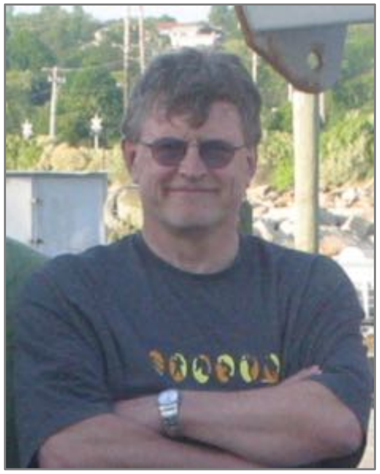
Autonomy in the Open
<Click Here to Download>

March 29th, 2017

Michael R. Benjamin
Henrik Schmidt
MIT Dept. of Mechanical Engineering
Laboratory for Autonomous Marine Sensing Systems
Computer Science and Artificial Intelligence Lab

mikerb@mit.edu
henrik@mit.edu

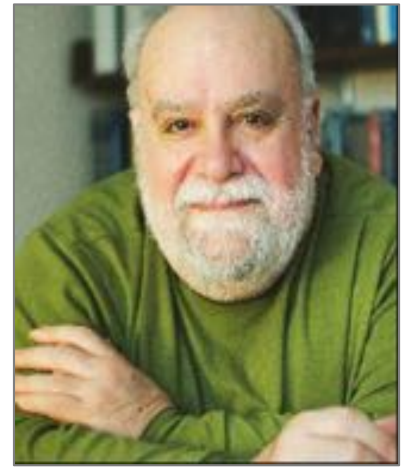
Collaborators



Prof. Henrik Schmidt (MIT)

Prof. John Leonard (MIT)

Prof. Paul Newman (Oxford)



Prof. Chryssostomidis (MIT)

Dr. Michael Novitzky

Dr. Paul Robinette

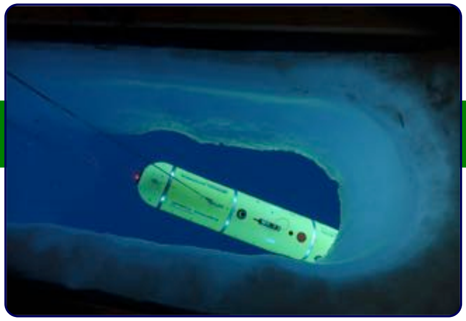
Thank You!!

Marine Autonomy

- Marine Autonomy – The Robots and Players

- Recent Past and Present. (And future?)

2006

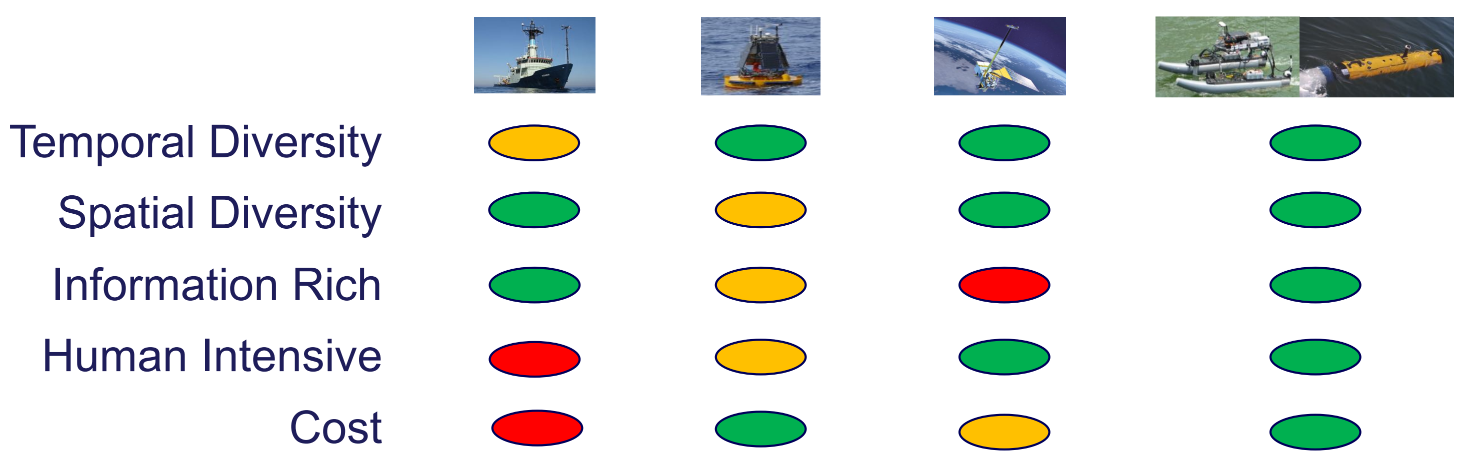


2016

2026?

- The Role of Open Source Software

How We Sense the Ocean

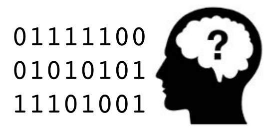


Who Has a Stake in Ocean Autonomy?

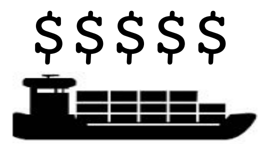
Who Has a Stake in Ocean Autonomy?

Defense

Science



Shipping



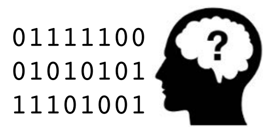
Oil & Gas

Which Autonomous Vehicles are Key? For Whom?

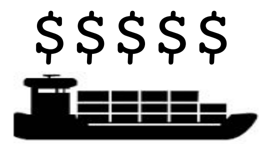
Which Autonomous Vehicles are Key?

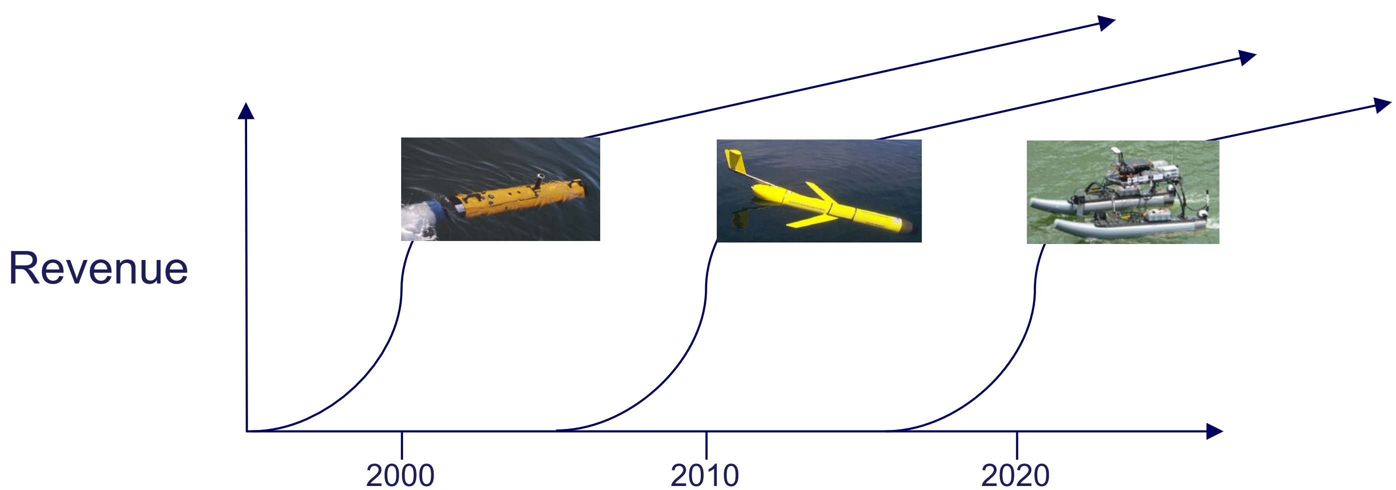
Defense

Science



Shipping



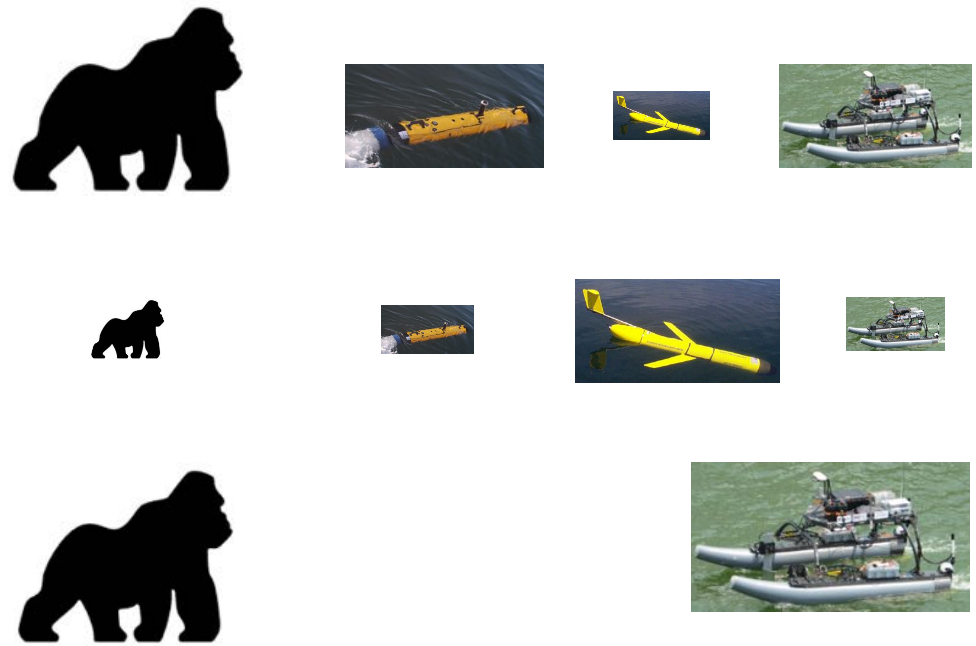


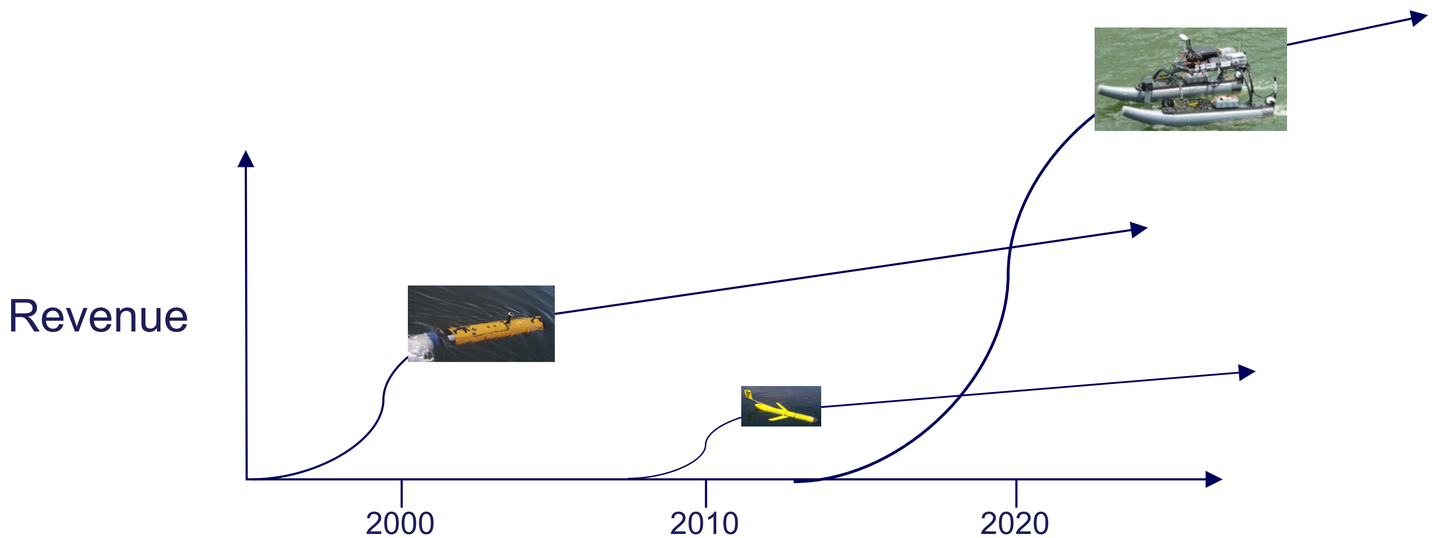
Which Autonomous Vehicles are Key?

Defense

Science

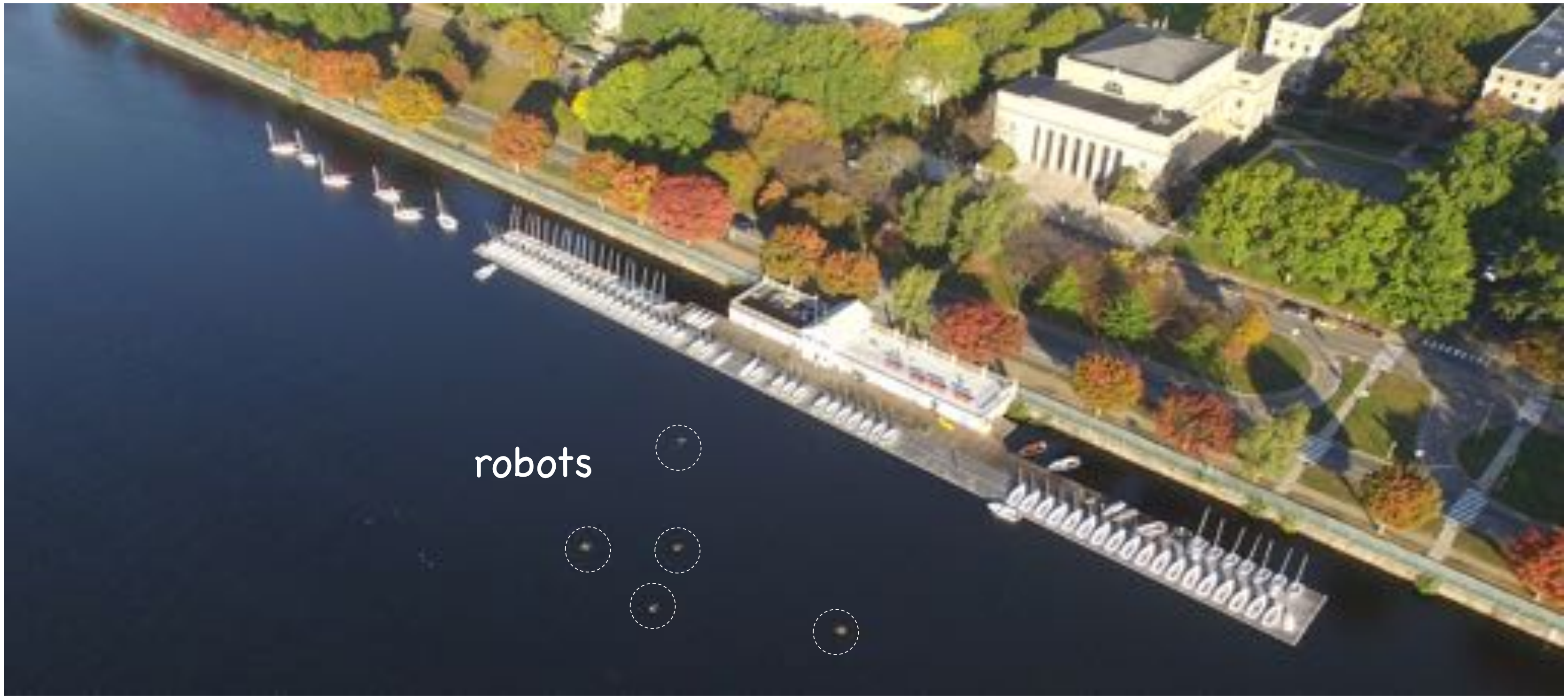
Shipping





The MIT Marine Autonomy Bay

http://oceanai.mit.edu/pavlab



- Laboratory for Autonomous Marine Sensing Systems (MECHE)

  henrik@mit.edu

- Marine Robotics Group (CSAIL)

  jleonard@mit.edu

- The AUV Laboratory (MIT Sea Grant)

  chrys@mit.edu

MIT Marine Robotic Platforms

The Bluefin SandShark One-Person Portable UUV

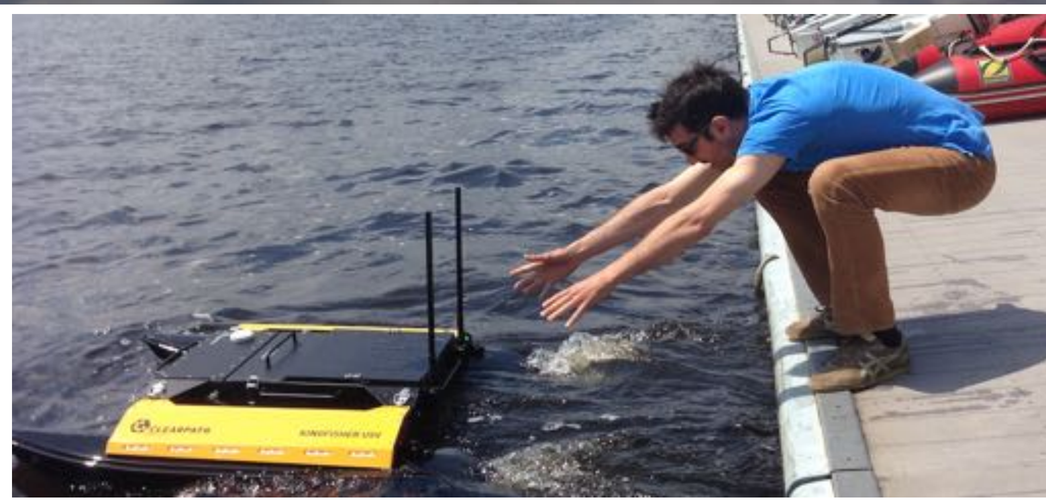


MIT Marine Robotic Platforms

The Bluefin 21-inch UUV (Macrura and Unicorn)

MIT Marine Robotic Platforms

The Clearpath Robotics Kingfisher USV



MIT Marine Robotic Platforms

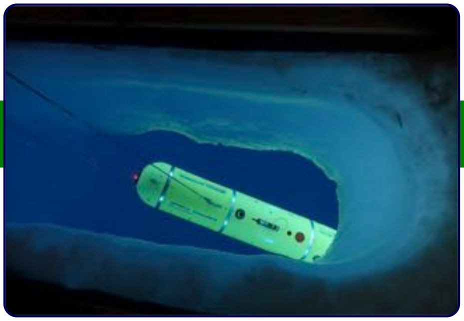
The WAM-V Unmanned Surface Vehicle

MIT 2.680
Marine Autonomy, Sensing and Communications

Marine Autonomy

- Marine Autonomy – The Robots and Players
- Recent Past and Present. (And future?)


2006


2016

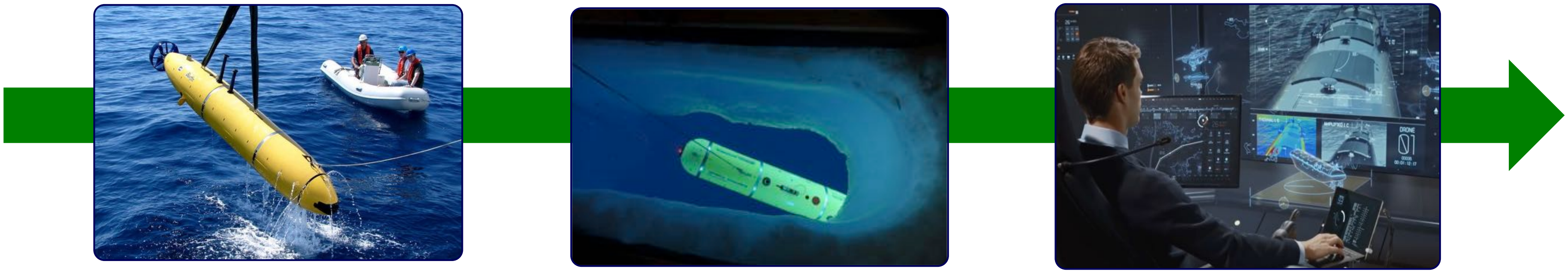

2026?

- The Role of Open Source Software

Monterey Bay 2006

PLUSNet Field Trials on the R/V New Horizon

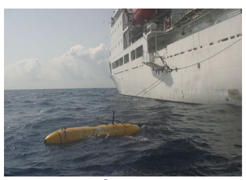
Marine Autonomy



2006 2016 2026?

- Payload Autonomy supported on virtually all commercial platforms.
- The MIT MOOS-IvP Open Source Project Launched.

(35 work-years, 130,000 lines of code, 30+ applications)


Bluefin-21


AMS Datamaran


Teledyne Gavia AUV


SeaRobotics SCOAP USV


Kingfisher M200


Kingfisher M100


Bluefin-9


MIT/Hover Kayak


REMUS 600


REMUS 100


Ocean Explorer


RMS Scouts


Iver-2


SeaRobotics USV

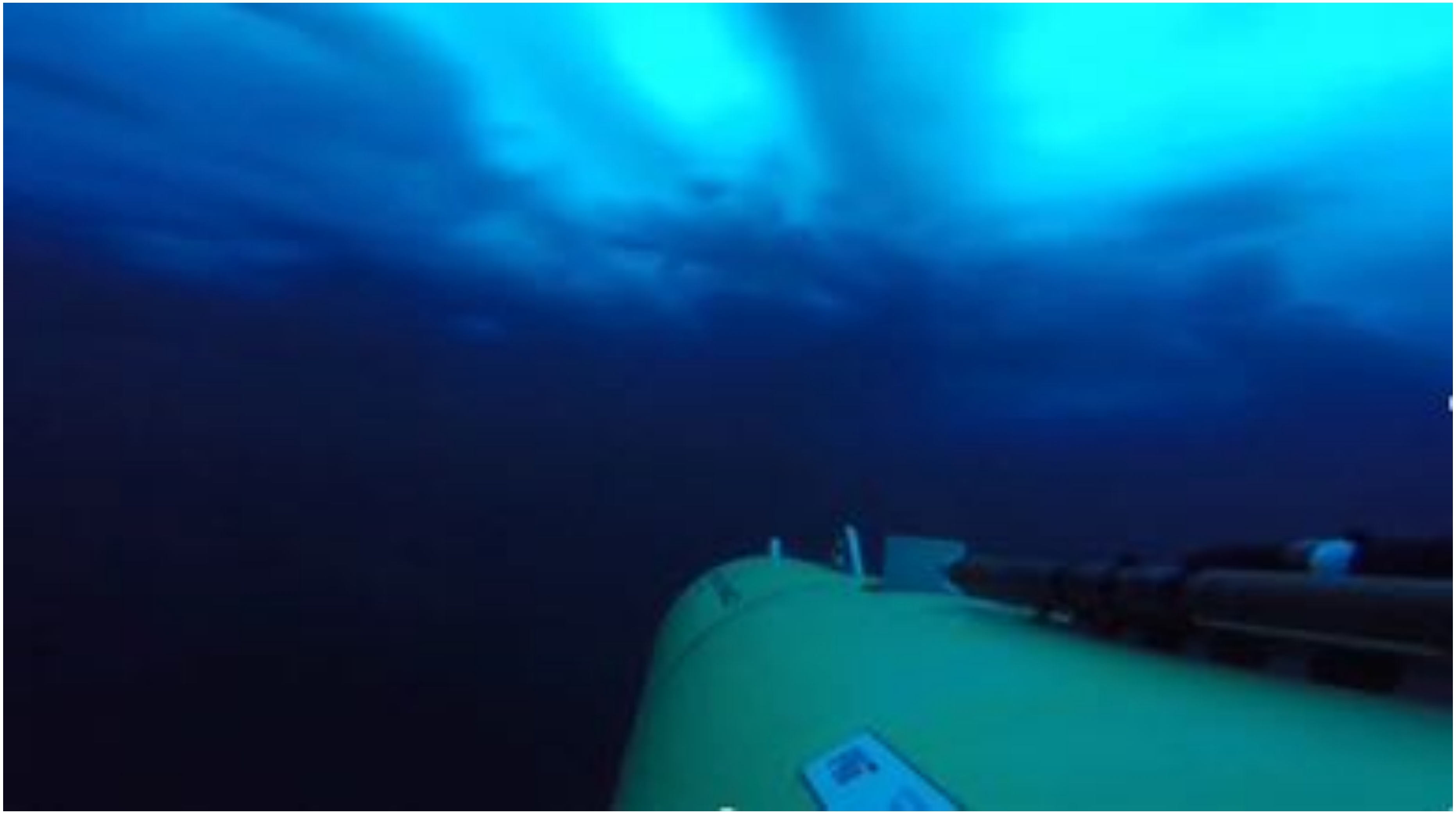

H-Scientific USV


Bobcat tractor


WAM-V


Folaga

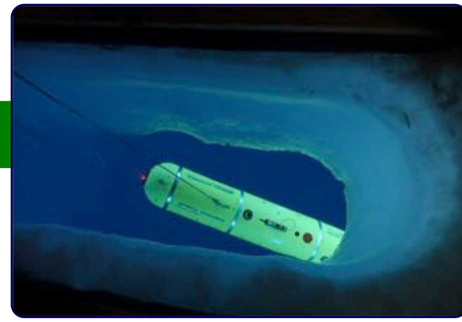
ICEX 2016



The MIT Bluefin 21-inch UUV

Marine Autonomy

2006

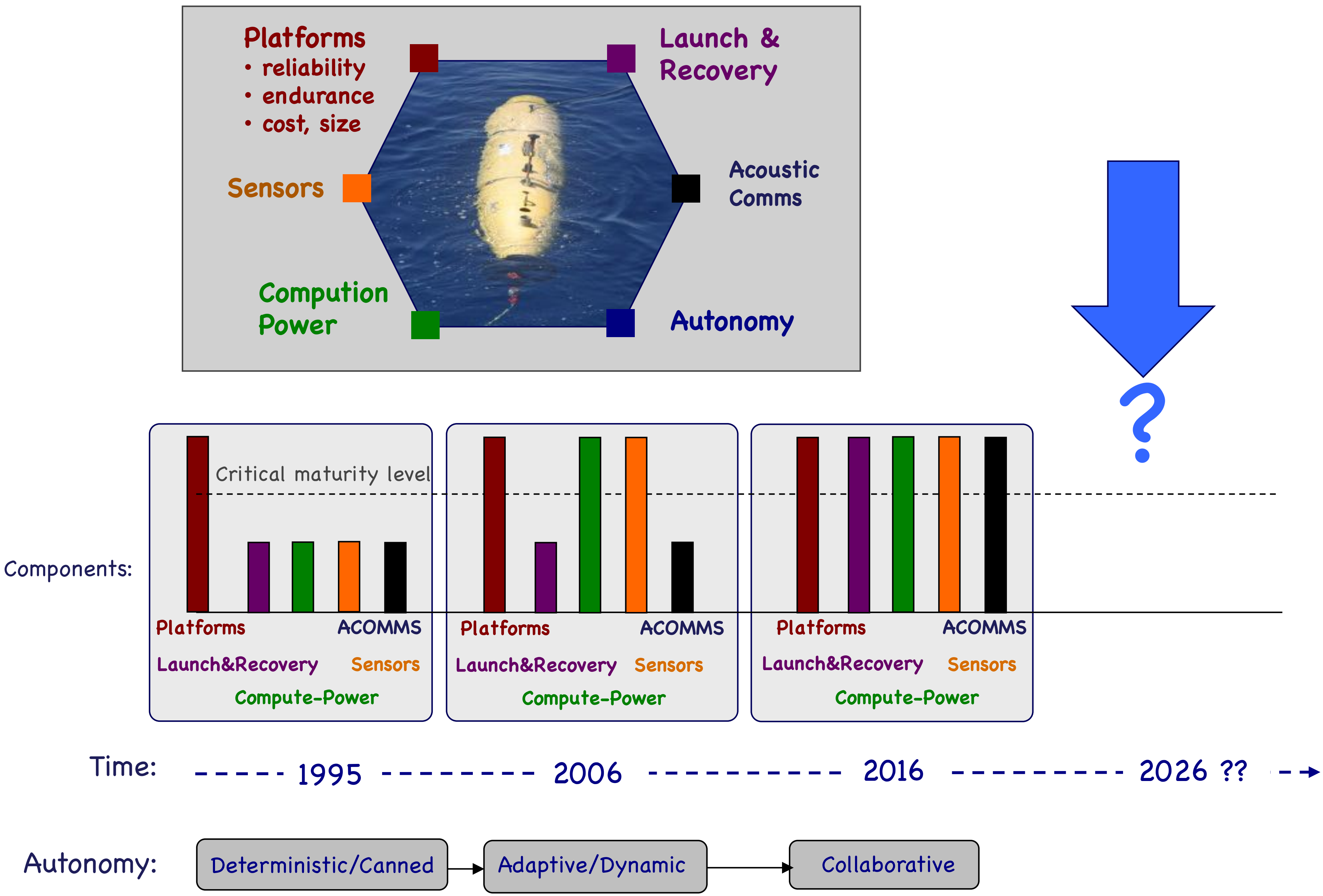


2016

2026?

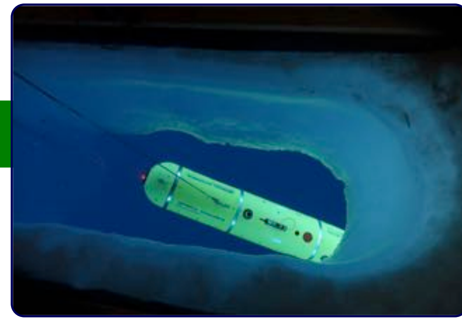
?

Trends in Component Technologies



Marine Autonomy

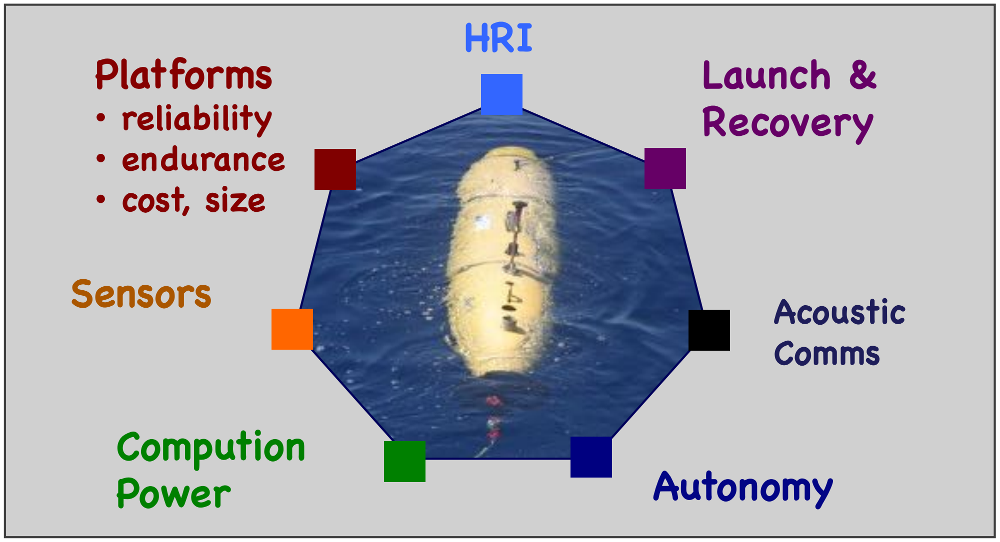
2006



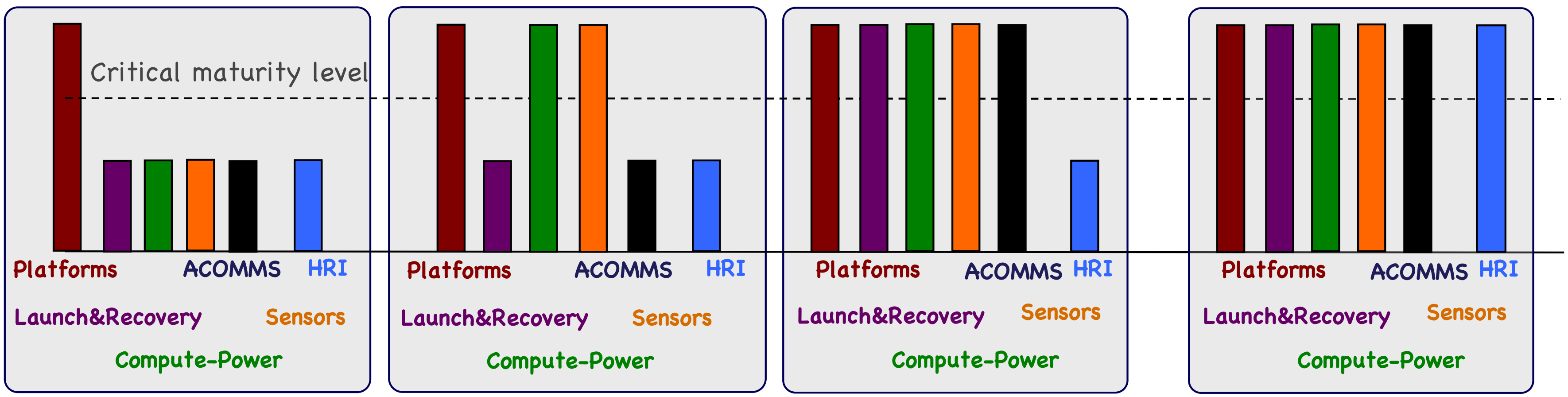
2016

2026?

Trends in Component Technologies



Components:



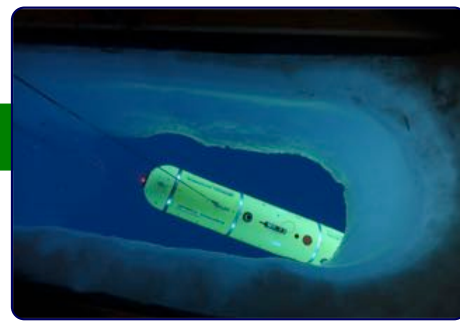
Time: ----- 1995 --------- 2006 ----------- 2016 ---------- 2026 ?? -->

Autonomy: Deterministic/Canned → Adaptive/Dynamic → Collaborative

Marine Autonomy

- Marine Autonomy – The Robots and Players
- Recent Past and Present. (And future?)

2006



2016

2026?

- The Role of Open Source Software

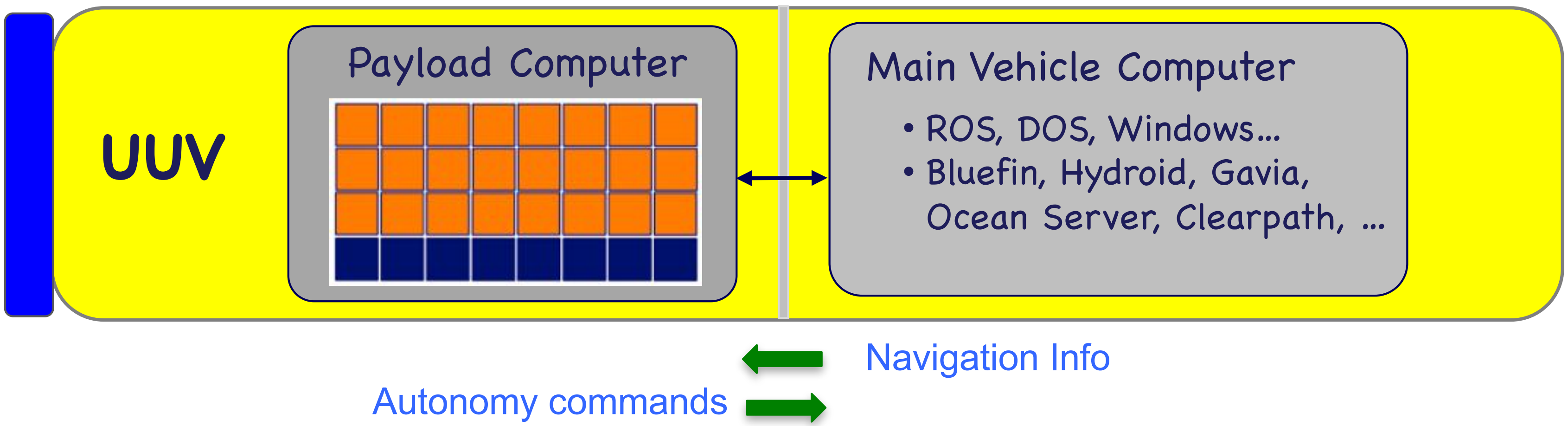
What is MOOS? MOOS-IvP?

- MOOS is Open Source Robot Middleware from Oxford
- MOOS is application communications architecture

- IvP is Open Source Autonomyware from MIT
- IvP is an autonomous decision-making architecture.

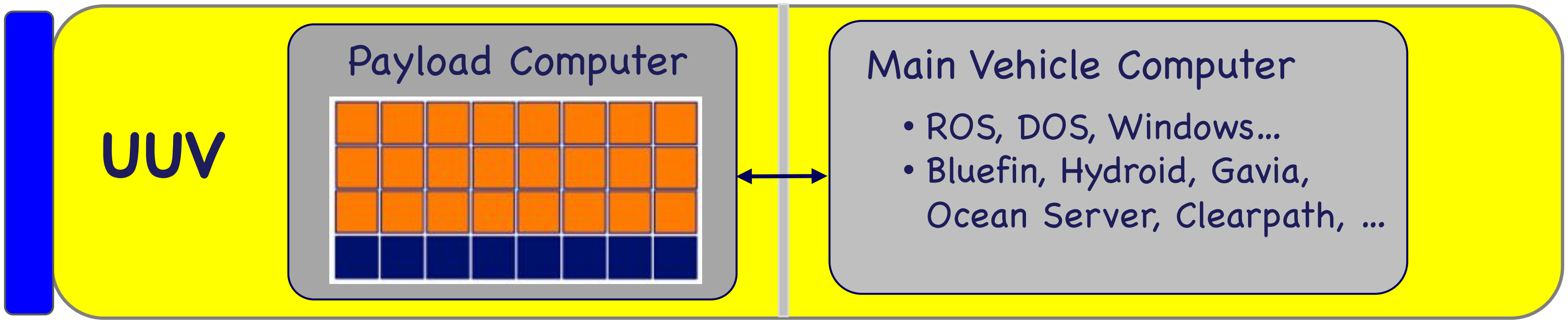
- Both are Open Source
- Both support layering of commercial, proprietary, even classified components built upon the Open Source libraries.

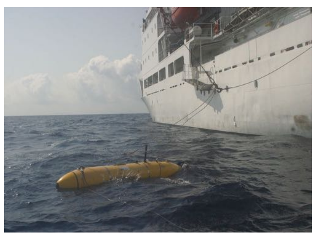
Payload UUV Autonomy
(Architecture Principle #1: Payload Autonomy)



Payload Autonomy

(Architecture Principle #1: Payload Autonomy)




Bluefin-21


AMS Datamaran


Teledyne Gavia AUV


SeaRobotics SCOAP USV


Kingfisher M200


Kingfisher M100


Bluefin-9


MIT/Hover Kayak


REMUS 600


REMUS 100


Ocean Explorer


RMS Scouts


Iver-2


SeaRobotics USV

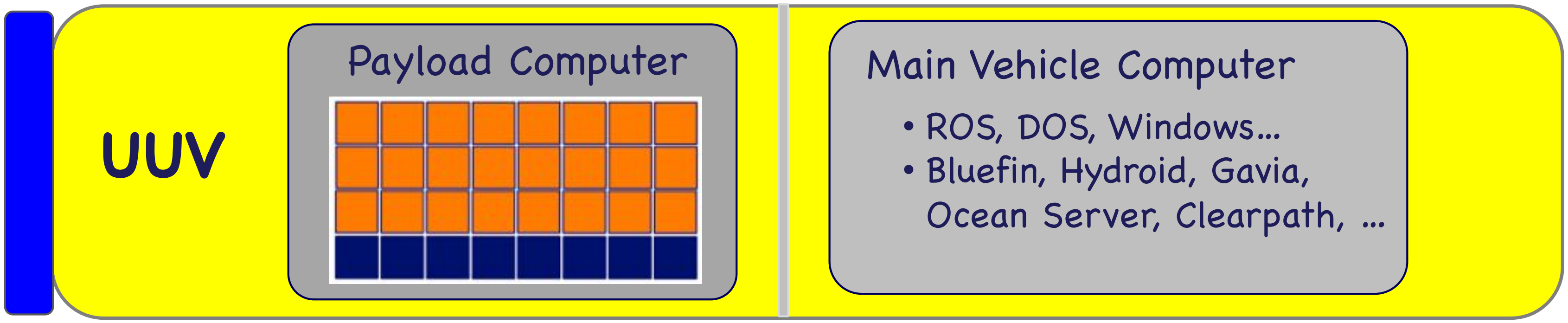

H-Scientific USV

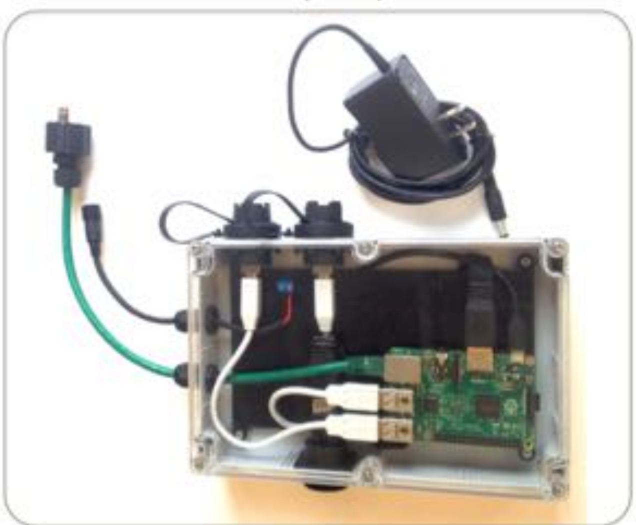

Bobcat tractor


WAM-V

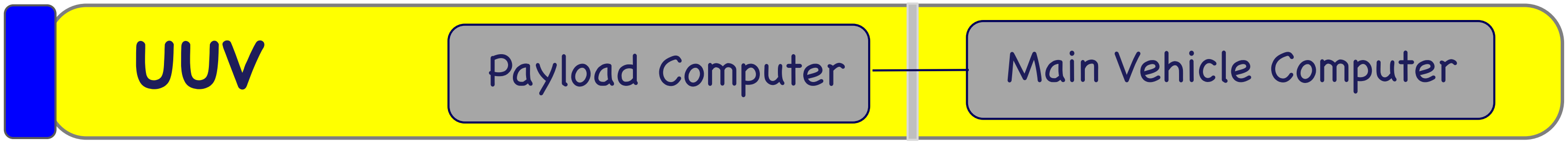

Folaga Glider

Payload Autonomy

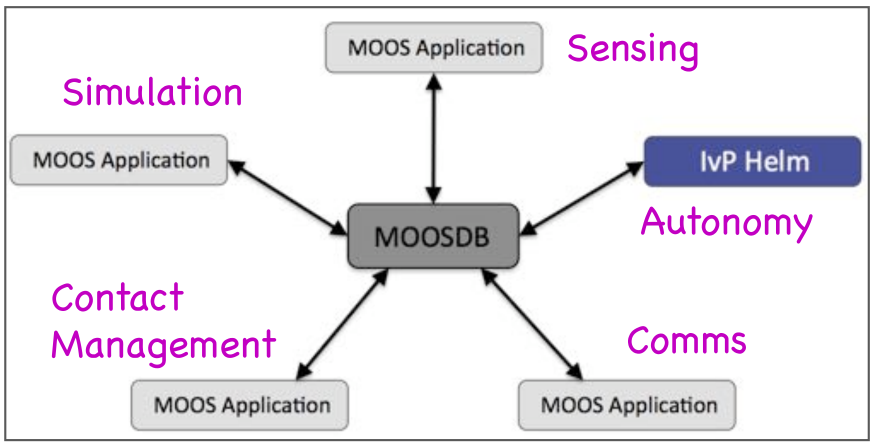




Payload Autonomy
(Architecture #2: Publish-Subscribe Middleware)



Payload Computer

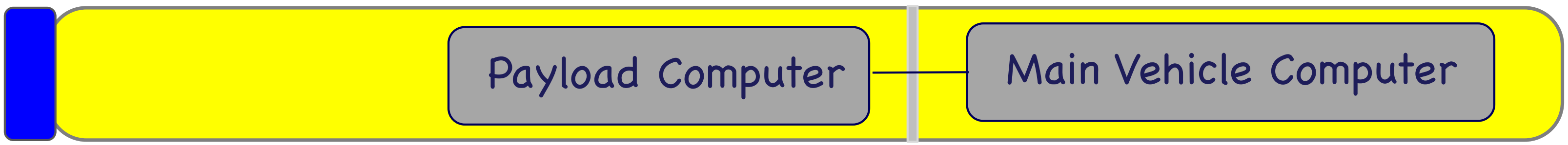


MOOS Middleware
MOOS Applications

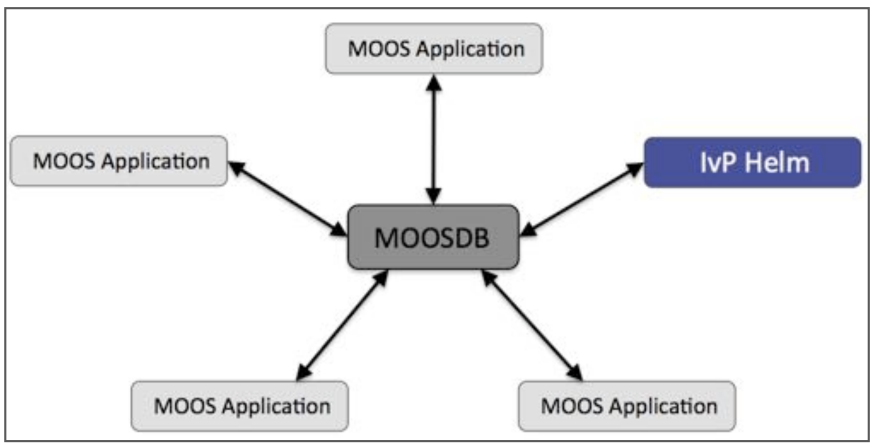
Architecture Principle #2
Autonomy System Middleware

De-couple Software Procurements
Sensing, Autonomy, Simulation, Comms...

Payload Autonomy
(Architecture #3: Behavior Based Decision Making)

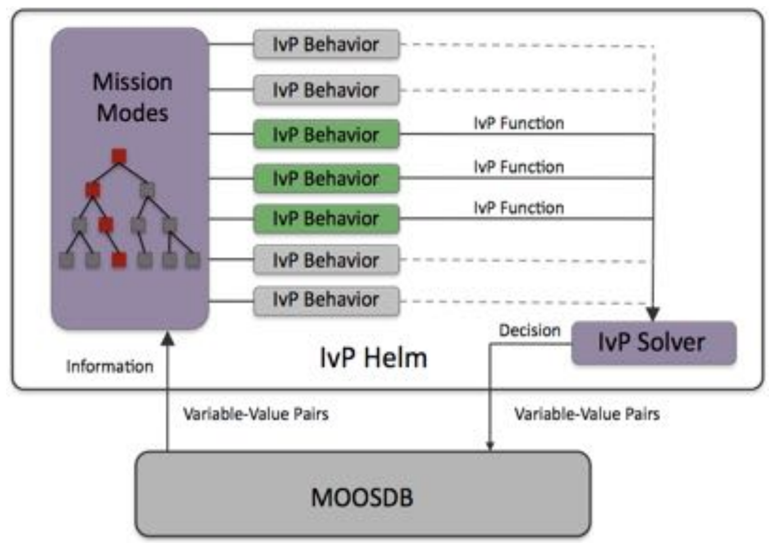


Payload Computer



MOOS Middleware
MOOS Applications

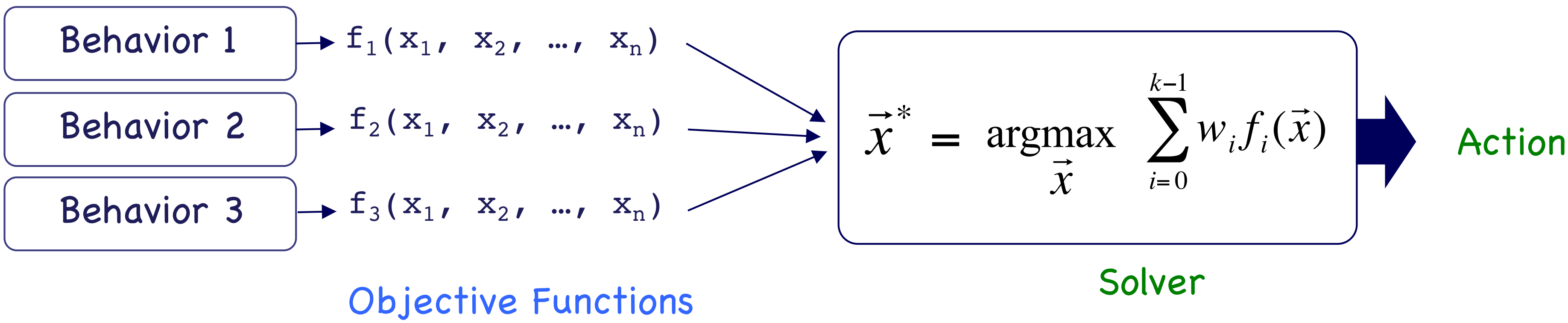
IvP Helm



Behavior-Based
Modular HELM

Overview of the IvP Helm

Behavior Output and Action Selection



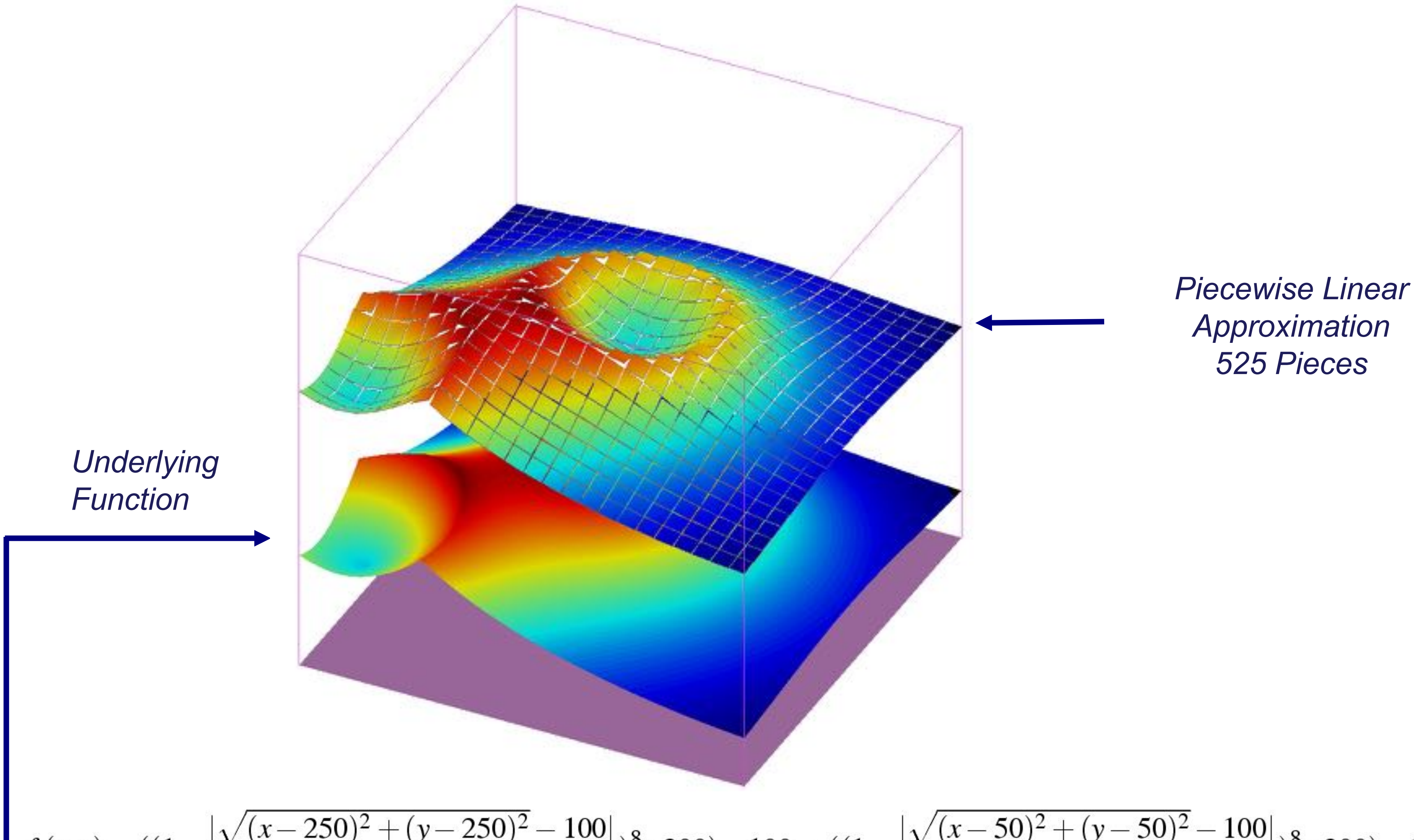
- The objective functions are called IvP functions - functions of a certain format.
- The Solver is called the IvP Solver - they exploit the IvP function structure.

- Typical Decision Space: Heading, Speed, Depth

IvP Functions

The IvP Function vs. Underlying Function

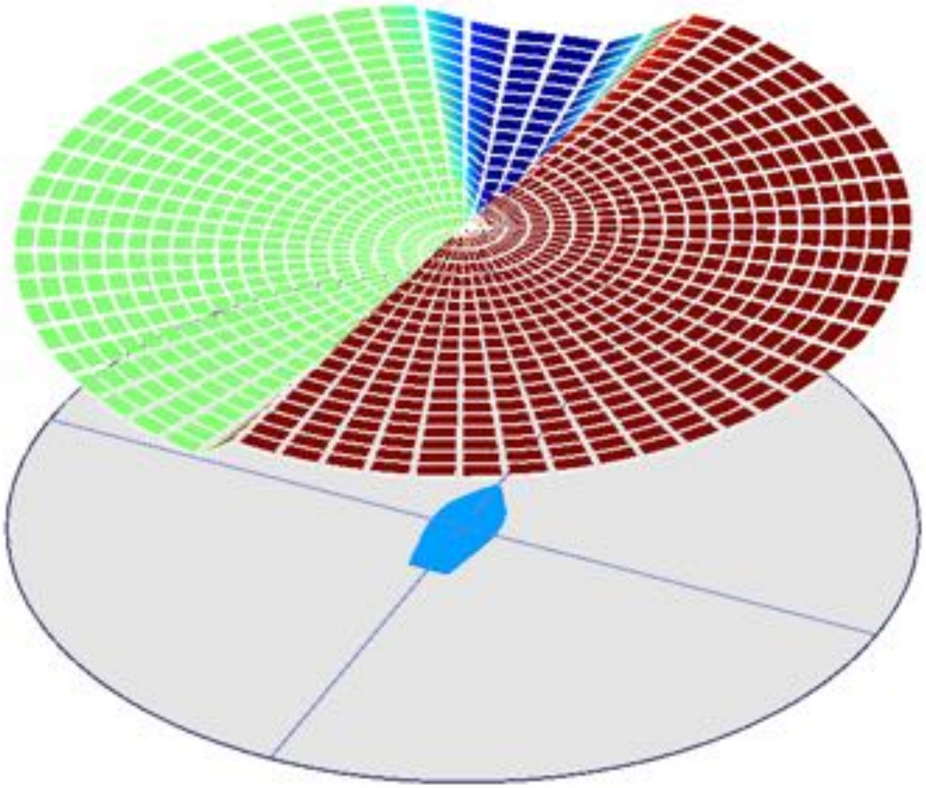
An *IvP function* is a piecewise linear approximation of an objective function, over a discrete decision space (domain).

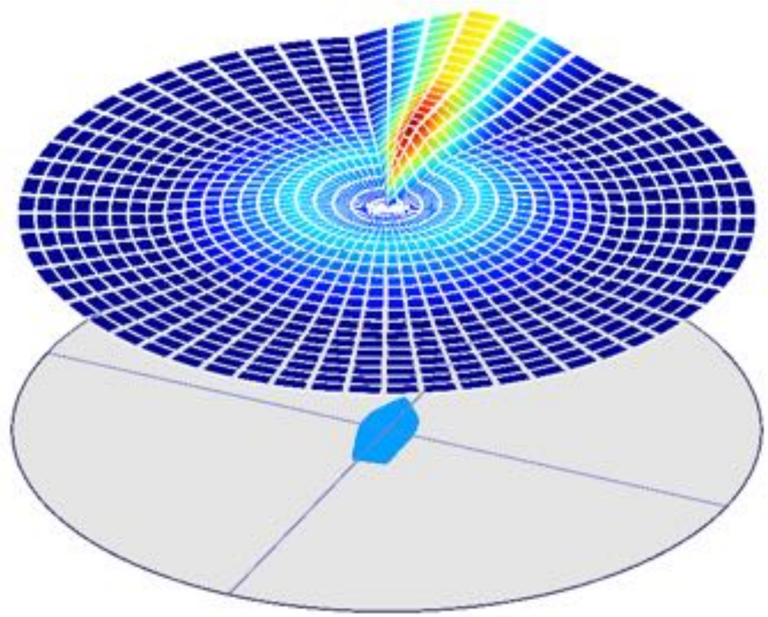


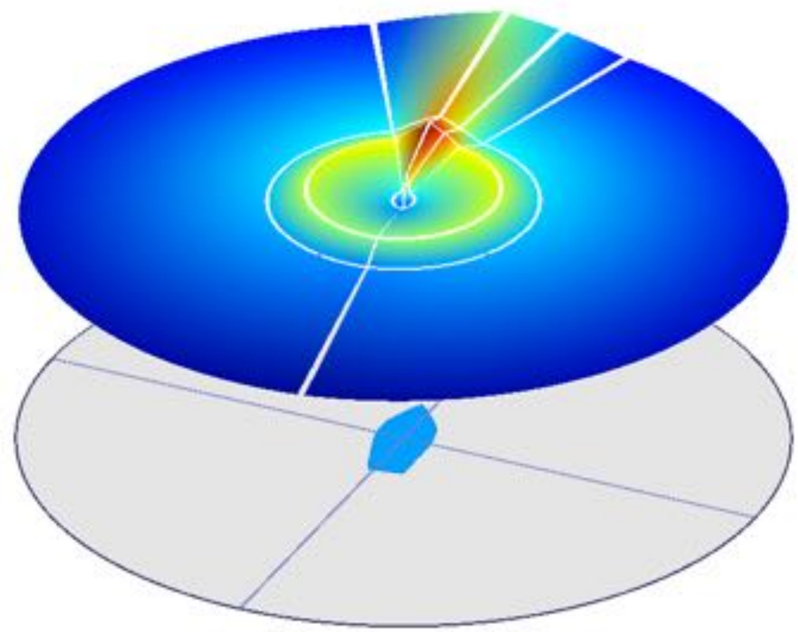
$$f_i(x,y) = ((1 - \frac{|\sqrt{(x-250)^2 + (y-250)^2} - 100|}{2500})^8 * 200) - 100, +((1 - \frac{|\sqrt{(x-50)^2 + (y-50)^2} - 100|}{2500})^8 * 200) - 100$$

Overview of the IvP Helm

Example IvP Functions for Collision Avoidance







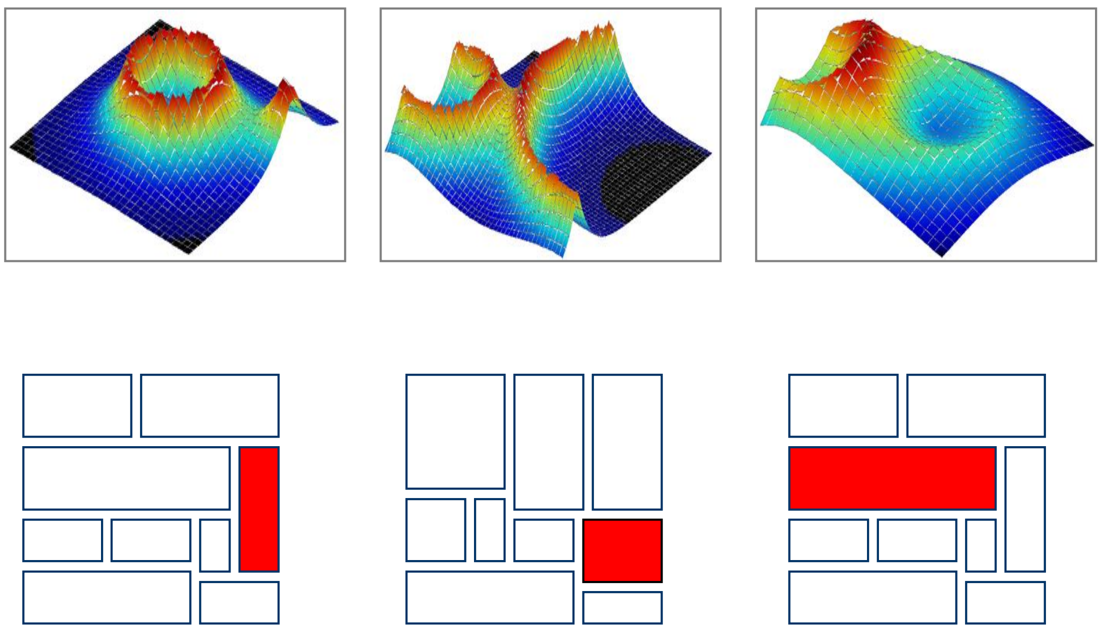
Interval Programming Solution Algorithms

Overview

An IvP problem consists of a set of k functions, each with a priority weighting. The solution is given by:

$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmax}} \sum_{i=0}^{k-1} w_i f_i(\vec{x})$$

f_1 f_2 f_3

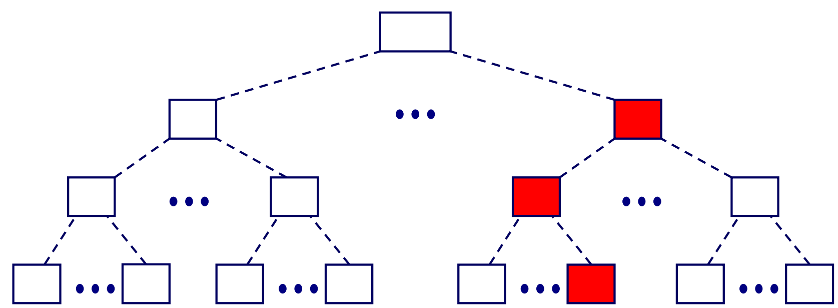


The Search Tree:

- 1 level for each function
- n^k leaf nodes (n pieces per function).

The Solution algorithm:

- Branch and bound
- Pruning based on intersection look-ahead.



Overview of the IvP Helm

Behavior Output and Action Selection

Behavior 1 → $f_1(x_1, x_2, \ldots, x_n)$

Behavior 2 → $f_2(x_1, x_2, \ldots, x_n)$

Behavior 3 → $f_3(x_1, x_2, \ldots, x_n)$

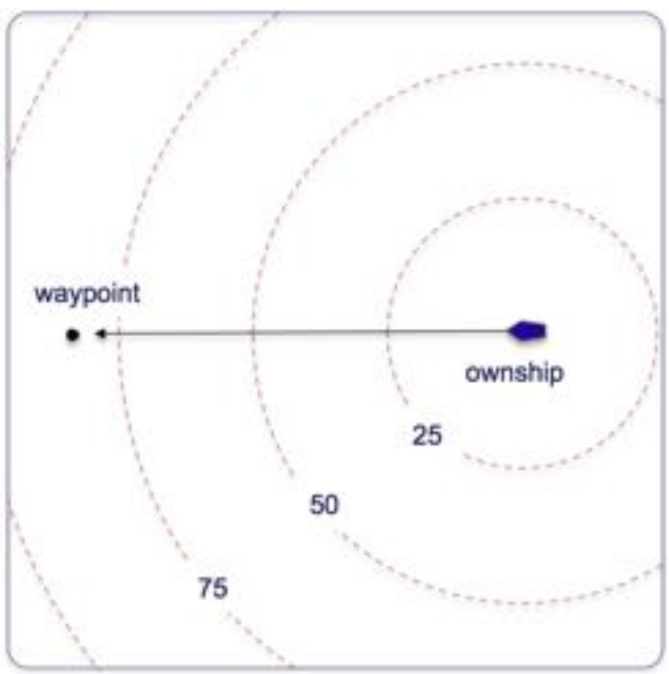
Objective Functions

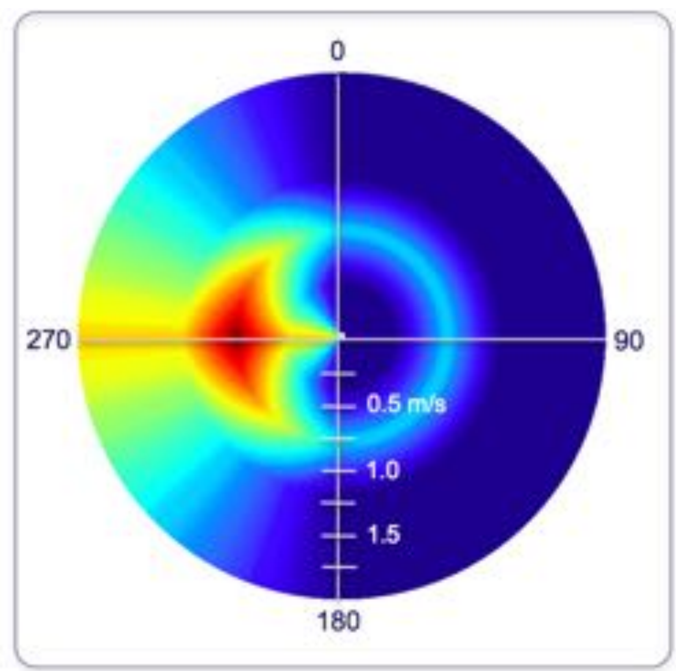
$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmax}} \sum_{i=0}^{k-1} w_i f_i(\vec{x})$$

Solver

Action

Example:
Waypoint Traversal
Behavior





Overview of the IvP Helm

Behavior Output and Action Selection

Behavior 1 → $f_1(x_1, x_2, \ldots, x_n)$

Behavior 2 → $f_2(x_1, x_2, \ldots, x_n)$

Behavior 3 → $f_3(x_1, x_2, \ldots, x_n)$

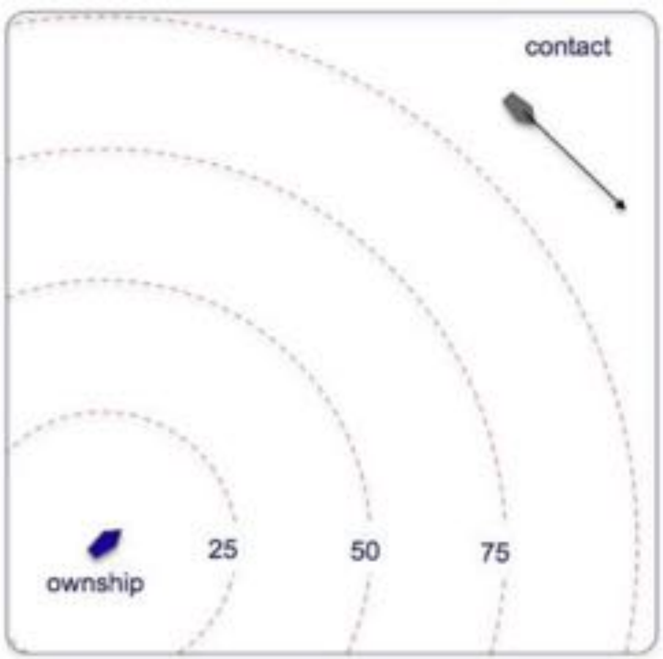
Objective Functions

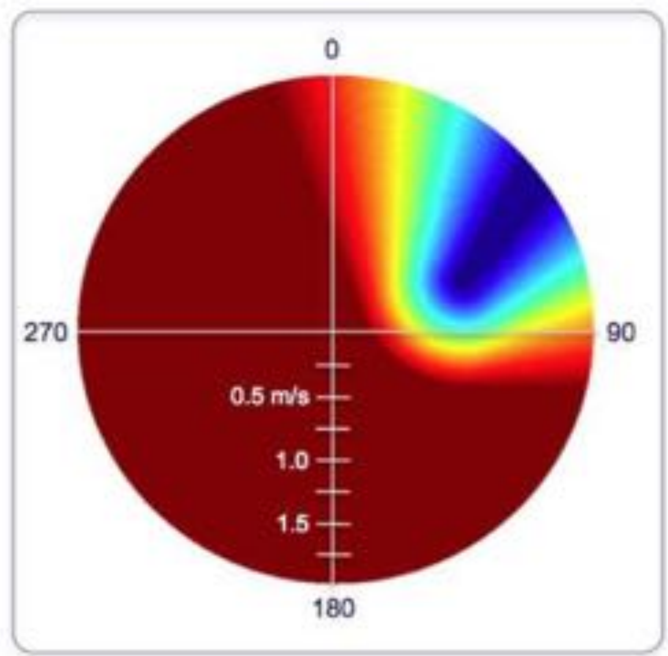
$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmax}} \sum_{i=0}^{k-1} w_i f_i(\vec{x})$$

Solver

→ Action

Example:
Collision Avoidance
Behavior

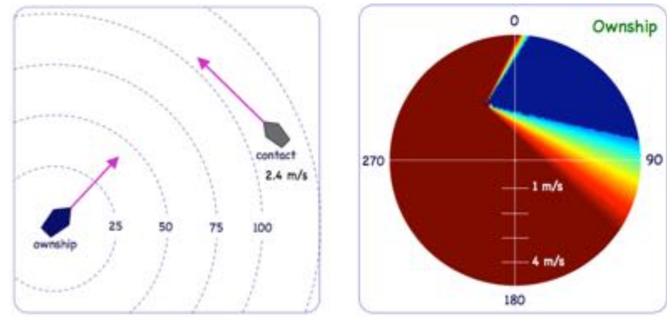




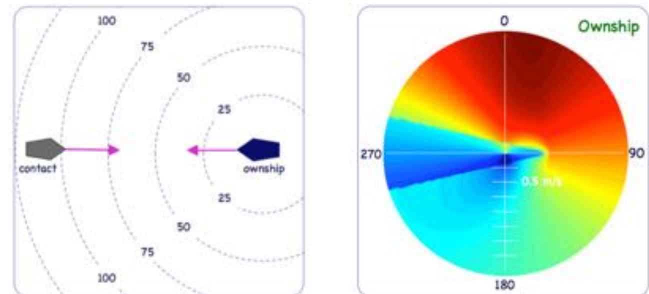
COLREGS Autonomy

- Funded by: Office of Naval Research (ONR)
- Idea:
  - Enable autonomous surface vehicles to obey the “Rules of the Road” COLREGS.
  - Establish a road test for validating the autonomous collision avoidance.
- Research Focus:
  - Map the protocols written for humans into algorithmic format.
  - Find minimal set of field tests that validate widest set of scenario permutations.

Crossing Situation



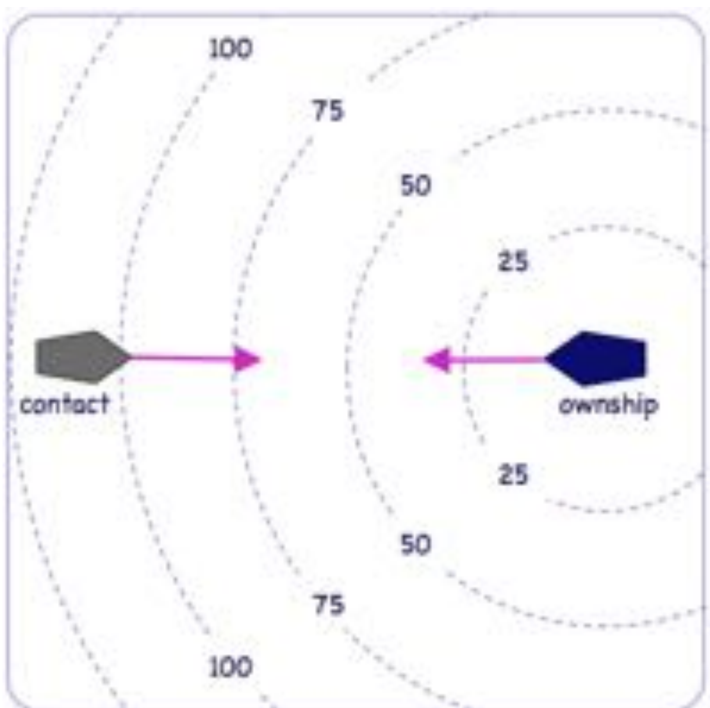
Head-on Situation

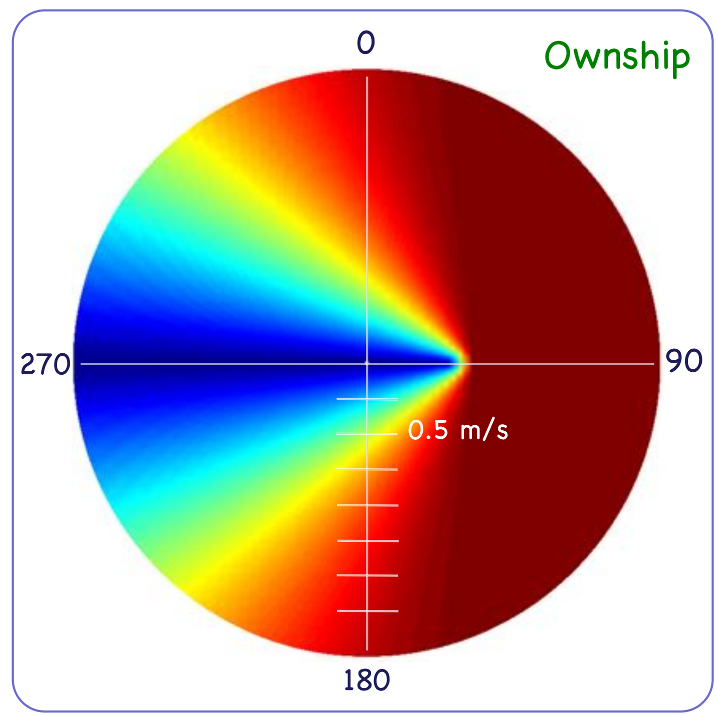


- Technical Approach:
  - Collision avoidance protocols mapped to set of modes, and submodes.
  - Modes map to a unique form of objective function. Multi-objective optimization with IvP to solve.
- Impact:
  - Autonomous long-duration coastal sampling with autonomous platforms.
  - Multi-vehicle/swarm capabilities can be built on COLREGS foundation.

What's Wrong with Non Protocol Based Collision Avoidance

- Consider the head-on situation
- If ownship rates candidate maneuvers based on closest point of approach, a maneuver to port or starboard looks equally good.



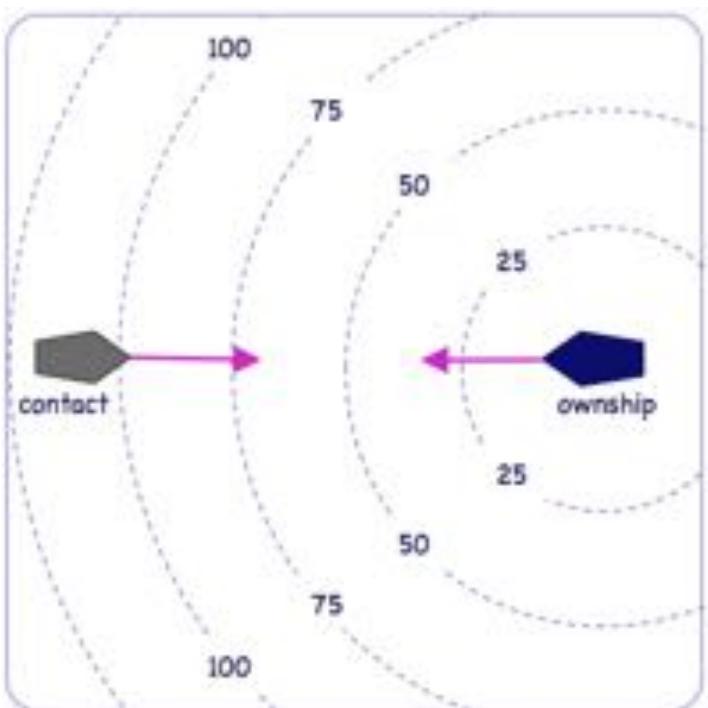


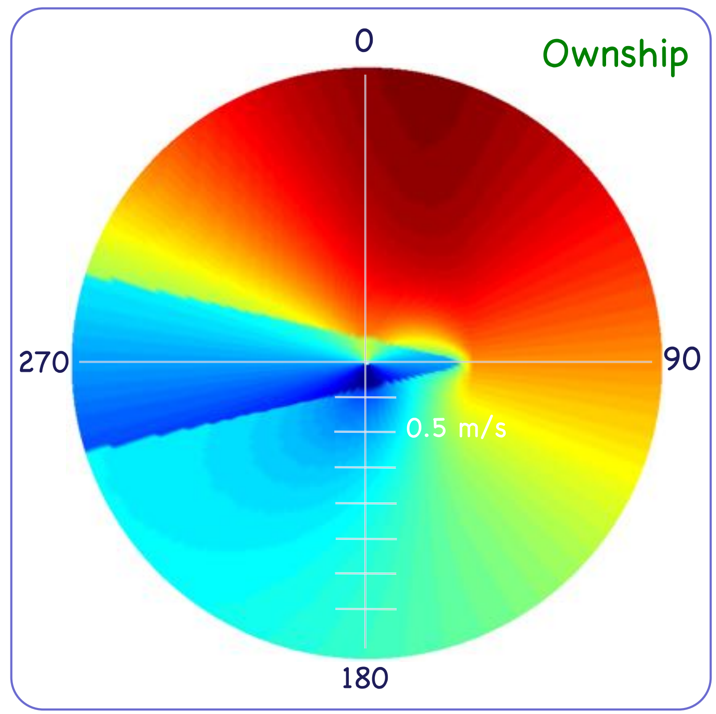
COLREGS Collision Avoidance

- The head-on situation is referenced in Rule 14 of the COLREGS.

> When two power-driven vessels are meeting on a reciprocal or nearly reciprocal courses so as to involve a risk of collision each shall alter her course to the starboard so that each shall pass on the port side of the other

- The COLREGS IvP Behavior on ownship heavily penalizes the "wrong" kind of turn.



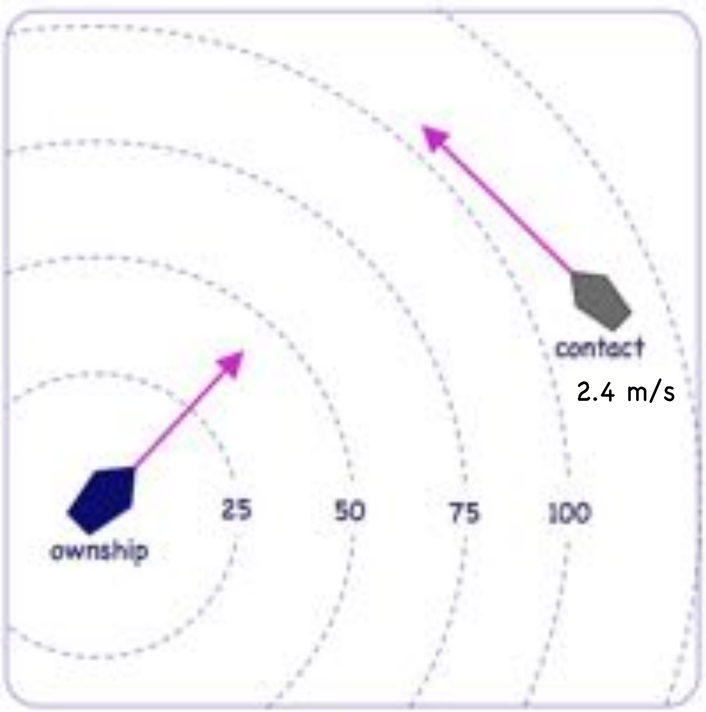


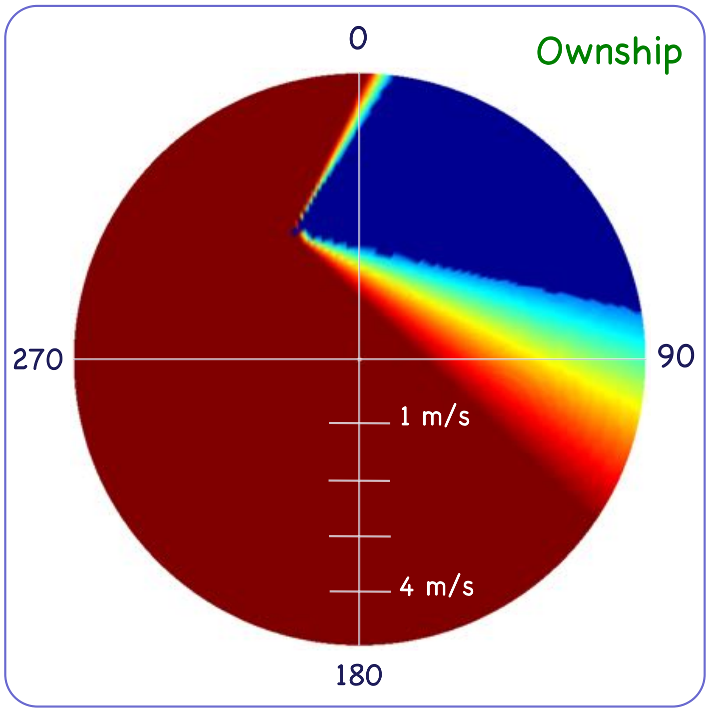
COLREGS Collision Avoidance

- The give-way (crossing) situation is referenced in Rule 15 of the COLREGS.

> When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

- The give-way vehicle may cross ahead of the other vessel if it clearly makes more sense.





COLREGS Collision Avoidance

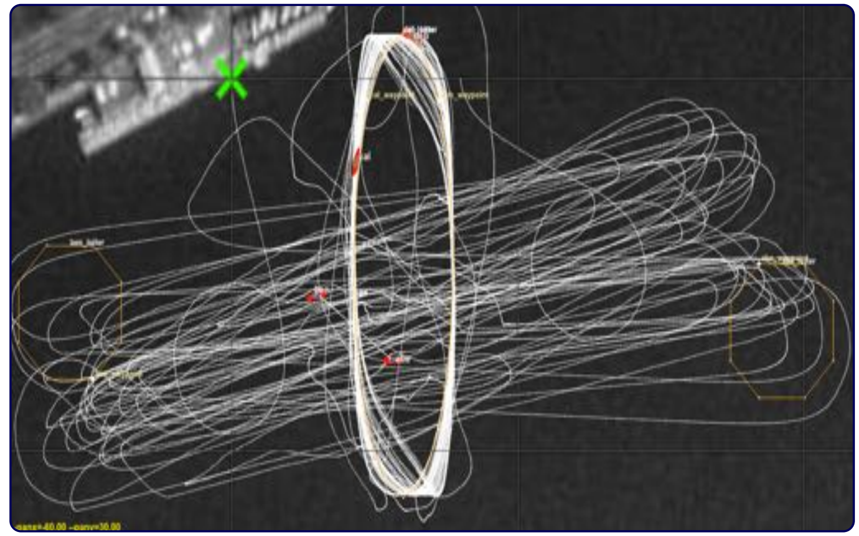
(sponsored by ONR)

- The objective is to avoid collisions with other autonomous and non-autonomous vehicles.
- COLREGS are the rules of the road for seagoing vessels.
- They provide a protocol of roles and required actions between vessels.

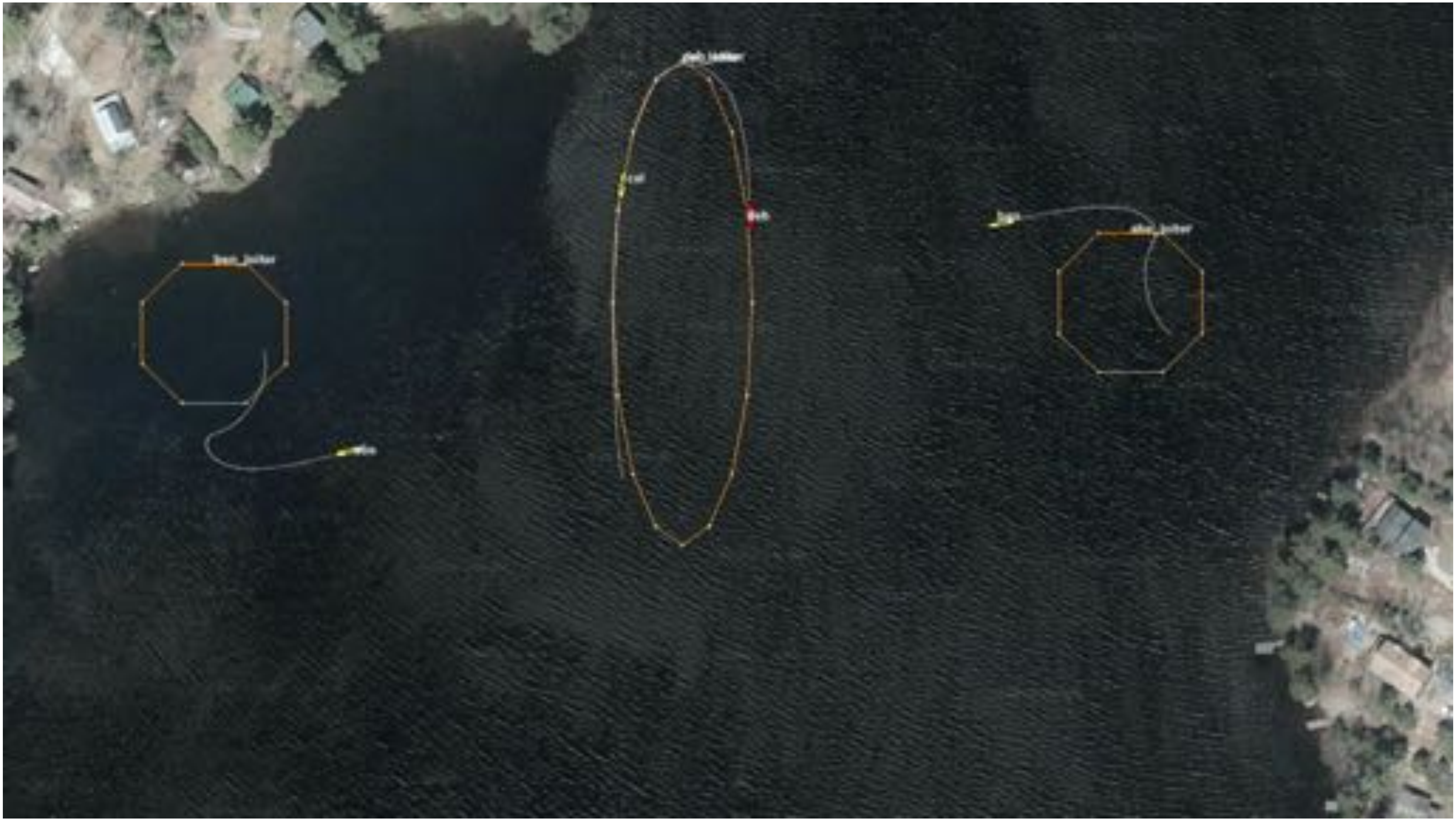
Collision Avoidance
WITH COLREGS

COLREGS
Testing / Validation

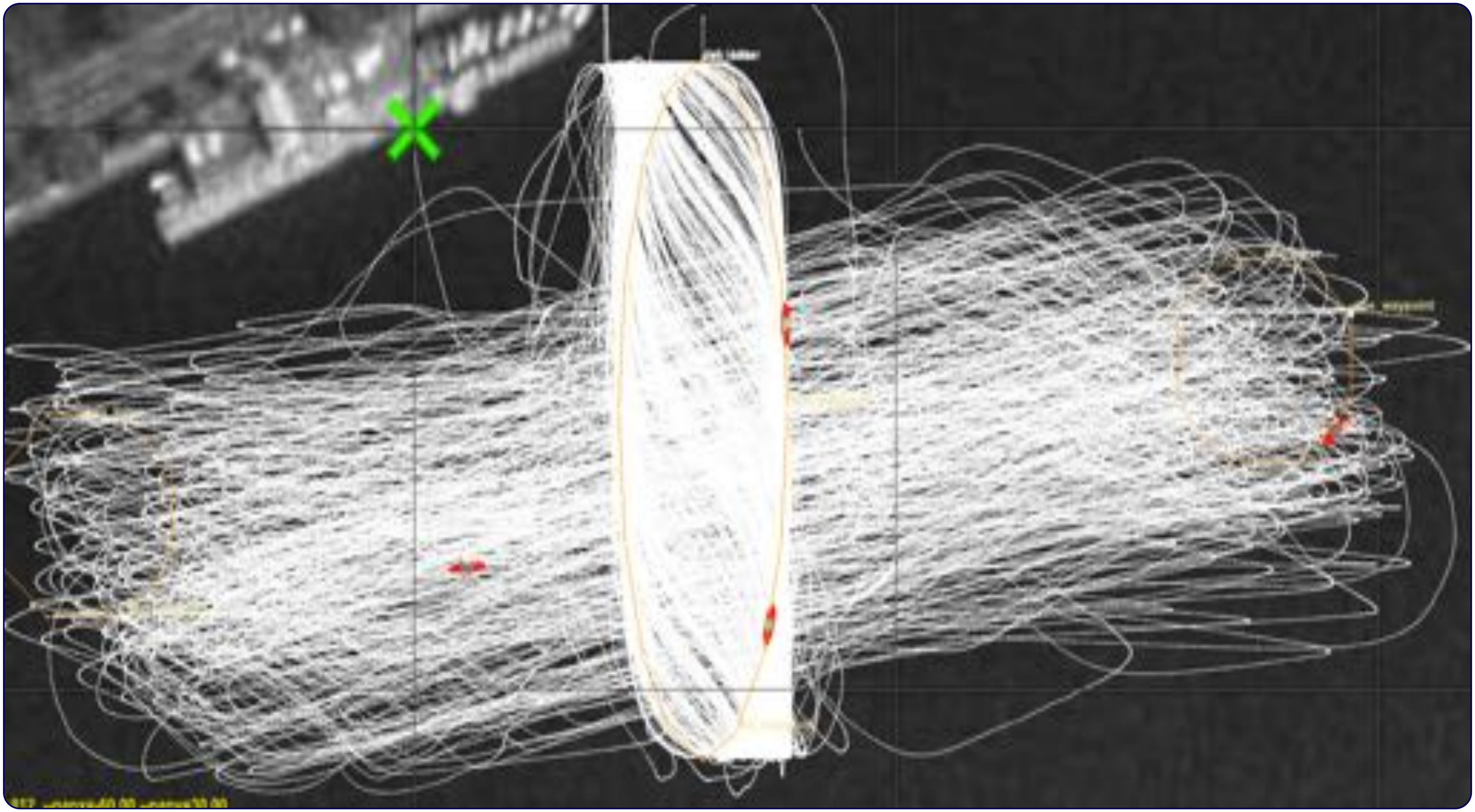
- Simulation
- In-Water Tests



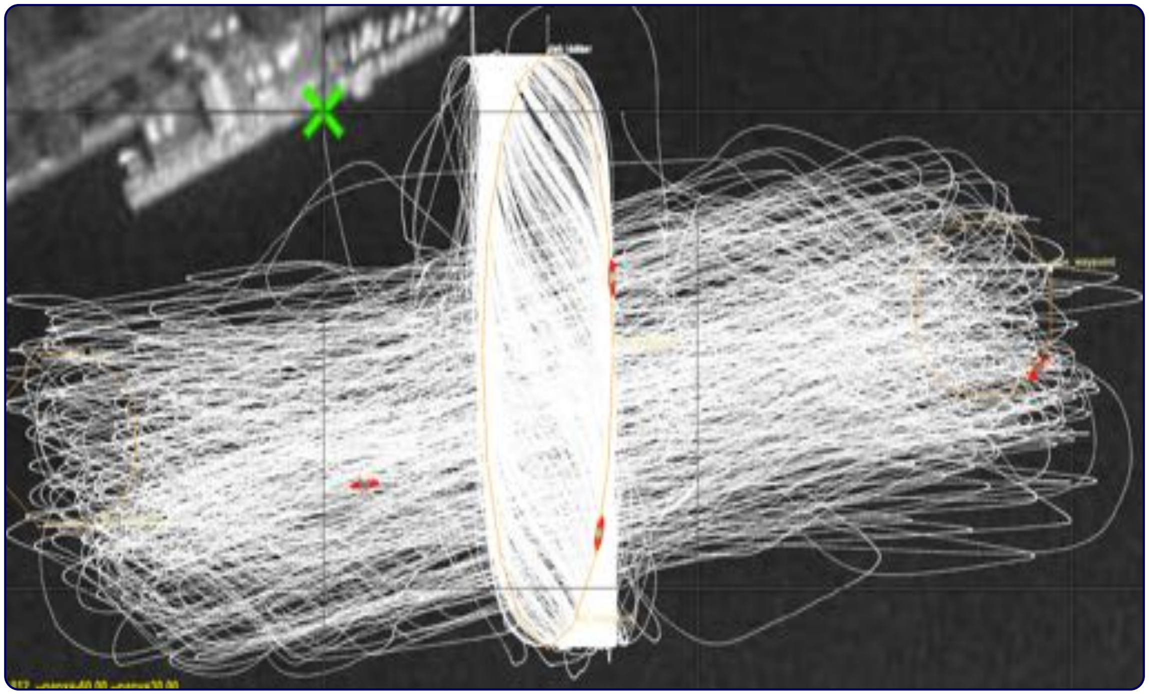
COLREGS
Testing / Validation



Testing / Validation



How do we learn from the data?



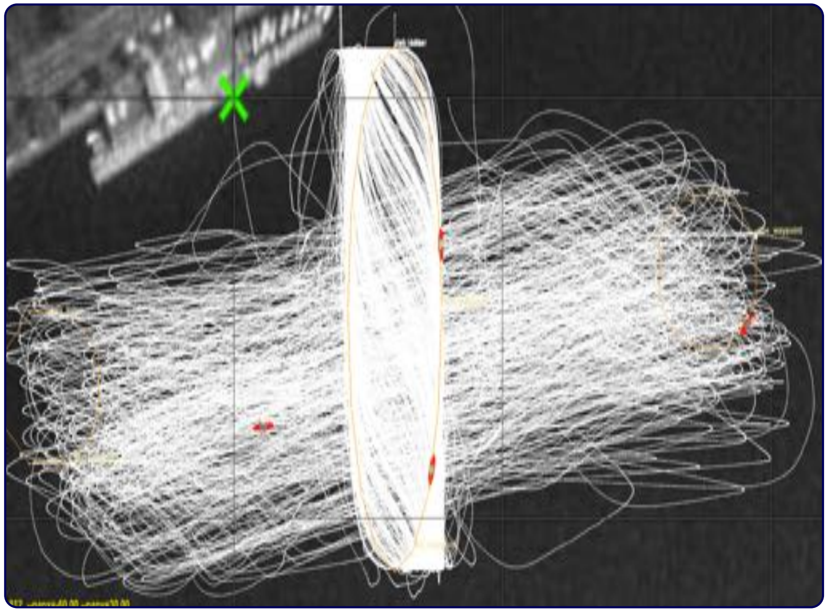
The Challenge:

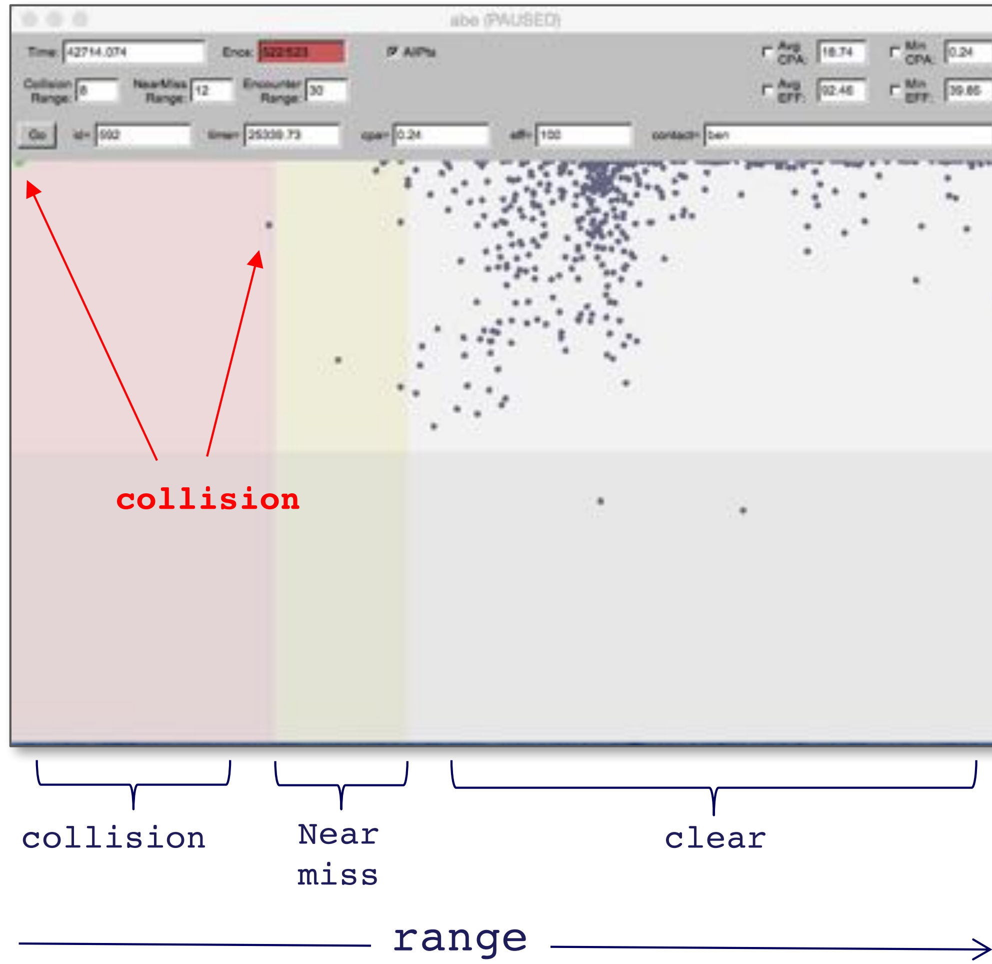
- 12 hours of simulation
- 4 vehicles
- 1009 encounters

RESULTS

- 1009 encounters
- 2 collisions
  - -one 0.95m
  - -one 7.5m
- 9 near misses (8-12m range)

Auto Generated Encounter Plots





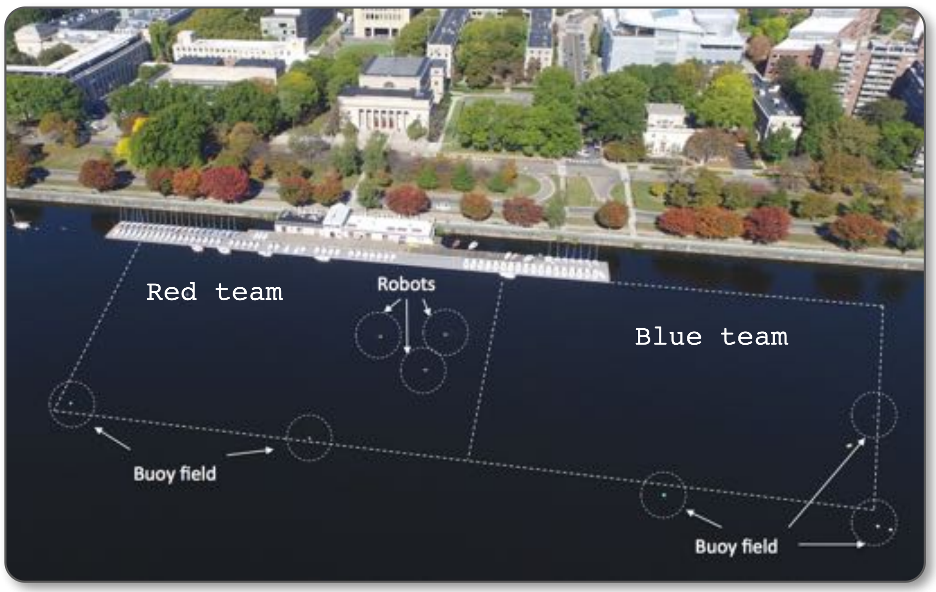
Encounter Plots

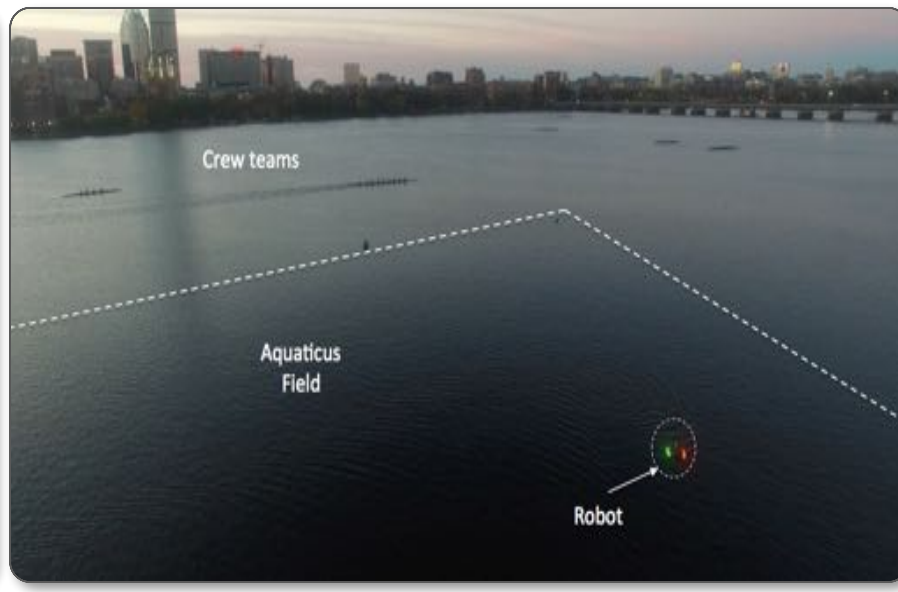
- Part of the alogview tool
- Open Source
- Works on any mission log file
- Developed Nov '15 to present.
- In next release of MOOS-IvP

Aquaticus.

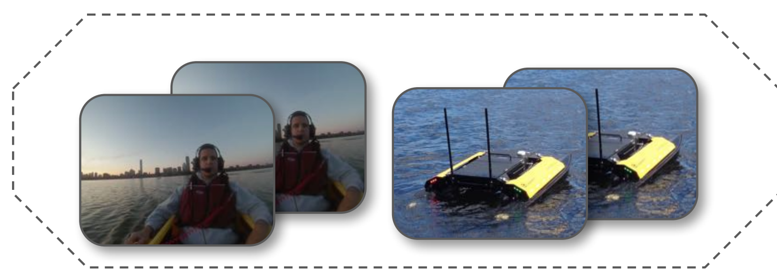
(funded by DARPA TTO)

- Aquaticus is a human-robot competition developed at MIT on the Charles River.
- It pits teams of humans and robots against other teams of humans and robots.
- It explores advanced marine *autonomy*, *human-robot trust*, *operator load* and the interface between robots and humans embedded in the field with robots.





VS.

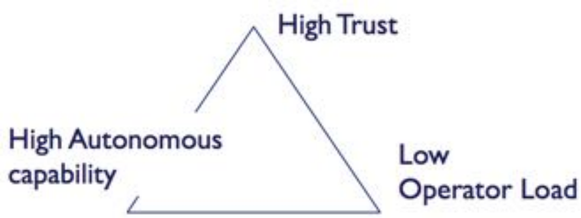


Aquaticus

Hypotheses:

The most effective Human-Robot systems are where the robot *augments* the human.

An effective robot is one that has high *autonomous capability*, high *operator trust*, and low *operator load*.



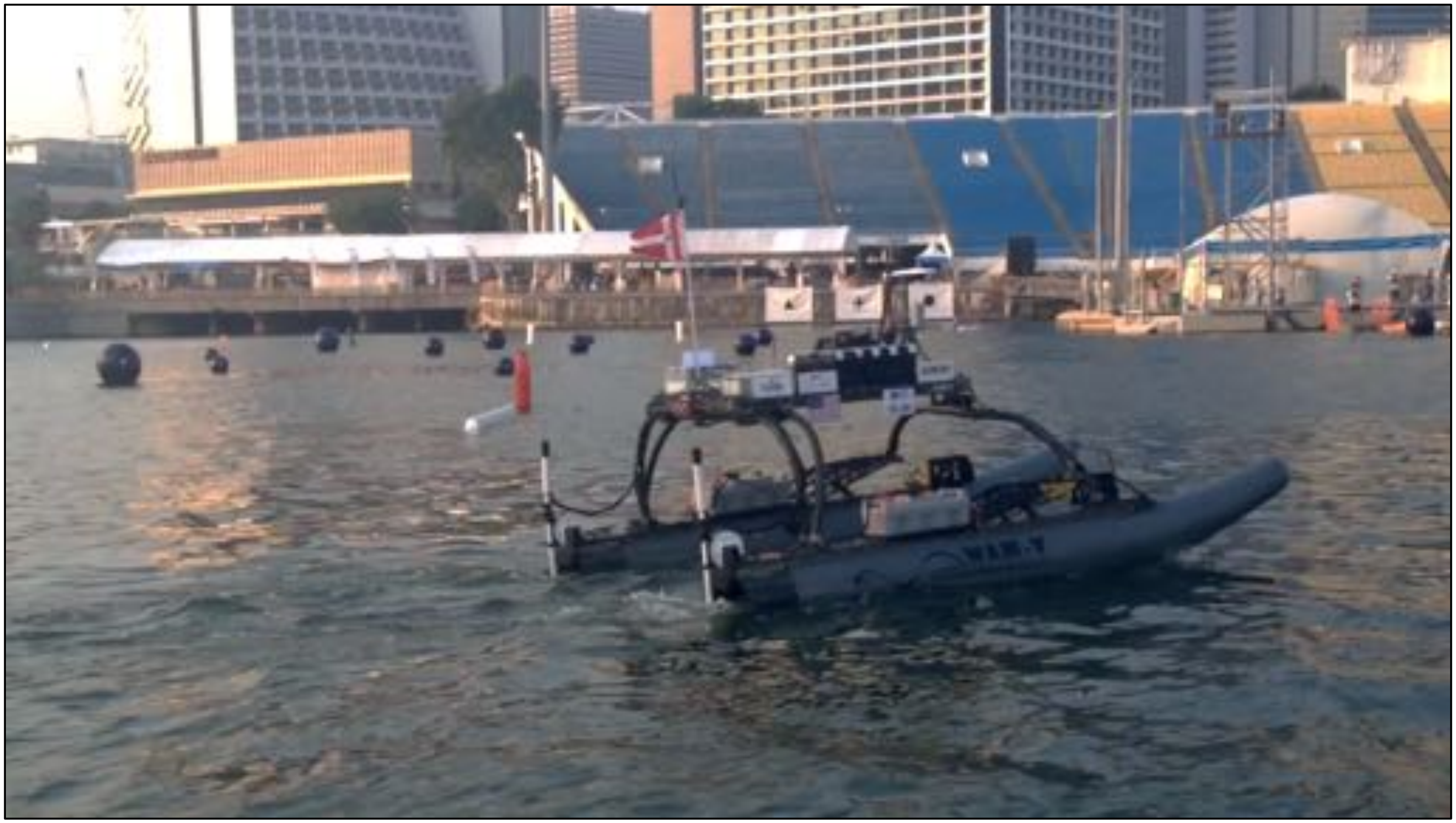
This is a tradeoff space – the right mix is not immediately obvious for a given application or set of humans.

Part of Aquaticus is to discover the basic relationships – to find that mix for *any* human-robot application.

Aquaticus

Boston Harbor Robo-Challenge

Remote Ocean Sensing Launched from MIT
Currently a seed project – Initial funding by Lockheed.
Looking for additional sponsors!



Boston – Harbor Robo-Challenge

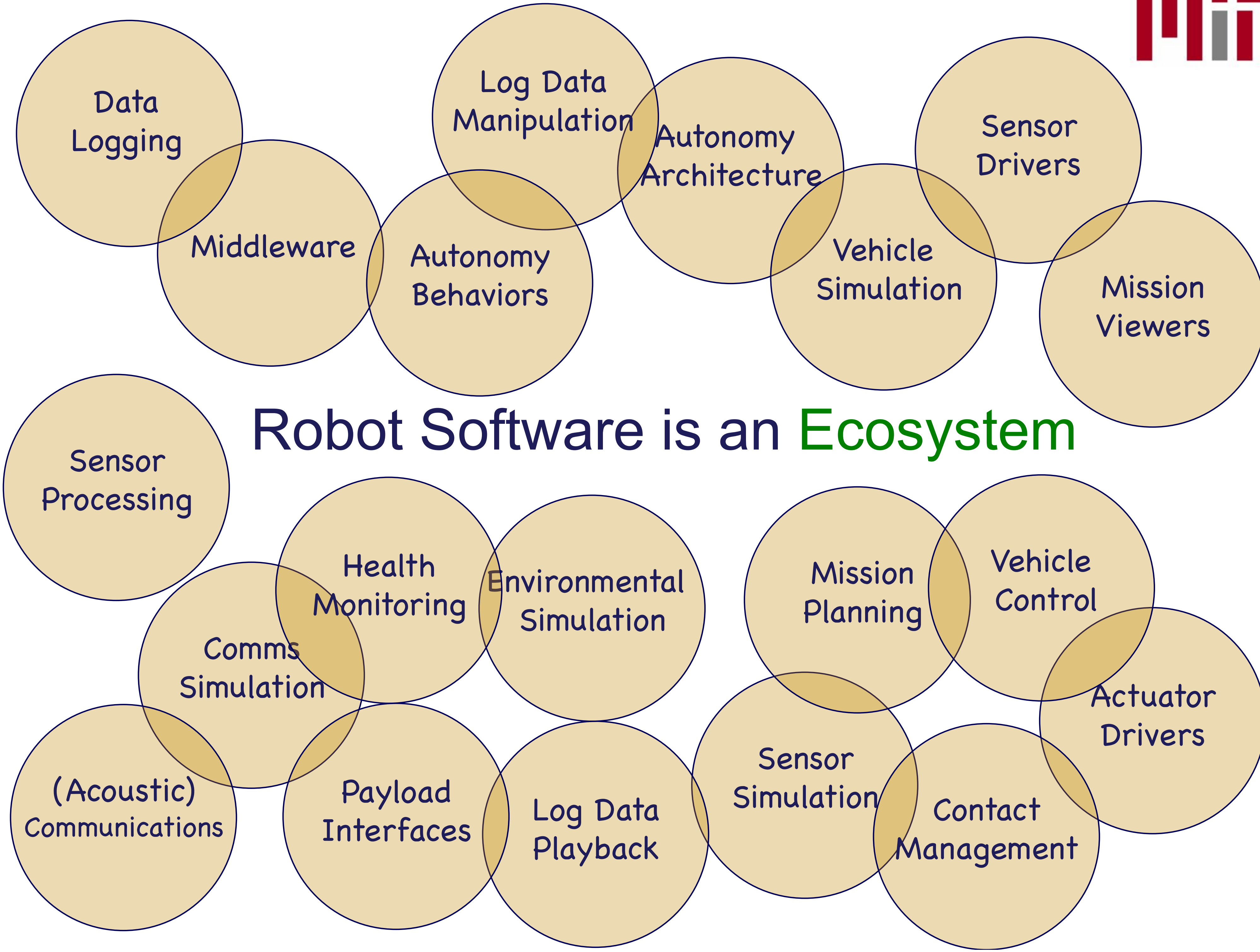
Overview of the route from MIT, through the Charles River, Boston Harbor and finally in Mass Bay. Roughly 5 kilometers.

Boston Harbor Robo-Challenge

Remote Ocean Sensing Launched from MIT
Currently a seed project - Initial funding by Lockheed.
Looking for additional sponsors!

Robot Software is an Ecosystem

Robot Software is an Ecosystem



The MOOS-IvP Ecosystem

- 800+ pages of text
- Many Working examples

- Agreement w/ Cambridge Univ. Press

- 35 workyears
- 120,000 lines of code

- 20 vehicle types
- 40+ projects
- 9+ countries

Example Missions

Documentation

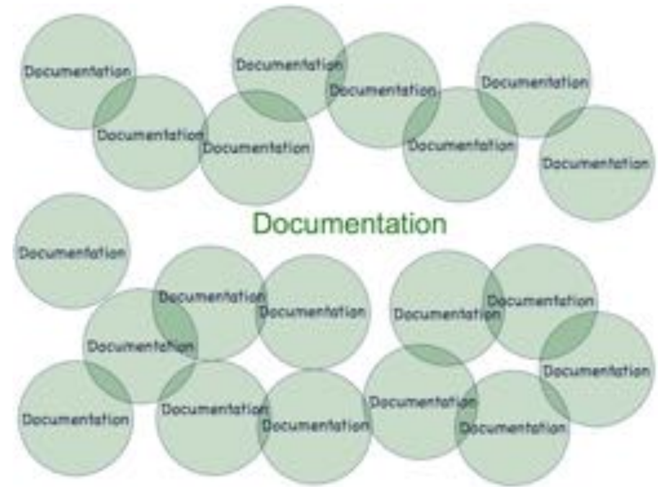
Tools
- Device/Sensor interfaces
- Vehicle Simulation
- Environmental Simulation
- Comms Simulation
- Debugging Tools
- Mission Planning Tools
- Mission Operation Tools

Autonomy Behaviors

Math and Utility Libraries

Autonomy Architecture

Robot Middleware



Evolution of the Ecosystem, Education and Research

2000 | 2005 | 2010 | 2017

- MOOS written
- IvP Integrated with MOOS
- Payload autonomy introduced
- First MOOS-DAWG Users workshop
- First offering of MIT 2.680

- Build 100% of the ecosystem yourself

- Faculty
- Post-docs

- Build 40% of ecosystem
- Learn from apprenticing

- Faculty
- Post-docs
- PhD students

- Build 20% of ecosystem
- Learn from documentation
- Learn from apprenticing

- Faculty
- Post-docs
- PhD students
- Masters students

- Build 5% of ecosystem
- Learn from coursework
- Learn from documentation
- Learn from apprenticing

- Faculty
- Post-docs
- PhD students
- Masters students
- Undergrads

END