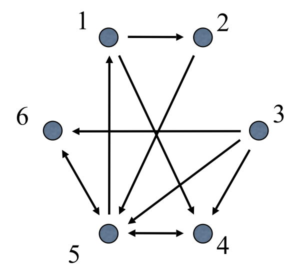
Goal: maintain position relative to neighbors

- "Neighbors" defined by graph
- Assume only sensed data for now
- Assume identical vehicle dynamics, identical controllers?

Example: hexagon formation

- Maintain fixed relative spacing between left and right neighbors

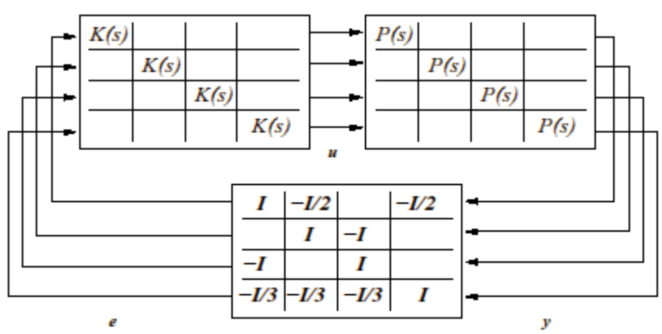
$$e_i = \sum_{j \in N_i} \underbrace{w_j}_{\text{weighting factor}} (\underbrace{y_i - y_j}_{\text{relative position}} - \underbrace{h_{ij}}_{\text{offset}})$$

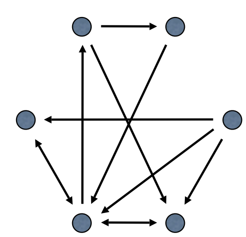


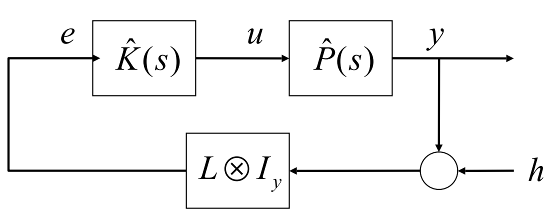
Can extend to more sophisticated "formations"

- Include more complex spatio-temporal constraints

Mathematical Framework







Analyze stability of closed loop

- Interconnection matrix, *L*, is the *weighted Laplacian* of the graph
- Stability of closed loop related to eigenstructure of the Laplacian

Fax and M
IEEE TAC 2004

Stability Condition

Agent dynamics

$$\dot{x}^i = Ax^i + Bu^i$$
$$y^i = Cx^i$$

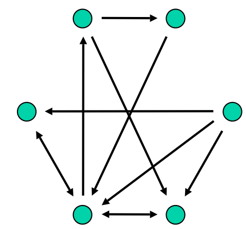
Control law

$$\dot{\xi}^i = F\xi^i + Gz^i$$
$$u^i = H\xi^i + Kz^i$$

Weighted error

$$z^i = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}(y^i - y^j)$$

- Agents have identical, linear dynamics
- Control law is dynamic compensator based on sum of relative errors on neighbors
- Can also allow feedback on internal state (fold into A)



$$\begin{bmatrix} 1 & -\frac{1}{2} & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 0 & 1 & -1 & 0 \\ -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Theorem *Let L be the weighted Laplacian of the communications graph $\mathcal{G}$. The closed loop system is (neutrally) stable iff the systems*

$$\dot{x}^i = Ax^i + Bu^i \qquad \dot{\xi}^i = F\xi^i + Gz^i$$
$$z^i = \lambda_j Cx^i \qquad u^i = H\xi^i + Kz^i$$

are stable for each eigenvalue λ_i of L.

Remarks

- Stability is based on check of *n* decoupled systems
- λ_i plays the role of a "loop gain": describes how your output affects your input

Sketch of Stability Proof

$$\dot{x}^i = Ax^i + Bu^i$$
$$z = L\widehat{C}x$$

$$\dot{\xi}^i = F\xi^i + Gz^i$$
$$u^i = H\xi^i + Kz^i$$

Notation

- $\widehat{A} = I_N \otimes A$: block diagonal matrix with A as elements
- $A_{(n)} = A \otimes I_n$: replace elemnts of A with $a_{ij}I_n$
- For $X \in \mathbb{R}^{r\times s}$ and $Y \in \mathbb{R}^{N\times N}$, $\widehat{X}Y_{(s)} = \widehat{Y}X_{(r)}$

Let T be a Schur transformation for L, so that $U = T^{-1}LT$ is upper triangular. Transform the (stacked) process states as $\tilde{x} = T_{(n)}x$ and the (stacked) controller states as $\tilde{\xi} = T_{(n)}\xi$. The resulting dynamics become

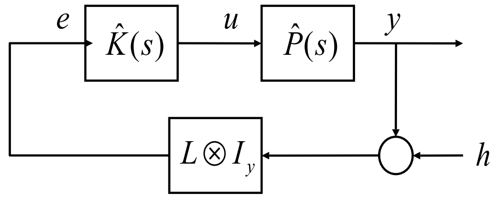
$$\frac{d}{dt}\begin{bmatrix}\tilde{x}\\ \tilde{\xi}\end{bmatrix} = \begin{bmatrix}\widehat{A} + \widehat{B}\widehat{K}\widehat{C}U_{(n)} & \widehat{B}\widehat{H}\\ \widehat{G}\widehat{C}U_{(n)} & F\end{bmatrix}\begin{bmatrix}\tilde{x}\\ \tilde{\xi}\end{bmatrix}.$$

This system is upper triangular, and so stability is determined by the elements on the (block) diagonal:

$$\frac{d}{dt}\begin{bmatrix}\tilde{x}_j\\ \tilde{\xi}_j\end{bmatrix} = \begin{bmatrix}A + BKC\lambda_j & BH\\ GC\lambda_j & F\end{bmatrix}\begin{bmatrix}\tilde{x}\\ \tilde{\xi}\end{bmatrix}.$$

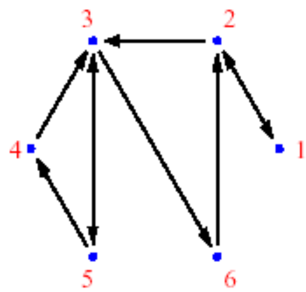
This is equivalent to coupling the process and controller with a gain λ_i.

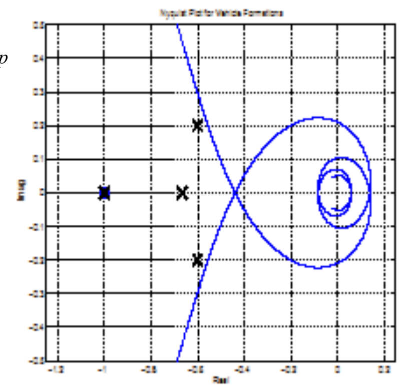
Frequency Domain Interpretation



Theorem *The closed loop system is (neutrally) stable iff the Nyquist plot of the open loop system does not encircle* $-1/\lambda_i(L)$*, where* $\lambda_i(L)$ *are the nonzero eigenvalues of* L*.*

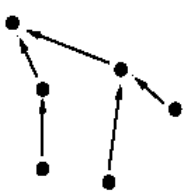
Example $P(s) = \dfrac{e^{-s\tau}}{s^2}$ $\quad K(s) = K_d s + K_p$



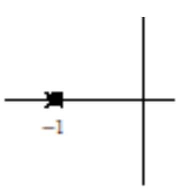


Spectra of Laplacians

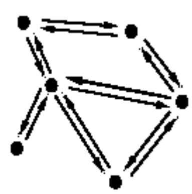
Unidirectional tree



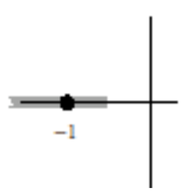
$\lambda = 0,1$



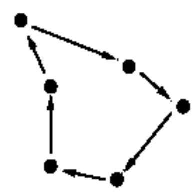
Undirected graph



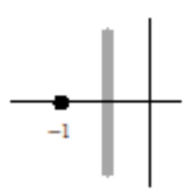
$\lambda \in [0,2]$



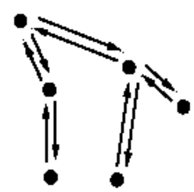
Cycle



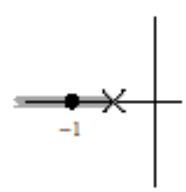
$\lambda_i = 1 - e^{2\pi(i-1)j/N}$



Periodic graph



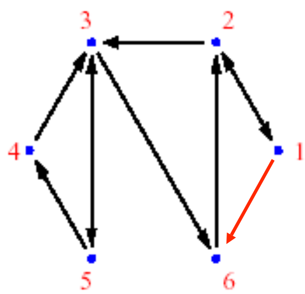
$\lambda_1 = 0, \lambda_N = 2$

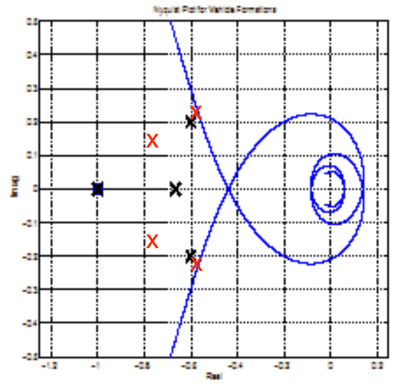


Example Revisited

Example

$$P(s) = \frac{e^{-s\tau}}{s^2} \qquad K(s) = K_d s + K$$





- Adding link increases the number of three cycles (leads to “resonances”)
- Change in control law required to avoid instability
- Q: Increasing amount of information available *decreases* stability (??)
- A: Control law cannot ignore the information ⇒ add’l feedback inserted

Fax and Murray
IFAC 02

Improving Performance through Communication

Baseline: stability only

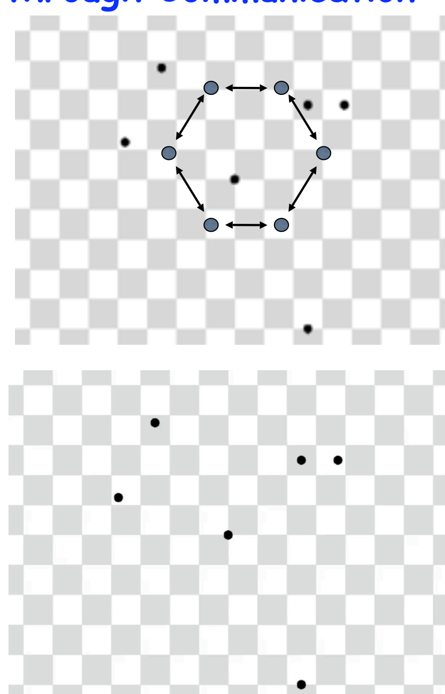
- Poor performance due to interconnection

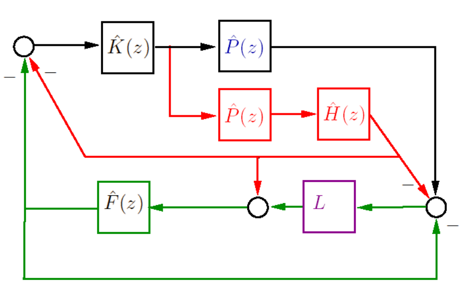
Method #1: tune information flow filter

- Low pass filter to damp response
- Improves performance somewhat

Method #2: consensus + feedforward

- Agree on center of formation, then move
- Compensate for motion of vehicles by adjusting information flow

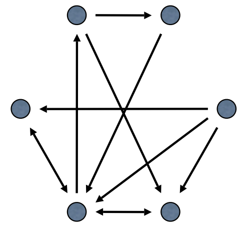


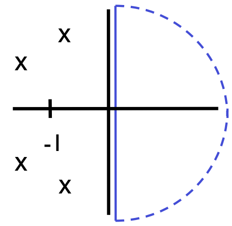


Special Case: (Asymptotic) Consensus

$$\dot{x}_i = \sum w_{ij}(x_j - x_i)$$

$$\dot{x} = -Lx$$





Consensus: agreement between agents using information flow graph

- Can prove asymptotic convergence to single value if graph is connected
- If w_{ij} = 1/(in-degree) + graph is *balanced* (same in-degree for all nodes) ⇒ all agents converge to average of initial condition

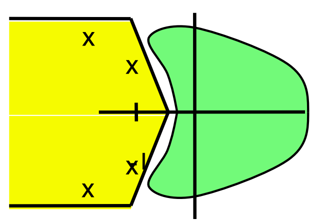
Extensions (Jadbabaie/Morse, Moreau, Olfati-Saber, Xiao, Chandy/Charpentier, ...)

- Switching (packet loss, dropped links, etc),time delays, plant uncertainty
- Nearest neighbor graphs, small world networks, optimal weights
- Nonlinear: potential fields, passive systems, gradient systems
- Distributed Kalman filtering, distributed optimization
- Self-similar algorithms for operation with varying connectedness

Open Problems: Design of Information Flow (graph)

How does graph topology affect location of eigenvalues of L?

- Would like to separate effects of topology from agent dynamics

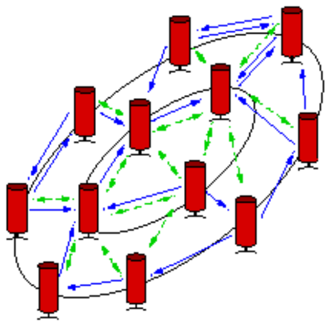


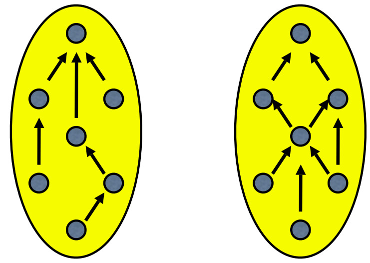
- Possible approach: exploit form of characteristic polynomial

$$\lambda(s) = \text{“}s^n + \Big(\sum w_i\Big)s^{n-1} + \Big(\sum_{\text{2 cycles}} w_i w_j\Big)s^{n-2} + \Big(\sum_{\text{3 cycles}} w_i w_j w_k\Big)s^{n-3} + \cdots + \Big(\sum_{\text{N cycles}} w_i \ldots w_N\Big)\text{”}$$

Performance

Look at motion between selected vehicles





G_1 - Control G_2 - Performance

Theorem

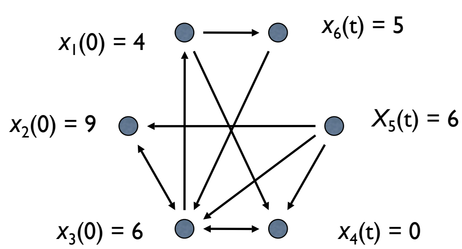
For a leader-follower formation with double-graph strategy, If $\|(1-\alpha)\cdot H(s)\|_\infty = M < 1$, *then*

$$\frac{\|E\|_\infty}{\|V\|_\infty} \le \left(\frac{1-M^n}{1-M} + \frac{\rho(\Theta_2)M^n}{1-\rho(\Theta_2)M}\right)\cdot \|N(s)\|_\infty$$

where $\rho(\Theta_2)$ *is the spectral radius of* Θ_2 *and* $N(s) = \frac{e_1(s)}{v_1(s)}$.

Robustness

What happens if a single node "locks up"



- Single node can change entire value of the consensus
- Desired effect for "robust" behavior: $\Delta x_l = \delta/N$

Different types of robustness (Gupta, Langbort & M)

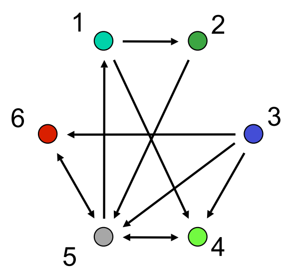
- Type I - node stops communicating (stopping failure)
- Type II - node communicates constant value
- Type III - node computes incorrect function (Byzantine failure)

Related ideas: delay margin for multi-hop models (Jin and M)

- Improve consensus rate through multi-hop, but create sensitivity to communcations delay

Cremean and M
CDC 2003

Stability of (Heterogeneous) Nonlinear Systems



Model as affine nonlinear system

$$\dot{x}_i = f_i(x_i) + B_i u$$

- allow agents to have different dynamics

Stability conditions

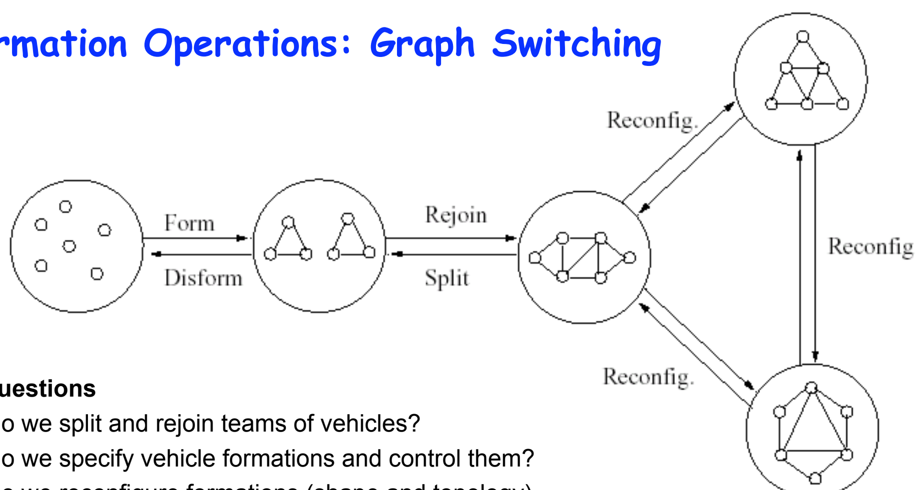
- Asy stable if

$$x_i^T f_i(x_i) < 0 \quad \text{for all } i$$

$$L \otimes BC \geq 0$$

- Fairly weak set of conditions: tells us when interconnection doesn't *destabilize* system

Formation Operations: Graph Switching



Control questions

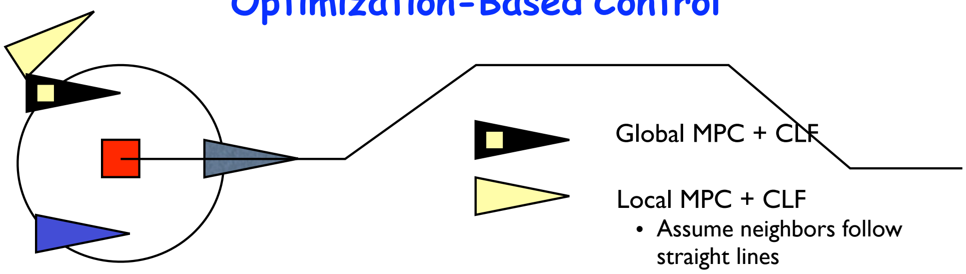
- How do we split and rejoin teams of vehicles?
- How do we specify vehicle formations and control them?
- How do we reconfigure formations (shape and topology)

Consensus-based approach using balanced graphs

- If each subgraph is balanced, disagreement vector provides common Lyapunov fcn
- By separately keeping track of the flow in and out of nodes, can preserve center of mass of of subgraphs after a split manuever

Dunbar and M
Automatica, 2006

Optimization-Based Control

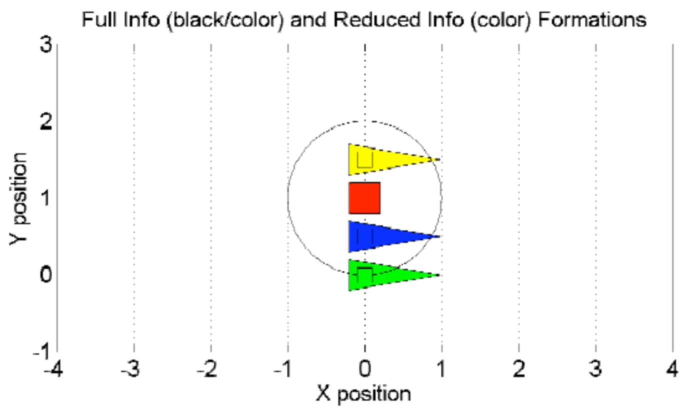


Task:

- Maintain equal spacing of vehicles around circle
- Follow desired trajectory for center of mass

Parameters:

- Horizon: 2 sec
- Update: 0.5 sec



Main Idea: Assume Plan for Neighbors

Individual optimization:

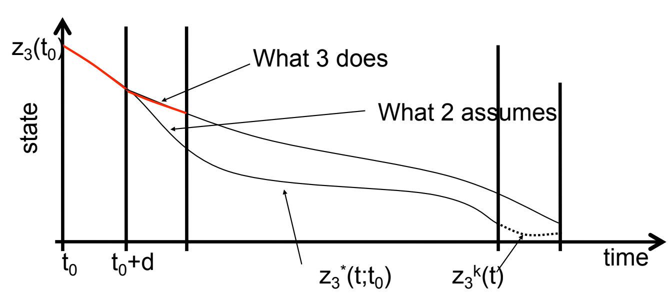
$$\min_{u_3(\cdot)} \left\{ \int_{t_k}^{t_k+T} L_3(z_3(\tau), \hat{z}_2(\tau), u_3(\tau))\ d\tau + G_3(z_3(t_k+T)) \right\}$$

$$\begin{aligned} \text{s.t.}\quad & \dot{z}_3(t) = f_3(z_3(t), u_3(t)) \\ & u_3(t) \in U_3, \quad z_3(t_k+T) \in Z_{f3} \\ & \|z_3(t) - \hat{z}_3(t)\| \le \delta^2 \kappa \end{aligned}$$

Compatibility constraint:
- each vehicle transmits plan to neighbors
- stay w/in bounded path of what was transmitted

Theorem. Under suitable assumptions, vehicles are stable and converge to globally optimal solution.

Pf Detailed Lyapunov calculation (Dunbar thesis)



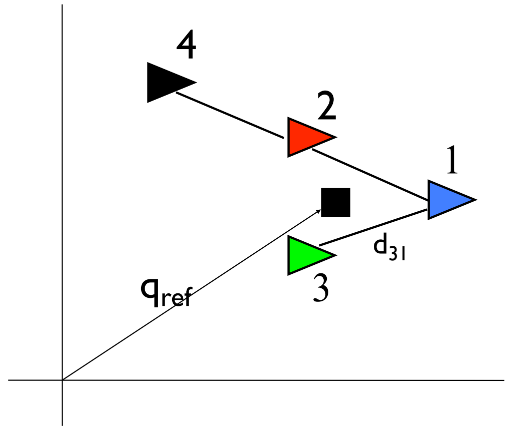
Example: Multi-Vehicle Fingertip Formation

Four vehicles with $\ddot{q}_i = u_i$.

Constraints:
$U_i = \{(v_1, v_2) \in \mathbb{R}^2 : -1 \leq v_{1,2} \leq 1\}$.

Formation defined by:

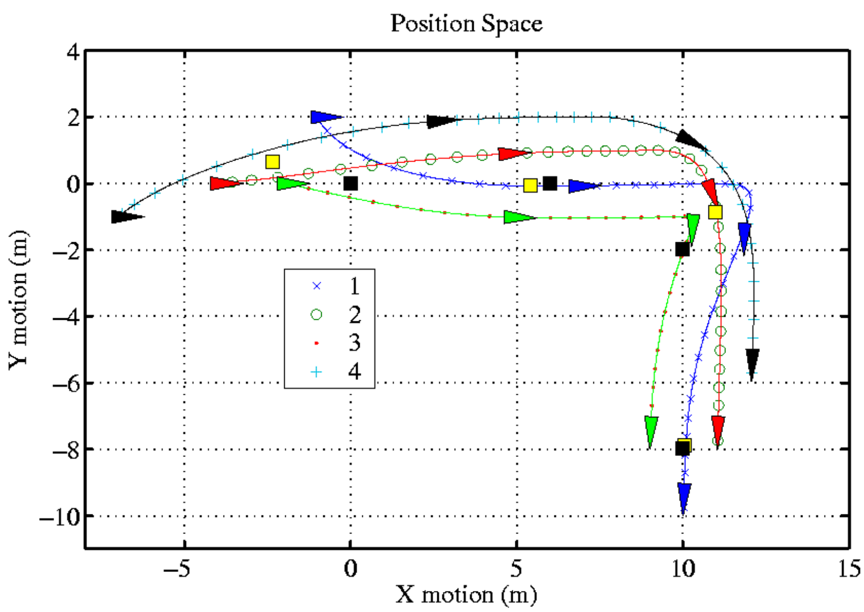
- Relative vectors d_{ij}
- COM of $\{1, 2, 3\}$ tracking signal $(q_{\text{ref}}, \dot{q}_{\text{ref}})$.



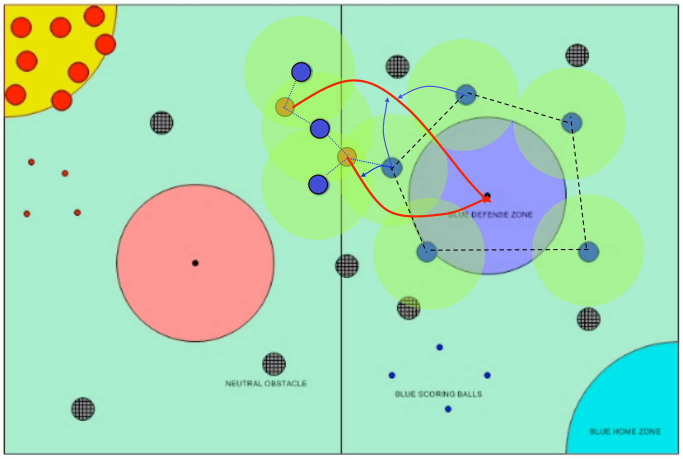
Coupled objective function

$$L(z,u) = ||q_3 - q_1 + d_{31}||^2 + ||q_2 - q_1 + d_{21}||^2 + ||q_4 - q_2 + d_{42}||^2$$
$$+ ||q_{\text{ref}} - (q_1 + q_2 + q_3)/3||^2 + \sum_{i=1}^{4} ||\dot{q}_{\text{ref}} - \dot{q}_i||^2 + ||u_i||^2.$$

Simulation Results



RoboFlag Subproblems

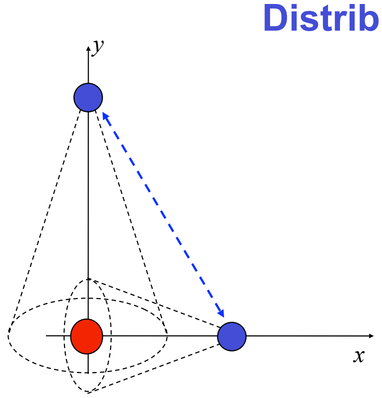


1. Formation control
- Maintain positions to guard defense zone

2. Distributed estimation
- Fuse sensor data to determine opponent location

3. Distributed assignment
- Assign individuals to tag incoming vehicles

Distributed Sensor Fusion

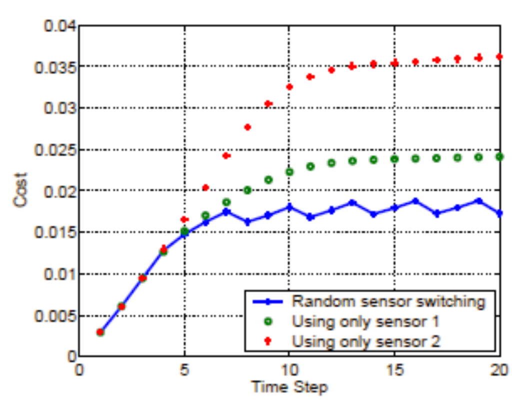


Two agents viewing single object

$$\dot{x} = Ax + Bu$$

$$\dot{y}_1 = C_1 x \qquad \dot{y}_2 = C_2 x$$

- Each sensor maintains its own estimate
- Sensors can communicate w/ packet loss

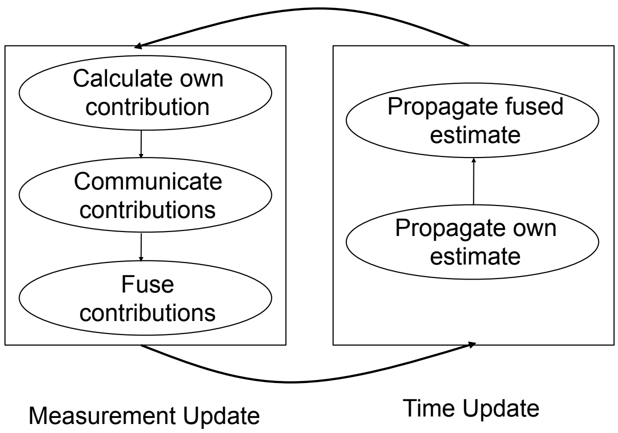


Simulation results
- Exchanging information, even intermittently, decreases error

Optimal estimation
- Q: what should sensors communicate?
- How should packet loss be handled?

Gupta, M & Hassibi
CDC 2004

Decentralized Estimation Algorithm



Two sensor case

- Optimal estimator can be decoupled into two contributions
- Sensor *i* can compute contribution to estimate *j* and transmit
- If information not received, use local info to propagate estimate
- In *n* sensor case, decom-position not as straight-forward; suboptimal

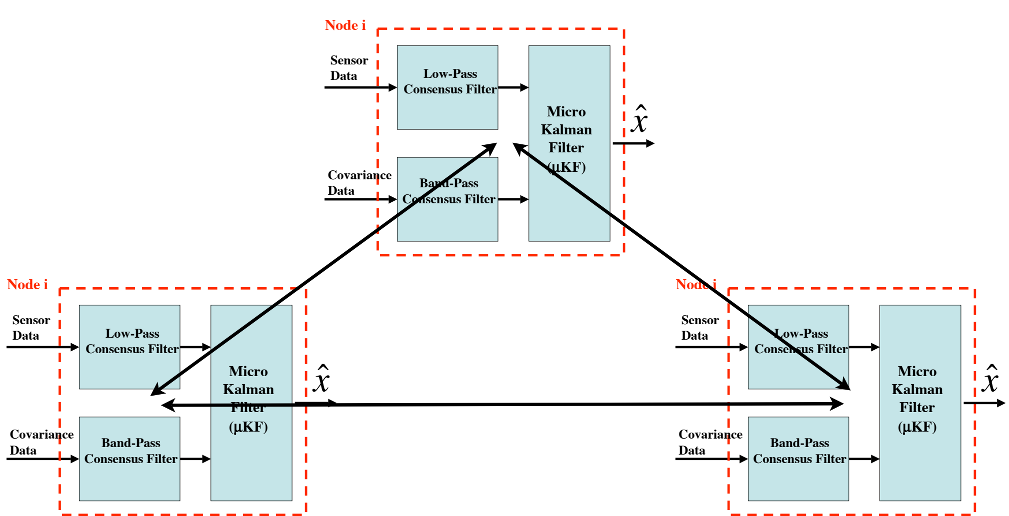
$$P_{k|k}^{-1}\hat{x}_{k|k} = \Gamma_k\Gamma_{k-1}\cdots\Gamma_1 P_{0|-1}^{-1}\hat{x}_{0|-1}^{-1} + \sum_i \left[A_k^i + \Gamma_k\Lambda_{k-1}^i + \cdots + (\Gamma_k\Gamma_{k-1}\cdots\Gamma_1)\Lambda_0^i\right]$$

$$\Lambda_k^i = \left(P_{k|k}^i\right)^{-1}\hat{x}_{k|k}^i - \left(P_{k|k-1}^i\right)^{-1}\hat{x}_{k|k-1}^i \qquad \Gamma_k = \left(P_{k|k-1}^i\right)^{-1} A P_{k-1|k-1}$$

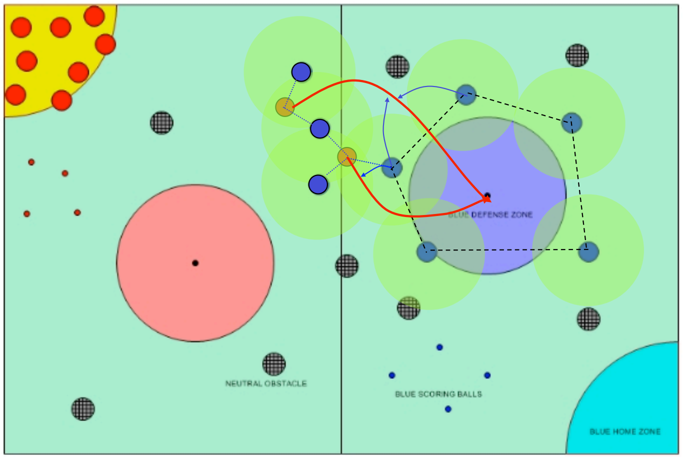
Distributed Kalman Filtering via Consensus

Distributed Kalman filtering

- Maintain local estimates of global average and covariance
- Need to be careful about choosing local rates of convergence
- Use consensus results to reason about stability, global rates of computers



RoboFlag Subproblems



1. Formation control
- Maintain positions to guard defense zone

2. Distributed estimation
- Fuse sensor data to determine opponent location

3. Distributed assignment
- Assign individuals to tag incoming vehicles

CCL: Computation and Control Language

Formal Language for Provably Correct Control Protocols

$\mathtt{P}(\mathtt{k}_1,\mathtt{k}_2)\ \mathtt{:=}\ \{$

$\quad\mathtt{initializers}$

$\quad\mathtt{guard}_1\mathtt{:rule}_1$

$\quad\mathtt{guard}_2\mathtt{:rule}_2$

$\quad\ldots$

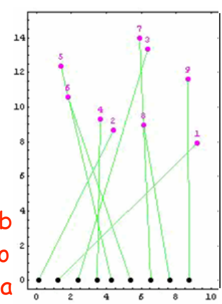
$\}$

"soup" of guarded commands

composition = union

non-shared variab remain local to component progra

$\mathtt{S}(\mathtt{k}_1,\mathtt{k}_2)\mathtt{:=P}(\mathtt{k}_1,\mathtt{k}_2)\mathtt{+C}(\mathtt{k}_1\mathtt{+1})\ \mathtt{sharing\ y,u}$



CCL Protocol for Decentralized Target Allocation

<u>CCL Interpreter</u>

Formal programming language for control and computation. Interfaces with libraries in other languages.

<u>Formal Results</u>

Formal semantics in transition systems and temporal logic. *RoboFlag* drill formalized and basic algorithms verified.

<u>Automated Verification</u>

CCL encoded in the *Isabelle* theorem prover; basic specs verified semi-automatically. Investigating various model checking tools.