LAIR
Lab for Autonomous and Intelligent Robotics

UNDERWATER ROBOTICS

Field Explorations in Marine Biology, Oceanography, and Archeology

COS 402: Artificial Intelligence - Sept. 2011

Christopher M. Clark

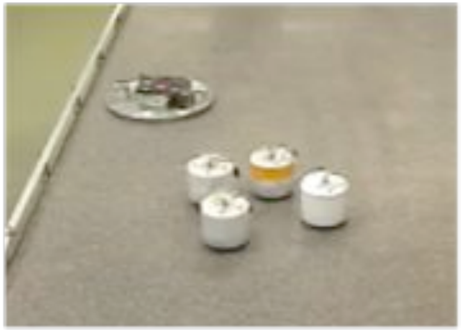
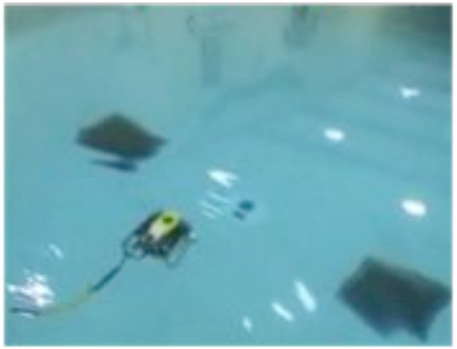
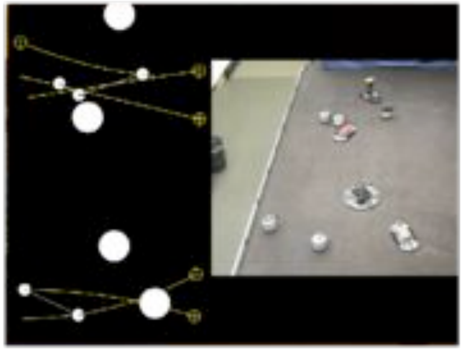
Outline

- Past Projects
- Maltese Cistern Mapping
- Distributed Multi Robot Boundary Tracking
- California Coastal Shark Tracking

LAIR
Lab for Autonomous and Intelligent Robotics

Past Projects

- Complete and Scalable multi-robot motion planning in tunnels
- Formation Planning with Non-Holonomic Constraints
- Autonomous Control of an ROV
- Altruistic Relationships between robots in multi-robot communities
- Arctic ice Deployments
- Autonomous Highway System – Decentralized lane level Control
- …

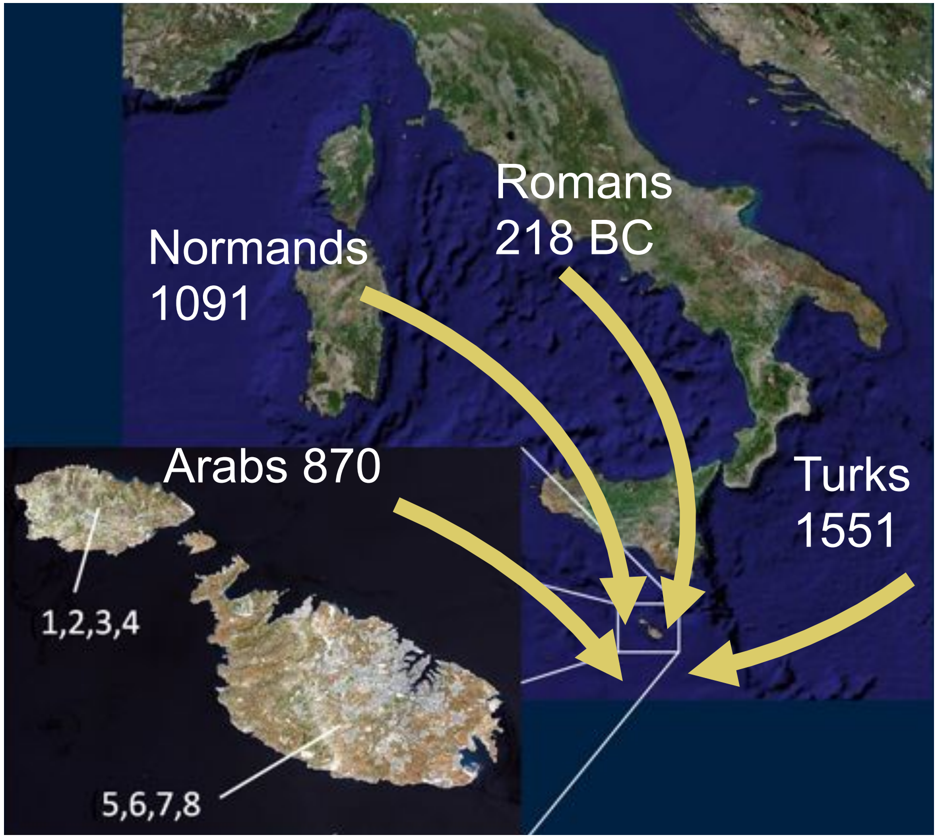





Outline

- Past Projects
- Maltese Cistern Mapping
- Distributed Multi Robot Boundary Tracking
- California Coastal Shark Tracking

LAIR
Lab for Autonomous and Intelligent Robotics

Maltese Cistern Mapping



Maltese Cistern Mapping

Maltese Cistern Mapping

Maltese Cistern Mapping




Maltese Cistern Mapping

Maltese Cistern Mapping

(a)

(b)

(c)

(d)

Maltese Cistern Mapping

- Localization
  - Determine the state of the robot in a *known environment* (e.g. map)
- Mapping
  - Build a map of the environment using measurements with respect to a *known robot state* (e.g. with GPS)
- SLAM – Simultaneous Localization And Mapping
  - Build a map of the environment using measurements with respect to a robot's state that is determined with respect to the map

[Thrun & Burgard 2005]

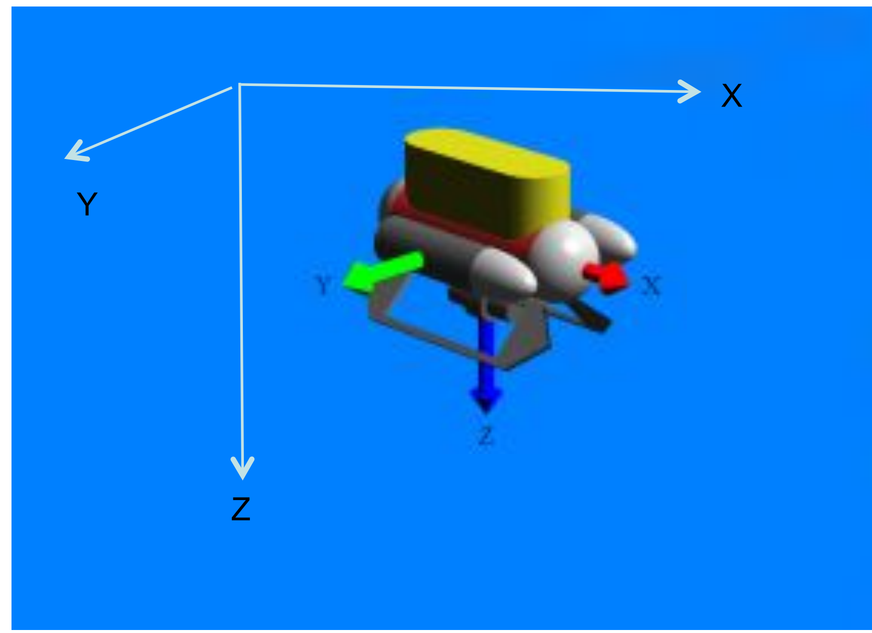
Malta Cistern Mapping

- Related Work:
  - Fairfield et. Al., *Real-time slam with octree evidence grids for exploration in underwater tunnels*, Journal of Field Robotics, 2006.
  - Ribas et. Al., *Underwater slam in man-made structured environments,* Journal of Field Robotics, 2008.
  - Mallios et. Al., *Pose-Based SLAM with Probabilistic Scan Matching Algorithm using a Mechanical Scanned Imaging Sonar*, IEEE OCEANS, 2009.

Maltese Cistern Mapping

- State Vector:

$$\boldsymbol{x} = [\underbrace{x\ y\ z\ \theta}_{\substack{\text{Position States}\\ \text{(Global Frame)}}}\ \underbrace{u\ v\ r\ \Theta}_{\substack{\text{Velocity States}\\ \text{(Robot Frame)}}}]$$



- Dynamic Model [Wang 2006]
  - Horiz. & Vert. Motion Decoupled
  - Quadratic Drag terms
  - No tether effects modeled

$$\underbrace{\boldsymbol{x}_{t+1}}_{\substack{\text{Current}\\ \text{State}}} = f(\underbrace{\boldsymbol{x}_t}_{\substack{\text{Previous}\\ \text{State}}},\ \underbrace{\boldsymbol{u}_{t+1}}_{\substack{\text{Control}\\ \text{Inputs}}})$$

Maltese Cistern Mapping

- Modeling Assumptions:
  - The ROV will usually move with low velocity when on mission
  - Almost three planes of symmetry;
  - Vehicle is assumed to be performing non-coupled motions.
  - Horizontal Plane:

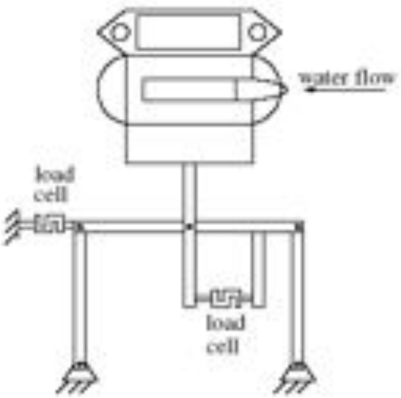
$$\begin{aligned} m_{11}\dot{u} &= -m_{22}vr + X_u u + X_{u|u|}u|u| + X \\ m_{22}\dot{v} &= m_{11}ur + Y_v v + Y_{v|v|}v|v|, \\ I\dot{r} &= N_r r + N_{r|r|}r|r| + N, \end{aligned}$$

  - Vertical Plan:

$$m_{33}\dot{w} = Z_w w + Z_{w|w|}w|w| + Z$$

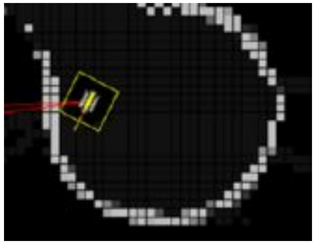
Malta Cistern Mapping

- Coefficients for the dynamic model are pre-calculated using strip theory;
- A series of tests are carried out to validate the hydrodynamic coefficients, including
  - Propeller mapping
  - Added mass coefficients
  - Damping coefficients



Maltese Cistern Mapping

- Occupancy Grid Mapping
  - Doesn't require knowledge of features!
  - The environment is discretized into a grid of equal sized cells.
  - Each cell (i, j) is assigned a likelihood $m_{ij} \in [0,1]$ of being occupied



- FastSLAM for occupancy grids- *[Thrun et al., 2005]*
  - Particle filter based SLAM
  - Occupancy grids representation

Maltese Cistern Mapping

- What is a Particle?
  - A particle is an individual state estimate.
  - In our SLAM, a particle i has three component

$$\{ \underbrace{\boldsymbol{x}^i}_{\text{State}} \ \underbrace{\boldsymbol{m}^i}_{\text{Map}} \ \underbrace{w^i}_{\text{Weight}} \}$$

1. The state is $\boldsymbol{x} = [x\ y\ z\ \theta\ u\ v\ r\ w]$
2. The map is an occupancy grid $\boldsymbol{m}$
3. The weight $\boldsymbol{w}$ that indicates it's likelihood of being the correct state.

Maltese Cistern Mapping

Alg. FastSLAM_occupancy_grids(X_{t-1}, u_t, z_t):

$X_t' = X_t = 0$

for $k = 1$ *to* M *do*

  x_t^k = **sample_motion_model**(u_t, x_{t-1}^k)

  w_t^k = **measurement_model_map**(z_t, u_t, m_{t-1}^k)

  m_t^k = **updated_occupancy_grid**(z_t, u_t, m_{t-1}^k)

  $X' = X' + \{ x_t^k, m_t^k, w_t^k \}$

endfor

Propagate Particles

for $k = 1$ *to* M *do*

  draw i *with probability* $\sim w_t^i$

  add $\{ x_t^i, m_t^i \}$ *to* X_t

Endfor

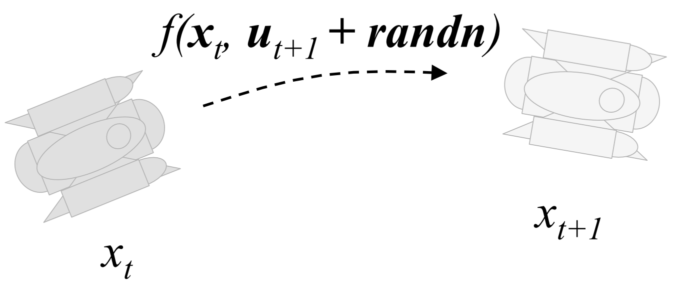
return X_t

Resampling

Maltese Cistern Mapping

- Step 4: **sample_motion_model**
  - The state vector is propagated forward in time to reflect the motion of the ROV based on control inputs and uncertainty
  - The dynamic model is used to propagate particle states,

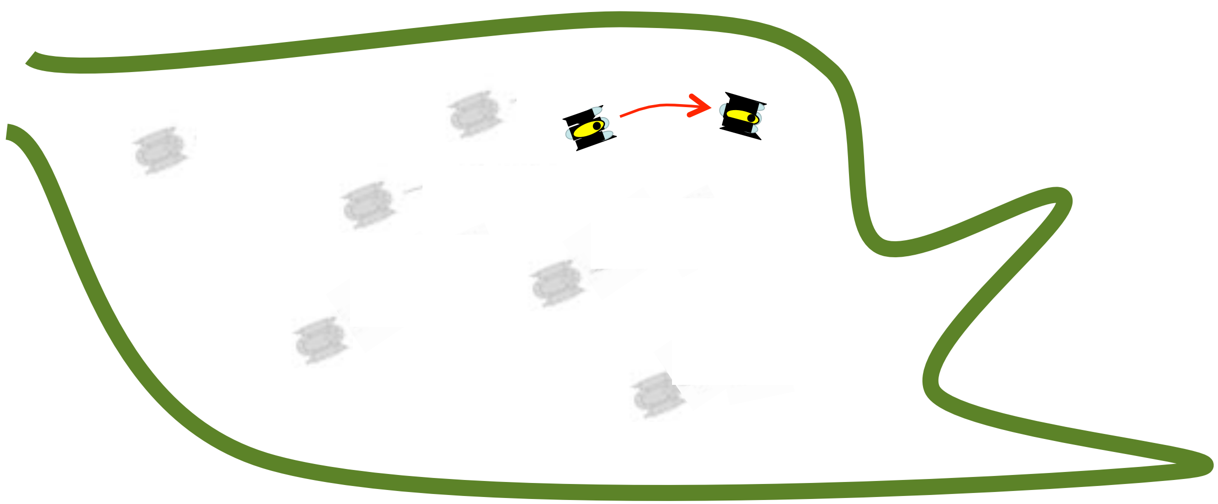
$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_{t+1} + \underbrace{\boldsymbol{randn(0,\sigma_u)}}_{\text{Experimentally Determined Process Noise}})$$



Maltese Cistern Mapping

- Step 4: **sample_motion_model**
  - The dynamic model is used to propagate particle states,

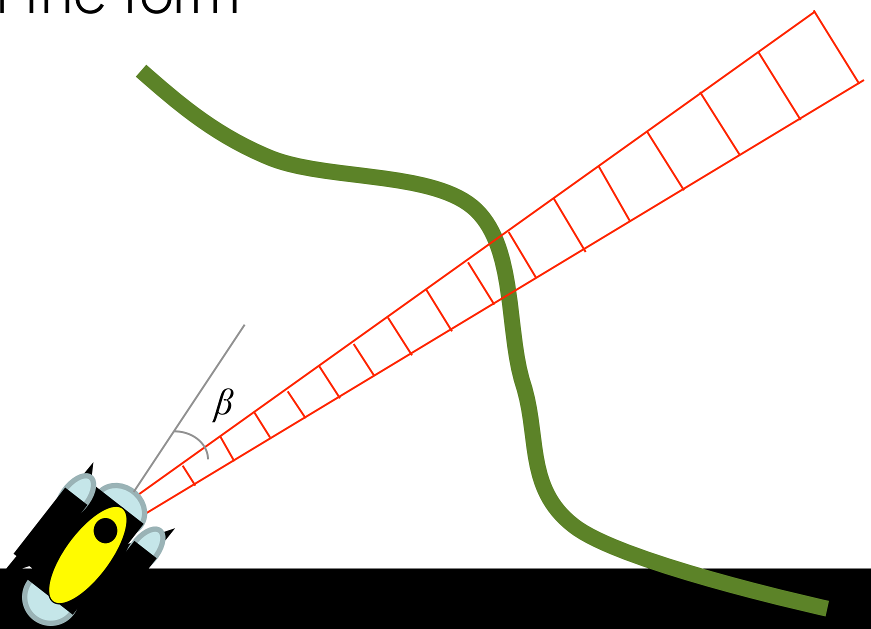
$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_{t+1} + \boldsymbol{randn})$$



Maltese Cistern Mapping

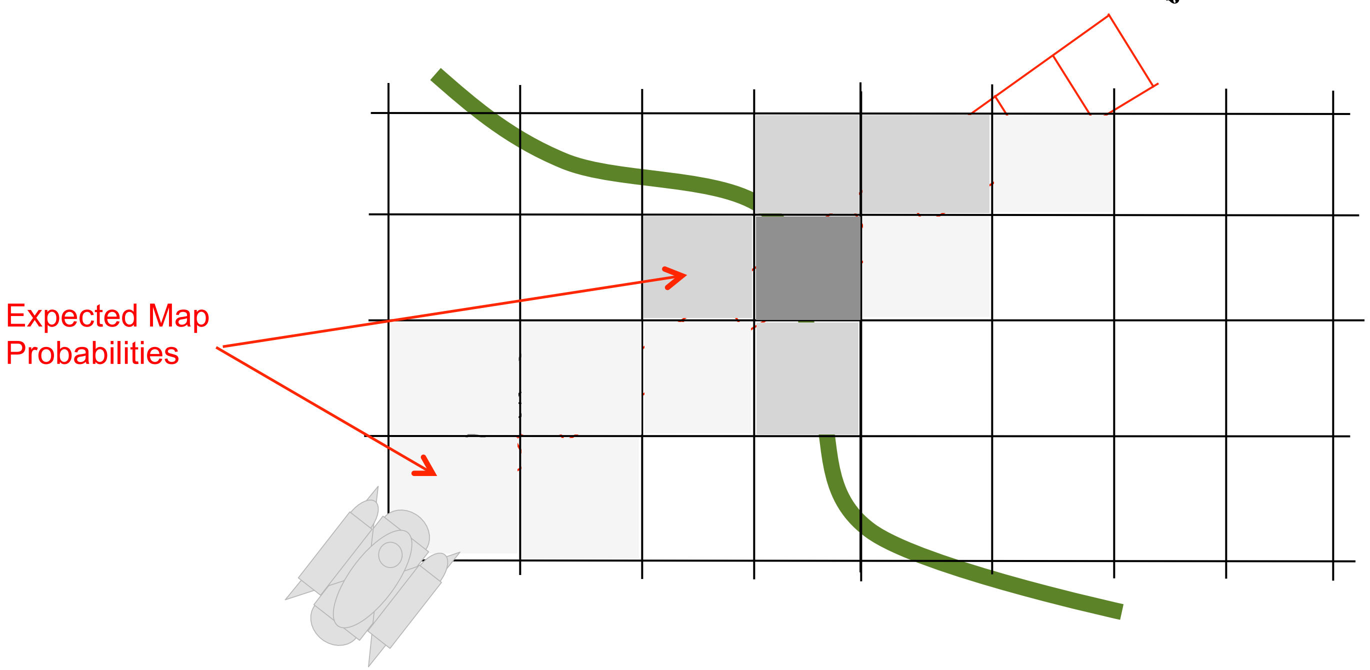
- Step 5: **measurement_model_map**
  - Particle weights are calculated by comparing actual sonar measurements with expected sonar measurements
  - Sonar measurements come in the form

$$z = [\,\underbrace{\beta}_{\text{sonar angle}}\ \underbrace{s^0\ s^1\ \ldots\ s^B}_{\text{Strength of returns for increasing range}}\,]$$



Maltese Cistern Mapping

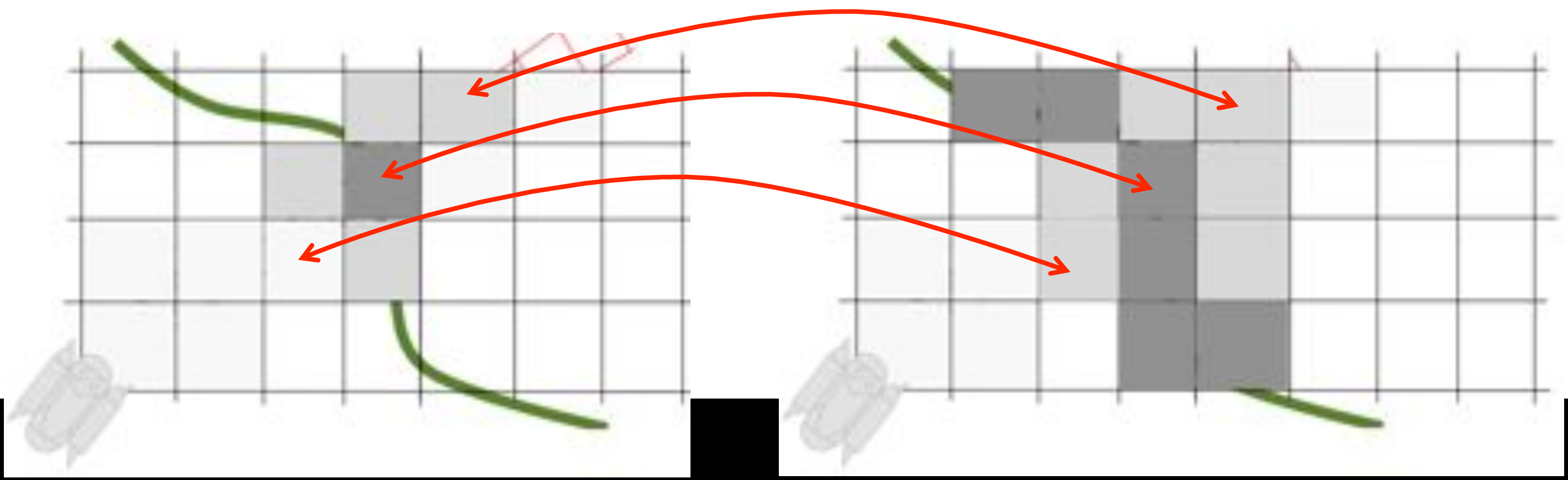
- Step 5: **measurement_model_map**
  - Given the state of the particle within a map, we can project which map cells the sonar would overlap, and calculate the *expected* map probabilities for those cells $\boldsymbol{p_z} = [\, p^0\ p^1\ \ldots\ p^{B'}]$



Maltese Cistern Mapping

- Step 5: **measurement_model_map**
  - Compare *expected* map probabilities with *existing* map probabilities.

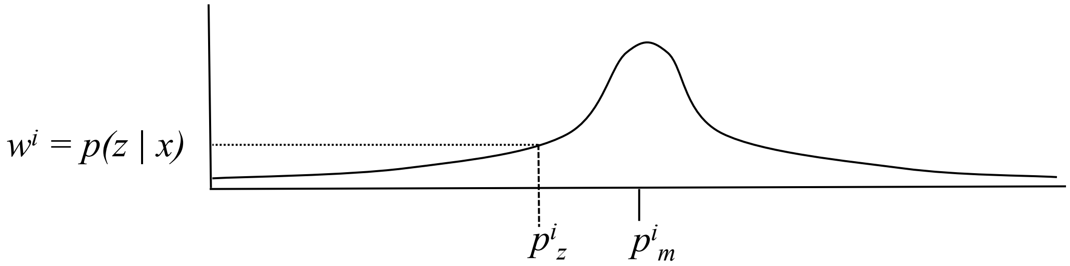
Expected Probabilities *Existing Map Probabilities*



Maltese Cistern Mapping

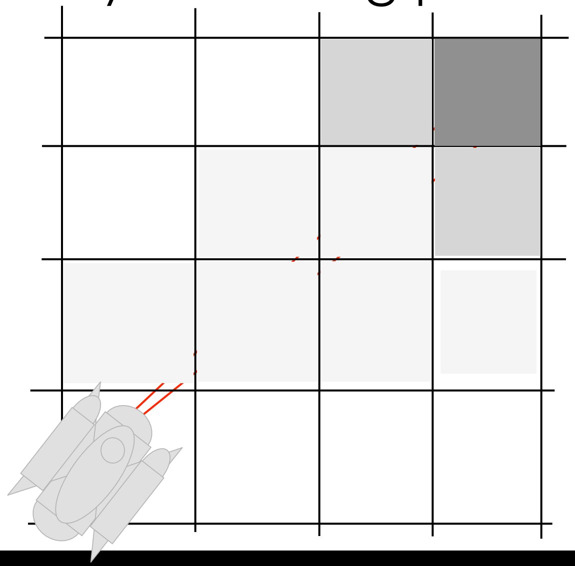
- Step 5: **measurement_model_map**
  - To calculate the particles weight w, we compare the expected map probabilities p_z with the map's current probabilities for the corresponding cells p_m

$$w = \sum_{i=1}^{B'} \frac{1}{\sigma\sqrt{2\pi}} exp \left(\frac{-(p_m^i - p_z^i)^2}{2\sigma^2} \right)$$



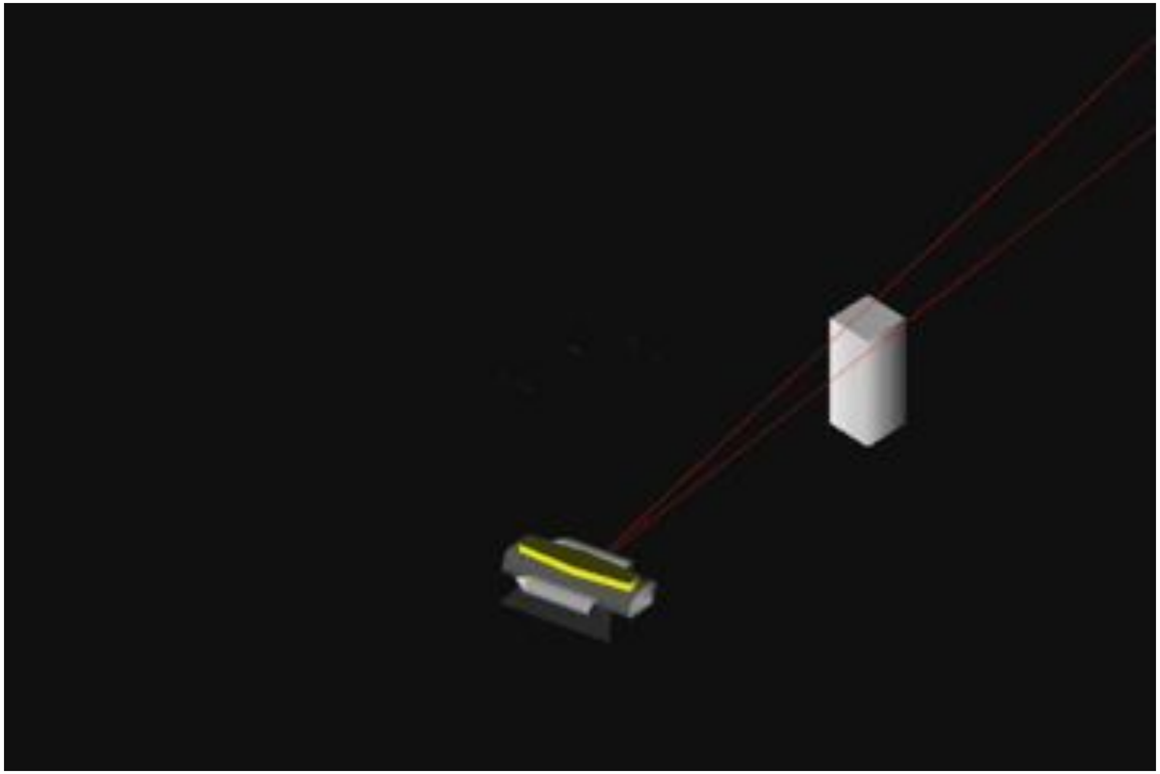
Maltese Cistern Mapping

- Step 6: **updated_occupancy_grid**
  - Modify the occupancy likelihood of each cell m_{ij} using sonar measurement z.
  - Add new probability to existing probability with *logit()* function



Maltese Cistern Mapping

- Results II: SLAM while moving



LAIR
Lab for Autonomous and Intelligent Robotics

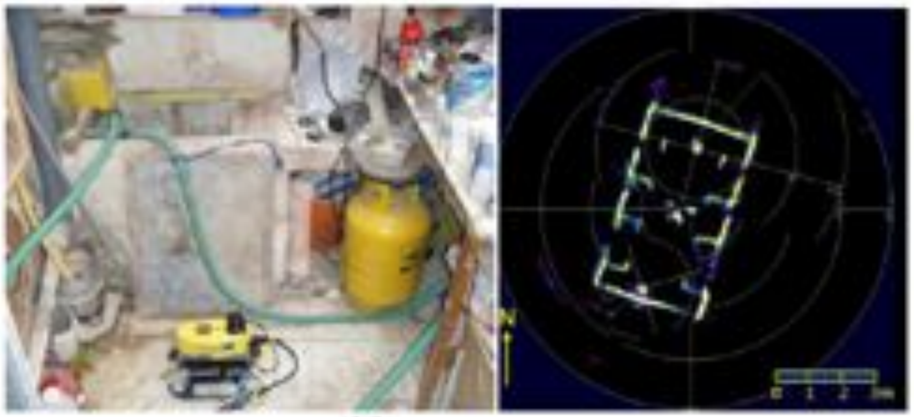
Maltese Cistern Mapping

- Results II: SLAM while moving

Table I. SLAM with stationary sonar scans vs. SLAM in motion.

	Site 24		Site 8		
Map type	Length (m)	Width (m)	Length (m)	Width (m)	Std. dev. (m)
Manual mosaics	5.6	1.4	8.9	2.3	0.18
Stationary SLAM	5.4	1.2	8.9	2.3	0.16
SLAM in motion	5.1	1.0	9.6	2.1	0.33

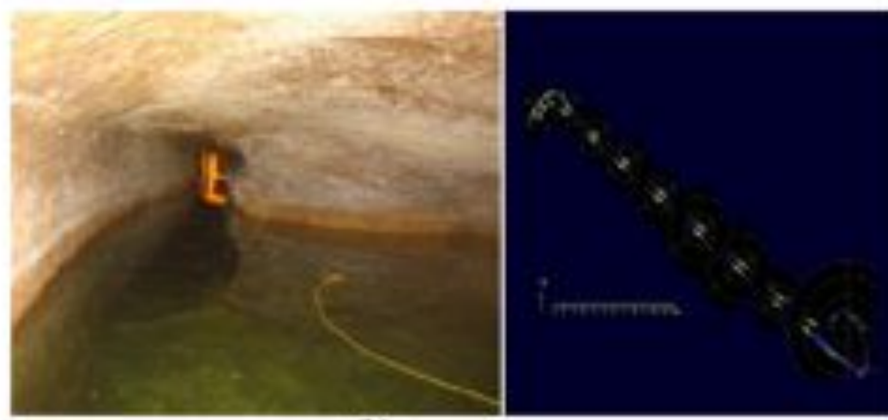
Maltese Cistern Mapping



(e)

(f)

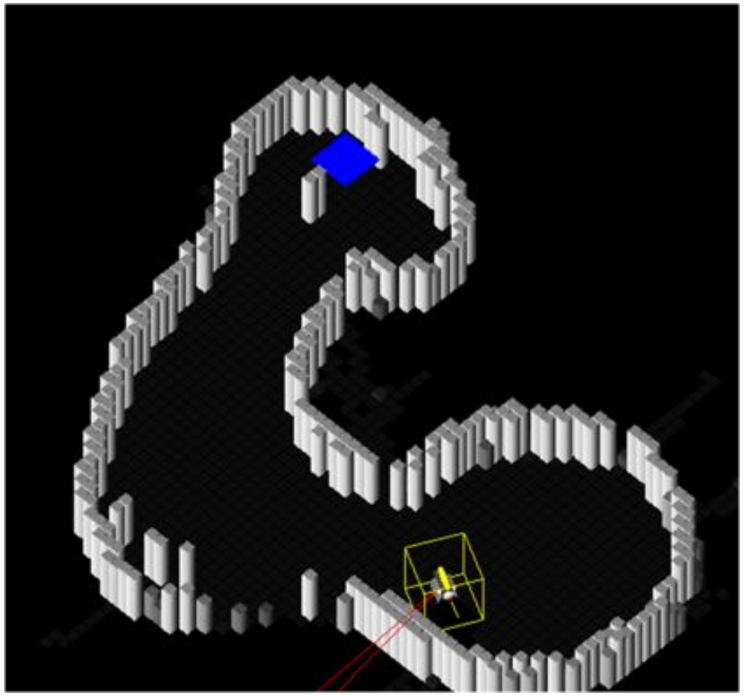
(g)

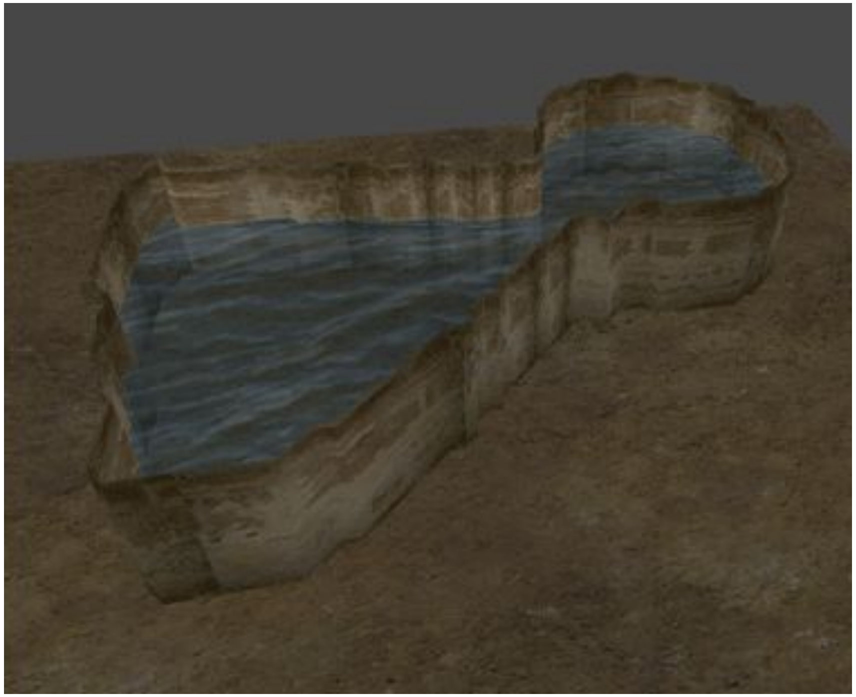


(h)

Maltese Cistern Mapping

- Results III: SLAM with stationary scans





Maltese Cistern Mapping

- Conclusions
  - First ever maps of ancient underwater cisterns created using robots
  - Sonar rotation head is slow, requiring slow motion of robot for accurate mapping
  - Tether snags, disturbances must be considered.
  - Over 33 maps created.

Outline

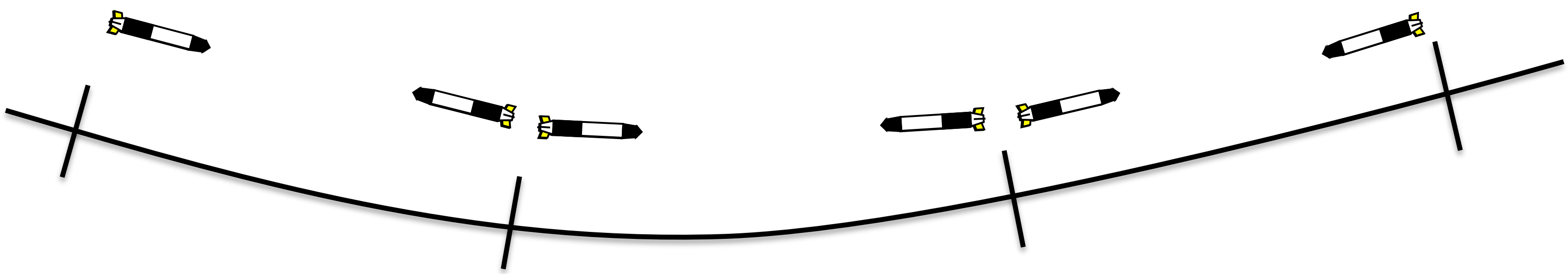
- Past Projects
- Maltese Cistern Mapping
- Distributed Multi Robot Boundary Tracking
- California Coastal Shark Tracking

Motivating Applications

- Applications
  - Oil Spill edge monitoring with AUVs
  - Border Security
  - Ice Edge Following
- Associated Issues
  - Some events to sample may be dynamic
  - Some areas of the edge may require more time to sample than others

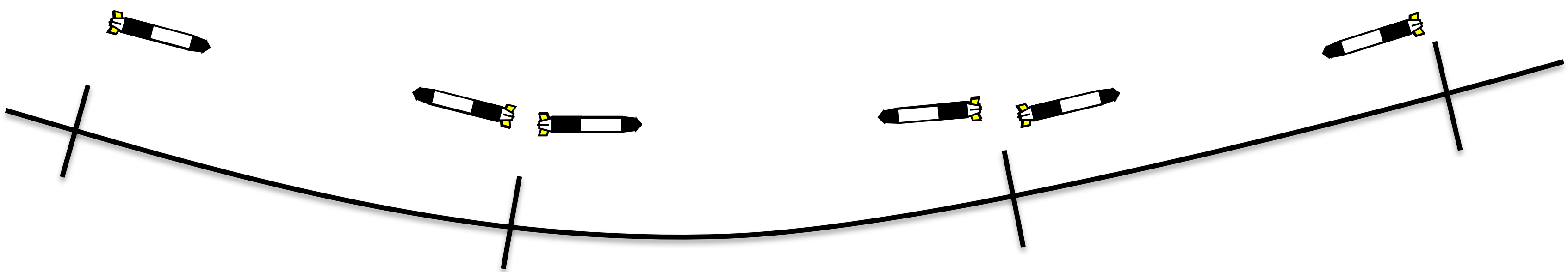
Motivating Applications

- Boundary Edge Sampling – 3 Robots
  - Out of Phase



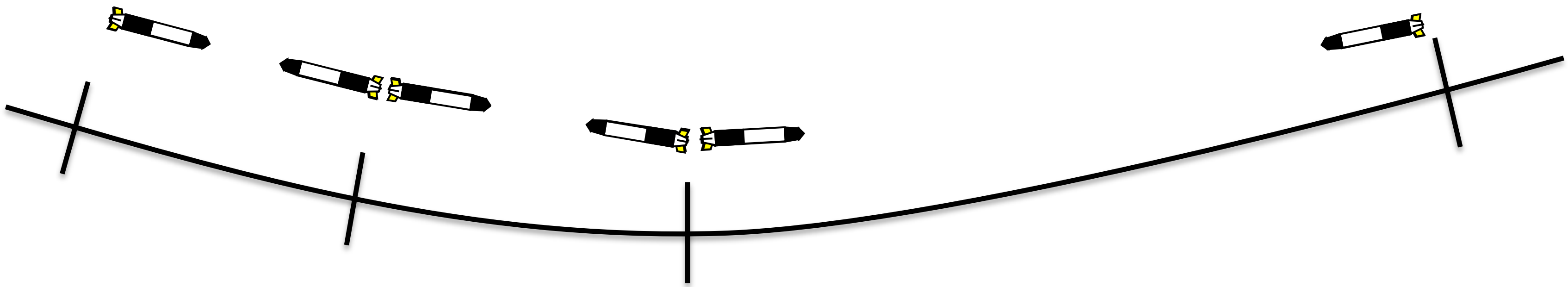
Motivating Applications

- Boundary Edge Sampling – 3 Robots
  - In Phase



Motivating Applications

- Boundary Edge Sampling – 3 Robots
  - In Phase
  - Balance Workloads



Motivating Applications

- Arctic Ice Edge Sampling
  - Arctic ice forms differently than in the past
  - Ice Algae forms integral part of ecosystem
  - Want to sample this algae and its activity
  - So, lets **track the edge** with **several AUVs**

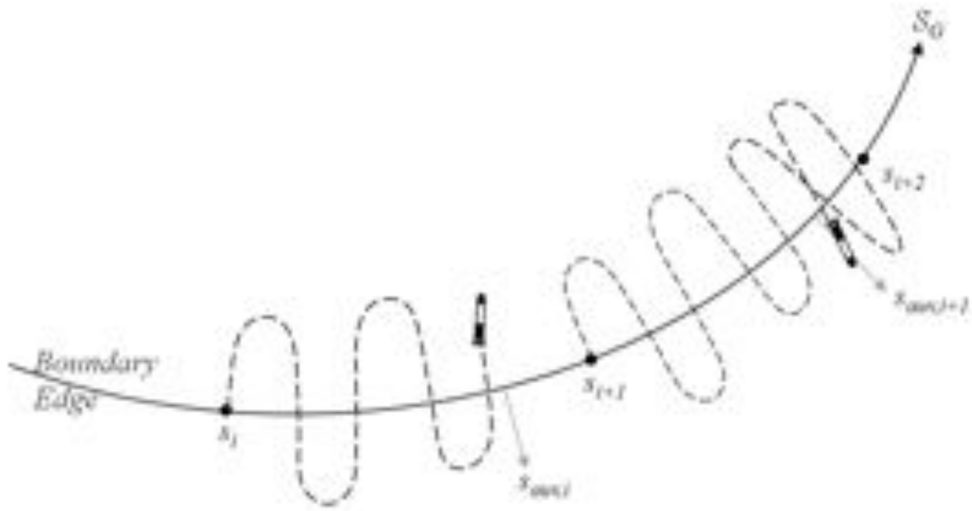
Photo by Doug Allen, CAMP Project

Related Work

- A. Joshi, et. al., "Experimental Validation of Cooperative Environmental Boundary Tracking with On-board Sensors", *in American Control Conference*, pp.2630-2635, June 2009.
- S. Charifa and M. Bikdash, "Adaptive boundary-following algorithm guided by artificial potential field for robot navigation", *IEEE Workshop on Robotic Intelligence in Informationally Structured Space*, pp.38-45, May 2009.
- D.A. Paley, et. al., "Cooperative Control for Ocean Sampling: The Glider Coordinated Control System", *IEEE Transactions on Control Systems Technology*, vol. 16, no.4, pp.735-744, July 2008.

Problem Definition

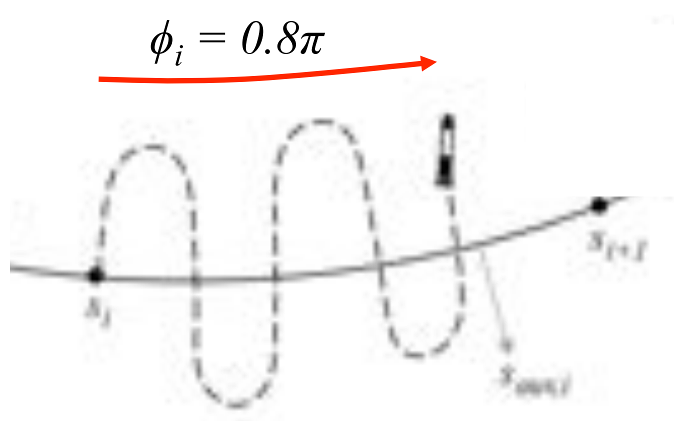
- Consider a continuous edge segment E
  - Define a coordinate frame where the S_G axis that follows E.
  - E is defined by two end points s_0 and s_n
  - E is partitioned into n sub-segments, each allocated to 1 robot.



Problem Definition

- Define Phase

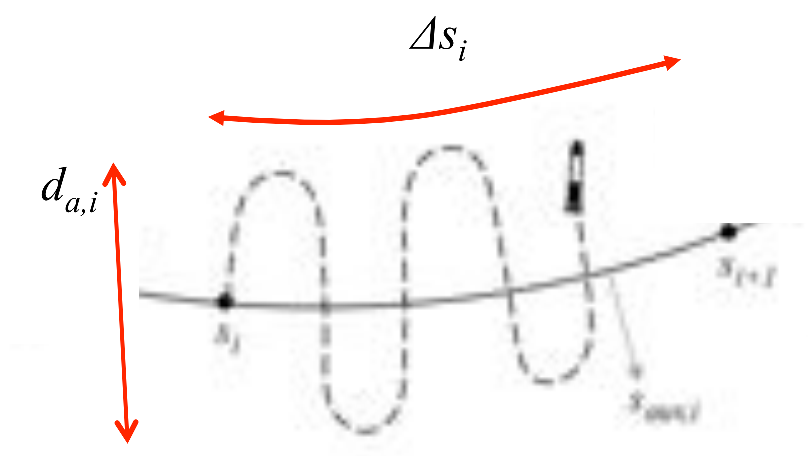
$$\phi_i = \left\{ \begin{array}{ll} \pi \frac{s_{rob,i} - s_i}{\Delta s_i} & if \;\; \dot{s}_{rob,i} > 0 \\ \pi \frac{s_{i+1} - s_{rob,i}}{\Delta s_i} + \pi & else \end{array} \right\}$$



Problem Definition

- Define Workload

$$\Psi_i \approx d_{a,i} \, \Delta s_i$$



Problem Definition

- For every robot pair, it is desirable to balance
  - Workload

$$e_{s,i} = \Psi_{i+1} - \Psi_i$$

  - Phase

$$e_{\phi,i} = \phi_{i+1} - \phi_i$$

Problem Definition

- To minimize these errors, we want each robot i to autonomously control its
  - Sub-segment boundary

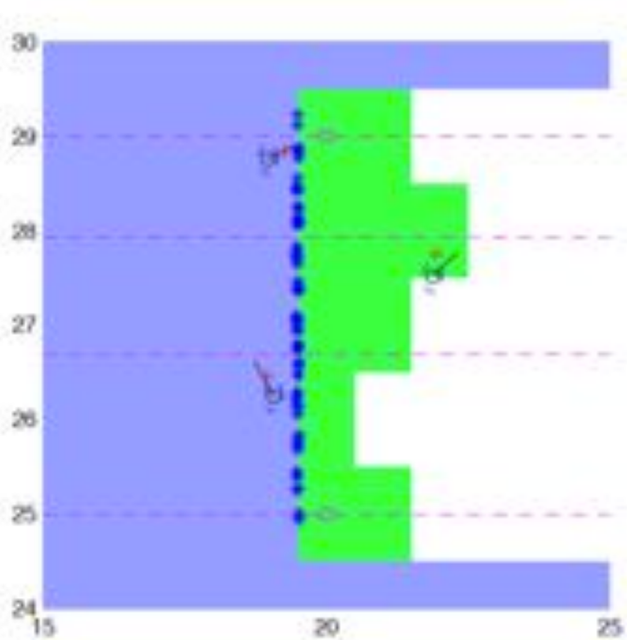
    $$s_i$$

  - Phase position

    $$\phi_i$$

Experiment Implementation

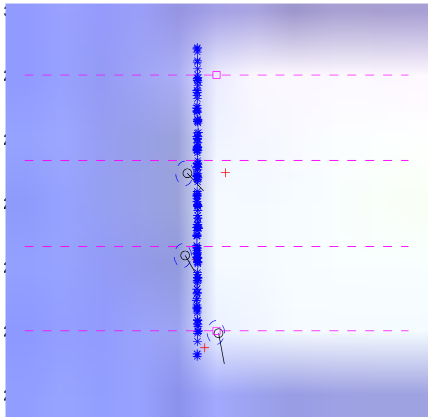
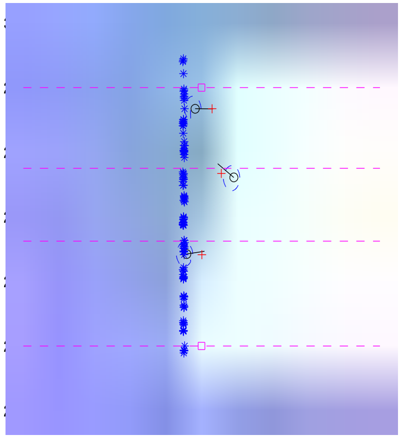
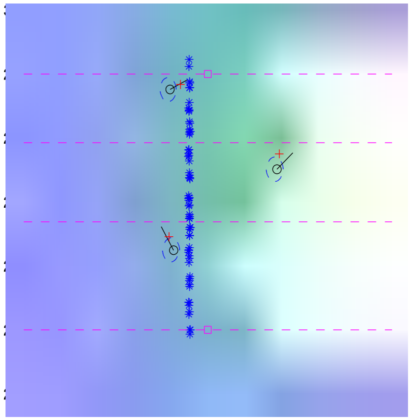
- Matlab Simulation

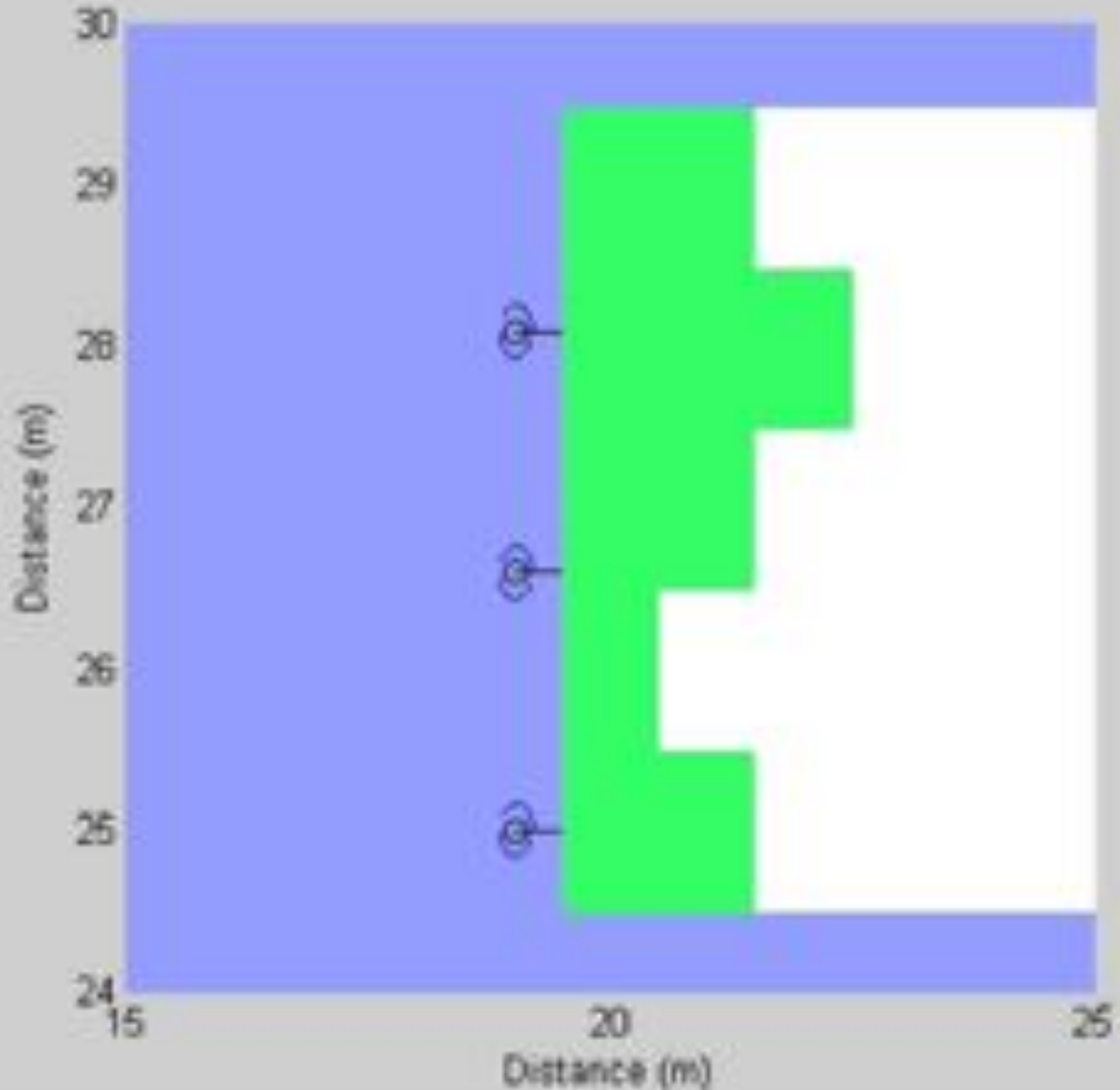


- iRobot Creates

Results

- Three different Simulations
  - No algae covering
  - Half algae covering
  - Random algae covering

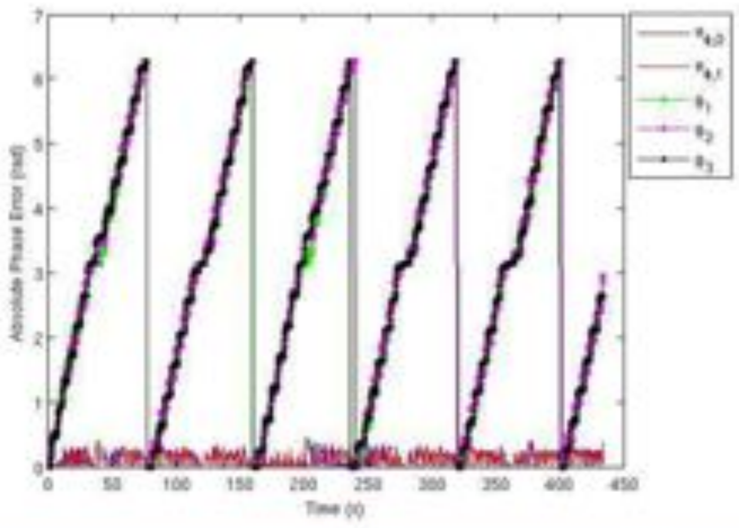


30
29
28
27
26
25
24
Distance (m)
15
20
25
Distance (m)

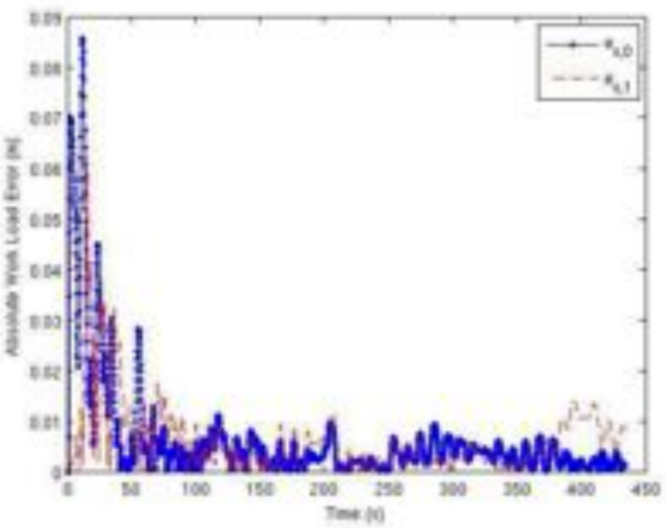
Results

- Experimental Results Example (no algae)

Phase



Workload



Conclusions

- Controller demonstrated to balance both workload and phase then tracking a boundary
- Tracking desired phase is not as accurate due to non linear paths
- Decentralized – controller only needs local information

Outline

- Past Projects
- Maltese Cistern Mapping
- Distributed Multi Robot Boundary Tracking
- California Coastal Shark Tracking

California Shark Tracking

- Improve accuracy of shark state estimation by first estimating the shark's behavioral mode.

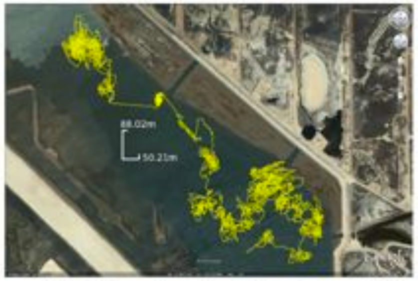
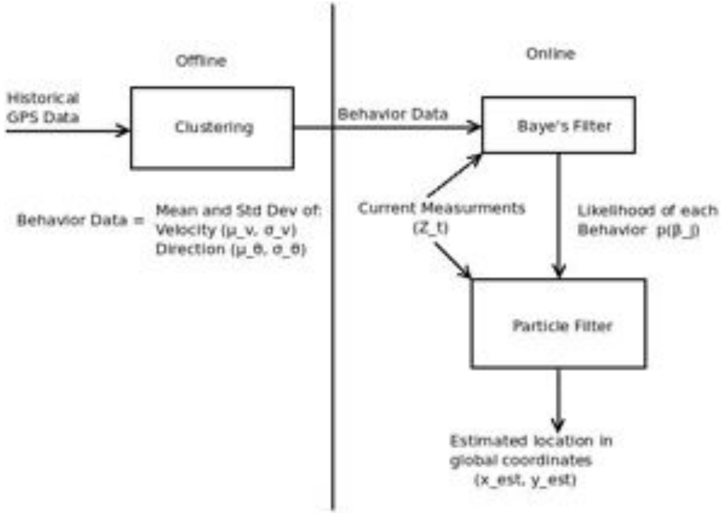


Fig. 1: Shovelnose Shark trajectory over 24 hours



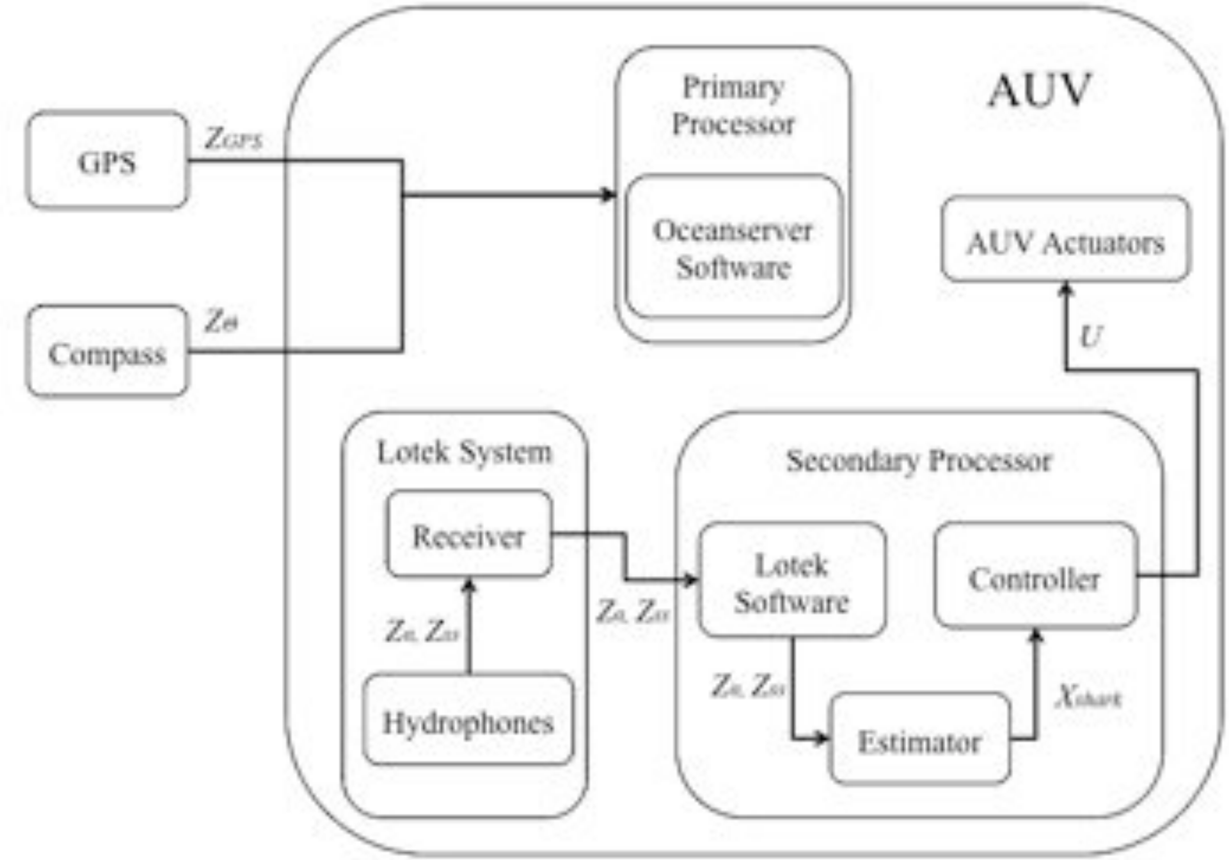
	Mean Error	Max Error
S.P.	1.34 m	117.17 m
L.A.	1.00 m	54.02 m

Problem Definition

- OceanServer Iver2 AUV
  - Sensors
    - GPS (when at surface)
    - 3 DOF compass
    - Altimeter
    - ADCP
    - Video Cameras
  - Actuators
    - Propeller
    - Control surfaces
  - Battery life 24 hours
  - Max Speed 4 knots
  - Depth Rated to 100 meters

Problem Definition

Problem Definition



Estimation

Given:

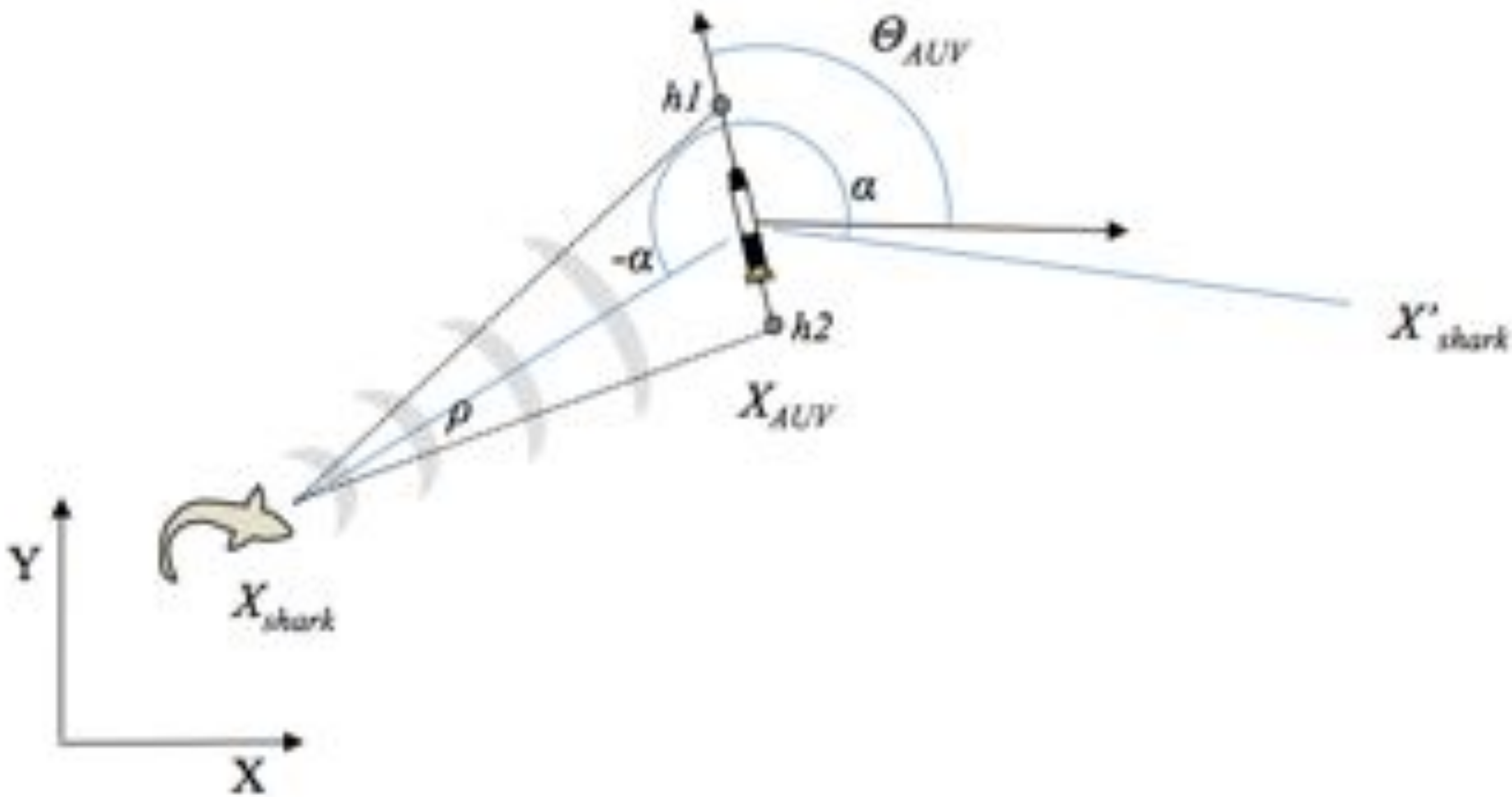
$$X_{auv,t} = [x_{auv}\ y_{auv}\ \theta_{auv}\ \dot{x}_{auv}\ \dot{y}_{auv}\ \dot{\theta}_{auv}]_t$$

$$Z_t = [Z_{ss}\ Z_{\alpha}]_t$$

Determine:

$$X_{shark,t} = [x_{shark}\ y_{shark}\ \theta_{shark}\ v_{shark}\ w_{shark}]_t$$

Estimation



Estimation

- What is a Particle this time?
  - A particle is an individual state estimate.
  - In our shark Tracking, a particle i has two components

$$\{ \underbrace{\boldsymbol{X_{shark}}^{i}}_{\text{State}} \quad \underbrace{w^{i}}_{\text{Weight}} \}$$

1. The state is $\boldsymbol{X_{shark}} = [x \;\; y \;\; \theta \;\; v \; w]$
2. The weight $\boldsymbol{w}$ that indicates it's likelihood of being the correct state.

Estimation

- Prediction Step
  - Let all particle states propagate according to a random sampling of a simple first-order motion model

Algorithm 1 PF_Shark_State_Estimator($\{X^p\}, X_{auv}, Z_\alpha$)

//Prediction
for all p particles **do**
  $v^p_{rand} \leftarrow v^p + \text{randn}(0, \sigma_v)$
  $\theta^p_{rand} \leftarrow \theta^p + \text{randn}(0, \sigma_\theta)$
  $x^p_{shark} \leftarrow x^p_{shark} + v^p_{rand} * \cos(\theta^p_{rand}) * \Delta t$
  $y^p_{shark} \leftarrow y^p_{shark} + v^p_{rand} * \sin(\theta^p_{rand}) * \Delta t$
  $v^p \leftarrow \gamma_{vt} * v^p + (1 - \gamma_{vt}) * \frac{\sqrt{(y^p_{shark} - y^p_{prev})^2 + (x^p_{shark} - x^p_{prev})^2}}{\Delta t}$
  $\theta^p \leftarrow \theta^p_{rand}$
  if α is valid **then**
    $\alpha^p_{exp} \leftarrow \text{atan2}(y_{auv} - y^p_{shark}, x_{auv} - x^p_{shark}) - \theta_{auv}$
    $\alpha^p_{exp} \leftarrow g(\alpha^p_{exp})$
    $w^p \leftarrow h(Z_\alpha, \alpha^p_{exp})$
  end if
end for

//Correction
if α is valid **then**
  $\{X^p\}_{temp} \leftarrow \{X^p\}$ for all p
  for all p particles **do**
    $X^p \leftarrow RandParticle(\{X^p\}_{temp})$
  end for
end if

Estimation

- Prediction Step
  - If a new Acoustic measurement is received, calculate the weight of each particle.

Algorithm 1 PF_Shark_State_Estimator($\{X^p\}$, X_{auv}, Z_α)

//Prediction
for all p particles **do**
$v^p_{rand} \leftarrow v^p + \text{randn}(0, \sigma_v)$
$\theta^p_{rand} \leftarrow \theta^p + \text{randn}(0, \sigma_\theta)$
$x^p_{shark} \leftarrow x^p_{shark} + v^p_{rand} * \cos(\theta^p_{rand}) * \Delta t$
$y^p_{shark} \leftarrow y^p_{shark} + v^p_{rand} * \sin(\theta^p_{rand}) * \Delta t$
$v^p \leftarrow \gamma_{vt} * v^p + (1 - \gamma_{vt}) * \frac{\sqrt{(y^p_{shark} - y^p_{prev})^2 + (x^p_{shark} - x^p_{prev})^2}}{\Delta t}$
$\theta^p \leftarrow \theta^p_{rand}$
if α is valid **then**
$\alpha^p_{exp} \leftarrow \text{atan2}(y_{auv} - y^p_{shark}, x_{auv} - x^p_{shark}) - \theta_{auv}$
$\alpha^p_{exp} \leftarrow g(\alpha^p_{exp})$
$w^p \leftarrow h(Z_\alpha, \alpha^p_{exp})$
end if
end for

//Correction
if α is valid **then**
$\{X^p\}_{temp} \leftarrow \{X^p\}$ for all p
for all p particles **do**
$X^p \leftarrow RandParticle(\{X^p\}_{temp})$
end for
end if

Estimation

- Correction Step
  - If a new Acoustic measurement is received, resample the distribution by randomly repopulating it with particles according to their weights.

Algorithm 1 PF_Shark_State_Estimator($\{X^p\}$, X_{auv}, Z_α)

//Prediction
for all p particles **do**
  $v^p_{rand} \leftarrow v^p + \text{randn}(0, \sigma_v)$
  $\theta^p_{rand} \leftarrow \theta^p + \text{randn}(0, \sigma_\theta)$
  $x^p_{shark} \leftarrow x^p_{shark} + v^p_{rand} * \cos(\theta^p_{rand}) * \Delta t$
  $y^p_{shark} \leftarrow y^p_{shark} + v^p_{rand} * \sin(\theta^p_{rand}) * \Delta t$
  $v^p \leftarrow \gamma_{vt} * v^p + (1 - \gamma_{vt}) * \frac{\sqrt{(y^p_{shark} - y^p_{prev})^2 + (x^p_{shark} - x^p_{prev})^2}}{\Delta t}$
  $\theta^p \leftarrow \theta^p_{rand}$
  if α is valid **then**
    $\alpha^p_{exp} \leftarrow \text{atan2}(y_{auv} - y^p_{shark}, x_{auv} - x^p_{shark}) - \theta_{auv}$
    $\alpha^p_{exp} \leftarrow g(\alpha^p_{exp})$
    $w^p \leftarrow h(Z_\alpha, \alpha^p_{exp})$
  end if
end for

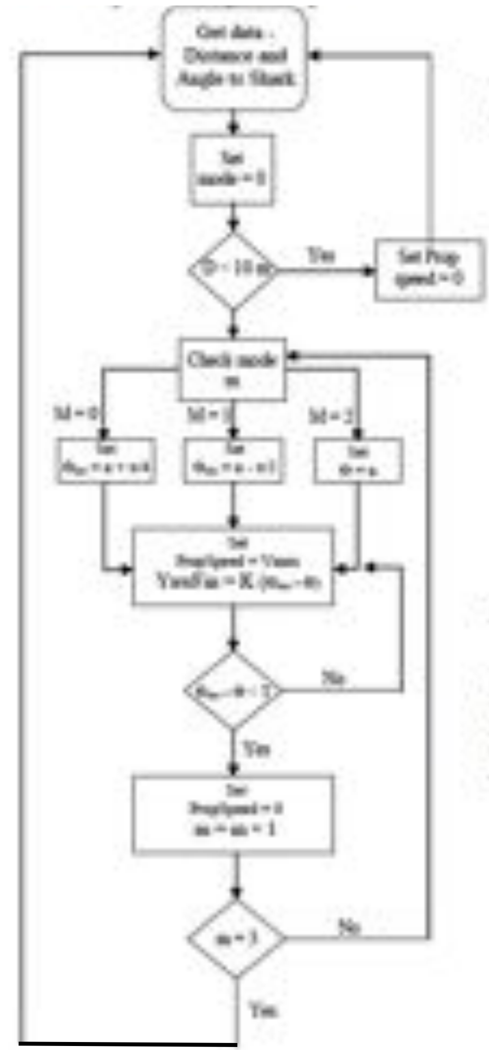
//Correction
if α is valid **then**
  $\{X^p\}_{temp} \leftarrow \{X^p\}$ **for all** p
  for all p particles **do**
    $X^p \leftarrow RandParticle(\{X^p\}_{temp})$
  end for
end if

LAIR
Lab for Autonomous and Intelligent Robotics

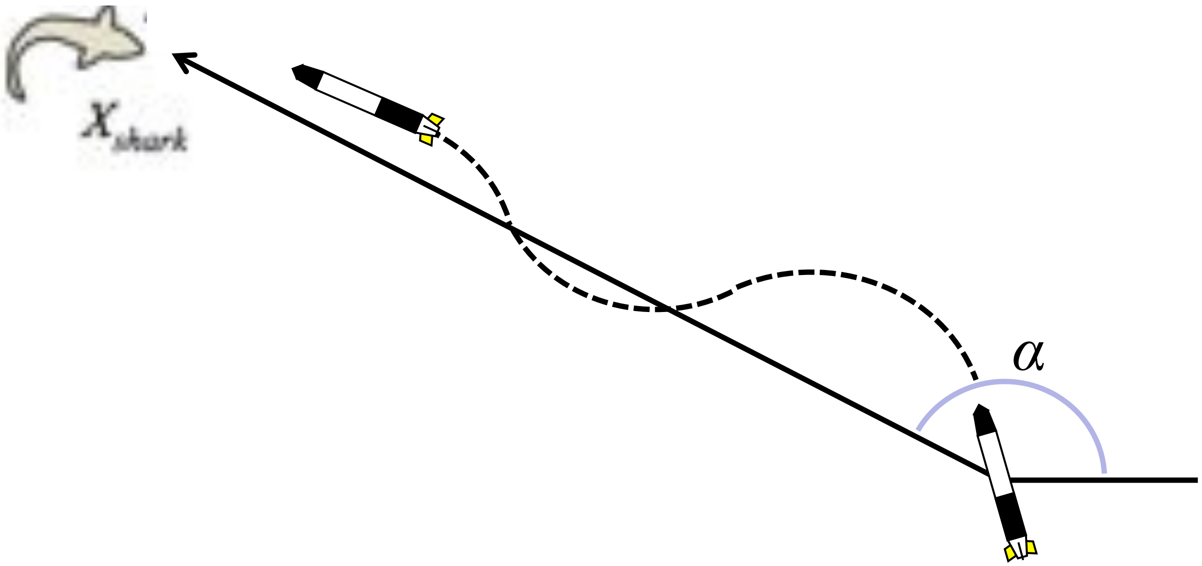
Control

- Three mode controller
  - Based on current estimate of bearing to shark α:

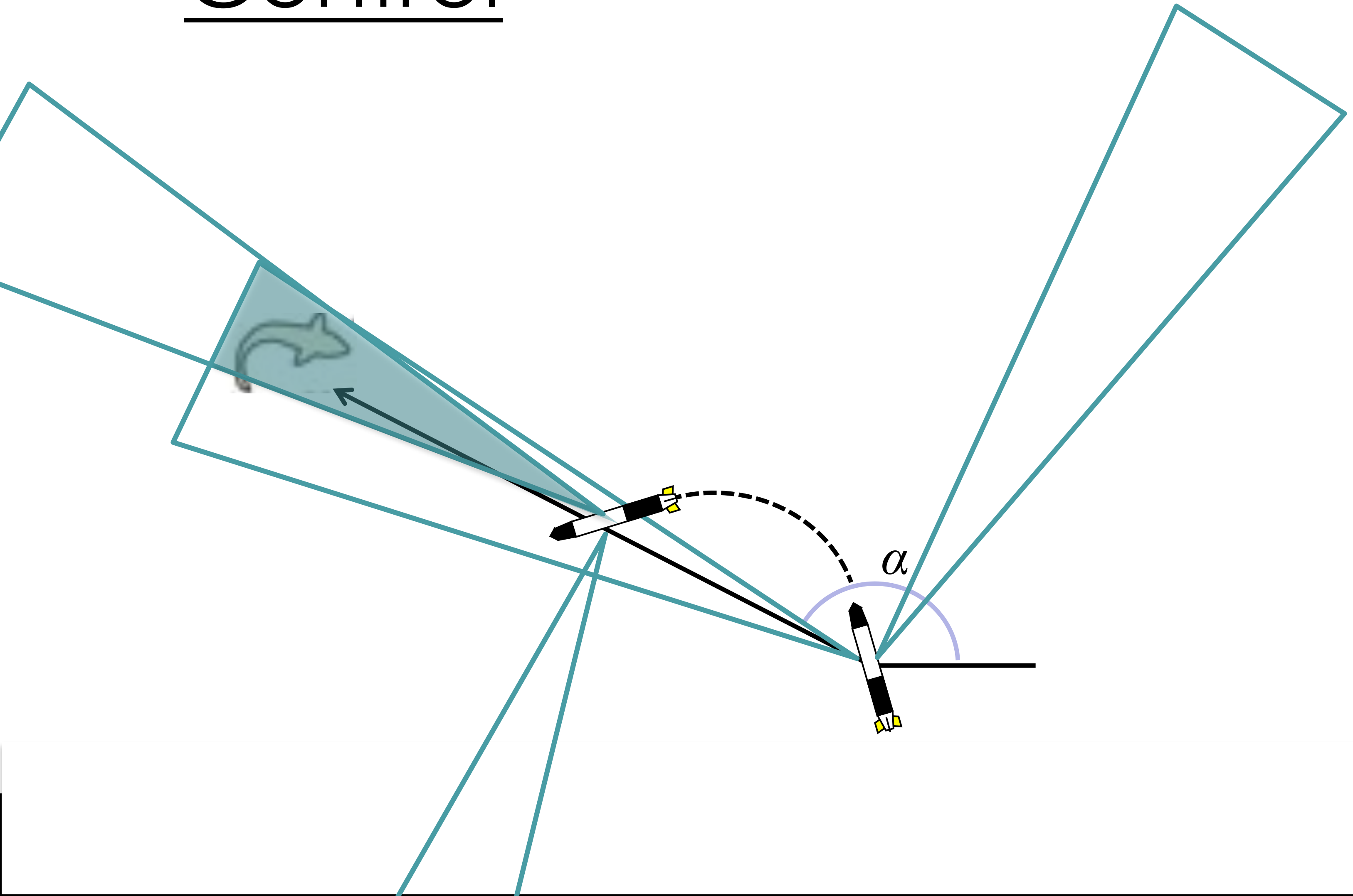
1. Track desired yaw angle
   $\theta_{des} = \alpha + \pi/4$
2. Track desired yaw angle
   $\theta_{des} = \alpha - \pi/4$
3. Track desired yaw angle
   $\theta_{des} = \alpha$



Control

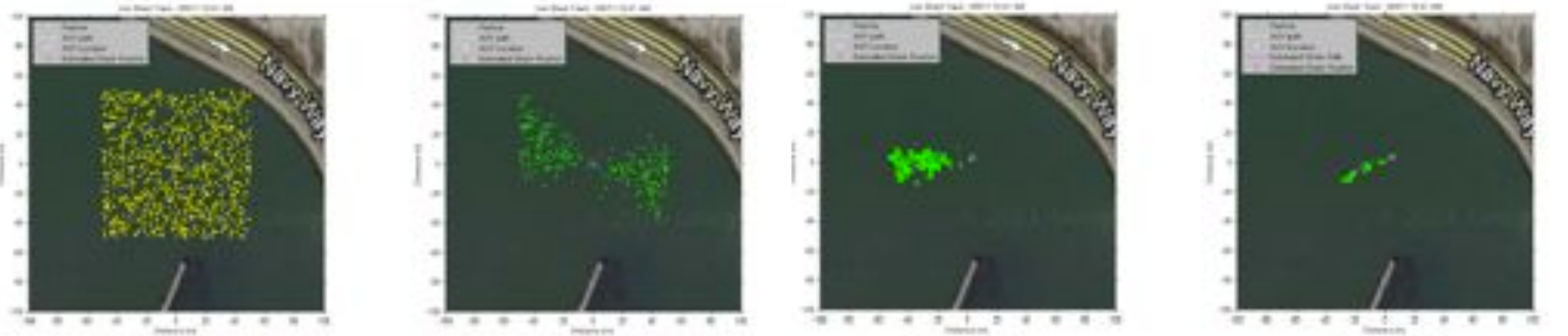


Control



Results

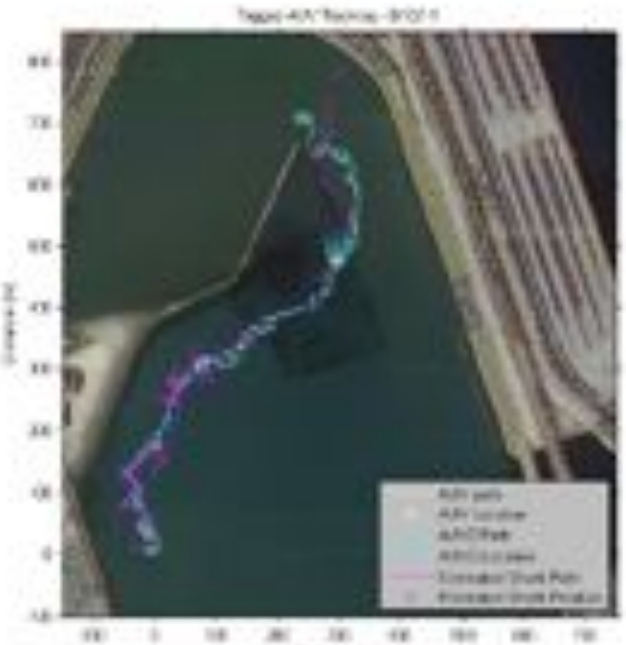
- One AUV tracking a tagged second AUV

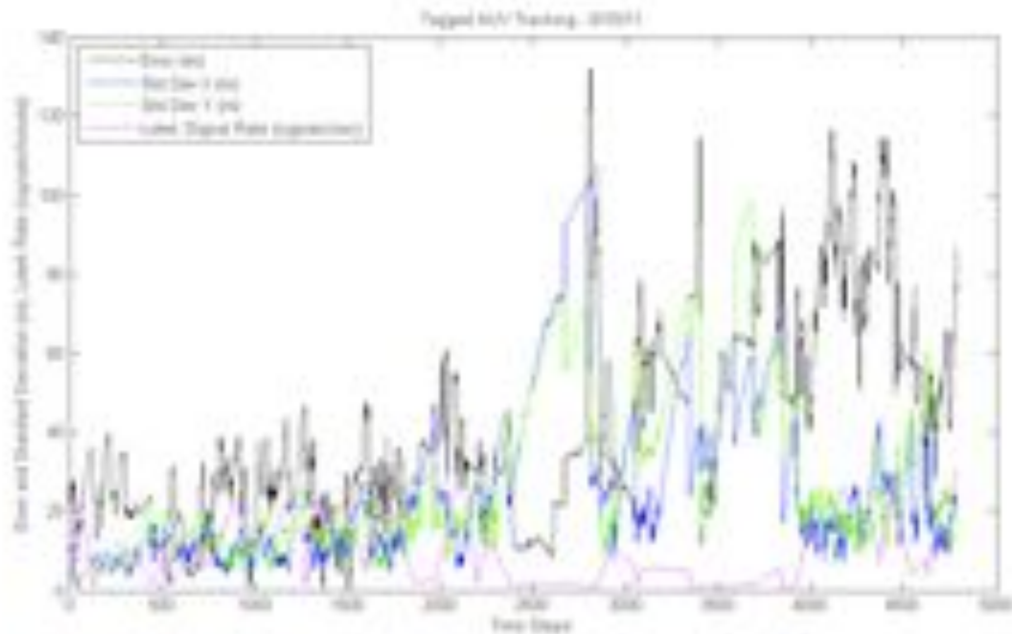


(a) t = 0.0 s (b) t = 2.54 s (c) t = 13.92 s (d) t = 32.37 s

Results

- One AUV tracking a tagged second AUV





Experiments

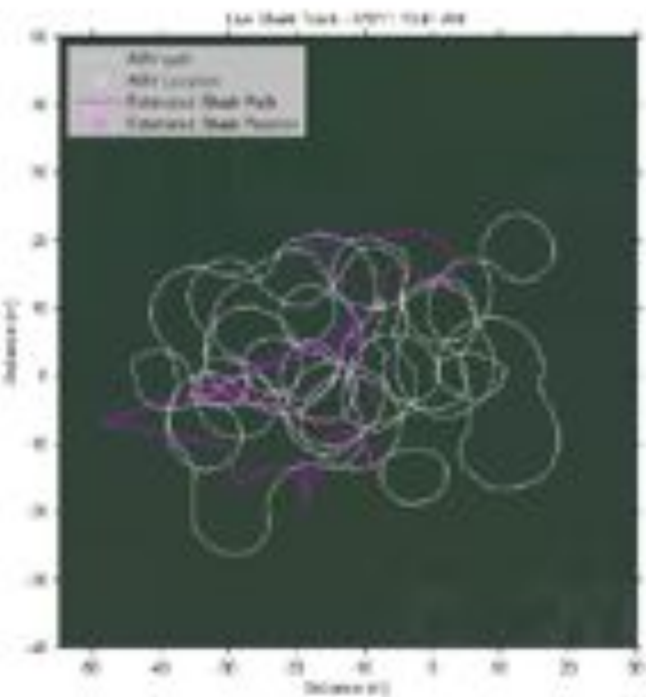
- SeaPlane Lagoon, Po [illegible] Angeles, CA

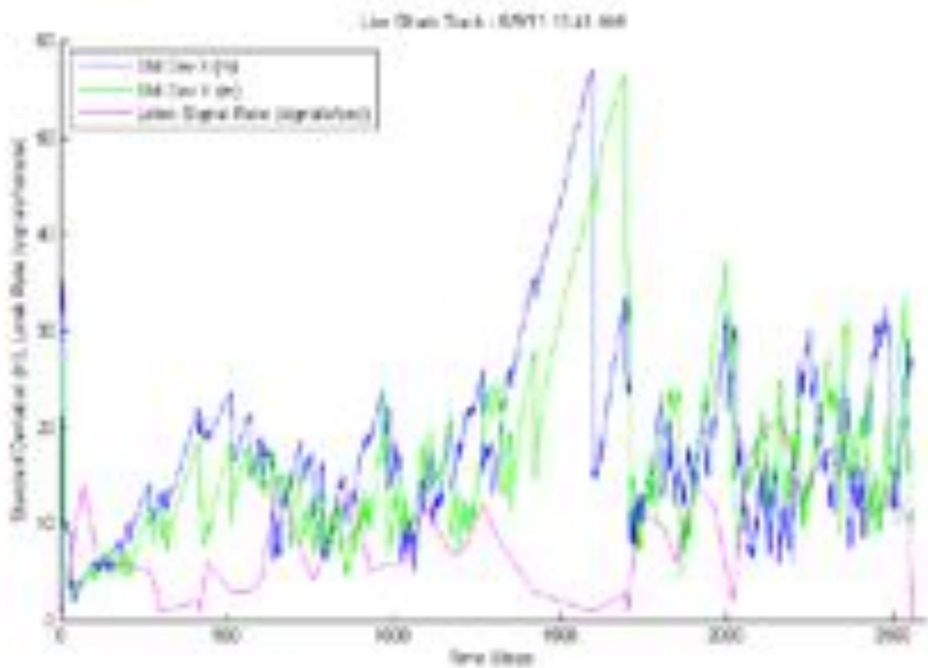
Experiments

- SeaPlane Lagoon, Po Angeles, CA

Results

- One AUV tracking a tagged Leopard shark





Results

- Summary

Mission Name	Date	Time	Mission Length min	Avg Error meters	Min Error meters	Max Error meters	Area Covered meters
darkTrackA	6/9/11	10:41 AM	48.16	n/a	n/a	n/a	104.29 x 85.72
darkTrackB	6/9/11	12:07 PM	37.25	n/a	n/a	n/a	42.87 x 90.10
darkTrackC	6/9/11	7:47 PM	41.77	n/a	n/a	n/a	120.14 x 81.54
darkTrackD	6/9/11	3:15 PM	1:41.27	n/a	n/a	n/a	103.42 x 44.69
outerTrack	6/10/11	11:19 AM	1:58.28	41.75	6.65	160.67	386.23 x 718.28
stationaryTrackA	6/7/11	4:19 PM	4.17	7.01	0.75	15.46	35.56 x 45.26
stationaryTrackB	6/7/11	4:34 PM	10.56	14.86	1.53	47.26	33.54 x 35.26
stationaryTrackC	6/7/11	4:47 PM	16.26	23.50	3.13	47.34	41.87 x 56.35

Future Work

- AUVs for Shark Tracking
- 3D Cistern & cave mapping
- Maltese Coastal Shipwreck mapping
- Arctic under ice sampling
- Generalize decentralized controller for multiple edges
- Decentralized Multi-Robot Motion Planning
- Estimation and control of fish swarms

Thank you

- For more details see:

http://lair.calpoly.edu