Goals:

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

Reading:


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
\[
\dot{x}^i = f^i(x^i, u^i) \quad x^i \in \mathbb{R}^n, u^i \in \mathbb{R}^m
\]
\[
\dot{y}^i = h^i(x^i) \quad y^i \in SE(3)
\]

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

Communications graph \( \gamma \)
- 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
\[
u^i = \gamma(x, \alpha) \quad \{ g_j^i(x, \alpha) : r_j^i(x, \alpha) \}\]
\[
\alpha' = \begin{cases} 
  r_j^i(x, \alpha) & g(x, \alpha) = \text{true} \\
  \text{unchanged} & \text{otherwise}
\end{cases}
\]

Decentralized strategy
\[
u^i(x, \alpha) = u^i(x^i, \alpha^i, x^{-i}, \alpha^{-i})
\]
\[
x^{-i} = \{ x^{j1}, \ldots, x^{jm} \}
\]
\[
j_k \in \mathcal{N}^i \quad m_i = |\mathcal{N}^i|
\]
- Similar structure for role update

Information Flow in Vehicle Formations

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 \( \Rightarrow \) information feedback loops

Example: satellite formation
- Blue links represent sensed information
- Green links represent communicated information

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

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

Relative position weighting factor 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
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
\[ z_i = \frac{1}{|N_i|} \sum_{j \in N_i} (y_j - y_i) \]

- 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)

Theorem
Let \( L \) be the weighted Laplacian of the communications graph \( G \). The closed loop system is (neutrally) stable iff the systems are stable for each eigenvalue \( \lambda_i \) of \( L \).

Remarks
- Stability is based on check of \( n \) decoupled systems
- \( \lambda_i \) plays the role of a "loop gain": describes how your output affects your input

Sketch of Stability Proof
\[ \dot{x} = Ax + Bu \]
\[ z = L\hat{C}x \]

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

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

The resulting dynamics become
\[ \frac{d}{dt} \begin{bmatrix} \hat{x} \\ \hat{\xi} \end{bmatrix} = \begin{bmatrix} \hat{A} + \hat{B}\hat{K}\hat{C}U_{(n)} & \hat{B}\hat{H} \\ \hat{G}\hat{C}U_{(n)} & F \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{\xi} \end{bmatrix} \]

This system is upper triangular, and so stability is determined by the elements on the (block) diagonal:
\[ \frac{d}{dt} \begin{bmatrix} \dot{x}_j \\ \dot{\xi}_j \end{bmatrix} = \begin{bmatrix} A + BK\lambda_j & BH \\ GC\lambda_j & F \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} \]

This is equivalent to coupling the process and controller with a gain \( \lambda_i \).
**Frequency Domain Interpretation**

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

**Example**

\[
P(s) = \frac{e^{-\pi}}{s^2} \quad K(s) = K_d s + K_p
\]

**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**
  - \(\lambda_i = 0, \lambda_N = 2\)
Example Revisited

Example

\[ P(s) = \frac{e^{-\pi}}{s^2} \quad K(s) = g_s + K \]

- 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

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 \( w_{ij} = 1/(\text{in-degree}) \) + graph is balanced (same in-degree for all nodes) \( \Rightarrow \) 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

\[ \lambda(s) = s^n + \left( \sum w_i \right) s^{n-1} + \left( \sum_{\text{2 cycles}} w_i w_j \right) s^{n-2} + \]
\[ \left( \sum_{\text{3 cycles}} w_i w_j w_k \right) s^{n-3} + \ldots + \left( \sum_{\text{N cycles}} w_1 \ldots w_N \right) ^n \]

- Possible approach: exploit form of characteristic polynomial
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
\[ \frac{\|E_{\infty}\|}{\|V_{\infty}\|} \leq \left( \frac{1 - M}{1 - M + \frac{\rho(\Theta_2)M^n}{1 - \rho(\Theta_2)M}} \right) \cdot \|N(s)\|_{\infty} \]

where \( \rho(\Theta_2) \) is the spectral radius of \( \Theta_2 \) and \( N(s) = \phi(s)^{-1}. \)

Robustness

What happens if a single node “locks up”

- Single node can change entire value of the consensus
- Desired effect for “robust” behavior: \( \Delta x_i = \delta/N \)

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 communications delay
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

\[ x_i^T f_i(x_i) < 0 \text{ for all } i \]

\[ L \otimes BC \succeq 0 \]

- 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 subgraphs after a split maneuver
### Main Idea: Assume Plan for Neighbors

**Individual optimization:**

\[
\min_{u_3(t)} \left\{ \int_{t_k}^{t_k+T} L_3(z_3(\tau), \dot{z}_3(\tau), u_3(\tau)) \, d\tau + G_3(z_3(t_k+T)) \right\}
\]

s.t.  
\[
\begin{align*}
\dot{z}_3(t) &= f_3(z_3(t), u_3(t)) \\
u_3(t) &\in U_3, \quad z_3(t_k+T) \in Z_{f_3} \\
\|z_3(t) - \hat{z}_3(t)\| &\leq \delta^2 \kappa
\end{align*}
\]

**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 $(q_{ref}, \dot{q}_{ref})$.

Coupled objective function

$$L(z, u) = ||q_3 - q_1 + d_{31}||^2 + ||q_2 - q_1 + d_{21}||^2 + ||q_4 - q_2 + d_{42}||^2$$

$$+ ||\dot{q}_{ref} - (q_1 + q_2 + q_3)/3||^2 + \sum_{i=1}^{4} ||\dot{q}_i - \dot{q}_{ref}||^2 + ||u_i||^2.$$
**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

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
\dot{y}_1 &= C_1 x \\
\dot{y}_2 &= C_2 x
\end{align*}
\]

- 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?
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

**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, decomposition not as straightforward; suboptimal

$$P_{k|k-1}^{-1}\hat{x}_{k|k} = \Gamma_k \Gamma_{k-1} \cdots \Gamma_1 P_{0|0}^{-1} \hat{\bar{x}}_{0|0}^{-1} + \sum_i \left[A_{k}^{i} + \Gamma_k A_{k-1}^{i} + \cdots + (\Gamma_k \Gamma_{k-1} \cdots \Gamma_1) A_0^{i}\right]$$

$$\Lambda_{k}^{i} = \left(P_{k|k}^{i}\right)^{-1} \hat{x}_{k|k}^{i} - \left(P_{k|k-1}^{i}\right)^{-1} \hat{x}_{k|k-1}^{i} \quad \Gamma_{k} = \left(P_{k|k}^{i}\right)^{-1} A P_{k-1|k-1}$$

**Decentralized Estimation Algorithm**
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

\[ P(k_1, k_2) := \{ \]
\[ \text{initializers} \]
\[ \text{guard}_1 : \text{rule}_1 \]
\[ \text{guard}_2 : \text{rule}_2 \]
\[ \ldots \]
\[ \} \]

\[ S(k_1, k_2) := P(k_1, k_2) + C(k_3 + 1) \text{ sharing } y, u \]

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

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

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