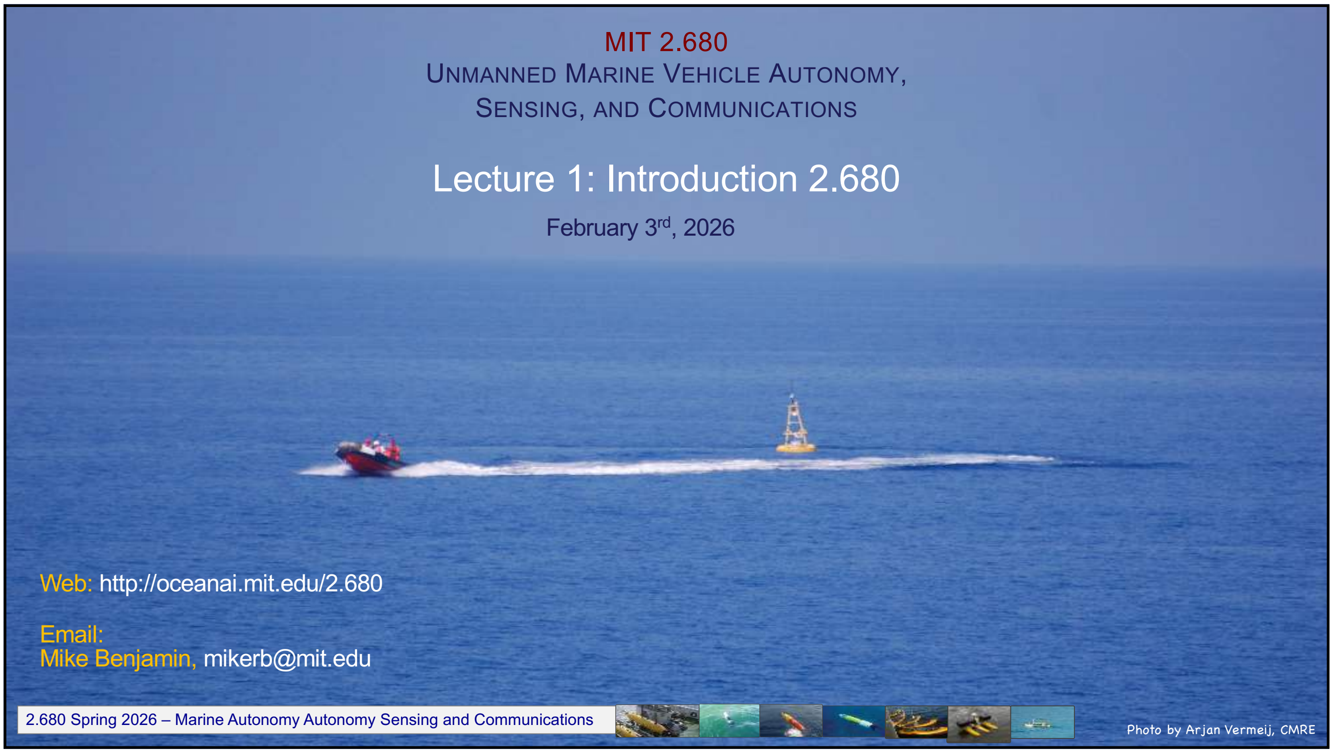


MIT 2.680

UNMANNED MARINE VEHICLE AUTONOMY,
SENSING, AND COMMUNICATIONS

Lecture 1: Introduction 2.680

February 3rd, 2026

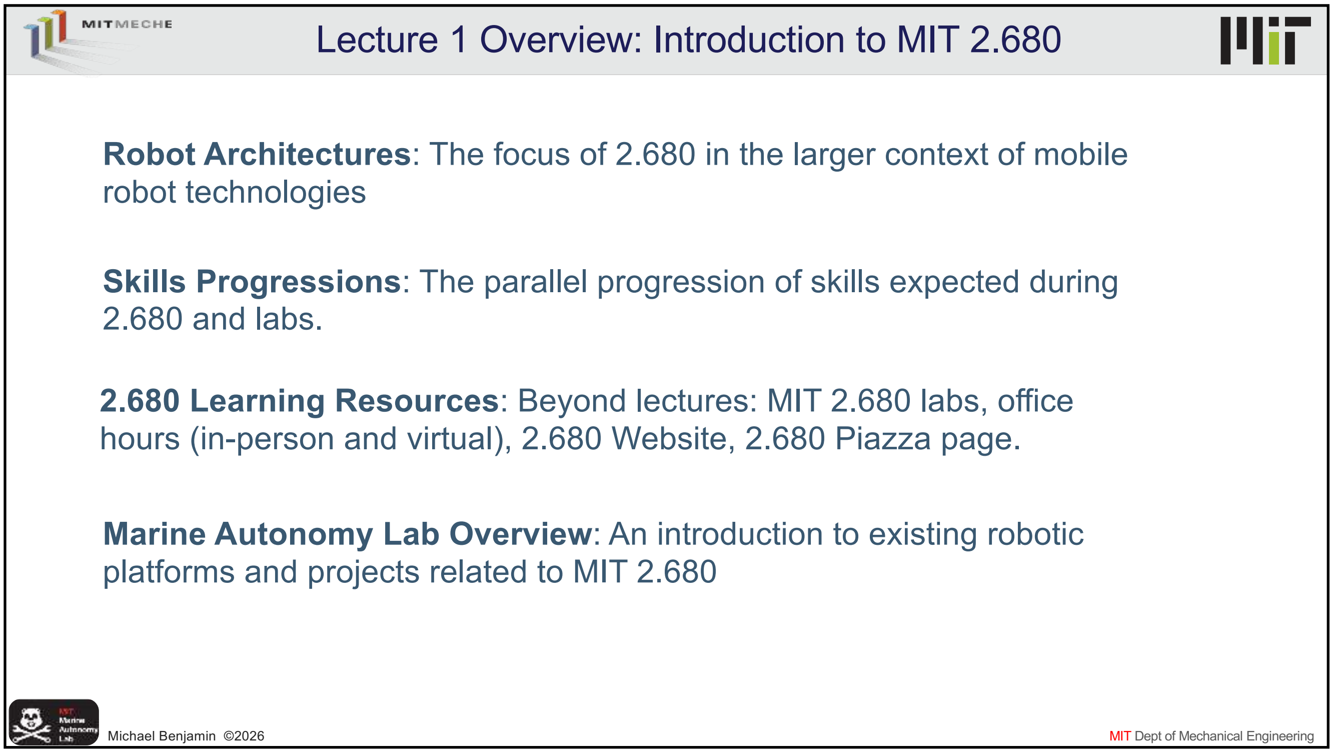
Web: http://oceanai.mit.edu/2.680

Email:
Mike Benjamin, mikerb@mit.edu

2.680 Spring 2026 – Marine Autonomy Autonomy Sensing and Communications

Photo by Arjan Vermeij, CMRE

1



Lecture 1 Overview: Introduction to MIT 2.680

Robot Architectures: The focus of 2.680 in the larger context of mobile robot technologies

Skills Progressions: The parallel progression of skills expected during 2.680 and labs.

2.680 Learning Resources: Beyond lectures: MIT 2.680 labs, office hours (in-person and virtual), 2.680 Website, 2.680 Piazza page.

Marine Autonomy Lab Overview: An introduction to existing robotic platforms and projects related to MIT 2.680

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

2

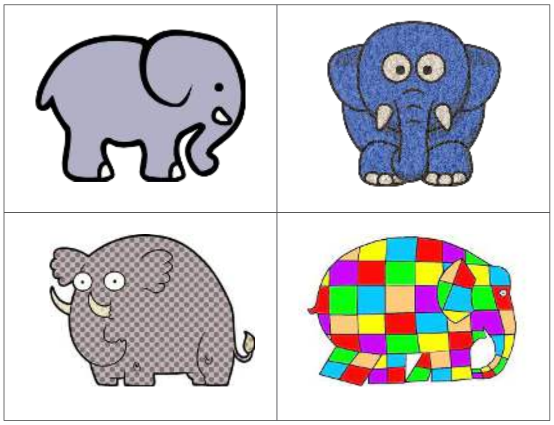
Robot Architectures

3

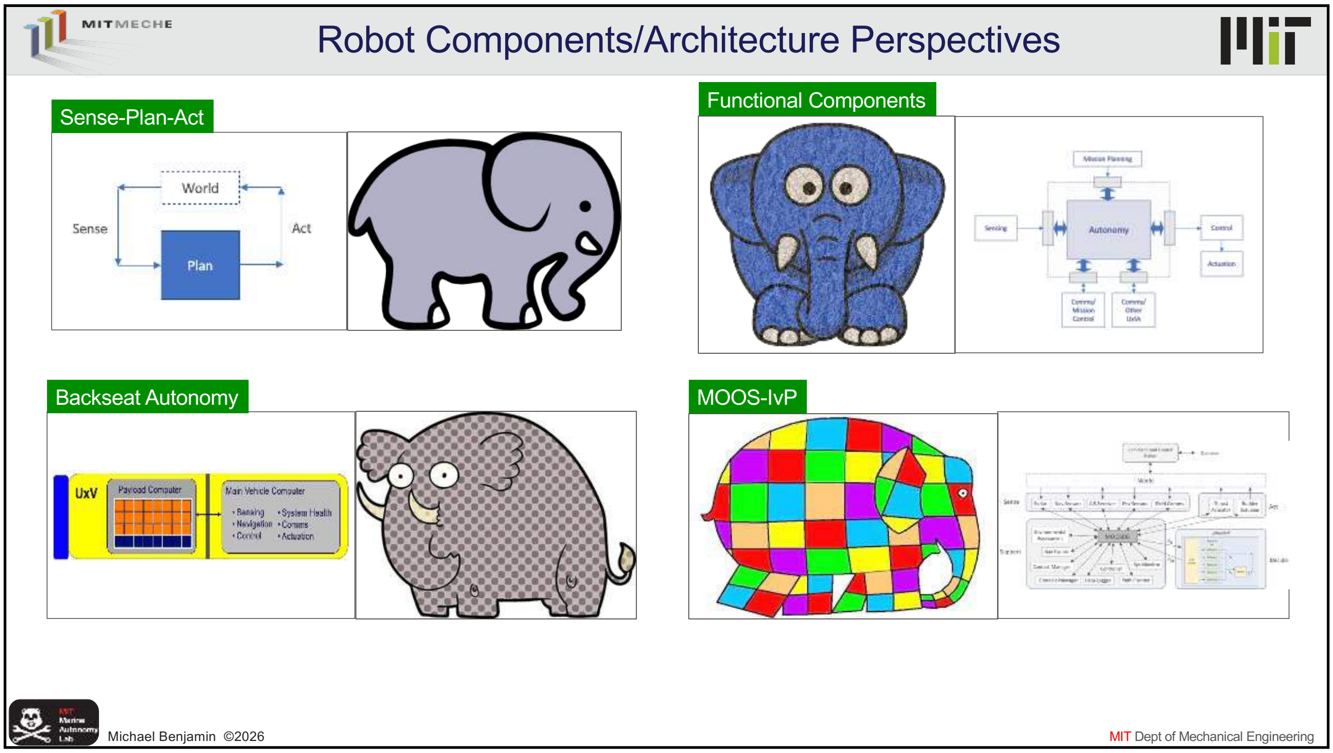
Robot Components/Architecture Perspectives

Reality

Multiple Perspectives



4



MITMECHE

Robot Components/Architecture Perspectives

Sense-Plan-Act

Functional Components

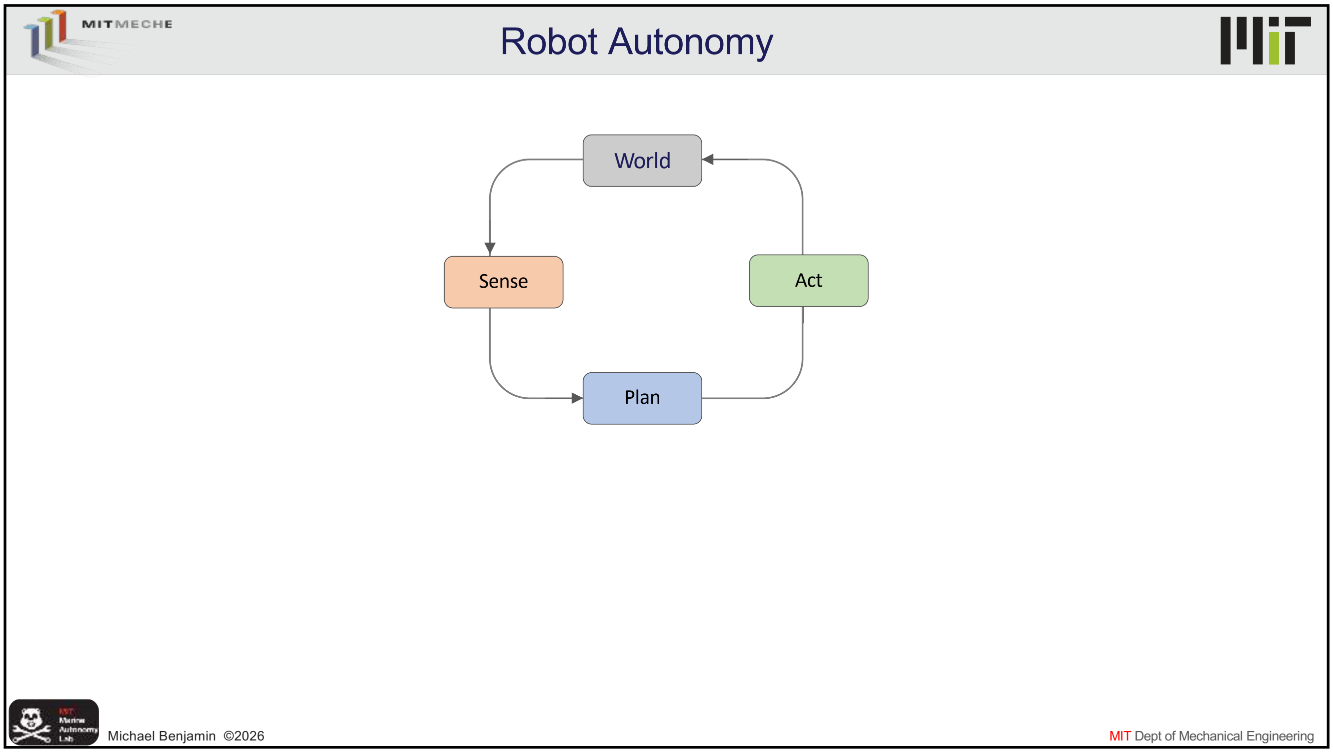
Backseat Autonomy

MOOS-IvP

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

5



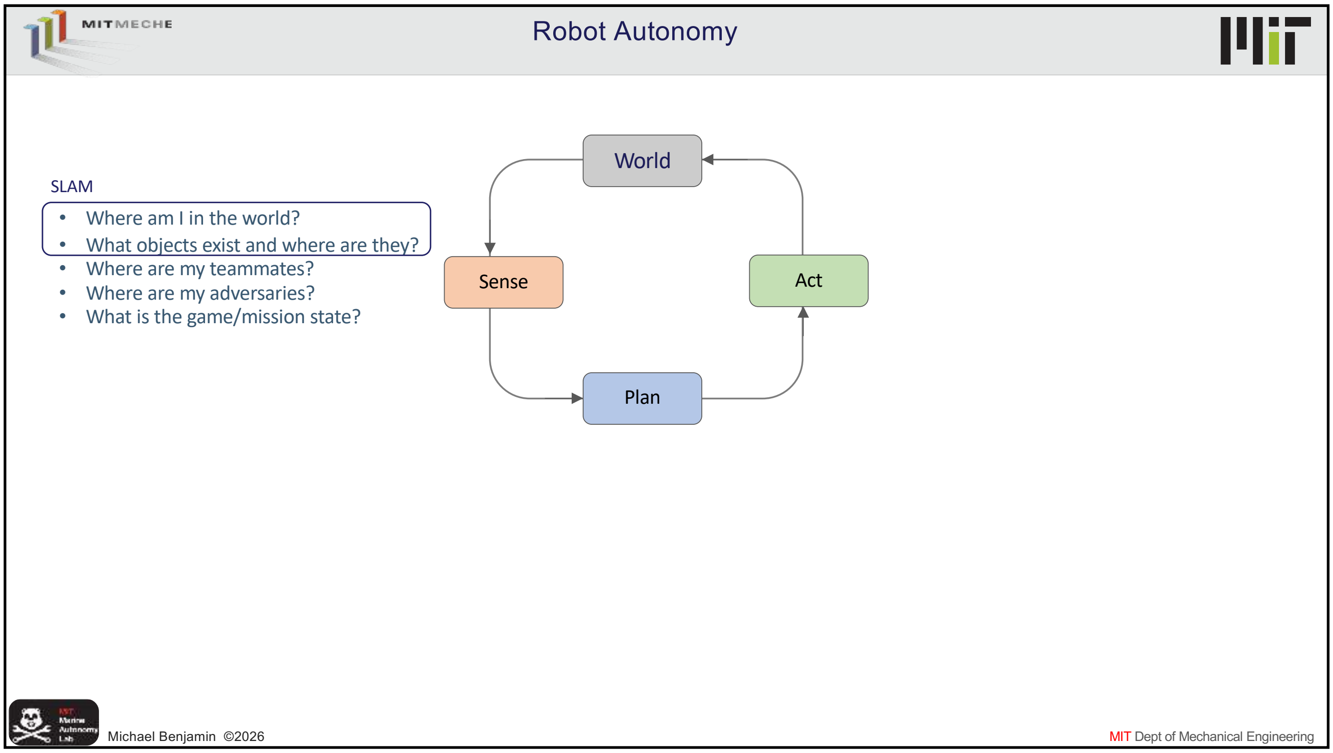
MITMECHE

Robot Autonomy

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

6



MITMECHE

Robot Autonomy

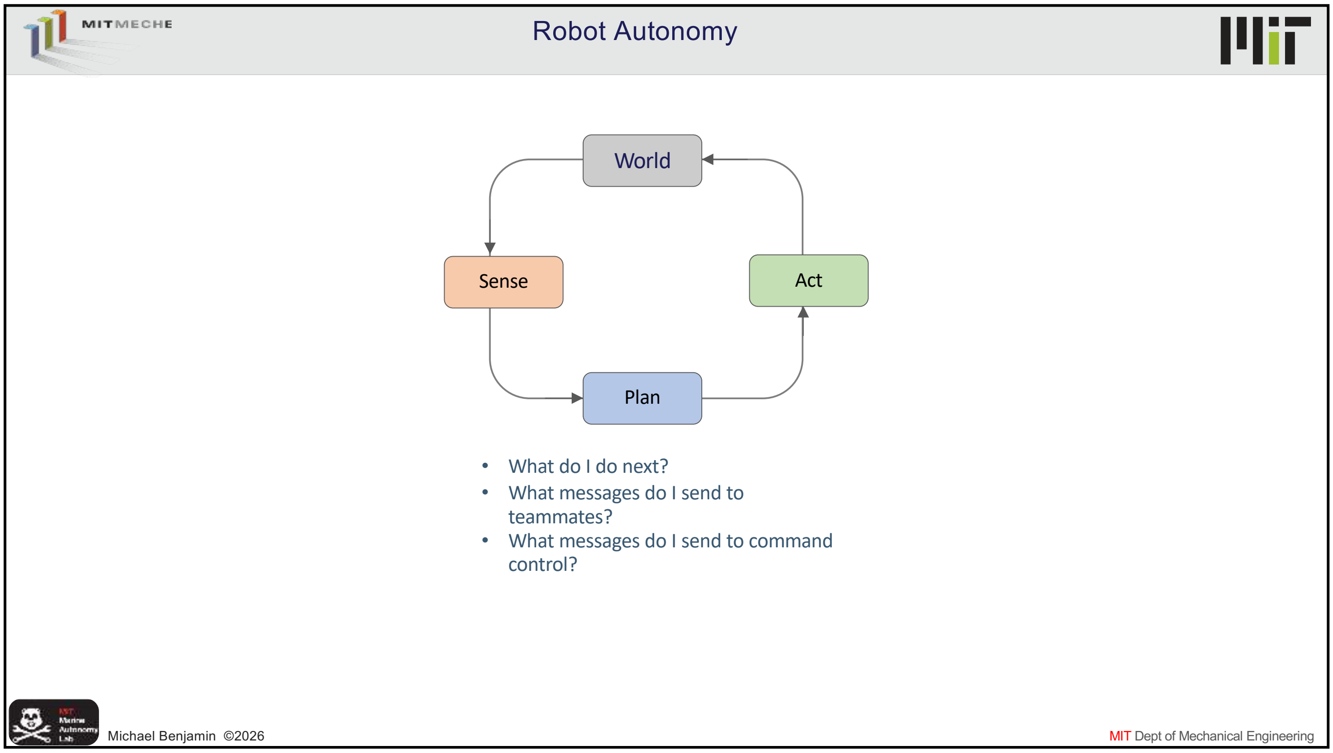
SLAM

- Where am I in the world?
- What objects exist and where are they?
- Where are my teammates?
- Where are my adversaries?
- What is the game/mission state?

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

7



MITMECHE

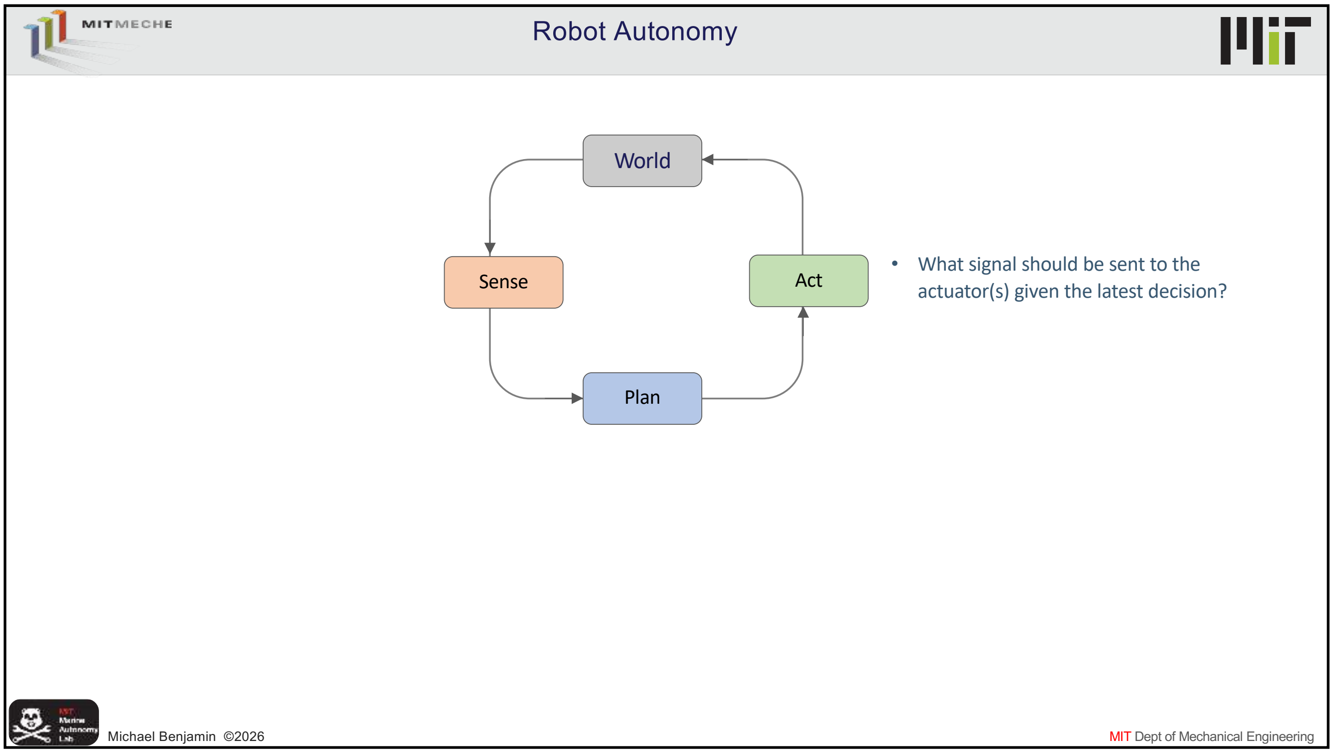
Robot Autonomy

- What do I do next?
- What messages do I send to teammates?
- What messages do I send to command control?

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

8



Robot Autonomy

World

Sense

Act

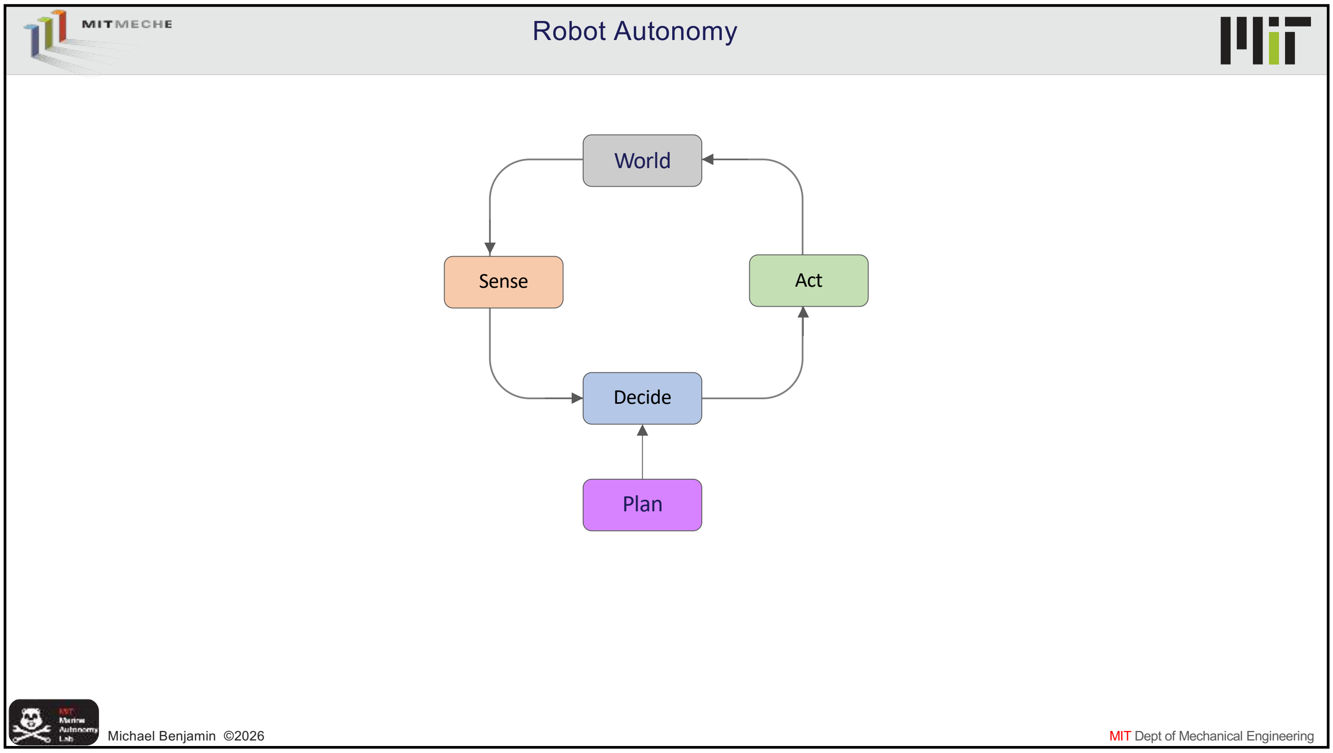
Plan

- What signal should be sent to the actuator(s) given the latest decision?

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

9



Robot Autonomy

World

Sense

Act

Decide

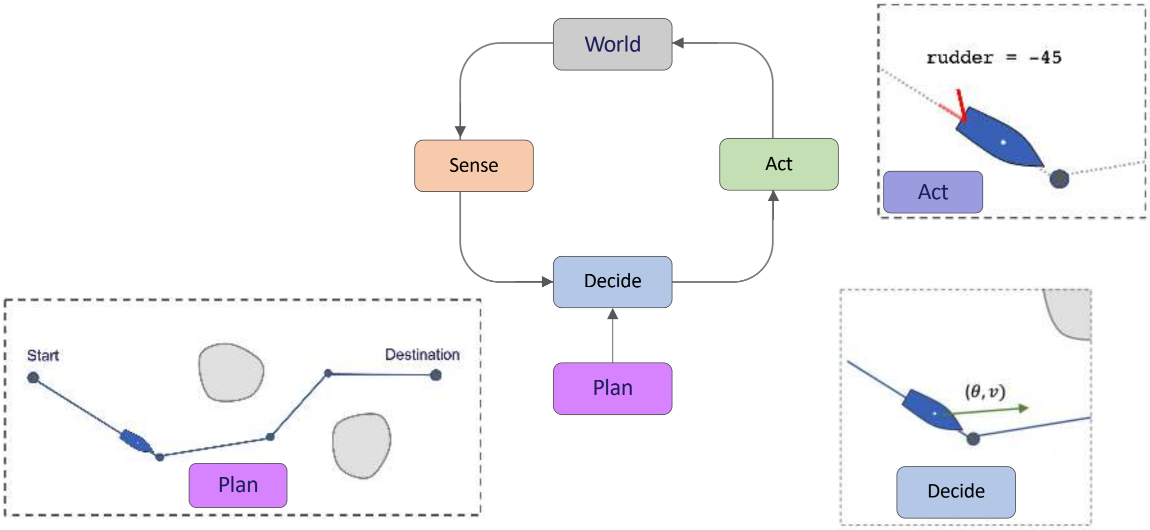
Plan

Michael Benjamin ©2026

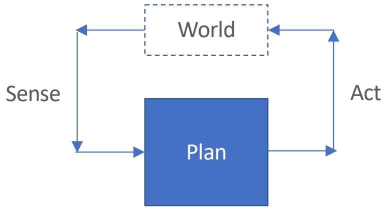
MIT Dept of Mechanical Engineering

10

Robot Autonomy

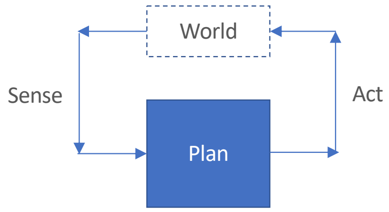


Sense – Plan - Act

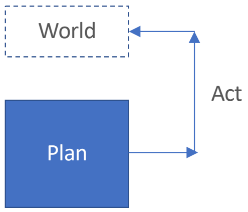


Virtually any robot can be described as sense-plan-act at some level.

Closed Loop vs Open Loop



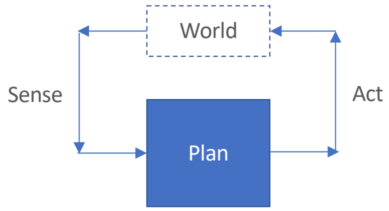
Closed Loop: The system has the ability to self-correct.



Open Loop: No position feedback, no sensed information of the environment.

13

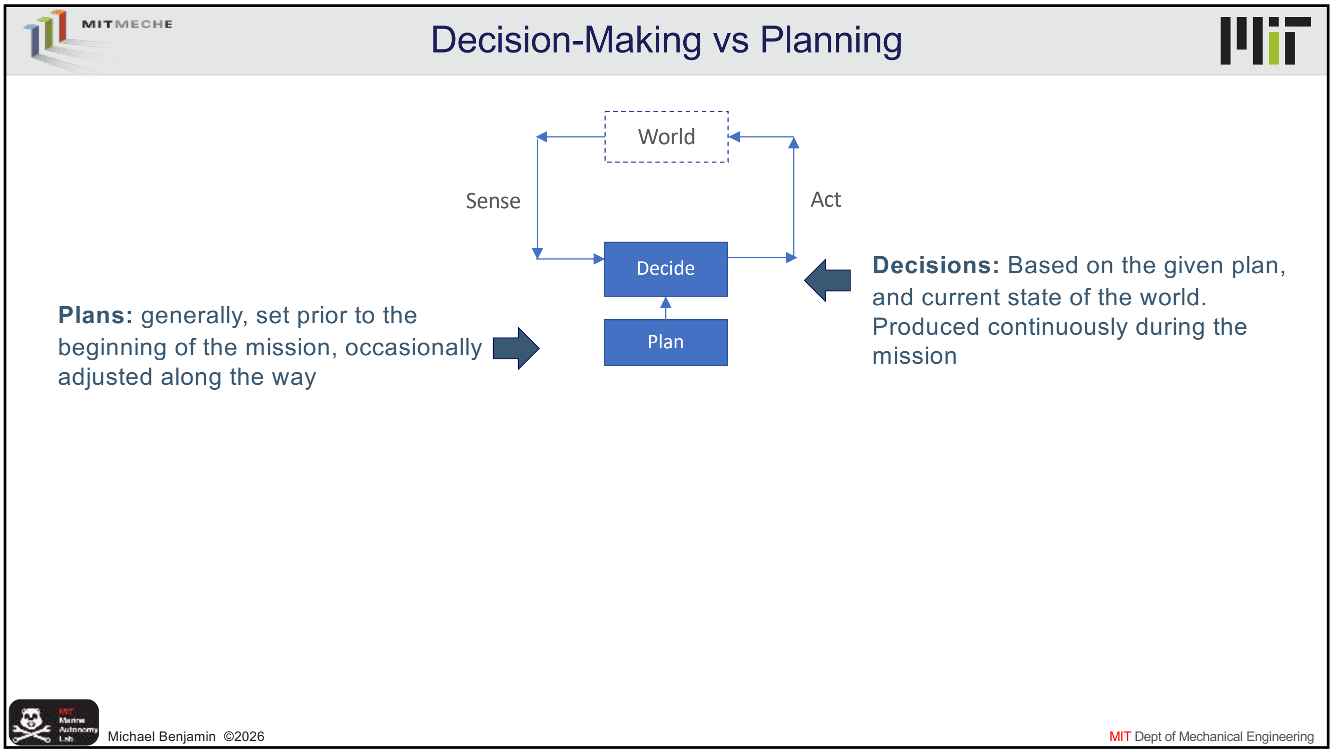
Closed Loop vs Open Loop



Closed Loop: The system has the ability to self-correct.

The Robots we consider in this class are all Closed Loop Systems

14

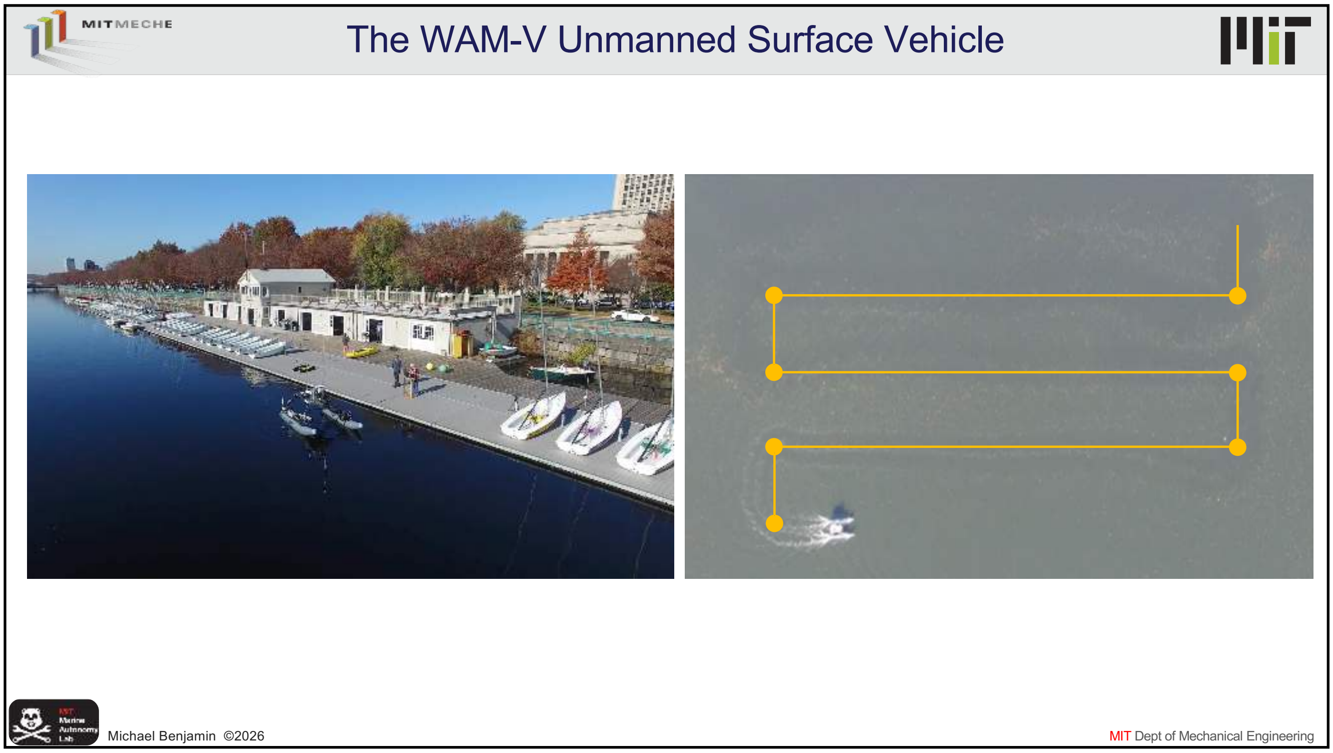


Decision-Making vs Planning

Plans: generally, set prior to the beginning of the mission, occasionally adjusted along the way

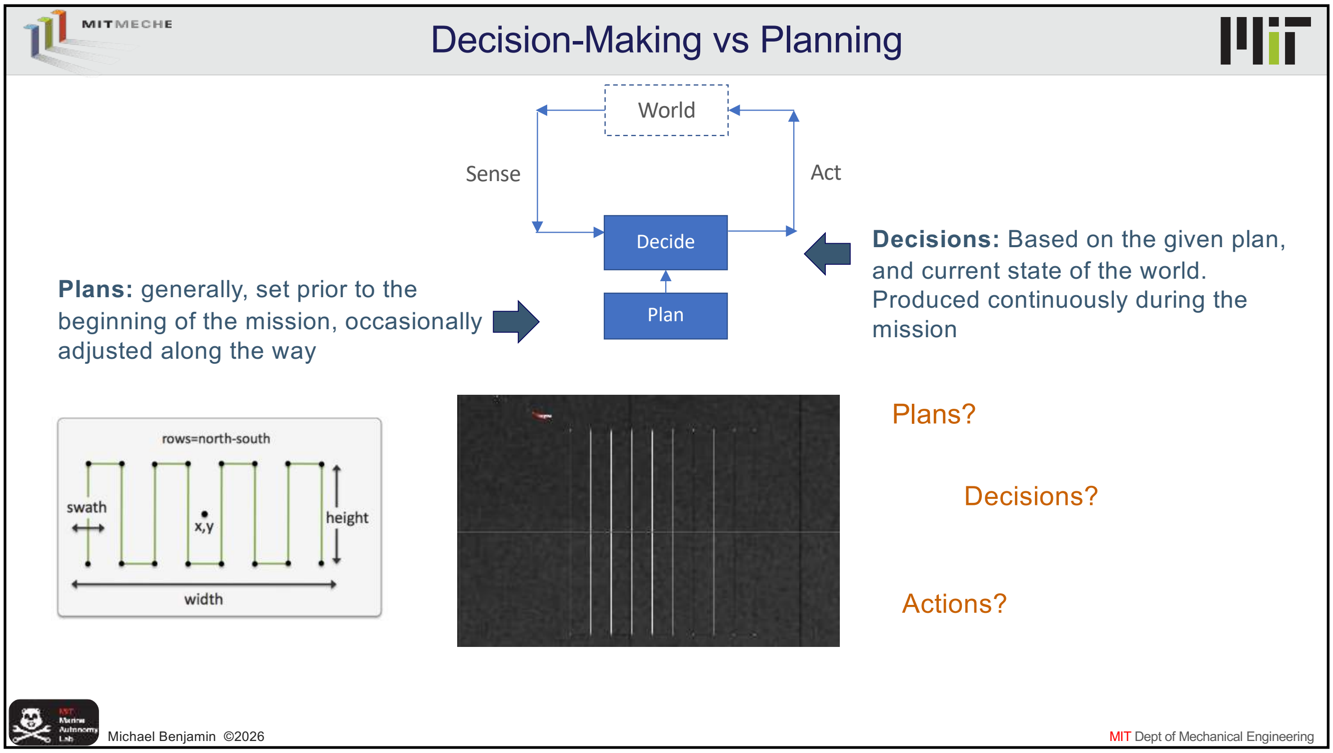
Decisions: Based on the given plan, and current state of the world. Produced continuously during the mission

15



The WAM-V Unmanned Surface Vehicle

16



Decision-Making vs Planning

Plans: generally, set prior to the beginning of the mission, occasionally adjusted along the way

Decisions: Based on the given plan, and current state of the world. Produced continuously during the mission

Plans?

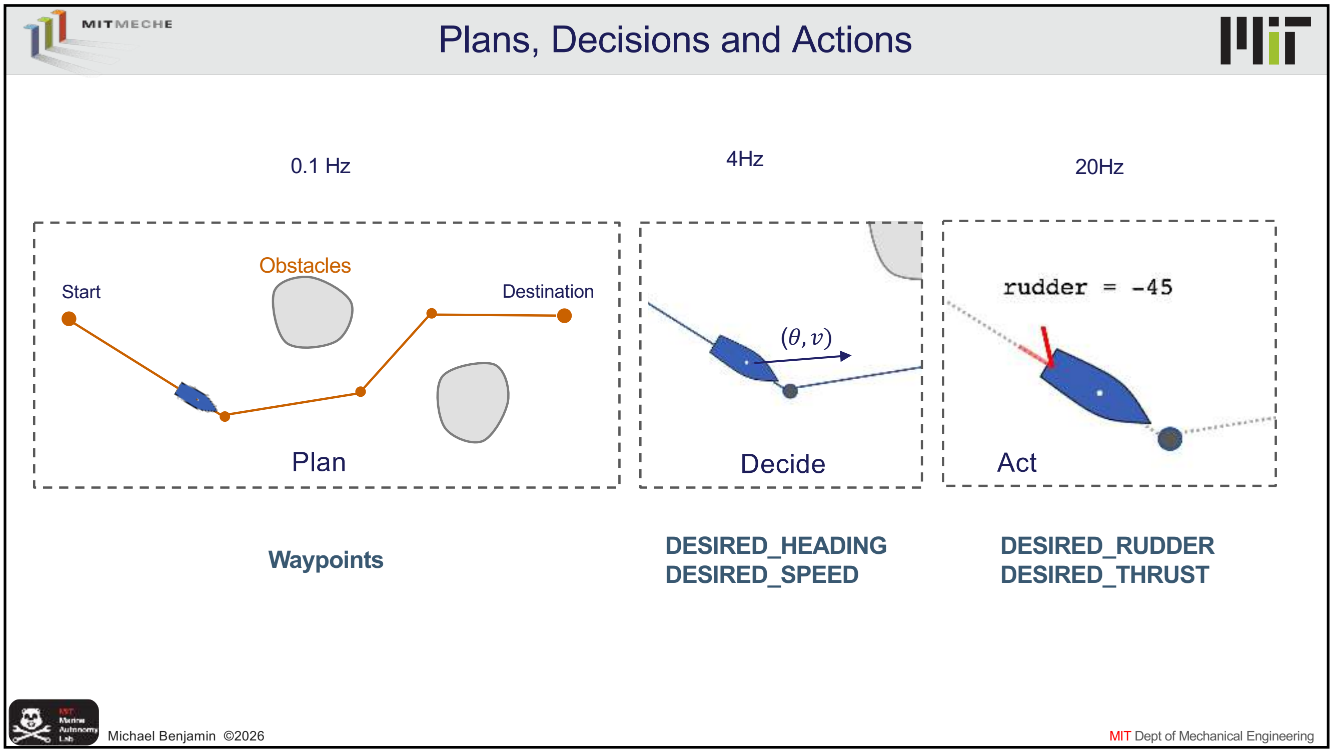
Decisions?

Actions?

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

17



Plans, Decisions and Actions

Waypoints

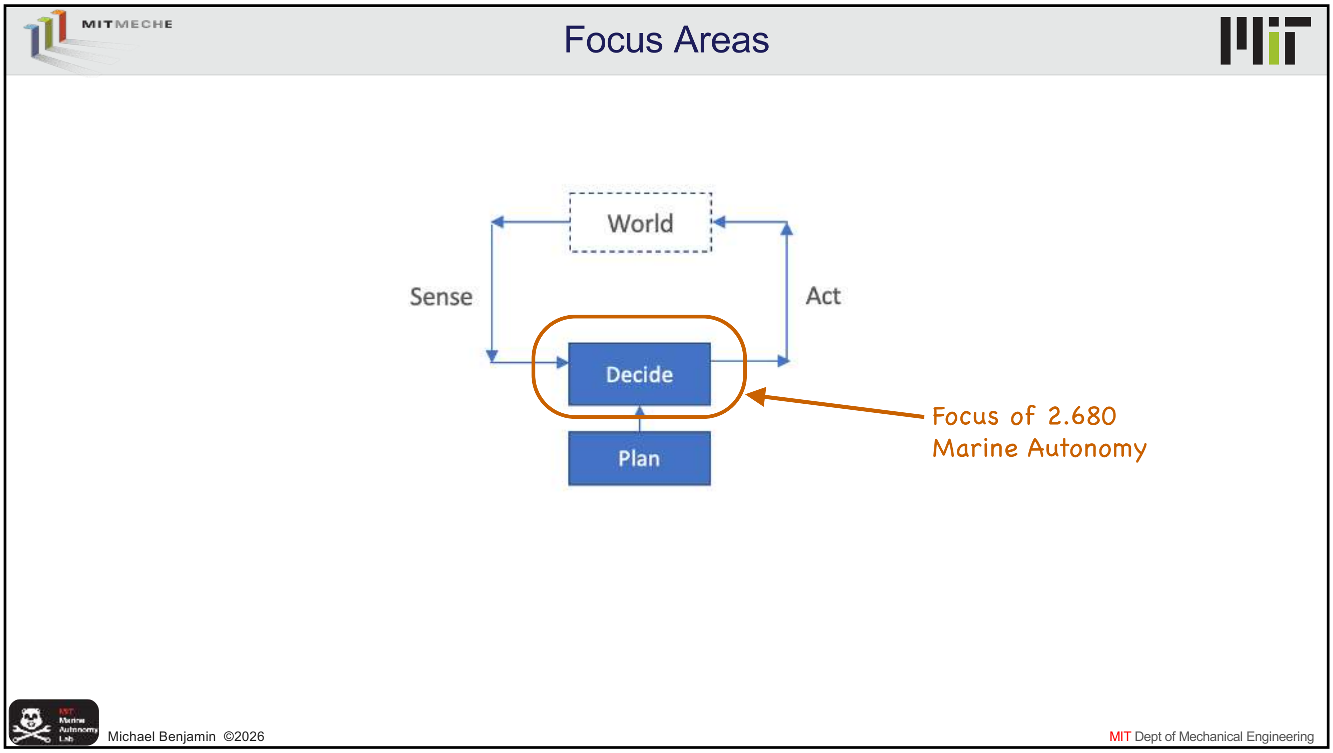
DESIRED_HEADING
DESIRED_SPEED

DESIRED_RUDDER
DESIRED_THRUST

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

18



Focus Areas

World

Sense

Act

Decide

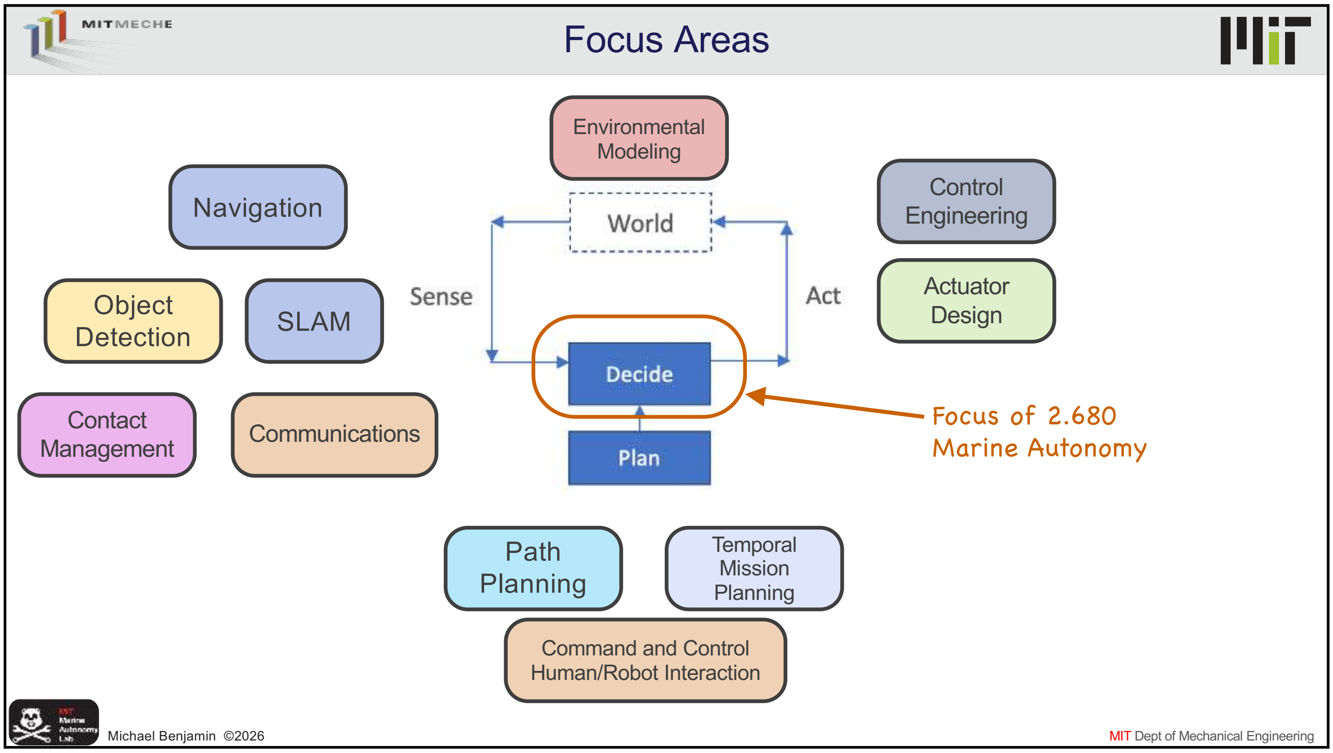
Plan

Focus of 2.680
Marine Autonomy

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

19



Focus Areas

Environmental Modeling

Navigation

Control Engineering

World

Object Detection

SLAM

Sense

Act

Actuator Design

Contact Management

Communications

Decide

Plan

Focus of 2.680
Marine Autonomy

Path Planning

Temporal Mission Planning

Command and Control
Human/Robot Interaction

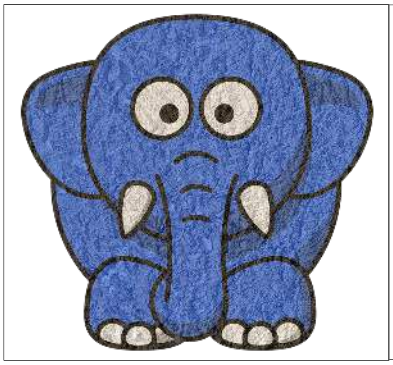
Michael Benjamin ©2026

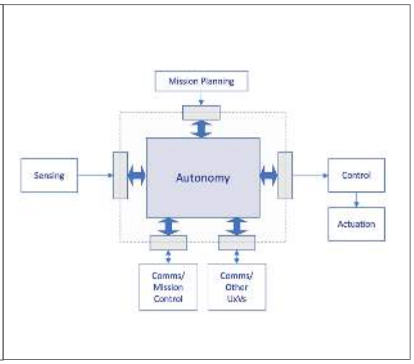
MIT Dept of Mechanical Engineering

20

Robot Components/Architecture Perspectives

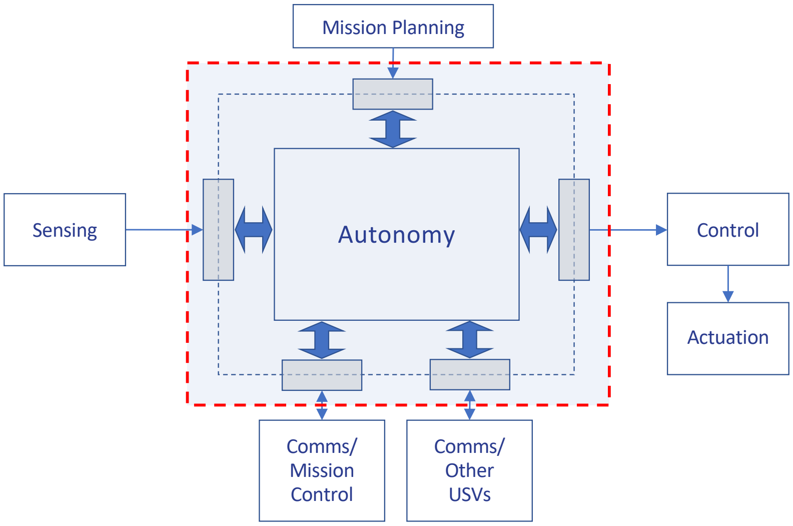
Functional Components





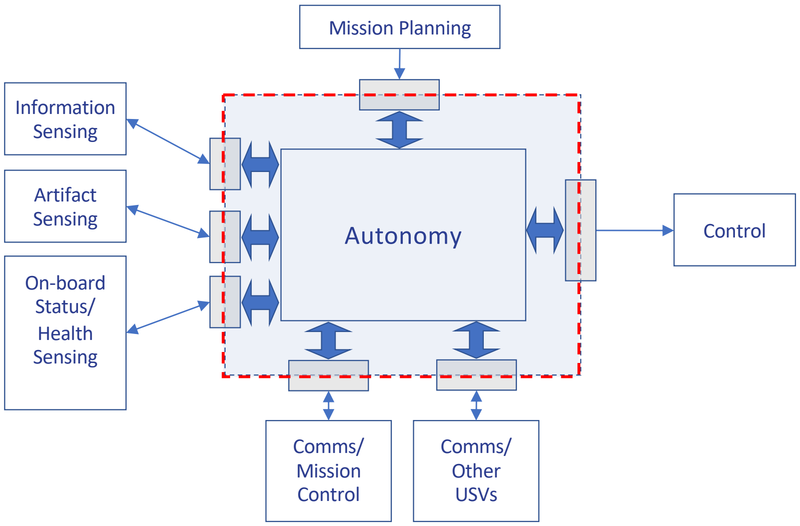
21

Functional Components



22

Functional Components



Mission Planning

Information Sensing

Artifact Sensing

On-board Status/ Health Sensing

Autonomy

Control

Comms/ Mission Control

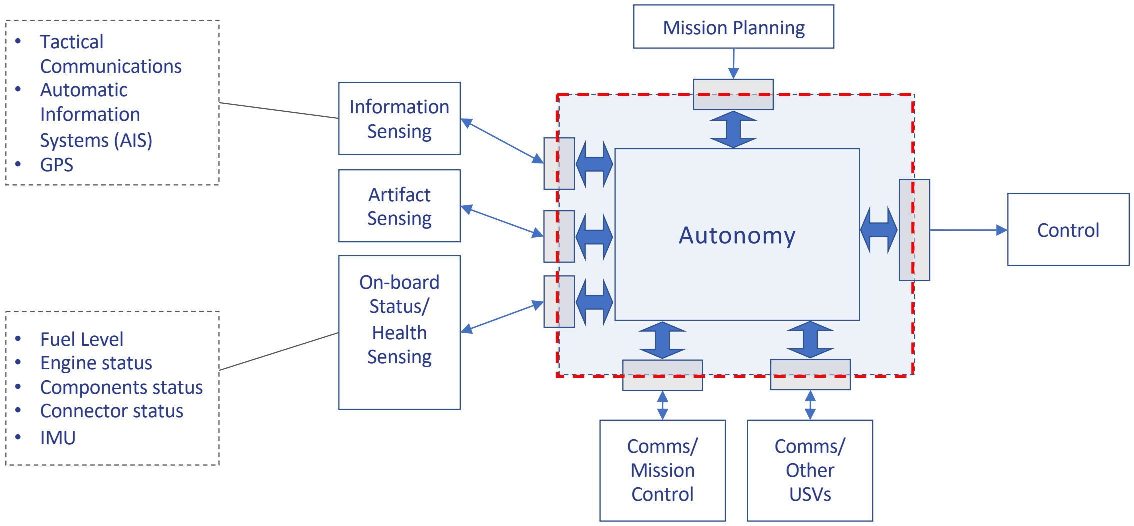
Comms/ Other USVs

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

23

Functional Components



- Tactical Communications
- Automatic Information Systems (AIS)
- GPS

- Fuel Level
- Engine status
- Components status
- Connector status
- IMU

Mission Planning

Information Sensing

Artifact Sensing

On-board Status/ Health Sensing

Autonomy

Control

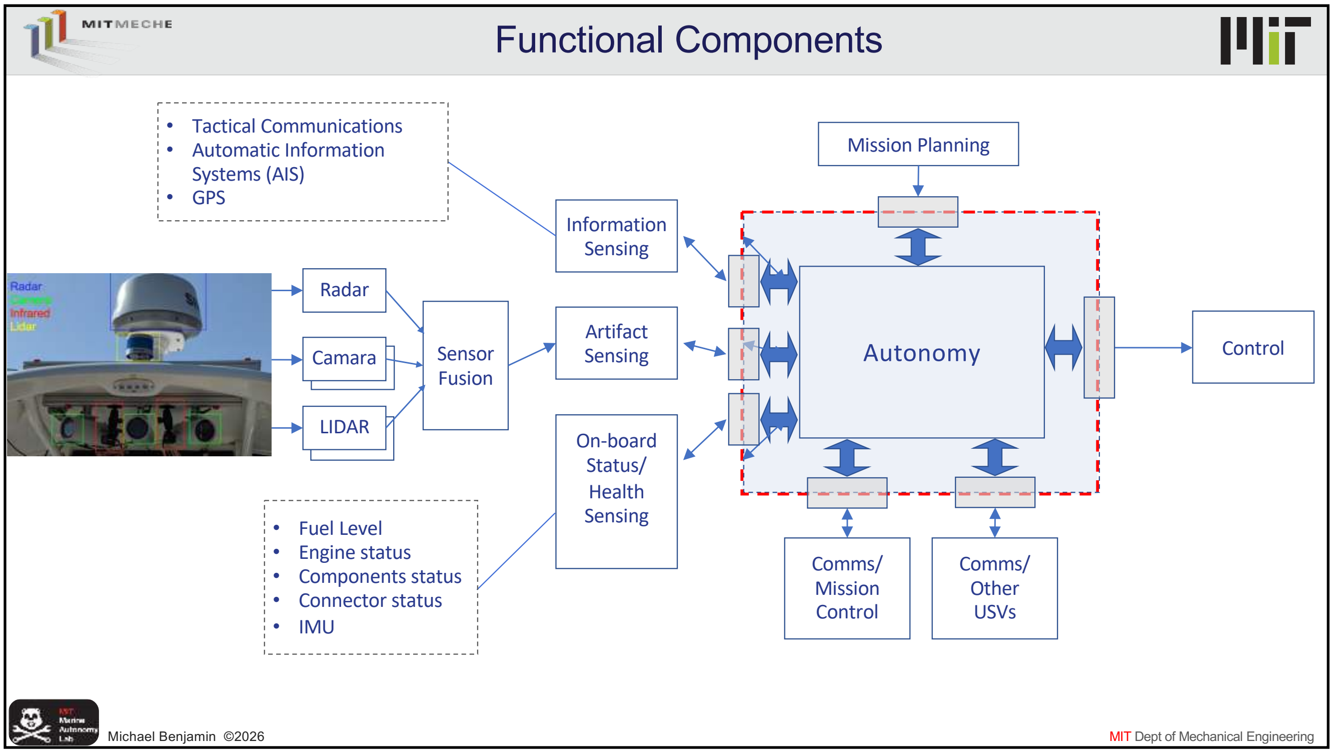
Comms/ Mission Control

Comms/ Other USVs

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

24



Functional Components

- Tactical Communications
- Automatic Information Systems (AIS)
- GPS

Radar

Camara

LIDAR

Sensor Fusion

Information Sensing

Artifact Sensing

On-board Status/ Health Sensing

- Fuel Level
- Engine status
- Components status
- Connector status
- IMU

Mission Planning

Autonomy

Control

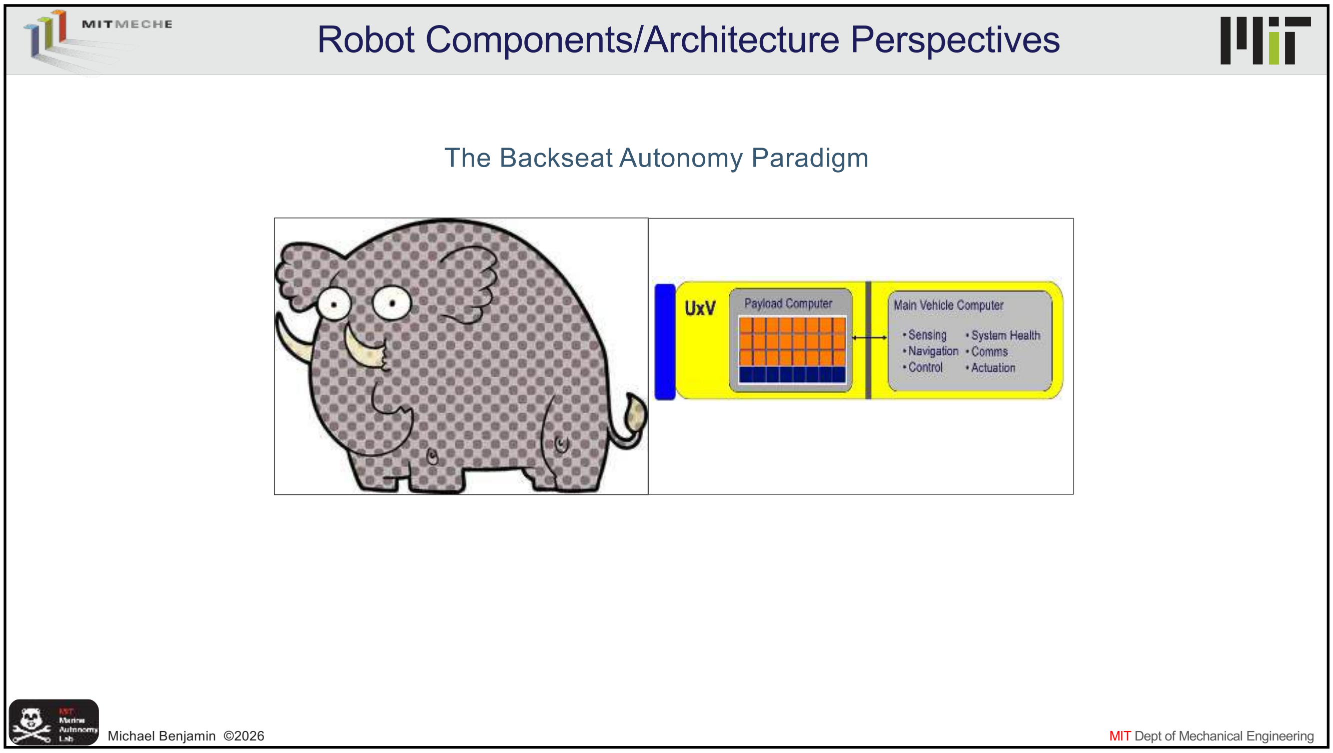
Comms/ Mission Control

Comms/ Other USVs

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

25



Robot Components/Architecture Perspectives

The Backseat Autonomy Paradigm

UxV

Payload Computer

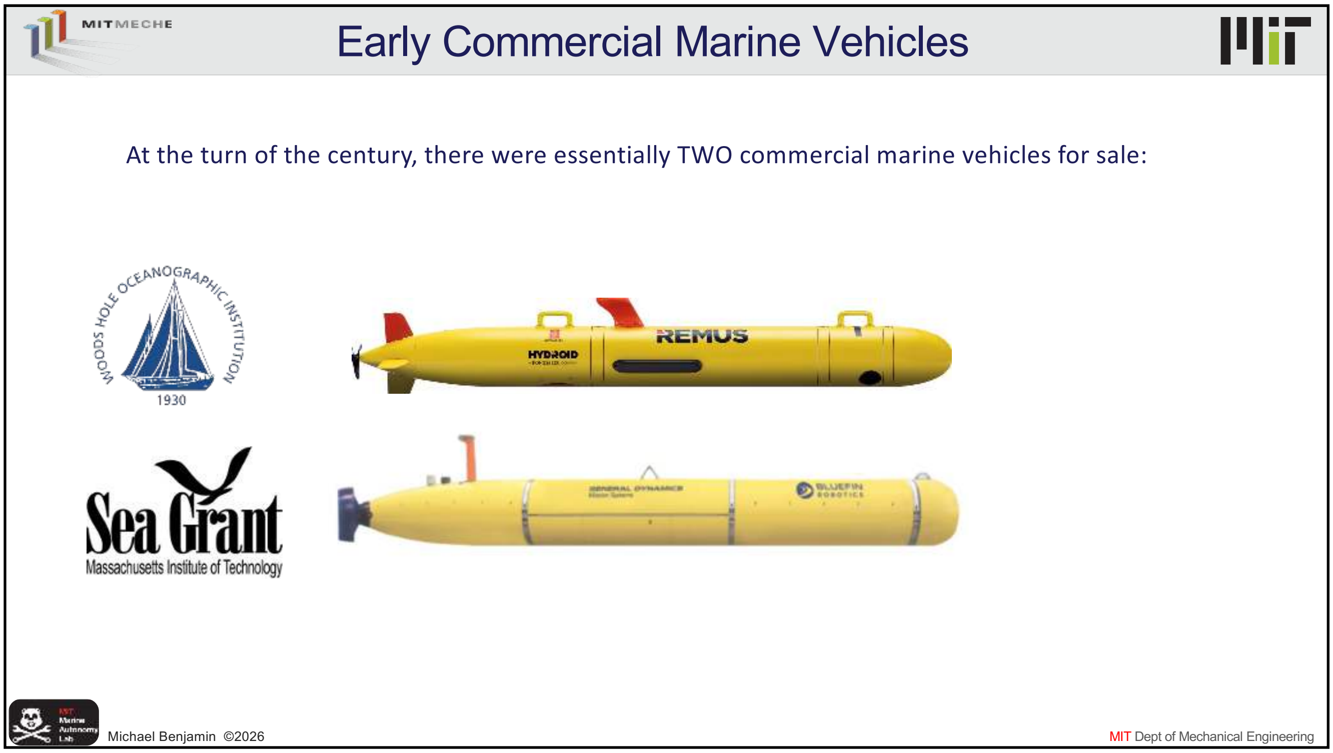
Main Vehicle Computer

- Sensing
- Navigation
- Control
- System Health
- Comms
- Actuation

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

26



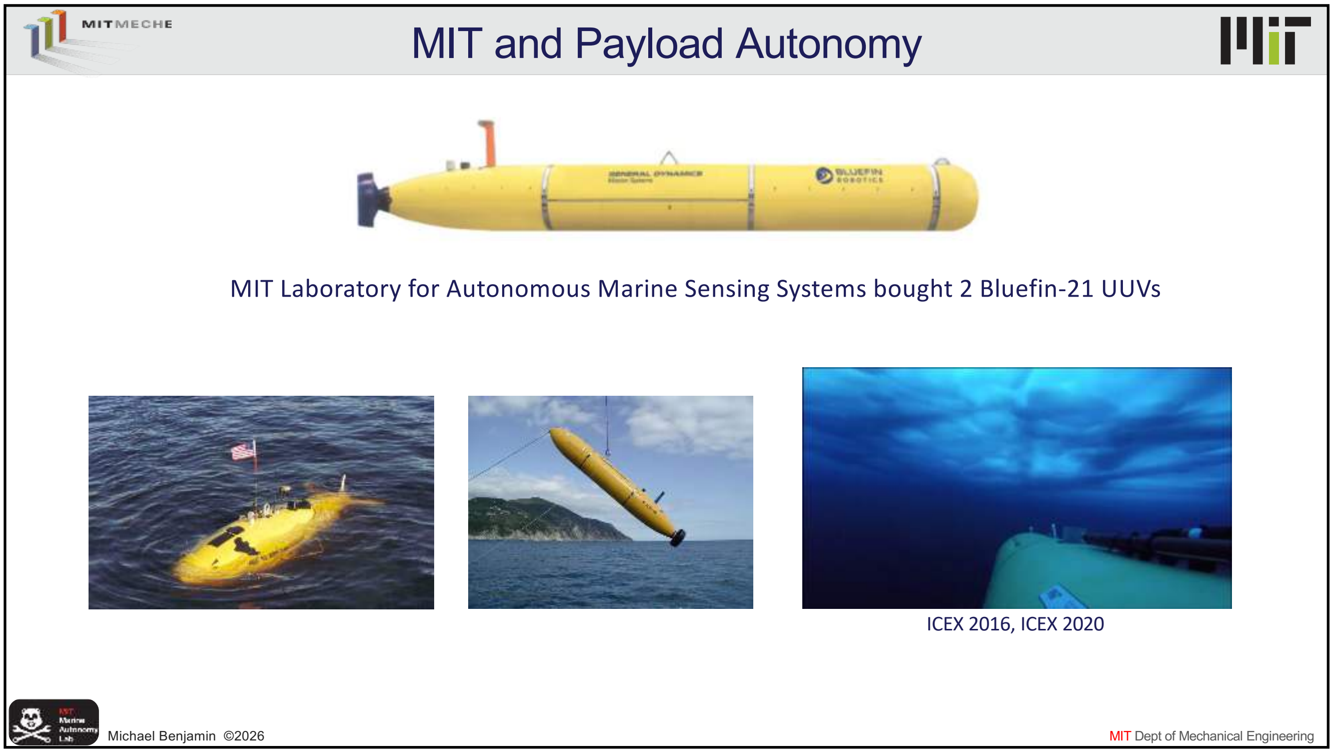
Early Commercial Marine Vehicles

At the turn of the century, there were essentially TWO commercial marine vehicles for sale:

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

27



MIT and Payload Autonomy

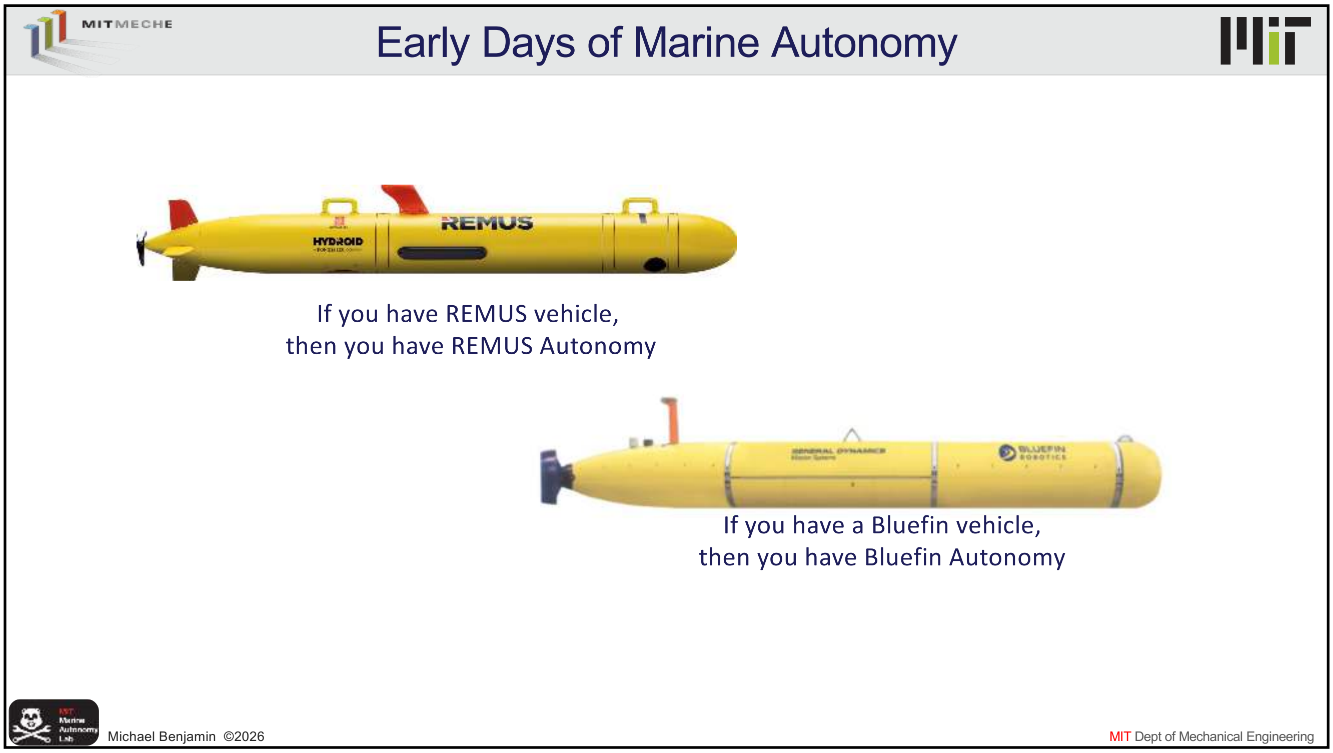
MIT Laboratory for Autonomous Marine Sensing Systems bought 2 Bluefin-21 UUVs

ICEX 2016, ICEX 2020

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

28



Early Days of Marine Autonomy

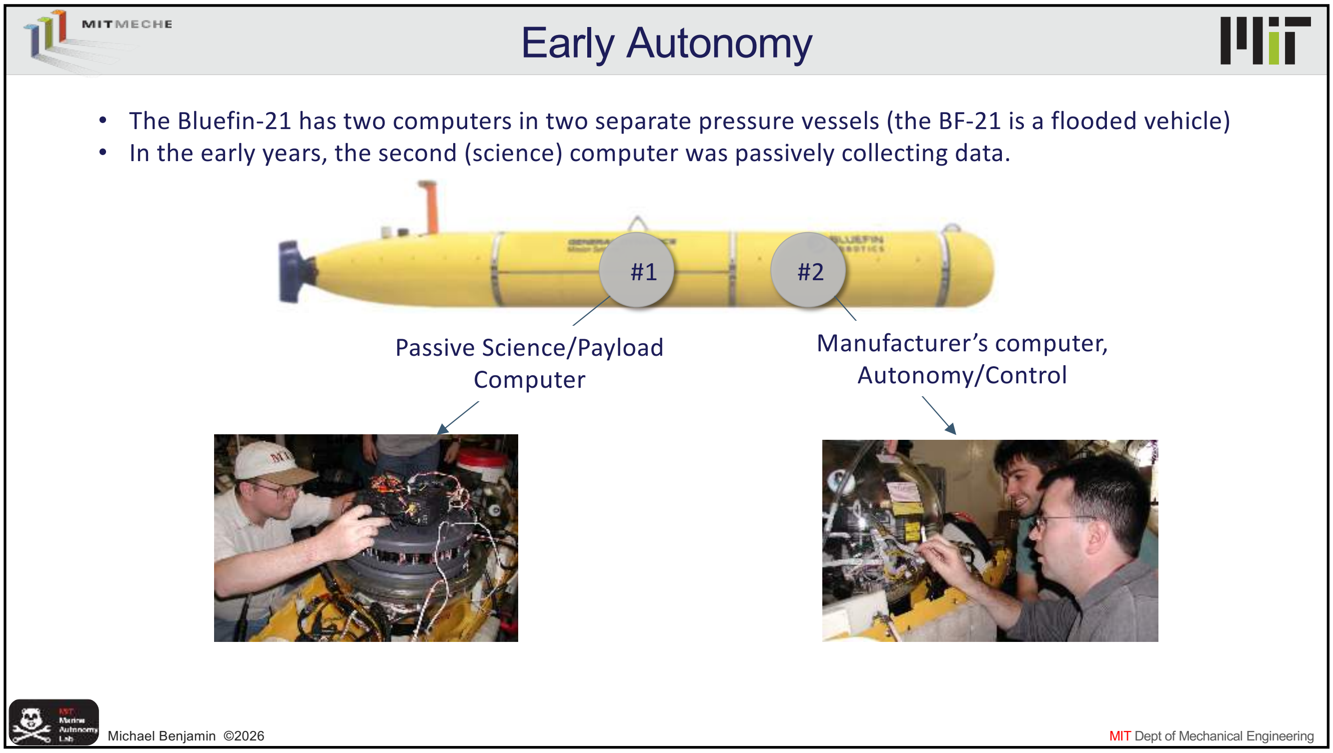
If you have REMUS vehicle,
then you have REMUS Autonomy

If you have a Bluefin vehicle,
then you have Bluefin Autonomy

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

29



Early Autonomy

- The Bluefin-21 has two computers in two separate pressure vessels (the BF-21 is a flooded vehicle)
- In the early years, the second (science) computer was passively collecting data.

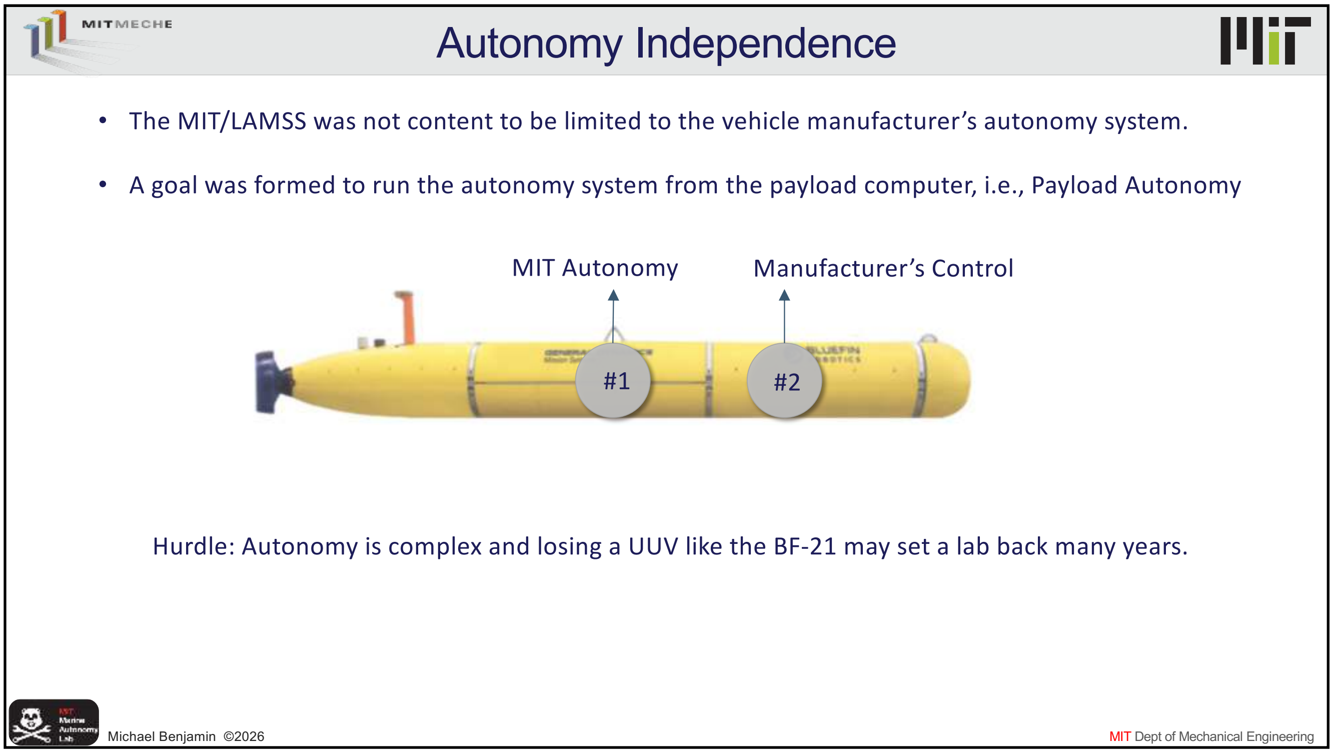
Passive Science/Payload
Computer

Manufacturer’s computer,
Autonomy/Control

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

30



Autonomy Independence

- The MIT/LAMSS was not content to be limited to the vehicle manufacturer’s autonomy system.
- A goal was formed to run the autonomy system from the payload computer, i.e., Payload Autonomy

MIT Autonomy

Manufacturer’s Control

#1

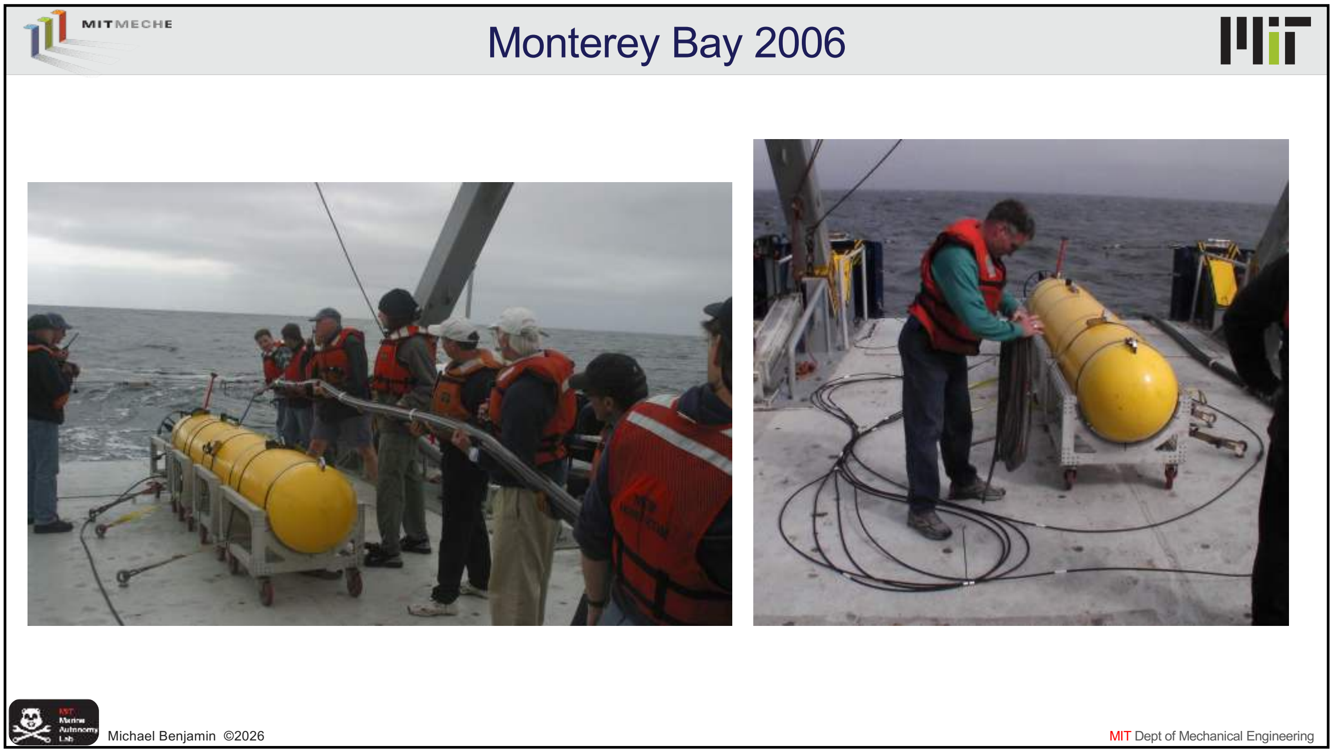
#2

Hurdle: Autonomy is complex and losing a UUV like the BF-21 may set a lab back many years.

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

31

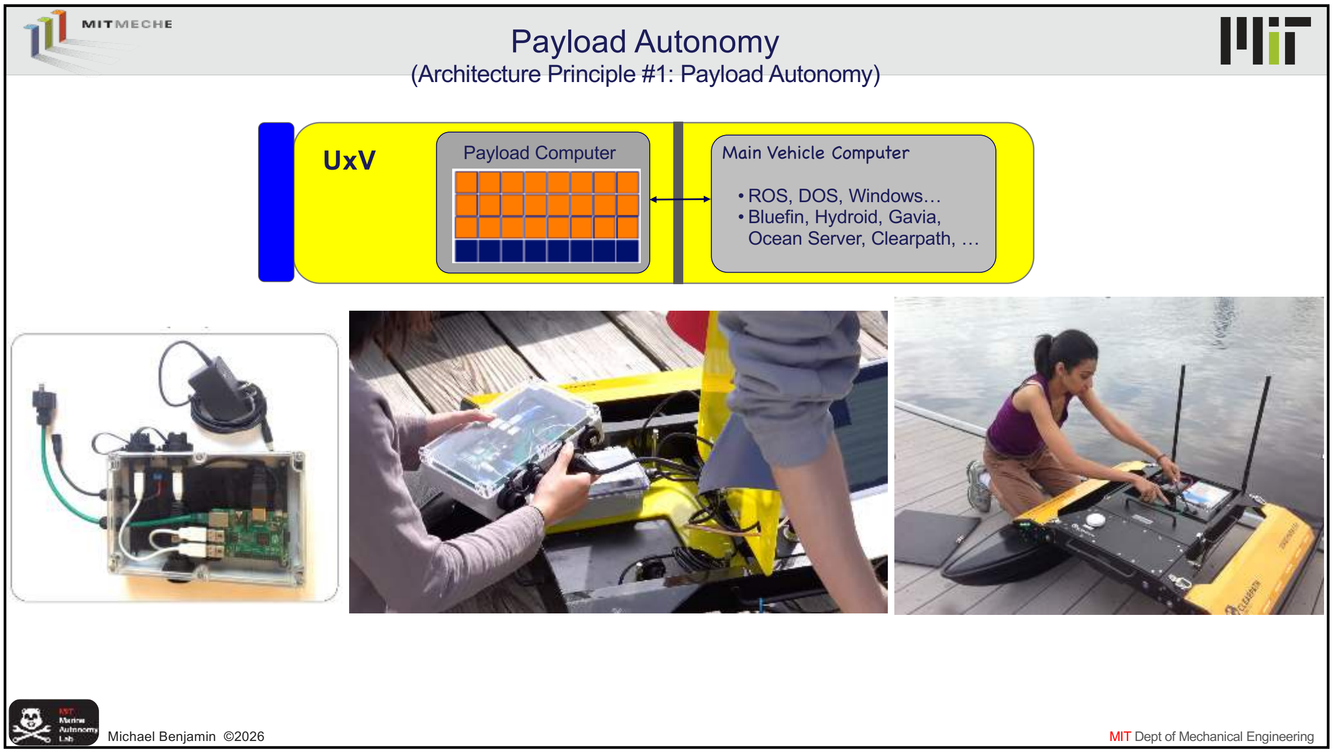


Monterey Bay 2006

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

32



Payload Autonomy

(Architecture Principle #1: Payload Autonomy)

UxV

Payload Computer

Main Vehicle Computer

- ROS, DOS, Windows…
- Bluefin, Hydroid, Gavia, Ocean Server, Clearpath, …

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

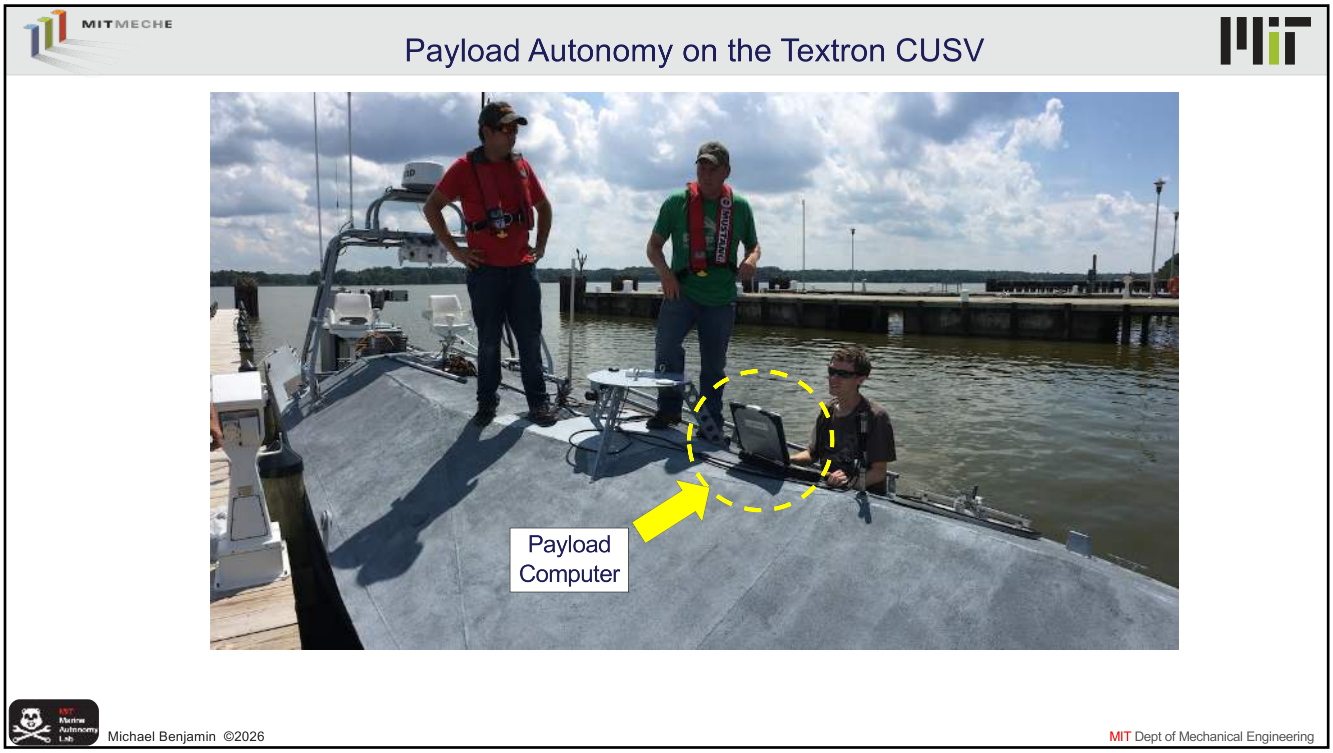
33

Payload Autonomy on the Textron CUSV

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

34

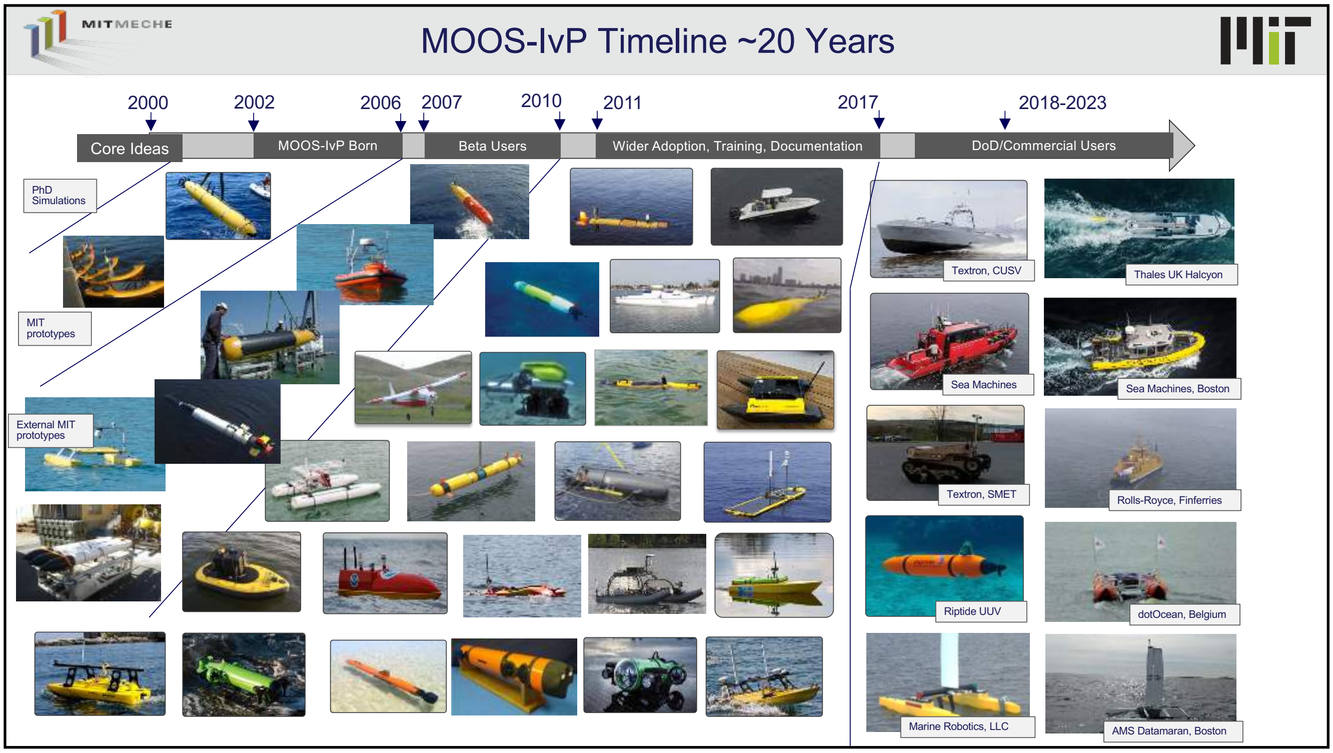


Payload Autonomy on the Textron CUSV

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

35

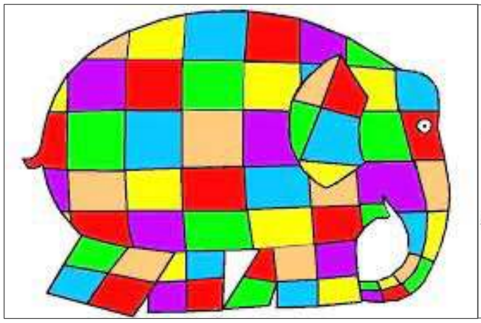


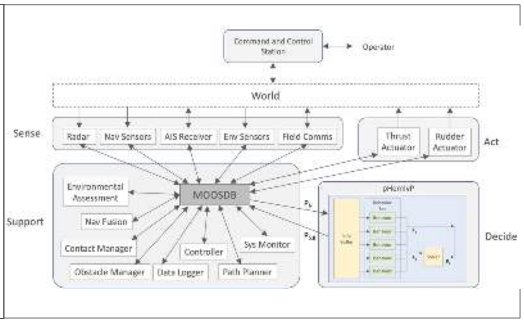
MOOS-IvP Timeline ~20 Years

36

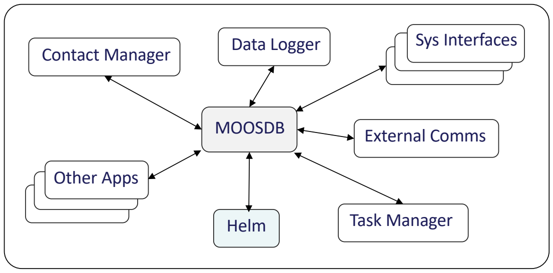
Robot Components/Architecture Perspectives

The MOOS and MOOS-IvP Architectures





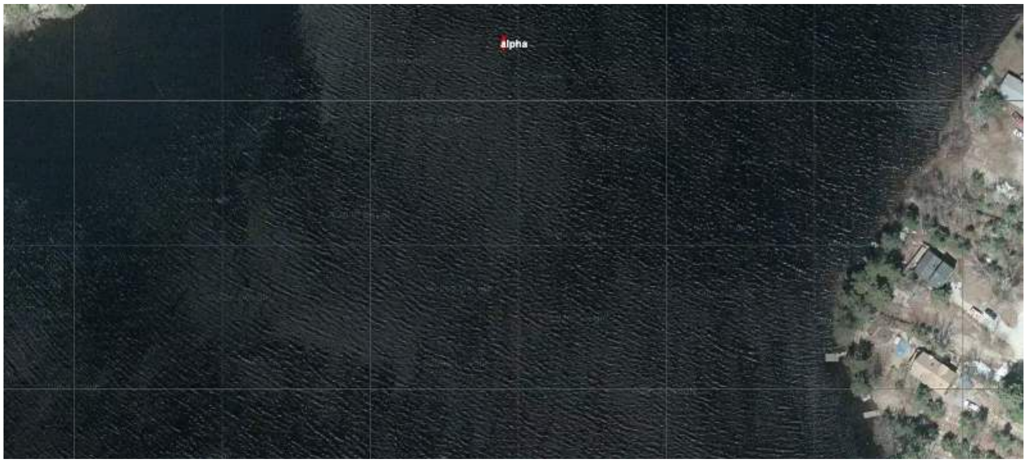
MOOS and MOOS Apps



MITMECHE

Example Mission Alpha

```
$ cd moos-ivp/ivp/missions/s1_alpha
$ ./launch.sh 10
```



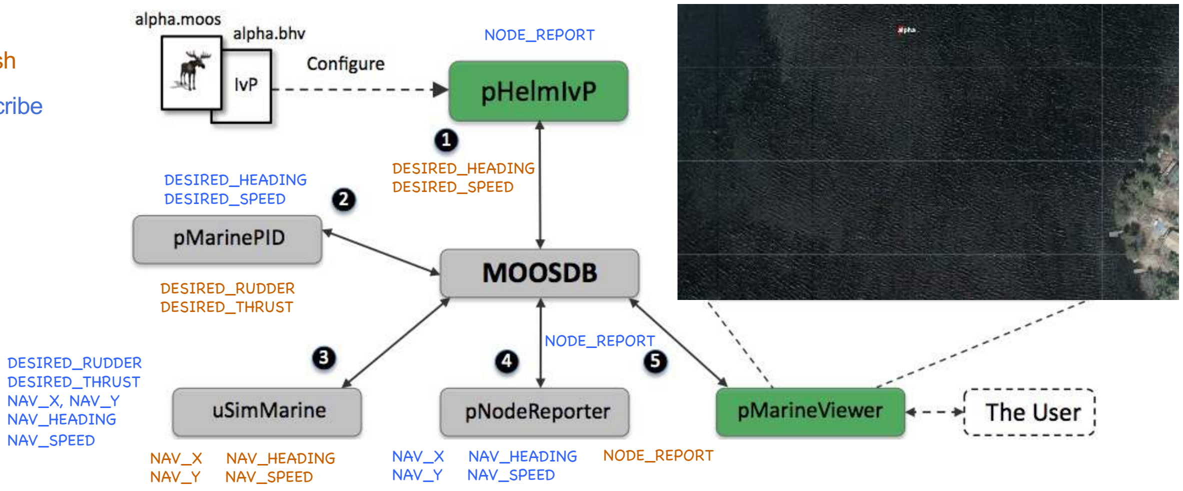
MITMECHE

Example Mission Alpha

```
$ cd moos-ivp/ivp/missions/s1_alpha
$ ./launch.sh 10
```

🟧 Publish

🟦 Subscribe


alpha.moos

alpha.bhv

IvP

Configure

NODE_REPORT

pHelmIvP

1

DESIRED_HEADING
DESIRED_SPEED

DESIRED_HEADING
DESIRED_SPEED

2

pMarinePID

MOOSDB

DESIRED_RUDDER
DESIRED_THRUST

NODE_REPORT

3

4

5

DESIRED_RUDDER
DESIRED_THRUST
NAV_X, NAV_Y
NAV_HEADING
NAV_SPEED

uSimMarine

pNodeReporter

pMarineViewer

The User

NAV_X
NAV_Y

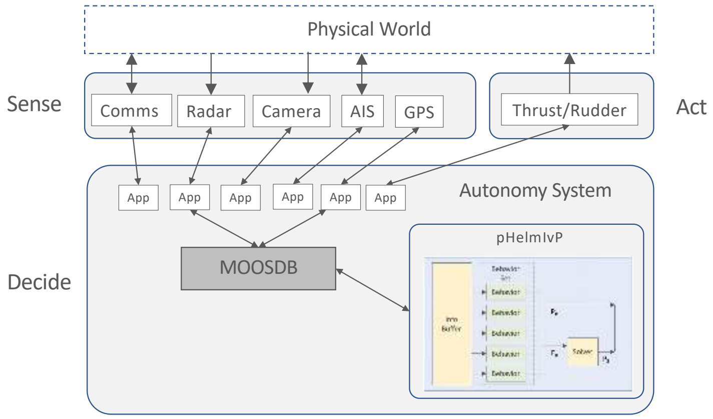
NAV_HEADING
NAV_SPEED

NAV_X
NAV_Y

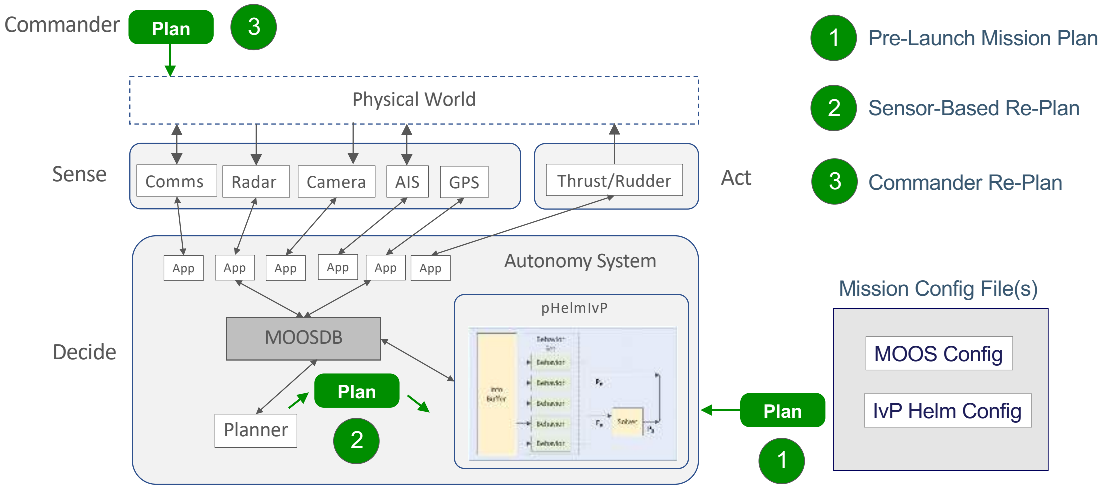
NAV_HEADING
NAV_SPEED

NODE_REPORT

Example Mission Alpha



Planning Sources



1 Pre-Launch Mission Plan

2 Sensor-Based Re-Plan

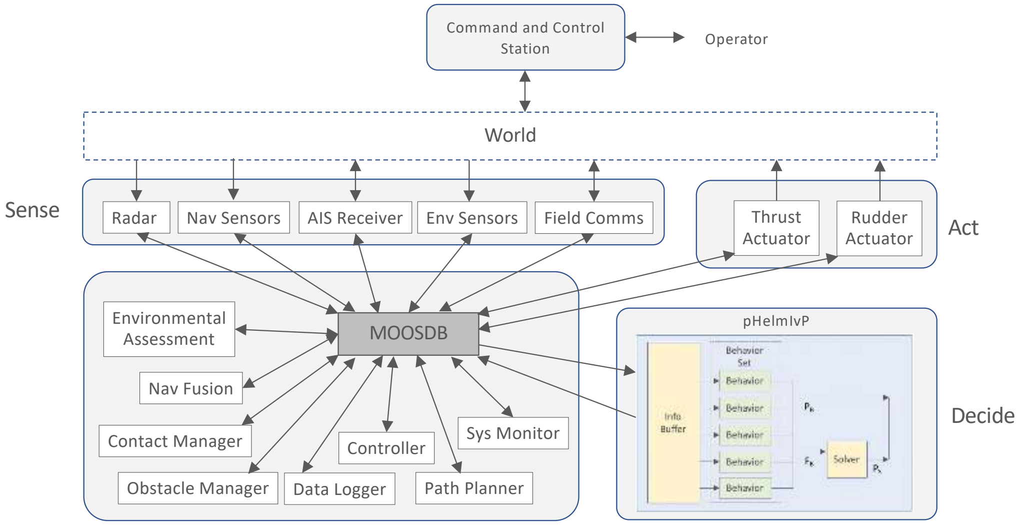
3 Commander Re-Plan

Mission Config File(s)

MOOS Config

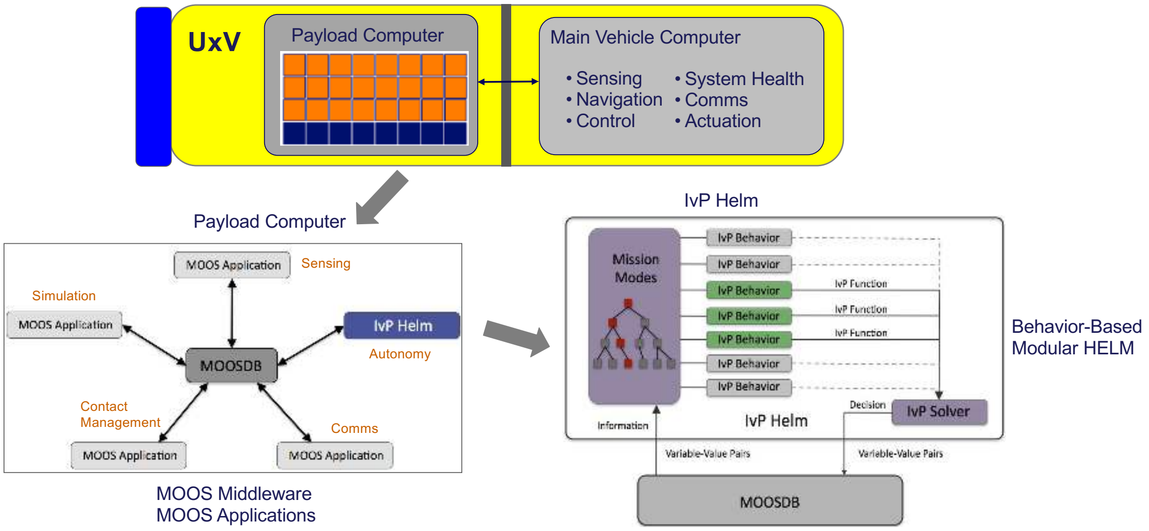
IvP Helm Config

Example Mission Alpha



43

MOOS-IvP Payload Autonomy



44

Lecture 1 Overview: Introduction to MIT 2.680

Robot Architectures: The focus of 2.680 in the larger context of mobile robot technologies

Skills Progressions: The parallel progression of skills expected during the course of 2.680 and labs.

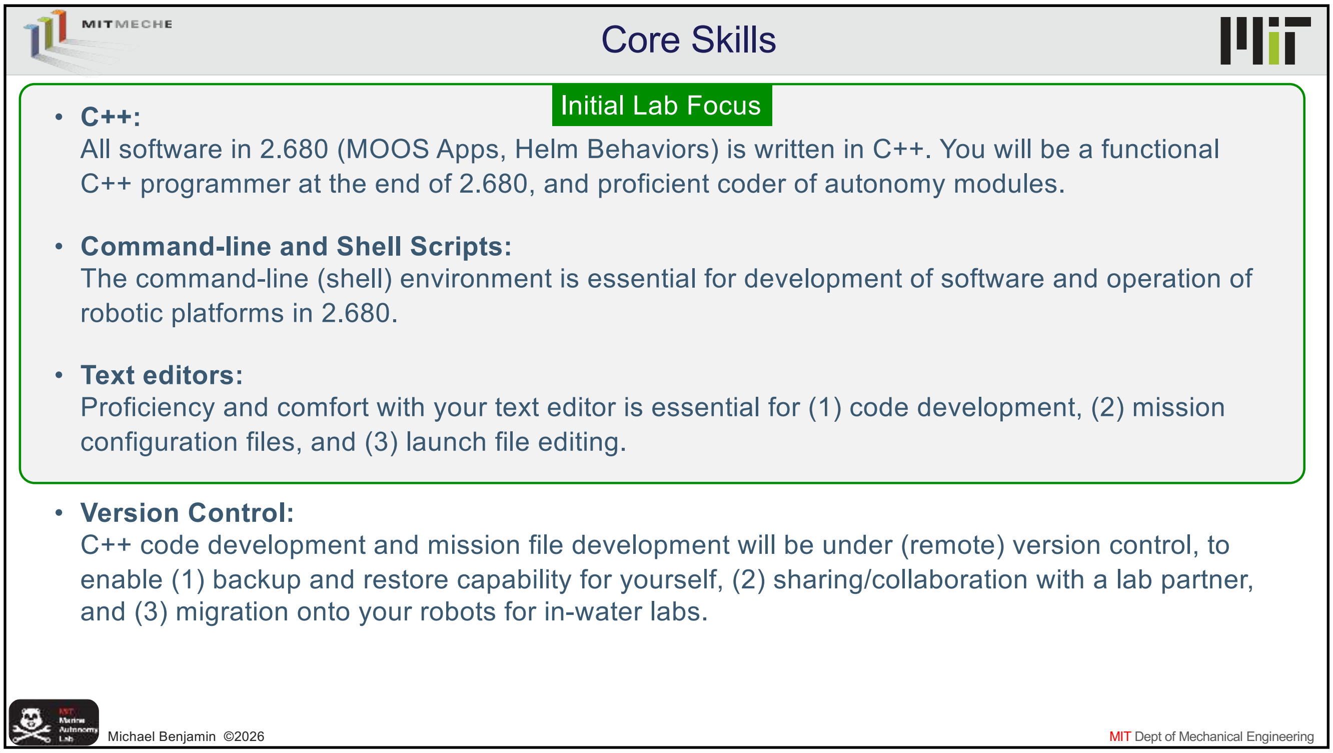
2.680 Learning Resources: Beyond lectures: MIT 2.680 labs, office hours (in-person and virtual), 2.680 Website, 2.680 Piazza page.

Marine Autonomy Lab Overview: An introduction to existing robotic platforms and projects related to MIT 2.680

45

Skills Progressions

46



Core Skills

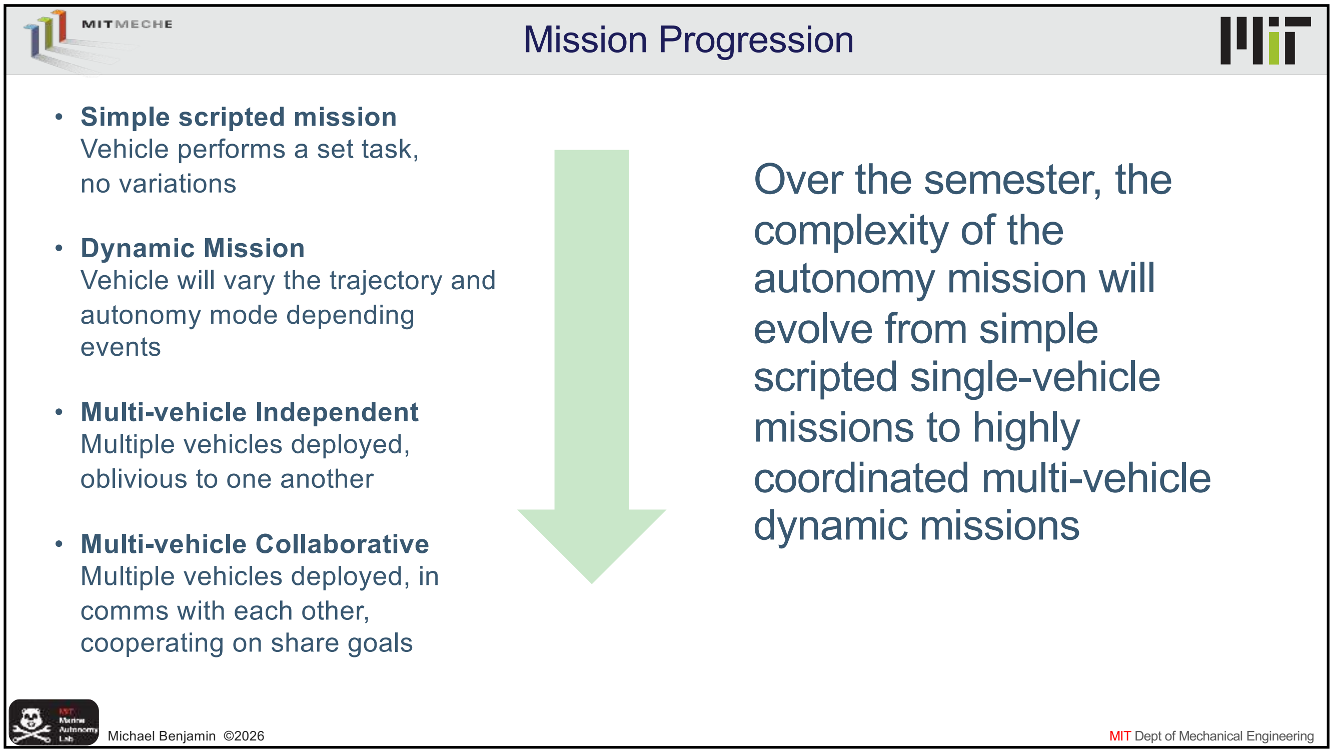
Initial Lab Focus

- **C++:**
  All software in 2.680 (MOOS Apps, Helm Behaviors) is written in C++. You will be a functional C++ programmer at the end of 2.680, and proficient coder of autonomy modules.

- **Command-line and Shell Scripts:**
  The command-line (shell) environment is essential for development of software and operation of robotic platforms in 2.680.

- **Text editors:**
  Proficiency and comfort with your text editor is essential for (1) code development, (2) mission configuration files, and (3) launch file editing.

- **Version Control:**
  C++ code development and mission file development will be under (remote) version control, to enable (1) backup and restore capability for yourself, (2) sharing/collaboration with a lab partner, and (3) migration onto your robots for in-water labs.



Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations

- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events

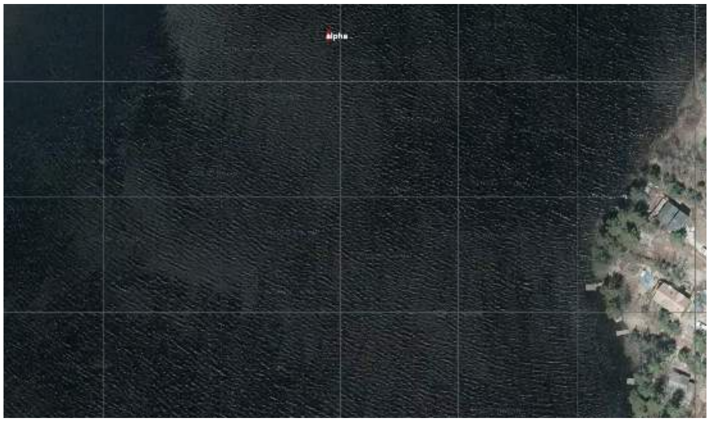
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another

- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals

Over the semester, the complexity of the autonomy mission will evolve from simple scripted single-vehicle missions to highly coordinated multi-vehicle dynamic missions

Mission Progression

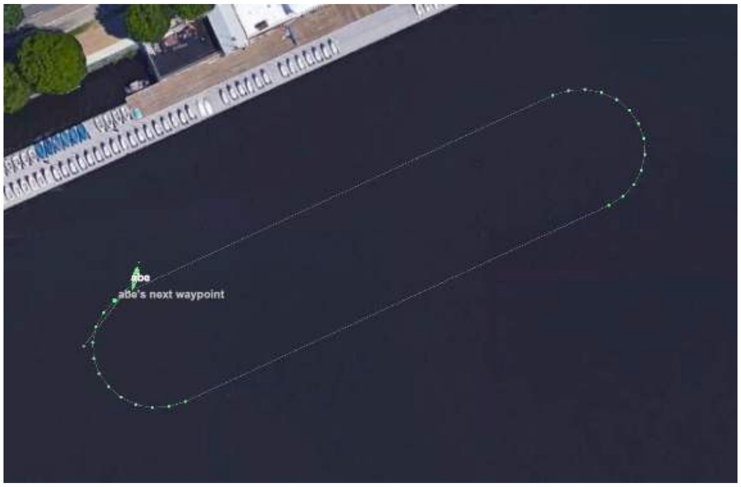
- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals



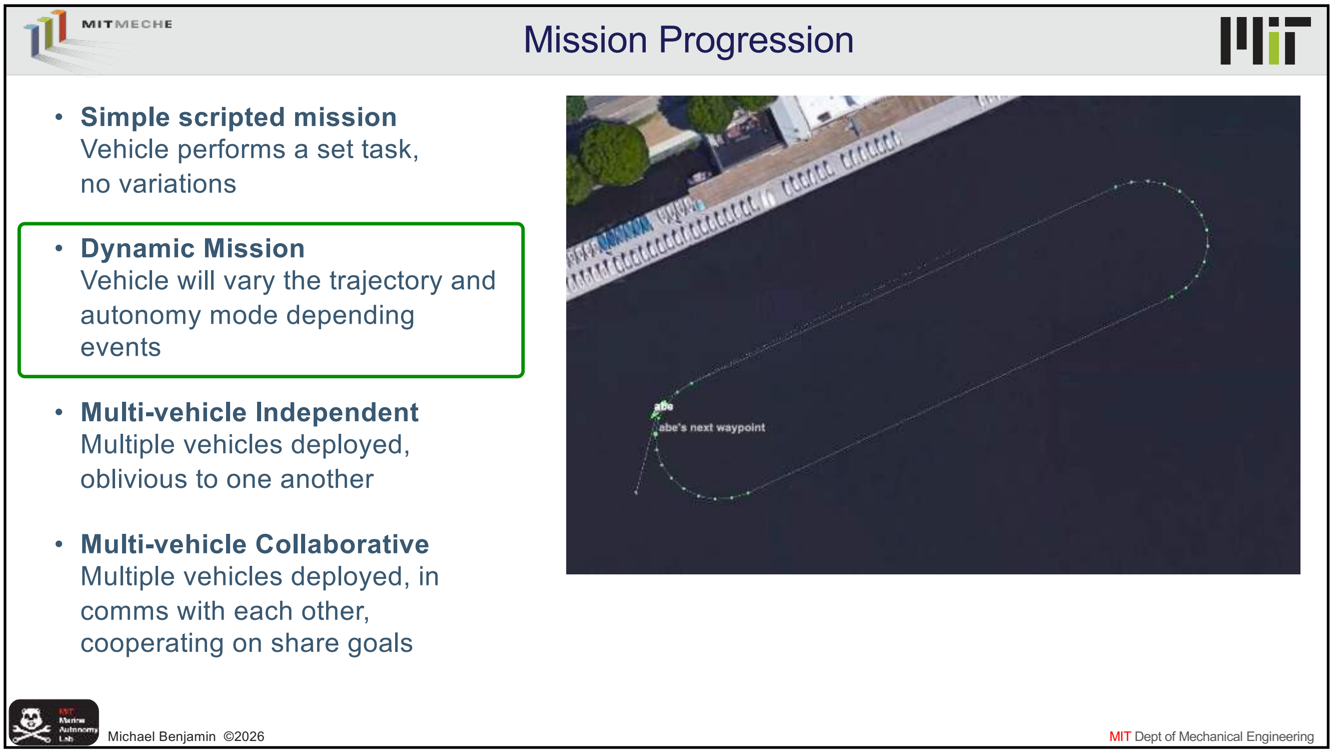
49

Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals



50



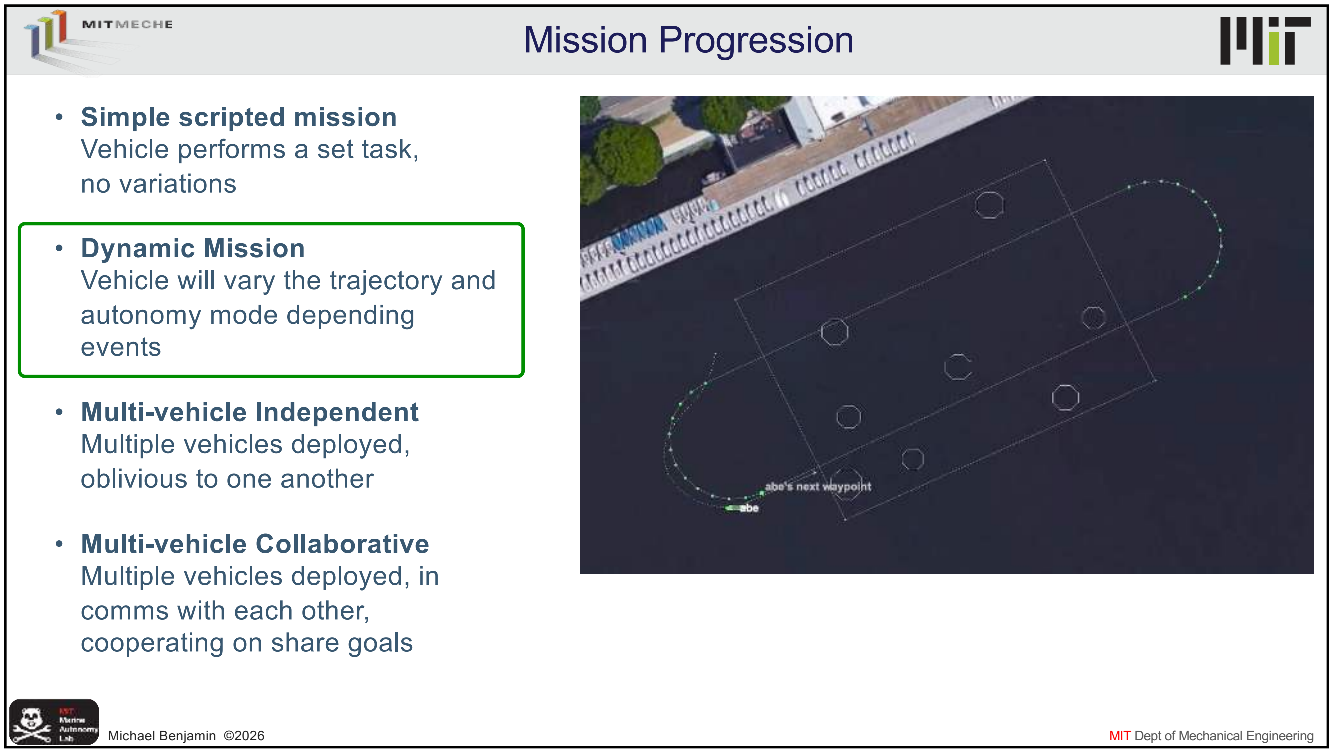
Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

51



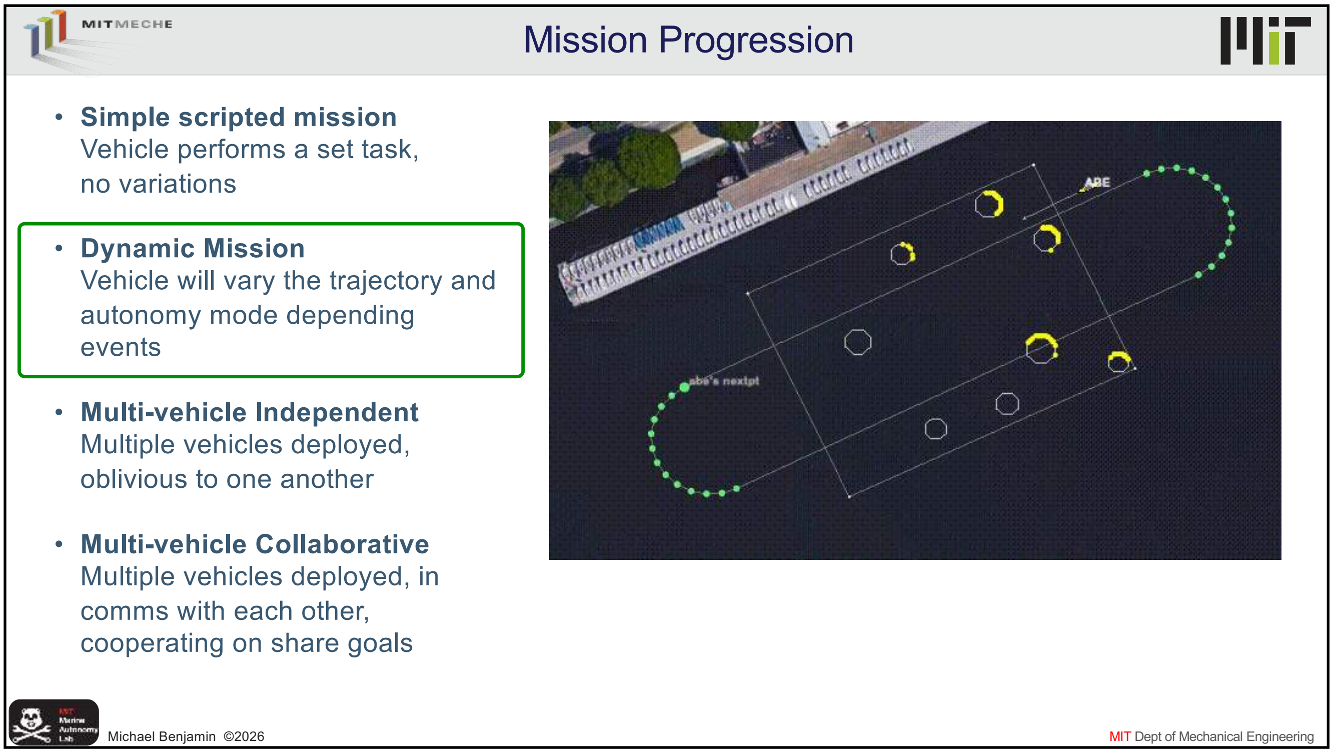
Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals

Michael Benjamin ©2026

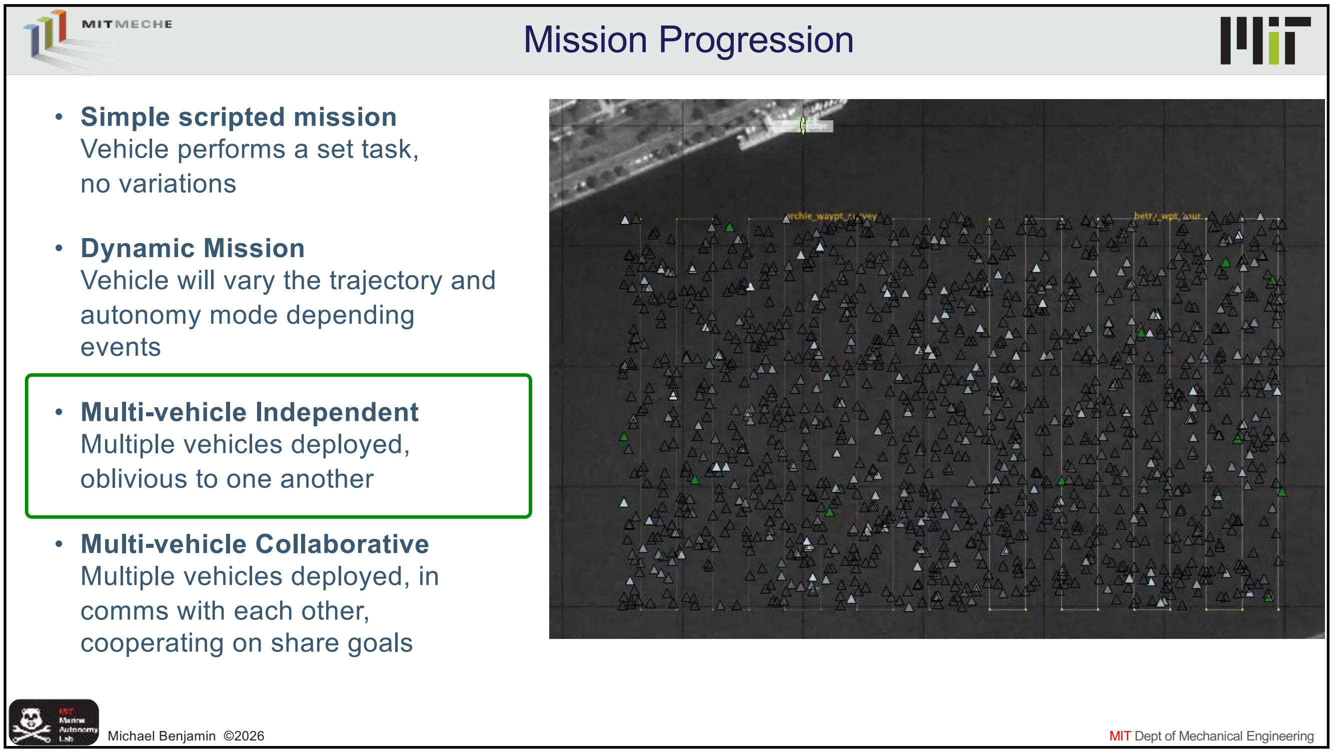
MIT Dept of Mechanical Engineering

52



Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals

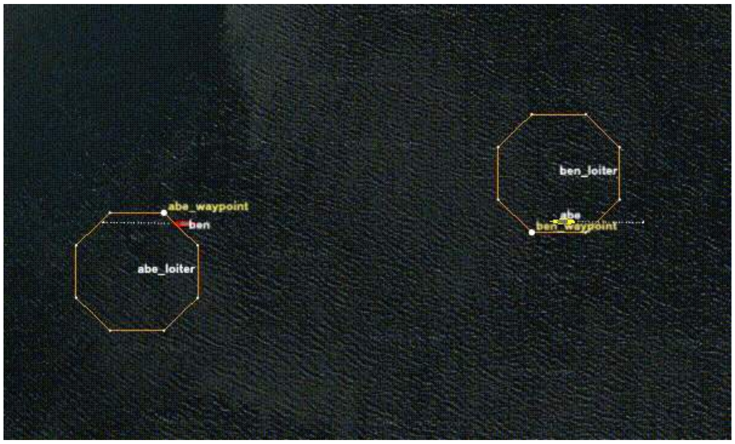


Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals

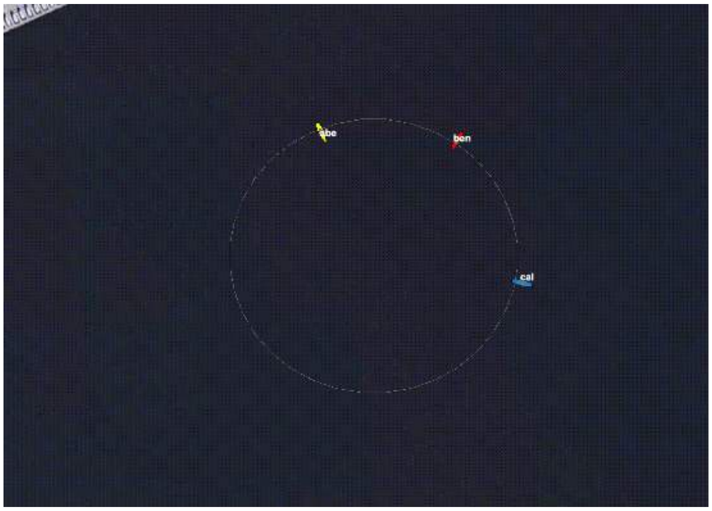
Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals



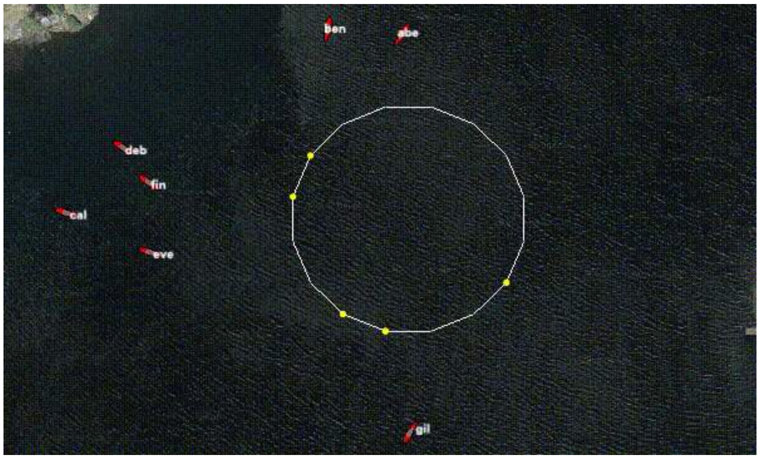
Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals



Mission Progression

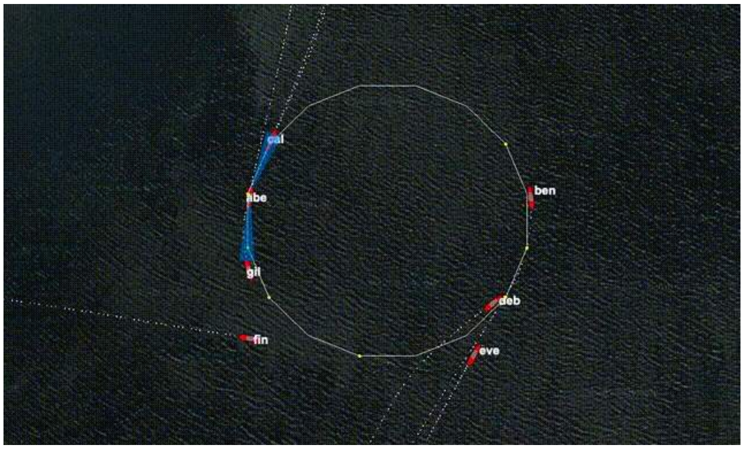
- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals



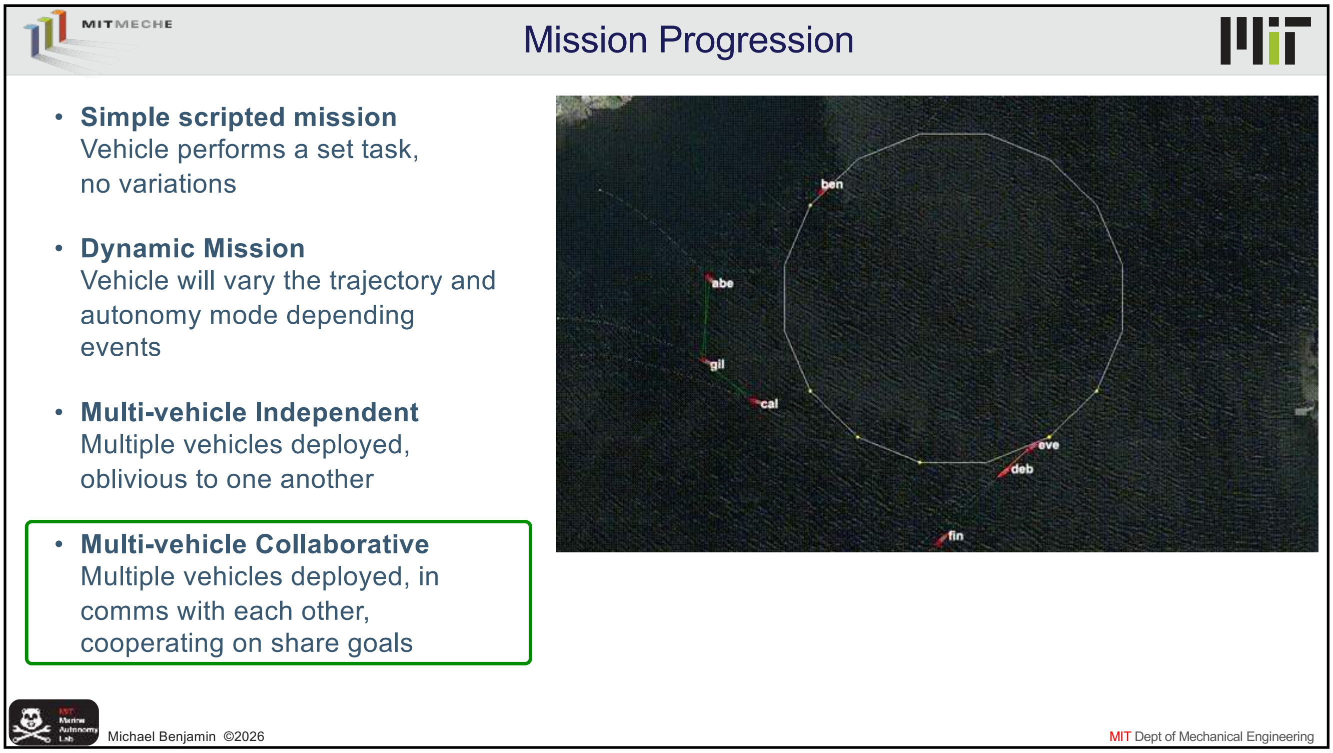
57

Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals



58



MITMECHE

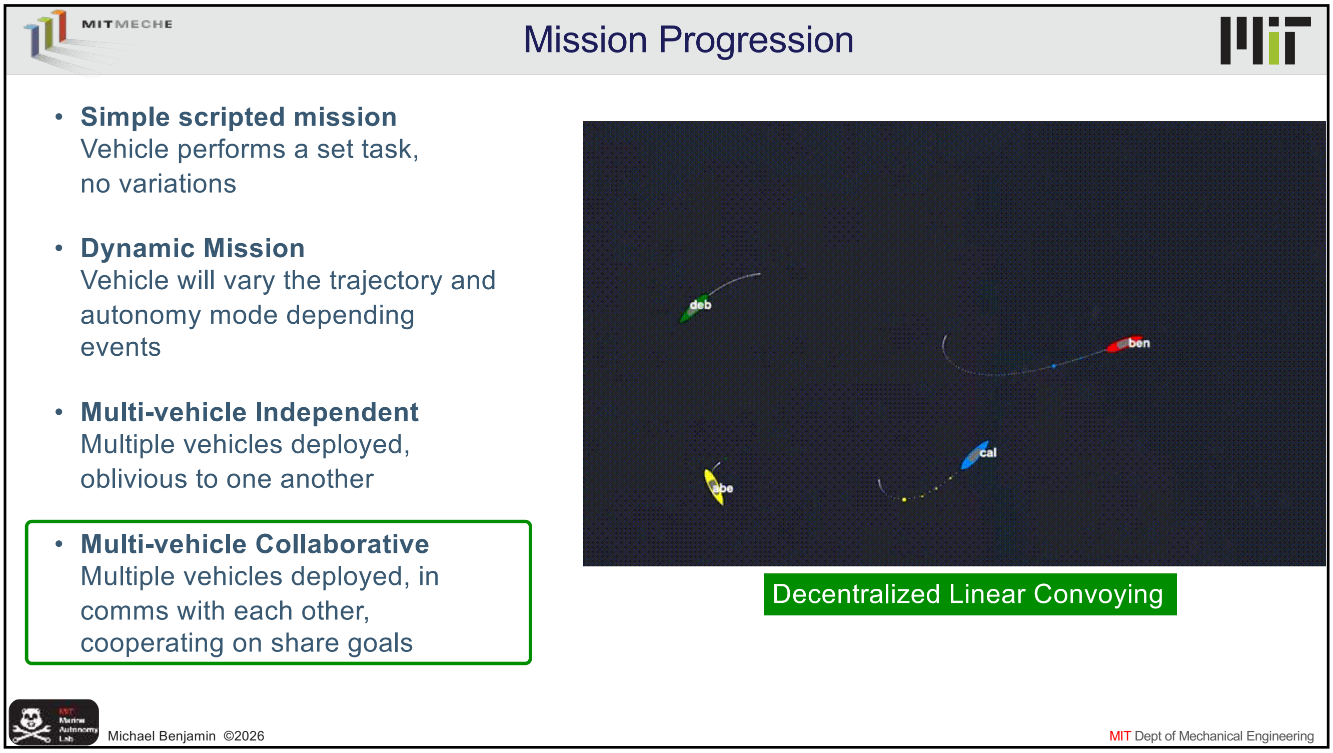
Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

59



MITMECHE

Mission Progression

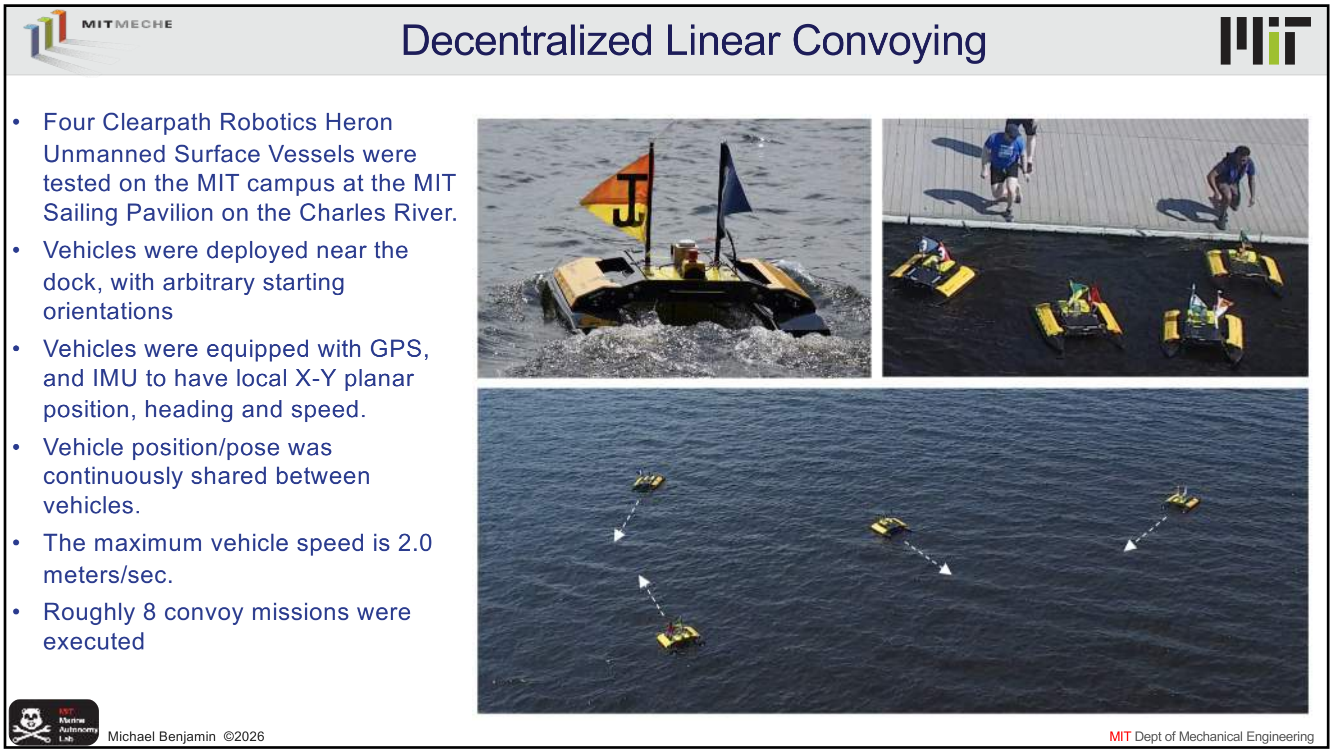
- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals

Decentralized Linear Convoying

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

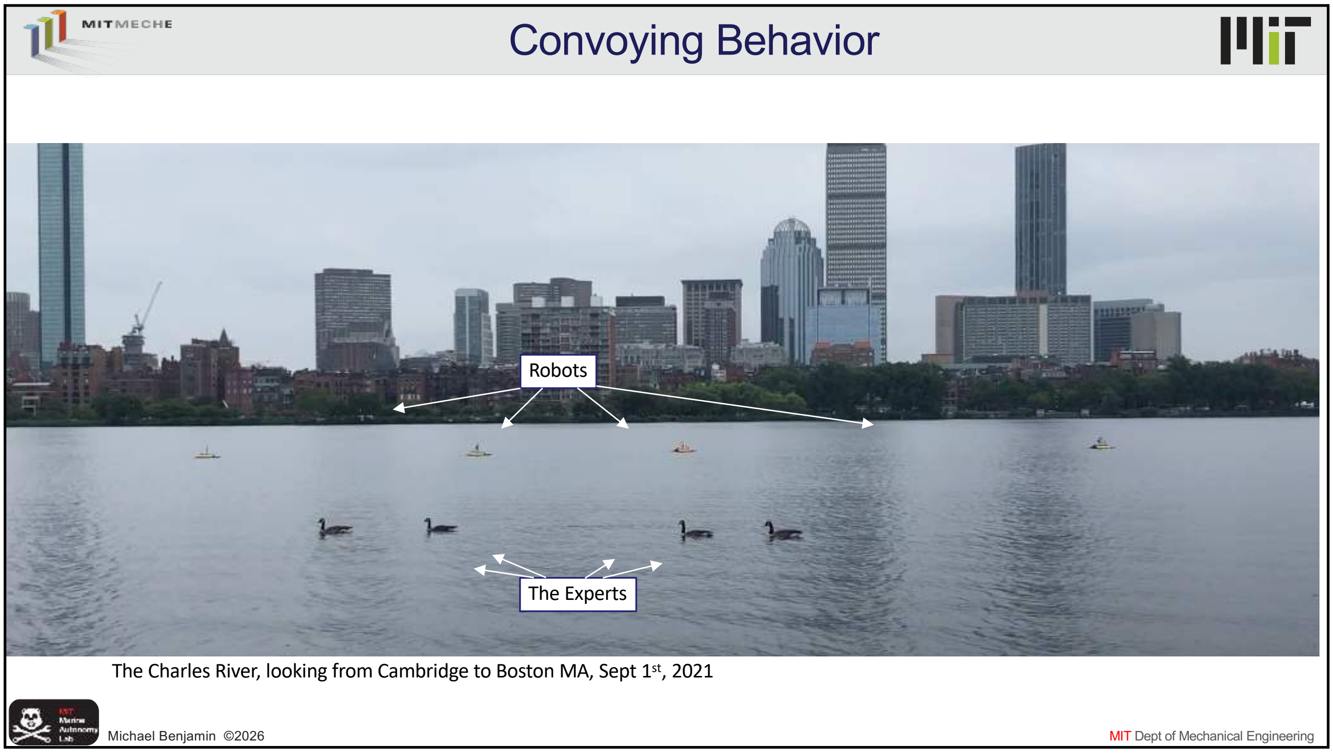
60



Decentralized Linear Convoying

- Four Clearpath Robotics Heron Unmanned Surface Vessels were tested on the MIT campus at the MIT Sailing Pavilion on the Charles River.
- Vehicles were deployed near the dock, with arbitrary starting orientations
- Vehicles were equipped with GPS, and IMU to have local X-Y planar position, heading and speed.
- Vehicle position/pose was continuously shared between vehicles.
- The maximum vehicle speed is 2.0 meters/sec.
- Roughly 8 convoy missions were executed

61



Convoying Behavior

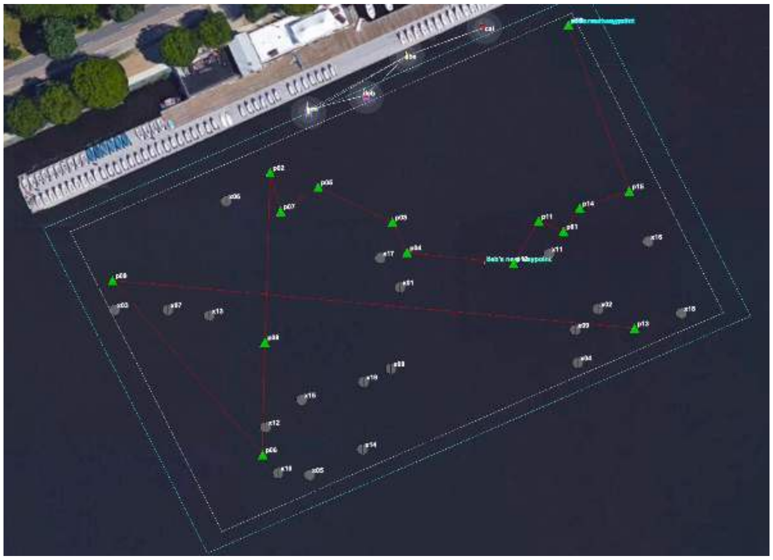
The Charles River, looking from Cambridge to Boston MA, Sept 1st, 2021

62

DARPA Sea Train Linear Convoying

Mission Progression

- **Simple scripted mission**
  Vehicle performs a set task, no variations
- **Dynamic Mission**
  Vehicle will vary the trajectory and autonomy mode depending events
- **Multi-vehicle Independent**
  Multiple vehicles deployed, oblivious to one another
- **Multi-vehicle Collaborative**
  Multiple vehicles deployed, in comms with each other, cooperating on share goals



MITMECHE

Run-Time Structure Progression

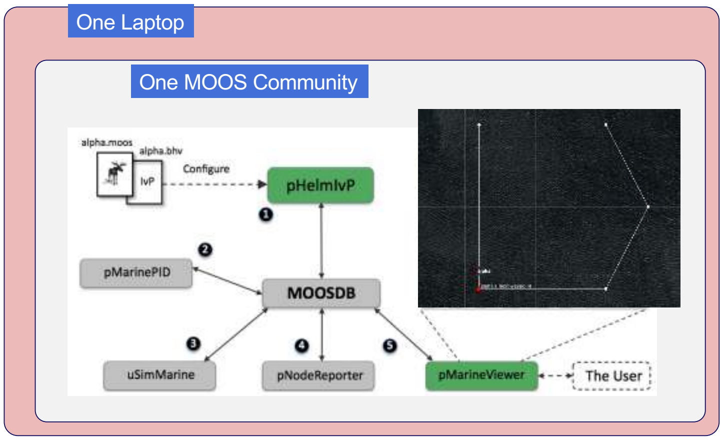
- **Single MOOS community**
  All robot and mission control in a single community
- **Multiple MOOS communities**
  Each vehicle has its own MOOS community and dedicated shoreside community on one laptop
- **Distributed MOOS communities**
  Simulations distributed over many laptops on the network
- **Deployed multi-vehicle ops**
  Multiple vehicles deployed, on the water, connected to shoreside command-and-control

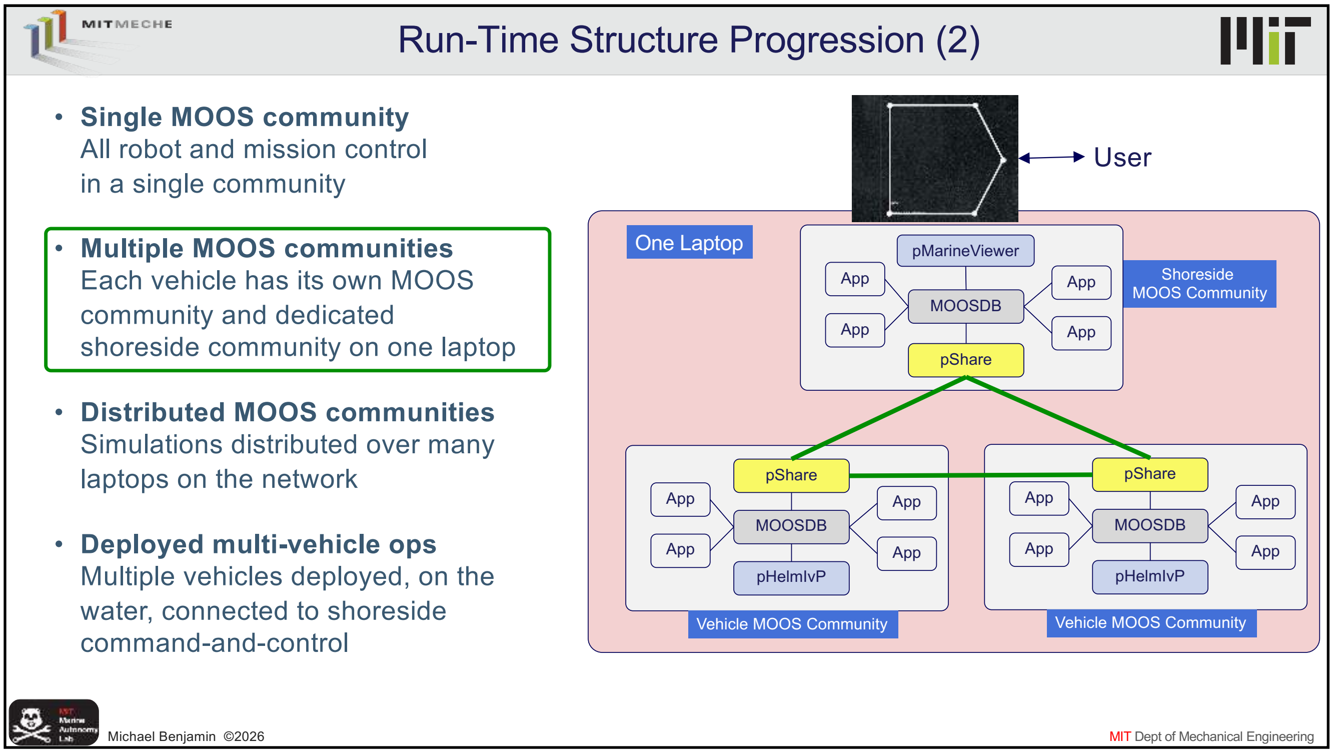
Over the semester, the way missions are run will evolve from a single MOOS community simulated on a single to multiple coordinated MOOS communities over several laptops or robots.

MITMECHE

Run-Time Structure Progression (1)

- **Single MOOS community**
  All robot and mission control in a single community
- **Multiple MOOS communities**
  Each vehicle has its own MOOS community and dedicated shoreside community on one laptop
- **Distributed MOOS communities**
  Simulations distributed over many laptops on the network
- **Deployed multi-vehicle ops**
  Multiple vehicles deployed, on the water, connected to shoreside command-and-control





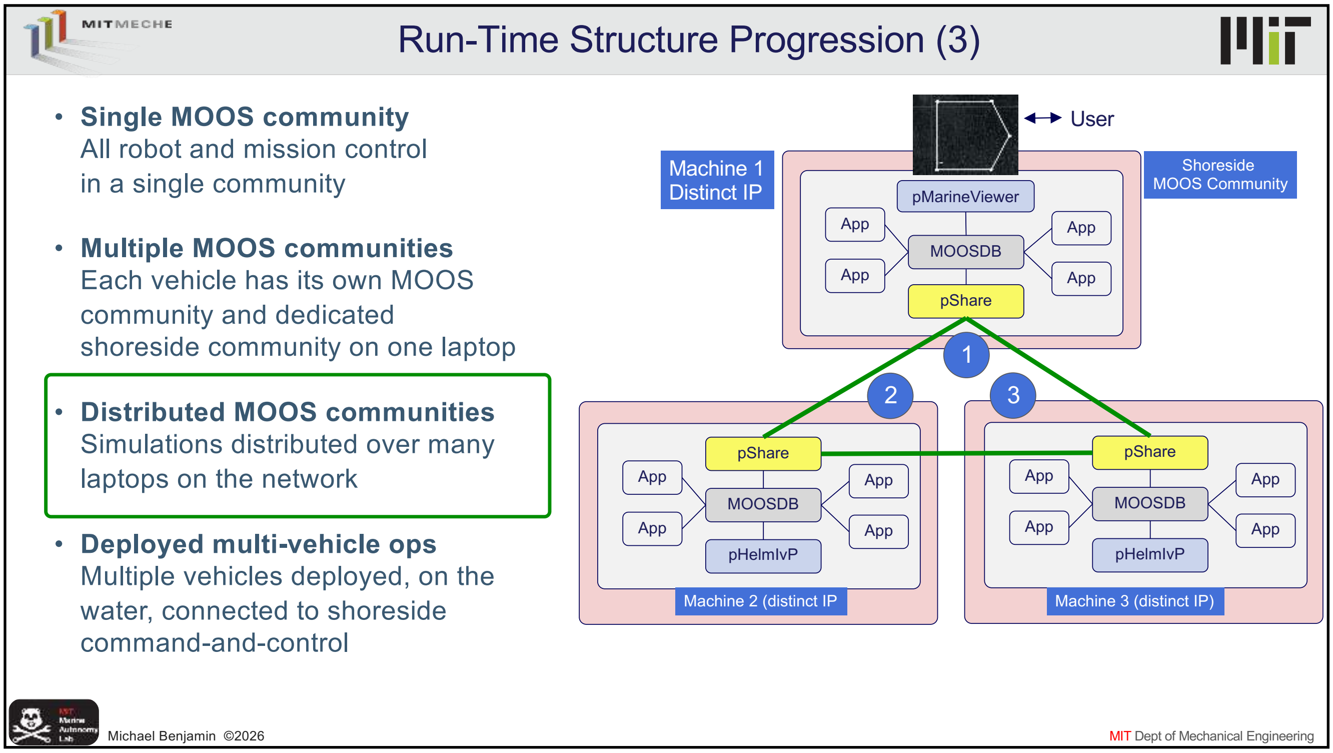
Run-Time Structure Progression (2)

- **Single MOOS community**
  All robot and mission control in a single community
- **Multiple MOOS communities**
  Each vehicle has its own MOOS community and dedicated shoreside community on one laptop
- **Distributed MOOS communities**
  Simulations distributed over many laptops on the network
- **Deployed multi-vehicle ops**
  Multiple vehicles deployed, on the water, connected to shoreside command-and-control

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

67



Run-Time Structure Progression (3)

- **Single MOOS community**
  All robot and mission control in a single community
- **Multiple MOOS communities**
  Each vehicle has its own MOOS community and dedicated shoreside community on one laptop
- **Distributed MOOS communities**
  Simulations distributed over many laptops on the network
- **Deployed multi-vehicle ops**
  Multiple vehicles deployed, on the water, connected to shoreside command-and-control

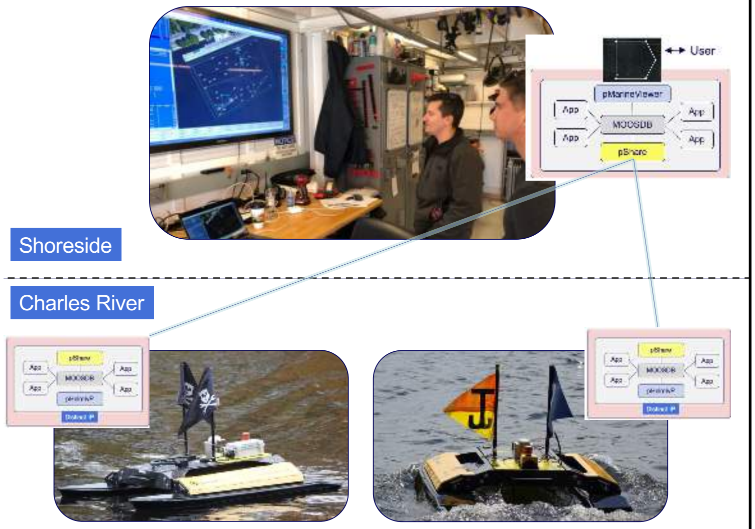
Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

68

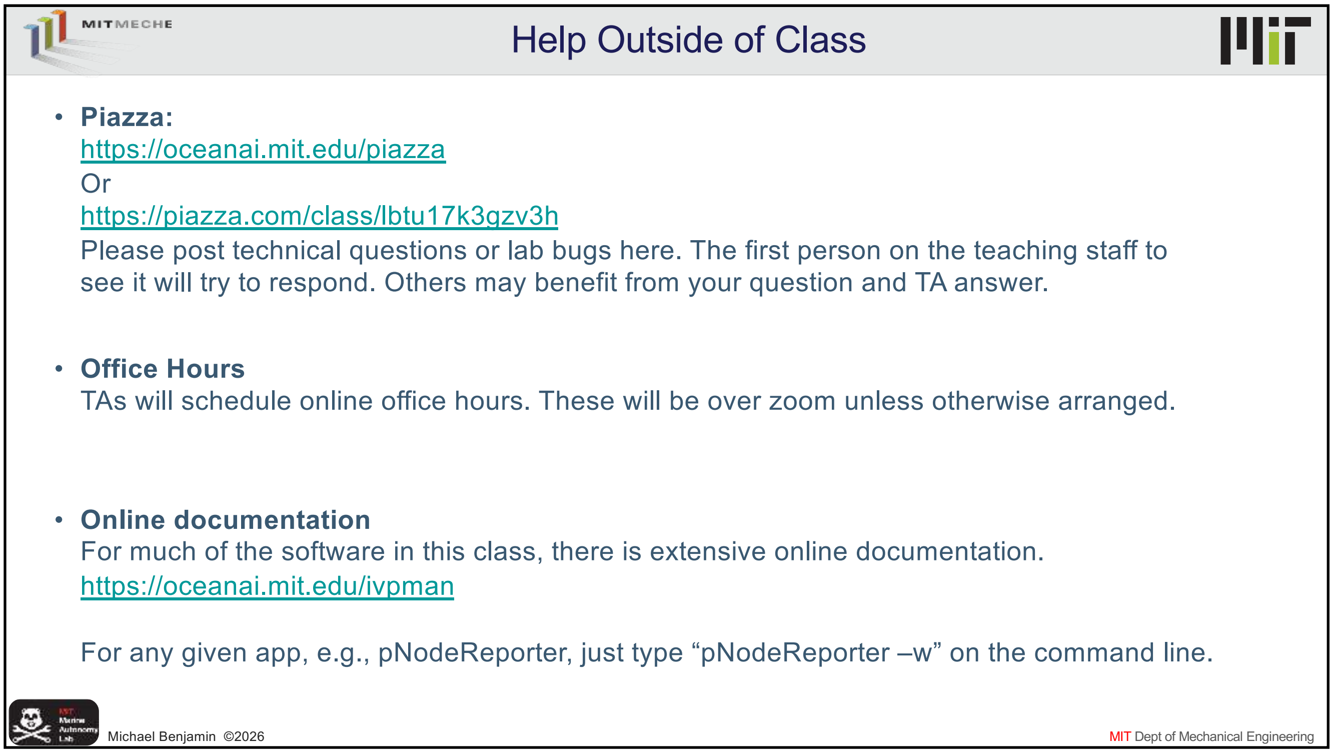
Run-Time Structure Progression (4)

- **Single MOOS community**
  All robot and mission control in a single community
- **Multiple MOOS communities**
  Each vehicle has its own MOOS community and dedicated shoreside community on one laptop
- **Distributed MOOS communities**
  Simulations distributed over many laptops on the network
- **Deployed multi-vehicle ops**
  Multiple vehicles deployed, on the water, connected to shoreside command-and-control



2.680 Learning Resources

- **Course Website:**
  https://oceanai.mit.edu/2.680
- **Course Schedule**
  https://oceanai.mit.edu/2.680/schedule
  Note the schedule is updated as the semester progresses.
  The current content/links for future lectures/labs are best projections.
  The lecture and lab content for future lectures/labs may show the previous year until replaced.
- **Course Lectures**
  https://oceanai.mit.edu/2.680/lectures
  Lectures are online (usually posted day-of). Printed handouts usually also brought to class
- **Course Labs**
  https://oceanai.mit.edu/2.680/labs
  Labs are online (usually posted day-of). Printed handouts usually also brought to class



Help Outside of Class

- **Piazza:**
  https://oceanai.mit.edu/piazza
  Or
  https://piazza.com/class/lbtu17k3gzv3h
  Please post technical questions or lab bugs here. The first person on the teaching staff to see it will try to respond. Others may benefit from your question and TA answer.

- **Office Hours**
  TAs will schedule online office hours. These will be over zoom unless otherwise arranged.

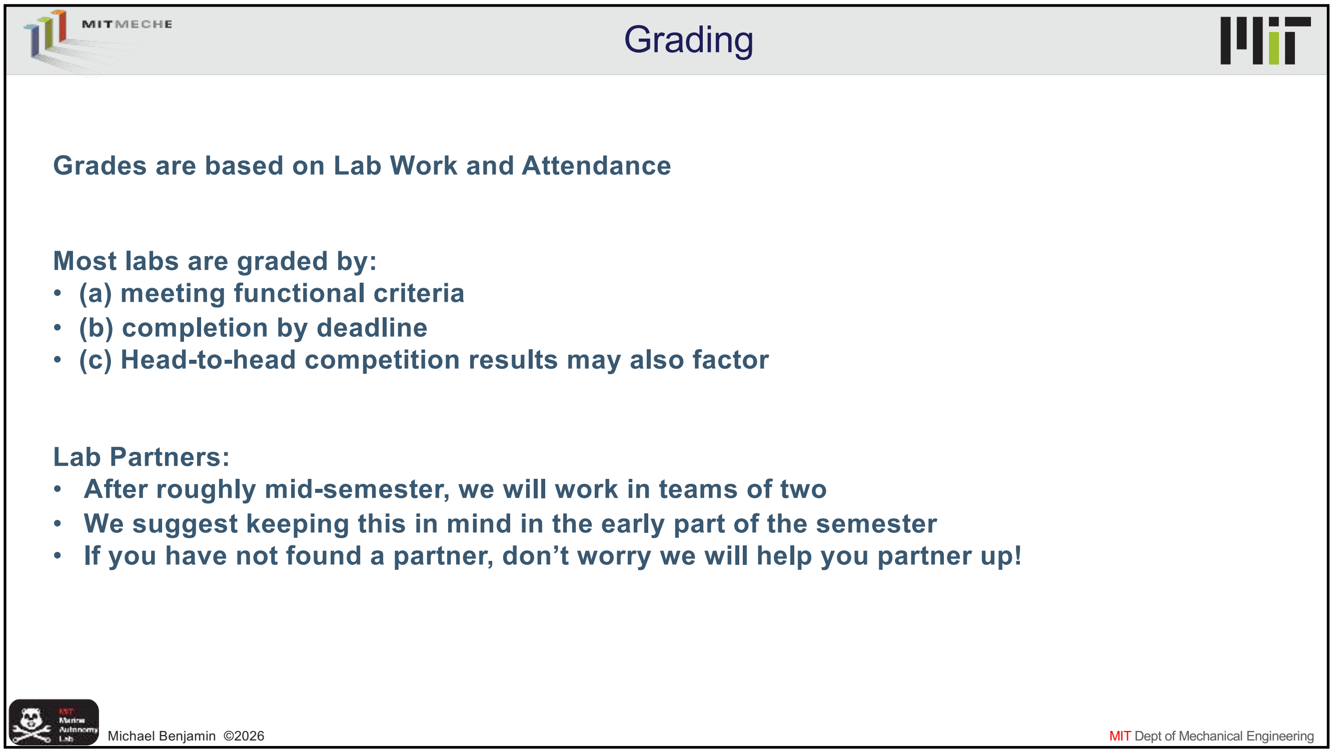
- **Online documentation**
  For much of the software in this class, there is extensive online documentation.
  https://oceanai.mit.edu/ivpman

  For any given app, e.g., pNodeReporter, just type “pNodeReporter –w” on the command line.

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

71



Grading

Grades are based on Lab Work and Attendance

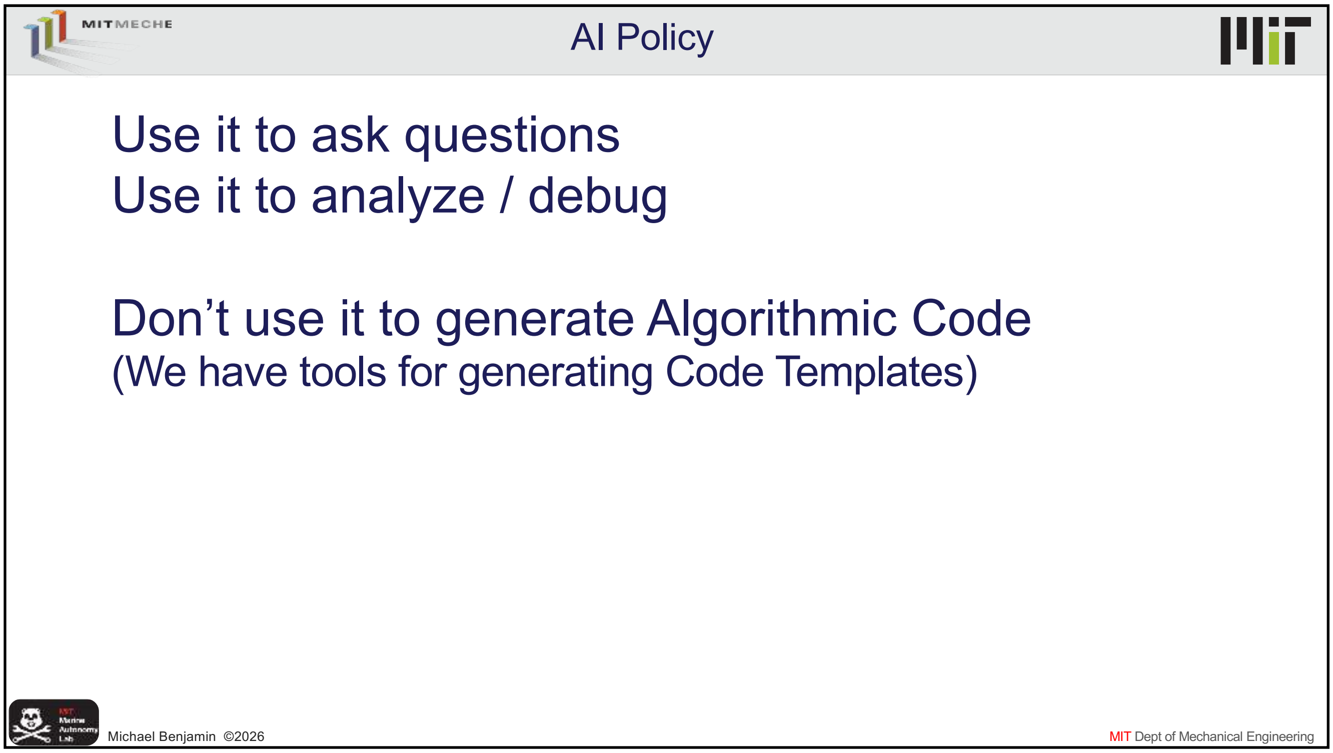
Most labs are graded by:
- **(a) meeting functional criteria**
- **(b) completion by deadline**
- **(c) Head-to-head competition results may also factor**

Lab Partners:
- **After roughly mid-semester, we will work in teams of two**
- **We suggest keeping this in mind in the early part of the semester**
- **If you have not found a partner, don’t worry we will help you partner up!**

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

72



AI Policy

Use it to ask questions
Use it to analyze / debug

Don't use it to generate Algorithmic Code
(We have tools for generating Code Templates)

Michael Benjamin ©2026

MIT Dept of Mechanical Engineering

73