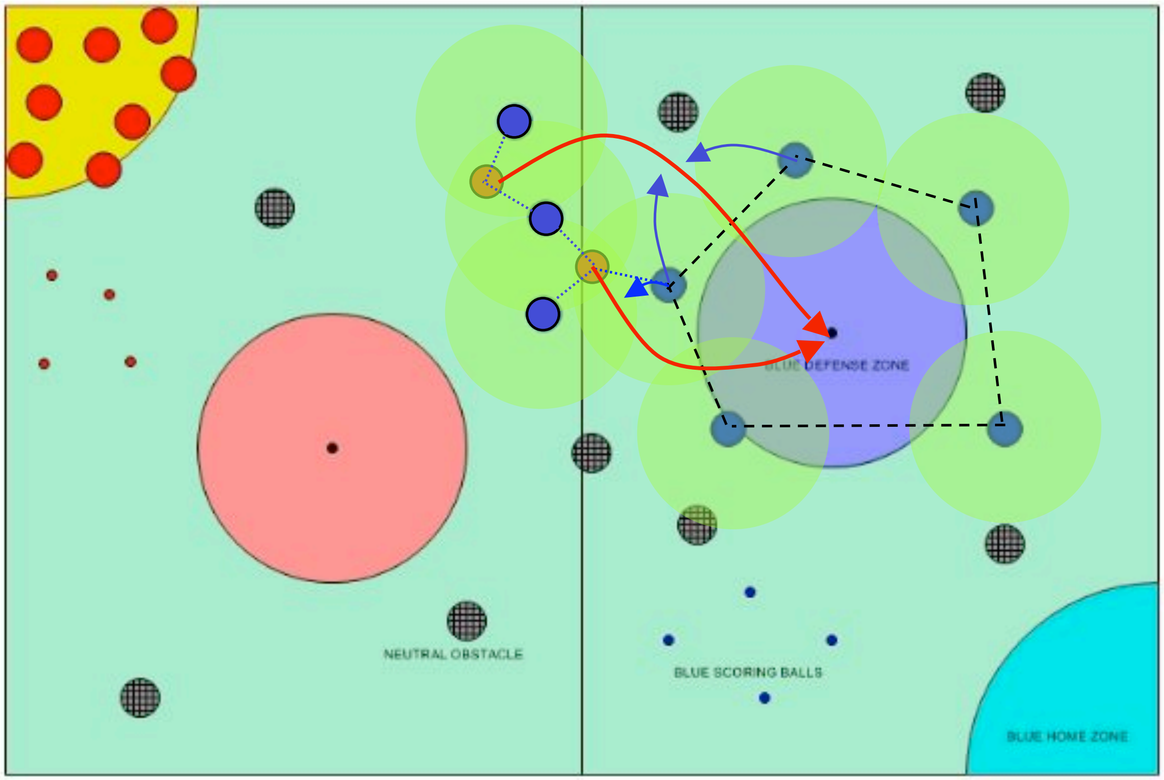
Distributed Consensus and Cooperative Estimation

Richard M. Murray
Control and Dynamical Systems
California Institute of Technology

Domitilla Del Vecchio (U Mich) Bill Dunbar (UCSC) Alex Fax (NGC) Eric Klavins (U Wash)
Reza Olfati-Saber (Dartmouth)

Vijay Gupta Zhipu Jin Demetri Spanos
Abhishek Tiwari Yasi Mostofi
Cedric Langbort (CMI/UIUC)

RoboFlag Subproblems



1. Formation control
- Maintain positions to guard defense zone

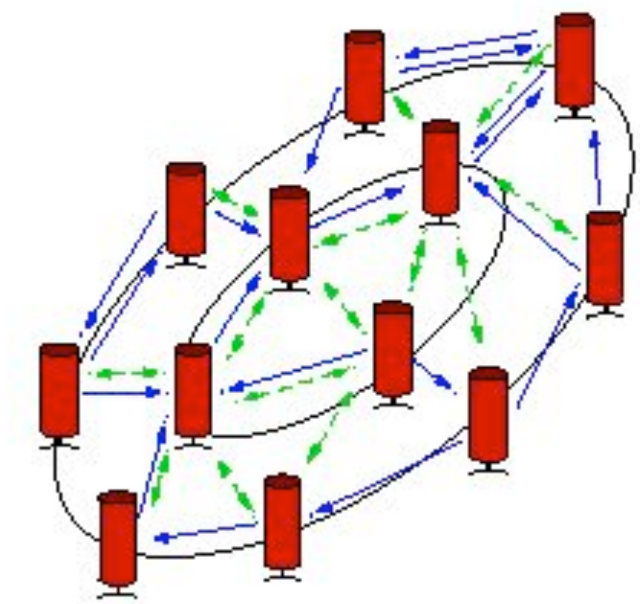
2. Distributed estimation
- Fuse sensor data to determine opponent location

3. Distributed consensus
- Assign individuals to tag incoming vehicles

Goal: develop systematic techniques for solving subproblems
- Cooperative control and graph Laplacians
- Distributed receding horizon control
- Verifiable protocols for consensus and control

→ Implement and test as part of annual RoboFlag competition

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

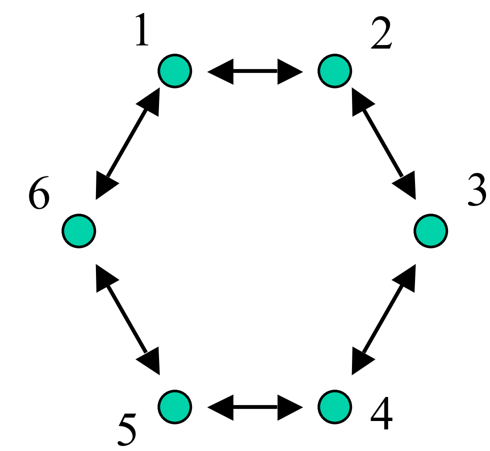
Topological features

- 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?

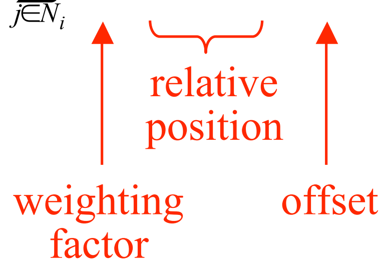
Sample Problem: Formation Stabilization

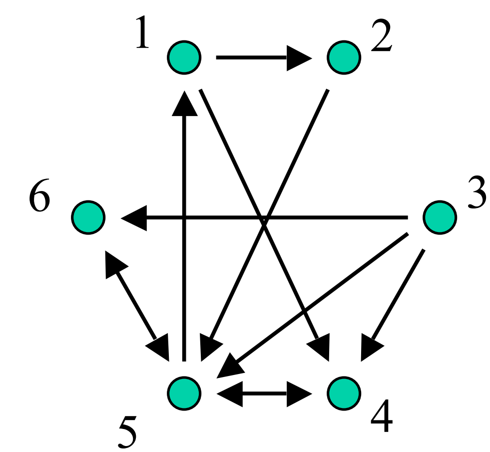
Goal: maintain position relative to neighbors
- “Neighbors” defined by graph
- Assume only sensed data for now
- Assume identical vehicle dynamics, identical controllers?



Example: hexagon formation
- Maintain fixed relative spacing between left and right neighbors

$$e_i = \sum_{j \in N_i} w_j (y_i - y_j - h_{ij})$$





Can extend to more sophisticated “formations”
- Include more complex spatia-temporal constraints

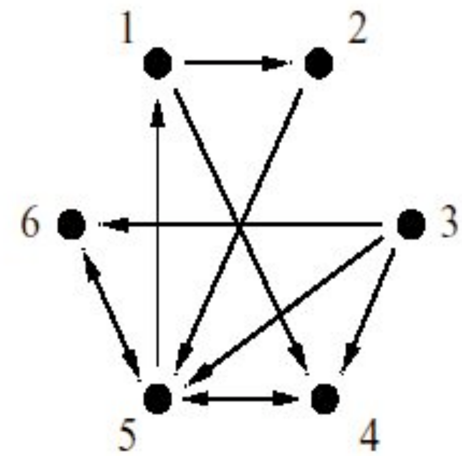
Graph Laplacian

Construction of (weighted) Laplacian

$$L = I - D^{-1}A$$

A = adjacency matrix

D = diagonal matrix, weighted by outdegree

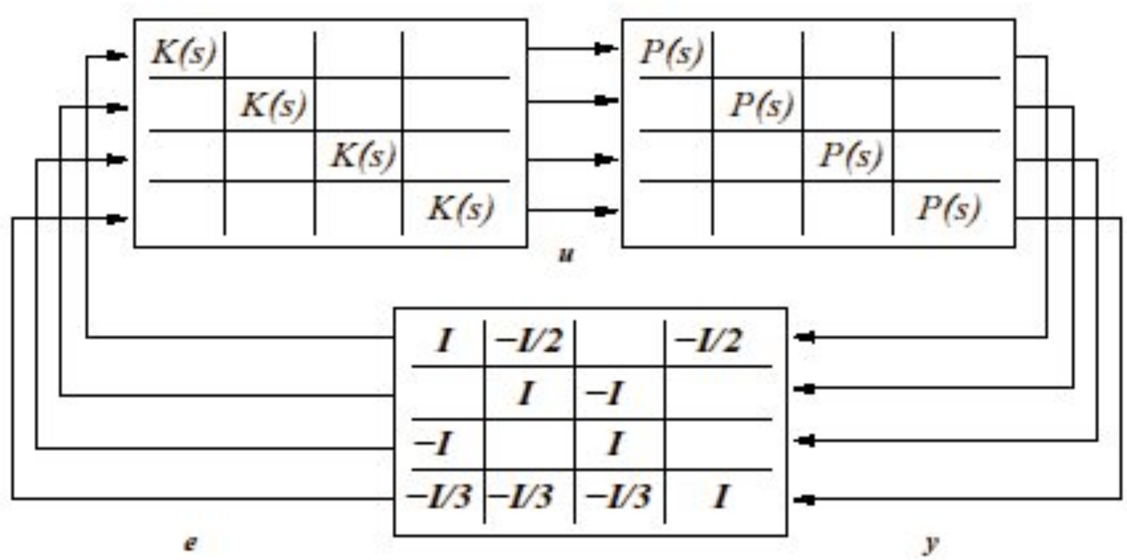


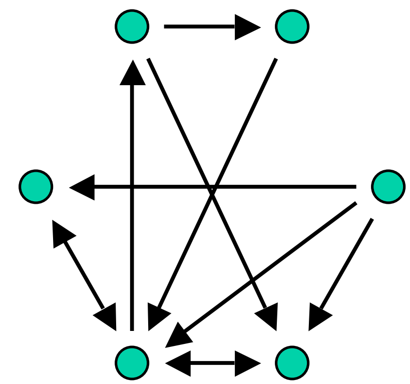
Properties of Laplacian

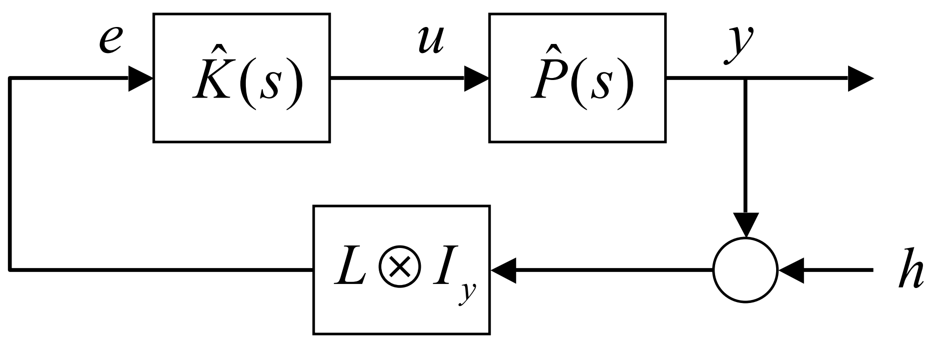
- Row sum equal 0 (stochastic matrix)
- All eigenvalues are non-negative, with at least one zero eigenvalue (from row sum)
- Multiplicity of 0 as an eigenvalue is equal to the number of strongly connected compo-nents of the graph
- All eigenvalues lie in a circle of radius one centered at 1 + 0i (Perron Frobenius)
- For bidirectional (eg, undirected) graphs, eigenvalues are all real, in [0,2]

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

Mathematical Framework





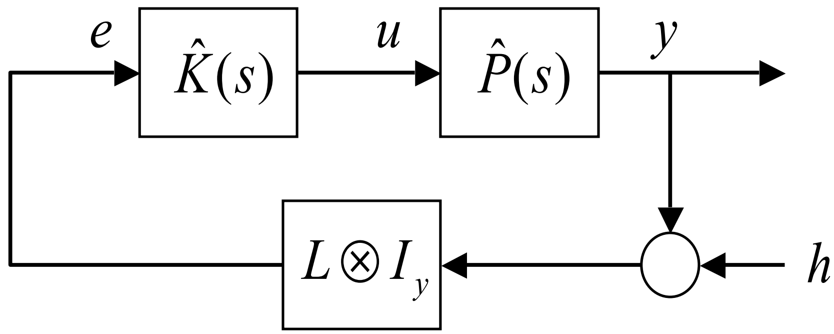


Analyze stability of closed loop

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

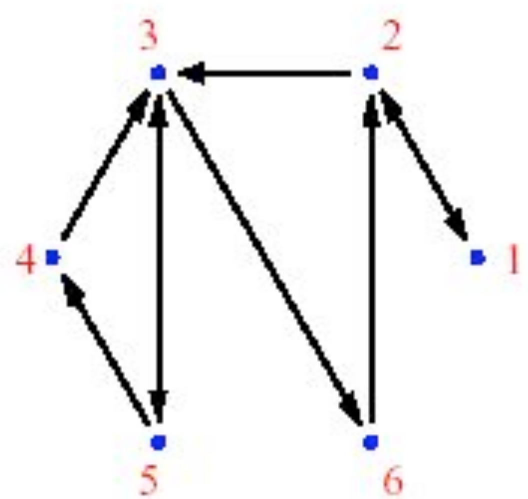
Fax and Murray
IFAC 02, TAC 04

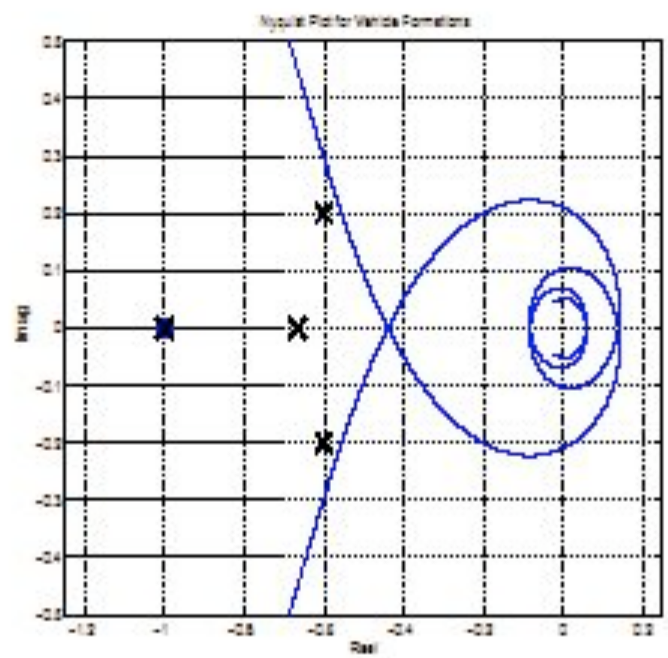
Stability Condition



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

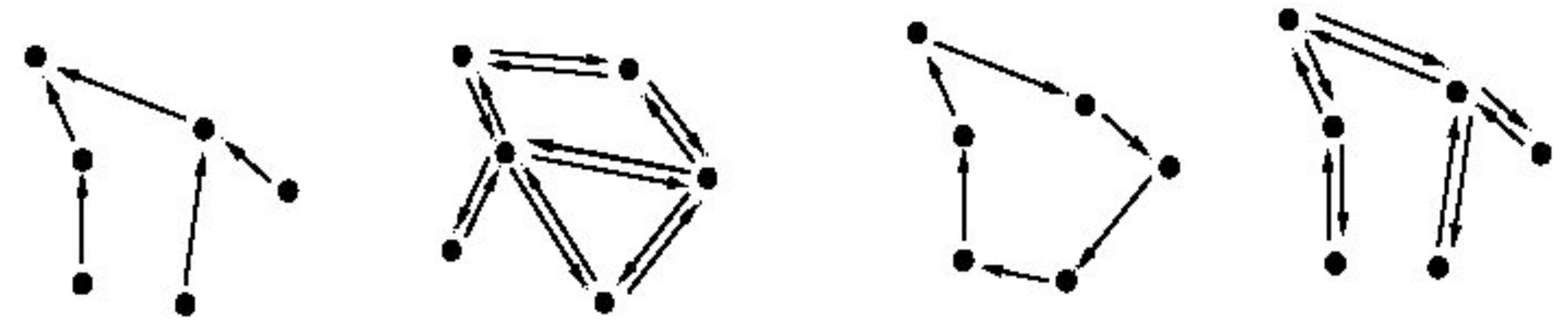
Example $P(s) = \frac{e^{-s\tau}}{s^2}$ $K(s) = K_d s + K_p$



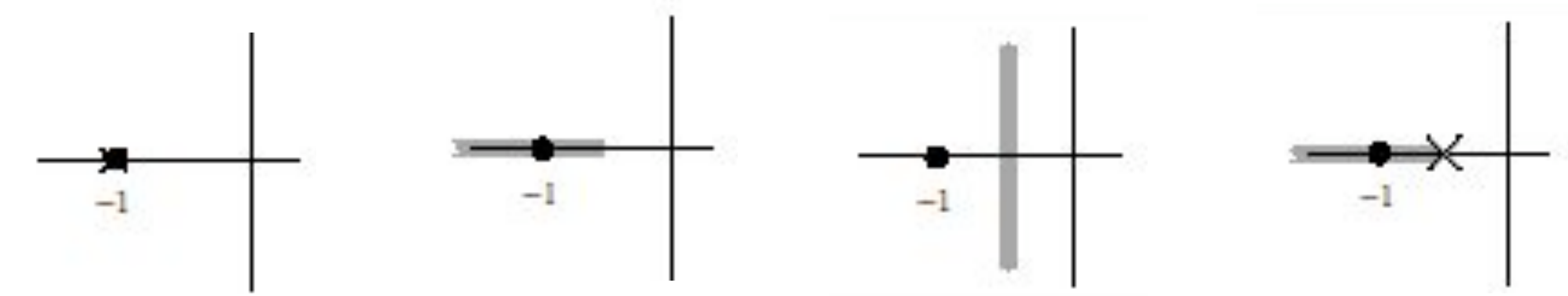


Spectra of Laplacians

Unidirectional tree	Undirected graph	Cycle	Periodic graph



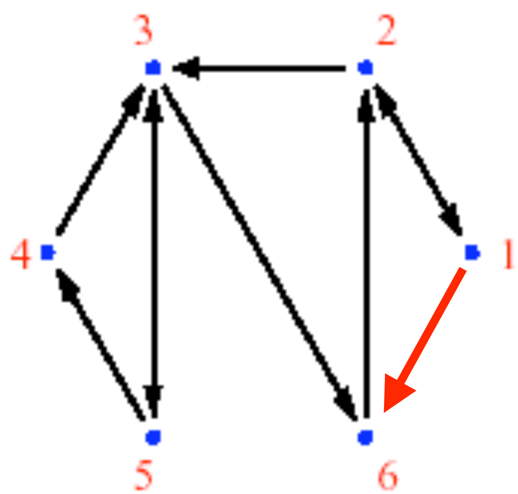
$\lambda = 0,1$	$\lambda \in [0,2]$	$\lambda_i = 1 - e^{2\pi(i-1)j/N}$	$\lambda_1 = 0, \lambda_N = 2$

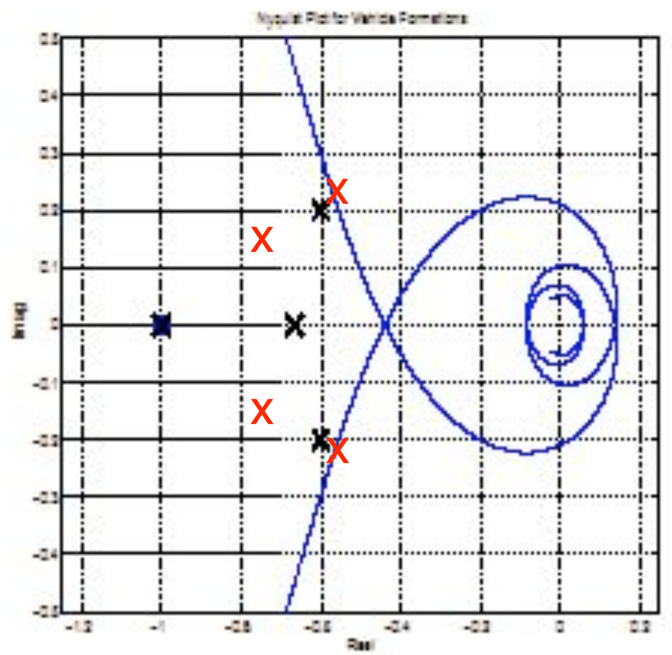


Example Revisited

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

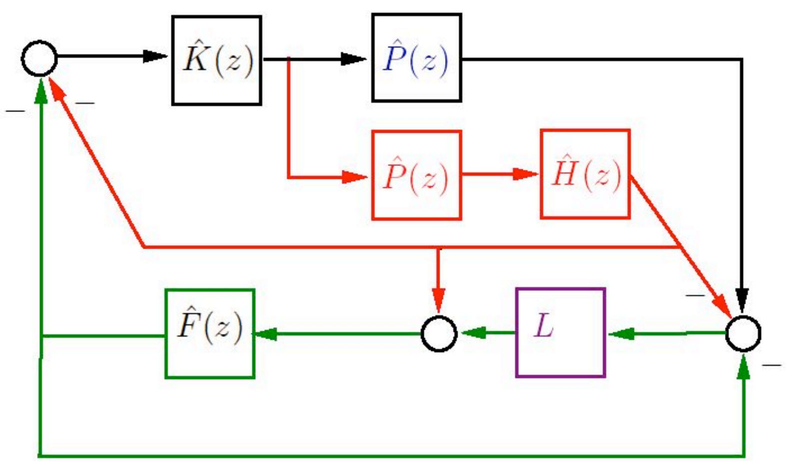
- Poor performance due to interconnection

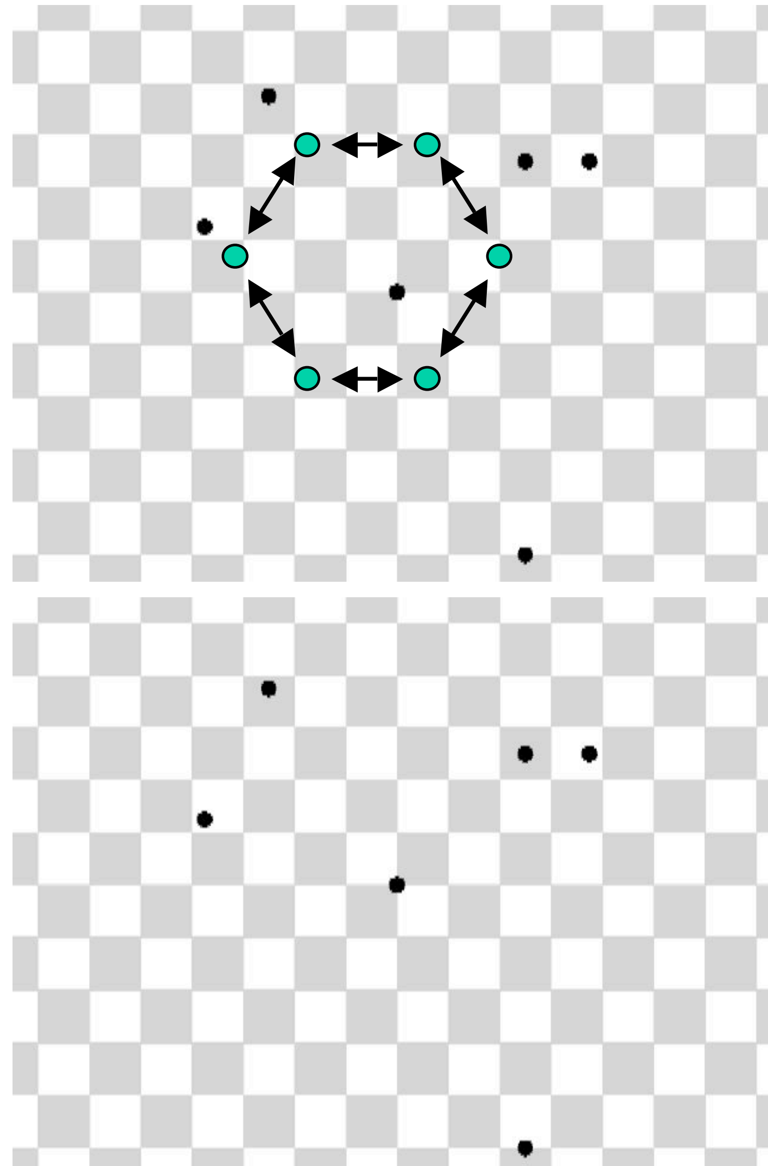
Method #1: tune information flow filter

- 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

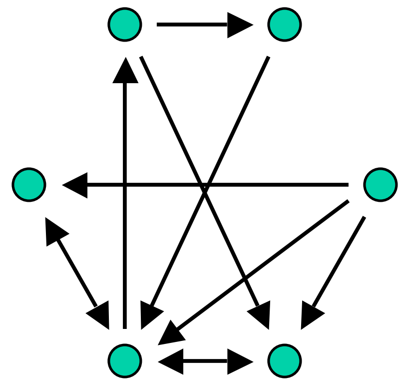


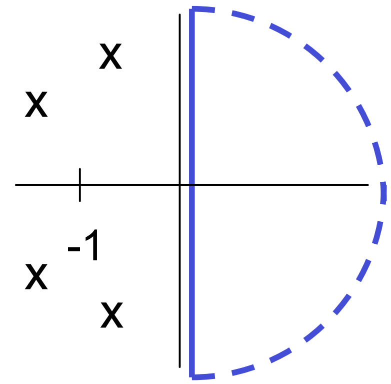


Special Case: Consensus

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

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





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) $\Rightarrow$ all agents converge to average of initial condition

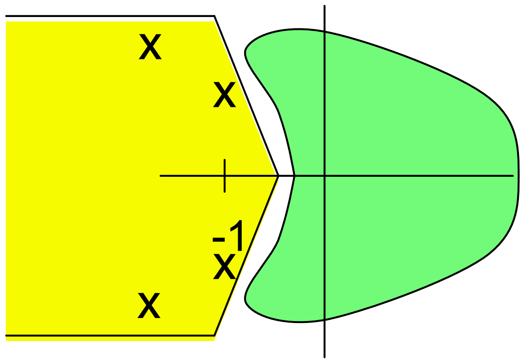
Variations and extensions (Jadbabaie, Leonard, Moreau, Morse, Olfati-Saber, Xiao, …)

- 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

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



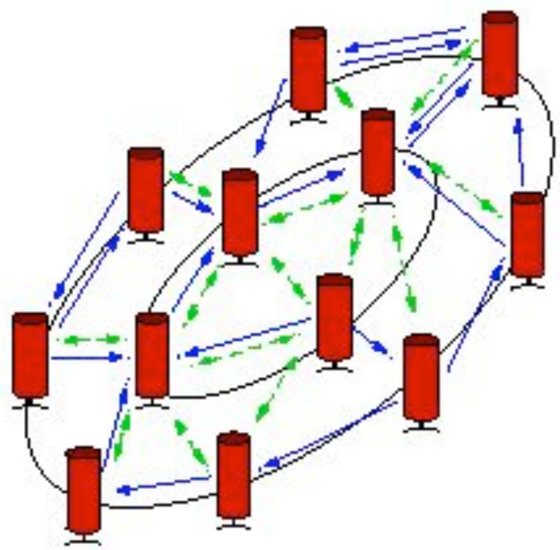
- Possible approach: exploit for of characteristic polynomial

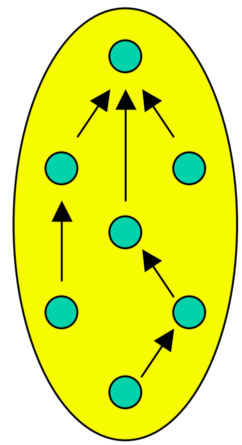
$$\lambda(s) = \text{``}s^n + \left(\sum w_i\right) + \Big(\sum_{2\ \text{cycles}} w_i w_j\Big) s^{n-2} + \Big(\sum_{3\ \text{cycles}} w_i w_j w_k\Big) s^{n-3} + \cdots + \Big(\sum_{N\ \text{cycles}} w_1 \cdots w_n\Big)\text{''}$$

Jin and Murray
CDC 04

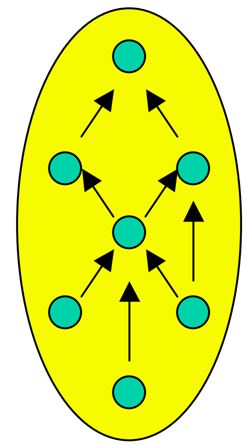
Performance

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*

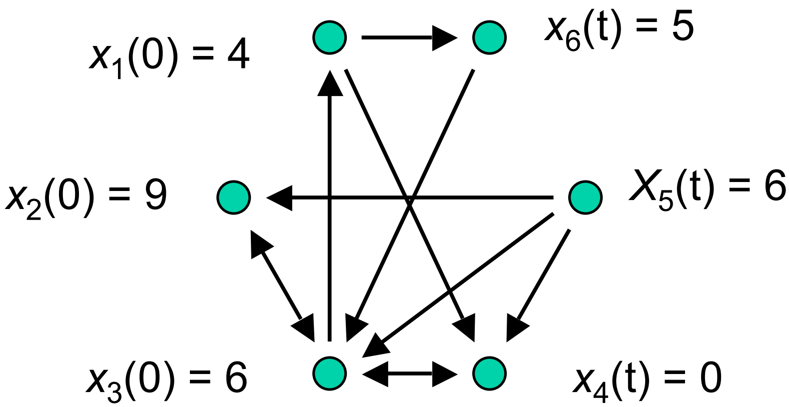
$$\frac{\|E\|_\infty}{\|V\|_\infty} \leq \left(\frac{1-M^n}{1-M} + \frac{\rho(\Theta_2)M^n}{1-\rho(\Theta_2)M}\right)\cdot \|N(s)\|_\infty$$

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

Gupta, Langbort and Murray
CDC 06

Robustness

What happens if a single node "locks up"



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

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

Alice Overview

Team Caltech

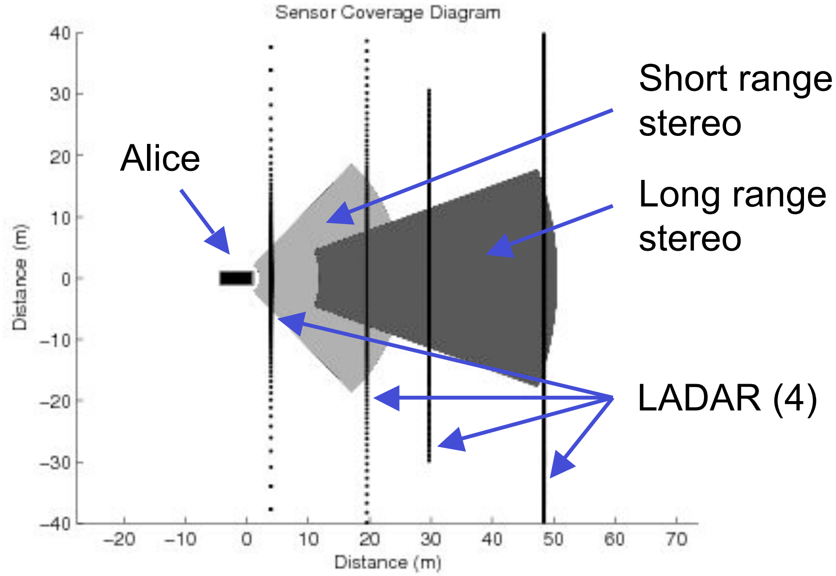
- 50 students worked on Alice over 1 year
- Course credit through CS/EE/ME 75
- Summer team: 20 SURF students + 6 graduated seniors + 4 work study + 4 grads + 2 faculty + 6 volunteers (= ~40)

Computing

- 6 Dell 750 PowerEdge Servers (P4, 3GHz)
- 1 IBM Quad Core AMD64 (fast!)
- 1 Gb/s switched ethernet

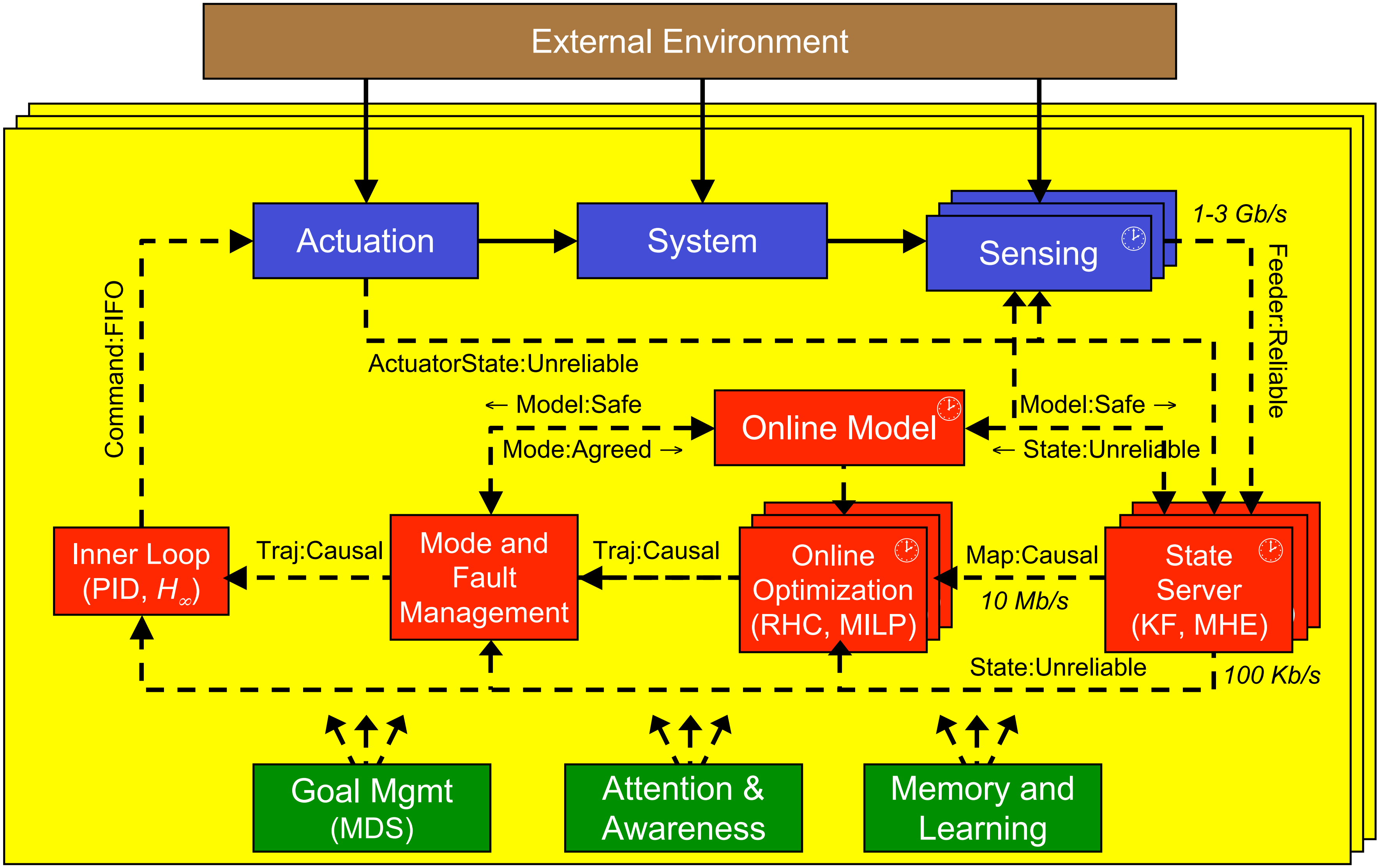
Software

- 15 individual programs with ~50 threads of execution
- Sensor fusion: separate digital elevation maps for each sensor; fuse @ 10 Hz
- Path planning: optimization-based planning over a 10-20 second horizon



An Architecture for Networked Control Systems

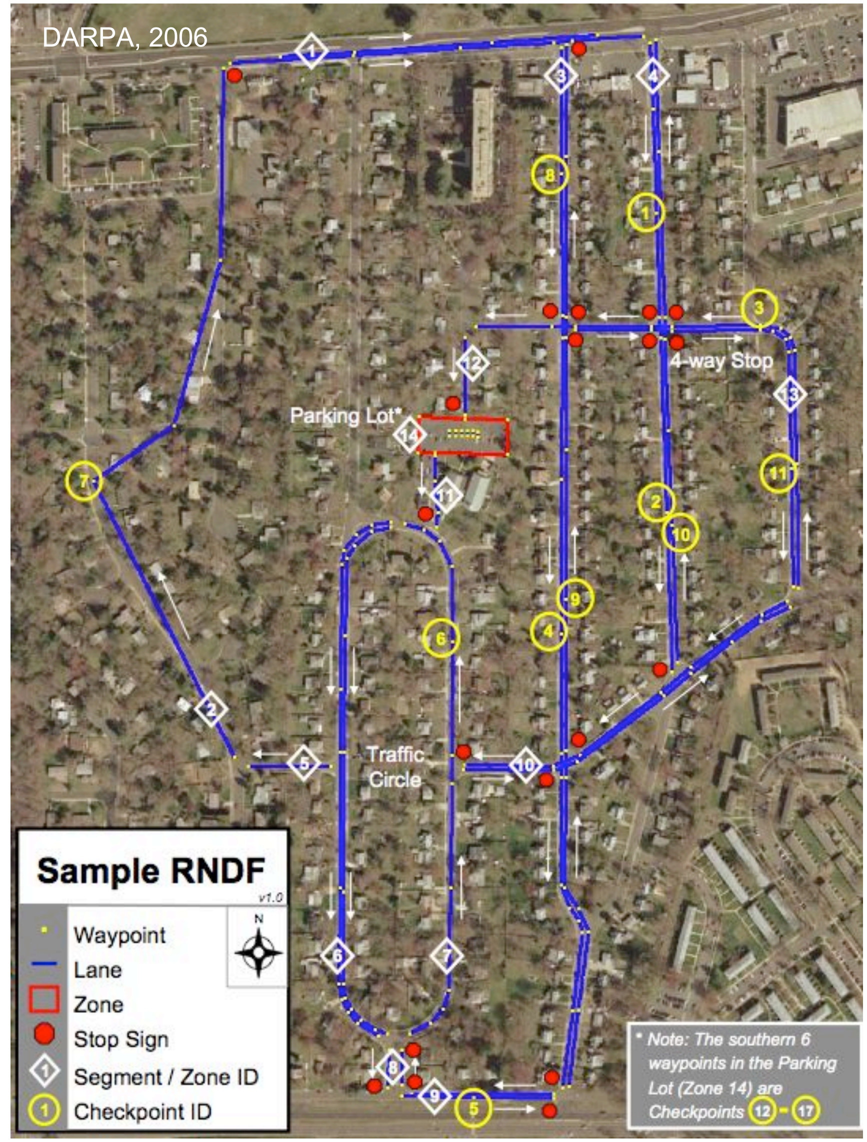
(following P. R. Kumar)



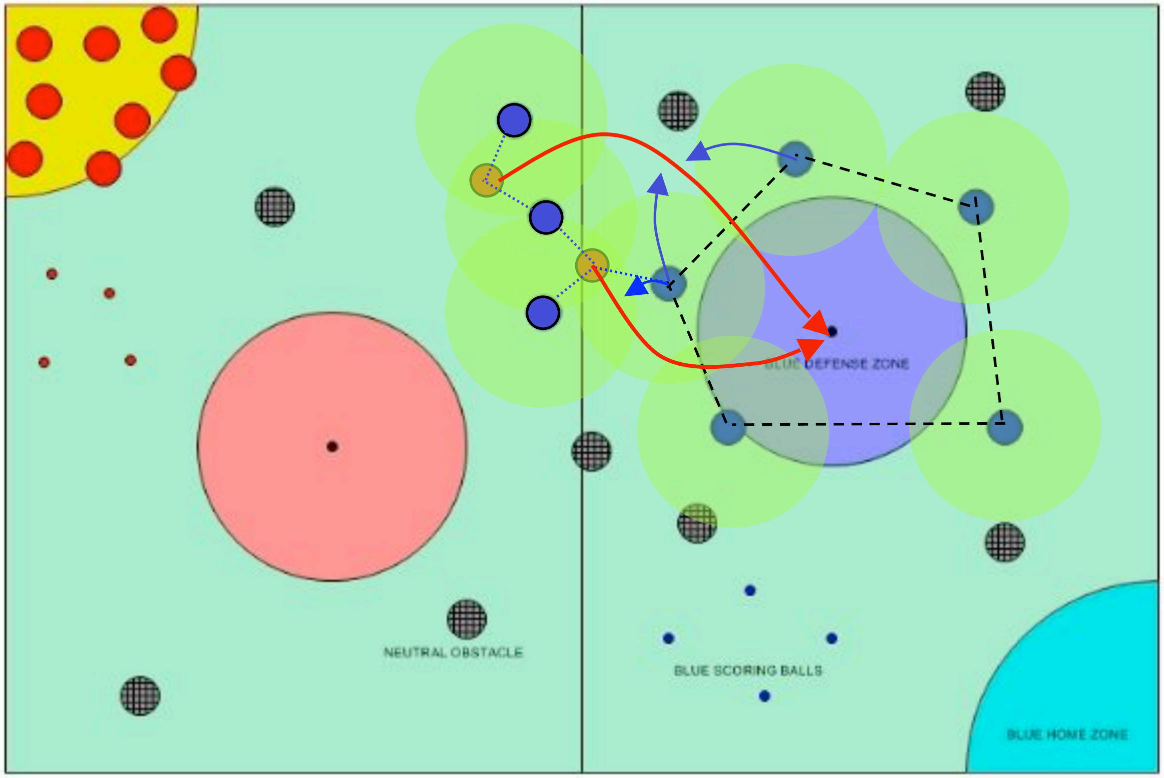
2007 Urban Challenge - 3 November 2007

Autonomous Urban Driving

- 60 mile course, less than 6 hours
- City streets, obeying traffic rules
- Follow cars, maintain safe distance
- Pull around stopped, moving vehicles
- Stop and go through intersections
- Navigate in parking lots (w/ other cars)
- U turns, traffic merges, replanning
- Prizes: $2M, $500K, $250K



Summary: Networked Control Systems

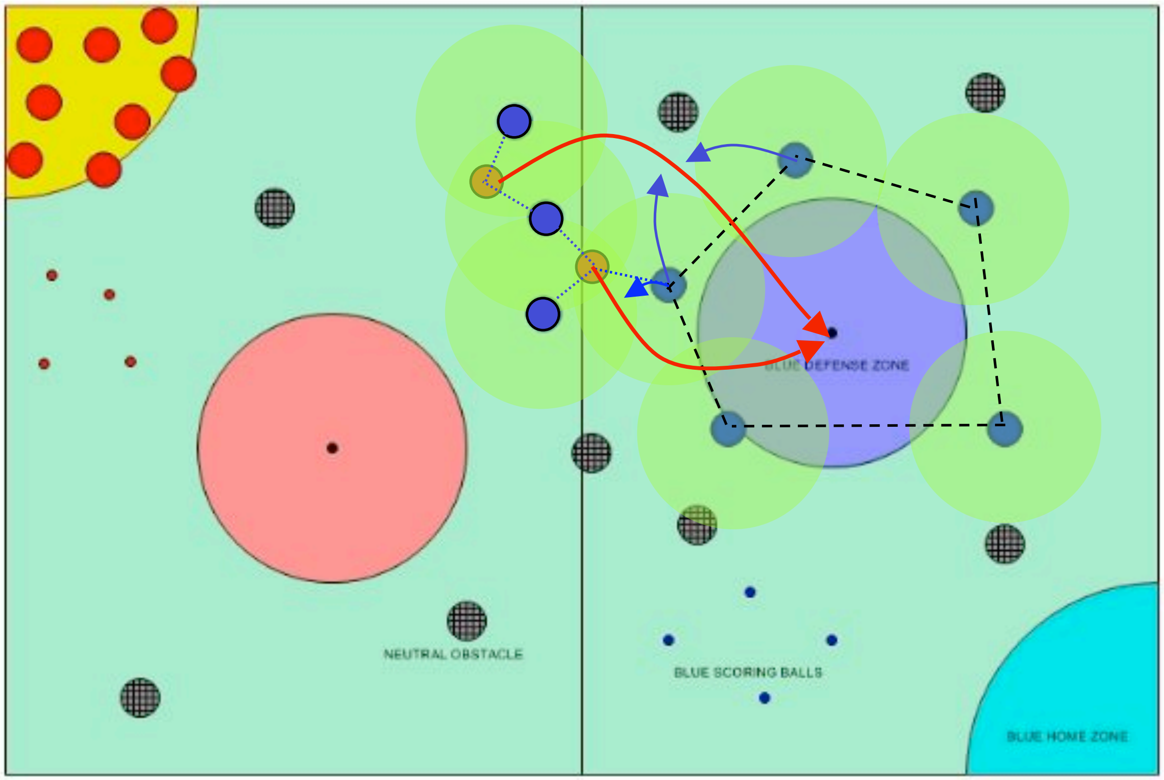


1. **Formation control**
   - Maintain positions to guard defense zone
2. **Distributed estimation**
   - Fuse sensor data to determine opponent location
3. **Distributed consensus**
   - Assign individuals to tag incoming vehicles

Integration of computer science, communications, and control

- Mixture of techniques from computer science, communications, control
- Increased need for reasoning at higher levels of abstraction (strategy)

RoboFlag Subproblems



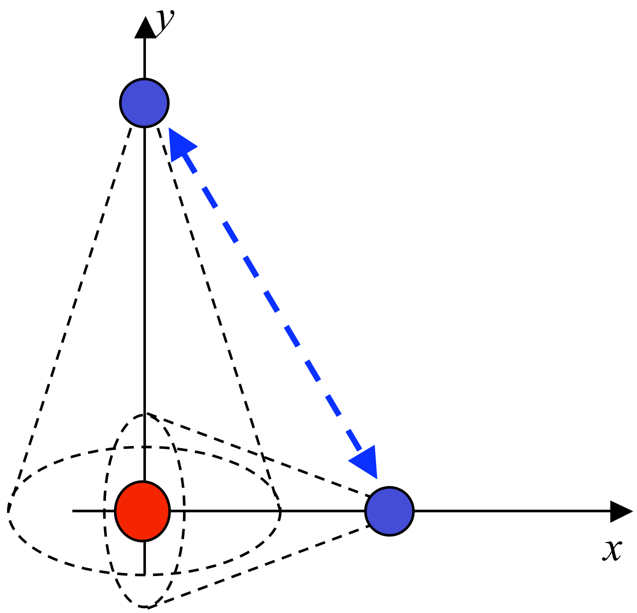
1. **Formation control**
   - Maintain positions to guard defense zone
2. **Distributed estimation**
   - Fuse sensor data to determine opponent location
3. **Distributed consensus**
   - Assign individuals to tag incoming vehicles

Goal: develop systematic techniques for solving subproblems

- Distributed receding horizon control
- Packet-based, distributed estimation
- Verifiable protocols for consensus and control

→ Implement and test as part of annual RoboFlag competition

Distributed Sensor Fusion



Two agents viewing single object

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

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

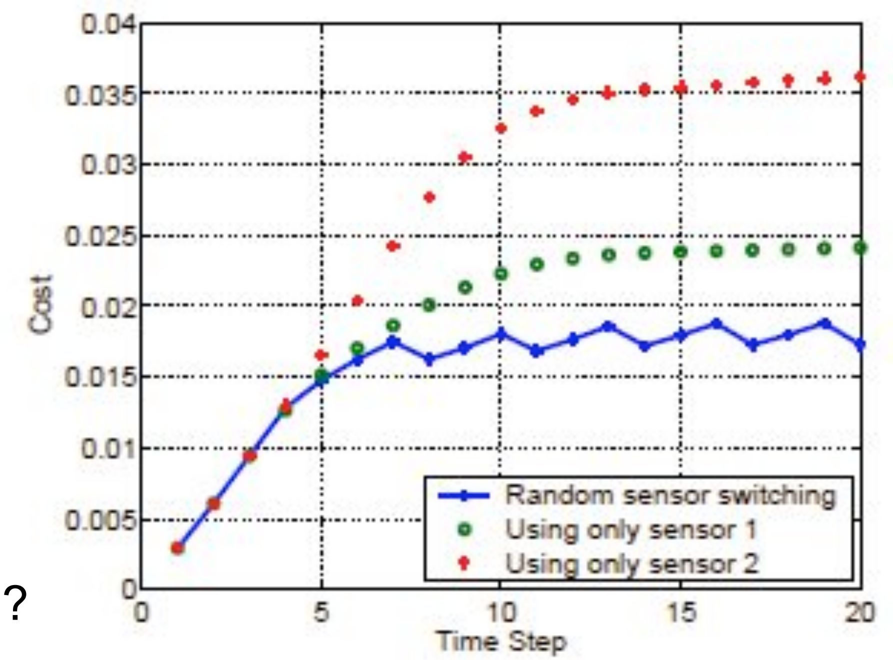
- Each sensor maintains its own estimate
- Sensors can communicate w/ packet loss

Simulation results

- Exchanging information, even intermittently, decreases error

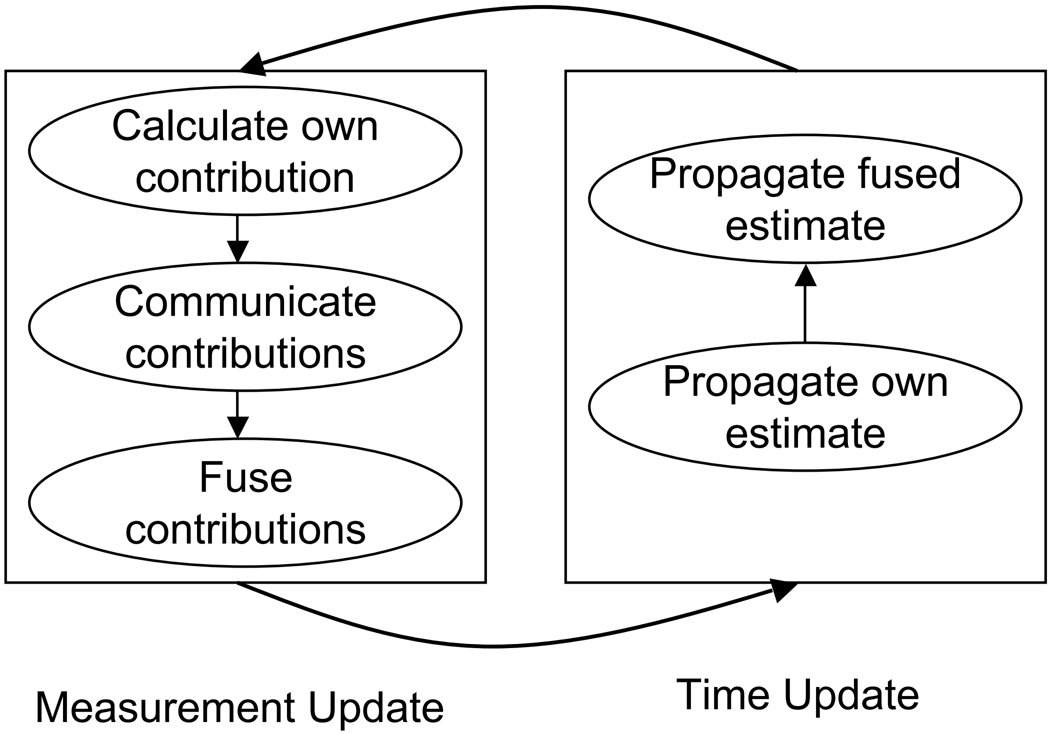
Optimal estimation

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



Gupta, M & Hassibi
CDC 2004 (s)

Decentralized Estimation Algorithm



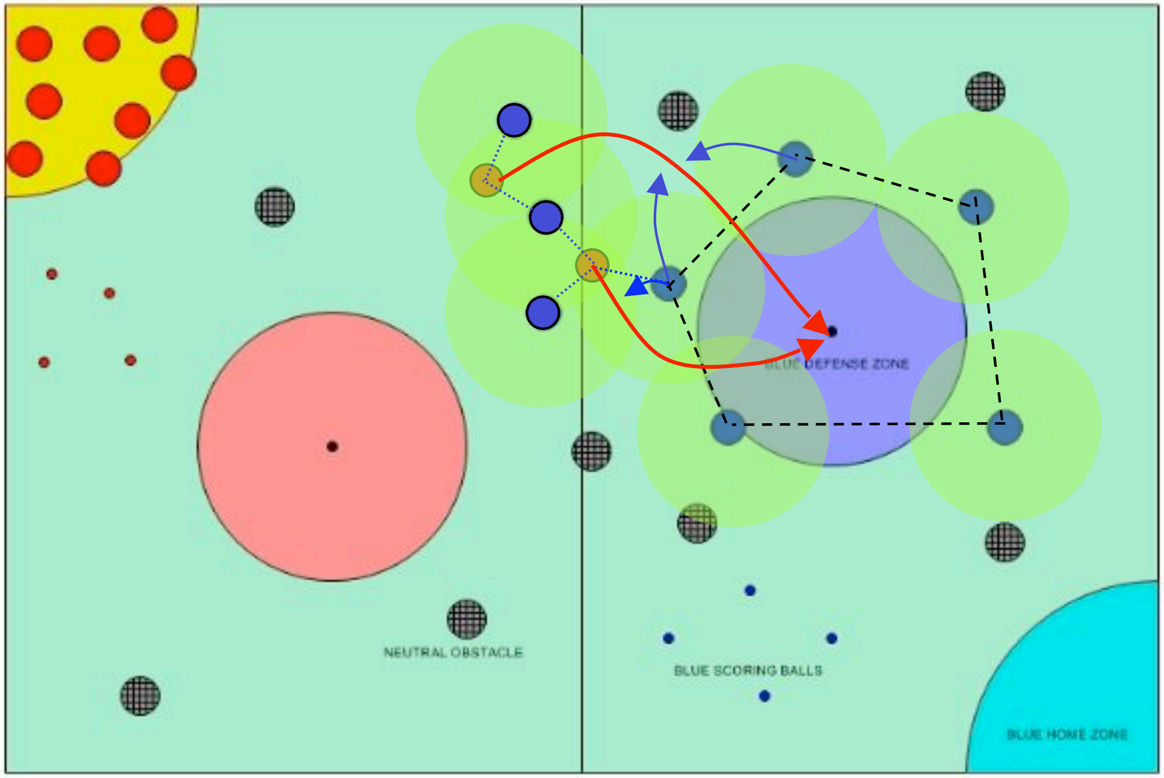
Two sensor case

- 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

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

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

RoboFlag Subproblems



1. **Formation control**
   - Maintain positions to guard defense zone
2. **Distributed estimation**
   - Fuse sensor data to determine opponent location
3. **Distributed consensus**
   - Assign individuals to tag incoming vehicles

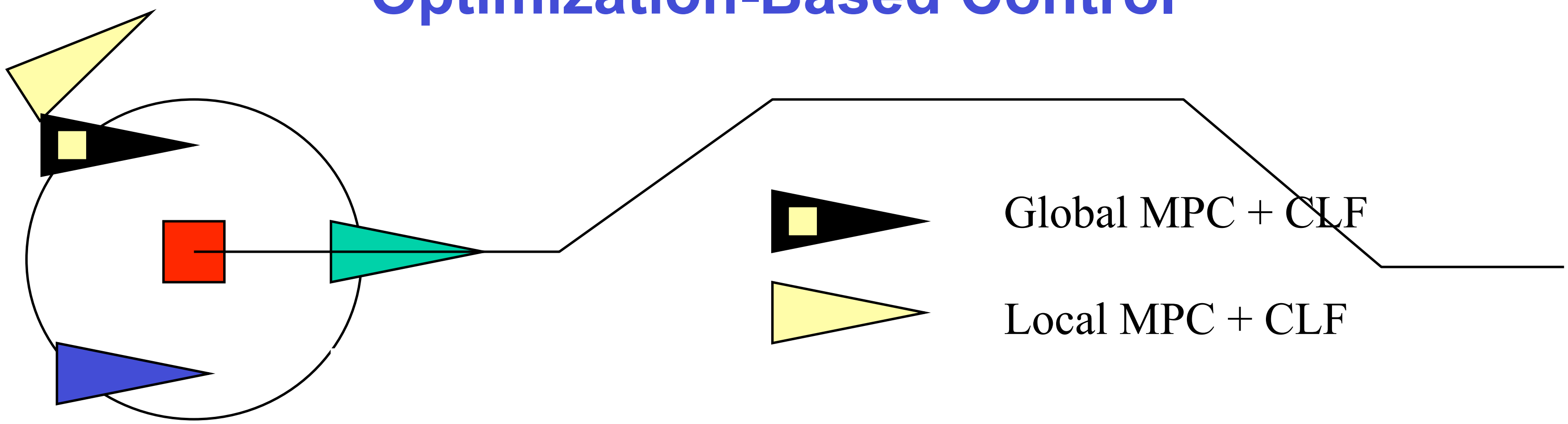
Goal: develop systematic techniques for solving subproblems

- Cooperative control and graph Laplacians
- Distributed receding horizon control
- Verifiable protocols for consensus and control

→ Implement and test as part of annual RoboFlag competition

Dunbar and M
IFAC 02

Optimization-Based Control

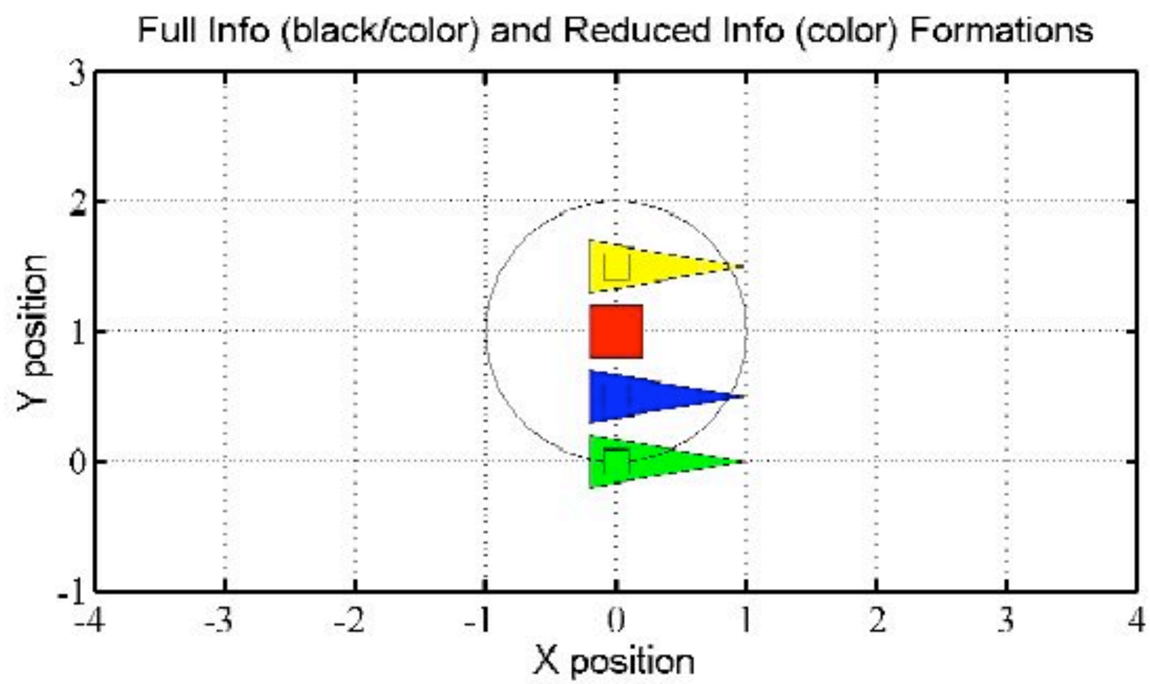


Task:

- 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:

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

$$\text{s.t.} \quad \dot{z}_3(t) = f_3(z_3(t), u_3(t))$$

$$u_3(t) \in U_3, \quad z_3(t_k+T) \in Z_{f3}$$

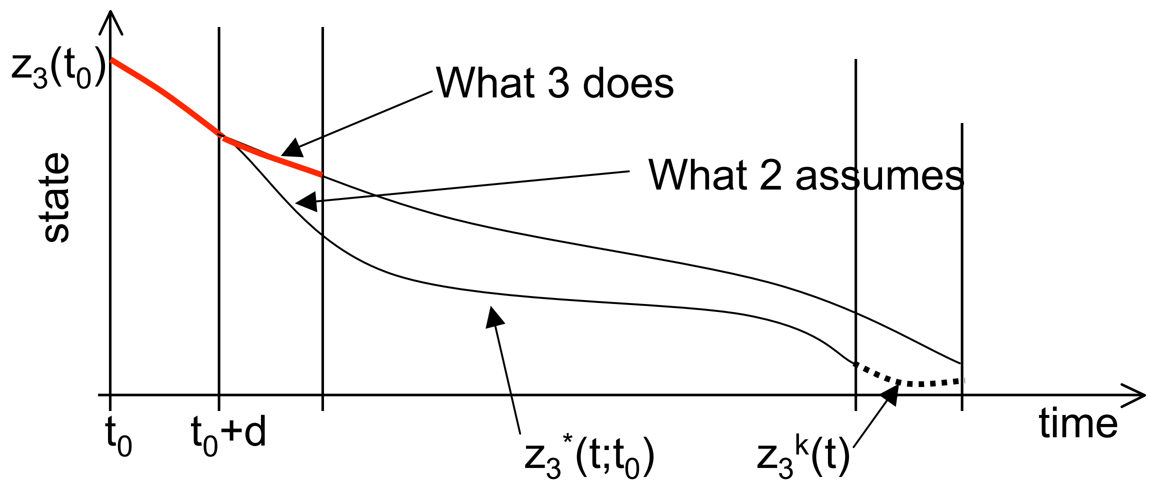
$$||z_3(t) - \hat{z}_3(t)|| \le \delta^2 \kappa$$

Compatibility constraint:

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

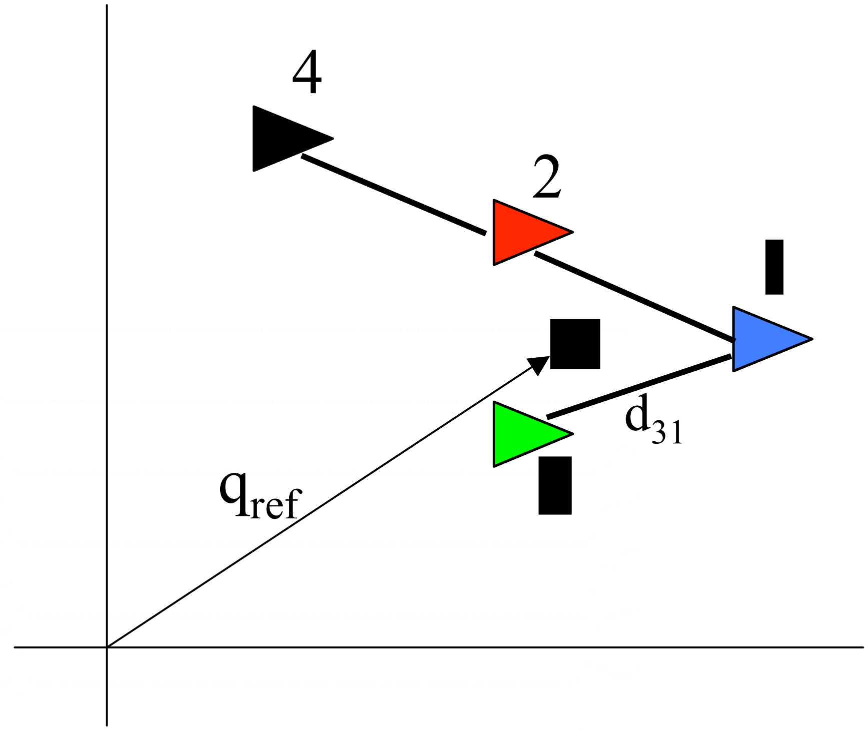
Theorem. Under suitable assumptions, vehicles are stable and converge to global optimal solution.

Pf Detailed Lyapunov calculation (Dunbar thesis)



Example: Multi-Vehicle Fingertip Formation

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



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

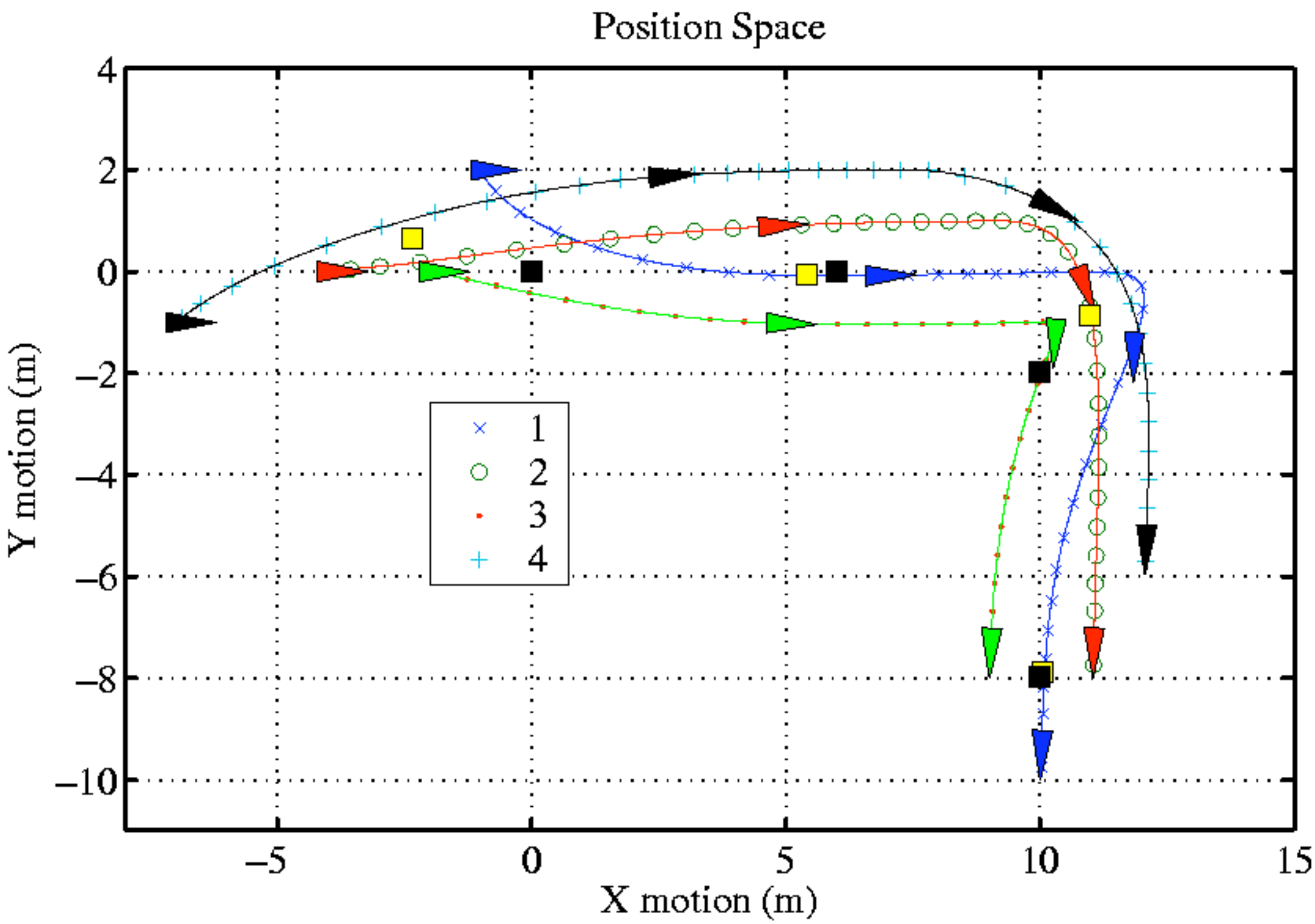
Formation defined by:

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

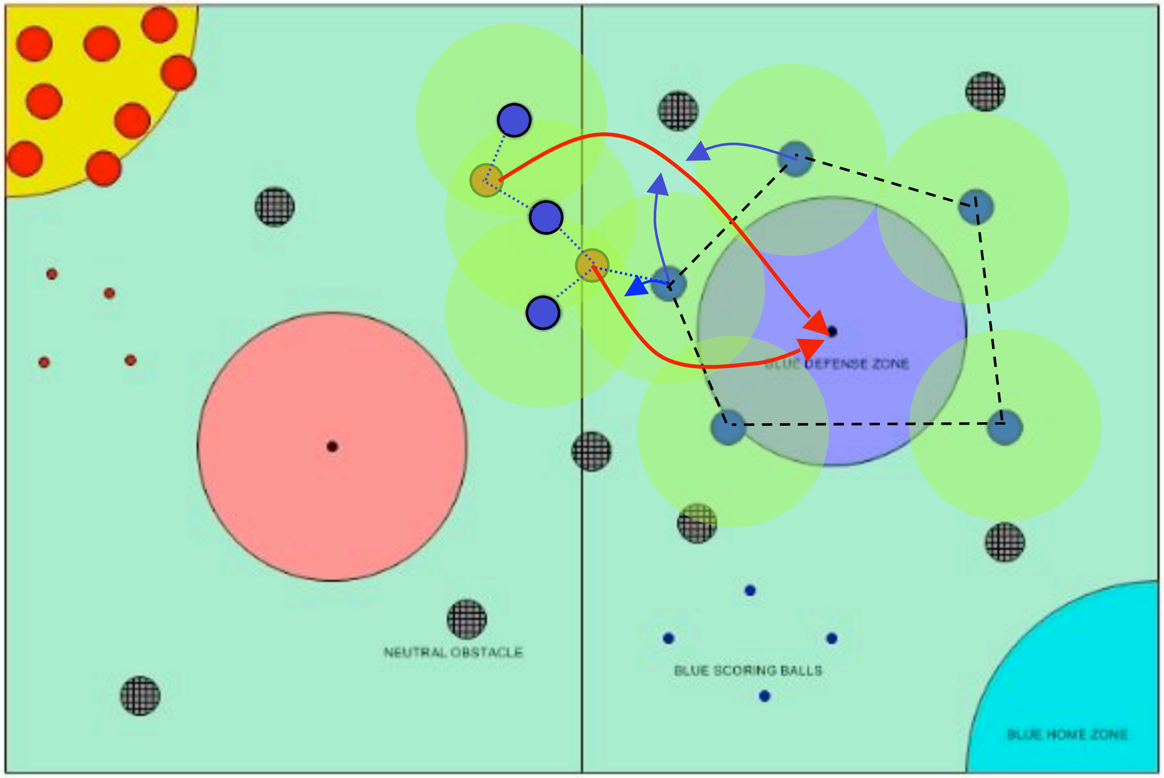
Coupled objective function

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

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 consensus**
   - 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

→ Implement and test as part of annual RoboFlag competition

CCL: Computation and Control Language

Formal Language for Provably Correct Control Protocols

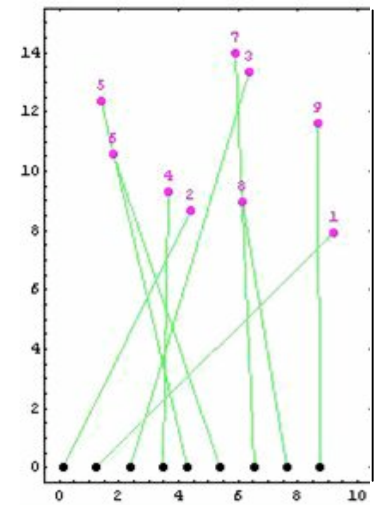
```
P(k1,k2) := {
  initializers
  guard1:rule1
  guard2:rule2
  ...
}
```

"soup" of guarded commands

composition = union

non-shared variables remain local to component programs

```
S(k1,k2):=P(k1,k2)+C(k1+1) 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.

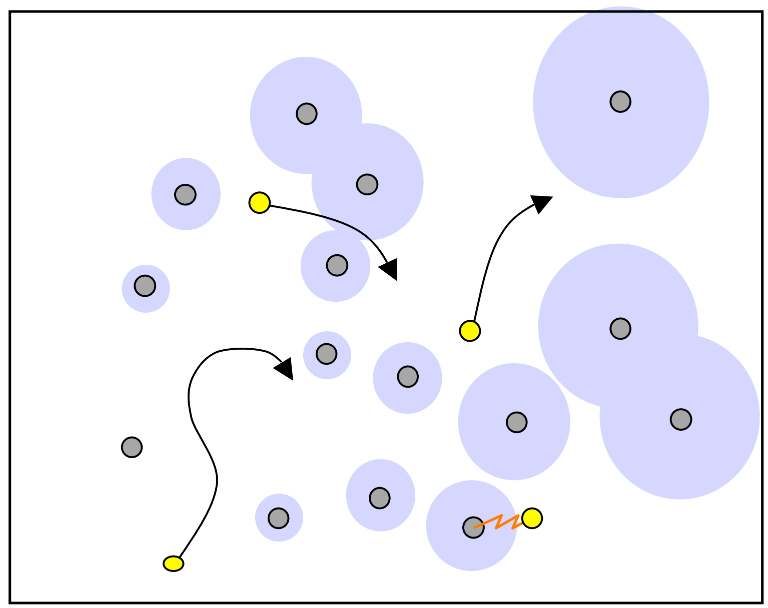
<u>Automated Verification</u>

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

Example #1: Situational Awareness

Communications complexity

- Maintain “situational awareness”
- Assume point-to-point communications and that each robot knows its own position
- Q: how many messages are required for each robot to keep track of all other robots w/in ε?
- A: $O(n^2)$ messages (worst case)



Method #1: Distance Modulated Communication - *O*(*n* log *n*)

- Maintain position estimates to within $k \Pi x_i - x_j \Pi$
- Communicate more often with robots that are closer

Method #2: Wandering Communication Scheme - *O*(*n*)

- Only moving robots need to keep track of position
- Robots transfer knowledge when they stop/start

Proof of correctness using CCL

Klavins WAFR 02

Klavins
CDC, 03

Example #2: RoboFlag Drill

Specification 1 Red Robot Dynamics: $\Pi_{red}(i)$

Initial:

$x_i \in [min, max] \wedge y_i > \max$

Clauses:

$y_i - \delta > 0 \; : \; y_i' = y_i - \delta$

Specification 2 Blue Robot Control: $\Pi_{blue}(i)$

Initial:

$z_i \in [min, max] \; \wedge \; z_i < z_{i+1}$

Clauses:

$z_i < x_{\alpha(i)} \wedge z_i < z_{i+1} - 2\delta \; : \; z_i' = z_i + \delta$

$z_i > x_{\alpha(i)} \wedge z_i > z_{i-1} + 2\delta \; : \; z_i' = z_i - \delta$

Specification 3 Assignment Protocol: $\Pi_{proto}(n)$

Initial:

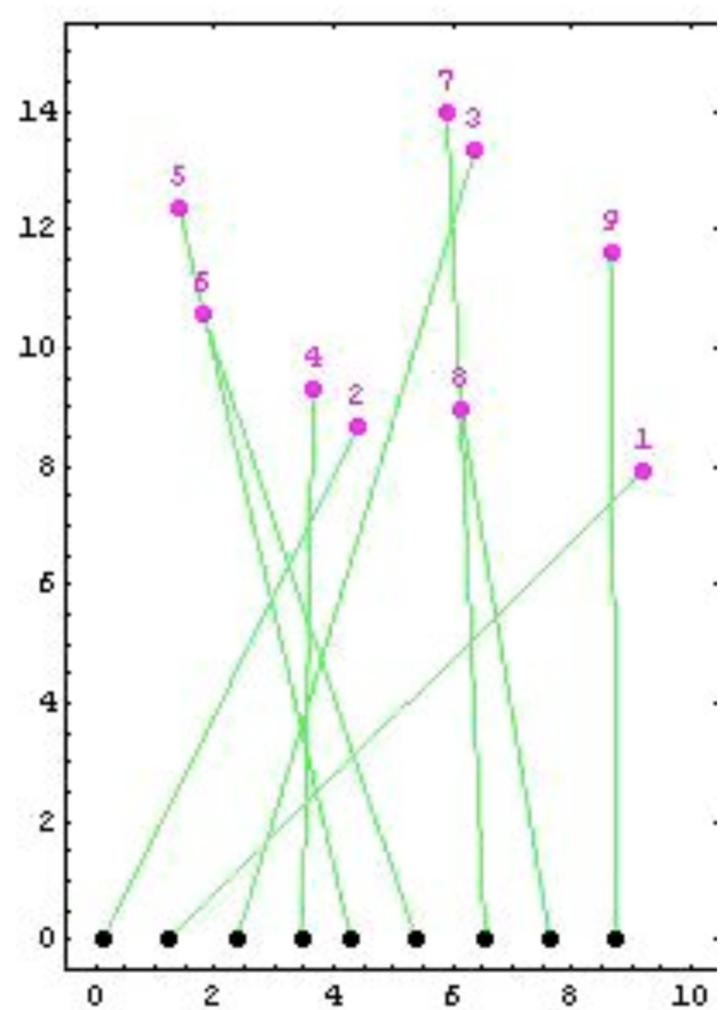
α is a bijection from $\{1, ..., n\}$ to $\{1, ...n\}$.

Clauses:

$switch_{1,2} \; : \; (\alpha(1)', \alpha(2)') = (\alpha(2), \alpha(1))$

...

$switch_{n-1,n} \; : \; (\alpha(n-1)', \alpha(n)') = (\alpha(n), \alpha(n-1))$



Formal Specifications

Safety (Defenders do not collide)

$$z_i < z_{i+1} \ \mathbf{co} \ z_i < z_{i+1}$$

Stability (switch stays false)

Robots are "far enough" apart.

$$\forall i \ . \ \underbrace{y_i > 2\delta \wedge z_i + 2\delta < z_{i+1}} \wedge \neg switch_{i,i+1} \ \mathbf{co} \ \neg switch_{i,i+1}$$

Lyapunov Stability:

$$\forall i \ . \ z_i + 2\delta m < z_{i+1} \wedge \exists j \ . \ switch_{j,j+1} \wedge V = m \ \mathbf{co} \ V < m$$

Example #3: Observation of CCL Programs

Del Vecchio & Klavins, CDC'03

Problem: Determine state of communications protocol used by a group of robots given their physical movements.

Assumptions: Protocol and motion control are described in CCL like language.

Results:
- Defns of observability, etc. for CCL programs
- Construction and analysis of observer that converges when the system is "weakly" observable
- Construction of an efficient observer for Roboflag drill in particular
- Everything specified in CCL