

e-Merge-ANT: December 2001

Stephen Fitzpatrick & Cordell Green (PIs)
Lambert Meertens

Kestrel Institute

<http://ants.kestrel.edu/>

ANTS PI Meeting, Scottsdale, AZ, 17-19 December 2001



Outline

- Project Summary
- Challenge Problem Year 2
 - metrics to guide coordination
 - localization
 - implementation details
 - experimental results
- Dynamics of distributed constraint optimization
 - asynchronous algorithm
 - dense graphs
- Plans

Project Summary

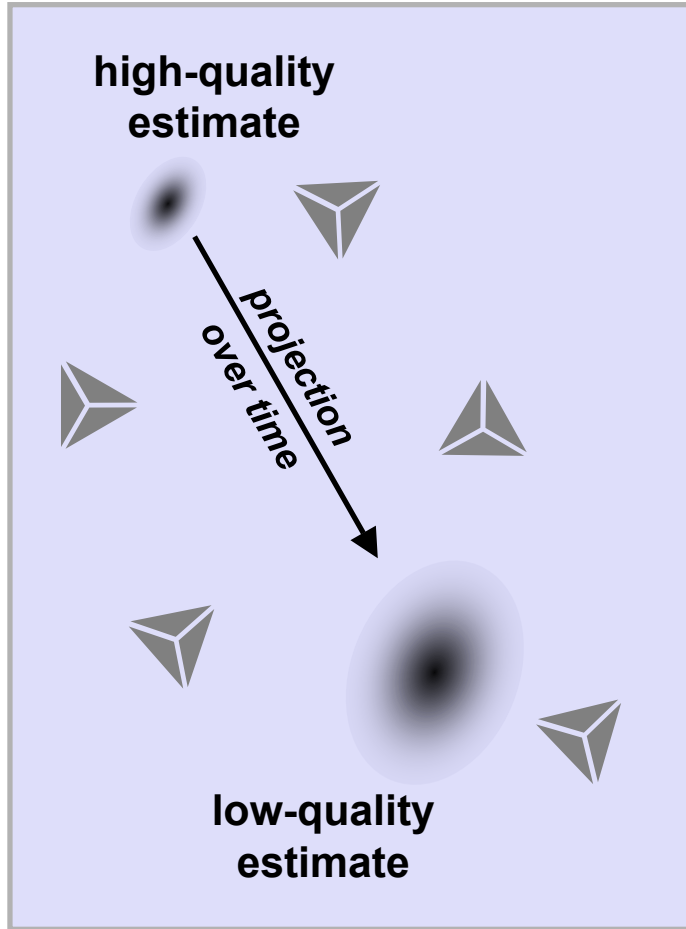
Year 1

- Formulated challenge problem as a distributed constraint optimization problem
 - isomorphic to graph coloring
- Developed distributed, scalable, anytime scheduler
 - soft graph colorer based on an iterative, local-repair algorithm
- Demonstrated scheduler on simulator & hardware
 - single target

Year 2

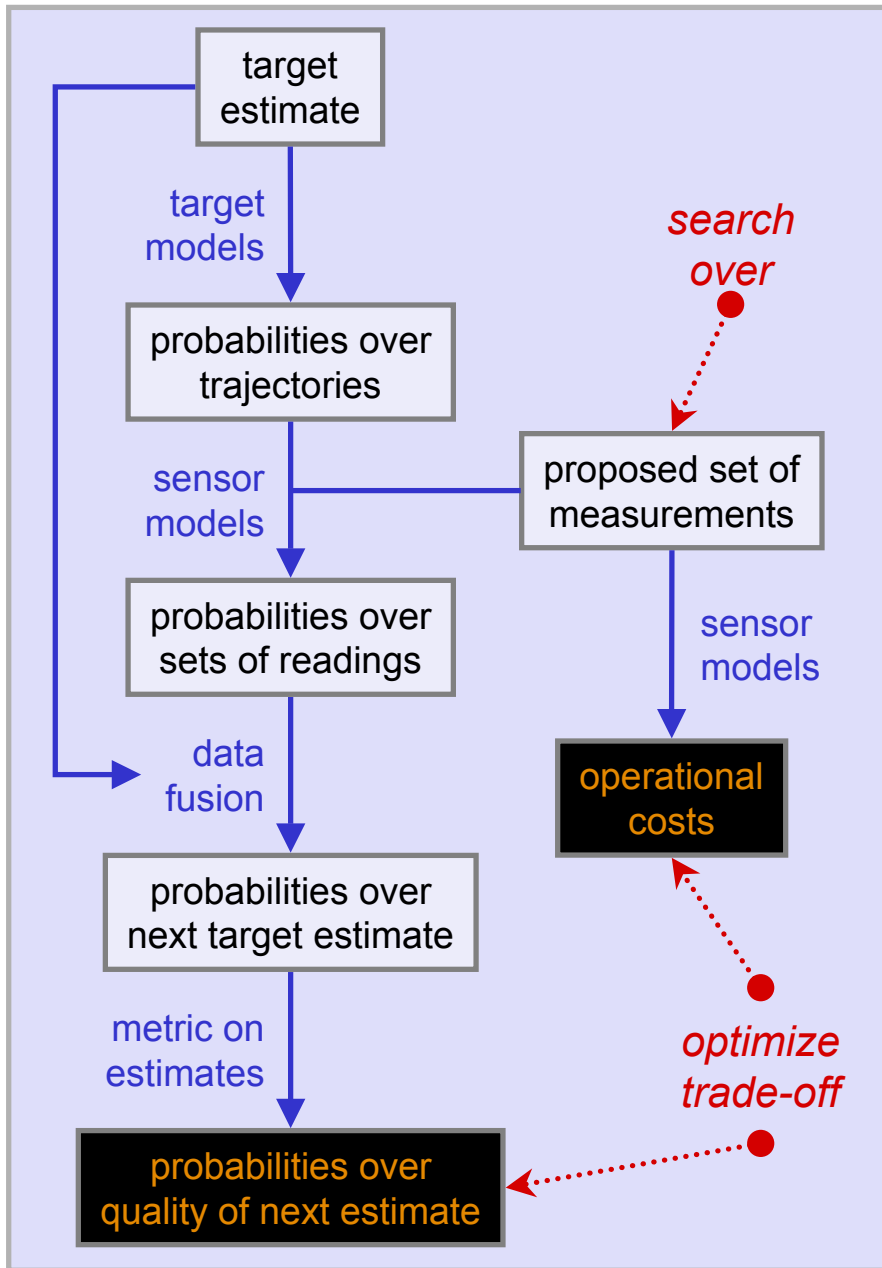
- Generalized formulation of challenge problem
 - more complex constraints/metrics allow more realistic representation of objectives
- Evaluated scheduler's performance on abstract graph coloring problems
 - scalable, low cost, robust
- Improved performance of distributed scheduler
 - simple stochastic component breaks symmetry to ensure convergence in parallel computations
- Demonstrated scheduler on simulator & hardware
 - multiple targets

Sensor Coordination



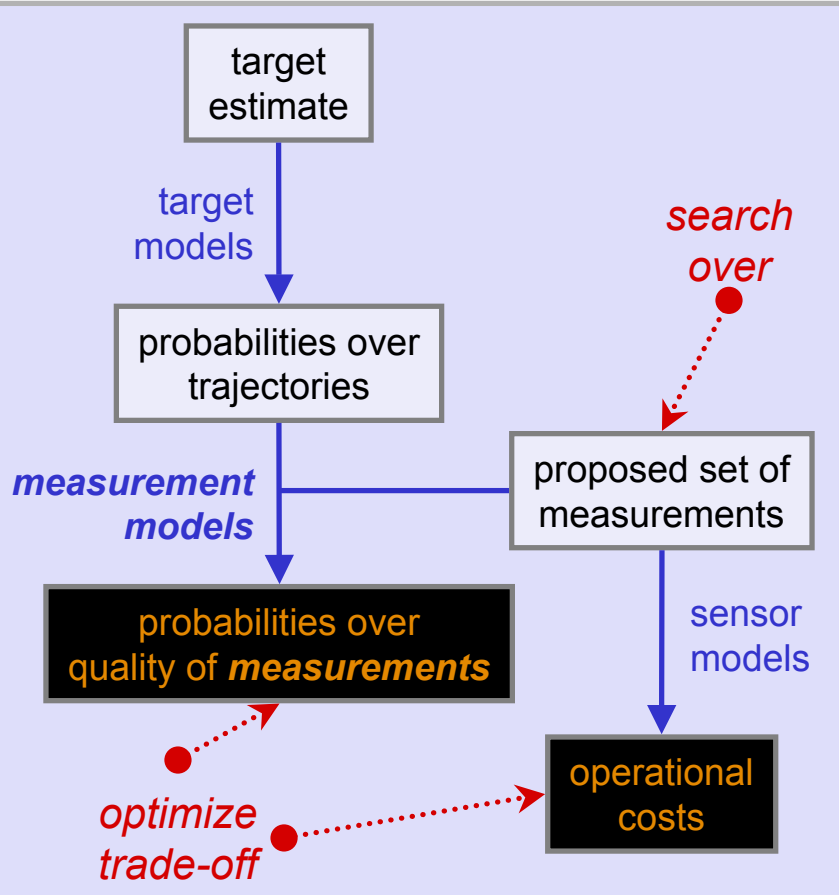
- A *target estimate* represents approximate knowledge about a target
 - probability density function over space of position \times velocity
- The *quality* of an estimate reflects its accuracy
 - e.g., standard deviation for normal distributions
- A *target model* predicts a target's future from an estimate
 - probability density function over *trajectories*
 - quality of predictions decreases further into the future
- Estimate quality is maintained by incorporating new measurements
- *Coordination* attempts to optimize the trade-off between quality of estimates and operational costs

Coordination based on Quality of Estimates



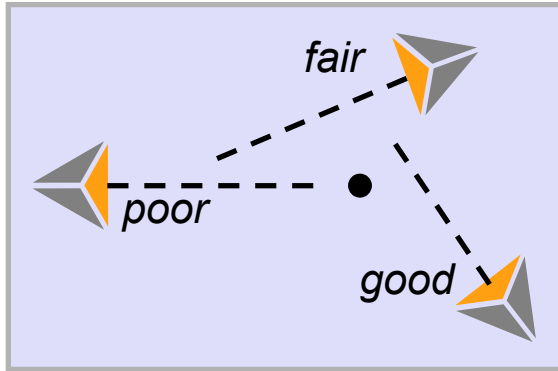
- Given a proposed set of measurements
 - determine expected quality of next estimate
 - determine costs
- Search over sets of measurements
 - optimize expected quality-cost trade-off
- Scalable
 - due to locality of sensor interactions
- But expensive!
 - large search spaces
 - expensive processes at each search node
 - not feasible for real-time, distributed coordination (yet)
- Won't work for BAE tracker
 - no quality metric available

Coordination based on Quality of Measurements



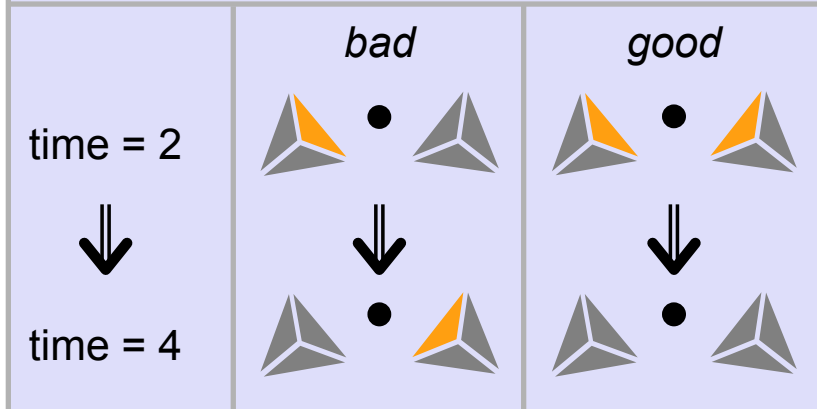
- Heuristic measurement models
 - determine quality of proposed set of measurements with respect to a trajectory
 - details on next slide
 - high-quality measurements assumed to lead to high-quality estimates
- Optimize trade-off between expected measurement quality and costs
- Much cheaper than using quality of estimate
 - presumably not as accurate
 - but will work for BAE tracker
 - no need for metric on estimate quality
 - BAE tracker gives most-likely trajectory

Measurement Metrics (for single target)



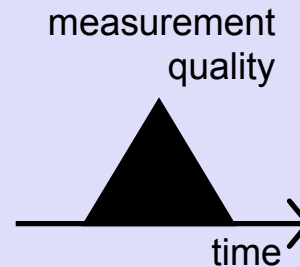
- Quality of single measurement derived from signal equation:
 - $s(R, \theta) = K \exp(\theta^2/A)/R^2$
 - $m(s) = \log_2[1 + \max(0, (s - s_b)/(s_m - s_b))]$
 - s_m, s_b = maximum, background signals

Simultaneity



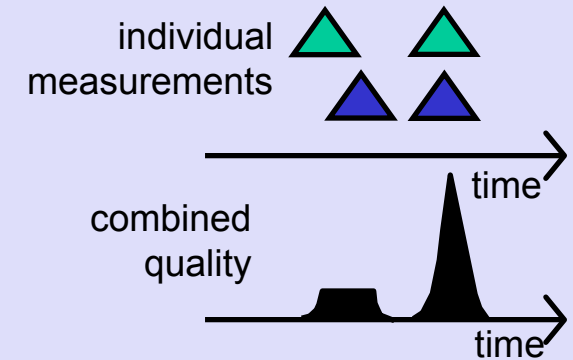
- Quality of multiple measurements reflects simultaneity

Persistence



persistence function associates a time window with a measurement

Adhesion



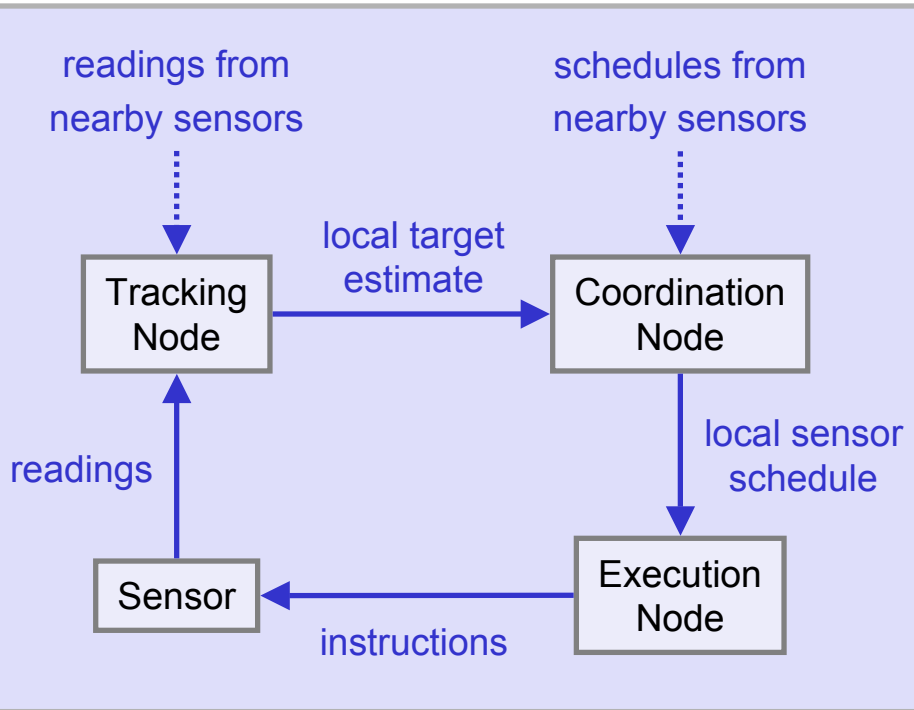
adhesion function combines measurements at every instant based on time windows

- Overall quality with respect to a trajectory
 - integral over time of instantaneous, combined quality

Multiple Targets

- Extend single-target concepts to multiple targets
 - a world state is a finite map from targets to single-target information
 - an estimate is a probability density function over world states
 - a trajectory is a timed sequence of world states
- A measurement may give information about any subset of the targets
 - a quality metric is a finite map from targets to single-target quality metrics
- Interference between targets is possible
 - for the challenge problem hardware $m_G(g) \equiv \max[0, m(g) - \sum_{g' \neq g} m(g')]$
 - where G is the set of all targets and $g, g' \in G$
- Multiple measurements are combined by combining the metrics for each target separately
 - the persistence and adhesion functions are lifted to finite maps

Localization



tightly-coupled: separate threads on same JVM

- Fully distributed, homogeneous architecture
 - scalable, robust
 - each sensor has local tracking, coordination and execution nodes
- Coordination occurs via exchange of *schedules*
 - each sensor independently executes its own schedule based on its local, synchronized clock
- Communication latency finessed
 - inter-sensor communication is infrequent
- Adaptation via continual rescheduling
 - convergence? ...
- Local coordination metrics
 - assume communication is possible where collaboration is useful

Coordination Nodes

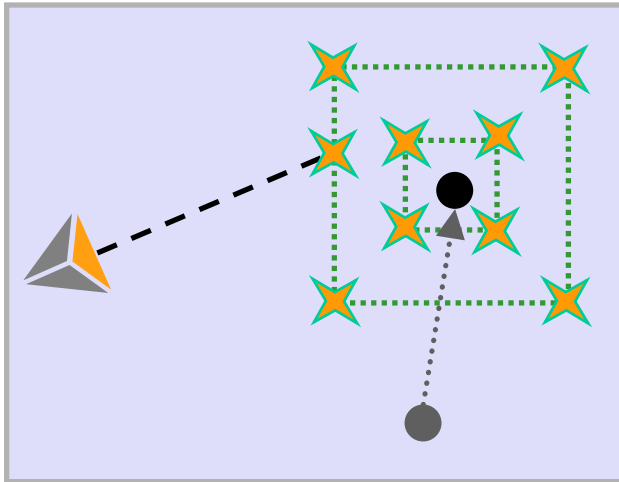
- Stochastic activation
 - periodically, each node randomly decides if it should activate
 - the *activation probability* determines the (mean) fraction of nodes activated
- Local schedule optimized by each activated node
 - given the current, local target estimate;
 - given the schedules that it has received from nearby sensors;
 - it computes a schedule of actions for its sensor
 - optimizes the trade-off between measurement quality and operational costs
 - it broadcasts the schedule (if changed)
- Convergence achieved by suitable activation probability
 - experimentally determined that 0.3 is a reasonable value
 - previously reported experiments with distributed, synchronous graph coloring
 - further experiments on *asynchronous* coloring show similar results
- Anytime process: can be interrupted when schedule is required
 - quality of schedule asymptotically improves over time

Tracking Nodes

- Each tracking node maintains tracks of nearby targets
 - ideally, we would have a multi-target tracker
 - instead, we tried a few heuristics to adapt BAE's tracker
 - each tracking node maintains one BAE tracker per target

- Measurement-track association

- given a measurement, the signal equation is used to try to determine which target might have been illuminated



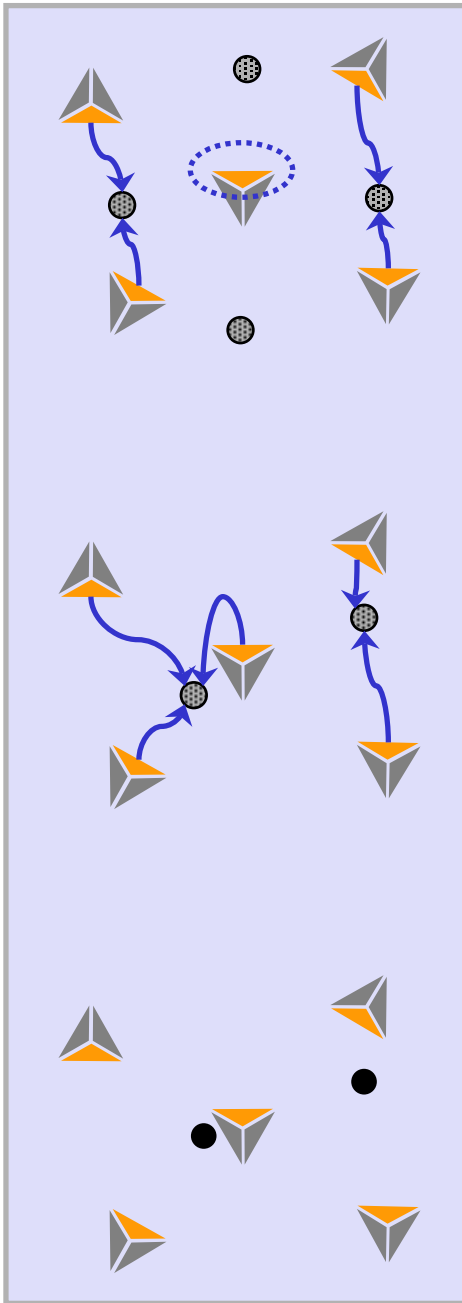
- project each target's position
 - compute theoretical signals at points on a narrow grid around target's expected position
 - determine if observed signal falls within theoretical range
 - if not, widen grid, up to some limit
- the measurement is associated with the target that gives the tightest match
 - if none match closely enough, measurement is unassociated ...

Track Initiation and Retirement

- Tracks initiated from unassociated measurements
 - measurements grouped using clustering
 - new tracks generated from significant groups
- Tracks are retired when they not updated for a certain time

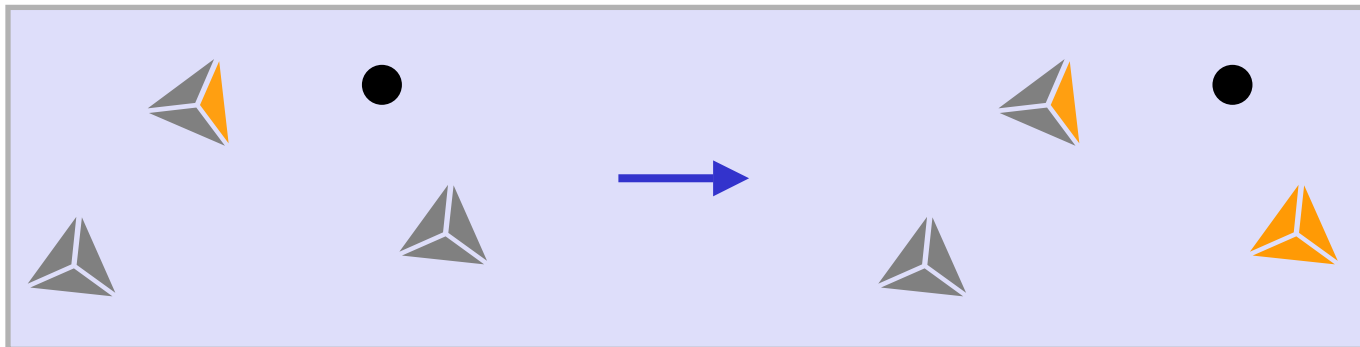
Clustering heuristic

- candidate positions are proposed at various points
- measurement association is attempted for these candidates
- new positions are computed from associated measurements
 - using a χ^2 -minimization test
- association retried with the original, unassociated measurements
- this process is repeated until a fix-point is reached



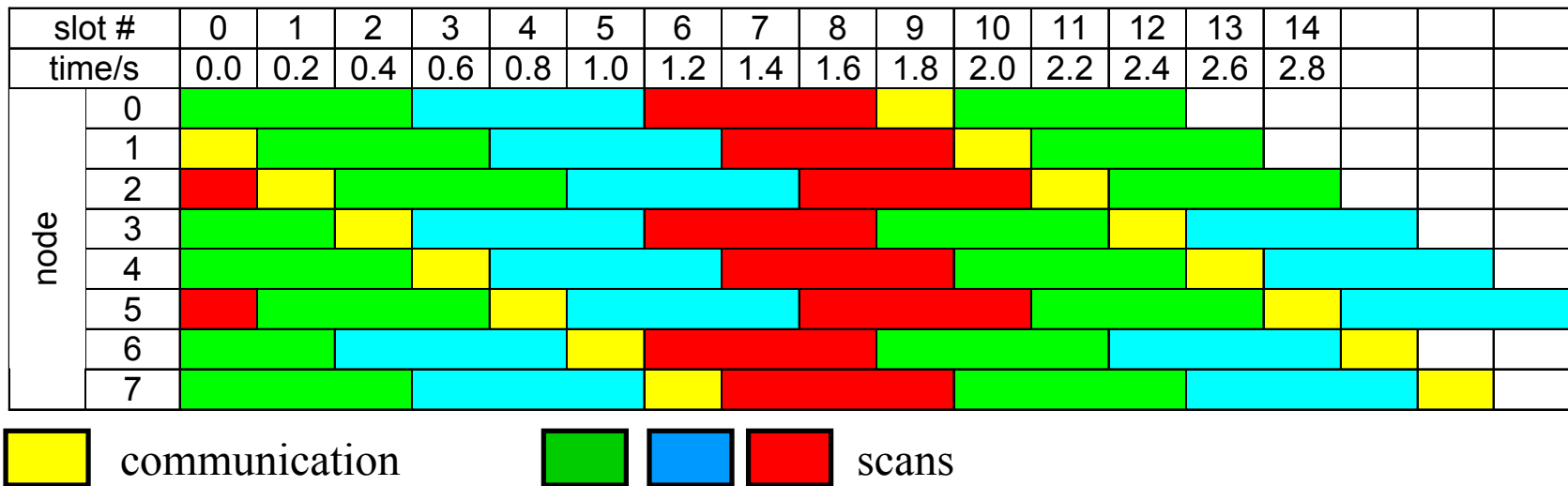
Control-based Supplements

- Cheap but coarse methods to supplement tracking-based method
- Exploit local measurements taken by sensor
 - a sector that gives a strong signal is a good candidate for another measurement
 - compare with signal predictions made during coordination to score tracker
- Use neighbor's measurements for proximity detection
 - allows some sensors to deactivate all sectors
 - simple scheme: a node reactivates if a neighbor that is within $1.5 \times$ (detection range) gets a strong signal.
 - finer scheme: a node reactivates if a neighbor that is within $1.5 \times$ (detection range) gets a strong signal from a sector that looks towards the node.



Communication

- Sensor schedule is integrated with a communication schedule
 - periodic schedule
 - 3 scan cycles of duration 0.6 seconds each
 - enough for 3 amplitude-only measurements
 - or 1 amplitude-and-frequency measurement
 - 1 broadcast cycle of duration 0.2 seconds
 - all nodes use the same frequency/channel
 - will need to be generalized for configurations with many more nodes
- Communication optimization
 - compression of multiple messages into single transmission
 - clock synchronization piggy-backed

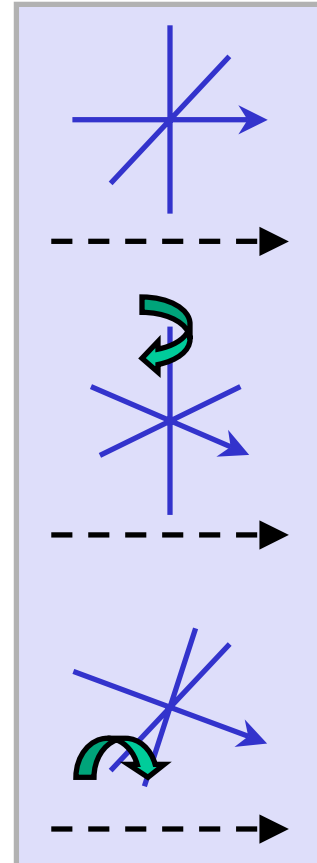
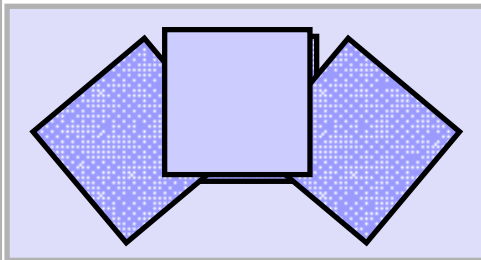


Calibration

- Theoretical signal model: $s(R,\theta) = K \exp(\theta^2/A)/R^2$
- Compare with observed signal
 - amplitude-only measurements every 0.5 seconds

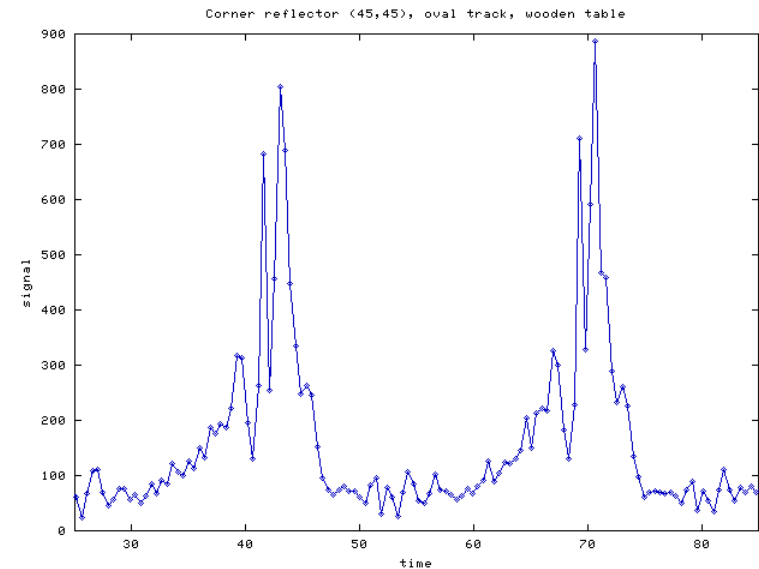
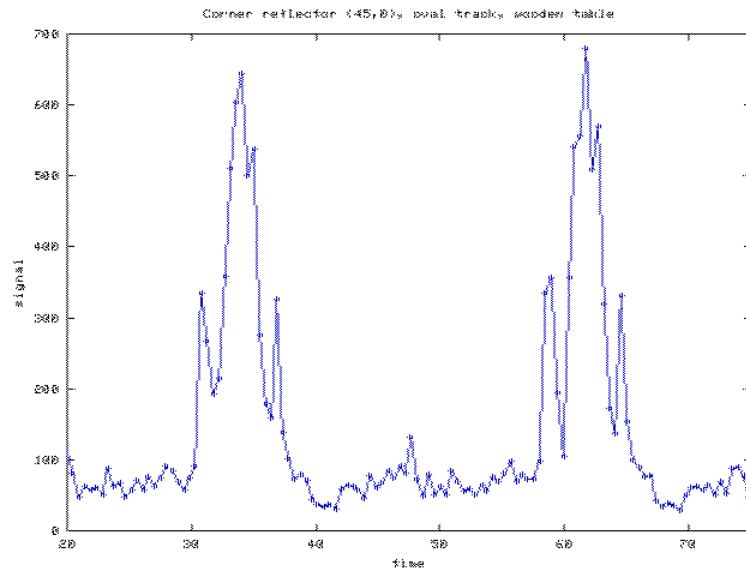
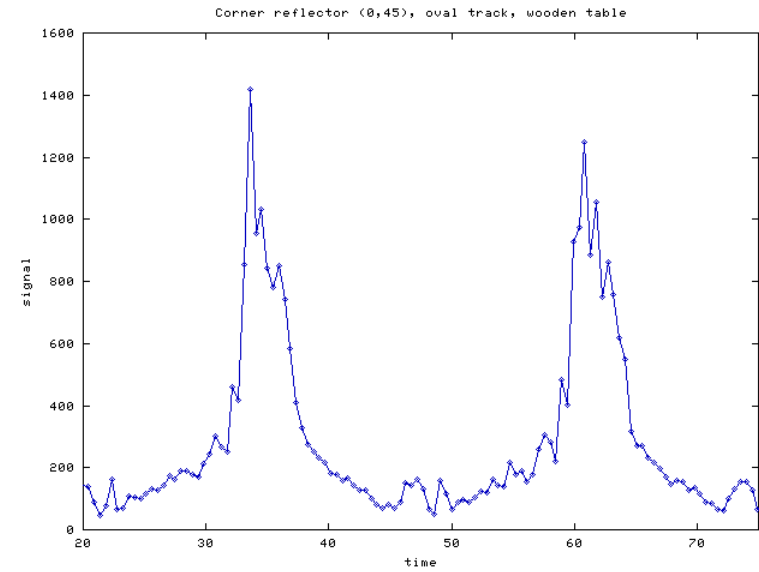
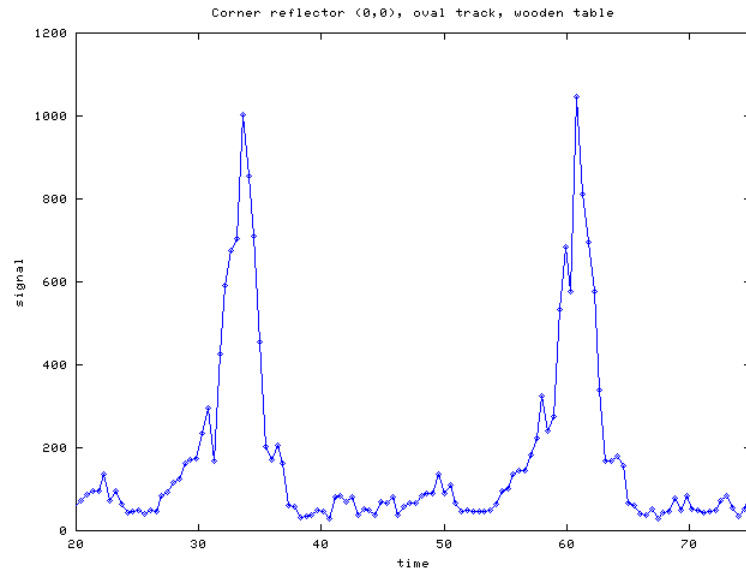


- Sensor raised ~2.5 feet on wooden table
- For some experiments, hood constructed around sensor from radar absorber
- Target moves along oval track
 - length 10 foot
 - width 4 foot
- Orientation of target varied
 - w.r.t. direction of travel



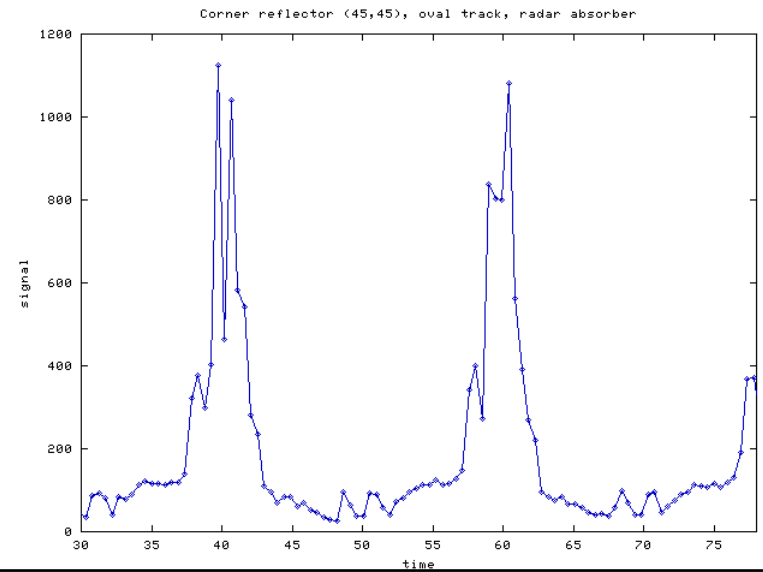
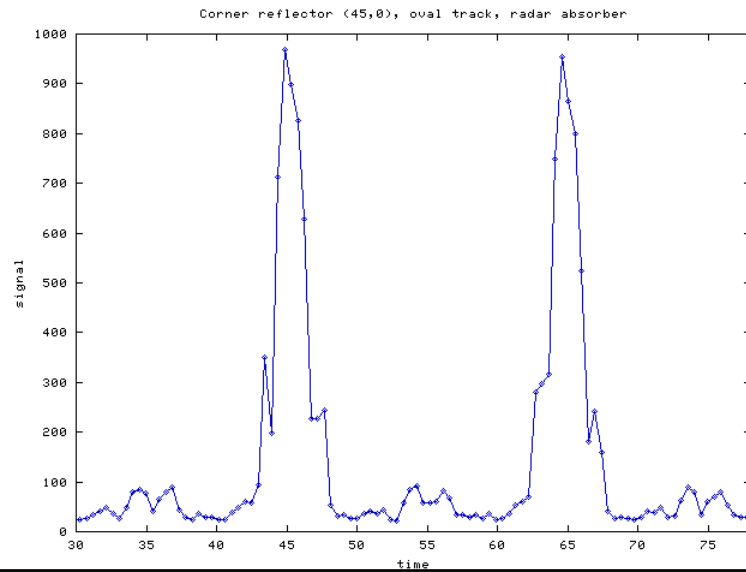
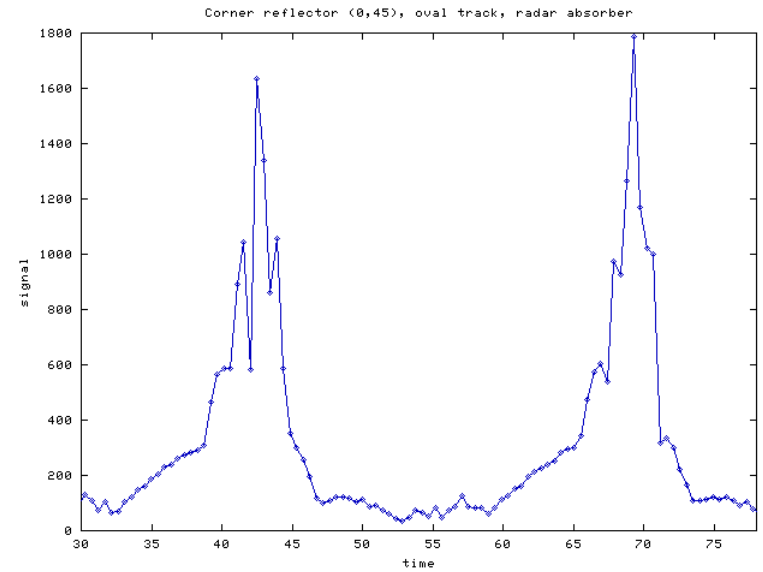
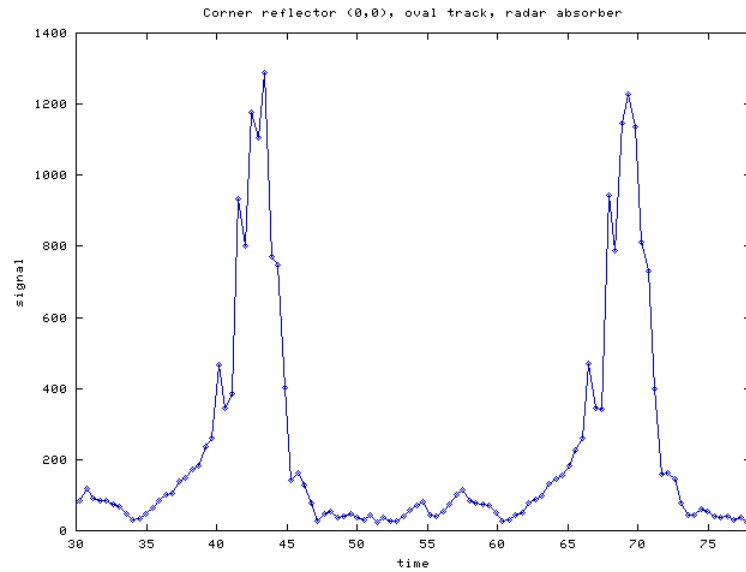
Calibration Results: No Absorber

- Multi-path reflection postulated as major (but not only) source of noise



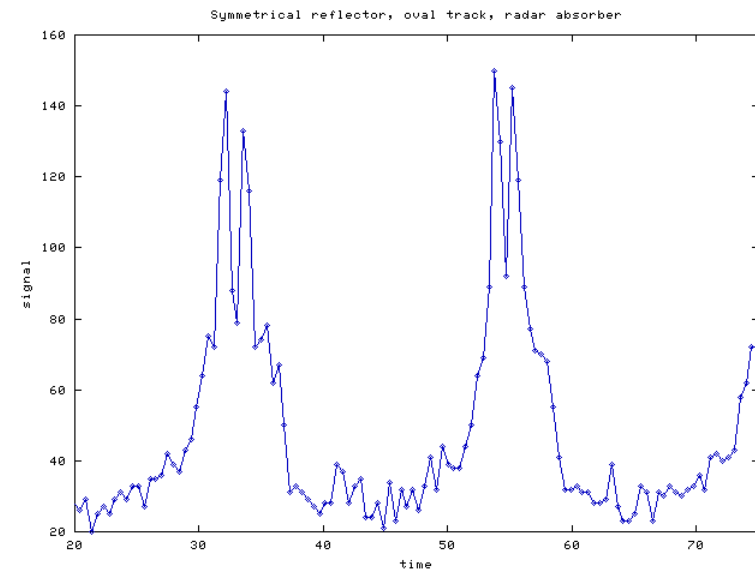
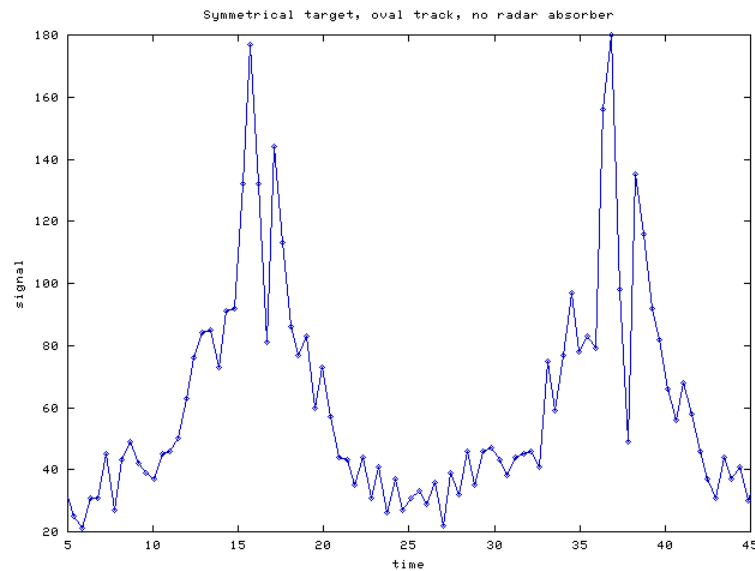
Calibration Results: With Absorber

- Should be no multi-path reflection – target orientation matters



Calibration Results: Spherical Reflector

- Spherical reflector
 - still observe troughs
 - probably due to zero-radial velocity
 - target wobble can also affect signal



In-Situ Calibration

- Measure signal as target moves along known track
 - measure sector 0 on all sensors, then sector 1, then sector 2
- Fit signals to $K \exp(\theta^2/A)/R^\gamma$ for each sensor & sector
 - for K, A & γ
- Some sector deliberately unused due to known reflection problems

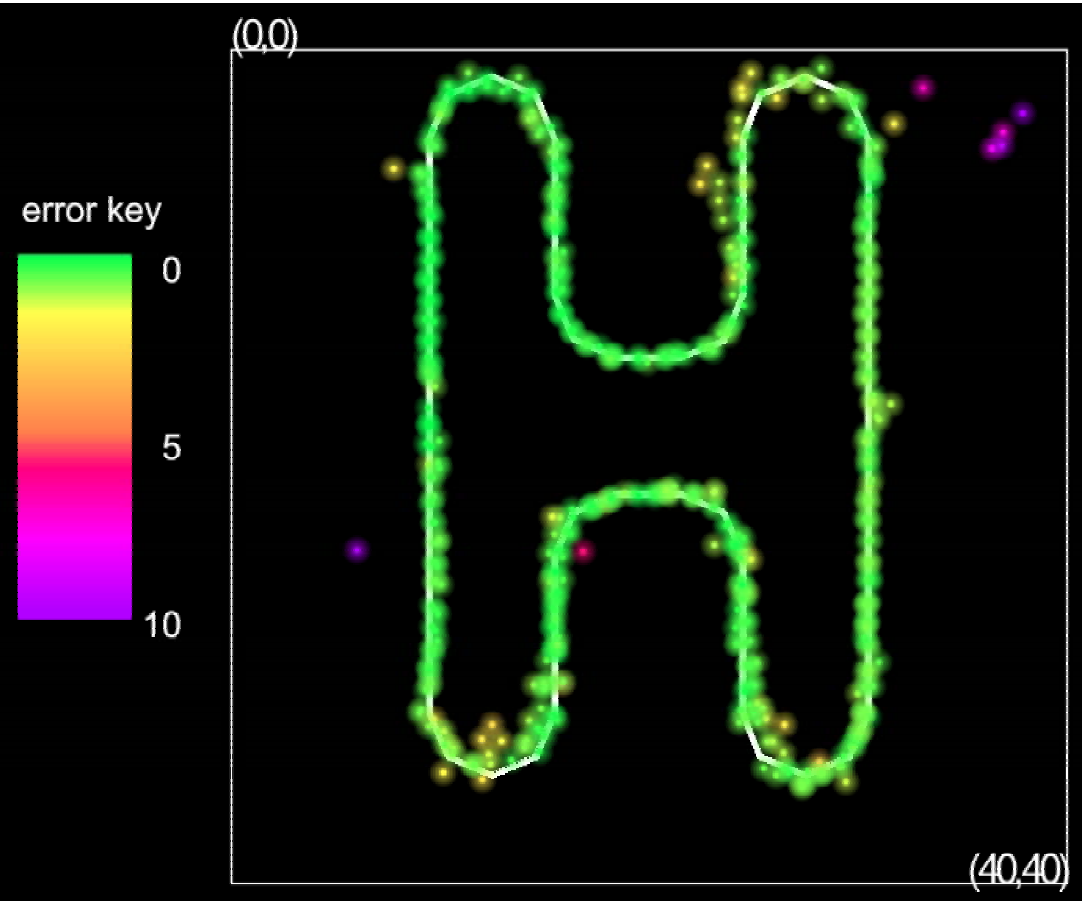
Node	0			1			2			3			
	Sector	0	1	2	0	1	2	0	1	2	0	1	2
K/1000		27.1	7.8		12.2	3.5			3.5	13.2	27.0		11.2
A/1000		0.3	1.8		1.4	2.2			1.1	0.9	1.0		1.5
γ		2.0	2.0		2.4	2.0			1.8	2.2	2.4		2.2
Node	4			5			6			7			
	Sector	0	1	2	0	1	2	0	1	2	0	1	2
K/1000		1.6	32.3		2.1	45.5	88.9	11.8		7.6		17.9	46.1
A/1000		6.5	1.4		6.1	1.1	0.6	1.5		1.5		1.5	1.2
γ		1.5	2.7		1.6	2.9	3.0	2.0		2.0		2.0	2.8

results from Mitre lab.

	K/1000	A/1000	γ
min	1.6	0.3	1.5
mean	21.1	1.8	2.2
max	88.9	6.5	3.0

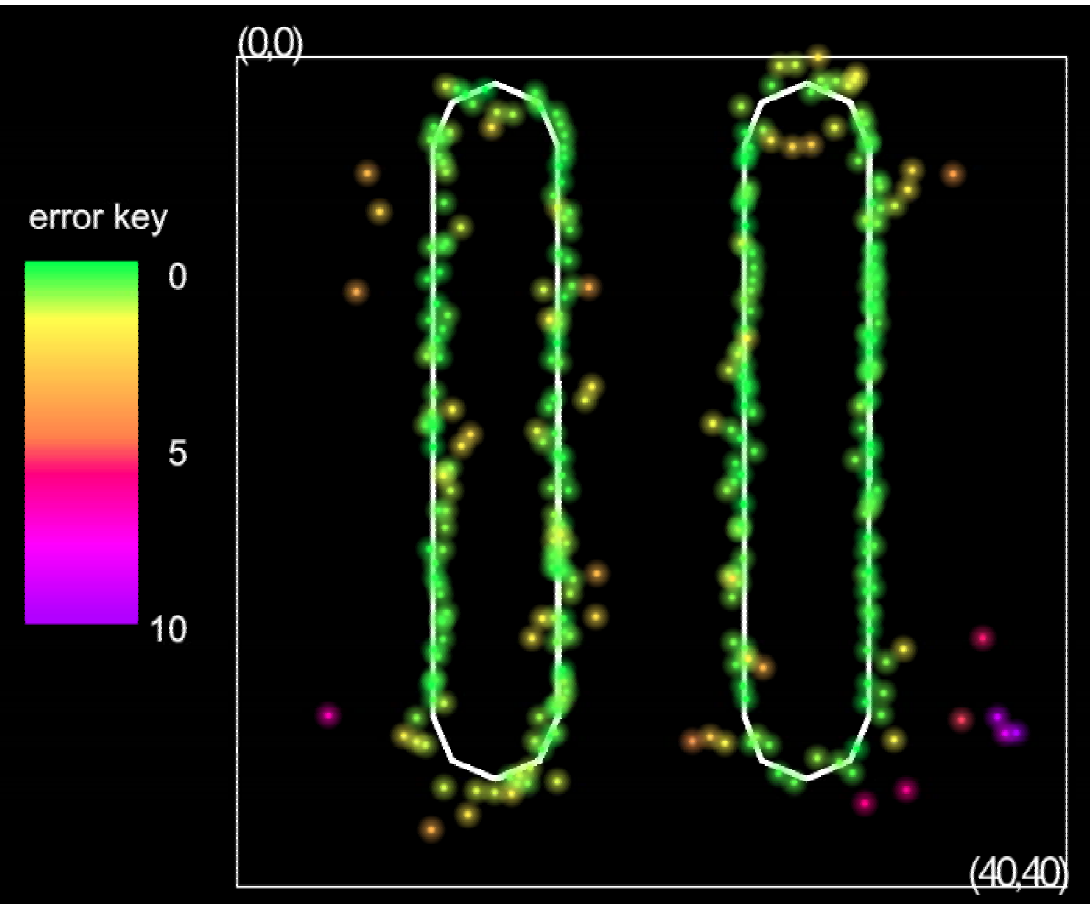
ranges

Results with Simulator: Single Target



- Mean track period = 2.0 seconds
- R.M.S. error = 1.6 feet
 - computed by projecting ground-truth to the time of each track point
 - Y1 result: 3.1 feet
- Mean power usage = 53%
 - should be able to achieve better
 - (beam seconds in RadSim log indicate ~20% usage)
 - Y1 result: 27%
- Communication usage = 0.48 messages per node per second

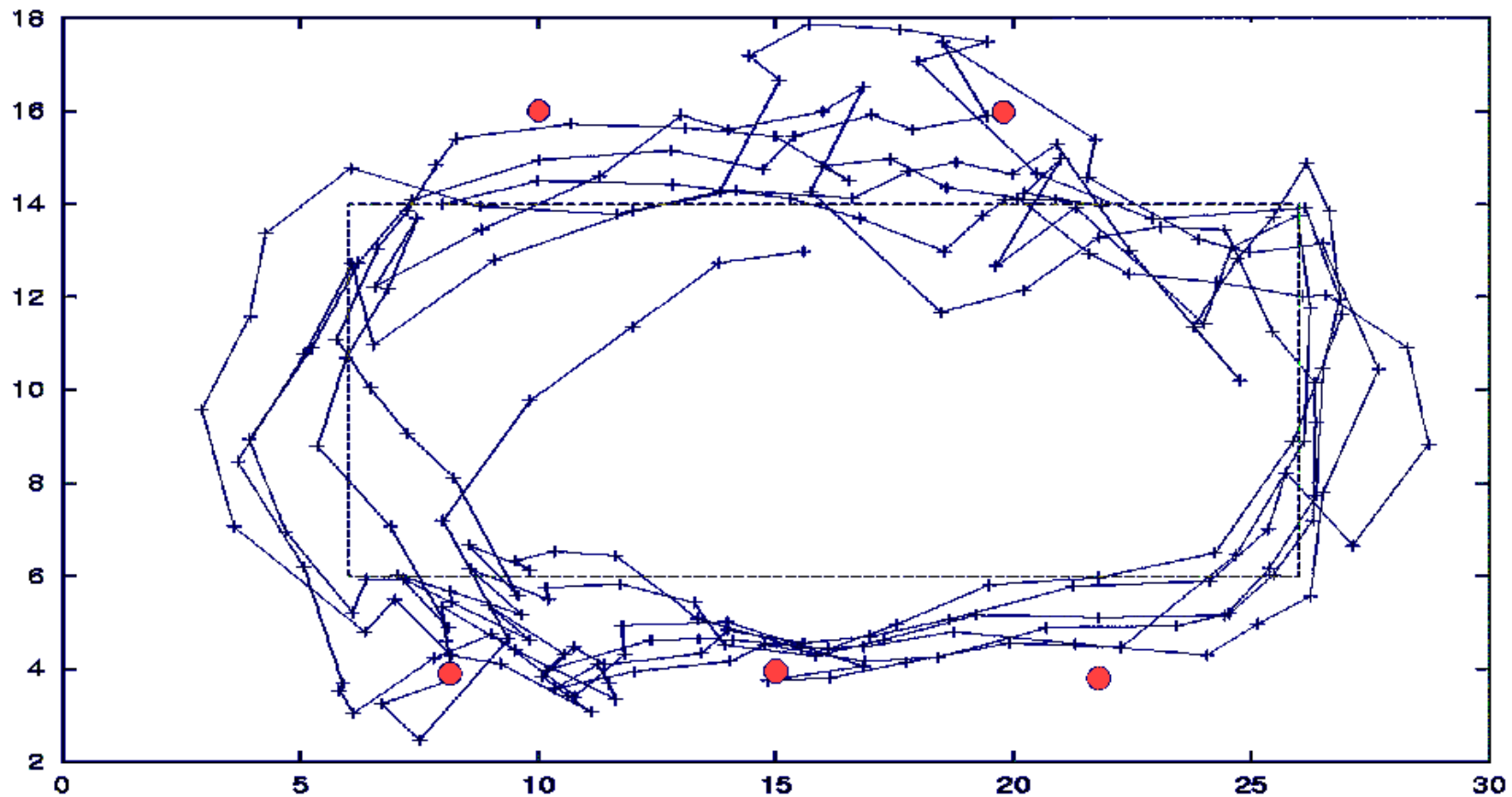
Results with Simulator: Two Targets



- Tracked simultaneously
- Mean track period per target
 - config 1: 1.1 seconds
 - config 2: 2.0 seconds
- R.M.S. error
 - config 1: 2.3 feet
 - config 2: 1.6 feet
 - each track point was assigned to the closer of the ground-truth targets
- Mean power usage
 - config 1: 53%
 - config 2: 61%
- Communication usage
 - config 1: 0.38 messages per node per second
 - config 2: 0.40 messages per node per second

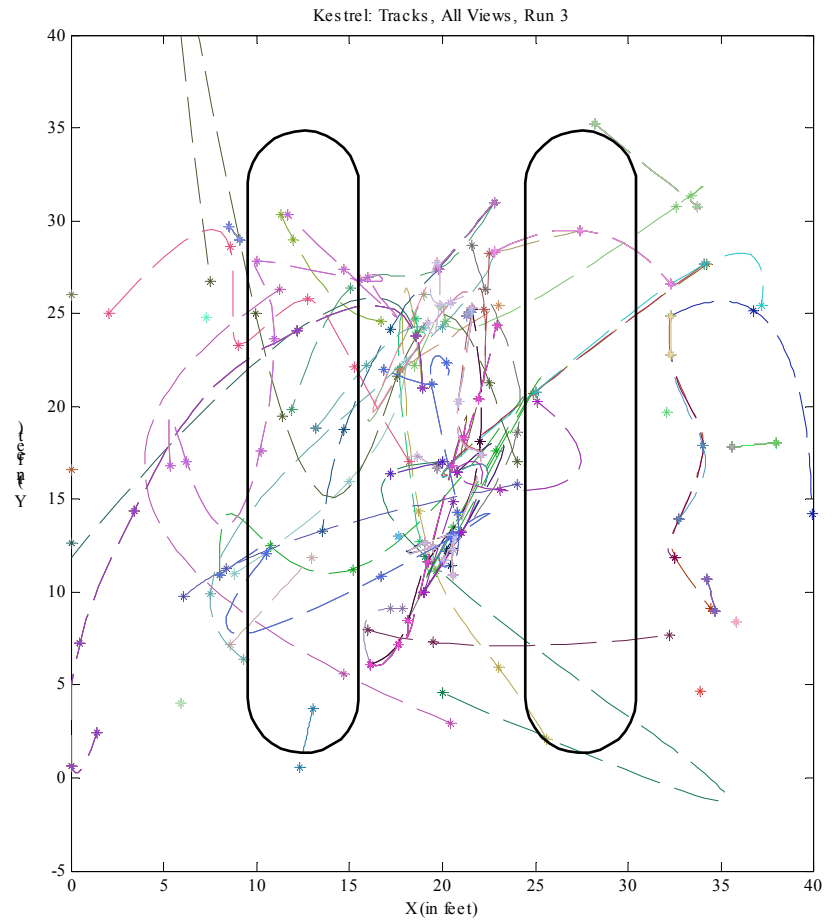
Results with Hardware at Kestrel

- Single target
 - 5 sensors, 400 seconds
 - this is the best performance, not typical



Results with Hardware at Mitre Lab.

- Mitre experiments
 - 0.31 messages sent per node per second
 - 21.4 bytes sent per node per second
 - not including system headers
 - 64% power usage
 - where 100% = 3 beams
- Tracking poor
 - many, many tracks generated
 - measurement noise interfered with track-measurement association

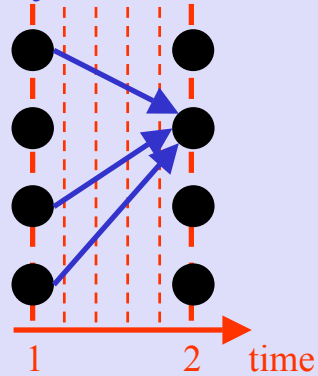


Dynamics of Distributed Constraint Optimization

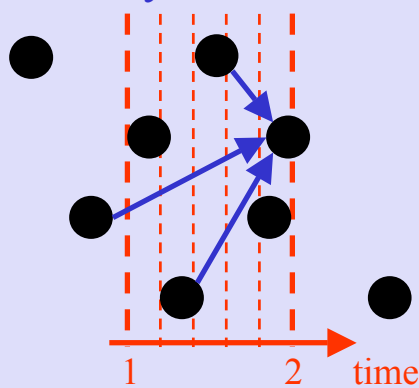
- Summary of previous work:
 - distributed graph colouring provides a clean benchmark for experimental assessment of distributed constraint optimization
 - essentially the same as the scheduling algorithm
 - metric being optimized (minimized) is the fraction of edges that are conflicts
 - i.e., that connect nodes of the same color
- Distributed, anytime coloring algorithm
 - each node chooses its own color
 - random initialization
 - stochastic loop in which each node chooses a color that minimizes its conflicts with its neighbors
 - informs its neighbors when its color changes
- Previously reported results for synchronous algorithm on sparse graphs
- What happens under other conditions? ...

Asynchronous Coloring

synchronous



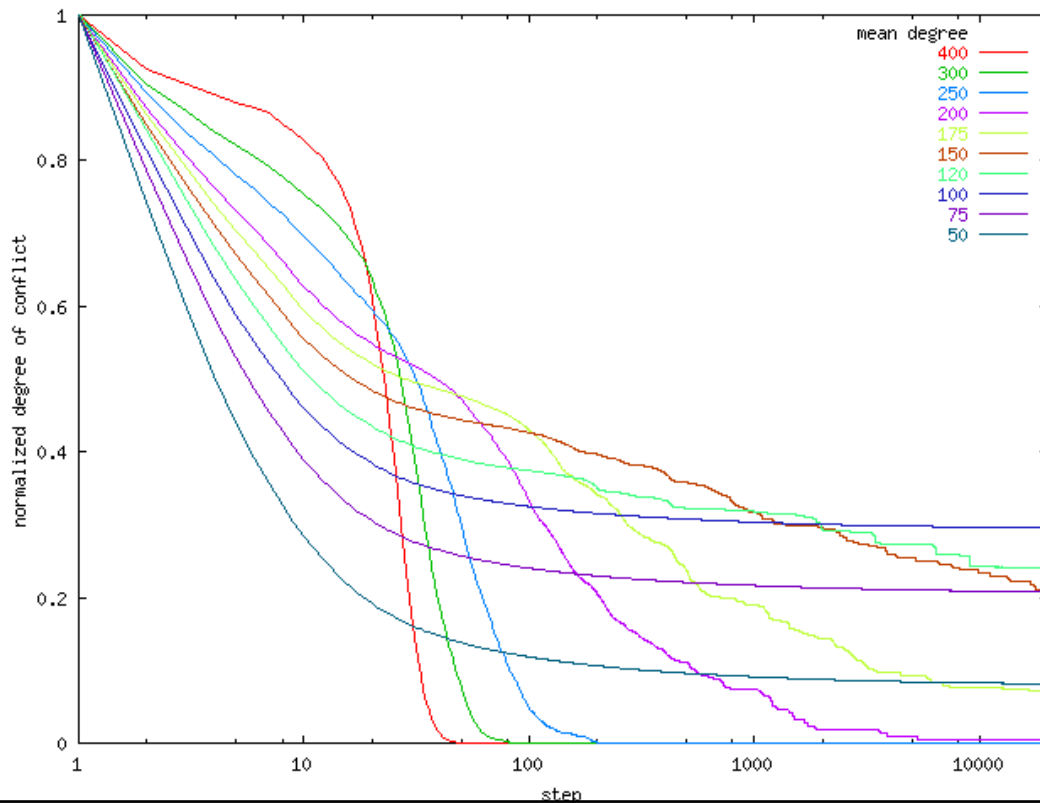
asynchronous



- Is strict synchronization needed for coordination?
- Each node updates its color with a mean period P
 - nodes initialized with random phases
 - each node has random jitter in its period, causing relative phase between nodes to drift
- If the communication latency is less than P , each node has *more* up-to-date information than in synchronous coloring
 - improves convergence

Dense Graphs

- Density corresponds to interconnectedness of sensor network
- Generate k -colorable random graphs, given mean degree D
 - randomly assign k colors to N disjoint nodes
 - randomly generate $DN/2$ edges between nodes of different colors
 - remove colors
 - resulting graph has a chromatic number of no more than k
 - the chromatic number is likely exactly k



- Averages over 20 graphs
 - each 10-colorable
 - mean size ~ 2000
 - mean degree from 50 to 400
- Fluffy comparison
 - for $D=400$
 - number of steps for proper coloring ~ 30
 - activation probability = 0.3
 - equivalent to ~ 18000 color changes for all 2000 nodes
 - equivalent branching factor b for a back-tracking algorithm

$$b^{2000} = 18000 \Rightarrow b \approx 1.005$$
 - don't take too seriously
 - not enough evidence yet

Plans

- Adapt experimental set-up to achieve convincing results on hardware
- Quantify effects of coordination using simulator
 - large scale experiments (e.g., 100 nodes)
 - compare with individual-sensor optimization
- Integrate S.C. tracker
- Extend results on dynamics
 - sparse graphs with local structure
 - compare distributed colorer with sequential colorer on dense graphs
- Develop new theoretical framework
 - position games – add dynamic strategies to games
- Investigate information theory for coordination metrics