Collaborative Nets

Eduardo Camponogara and Sarosh Talukdar

Institute for Complex Engineered Systems
Carnegie Mellon University

CDNA Project Meeting
Pittsburgh, October 1999

Controlling Large Networks

Operating goals fall into categories:

- Costs & profits
- Safety
- Regulations
- Equipment Limits

Limitations:

- No organization can cope with all operating goals
- Need of diverse skills
- Multitudes of agents

Control Solution:

- Delegate goals to separate organizations

Organization: A network of agents and communication links.

Agent: Any entity that makes and implements decisions such as relays, control devices, and humans.

Multiple Organizations in the Power Grid

	Protection Systems	Generator Control	Security Systems
Agents	Relays	Governors, exciters optimization soft.	Simulation & learn. tools, humans
Goals	Keep equipment under limits	Reduce cost s.t. constraints	Prevent cascading failures
Reaction Time	0.01 to 0.1secs	Seconds	Hours, days

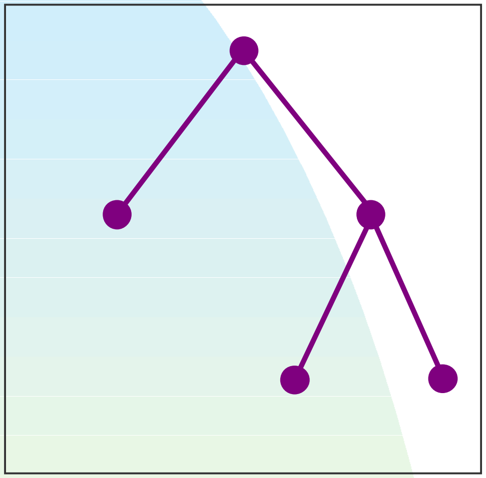
Low ⇐ Agent Skills ⇒ High

Large ⇐ Number of Agents ⇒ Small

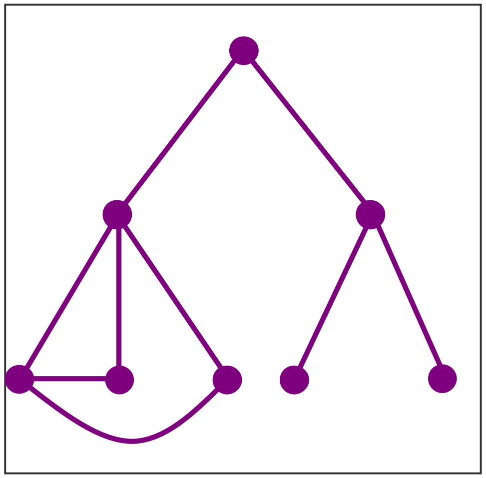
Fast ⇐ Agent Speed ⇒ Slow

Organizations Do Not Collaborate

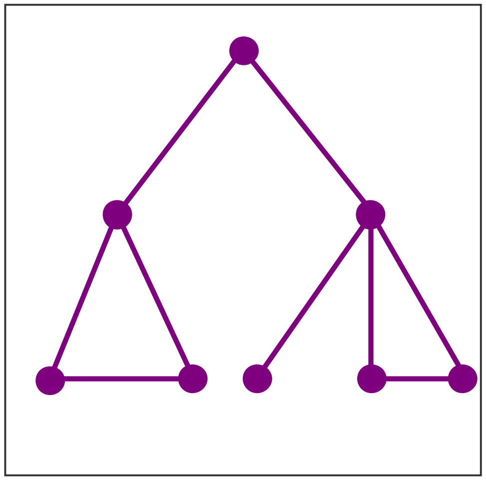
Protection Systems



Generator Control



Security Systems



Current Scenario:

- Agents in separate organizations do not “talk”
- Agents might work at cross-purpose
- Organizations might interfere with one another

How do we make individual agents more effective?

How do we prevent interference between organizations?

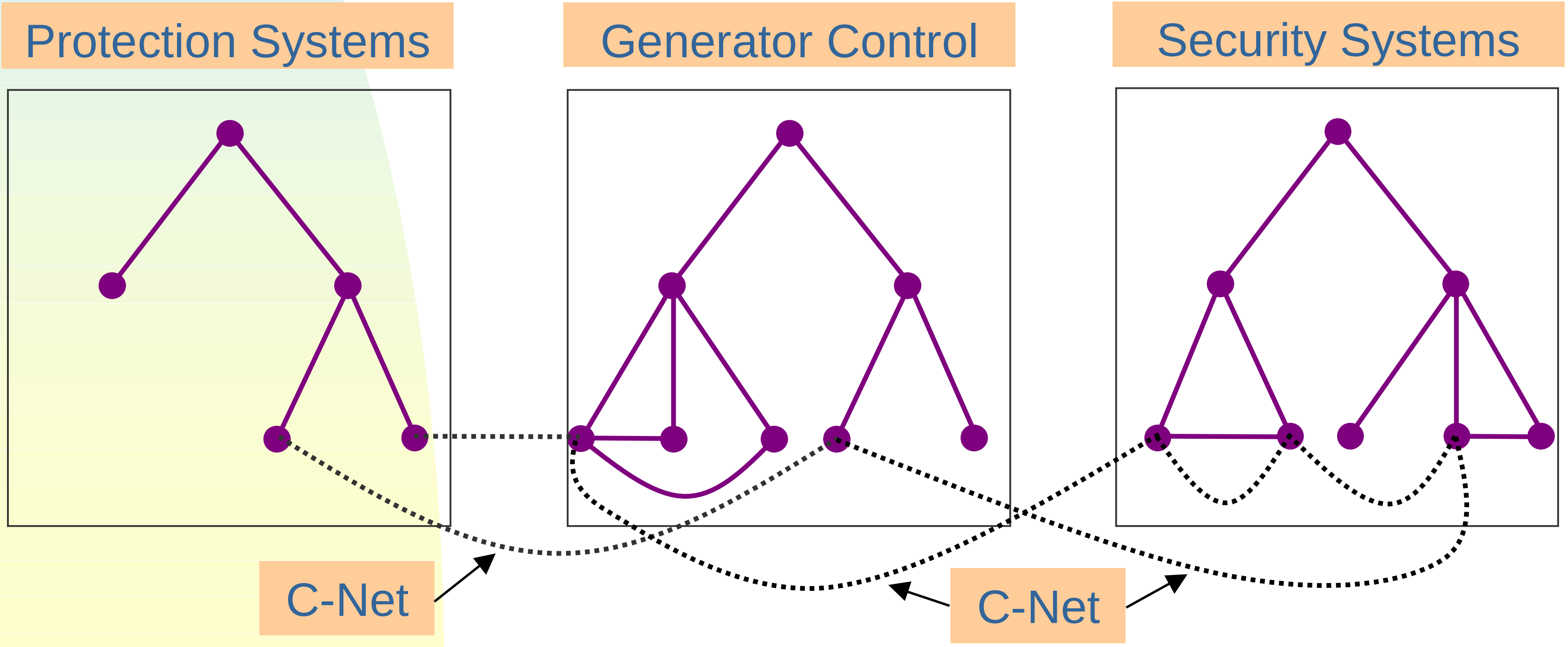
Improving Overall Performance of Nets

The suggested answer is based on:

1.) The use of a common framework to specify agent tasks.

2.) The implementation of a sparse, collaborative net that can cut across the hierarchic organizations.

3.) The design of collaboration protocols to promote effective exchange of information.



Protection Systems

Generator Control

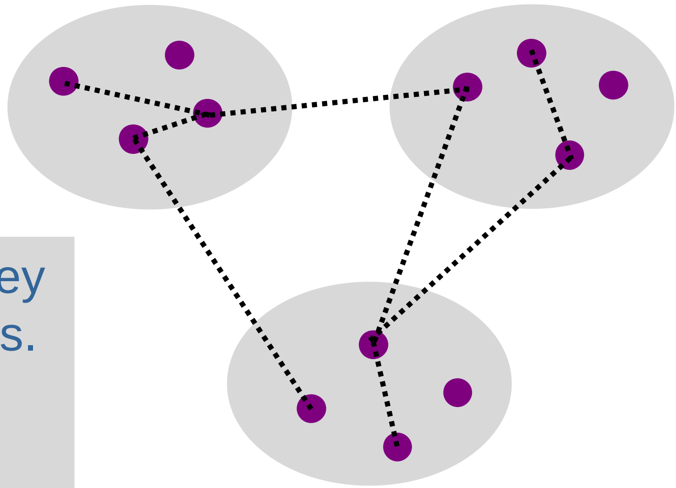
Security Systems

C-Net

C-Net

What Is A Collaborative Net?

A flat organization of dissimilar agents that can integrate hierarchic organizations.



Properties:

- Agents are autonomous within the C-Net. They have initiative, make and implement decisions.
- Agents collaborate with their neighbors.

  The collaboration protocol determines:
  - what information is exchanged,
  - in which way, and
  - how agents make use of it.

Advantages:

- Quick
- Fault Tolerant
- Open

Disadvantages:

- No structural coordination. If necessary, it can emerge from the collaboration protocol.
- Unfamiliar.

The Rolling Horizon Formulation

A framework to solve dynamic control problems as a series of static optimization problems.

The dynamic control problem

Minimize $f(x, dx/dt, u, t)$

Subject to $h(x, dx/dt, u, t) = 0$

$g(x, dx/dt, u, t) <= 0$

The steps of the rolling horizon formulation:

1.) Choose a horizon $[t_0,..,t_N]$, i.e. a set of time points where t_0 is the current time.

2.) Let $x(t_n)$ be the state predicted at time t_n.

x(t_0) is the current state.

3.) Let $u(t_n)$ be the planned actions at time t_n.

4.) Let X=$[x(t_0),\ldots,x(t_N)]$ and U=$[u(t_0),\ldots,u(t_N)]$

5.) Choose a model to predict $x(t_{n+1})$ from $x(t_n)$ and $u(t_n)$. Possibly, a discrete approximation of the dynamic equations (e.g., Euler's step).

The static opt. problem (P)

Minimize $f(X,U)$

Subject to $H(X,U) = 0$

$G(X,U) <= 0$

The prediction model is embedded in $H(X,U)$.

$G(X,U)$ approximates the operating constraints in $g(x, dx/dt, u, t)$.

The Rolling Horizon Algorithm

- A model is used to predict the future state of the physical network over a set of discrete points in time (horizon).
- An optimization procedure computes the control actions, over the horizon, that minimize “error.”

Steps of the Algorithm:

1.) The current time is t_0.

2.) Sense the current state $x(t_0)$

3.) Instantiate the static optimization problem (P).

4.) Solve (P) to obtain the control actions $U=[u(t_0),\dots,u(t_N)]$.

5.) Implement the control action $u(t_0)$.

6.) Pause and let the physical network progress in time. The horizon “rolls” forward.

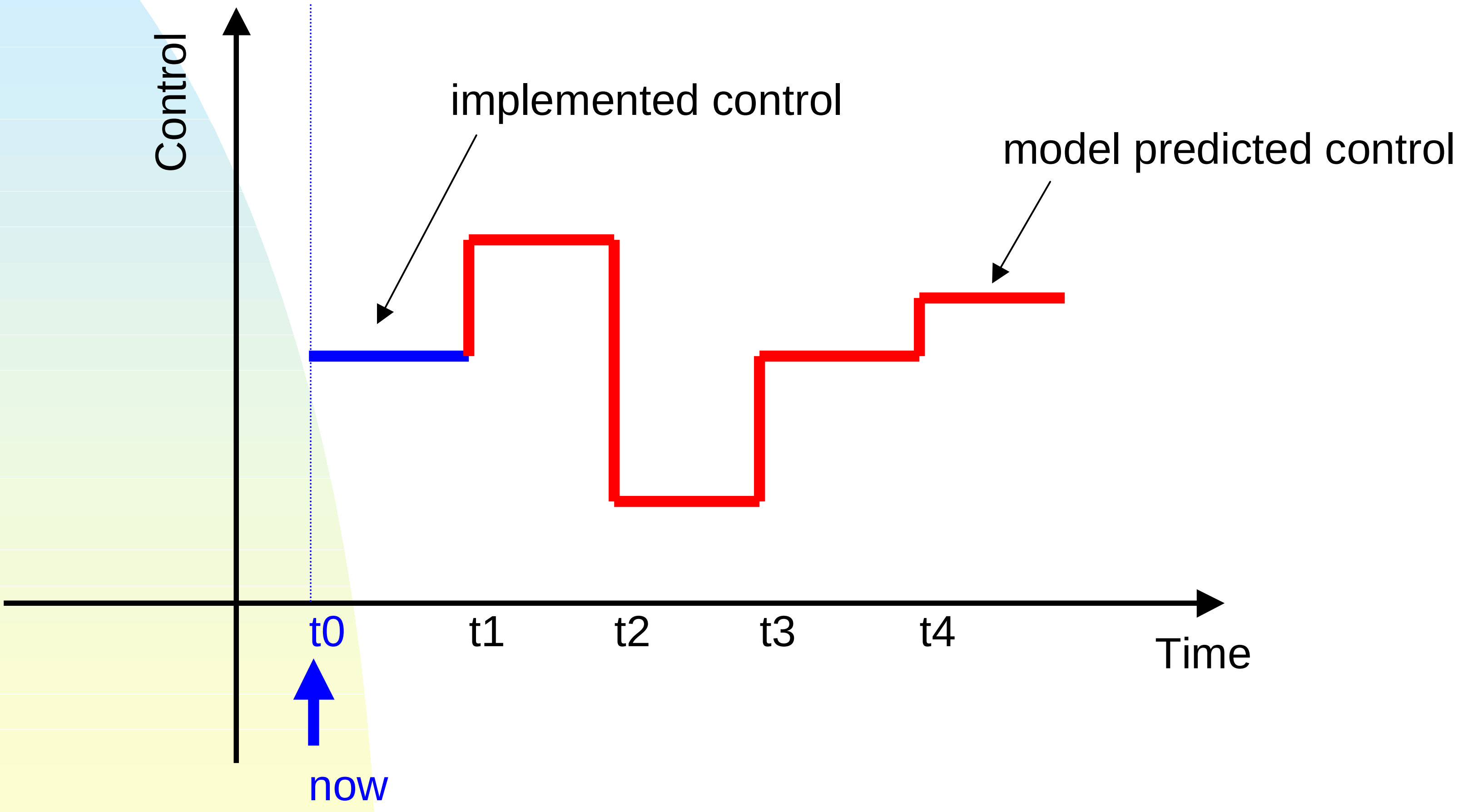
7.) Repeat from step 1.

Design Issues:

- The horizon has to be long enough to avoid present actions with poor long-term effects.
- Accuracy of the prediction model.

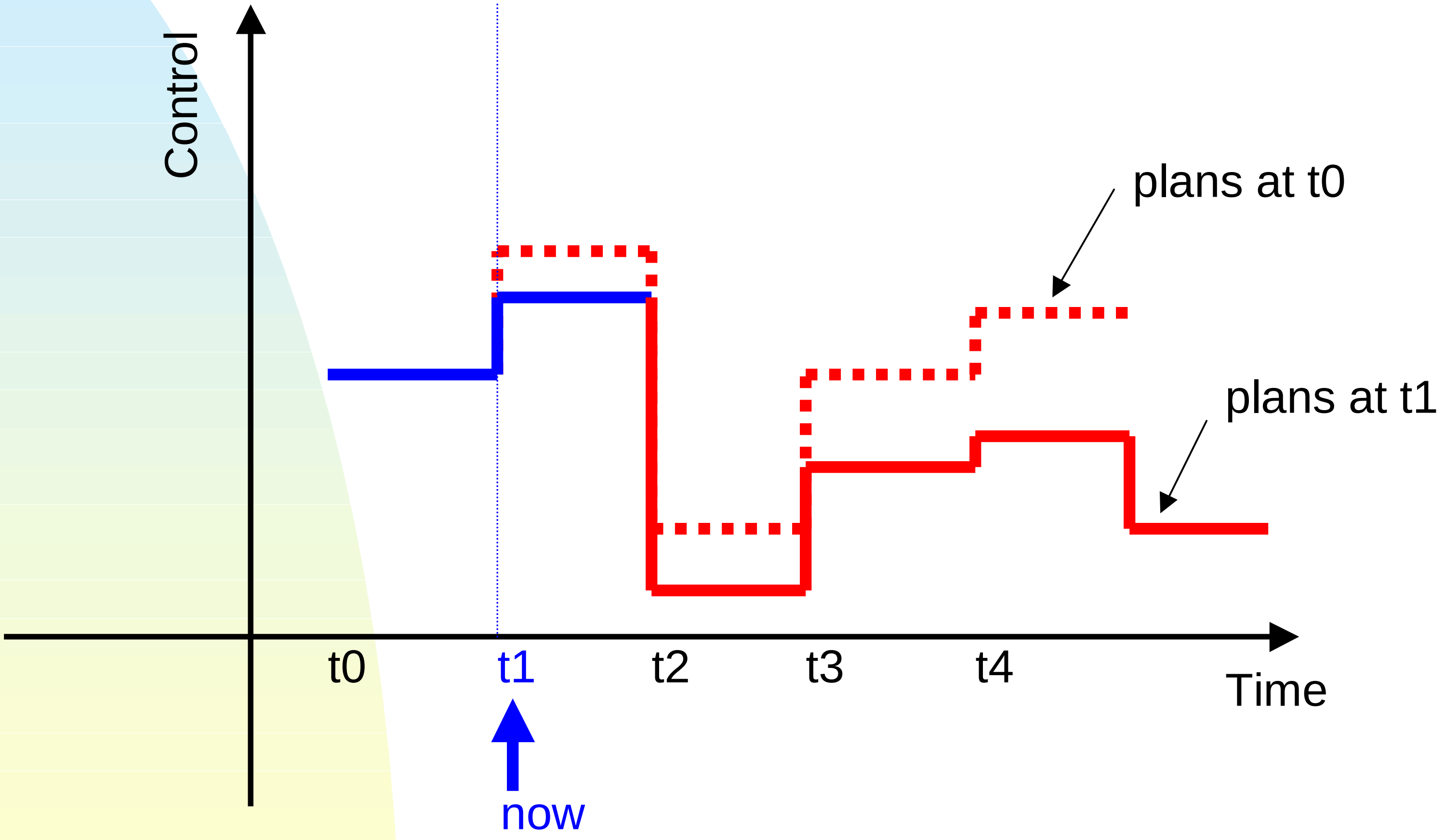
The Rolling Horizon

Plan ahead



The Rolling Horizon

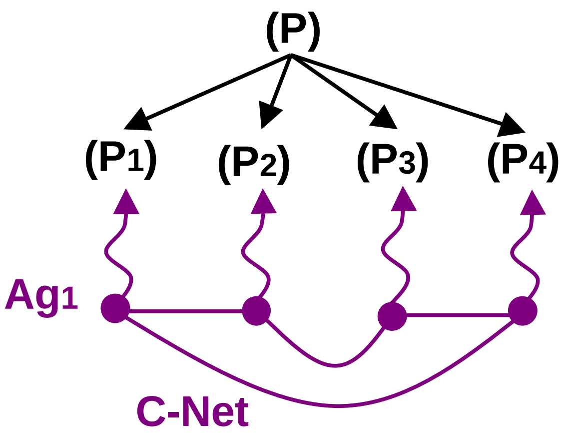
Update plans frequently



A Framework for Specifying Agent Tasks

Break up the static optimization problem, (P), into a set of M small, localized subproblems, {(P_m)}.

Assemble M agents into a C-Net, so that each agent matches one subproblem.



Agent-m and its subproblem (P_m)

It has partial perception of, and limited authority over, the physical network.

Proximate variables (x_m,u_m):
It senses the values of a subset x_m of x.
It sets the values of a subset u_m of u.

Neighborhood variables (y_m)
Variables sensed or set by neighbors.

Remote variables (z_m):
All the other variables.

Matching Agents to Subproblems

The rolling horizon formulation of (P_m)

Minimize $f_m(X_m, U_m, Y_m, Z_m)$

Subject to $H_m(X_m, U_m, Y_m, Z_m) = 0$

$G_m(X_m, U_m, Y_m, Z_m) <= 0$

The matching between agent-m and its subproblem (P_m)

Exact: If it is not sensitive to remote vars.

$f_m = f_m(X_m, U_m, Y_m)$

$H_m = H_m(X_m, U_m, Y_m)$

$G_m = G_m(X_m, U_m, Y_m)$

Near: If it is weakly sensitive to remote vars.

Collaboration Protocols

A protocol prescribes: a) the data exchanged by agents,
b) in which way, and
c) how agents use the data to solve their problems.

Two protocols

Voting

In setting the values of its controls, each agent takes the votes of its neighbors into account.

Proximate Exchange

Each agent broadcasts its plans to nearby agents which, in turn, take these plans into account.

Versions

- Semi-synchronous, semi-parallel (mutual help).
  - Synchronization between neighbors.
  - Parallel work if agents are non-neighbors.
- Asynchronous, parallel.

Equivalence and Convergence

Two Questions:

Equivalence:

When are the solutions to the network of subproblems, $\{(P_m)\}$, solutions to (P)?

Convergence:

When does the effort of the collaborative agents converge to a solution of $\{(P_m)\}$?

Sufficient conditions for equivalence and convergence:

1.) Coverage: The C-Net must provide complete coverage of the network.

2.) Density: The matching of agents to subproblems must be exact.

3.) Convexity: (P) must be convex.

4.) Feasibility: (P) must be strictly feasible.

5.) Int-Pt-Mtd: The agents must use an interior-point-method.

6.) Serial Work: The agents run the semi-synchronous, semi-parallel protocol.

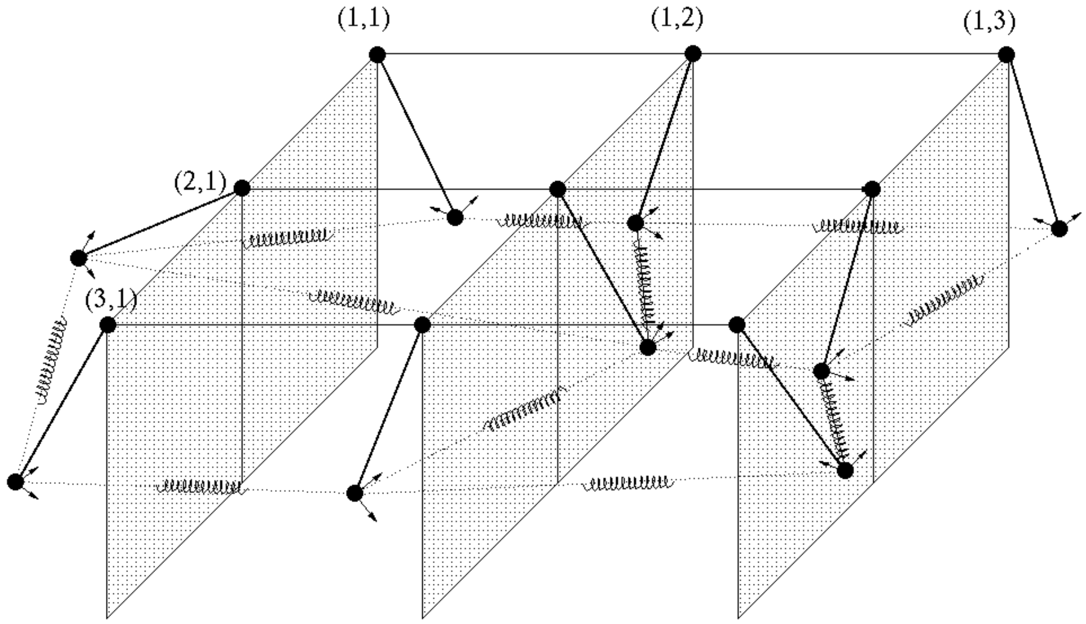
Relaxing Sufficient Conditions in Practice

We believe that the following conditions can be relaxed in practice:

1.) Density:	Near matching of agents to problems are likely to be adequate.
2.) Convexity:	It is impractical in real-world networks.
3.) Serial work:	Serial work within a neighborhood is too slow.

A prototypical network: A forest of pendulums.

- One agent at each pend.
- Agents control two forces:
  Horizontal & Orthogonal.
- Agents collaborate with
  nearest neighbors.



The Dynamic Control Problem

Problem: Drive pendulums to the pre-disturbance mode, that is, minimize cumulative error (from desired trajectory) and total control-input cost.

Minimize

$$f(x,u) = \int_{t=0}^{t=\infty} \|x-\tilde{x}\|^2 dt + b \int_{t=0}^{t=\infty} \|u\|^2 dt$$

Subject to

$$h(x,\dot{x},u) = 0$$

Three Control Solutions:

- **C1** A centralized, nonlinear optimization package to solve the stat. opt. prob. (P).
- **C2** A centralized, feedback linearization controller.
- **C-Net** A collaborative net, with one agent at each pendulum, that solves {(P$_m$)}.

C-Net and C1: Experimental Set-up

Goal: Evaluate the loss in quality of the Collaborative Net solution.

Set-up: C-Nets and C1s restore synchronous mode of pendulums.

At each sample time t,

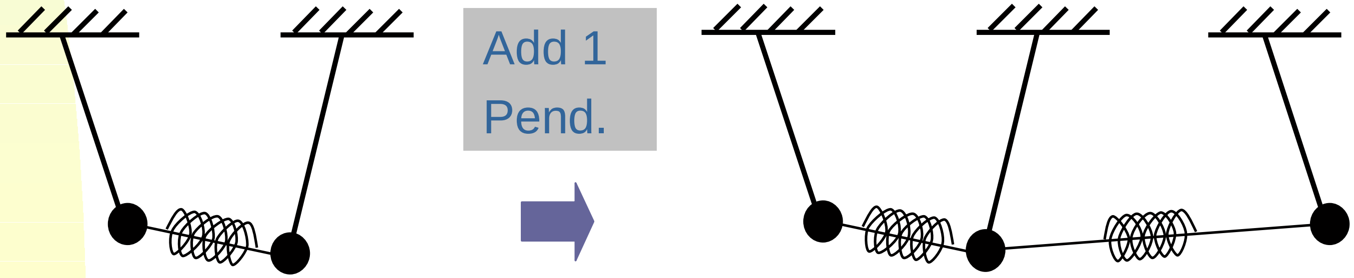
1.) solve the network of subproblems, $\{(P_m)\}$, with the C-Net,
2.) record the obj-function evaluation of the C-Net, F(C-Net),
3.) solve the static optimization problem, (P), with C1, and
4.) record the obj-function evaluation of C1, F(C1).

Output Data: A list of obj-function-evaluation pairs [F(C-Net),F(C1)].

Scenarios: Place pendulums in a line to form forests of 2 to 9 pendulums.

2-Pendulum Forest

3-Pendulum Forest



Add 1 Pend.

C-Net and C1: Results

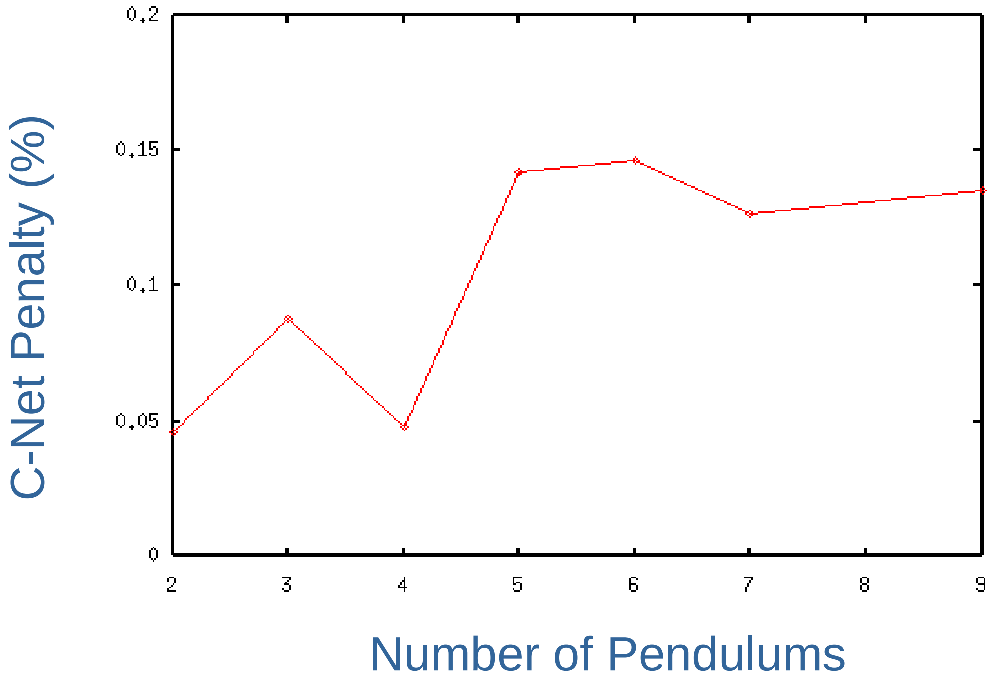
C-Net Excess: The difference in quality between the C-Net and C1 solutions.

F(C-Net) is the obj-function evaluation attained by the C-Net.
F(C1) is the obj-function evaluation attained by controller C1.

C-Net excess = [F(C-Net) – F(C1)] / F(C1)

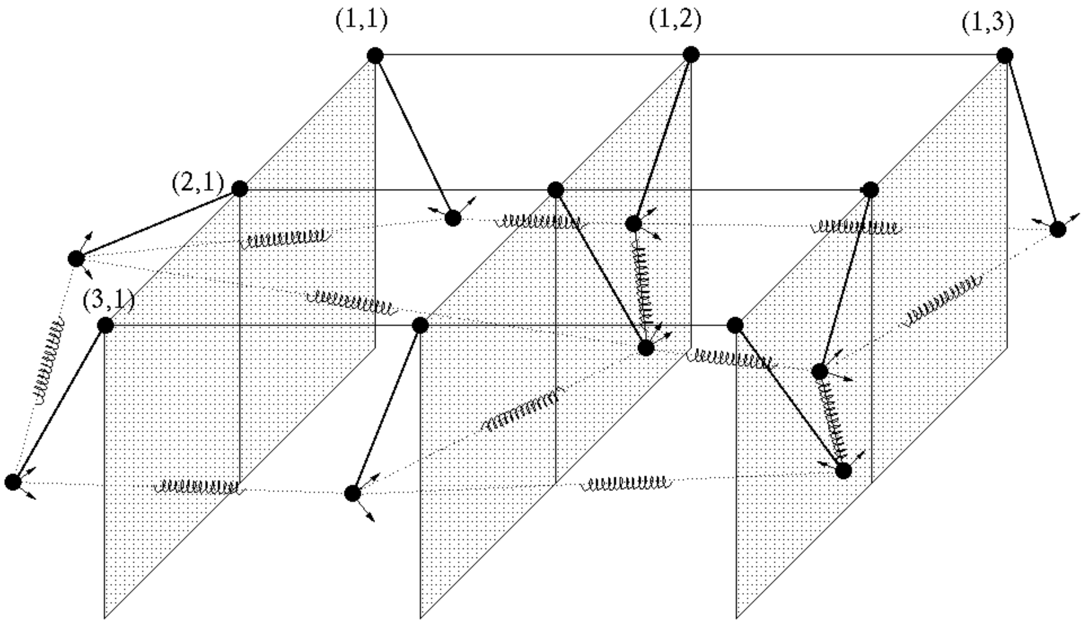
C-Net Penalty: The mean value of the C-Net excess.

C-Net penalty is low



C-Net and C2: Experimental Set-up

Goal:	Evaluate the performance of the C-Net and the feedback linearization controller, C2, a traditional control technique.
Set-up:	C-Net and C2 restore synchronous mode of pendulums.
Output Data:	The cumulative error and input-cost, *f*(*x*,*u*), for the C-Net & C2.
Scenario:	A forest with 9 pendulums placed in a uniform grid.



C-Net and C2: Results

Objective:

Minimize $$f(x,u) = \int_{t=0}^{t=\infty} \left\| x - \tilde{x} \right\|^2 dt + b \int_{t=0}^{t=\infty} \left\| u \right\|^2 dt$$

Control-Input Cost (*b*)	Objective Function Evaluation: *f*(*x*,*u*)	
	C2 (feedback lin)	C-Net
10e-4	9.56	11.89
10e-3	10.49	12.32
10e-2	17.05	16.00
10e-1	82.64	32.07

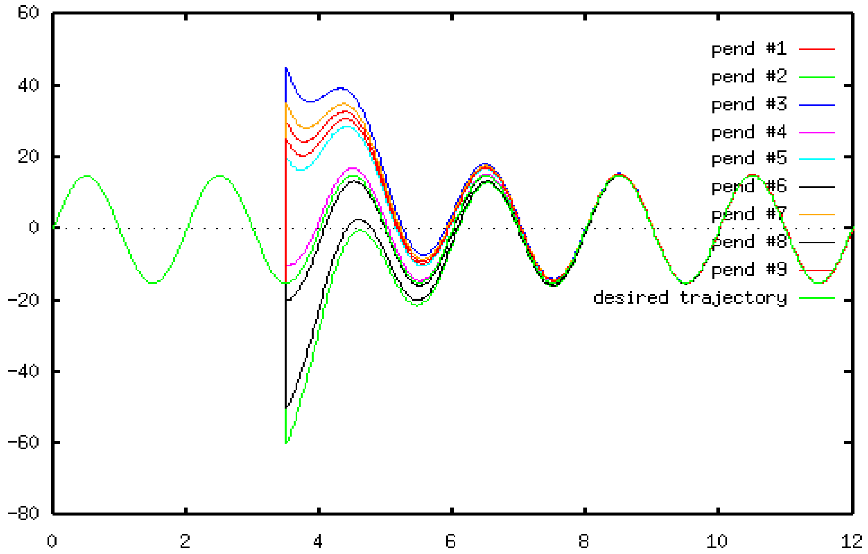
The lower the *f*(*x*,*u*), the better the solution

C-Net performance improves

C-Net and C2: Trajectory of Pendulums

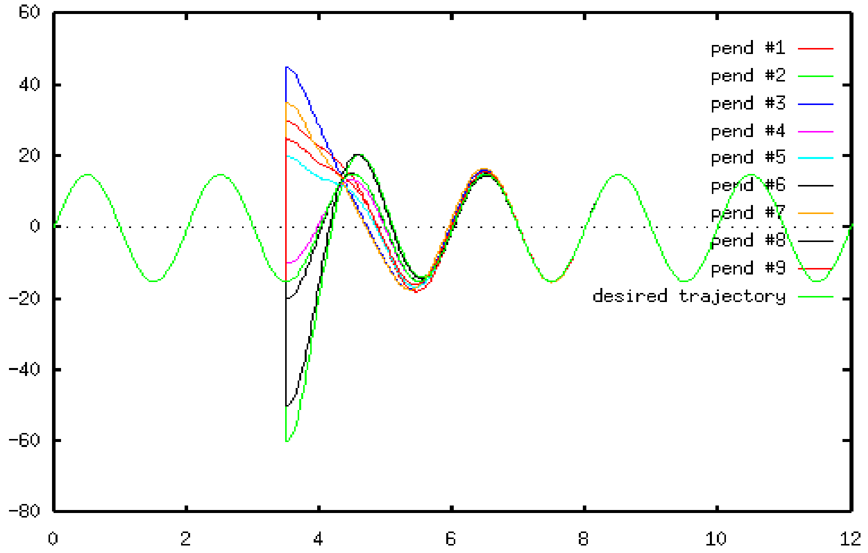
Pendulums under control of C2 (feedback linearization)

C2 immediately drives the pendulums to the desired trajectory.



Pendulums under control of the C-Net

The C-Net waits until it becomes cheaper to drive pendulums.



Conclusions and Future Work

The experiments show that C-Nets are promising.

Current research effort:

- Development of collaboration protocols that allow agents to work asynchronously and in parallel, at their own speed.
  - Use of safety margins to guarantee feasibility, and foster effective work between slow and fast agents.
- A taxonomy of collaboration protocols.

What else have we done?

- Employed C-Nets to recover synchronous operation of generators in power networks IEEE-14, -30, -57.
- Preliminary work on the decomposition of (P) into $\{(P_m)\}$:
  - Models and algorithms to specify “neighborhood” perception.