Real-Time Trajectory Generation and Tracking for Cooperative Control Systems

Richard Murray   Jason Hickey
California Institute of Technology

MURI Kickoff Meeting
14 May 2001

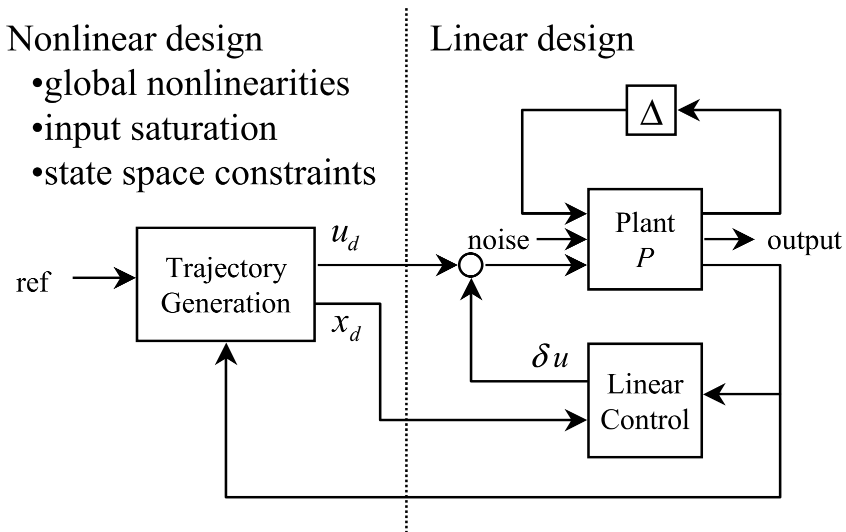
Outline

I. Review of previous work in trajectory generation and tracking

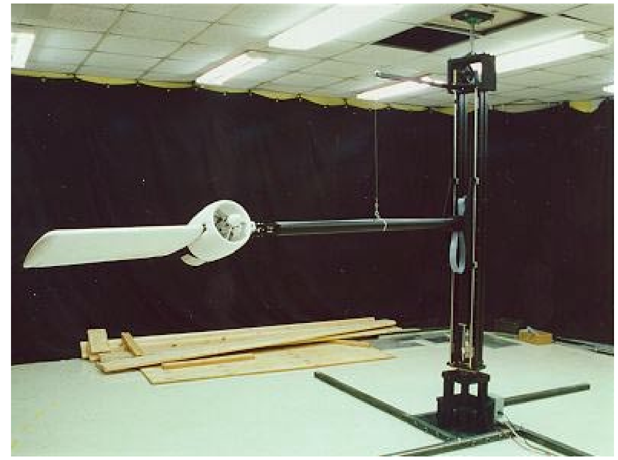
II. Cooperative trajectory generation via optimization-based control

III. Research plan and integration

Trajectory Generation and Tracking Using Differential Flatness



Caltech Ducted Fan



Approach: Two Degree of Freedom Design

- Use online trajectory generation to construct feasible trajectories
- Use (scheduled) linear control for performance and robustness
- For many flight vehicles, system is differentially flat $\Rightarrow$ reduce dynamic system to algebraic equivalent and generate feasible trajectories in real time

Results (PRET + SEC)

- Framework for exploiting differential flatness in real-time trajectory generation
- Necessary and sufficient conditions for flatness classes of (mechanical) systems
- NTG software package for finite time optimal control in presence of constraints
- Implementation and testing on Caltech ducted fan
- Transitions in progress to SEC, Raytheon

Real-Time Trajectory Generation / Optimization

$$\dot{x} = f(x,u)$$

Collocation

$$(x,u) = \sum \alpha_i \psi^i(t)$$
$$\dot{x}(t_i) = f(x(t_i),u(t_i))$$

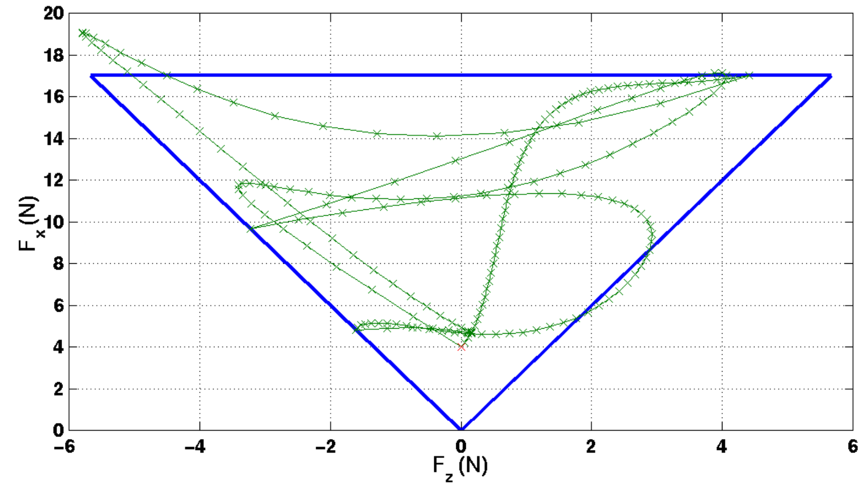
Flatness

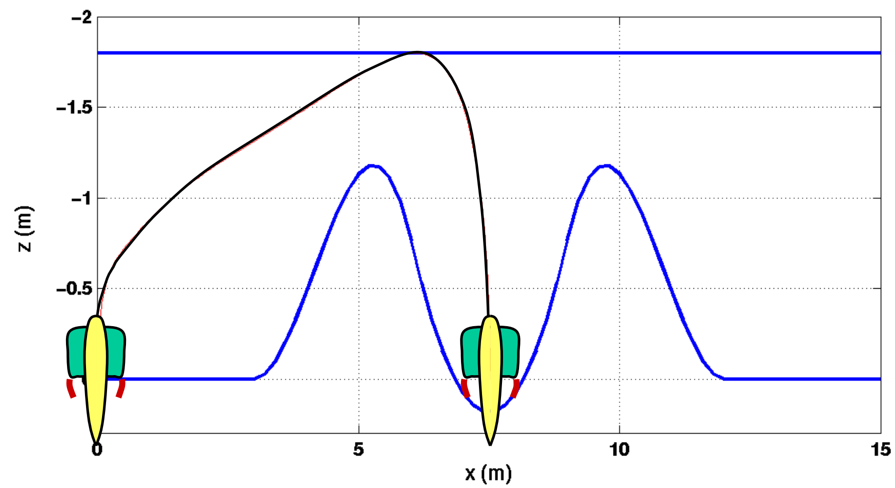
$$z = z(x,u,\dot{u},\ldots,u^{(p)})$$
$$x = x(z,\dot{z},\ldots,z^{(q)})$$
$$u = u(z,\dot{z},\ldots,z^{(q)})$$
$$z = \sum \alpha_i \psi^i(t)$$

Quasi-collocation

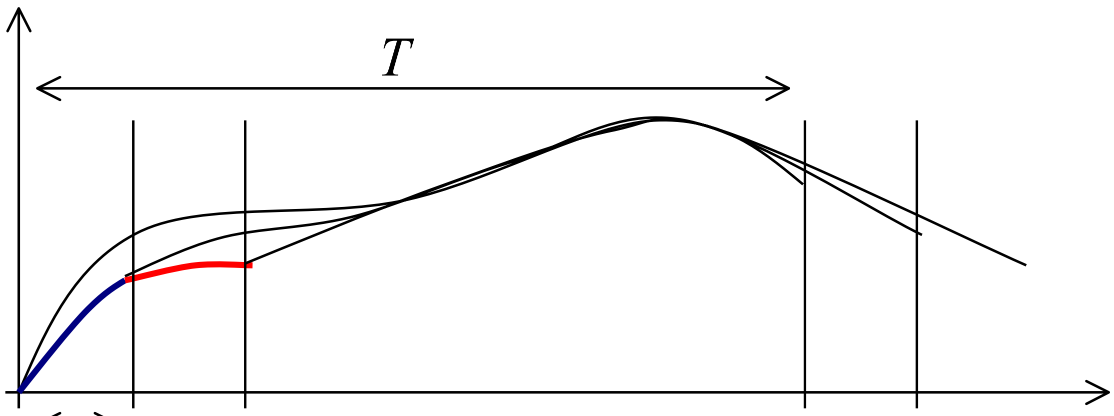
$$y = h(x)$$
$$(x,u) = \Gamma(y,\dot{y},\ldots,y^{(q)})$$
$$0 = \Phi(y,\dot{y},\ldots,y^{(p)})$$

Ducted Fan Terrain Avoidance





Optimization-Based Control: MPC + CLF



$$u_{[t,t+\Delta T]} = \arg\min \int_t^{t+T} q(x(\tau),u(\tau))\,d\tau + V(x(t+T))$$

$$x_0 = x(t) \quad x_f = x_d(t+T)$$

$$\dot{x} = f(x,u) \quad g(x,u) \le 0$$

Online control customization

- System: f(x,u)
- Constraints/environment: g(x,u)
- Misssion: L(x,u)

Update in real-time to achieve *reconfigurable* operation

Theory: MPC + CLF Approach

$$J_T^*(x_0) = \min_{u(\cdot)} \int_0^T q(x(\tau), u(\tau))\, d\tau + V(x(T))$$

Finite horizon

CLF
≈ "cost to go"

$$\inf_u \; (\dot{V} + q)(x, u) \;\leq\; 0$$

Basic Idea

- Use online models to compute receding horizon optimal control
- CLF-based terminal cost gives stability + short time horizons

Properties

- Can prove stability (in absence of constraints)
- Incremental improvement property ⇒ finite iterations OK
- Increased horizon ⇒ larger region of attraction

Experimental Results: Caltech Ducted Fan

NTG +
MPC +
CLF

Pitch
Control

dSPACE
RTOS

Multi-Vehicle Optimization-Based Control

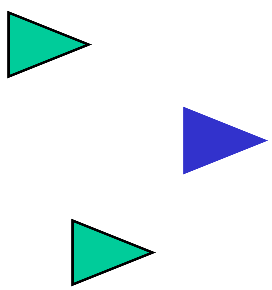
Assume we have real-time, finite horizon optimal control as a *primitive*

$$\left.\begin{aligned} u_{[t,t+\Delta T]} &= \arg\min \int_t^{t+T} L(x,u)\,dt + V(x(t+T)) \\ x_0 = x(t) &\quad x_f = x_d(t+T) \\ \dot{x} = f(x,u) &\quad g(x,u) \le 0 \end{aligned}\right\}$$

Choose f, g, L to represent the *coupling* between the various subsystems

Cooperation depends on how we model "rest of the world"

$$\ddot{q}_1 = 0 \quad |u_1| \le L$$

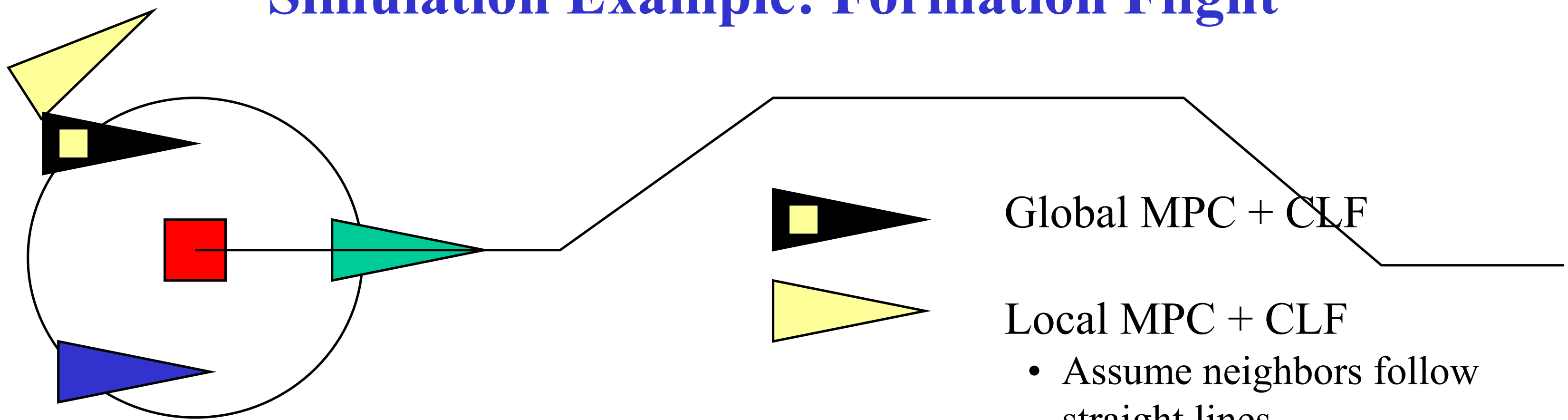


$$\boxed{\begin{aligned} u_{2,[t,t+\Delta T]} &= \arg\min \int_t^{t+T} L(q_1, x_2, q_3, u_2)\,dt + V \\ \dot{x}_2 = f(x_2, u_2) &\quad g(x_2, u_2) \le 0 \end{aligned}}$$

$$\ddot{q}_3 = 0 \quad |u_3| \le L$$

Reconfigurable based on condition, mission, environment

Simulation Example: Formation Flight



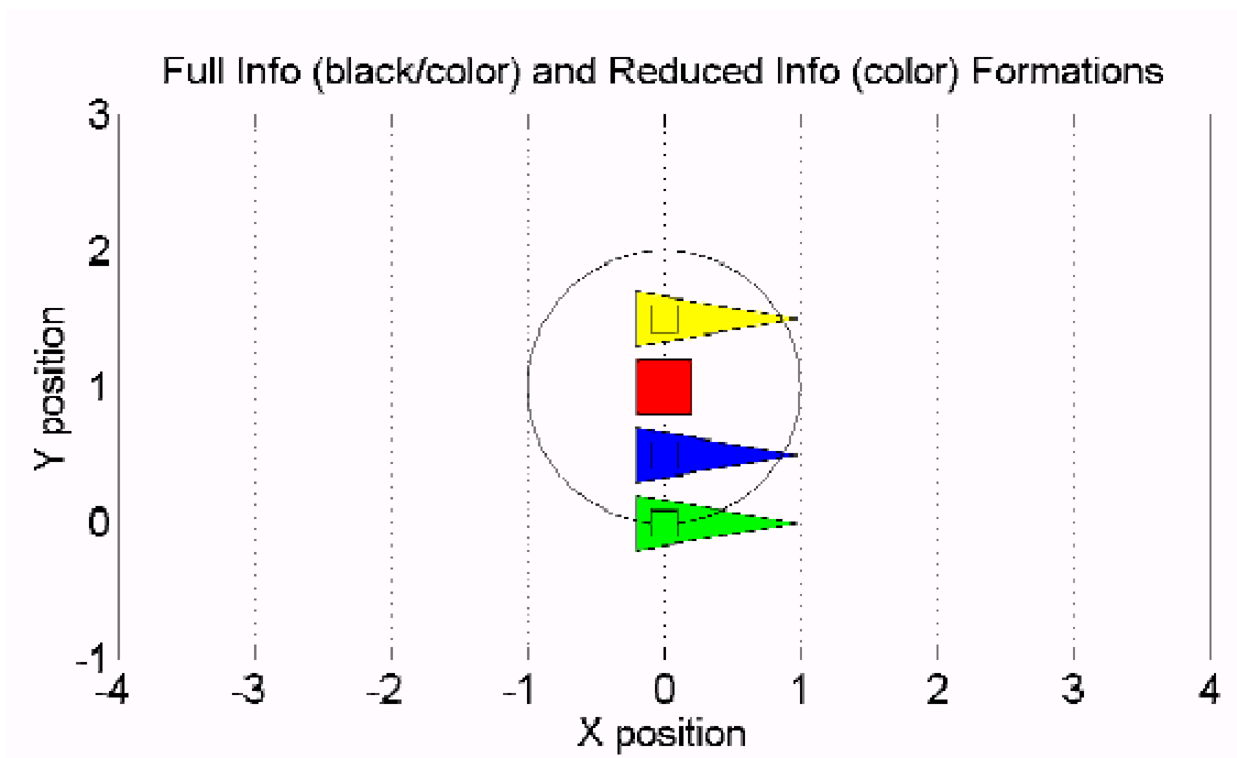
- Assume neighbors follow straight lines

Task:

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

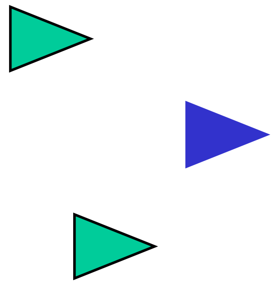
Parameters:

- Horizon: 2 sec
- Update: 0.5 sec
- **High damping**



Multi-Vehicle Optimization-Based Control

$$\ddot{q}_1 = 0 \quad |u_1| \leq L$$



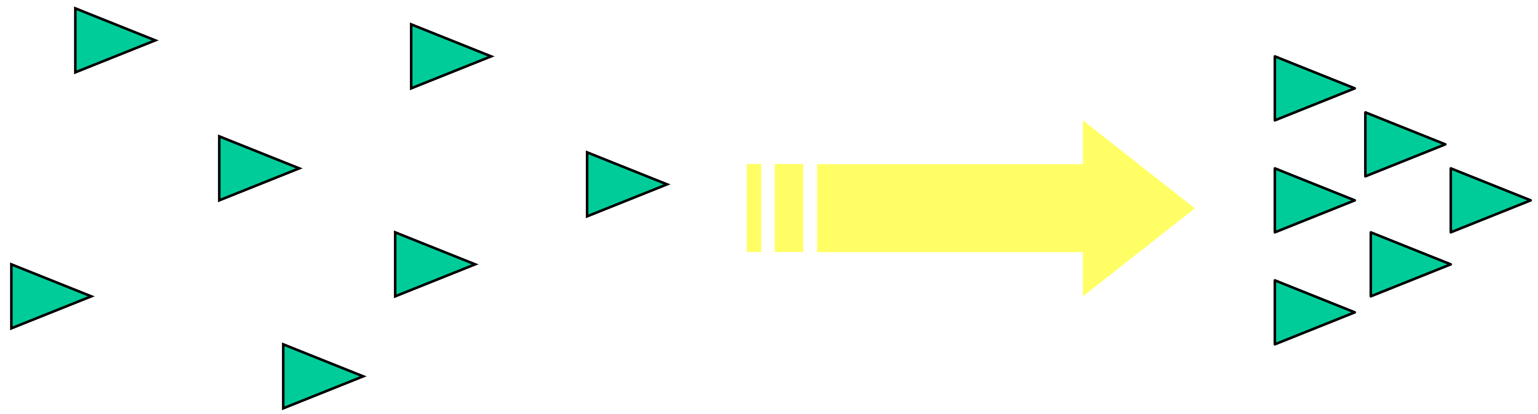
$$\boxed{\begin{array}{c} u_{2,[t,t+\Delta T]} = \arg\min \int_t^{t+T} L(q_1, x_2, q_3, u_2)dt + V \\ \dot{x}_2 = f(x_2, u_2) \quad g(x_2, u_2) \leq 0 \end{array}}$$

$$\ddot{q}_3 = 0 \quad |u_3| \leq L$$

Open Questions: "Low Level" Cooperation

- How do we coordinate motion between multiple vehicles?
- How do we aggregate cost functions into hierarchies?
- How do we provide redundancy and failure tolerance?
- How do we communicate between vehicles and how often?
- How do we insure scalability to large numbers of agents?
- How do we incorporate adversarial actions?

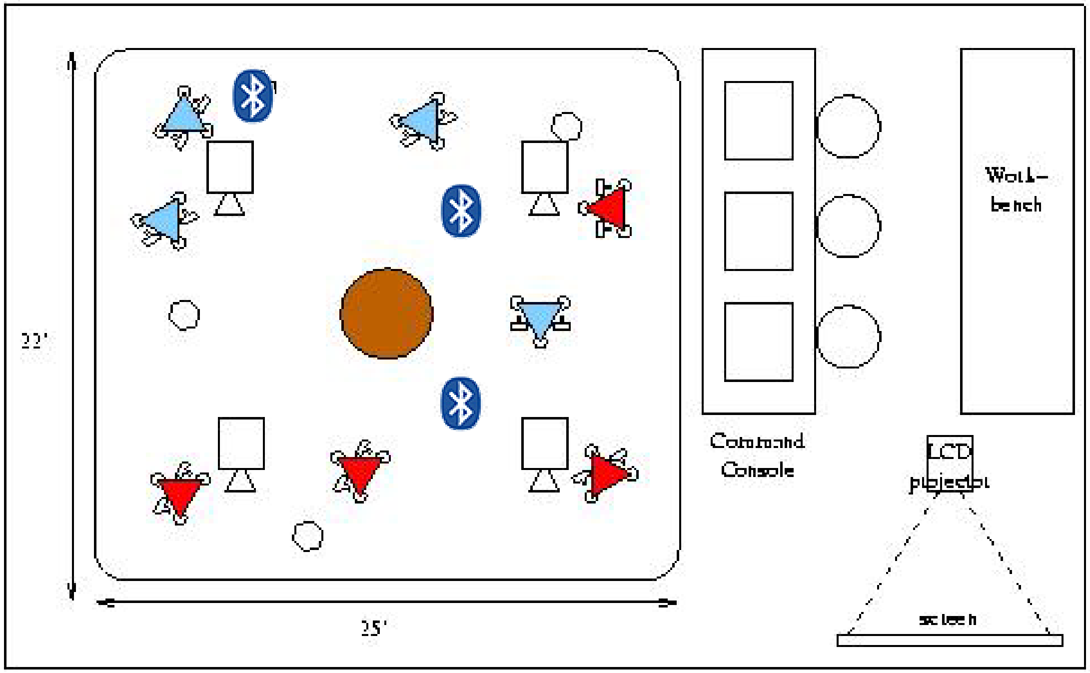
Higher Level Cooperation: Rejoin Exapmle

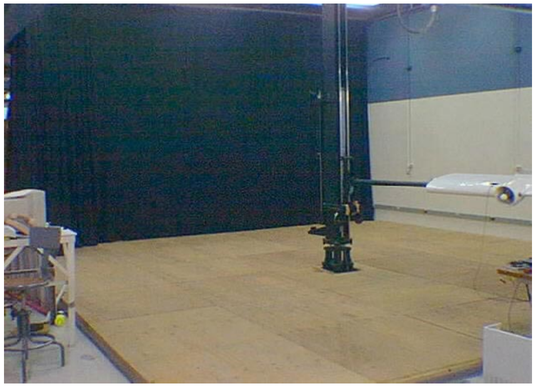


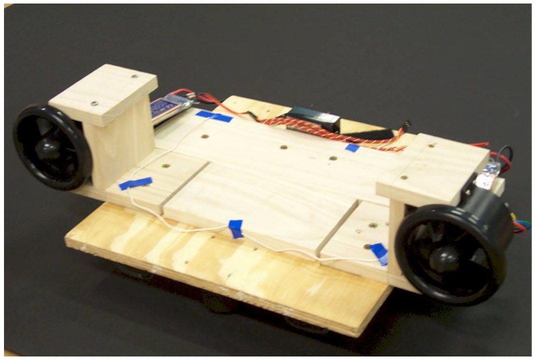
Open Questions: Higher Level Issues

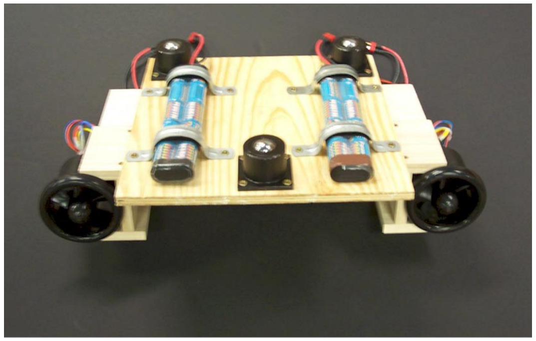
- How do we collectively agree to rejoin in a robust manner?
- How do we integrate “protocol stack” with “trajectory generation/tracking”
- How do we describe the “specification” of the task?
- How do we prove that solution (code) satisfies the specification?
- How do we prove “stability” of the solution?
- How do we verify and validate the solution?
- How do we insure all of this works in the presence of adversaries?

Multi-Vehicle Wireless Testbed for Integrated Control, Communications and Computation (DURIP)









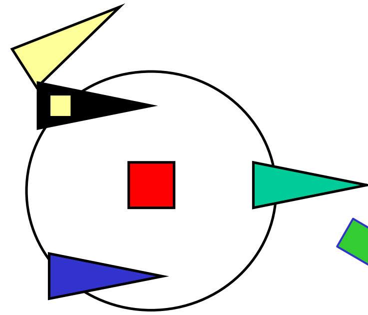
Testbed features

- Distributed computation on individual vehicles + command and control console
- Point to point, ad-hoc networking (bluetooth) + local area networking (802.11)
- Cooperative control in dynamic, uncertain, and adversarial environments

Research Plan and Integration: Cooperative Control in Dynamic, Uncertain and Adversarial Environments

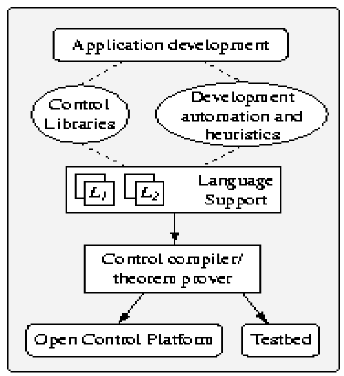
Optimization-Based Control

- Real-time model predictive control for online control customization: theory and software
- Online implementation on Caltech Ducted Fan



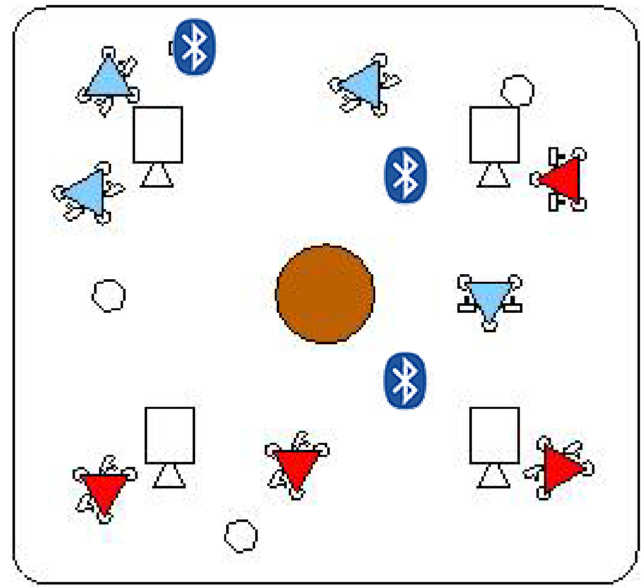
Cooperative Control

- Linked cost functions



Software Environments

- Logical programming environments for embedded control systems design



Multi-Vehicle Testbed (DURIP)

- Implementation on multi-vehicle, wireless testbed using Open Control Platform
- Bluetooth-based point to point communications with ad-hoc networking