LISC

LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

Optimal Control of Mobile Sensor Networks

Silvia Ferrari
Laboratory of Intelligent Systems and Controls (LISC)
Department of Mechanical Engineering and Materials Science
Duke University

MAE Seminar Series
Department of Mechanical and Aerospace Engineering
Princeton University

November 20, 2009

Outline

- Introduction
- Motivation and Applications
- Optimal Control Framework
- Geometric Models of Mobile Sensors
- Cooperative Track Detection Problem
- Treasure Hunt Problem
- Marco Polo Problem
- Conclusions and Q&A

Introduction

- **Modern Surveillance Systems** – multiple sensors installed on mobile platforms

  -- landmine detection and identification

  -- monitoring of endangered species

  -- monitoring of urban environments

- Traditional paradigm: sensor information is used as feedback to the vehicle in order to support the vehicle navigation

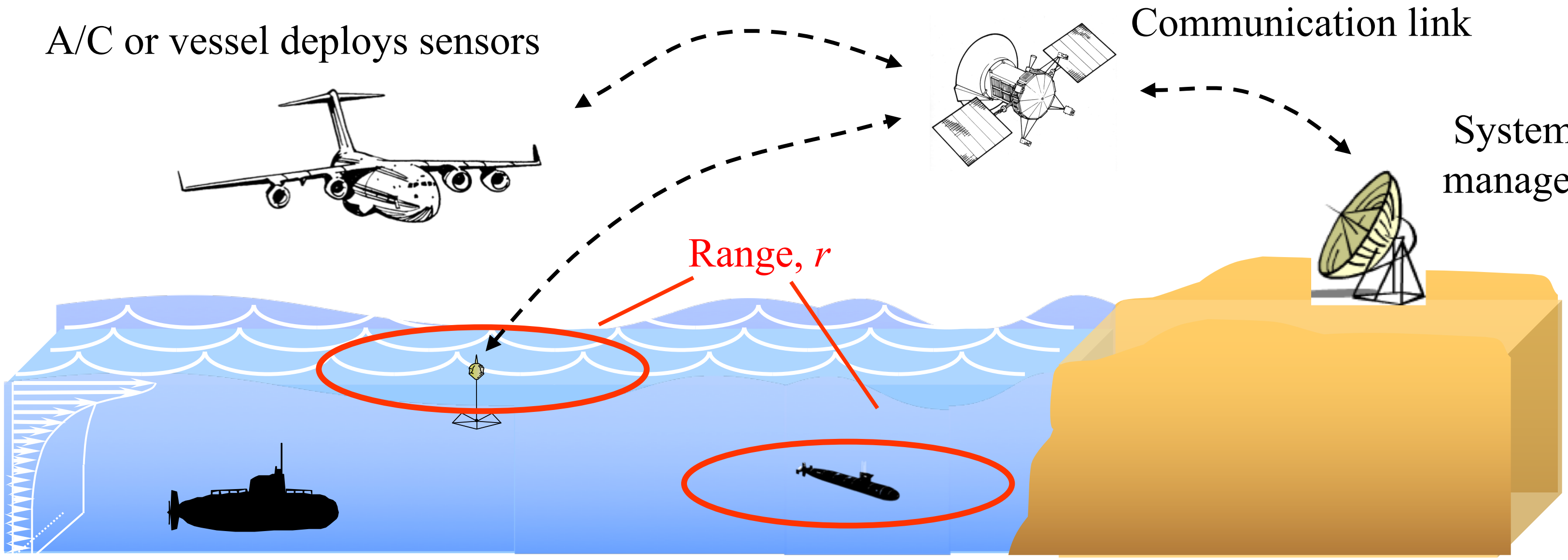
- New paradigm: the sensor motion is planned in view of the expected measurement process, in order to support the sensing objectives

- **LISC Research Emphasis:** Geometric sensor path planning

  -- Address couplings between sensor measurements and sensor dynamics

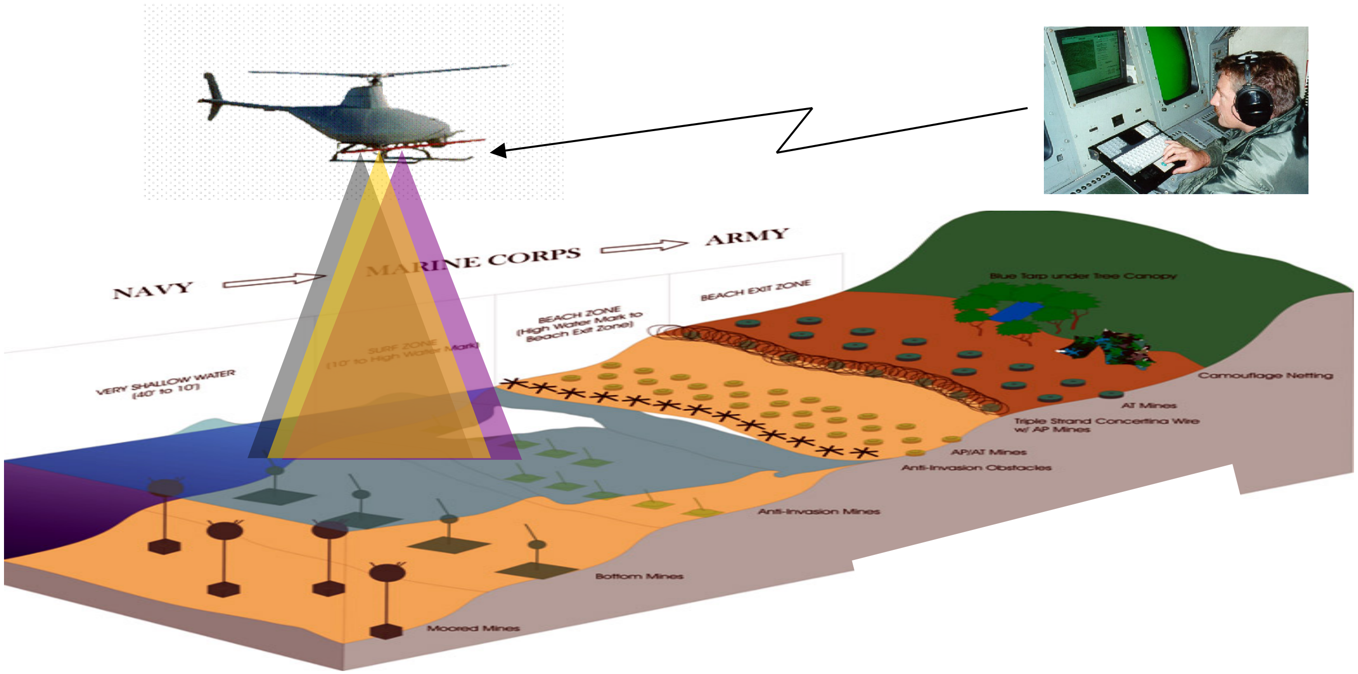
  -- Optimize sensing objectives (e.g., detection, classification, tracking..)

Applications: Undersea Surveillance



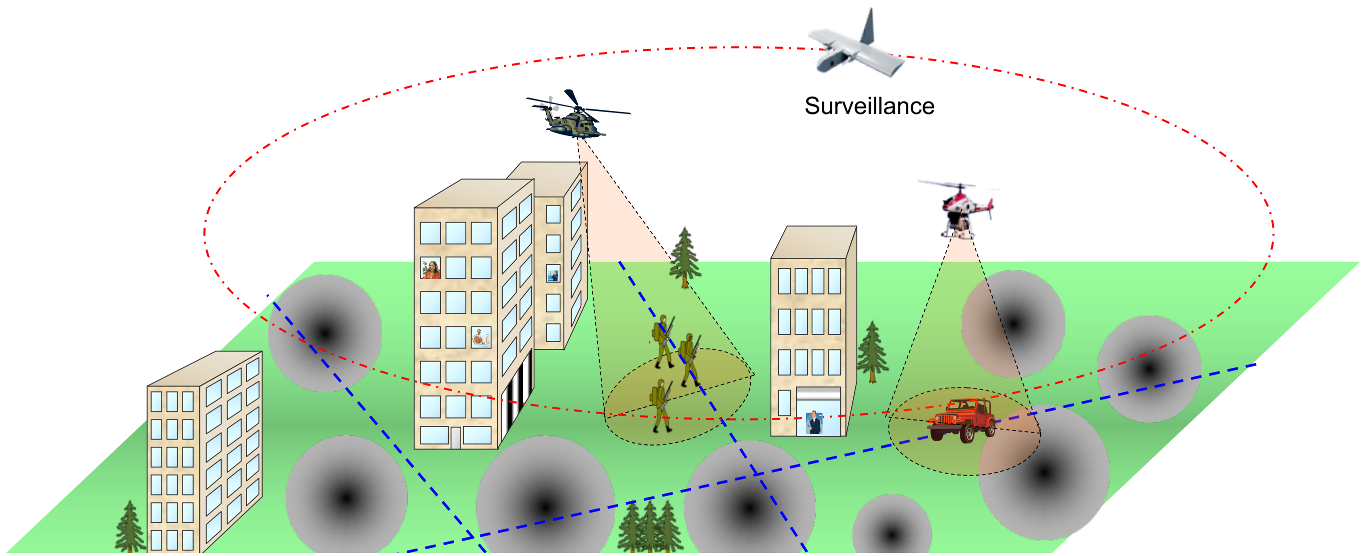
- Sensors: acoustic, w./ GPS, limited micro-level processing, mobile
- Targets: passive, mobile, unauthorized
- Environment: heterogeneous bathymetry and ambient properties, currents
- Sensing objectives: coverage, tracking, detection, classification

Applications: UAV Demining



- Sensors: Cameras, IR, GPR, EMI, synthetic aperture radar (SAR)
- Targets: static, hidden, hazardous
- Environment: heterogeneous soils, weather, time of day, obstacles
- Sensing objectives: detection, classification

Applications: Urban Monitoring



- Sensors: Cameras, IR, synthetic aperture radar (SAR)
- Targets: static, hidden, mobile, evading
- Environment: time of day, obstacles
- Sensing objectives: detection, classification, tracking

Optimal Control Framework

Performance measure to be optimized w.r.t. $\mathbf{u}(t)$,

$$J = \phi\left[\mathbf{x}(t_f), t_f\right] + \int_{t_0}^{t_f} L[\mathbf{x}(t), \mathbf{u}(t), t]\,dt, \quad \text{with I.C. } \mathbf{x}(t_0)$$

subject to nonlinear time-varying dynamics,

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t), t]$$

and subject to equality and inequality constraints

$$\mathbf{c}[\mathbf{x}(t), \mathbf{u}(t)] \geq 0$$

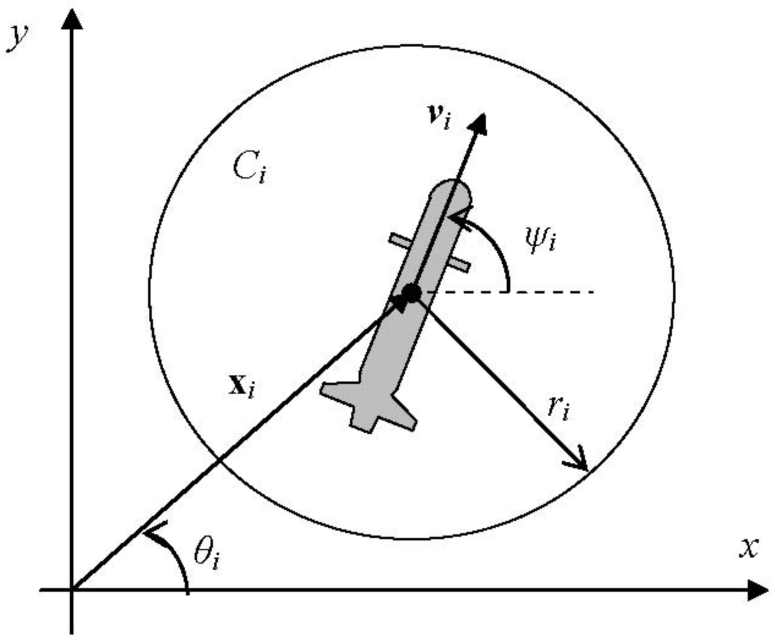
- Vehicles control vector: $\mathbf{u}[\mathbf{x}(t), t]$
- Vehicles positions: $\mathbf{x}(t)$
- Environmental and sensing parameters: $\mathbf{p}(t)$

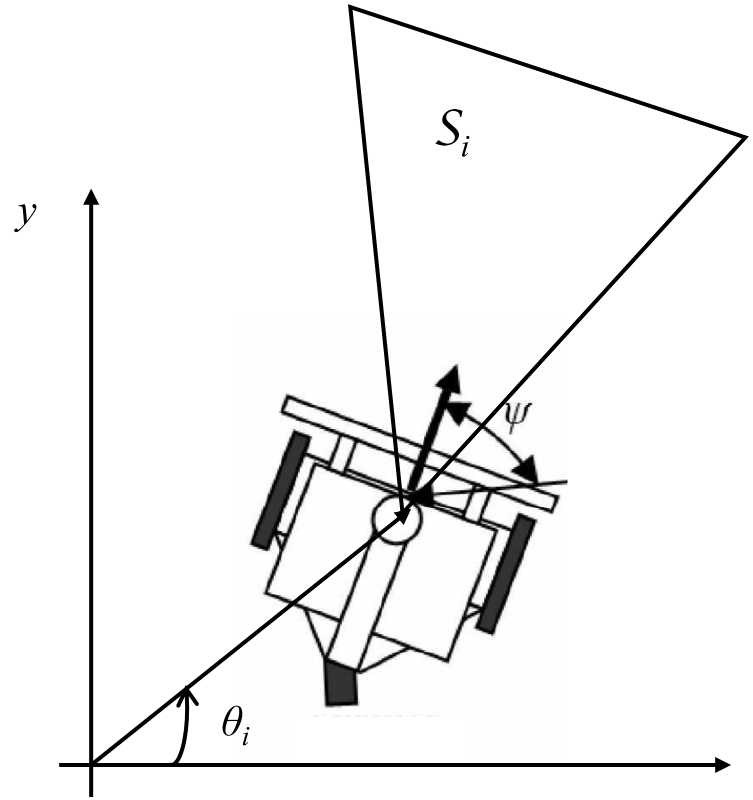
Modeling of Mobile Sensor Networks

Mobile Sensor Model

- The sensor is characterized by a discrete field-of-view (FOV) geometry, and by a joint probability density or mass function (PDF or PMF).
- The platform is characterized by a discrete vehicle geometry and a dynamic equation.

Examples:

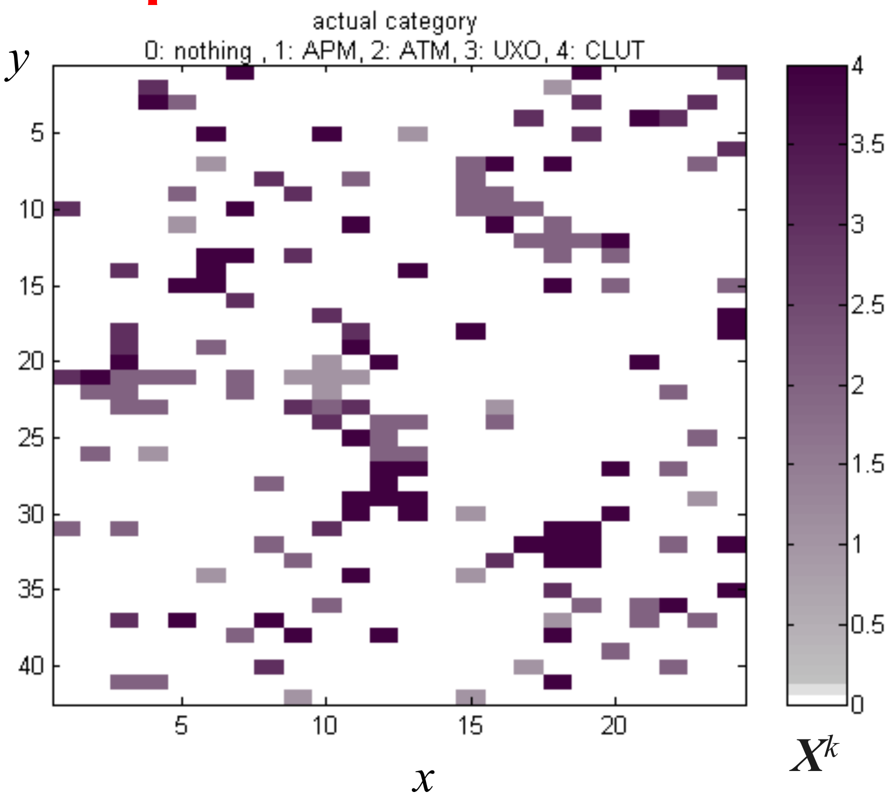


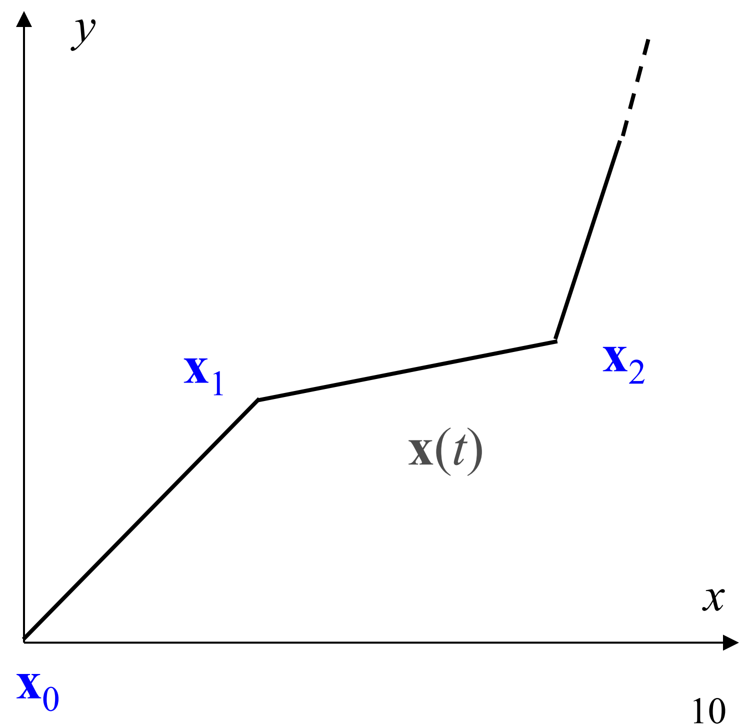


Targets and Workspace

- The workspace may contain obstacles and changing environmental conditions
- Stationary targets are characterized by discrete geometries, and a prior probability density or mass function or *prior*: $p(X^k)$, $p(\boldsymbol{\lambda}^k)$
- Moving targets are characterized by a Markov motion model: $p(\mathbf{x}_j, \mathbf{v}_j, \theta_j)$, $j = 1, \ldots$

Examples:





Probabilistic Model of Sensor Measurements

Classical Sensor Model (Estimation Theory):

$$\mathbf{Z}^k = \mathbf{h}(\mathbf{X}^k, \boldsymbol{\lambda}^k)$$ (Discrete time)

- Measurement vector: $\boldsymbol{Z}^k = [z_1(t_k) \ \ldots \ z_r(t_k)]^T$ (Continuous variables)
- Target state: $\boldsymbol{X}^k = [x_1(t_k) \ \ldots \ x_n(t_k)]^T$
- (p x 1)-Vector of sensor characteristics, e.g., noise and measurement errors: $\boldsymbol{\lambda}^k$
- Deterministic, nonlinear vector function: $\mathbf{h}$: $\Re^{n \times p} \rightarrow \Re^r$

Probabilistic Sensor Model:

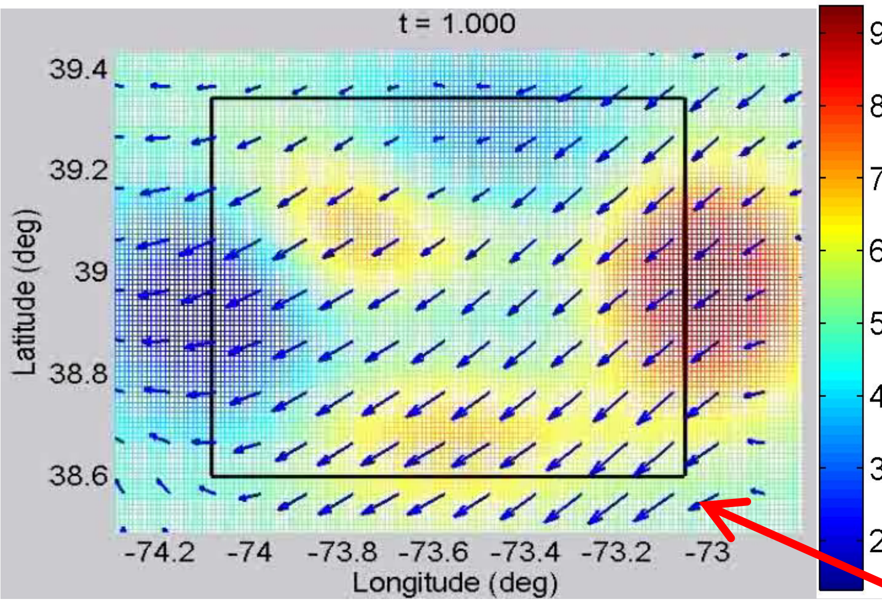
$$p(\mathbf{Z}^k, \mathbf{X}^k, \boldsymbol{\lambda}^k) = p(\mathbf{Z}^k \mid \mathbf{X}^k, \boldsymbol{\lambda}^k) p(\mathbf{X}^k) p(\boldsymbol{\lambda}^k)$$ (Discrete time)

- Set of random measurement or observations at t_k: $\boldsymbol{Z}^k$ (Discrete variables)
- Set of random target state variables at t_k: $\boldsymbol{X}^k$
- Set of random sensor characteristics at t_k: $\boldsymbol{\lambda}^k$
- Joint probability density or mass function (PDF or PMF): $p(\mathbf{Z}^k, \mathbf{X}^k, \boldsymbol{\lambda}^k)$

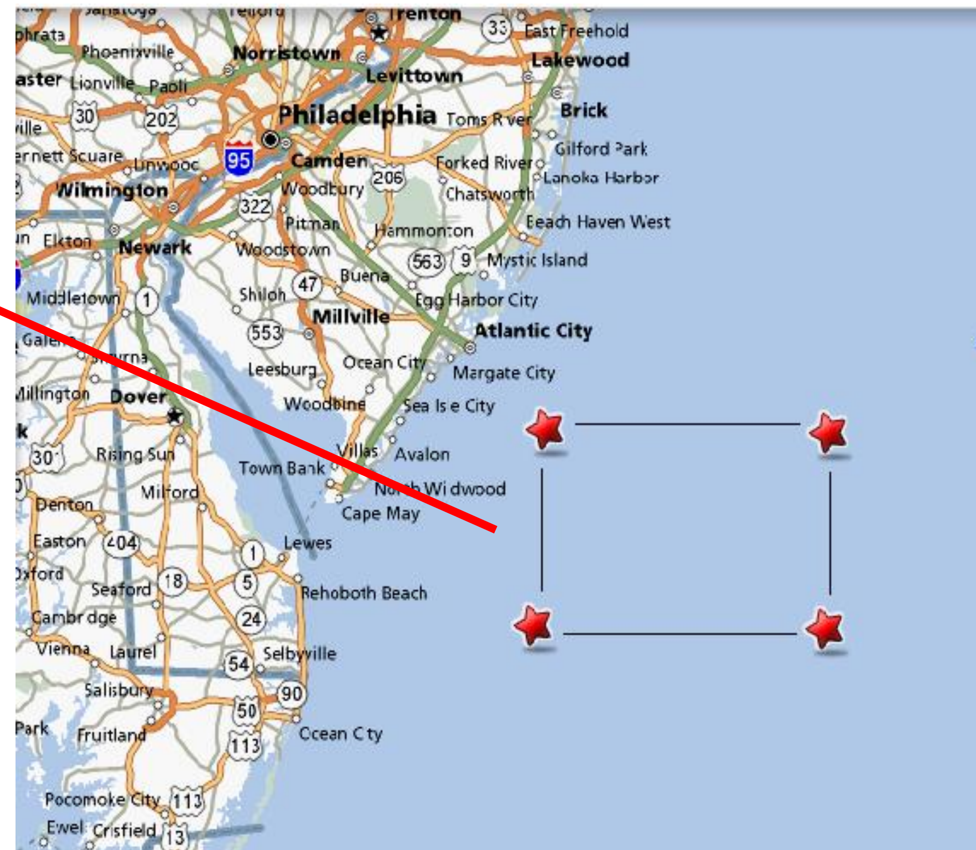
Underwater Sensor Networks for Cooperative Track Detection

Workspace and Environmental Conditions

Current Vector Field Over a 5-Day Period



Environmental conditions influence, *r* (Km)



Real CODAR-Measured Current Field (100 naut-NJ Coast)†

†[COOL, Rutgers University]

LISC
LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

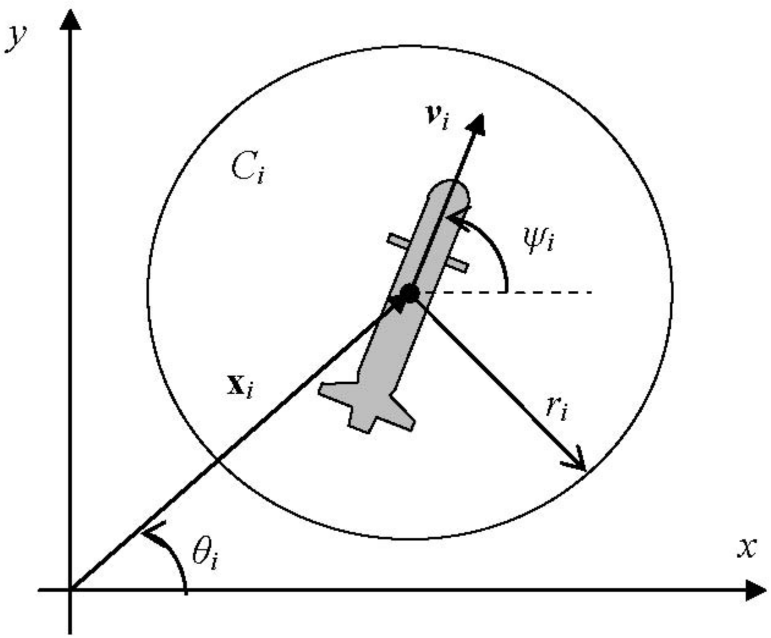
Underwater Sensor Model

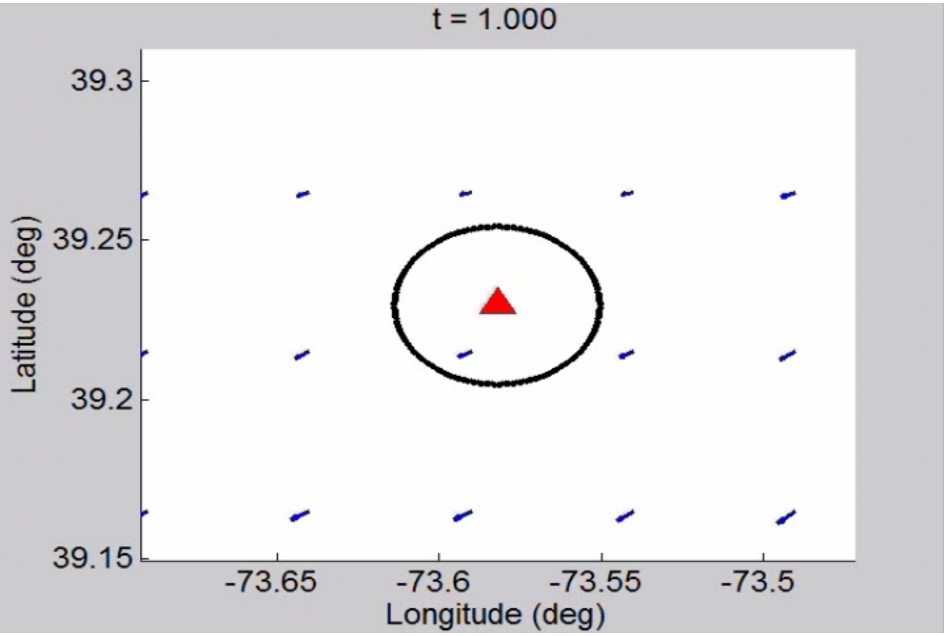
Underwater Vehicle Dynamics:

$$\mathbf{M}_i\,\dot{\boldsymbol{\nu}}_i + \mathbf{C}_i(\boldsymbol{\nu}_i)\,\boldsymbol{\nu}_i + \mathbf{D}_i(\boldsymbol{\nu}_i)\,\boldsymbol{\nu}_i + \mathbf{g}_i(\boldsymbol{\xi}_i) = \tilde{\mathbf{B}}_i\,\mathbf{T}_i(\boldsymbol{\nu}_i, \mathbf{u}_i), \quad i = 1, ..., n$$

Vehicle's control bounds: $\| \mathbf{u} \| \leq V_{\max}$ Sensor's effective range: $r_{\max} = r\,(\mathbf{x}, \mathbf{u}, \alpha)$

Simulation: sensor motion and effective range





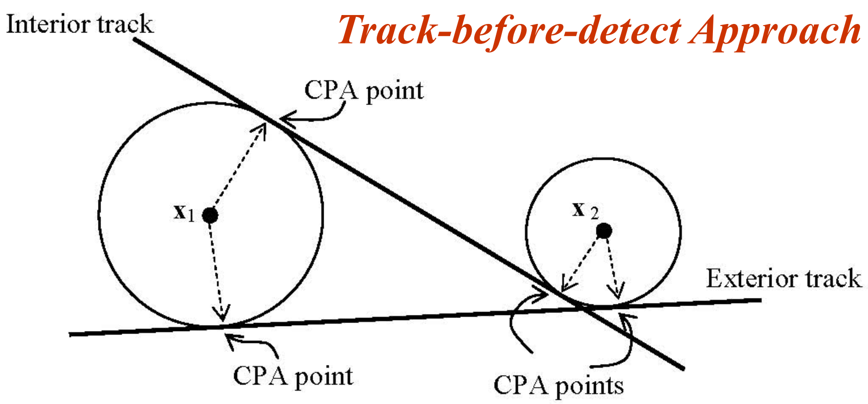
K. C. Baumgartner, S. Ferrari, and A. Rao, "Optimal Control of a Mobile Sensor Network for Cooperative Target Detection," *IEEE Journal of Oceanic Engineering*, in press, now available on line.

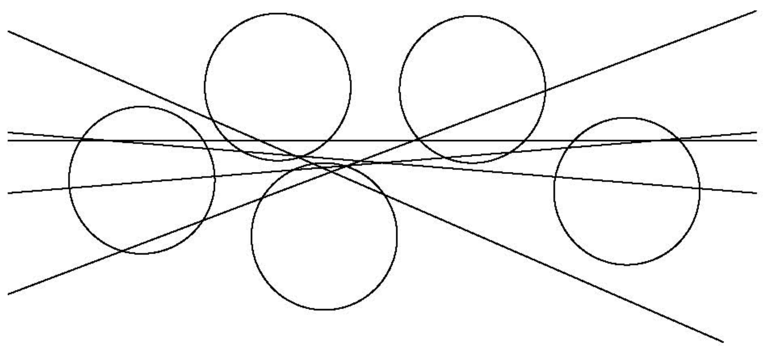
LISC
LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

Objective: Cooperative Track Detection

***Track Coverage*:** Sensors' ability to cooperatively detect target tracks

Track-before-detect Approach





k-Transversals or Stabbers of a family of 5 circles ($k = 2$)

Geometric Transversal Theory

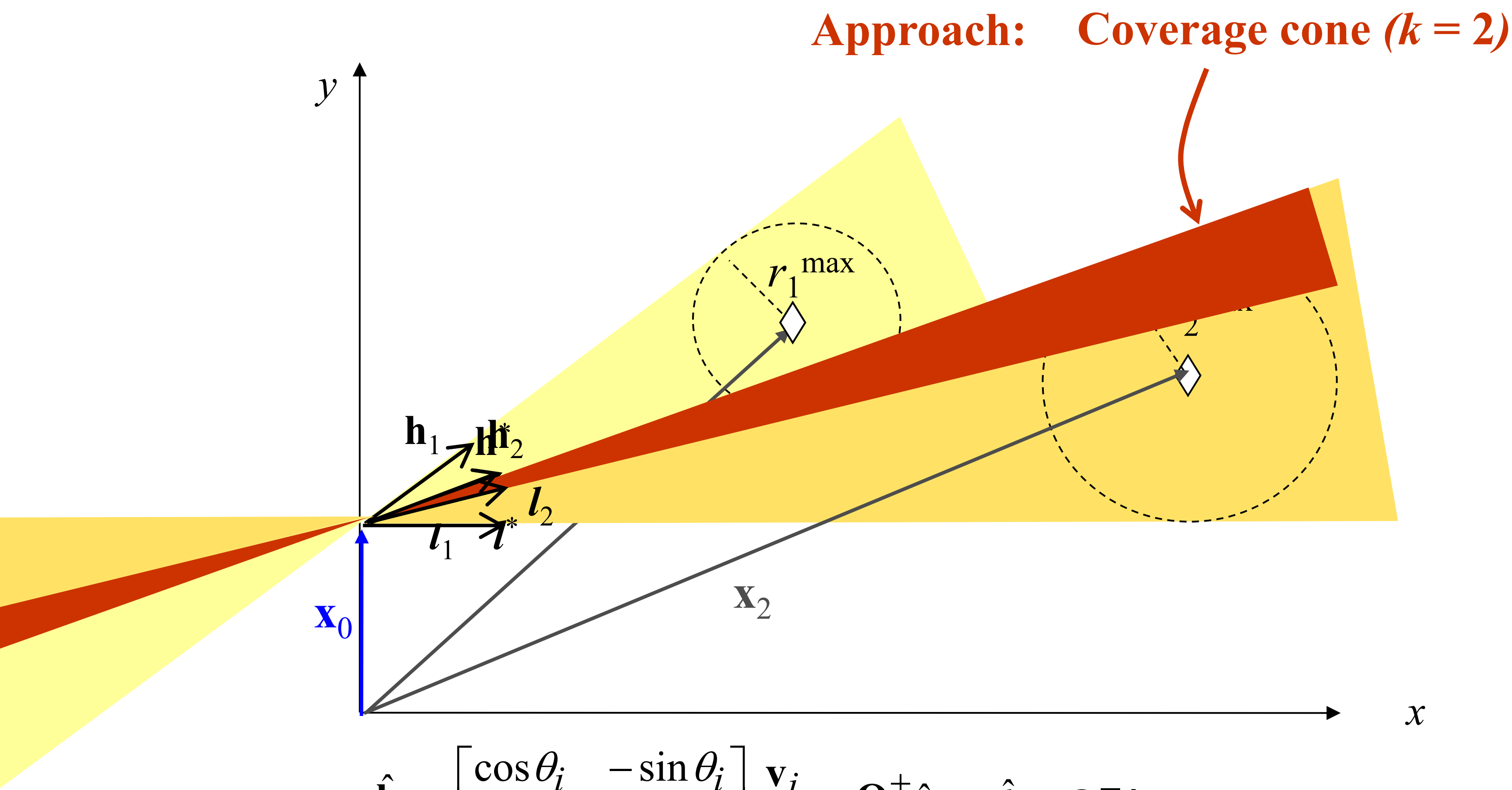
- "Divide and conquer" algorithm for constructing the space of line transversals to n line segments in $\Re^2$ [Edelsbrunner, 1982].
- $O(r)$ algorithm for finding the slope of a line transversal to a family of n disjoint convex translates [Egyed and Wenger, 1989].
- Finding a line transversal for a family of n line segments or a family of n circles in $\Re^2$ takes $\Omega(n \log n)$ time on an algebraic decision tree [Avis et al., 1984].

S. Ferrari, "Track Coverage in Sensor Networks," *Proc. American Control Conference*, Minneapolis, MN, 2006, pp. 2053-2059.

LISC
LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

Track Coverage Cones

- Represent space of line transversals for k circles belonging to a family of n (non-translates) circles as a function of their location in $A \in \Re^2$.



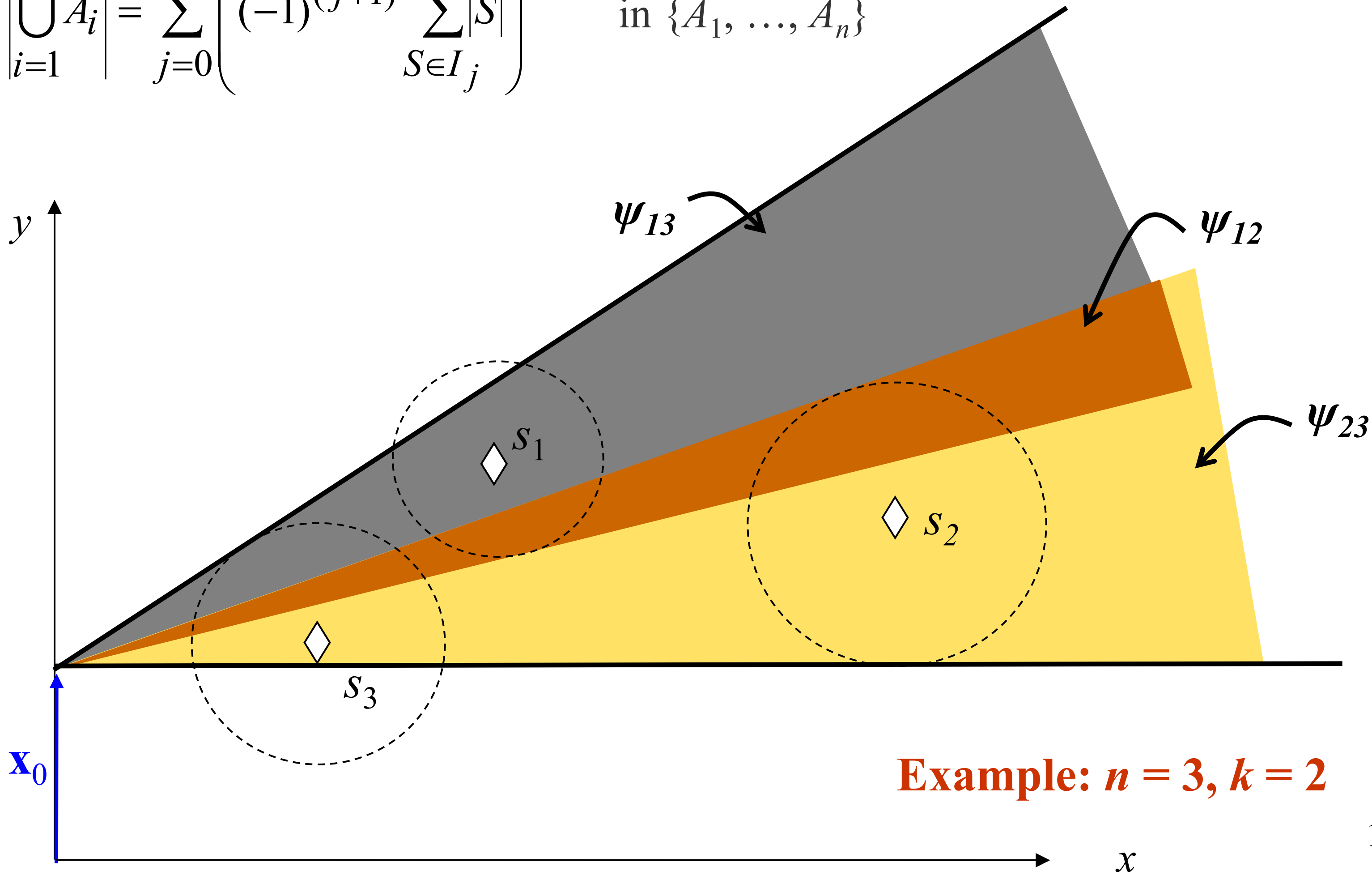
$$\hat{\mathbf{h}}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} = \mathbf{Q}_i^+ \hat{\mathbf{v}}_i, \quad \hat{\boldsymbol{l}}_i = \mathbf{Q}_i^- \hat{\mathbf{v}}_i.$$

Redundant Track Coverage ($n > k$)

Principle of union-exclusion → union of possible non-disjoint sets

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{j=0}^{n}\left((-1)^{(j+1)} \sum_{S\in I_j} |S|\right)$$

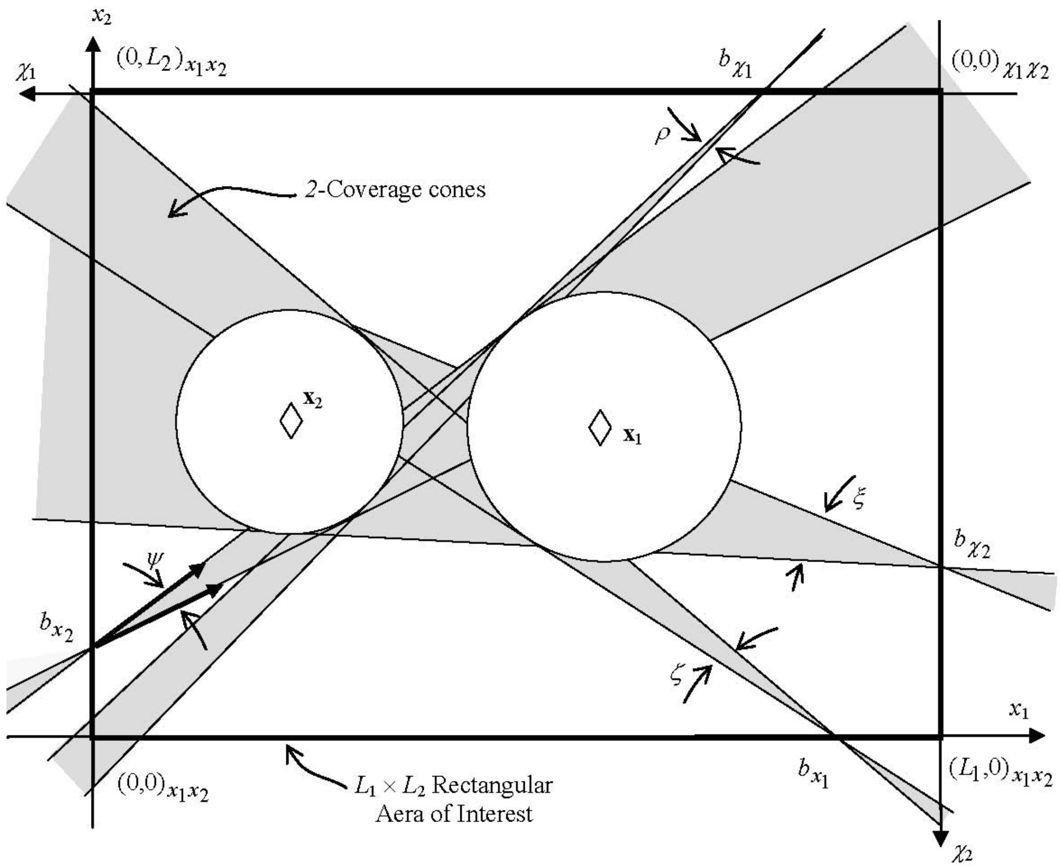
I_k = set of k fold intersections of members in $\{A_1, \ldots, A_n\}$



Example: $n = 3$, $k = 2$

Track Coverage in a Rectangular Workspace

- Opening angles: Lebesgue measure on the sets of line transversals.



K. C. Baumgartner and S. Ferrari, "A Geometric Approach to Analyzing Track Coverage in Sensor Networks", *IEEE Transactions on Computer*, Vol. 57, No. 8, pp. 1113-1128, August 2008.

Theorem: Probability of Track Detection

THEOREM 3.6. *The probability of detection of unobserved tracks for a set $\mathcal{P}$ of N pursuers with fields-of-view $\mathcal{D}_1, \dots, \mathcal{D}_N$, in a square game area $\mathcal{S}$ of dimensions $L \times L$, is a multivariate probability density function of the sensors' positions $\mathcal{X} = \{p_1, \dots, p_N\}$ given by a Lebesgue measure on this union,*

$$P_{\mathcal{S}}^k(\mathcal{X}) = \frac{\delta b}{4\pi L} \sum_{\ell=1}^{L/\delta b} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \le i_1 < \cdots < i_j \le m} [\psi(D_p^{i_{1,j}}, b_y^\ell) + \varphi(D_p^{i_{1,j}}, b_{y'}^\ell)]$$

$$+ \frac{\delta b}{4\pi L} \sum_{\ell=0}^{(L/\delta b - 1)} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \le i_1 < \cdots < i_j \le m} [\zeta(D_p^{i_{1,j}}, b_x^\ell) + \rho(D_p^{i_{1,j}}, b_{x'}^\ell)]$$

$$\text{with} \quad m = \frac{N!}{(N-k)!k!}, \qquad D_p^{i_{1,j}} \equiv \{D_k^{i_1} \cup \cdots \cup D_k^{i_j}\},$$

where the summation $\sum_{1 \le i_1 < \cdots < i_j \le m}$ is a sum over all the $[m!/(m-j)!\, j!]$ distinct integer j-tuples $(i_1, \dots, i_j)$ satisfying $1 \le i_1 < \cdots < i_j \le m$, $D_k^{i_l}$ denotes the i_lth k-subset of D, and $D_p^{i_{1,j}}$ is a p-subset of D, with $k \le p \le n$.

S. Ferrari, R. Fierro, B. Perteet, C. Cai, and K. C. Baumgartner, "A Geometric Optimization Approach to Detecting and Intercepting Dynamic Targets using Mobile Sensor Network," *SIAM Journal on Control and Optimization*, Vol. 48, No. 1, pp. 292-320, 2009.

Optimal Control Problem

Sensing performance metric to be optimized w.r.t. $\mathbf{c}[\cdot]$, $\mathbf{x}(\cdot)$

$$J = \phi\left[\mathbf{x}\left(t_f\right), t_f\right] + \int_{t_0}^{t_f} \left\{ w_T \cdot P_S^k\left[\mathbf{x}(t), \mathbf{r}(t), \mathbf{u}(t)\right] + w_E \cdot \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t) \right\} dt,$$

subject to sensor network dynamics,

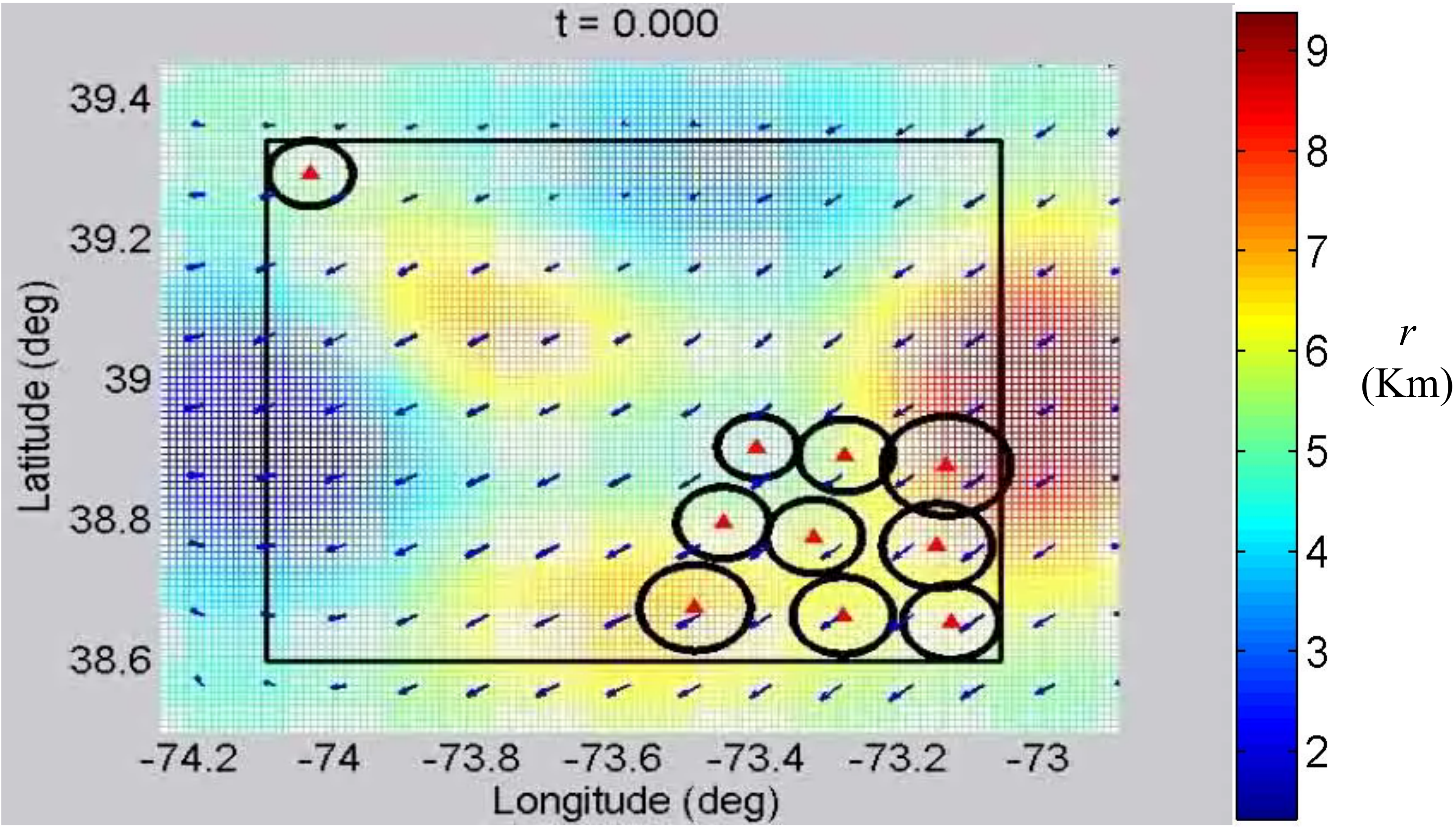
$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t), t\right]$$

and subject to equality and inequality mission constraints

$$\mathbf{g}\left[\mathbf{x}(t), \mathbf{u}(t)\right] \geq 0$$

- n vehicles controls: $\mathbf{u}(t) = \mathbf{c}[\mathbf{x}(t), t]$
- n vehicles positions: $\mathbf{x}(t)$
- Environmental and sensing parameters: $\mathbf{p}(t)$

Optimal Sensor Network Trajectories



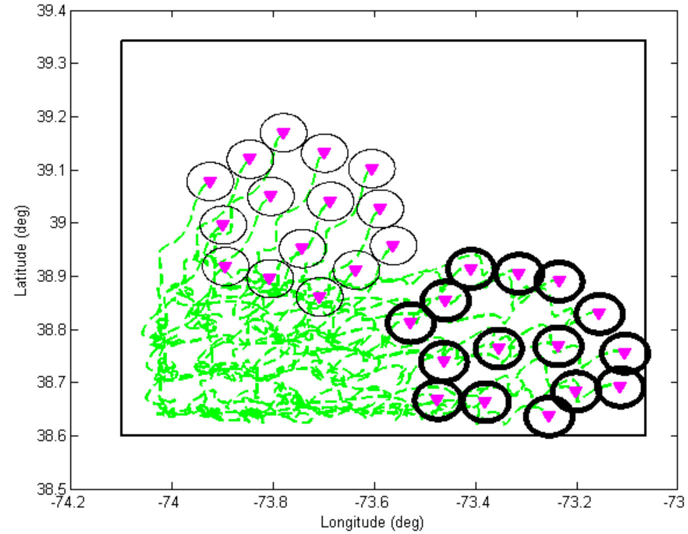
K. C. Baumgartner, S. Ferrari, and A. Rao, "Optimal Control of a Mobile Sensor Network for Cooperative Target Detection," *IEEE Journal of Oceanic Engineering*, in press, available online.

LISC
LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

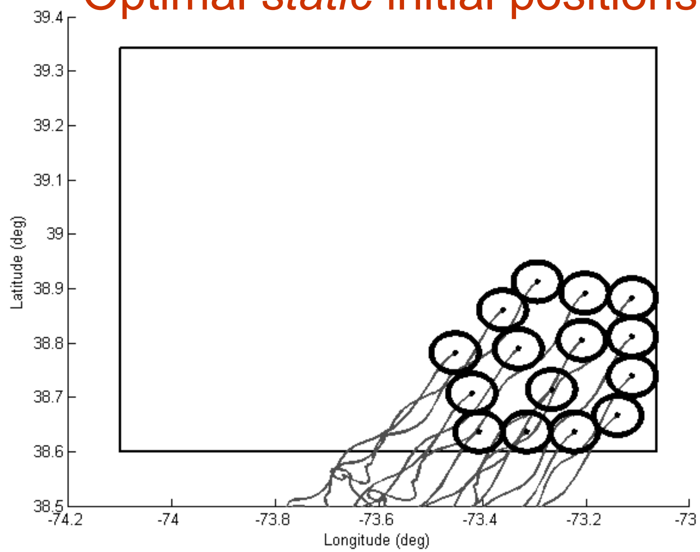
Comparison of Optimal Sensor Deployments

- $L_1 \times L_2 =$ 90 x 82.5 Km
- $\Delta T =$ **9 days**
- $n = 15$, $k = 3$,
- $r = 4$ Km
- ▼ UUV ● Buoys

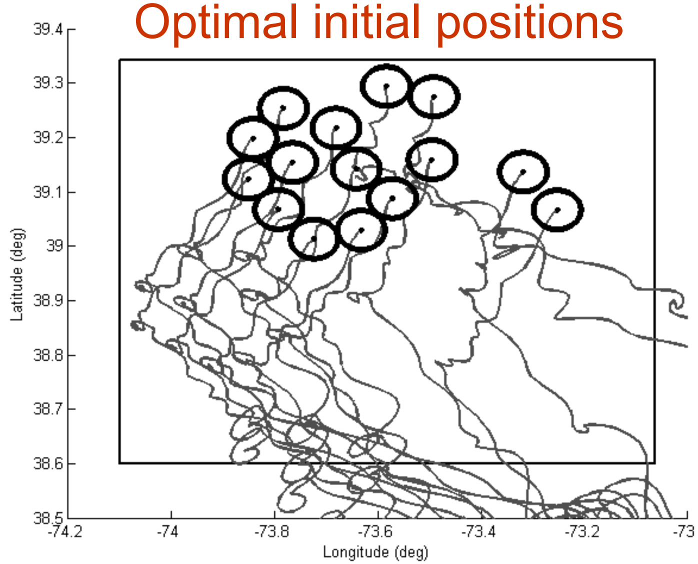
Optimal control and initial positions



Optimal *static* initial positions

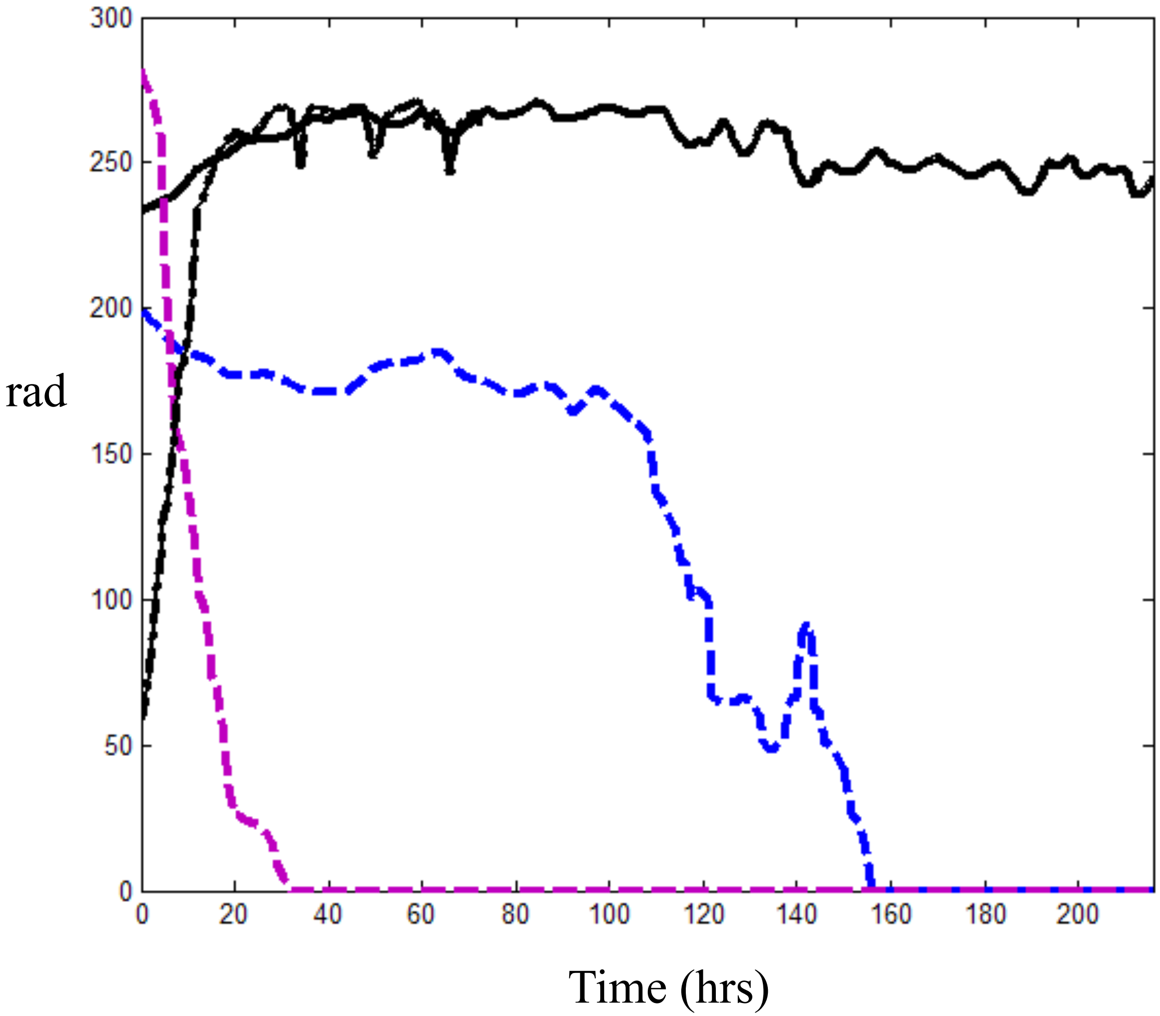


Optimal initial positions



Sensing Performance Comparison

Track Coverage Time History



Legend:

- Solid black line: Optimal control and initial conditions ($\mathbf{x}_0^*$)
- Black dash-dot line: Optimal control, given $\mathbf{x}_0$
- Blue dashed line: Optimal initial conditions ($\mathbf{x}_0^*$), no control (buoys)
- Magenta dashed line: Static optimal

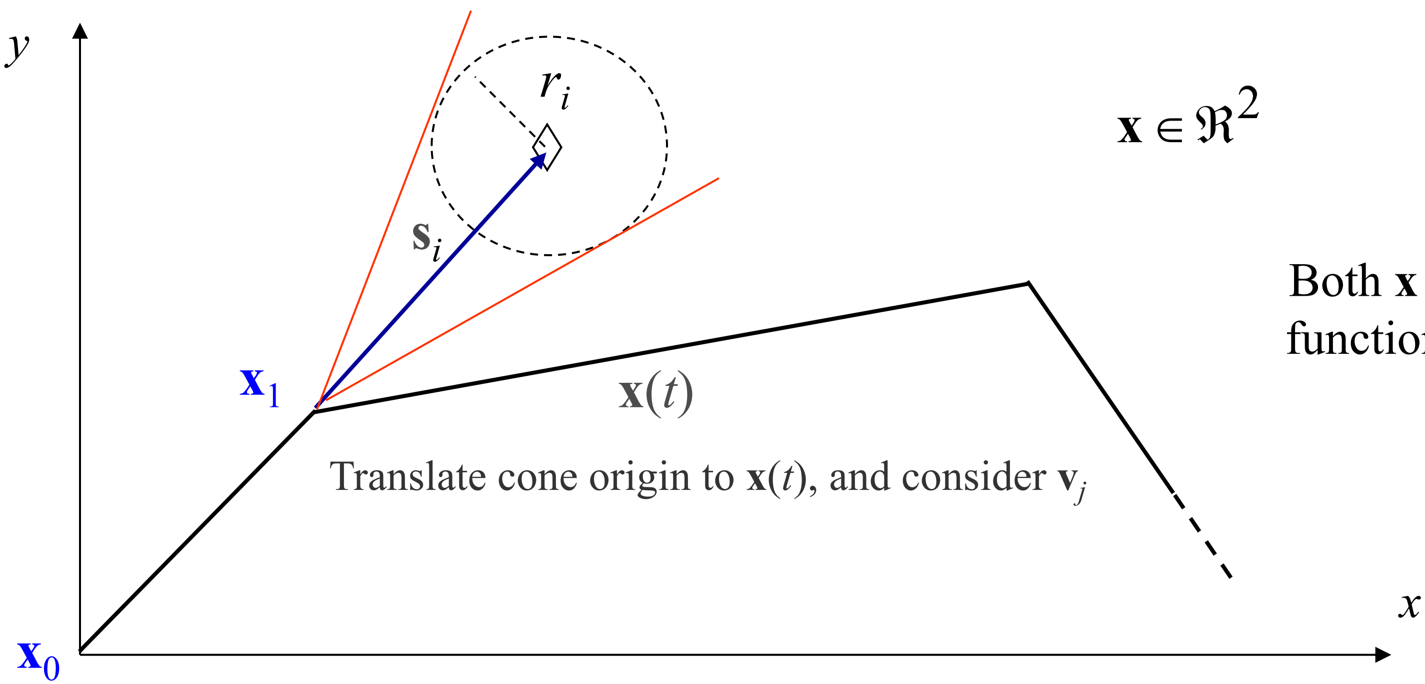
Extension to Maneuvering Targets

Track Coverage for Maneuvering Targets

Markov Target Model:

$$\mathbf{x}(t) = \mathbf{x}_j + \mathbf{v}_j(t - t_j)[\cos\theta_j \ \sin\theta_j]^T, \qquad \mathbf{x}_j = \mathbf{x}(t_j), \ t_j \le t < t_{j+1}, \ j = 1, 2, \ldots$$

Markov motion parameter values: $\{\mathbf{x}_j, \mathbf{v}_j, \theta_j\}_{j = 1, 2, \ldots}$



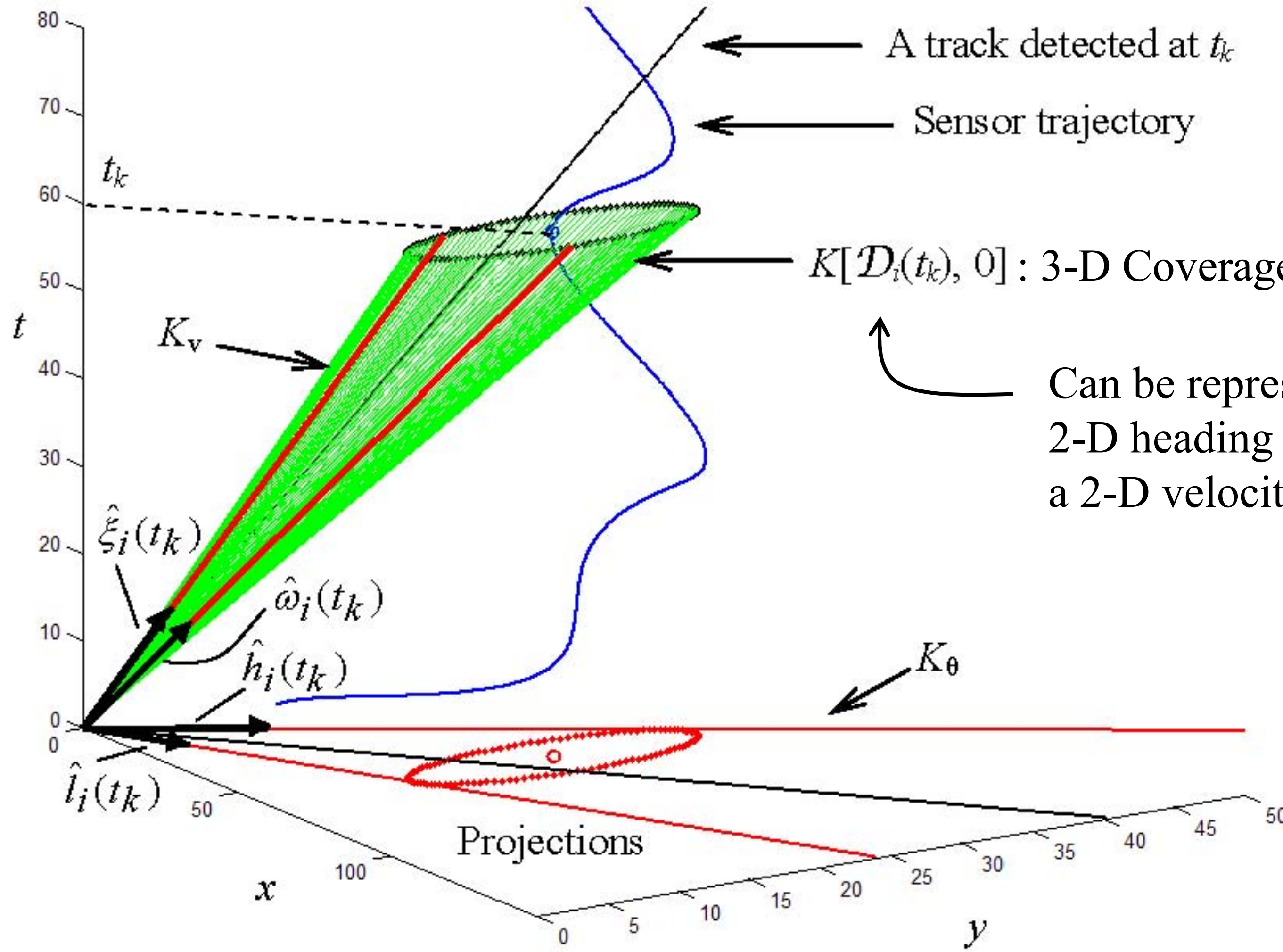
$\mathbf{x} \in \Re^2$

Both $\mathbf{x}$ and $\mathbf{s}_i$ are functions of time

S. Ferrari, R. Fierro, and D. Tolilc "A Geometric Optimization Approach to Tracking Maneuvering Targets Using a Heterogeneous Mobile Sensor Network," *Proc. Conference on Decision and Control*, Cancun, MX, 2009.

Spatio-Temporal (ST) Coverage Cones

Markov trajectory amounts to straight-line segments in the space $\Re^2 \times [t_0, t_f]$



Can be represented by a 2-D heading cone K_θ, and a 2-D velocity cone K_v

Heading and Velocity Cones

The 3D coverage cone $K[D_i, \mathbf{x}_j]$ can be represented by its projection onto the xy-plane, K_θ , and by K_v, which denotes its intersection with the

velocity plane: $(\sin\theta_j)x + (\cos\theta_j)y = 0$

K_v is finitely generated by the unit vectors,

$$\hat{\xi}_i(t) = \begin{bmatrix} \cos\theta_j \sin[\pi/2 - \eta_i(t)] \\ \sin\theta_j \sin[\pi/2 - \eta_i(t)] \\ \cos[\pi/2 - \eta_i(t)] \end{bmatrix} \quad \text{and} \quad \hat{\omega}_i(t) = \begin{bmatrix} \cos\theta_j \sin[\pi/2 - \mu_i(t)] \\ \sin\theta_j \sin[\pi/2 - \mu_i(t)] \\ \cos[\pi/2 - \mu_i(t)] \end{bmatrix}$$

where:

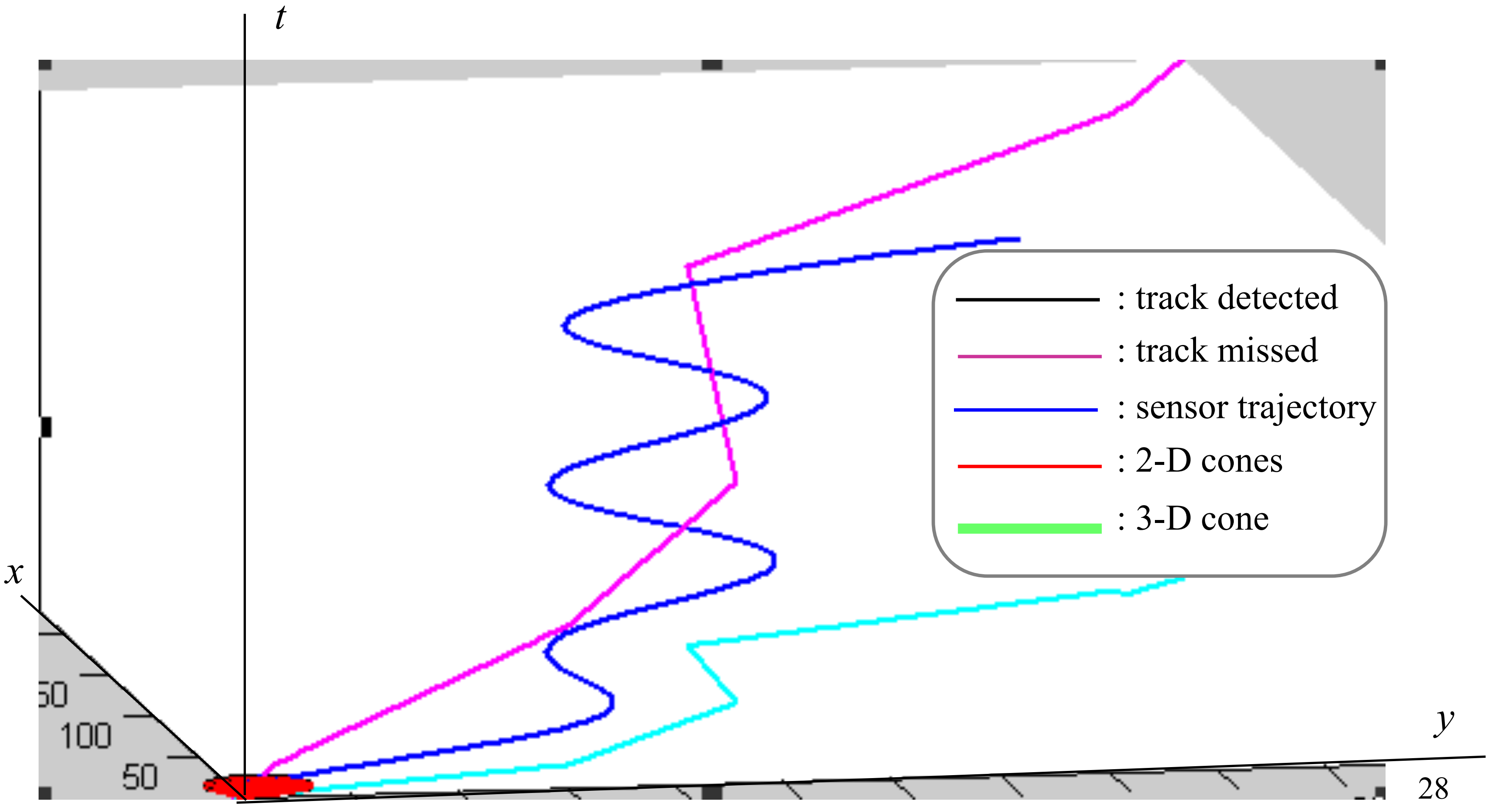
$$\eta_i, \mu_i = \tan^{-1}\left\{ t \left[x_i \cos\theta_j + y_i \sin\theta_j \mp \sqrt{r_i^2 - (x_i \sin\theta_j + y_i \cos\theta_j)^2} \right]^{-1} \right\}$$

K_θ and K_v are a function of the sensor coordinates $x_i(t)$ and $y_i(t)$ and of the Markov motion parameter values: $\{\mathbf{x}_j, \mathbf{v}_j, \theta_j\}_{j = 1, 2, \ldots}$

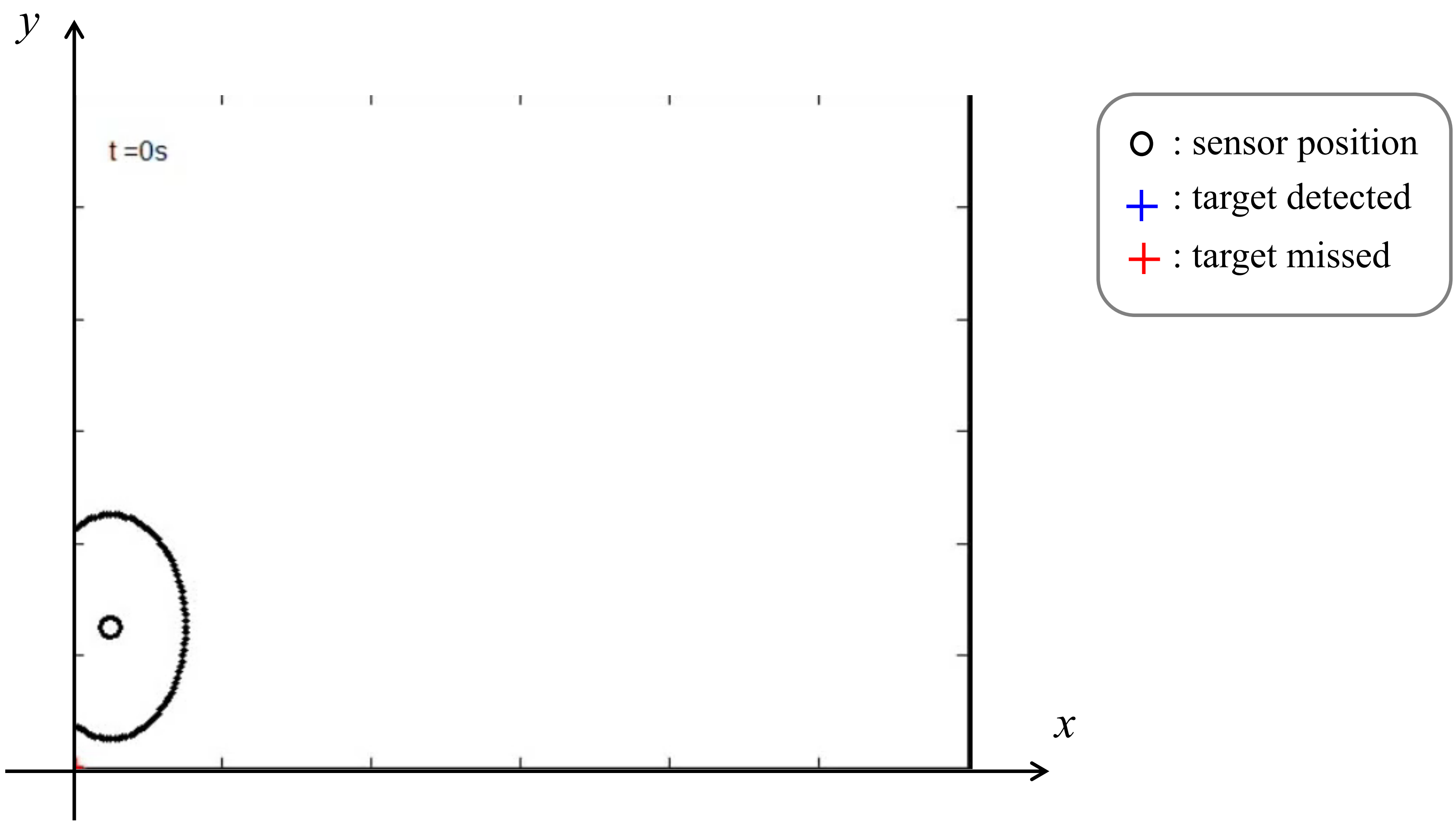
LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

Coverage Cone Interpretation

At every time $t_j \leq t < t_{j+1}$, the 3D spatio-temporal cone $K[D_i, \mathbf{x}_j]$ contains all Markov pwl. tracks that originate at $\mathbf{x}_j$ and are detected by the i^{th} sensor at $\mathbf{x}_i$.



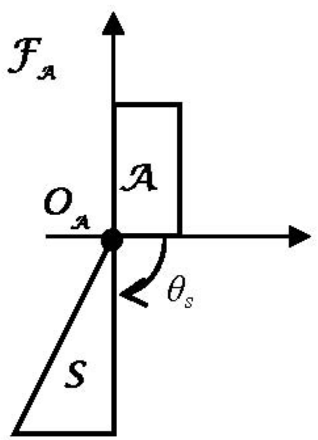
Example: Application of ST cone at t = 60 s

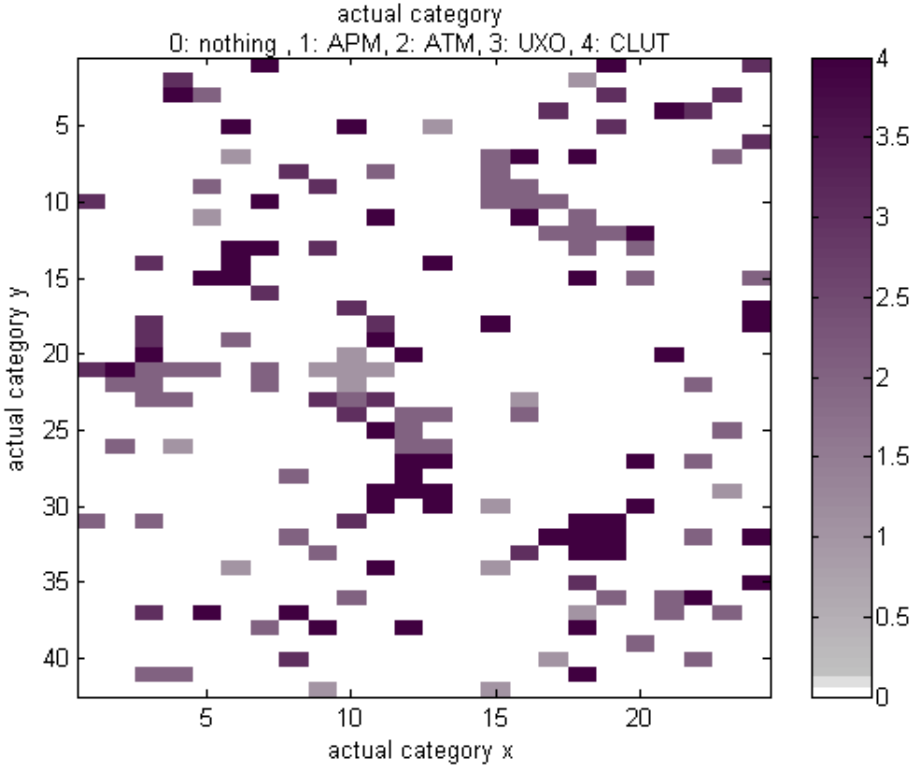


S. Ferrari, R. Fierro, and D. Tolic "A Geometric Optimization Approach to Tracking Maneuvering Targets Using a Heterogeneous Mobile Sensor Network," *Conference on Decision and Control*, Shanghai, China, 2009, to appear.

Treasure Hunt Problem

For a given layout $\mathcal{W} \subset \Re^2$ with r targets and n obstacles and a given joint probability mass function $P(y, m_1, \ldots, m_r)$ of an hypothesis variable, y, and r measurements, find the obstacle-free path that minimizes the distance traveled by a robot $\mathcal{A}$, between two configurations q_0 and q_f and maximizes the information value for a sensor with field of view $\mathcal{S}$, installed on $\mathcal{A}$.





S. Ferrari and C. Cai, "Information-Driven Search Strategies in the Board Game of CLUE®," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, Vol. 39, No 3, June 2009.

Definitions

Information Value: Expected Entropy Reduction (EER):

$$\Delta H(X^k; Z^k \mid Z^1,\ldots,Z^{k-1},\lambda^k) \equiv H(X^k \mid Z^1,\ldots,Z^{k-1},\lambda^k) - \sum_{Z^k} H(X^k \mid Z^1,\ldots,Z^k,\lambda^k)\, p(Z^k \mid Z^1,\ldots,Z^{k-1},\lambda^k)$$

Advantage: additive, symmetric, non-myopic, ..

Definition 4.1 (Field of View): The field of view of a sensor mounted on $\mathcal{A}$ is a closed and bounded subset $\mathcal{S}(q) \subset \mathcal{W}$ such that the measurement set of a target located at any point $p \in \mathcal{S}(q)$ can be obtained by the sensor when the robot occupies the configuration $q \in \mathcal{C}$.

Definition 4.2 (C-Target): The target $\mathcal{T}_i$ in $\mathcal{W}$ maps in the robot's configuration space, $\mathcal{C}$, to the C-target region $\mathcal{CT}_i = \{q \in \mathcal{C} \mid \mathcal{S}(q) \cap \mathcal{T}_i \neq \varnothing\}$.

Definition 4.3: A *void cell* is a convex polygon κ in $\mathcal{C}_{free}$ with the property that none of the targets are observable from any of the configurations in κ.

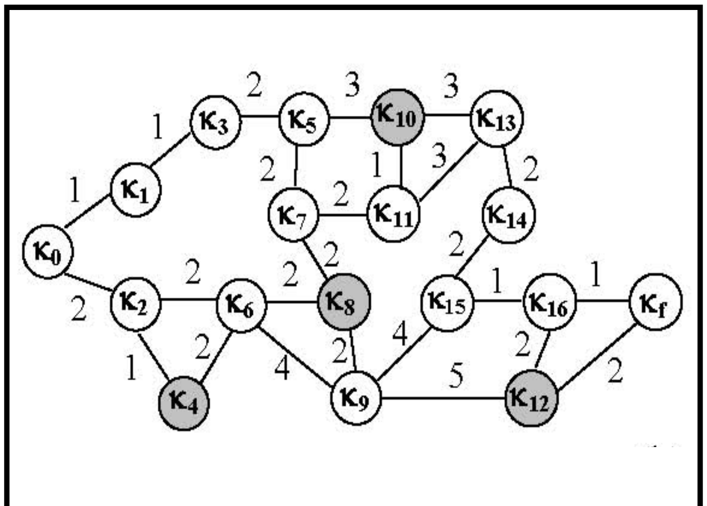
Definition 4.4: An *observation cell* is a convex polygon $\bar{\kappa}$ in $\mathcal{C}_{free}$ with the property that every configuration in $\bar{\kappa}$ enables a non-empty set of measurements $Z(\bar{\kappa}) = \{M_i \mid q \in \bar{\kappa},\ q \in \mathcal{CT}_i\}$.

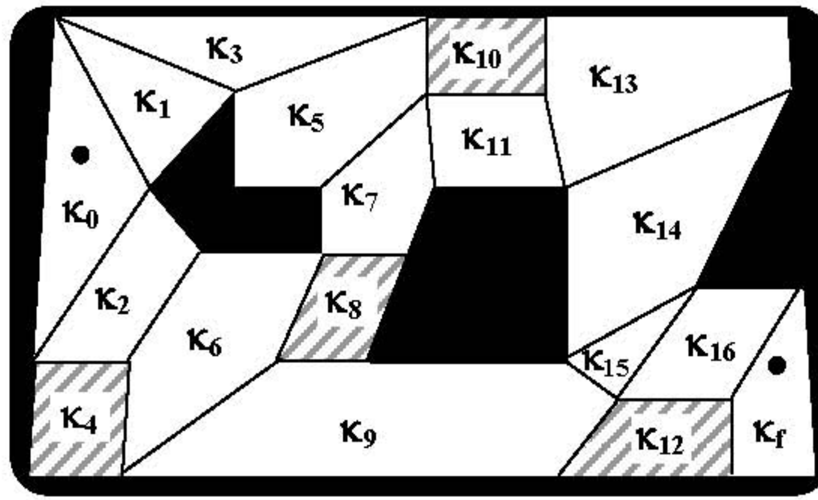
S. Ferrari and C. Cai, "Information-Driven Search Strategies in the Board Game of CLUE®," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, Vol. 39, No 3, June 2009.

Sensor Path Planning

- Develop a cell decomposition method that accounts for the geometries of the targets and the sensor FOV

New **cell decomposition:** $\mathcal{W}$



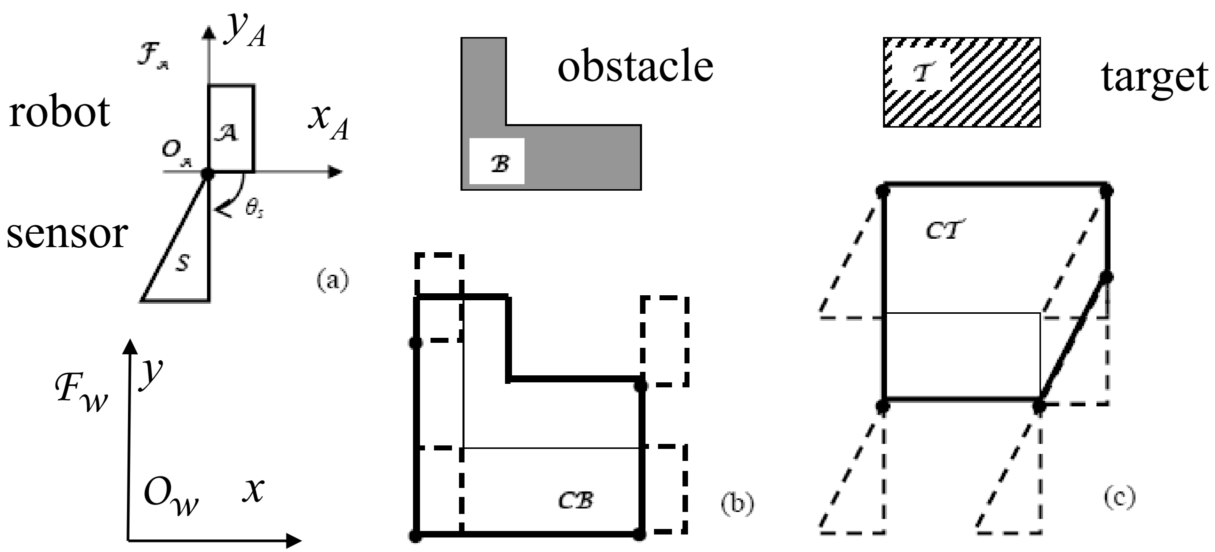


Connectivity graph $\mathcal{G}$ with **observation cells** labeled in grey and void cells labeled in white

Approximate-and-Decompose Method

- Configuration q of robot $A(q)$: $q = (x, y, \theta)$ with orientation θ
- Configuration space, C : the space of all the possible configurations of A

Examples of C-obstacle, *CB*, and C-target, *CT*, obtained for a sensor with FOV, $S(q)$:



- Bounding rectangloid approximation $\mathcal{RB}$ of $\mathcal{CB}$
- Bounded rectangloid approximation $\mathcal{R'T}$ of $\mathcal{CT}$

Decomposition Procedure

1. Decompose the range of robot orientations $[\theta, \theta']$ into non-overlapping intervals

   $I_u = [\gamma_u , \gamma_{u+1}]$, $\quad \kappa^u = [x_\kappa, x'_\kappa] \times [y_\kappa, y'_\kappa] \times I_u$

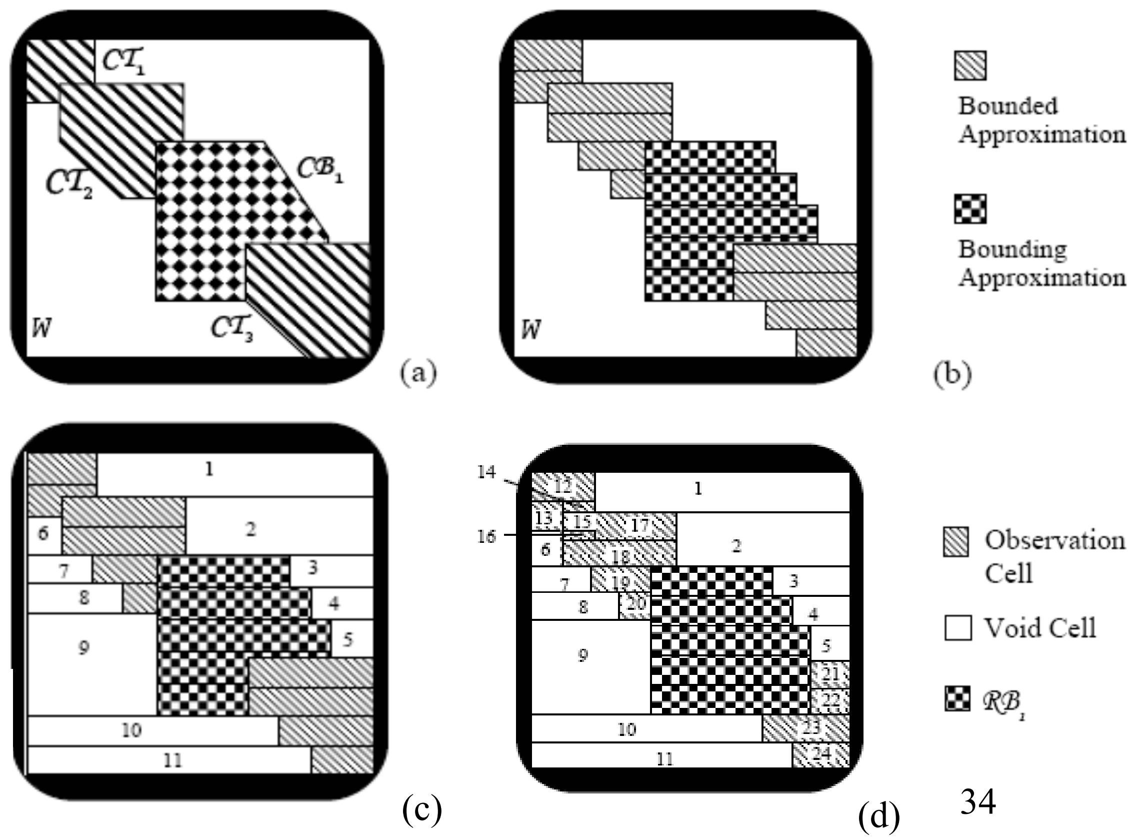
2. Compute $CB_j[k^u]$ and $CT_i[k^u]$, then $RB_j[k^u]$ and $R'T_i[k^u]$

3. Obtain void cells decomposition $\mathcal{K}_{void}$ of void configuration space C^u_{void}

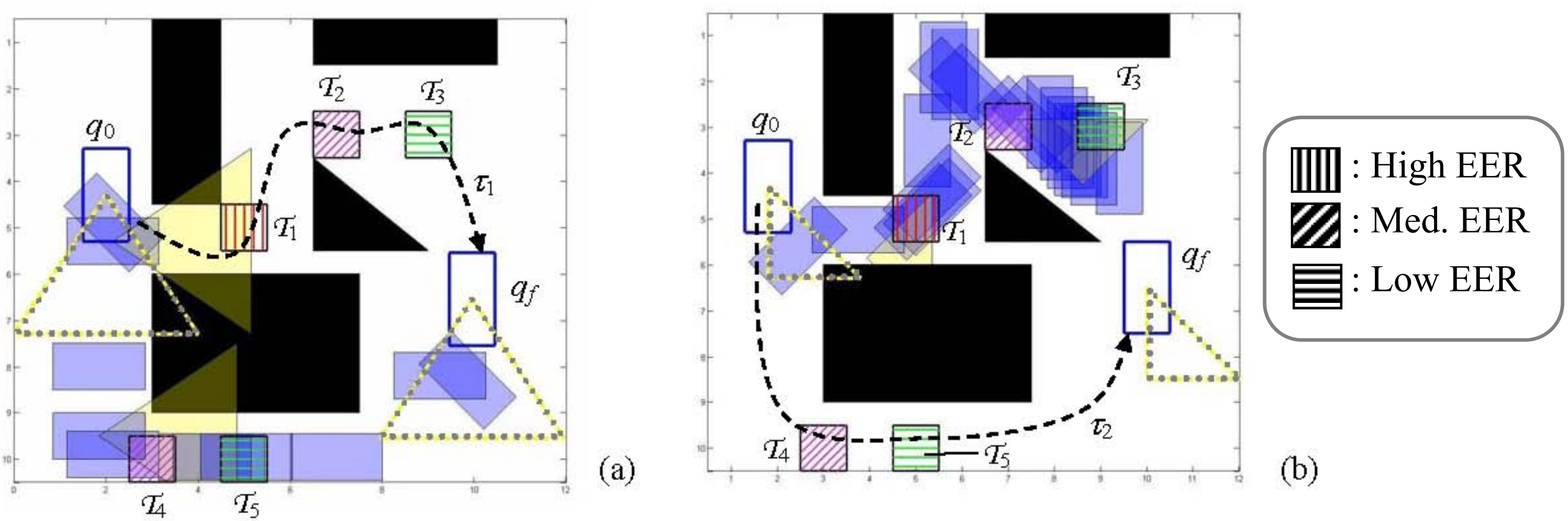
$$\mathcal{C}^u_{void} = \kappa^u \setminus \{\bigcup_{j=1}^{n} \mathcal{RB}_j[\kappa^u] \cup \bigcup_{i=1}^{r} \mathcal{R}'\mathcal{T}_i[\kappa^u]\}$$

4. Obtain observation cells decomposition $\mathcal{K}_z$ of C^u_z

$$\mathcal{C}^u_z = \bigcup_{i=1}^{r} \mathcal{R}'\mathcal{T}_i[\kappa^u] \setminus \bigcup_{j=1}^{n} \mathcal{RB}_j[\kappa^u]$$



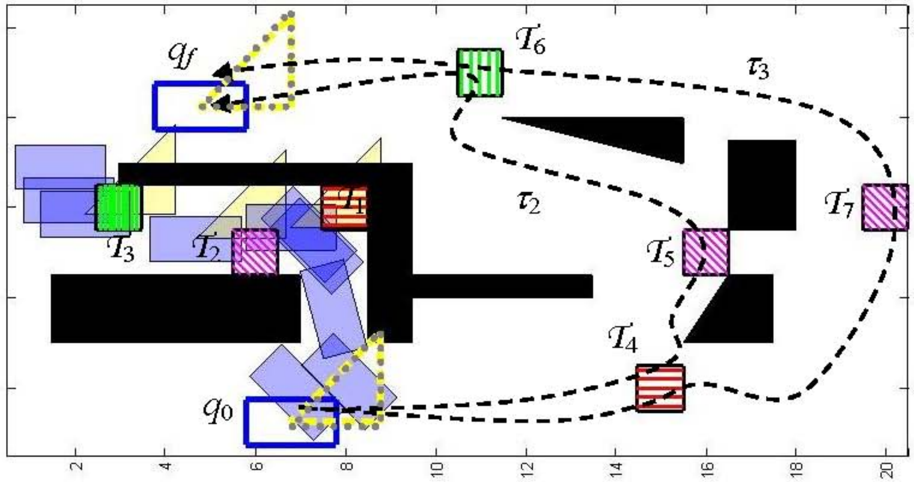
Influence of Sensor Geometry



Sensor geometry S_1				
Path	Targets	D_{tot}	B_{tot}	η_{CL}
τ^*	$\mathcal{T}_1, \mathcal{T}_4, \mathcal{T}_5$	21.3	14.4	0.0516
τ_1	$\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$	22.6	14.6	0.0381

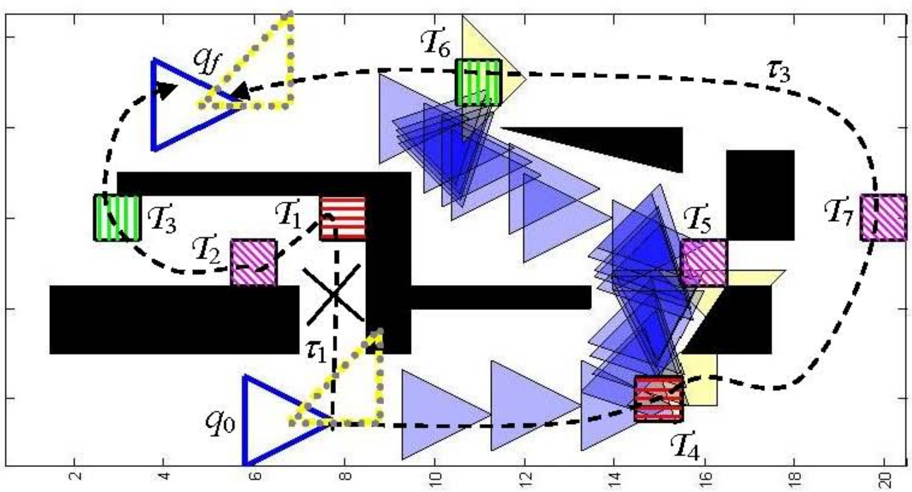
Sensor geometry S_2				
Path	Targets	D_{tot}	B_{tot}	η_{CL}
τ^*	$\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$	22.6	14.6	0.0381
τ_2	$\mathcal{T}_4, \mathcal{T}_5$	15.1	4.67	0.0226

Influence of Robot Geometry



(a)

Robot geometry $\mathcal{A}_1$			
Path	$\tau^* = \tau_1$	τ_2	τ_3
Targets	$\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$	$\mathcal{T}_4$, $\mathcal{T}_5$, $\mathcal{T}_6$	$\mathcal{T}_4$, $\mathcal{T}_6$, $\mathcal{T}_7$
D_{tot}	20.47	27.50	32.95
B_{tot}	9.93	9.71	9.91

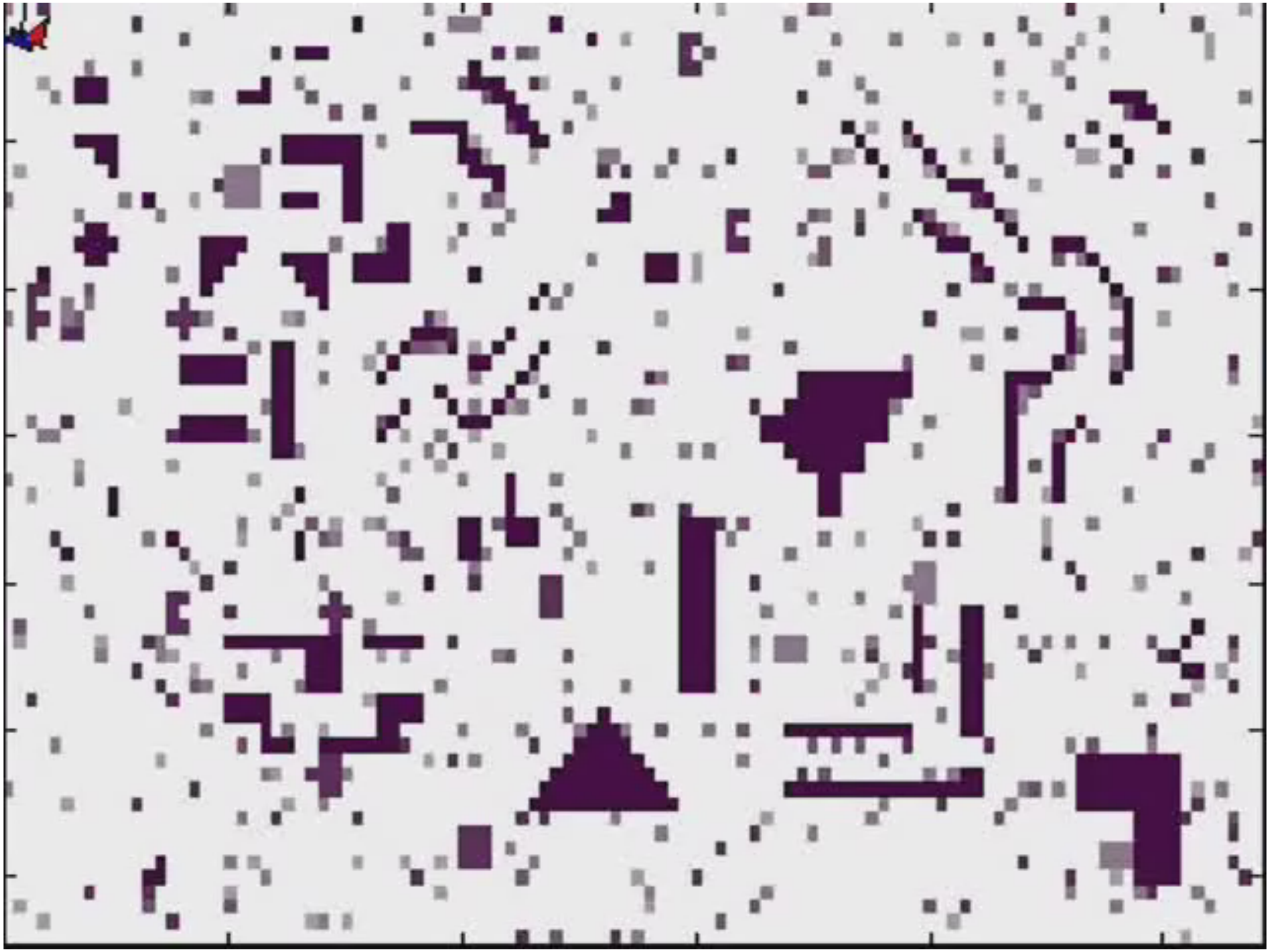


(b)

Robot geometry $\mathcal{A}_2$		
Path	$\tau^* = \tau_2$	τ_3
Targets	$\mathcal{T}_4$, $\mathcal{T}_5$, $\mathcal{T}_6$	$\mathcal{T}_4$, $\mathcal{T}_6$, $\mathcal{T}_7$
D_{tot}	30.94	33.59
B_{tot}	9.71	9.91

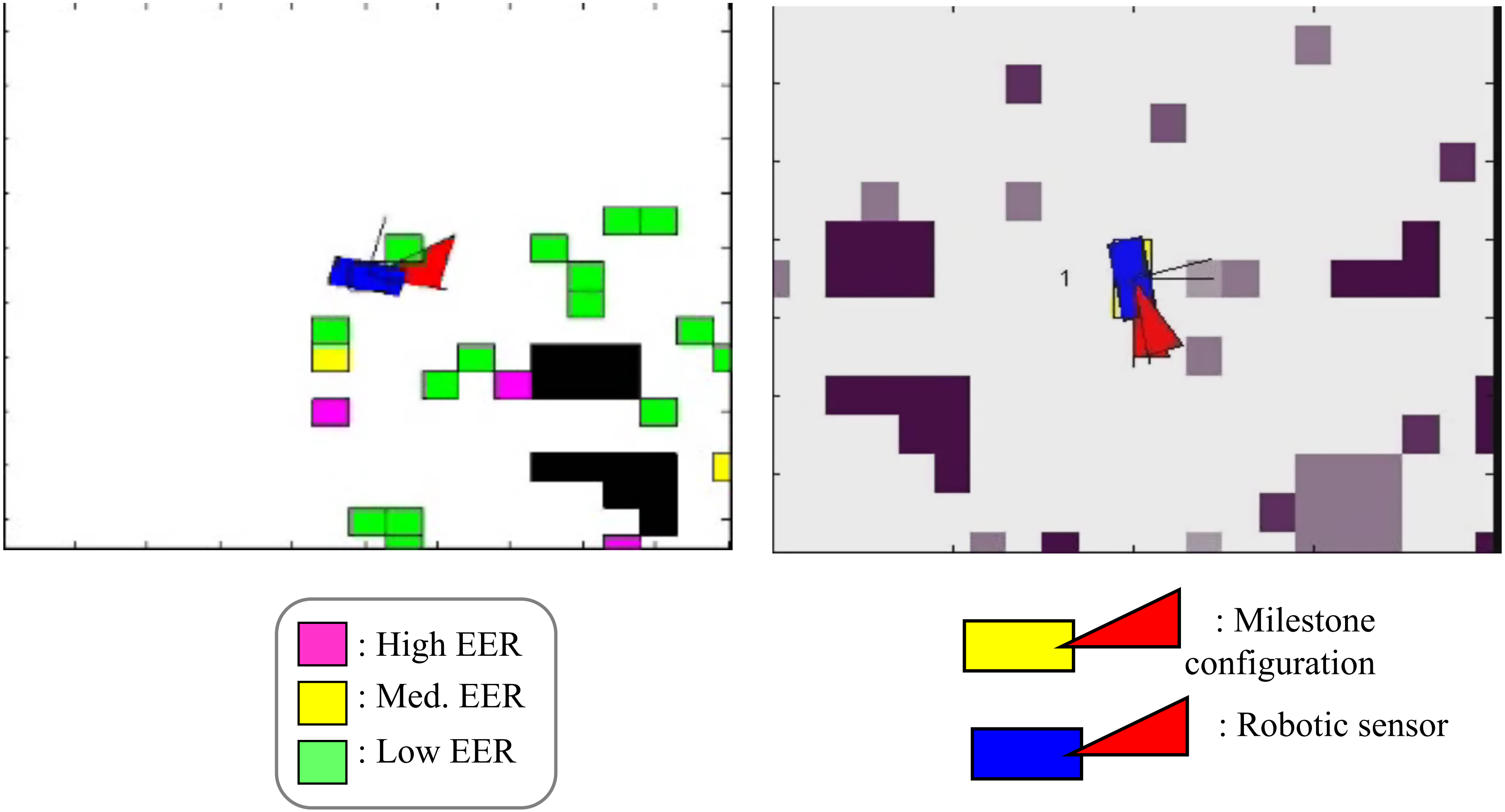
C. Cai and S. Ferrari, "Information-Driven Sensor Path Planning by Approximate Cell Decomposition," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, Vol. 39, No 3, 2009.

Large Workspace (Information Roadmap)



LISC
LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

Information Value and Milestones



G. Zhang, S. Ferrari, and M. Qian, ”Information Roadmap Method for Robotic Sensor Path Planning,” *Journal of Intelligent and Robotic Systems*, Vol. 56, pp. 69-98, 2009.

Performance Comparison

Efficiency Metric	Method				
	Optimal Strategy, σ^*	Shortest Path (σ^* Improvement)	Complete Coverage (σ^* Improvement)	Random Search (σ^* Improvement)	Grid Search (σ^* Improvement)
η_N	0.4610	0.3053 (51.0%)	0.2683 (71.8%)	0.1441 (219.9%)	0.2321 (98.6%)
η_y	0.0595	0.0407 (46.2%)	0.0055 (981.8%)	0.0114 (421.9%)	0.0122 (387.7%)
η_{CL}	0.0446	0.0157 (184.1%)	0.0153 (191.5%)	0.0098 (355.1%)	0.0133 (235.3%)
η_H	0.0599	0.0330 (81.5%)	0.0410 (46.1%)	0.0244 (145.5%)	0.0343 (74.6%)

C. Cai and S. Ferrari, "Information-Driven Sensor Path Planning by Approximate Cell Decomposition," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, Vol. 39, No. 3, June 2009.

Marco Polo Problem

Given a set $\mathcal{P}$ of N pursuers and a set $\mathcal{T}$ of M targets moving within an obstacle-populated game area $\mathcal{S}$, find a set of policies which maximize the total sensing reward, and minimize the total time required to capture targets in $\mathcal{T}$ that have been positively detected.

Assumptions

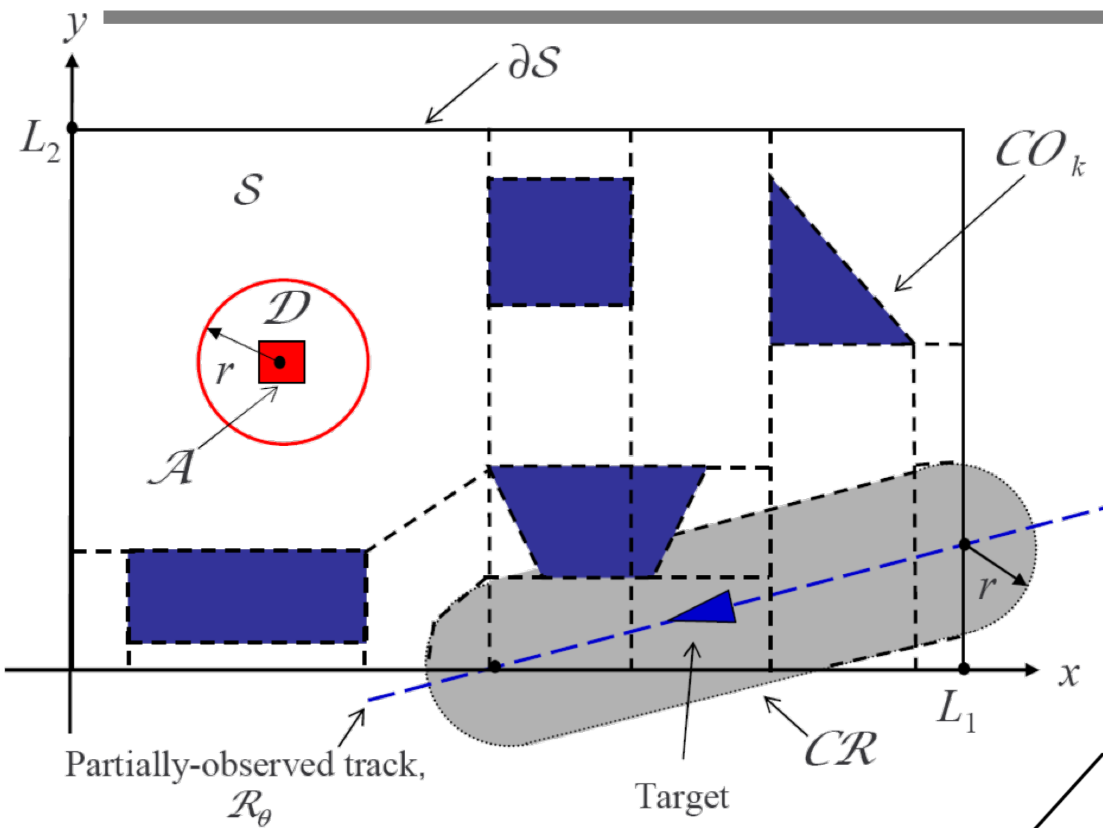
1. Targets travel in straight lines with constant velocity
2. Targets are observed intermittently by multiple sensors that measure only position
3. Pursuers have two modes: *detection* and *pursuit*
4. Tracks may be *unobserved*, *partially-observed* ($< k$), or *fully-observed* ($\geq k$)
5. Pursuers can always move faster than the targets

Objectives

- Maximize the probability of detecting unobserved tracks
- Maximize the probability of detecting partially-observed tracks
- Minimize the distance traveled to detect and capture targets

LISC
LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

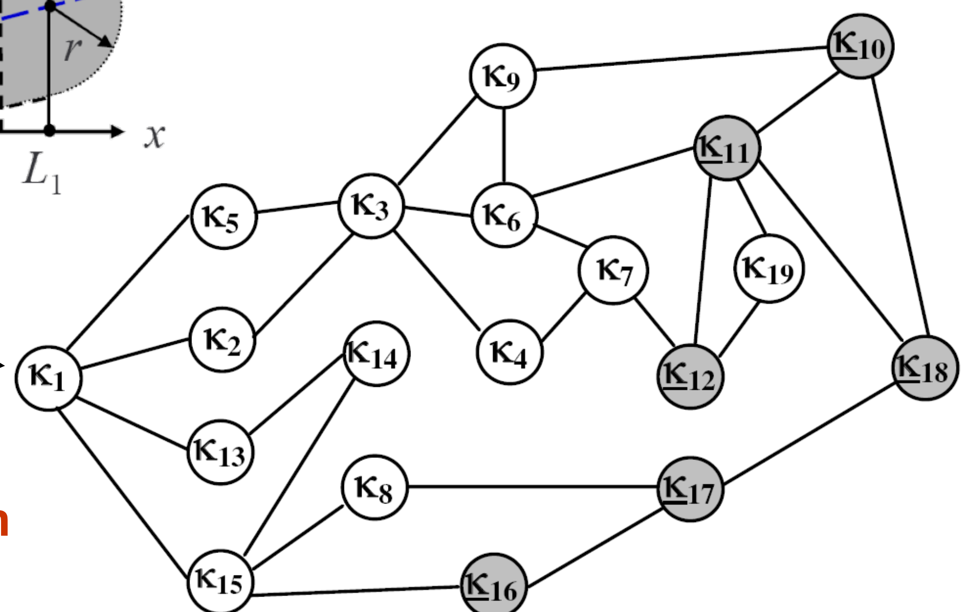
Sensor Roadmap



Example of cell decomposition

- Rectangular workspace (L_1 x L_2)
- Four C-obstacles
- One target with 2 < *k* detections
- One sensor with range *r*

Obstacle free cells



Connectivity Graph

Observation cells

Performance Analysis

THEOREM 4.1. *The pursuit-evasion game in Problem 2.1 is guaranteed to terminate provided,*

$$N \geq N_{\min} = \frac{1}{2}\left[\left\lfloor \frac{2L}{r} \right\rfloor + k - 1 + \left| \left\lfloor \frac{2L}{r} \right\rfloor - k + 3 \right|\right] \tag{4.1}$$

and requires a time,

$$t_f \leq T_u = \frac{(\sqrt{2}L - 2r)}{V_{\mathcal{T}_{\min}}} + \left[\left\lfloor \frac{(k-2)M}{N} \right\rfloor + 1\right] \frac{(\sqrt{2}L - r)}{\bar{V}_p} \tag{4.2}$$

$$+ \frac{r}{(V_{p_{\max}}^2 - \bar{V}_{\mathcal{T}}^2)} + \left\lfloor \frac{M}{N} \right\rfloor \frac{(\bar{V}_{\mathcal{T}} + \sqrt{2V_{p_{\max}}^2 - \bar{V}_{\mathcal{T}}^2})}{(V_{p_{\max}}^2 - \bar{V}_{\mathcal{T}}^2)^2} L \tag{4.3}$$

to capture all M targets in $\mathcal{T}$. If the network contains at least

$$N_\ell = \frac{1}{2}\left[\ell \left\lfloor \frac{2L}{r} \right\rfloor - 4\ell(\ell - 1) + (k-2)M + \left| \ell \left\lfloor \frac{2L}{r} \right\rfloor - 4\ell(\ell - 1) - (k-2)M \right|\right] \tag{4.4}$$

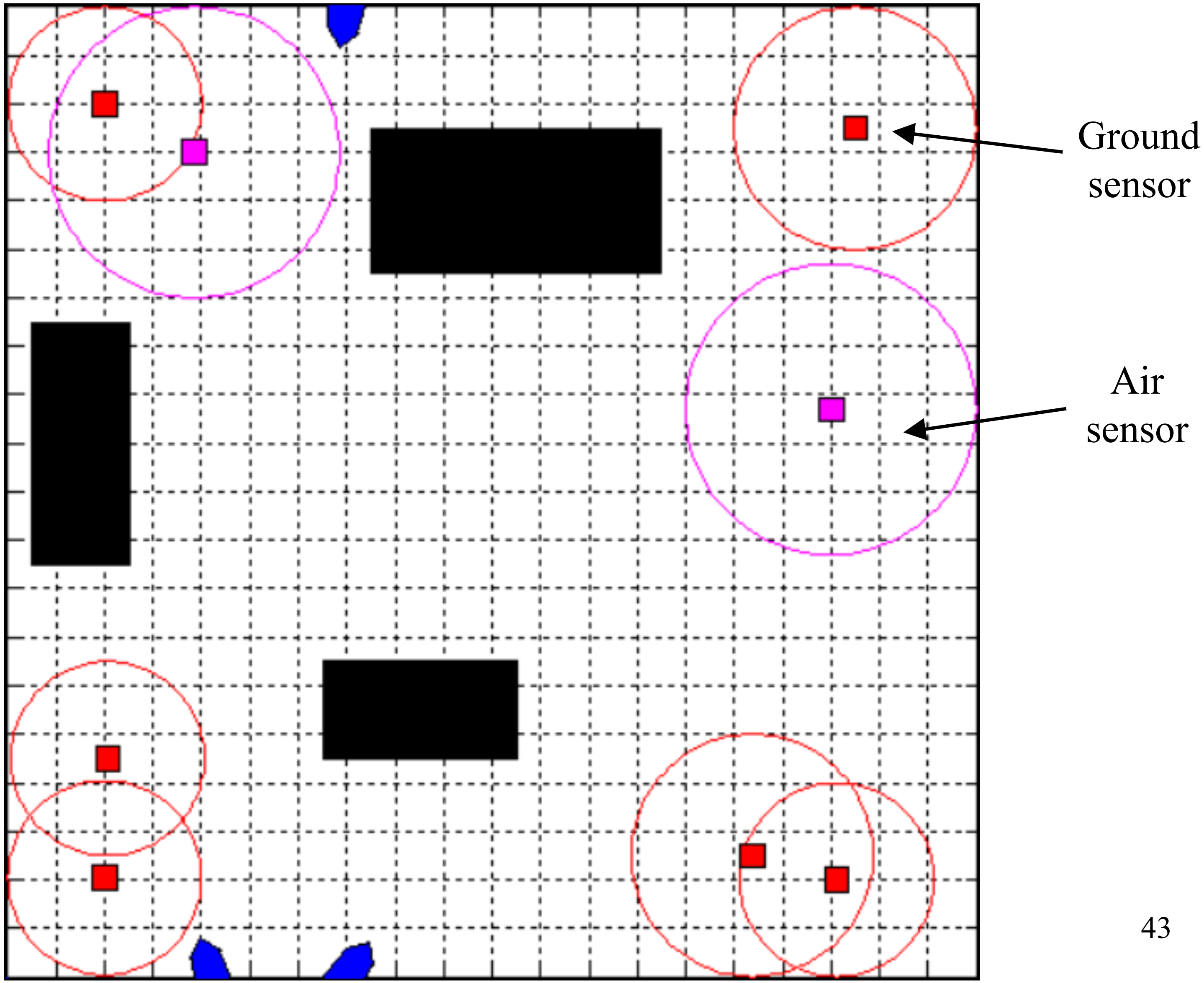
sensors, with $\ell = 1, \ldots, \lfloor L/4r \rfloor$, then all targets in $\mathcal{T}$ can be captured in a time,

$$t_f \leq T_\ell = \frac{1}{V_{\mathcal{T}_{\min}}}\left\{\frac{\sqrt{2}}{2}L - 2\sqrt{2}r(\ell - 1) + \left| 2r[1 + \sqrt{2}(\ell - 1)] - \frac{\sqrt{2}}{2}L \right|\right\}$$

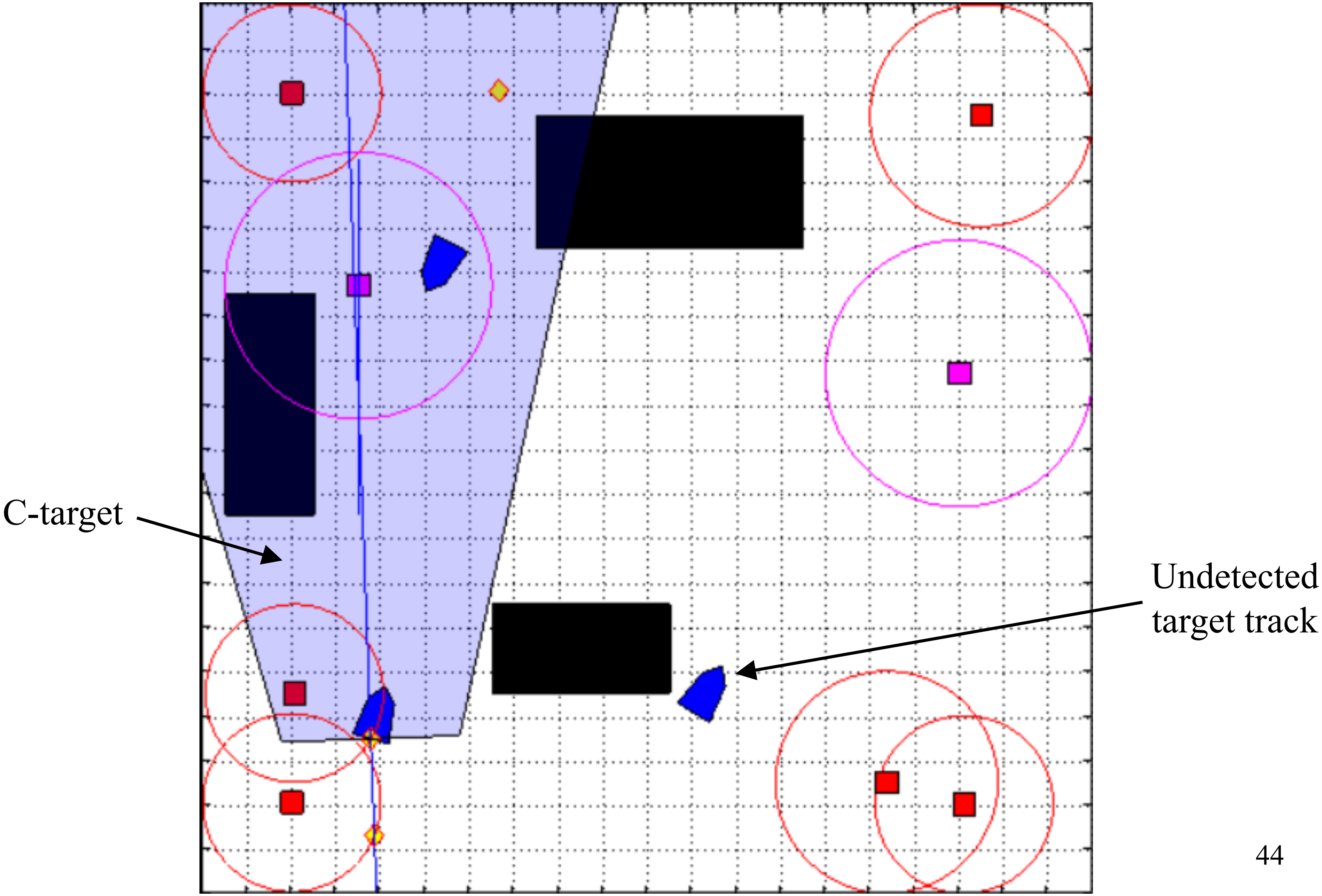
$$+ \frac{(\sqrt{2}L - r)}{\bar{V}_p} + \frac{r}{(V_{p_{\max}} - \bar{V}_{\mathcal{T}})} \tag{4.5}$$

and the game terminates in $t_f \leq T_\ell \leq T_u$, where $T_\ell = T_u$ when $\ell = 1$ and $k = 3$.

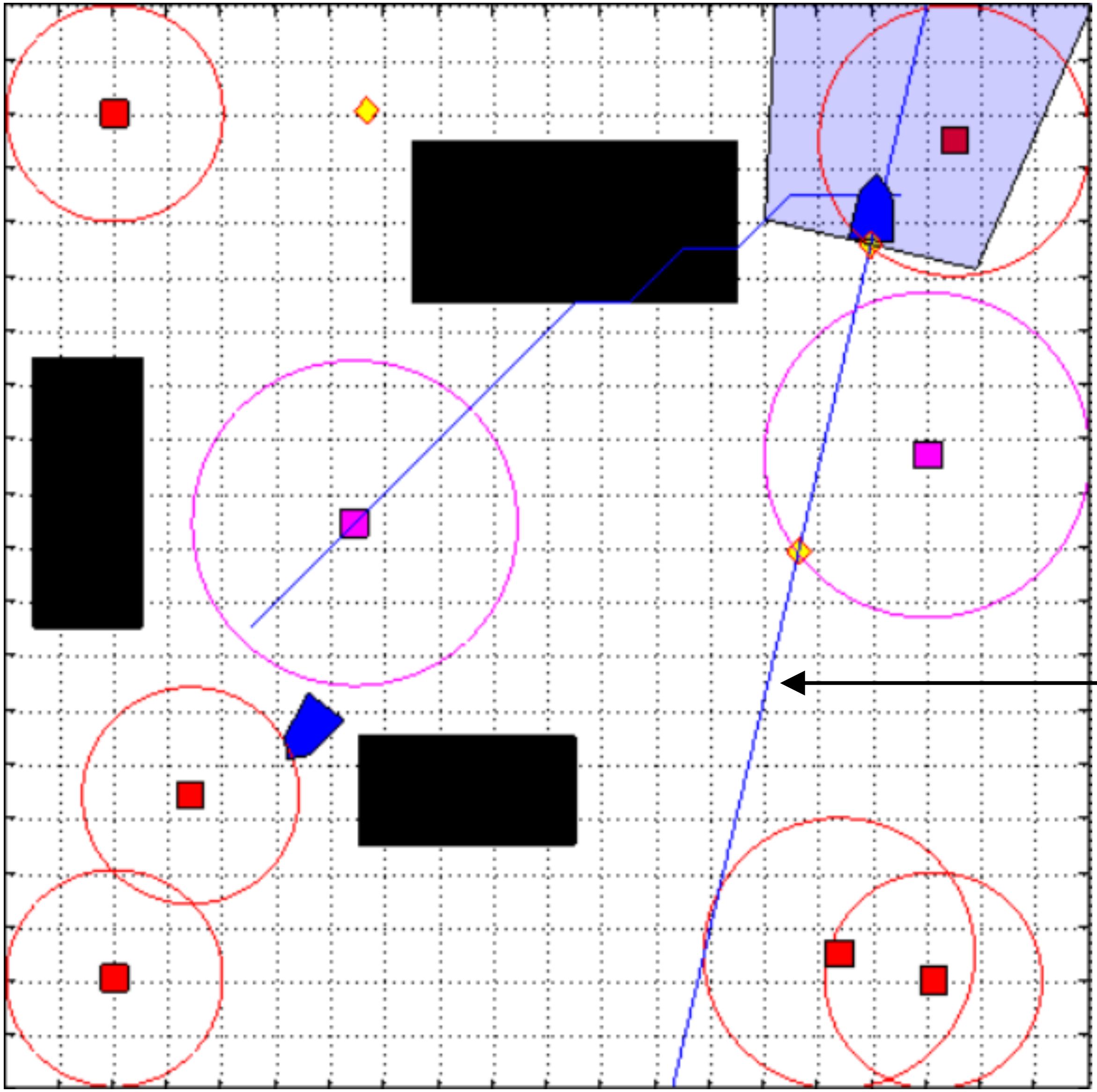
Heterogeneous Network



Heterogeneous Network

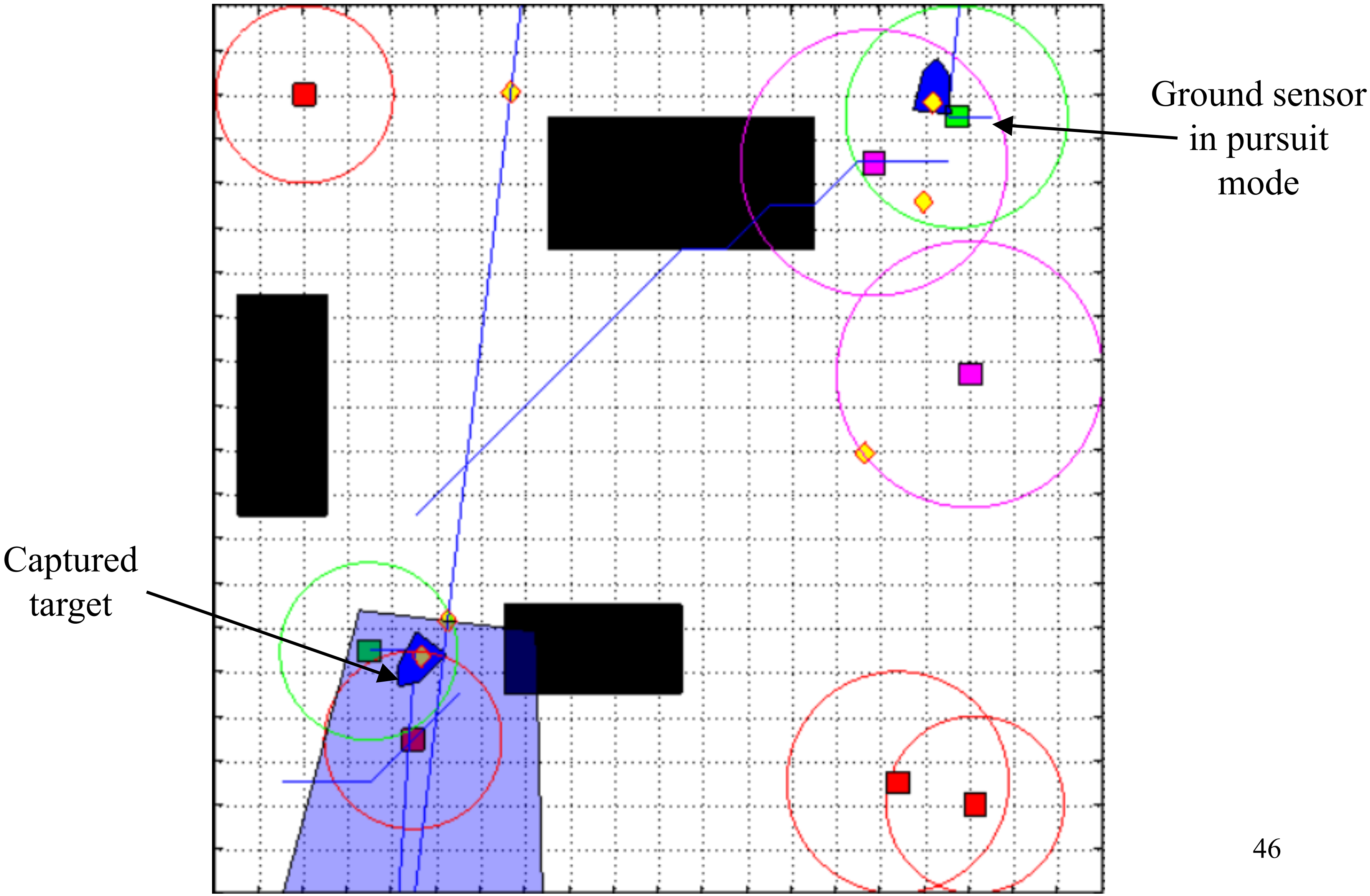


Heterogeneous Network

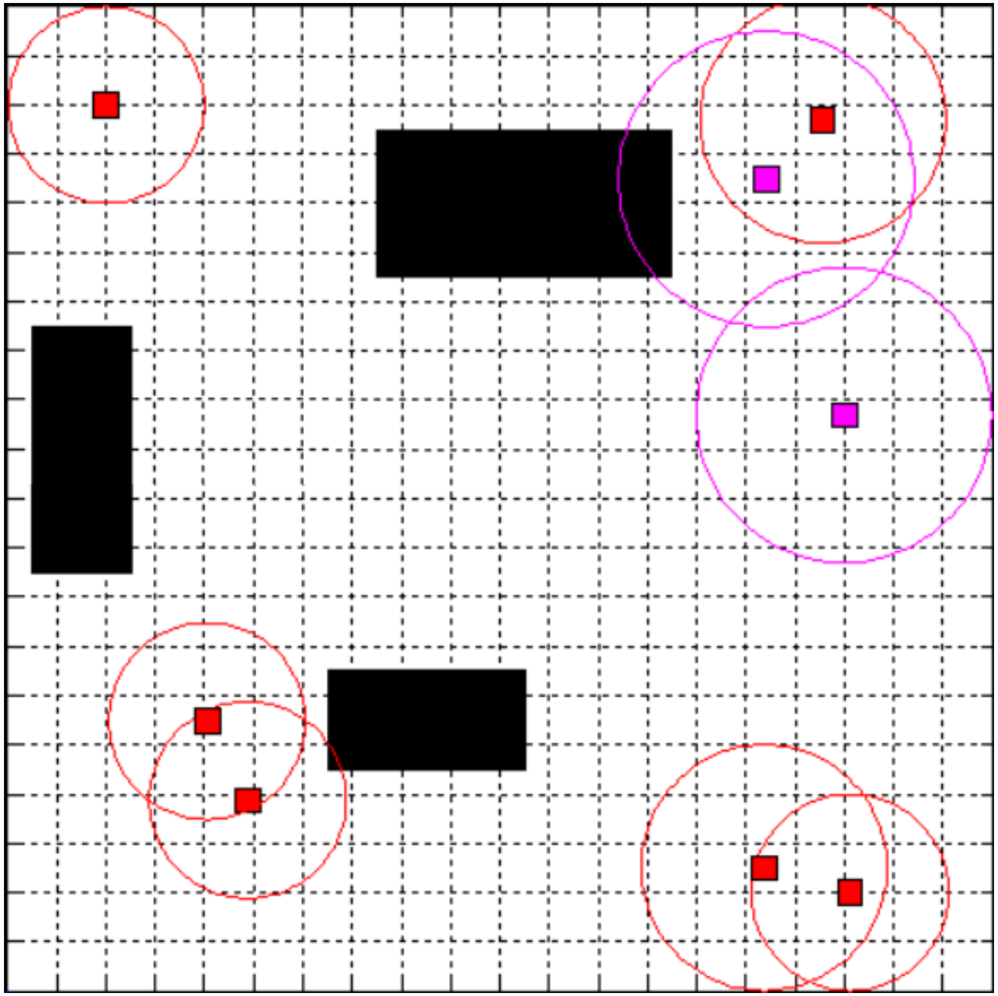


Partially-observed track

Heterogeneous Network



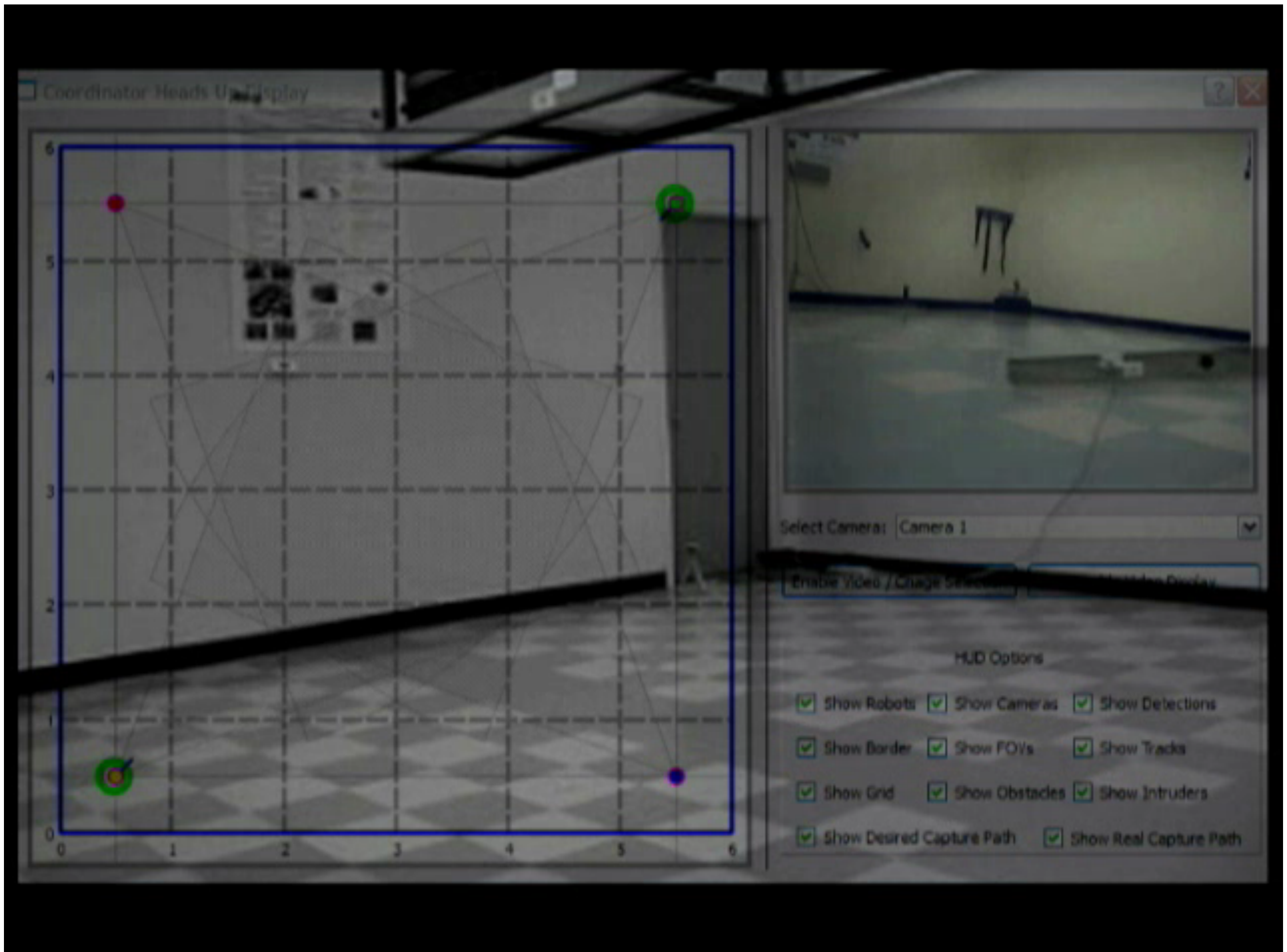
Game ends!



D. Tolic, R. Fierro, and S. Ferrari, "Cooperative multi-target tracking via hybrid modeling and geometric optimization," Proc. Mediterranean Conference on Control and Automation (MED'09), Thessaloniki, Greece, January 2009, pp. 440-445.

Experiments

- Conducted by Prof. Rafael Fierro and Brent Perteet, University of New Mexico



Conclusions

- Geometric and probabilistic sensor models
- Track Coverage Functions
- Information Value Functions
- Optimal Control of Cooperative Sensor Networks
- Underwater, ground, and air robots

Work in progress:

- Maneuvering targets
- Path Exposure
- Online Learning and Fusion
- Optimal Control of Distributions

L
I
S
C

LISC Research

Active Research Areas

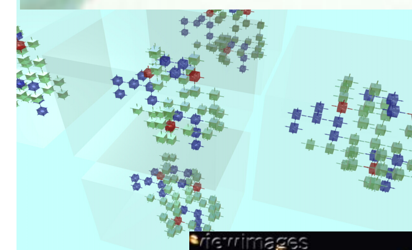
Approximate dynamic programming

Adaptive control of aircraft

Learning

Artificial and spiking neural networks

Games (CLUE®, Marco Polo, Pacman®)



Acknowledgments

Collaborators:

Dr. Thomas Wettergren, Naval Undersea Warfare Center

Prof. Rafael Fierro, University of New Mexico

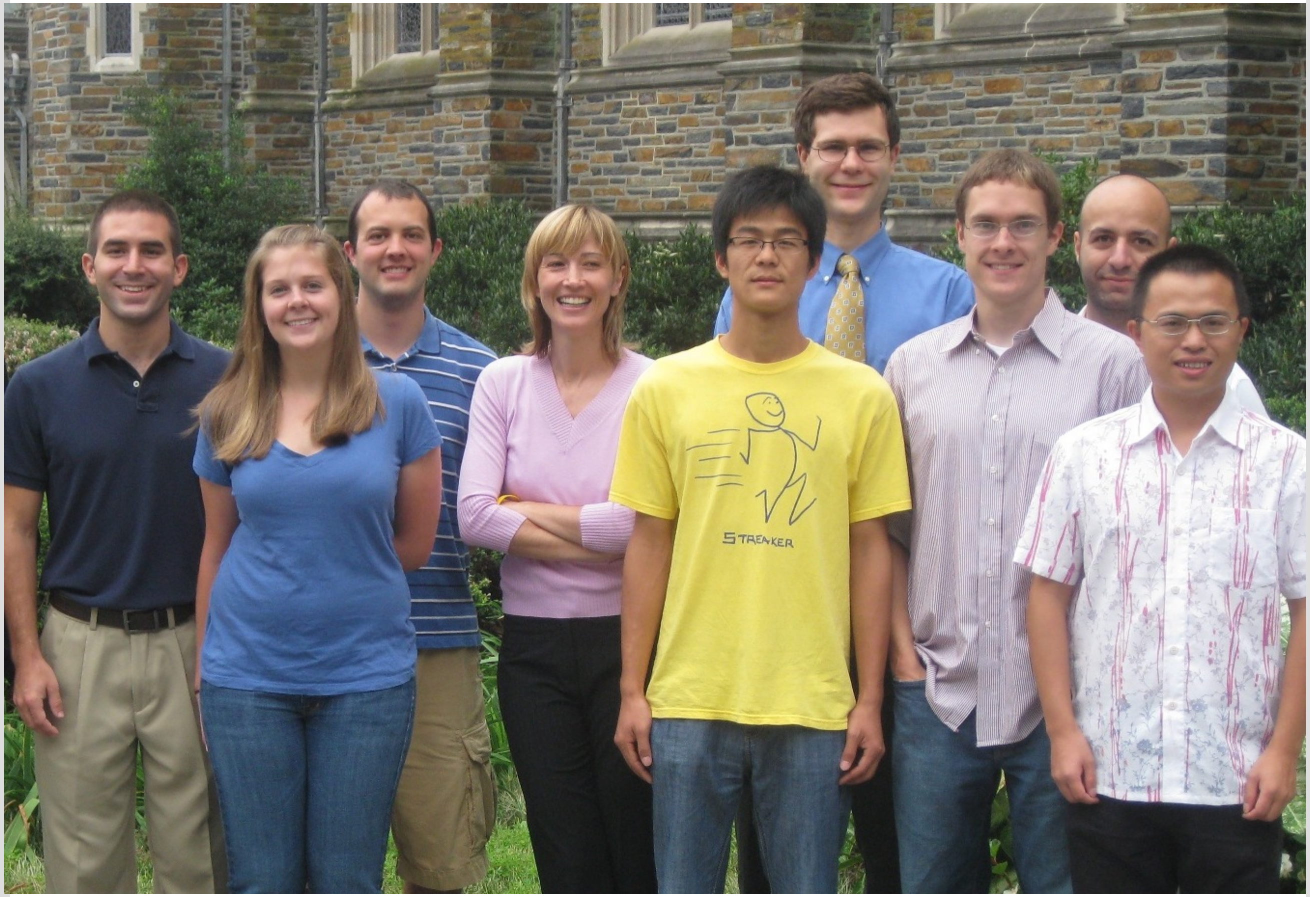
Prof. Anil Rao, University of Florida

Sponsors:

This research is funded by the ONR (Code 321), NASA, and the NSF ECS Division

L
I
S
C

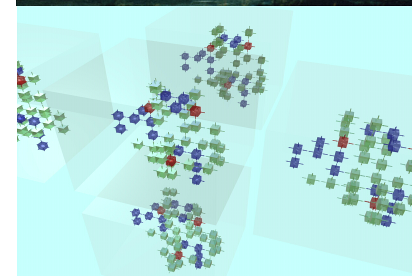
LISC Members



Dr. Chenghui Cai	Ashleigh Swingler
Dr. Kelli Baumgartner	Greg Foderaro
Gianluca Di Muro	Greyson Daugherty
Guoxian Zhang	Brian Bernard
Andrew Tremblay	Wenjie Lu

L
I
S
C

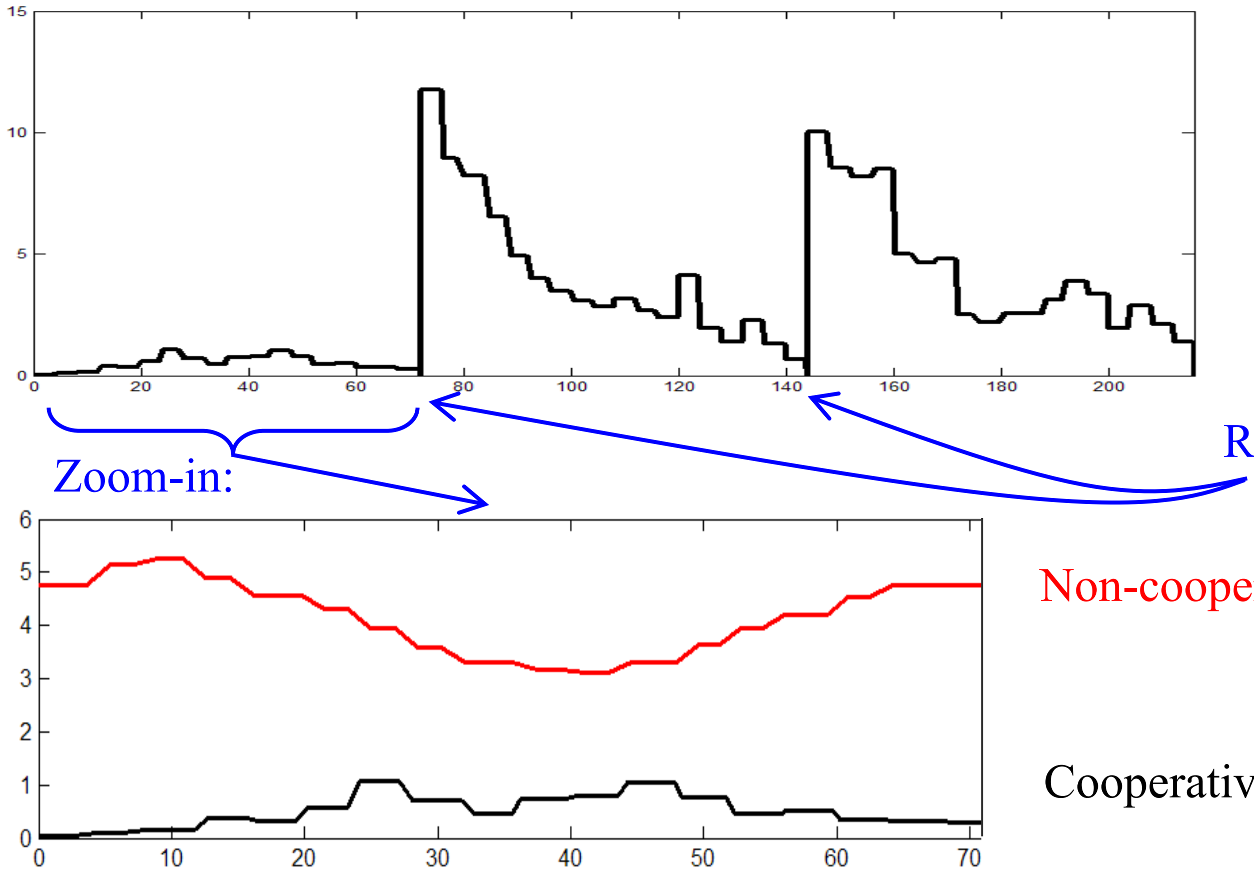
Back-up Slides



Energy Expenditure

Sensor network

Cooperative control



Re-planning

i^{th} sensor

Non-cooperative control

Cooperative control

Legend:

▬ (black)	: Cooperative optimal control ($\mathbf{x}_0^*$)
▬ (red)	: Non-cooperative path planning, given $\mathbf{x}_0$ and $\mathbf{x}_f$

Comparison with Other Deployments

Mission Parameters	*Performance Metrics*	*Optimal Control & x_0^**	*Optimal Control*	*Path Planning*	*Optimal Buoys*
$(n, k) = (15, 3)$	Track Coverage	$2.52 \cdot 10^4$	$1.74 \cdot 10^4$	$8.87 \cdot 10^3$	$1.99 \cdot 10^3$
$\Delta T = 3$ days	Energy	165	$1.31 \cdot 10^2$	604	0
	Total Performance	$2.50 \cdot 10^4$	$1.73 \cdot 10^4$	$8.27 \cdot 10^3$	$1.99 \cdot 10^3$
	OC* Improvement	n/a	44%	202.3%	1,156 %

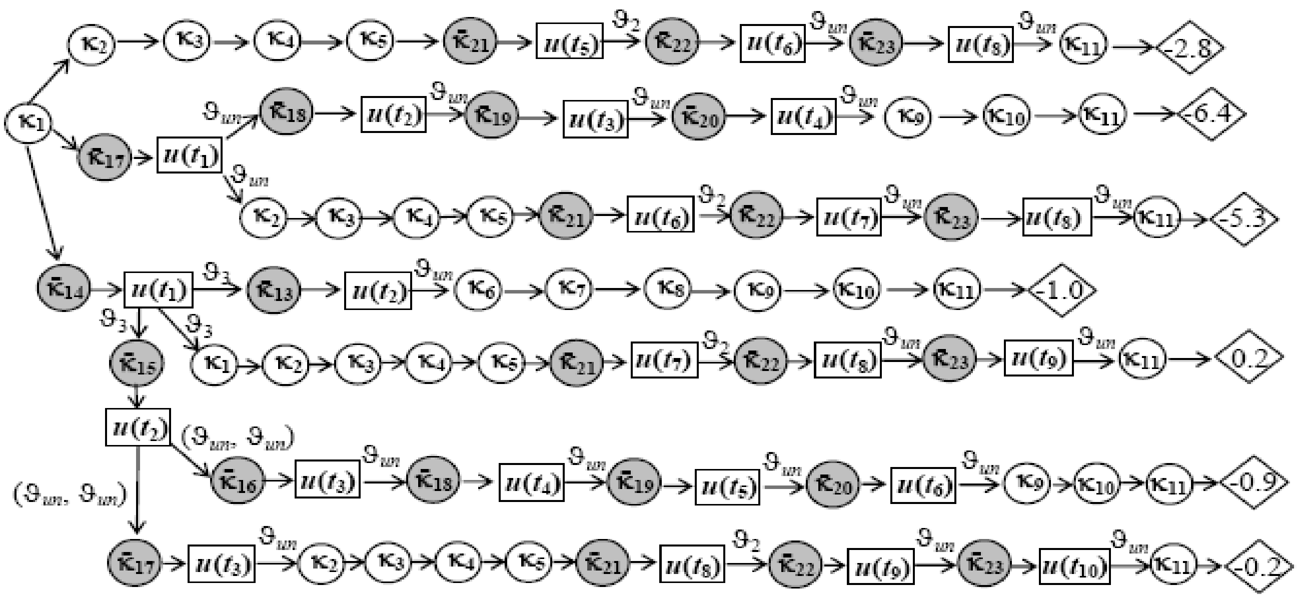
Where:

- Optimal Control & $\mathbf{x}_0^*$ (OC*): Sensor network's initial positions, control and state histories are optimized simultaneously.
- Optimal Control: Sensor network's initial positions are given, and the control and state histories are optimized simultaneously.
- Path Planning: Sensor network's initial and final position are optimized with respect to SS, and the control and state histories are optimized with respect to Energy.
- Zero Control: Sensor network's initial positions are optimized with respect to the currents and SS, but the sensors have no on-board controls (e.g., buoys).

Decision Tree and Optimal Sensor Path

The decision tree *DT* obtained from T_r is a tuple $\{U_C, U_D, R, A\}$ with κ_0 as the root, and the value of reward function *R* as the leaves. Where,

U_C: set of chance nodes (round); U_D: set of test-decision nodes (squares); *A*: directed arcs.



optimal

The **optimal path** in the decision tree is found using a rolling-back procedure that determines the optimal strategy by recursively estimating the utility of each branch.

S. Ferrari and C. Cai, "Information-Driven Search Strategies in the Board Game of CLUE®," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, Vol. 39, No 3, June 2009.

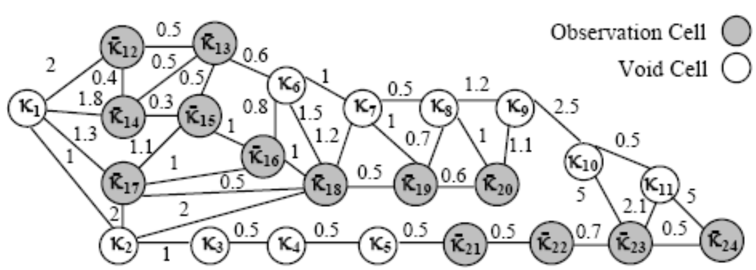
Pruned Connectivity Tree

The connectivity tree T_r associated with $\mathcal{G}$ and two cells $\kappa_0 \ni q_0$, $\kappa_f \ni q_f$, is a tree graph with κ_0 as the root, κ_f as the leaves, a cost d attached to each arc, and with the following properties:

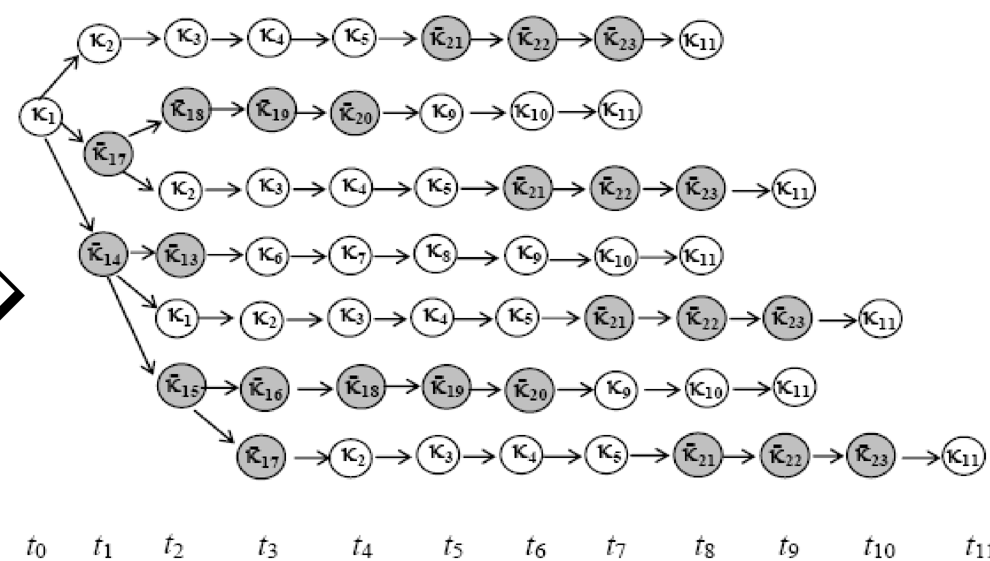
- A branch τ in T_r represents a channel joining κ_0 to κ_f in $\mathcal{G}$.
- Two branches are said to be *information equivalent* if they join the same cells, κ_i and κ_l, and contain the same set of observation cells, regardless of the order.
- A branch in T_r connecting any two cells κ_i and κ_l has the <u>smallest overall</u> cost of any other information-equivalent branch in $\mathcal{G}$.

Label-correcting pruning algorithm:

Connectivity graph, $\mathcal{G}$



Connectivity tree, T_r



Performance Analysis

Performance of pruning algorithm:

Method	Nodes | Arcs in $\mathcal{G}$	Observation cells in $\mathcal{G}$	Branches in T_r	Computation time
Pruning Algorithm	531 | 686	270	48	79 s
Exhaustive Search	51 | 106	43	51	87 s

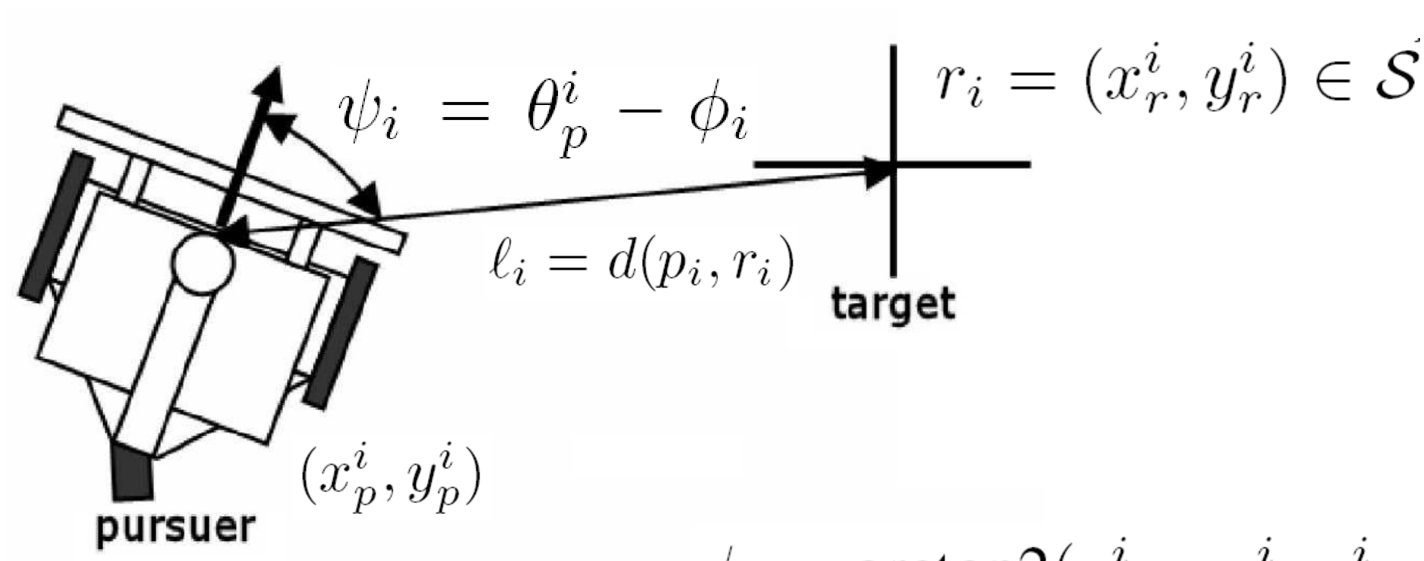
Method	d_M	Branches in T_r	Time slices in T_r	Computation time
Pruning Algorithm	None	16	17	7 s
	10	8	15	3 s
	5	4	9	1 s
Exhaustive Search	None	34835	19	1570 s
	10	27452	19	1064 s
	5	13693	17	378 s

Complexity of decomposition: $O(n_{\mathcal{T}}(n_{\mathcal{B}} + n_{\mathcal{T}})) + O((n_{\mathcal{B}} + n_{\mathcal{T}}) \log (n_{\mathcal{B}} + n_{\mathcal{T}}))$

$n_{\mathcal{A}}$, # of edges defining robot $\mathcal{A}$; $n_{\mathcal{B}}$, # of total edges of n convex obstacles; $n_{\mathcal{T}}$, # of total edges of r convex targets

C. Cai and S. Ferrari, "Information-Driven Sensor Path Planning by Approximate Cell Decomposition," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, Vol. 39, No. 3, June 2009.

Pursuer Control Policy



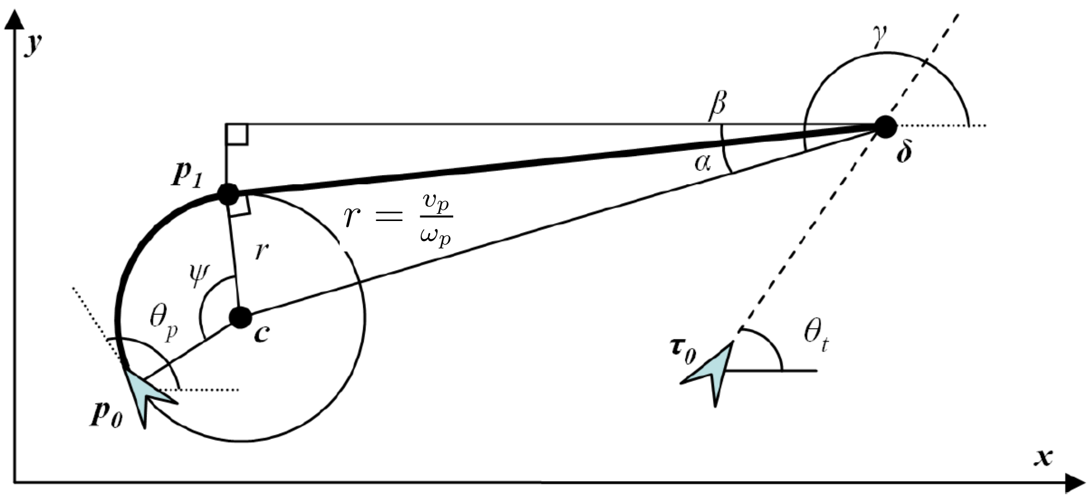
$$\phi_i = \text{arctan2}(y_r^i - y_p^i, x_r^i - x_p^i)$$

Proportional Controller: $u_p^i = K_p \zeta_i$

where $\zeta_i = [\ell_i \quad \psi_i]^T$ is the error vector and

$K_p = \text{diag}(k_v, k_\omega)$ is the diagonal control constant matrix

Policy in Pursuit Mode



$$\delta(t_c) = \begin{bmatrix} x_t + t_c v_t \cos\theta_t \\ y_t + t_c v_t \sin\theta_t \end{bmatrix}$$

$$t_c = \frac{r\psi + \|c - \delta(t_c)\| \cos\alpha}{v_p}$$

$$c_R = \begin{bmatrix} p_{0x} + r\cos(\theta_p - \frac{\pi}{2}) \\ p_{0y} + r\sin(\theta_p - \frac{\pi}{2}) \end{bmatrix} \quad c_L = \begin{bmatrix} p_{0x} + r\cos(\theta_p + \frac{\pi}{2}) \\ p_{0y} + r\sin(\theta_p + \frac{\pi}{2}) \end{bmatrix} \quad c = \begin{cases} c_R, & \text{if } \|c_R - \delta\| \le \|c_L - \delta\|, \\ c_L, & \text{if } \|c_R - \delta\| > \|c_L - \delta\|. \end{cases}$$

Interception point is computed using Newton's method

S. Ferrari, R. Fierro, B. Perteet, C. Cai, and K. C. Baumgartner, "A Geometric Optimization Approach to Detecting and Intercepting Dynamic Targets Using a Mobile Sensor Network," *SIAM Journal on Control and Optimization*, Vol. 48, No. 1, pp. 292-320, 2009.

Definitions

Target Tracks

1. An **unobserved track** is the path of a target *j* for which there are no detections at the present time, *t*

2. A **partially-observed track** is the path of a target that is estimated from $0 < l < k$ individual sensor detections obtained up to the present time, *t*

3. A **fully-observed track** is the path of a target that is estimated from at least *k* individual sensor detections obtained up to the present time, *t*

Cells

1. A **void cell** is a convex polygon $\kappa \subset \mathcal{C}_{free}$ with the property that for every configuration $q_i \in \kappa$ the pursuer *i* has zero probability of detecting a partially-observed target.

2. An **observation cell** is a convex polygon $\underline{\kappa} \subset \mathcal{C}_{free}$ with the property that for every configuration $q_i \in \kappa$ the pursuer *i* has a non-zero probability of detecting a partially-observed target.

Control Policy for Sensors in Detection Mode

Reward function:

$$R(\kappa_l, \kappa_\imath) = w_1 P_d(\kappa_l) + w_2 \Delta \mathcal{P}_{\mathcal{S}}^k(\kappa_l, \kappa_\imath) - w_3 d(\kappa_l, \kappa_\imath)$$

where $\Delta \mathcal{P}_{\mathcal{S}}^k(\kappa_l, \kappa_\imath)$ is the change in the network track-coverage,

$d(\kappa_l, \kappa_\imath)$ is the Euclidean distance between cells,

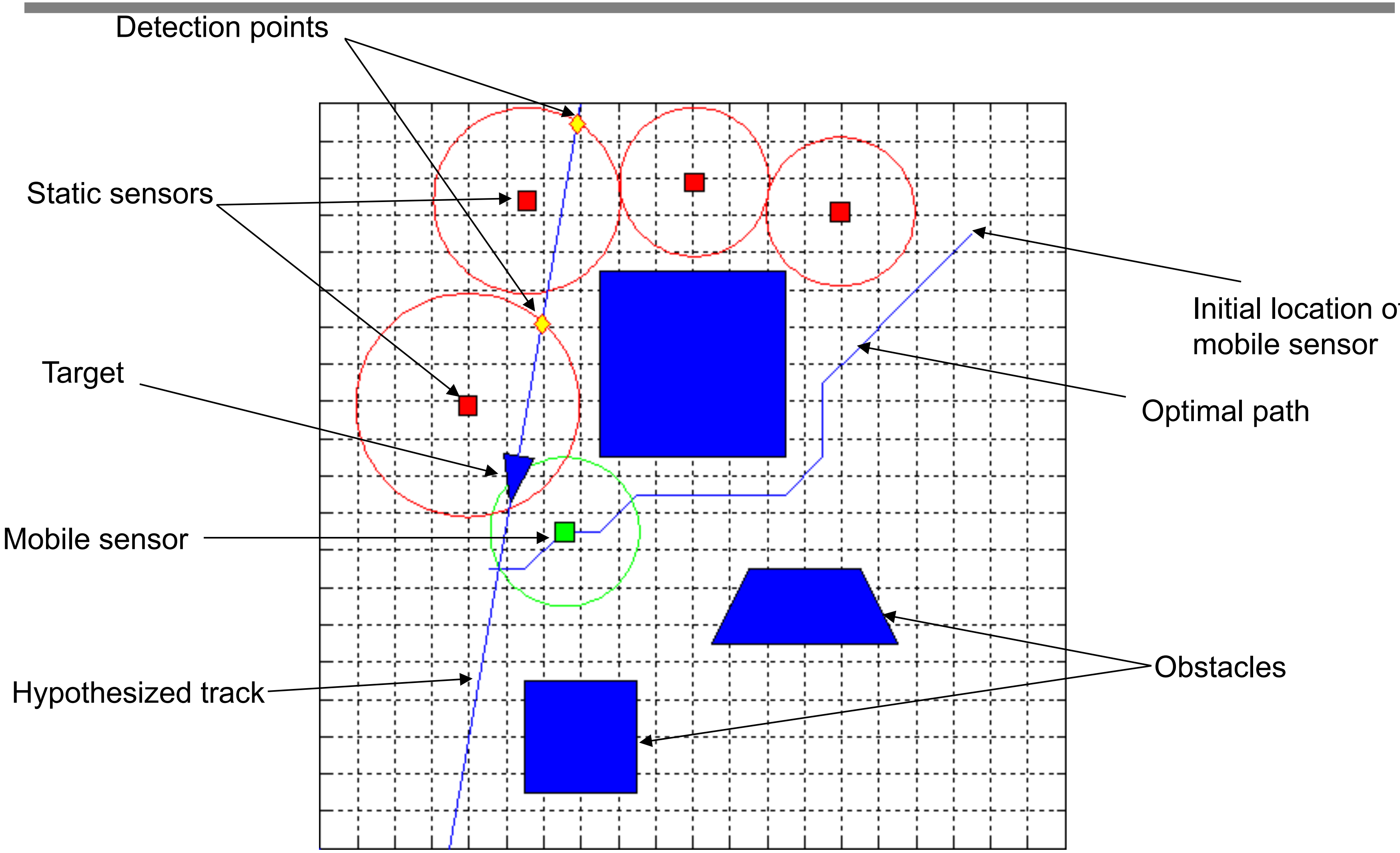
$P_d(\kappa_l)$ is the probability of detecting a target inside a given cell (assuming a binary sensor model), and

w_1, w_2, and w_3 are weighting parameters.

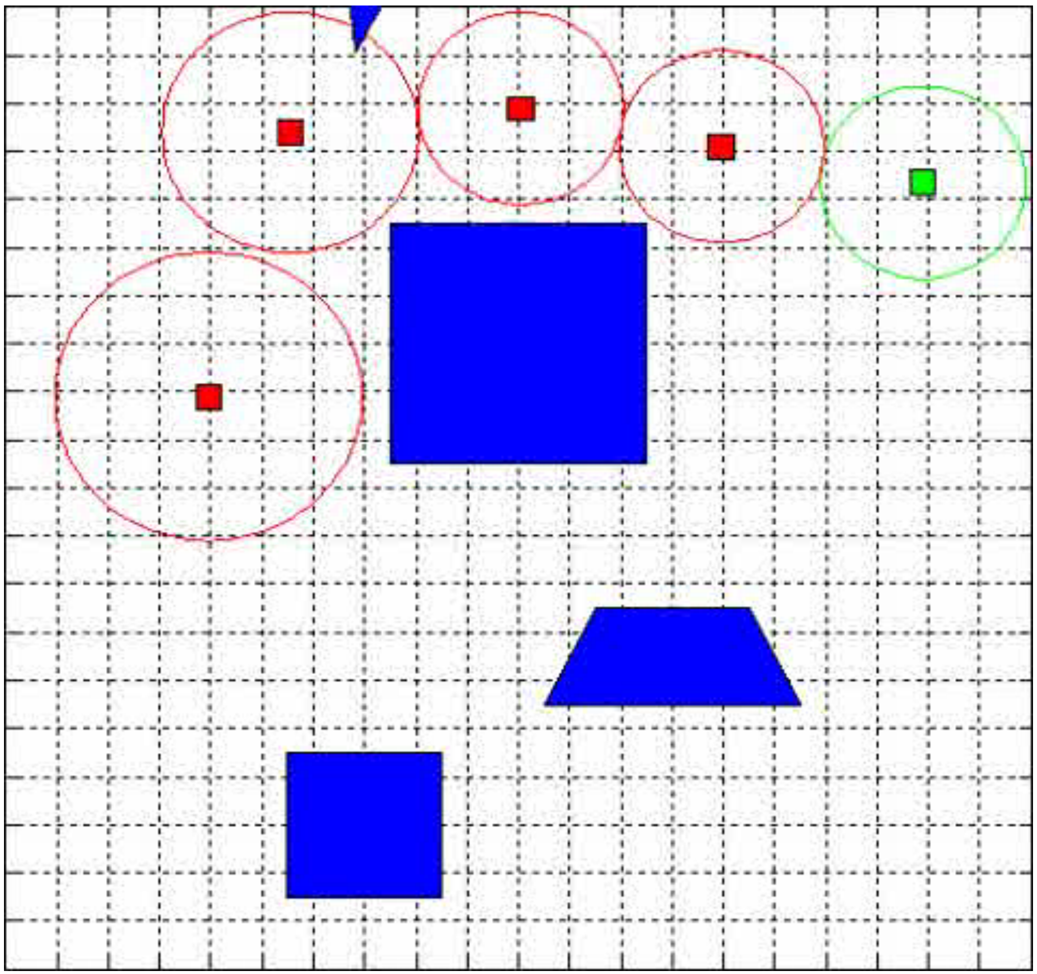
Using a graph searching algorithm such as A*, the optimal sequence or channel of cells which maximizes the reward is

$$\mu^* \equiv \{\kappa_0, \ldots, \kappa_f\}^* = \arg\max_{\mu} \sum_{(\kappa_l, \kappa_\imath) \in \mu} R(\kappa_l, \kappa_\imath)$$

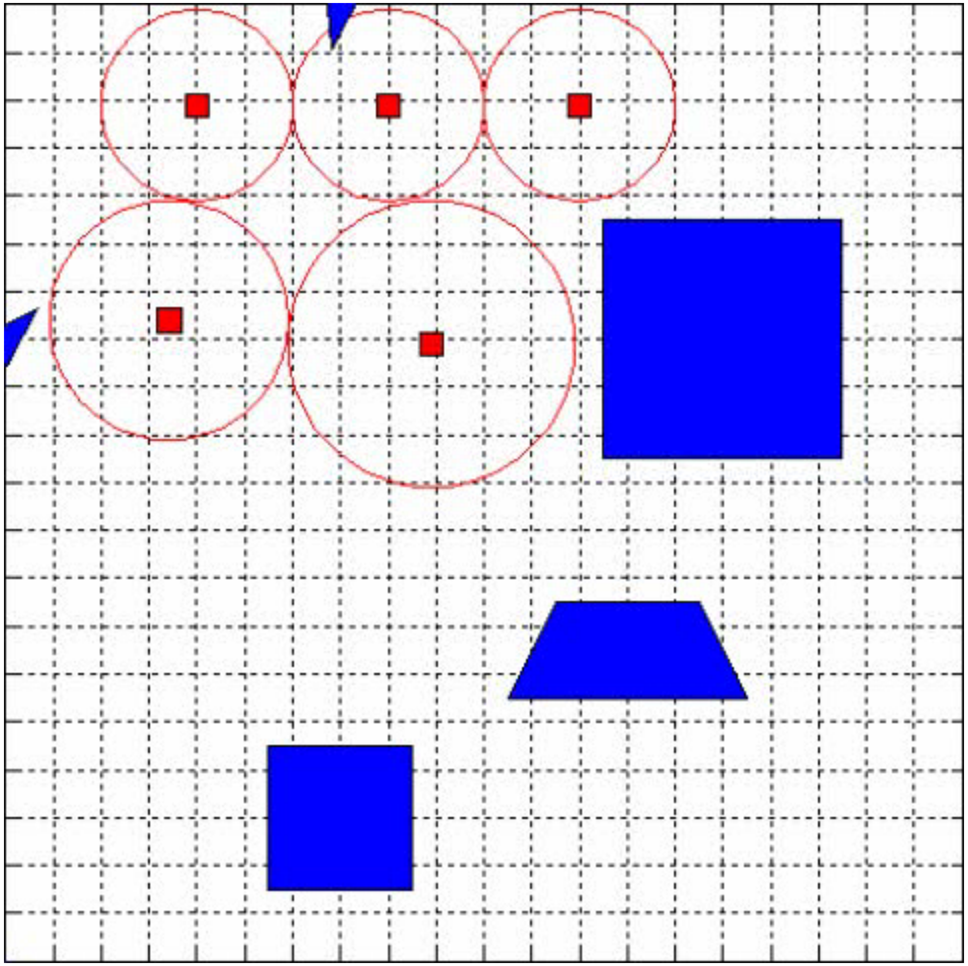
Results: 5 Sensors and 1 Target



Results: 5 Sensors and 1 Target

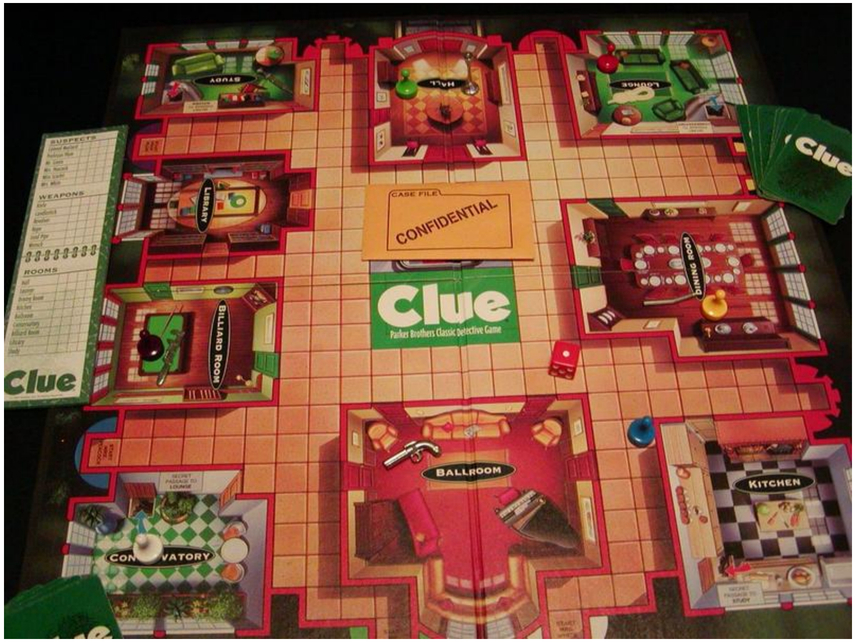


Results: 5 Sensors and 2 Targets



Benchmark Example -- Board Game of CLUE®

CLUE® is a benchmark example of treasure-hunt problem, because the information (or evidence) that can be obtained about the hidden variable, depends on the position of the pawn on the gameboard: coupled motion planning and inference problems.



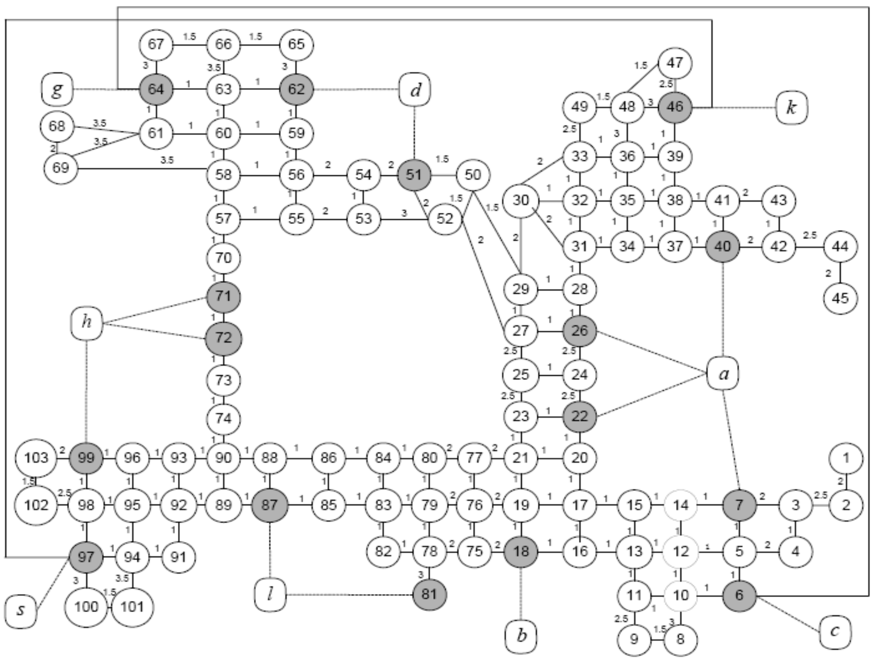
CLUE® game board. CLUE® & ©2006 Hasbro, Inc. Used with permission.

Game characteristics:

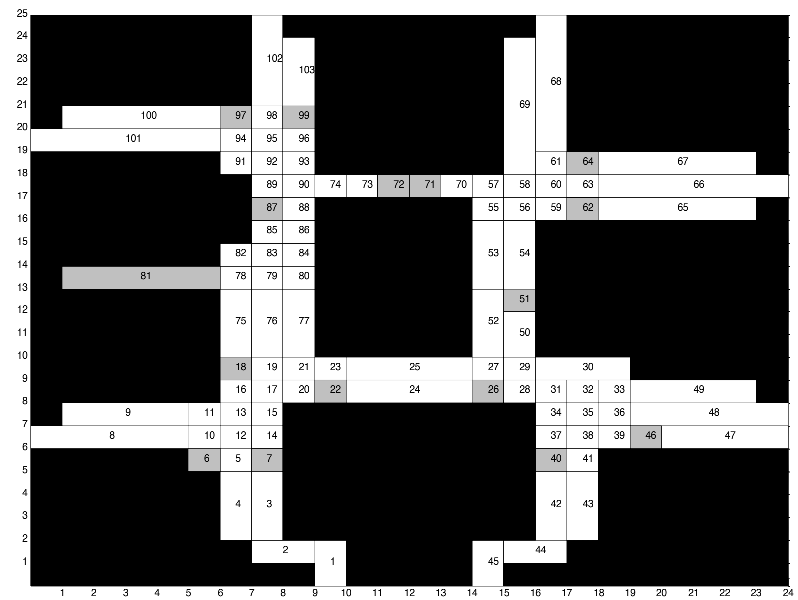
- "who, how, and what room?"
- 6 suspects, 6 weapons, 9 rooms
- Movement
- Suggestion decision → evidence
- Inference of hidden cards
- CLUE® ↔ surveillance systems

C. Cai and S. Ferrari, "A *Q*-Learning Approach to Developing an Automated Computer Player for the Board Game of CLUE®," *Proc. International Joint Conference on Neural Networks*, Hong Kong, 2008, pp. 2347-2353.

Exact Cell Decomposition



Connectivity Graph

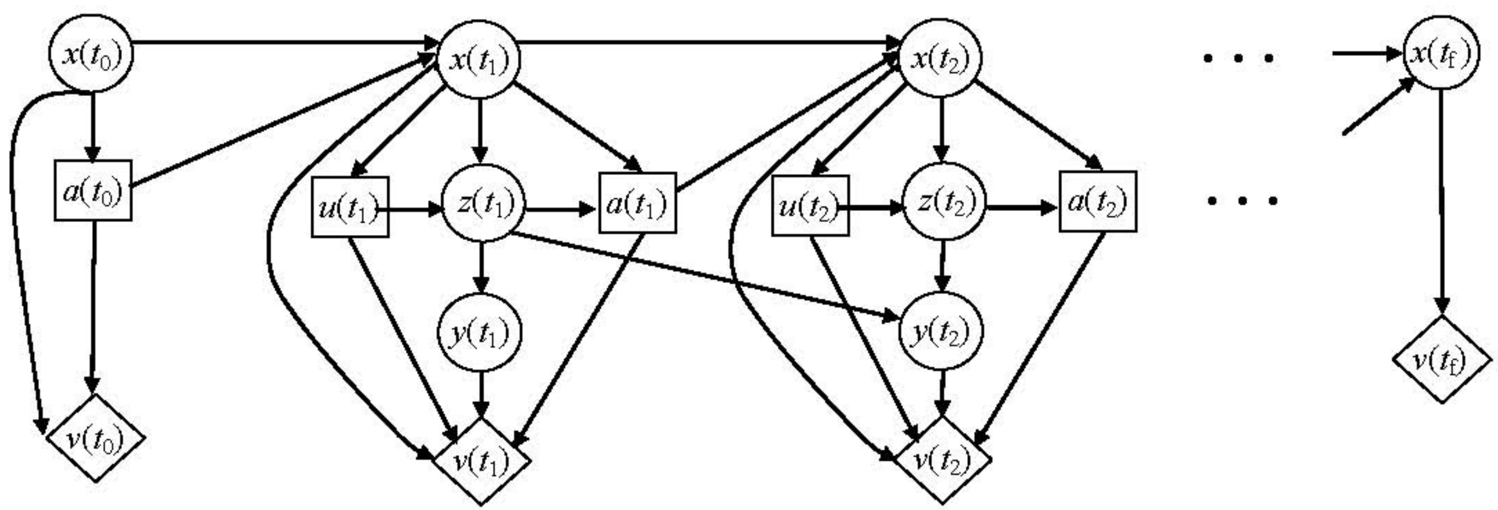


Convex Polygonal Decomposition

S. Ferrari and C. Cai, "Information-Driven Search Strategies in the Board Game of CLUE®," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, to appear in Vol. 39, No 3, June 2009.

Treasure Hunt Influence Diagram

- Connectivity tree, T, is folded into an influence diagram (action decisions, a_k, observable state, x_k)
- The observation cells in $\Omega(x_k)$ specify the admissible set of test decisions, u_k, and the domain of the non-observable state, $\Omega(z_k) = \{m_i, \ldots, m_j\}$
- $Z_T = \{z_1, z_2, \ldots, z_{f-1}\}$ a sequence of measurements about y over $\{t_1, t_2, \ldots, t_f\}$
- **Profit of Observation:** $v(t_k) = R(t_k) = w_B.B(t_k) - w_J.J(t_k) - w_D.D(t_k)$
  where $B(t_k)$ is the expected entropy reduction (EER),
  $\Delta H(t_k) = H(\mathrm{y} \mid z_{k-1}, z_{k-2}, \ldots, z_1) - H(\mathrm{y} \mid z_k, z_{k-1}, \ldots, z_1) = I(\mathrm{y}; z_k \mid z_{k-1}, \ldots, z_1)$



Influence Diagram Representation of Underlying POMDP

Optimal CLUE® Game Strategy

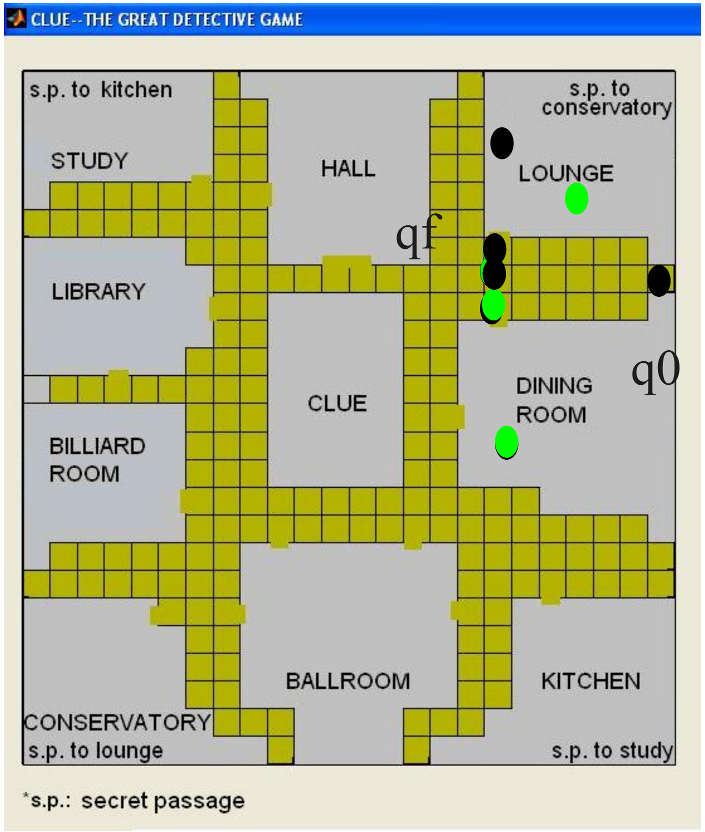
Profit of Observation:

$R(t_k) = w_B.B(t_k) - w_J.J(t_k) - D(t_k)$

- ICP

q0=66

qf=64



Shortest path (w_B= 0; w_J=0):

[66 63 64]; R = -7;

Aggressive path (w_B= 12):

[66 63 62 d 62 63 64];

R =12×0.41- 0 - 7= -2.08;

Suggestion:

{Mrs. Peacock; Revolver; Dining Room}

{Mr. Green; Rope; Lounge}

P3's Response: {Revolver}

P2's Response: {Mr. Green}

Posterior before evidence:

Posterior after evidence:

[0.25 0.25 0.25 0.25] [illegible]

[0.33 0.33 0.33] [illegible]

[0.14 0.14 0.14 0.14 0.14 0.14 0.14] [illegible]

Game Results

Players:	Games won / games played	Winning rate	Time to determine y^r	Time to win the game
ID player	14 / 25	56 %	8.21 turns	8.57 turns
Humans	10 / 25	40 %	n.a.	12.7 turns
CS player	1 / 25	4 %	12 turns	12 turns
BN player	20 / 37	54 %	10.2 turns	10.9 turns
Humans	16 / 37	43 %	n.a.	12.9 turns
CS player	1 / 37	2.7 %	4 turns	4 turns

Players:	Games won / games played	Winning rate	Time to determine y^r	Time to win the game
ID player	21 / 43	48.8 %	10.5 turns	10.7 turns
Humans	19 / 43	44 %	n.a.	8.89 turns
CS player	3 / 43	6.98 %	8.33 turns	8.33 turns
BN player	18 / 55	32.7 %	11.7 turns	12.0 turns
Humans	32 / 55	58 %	n.a.	11.4 turns
CS player	5 / 55	9 %	10.6 turns	10.6 turns