U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT COMMAND – ARMY RESEARCH LABORATORY

Tactical Behaviors for Autonomous Maneuver Collaborative Research Program (TBAM-CRP)

John Rogers, Ph.D. – TBAM-CRP Collaborative Alliance Manager
Derrik Asher, Ph.D. – Research Psychologist / Multi-Agent Collaboration Researcher
Luke Frerichs – AI for Mobility and Maneuver Program Officer
David Baran – Chief, Intelligence for Robotics Branch

DEVCOM Army Research Laboratory, Computational and Information Sciences Directorate
29 April 2022

APPROVED FOR PUBLIC RELEASE

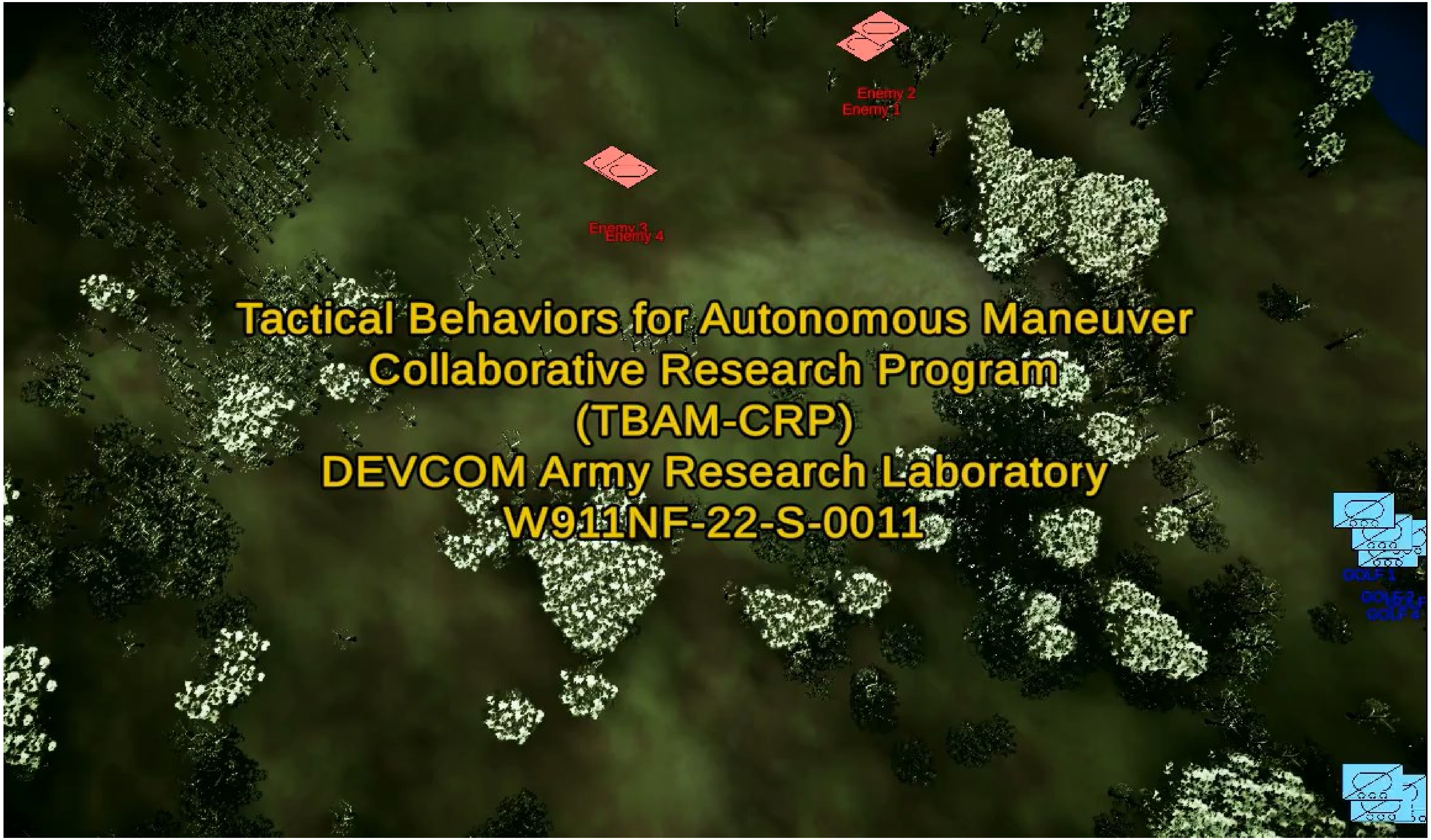
AGENDA

- **Program motivation**
- **Army doctrine background**
- **Desired outcomes/ metrics**
- **GFE Software**
  - ARL Autonomy Stack
  - ARL Unity Simulator
  - MMAE (MITRE Multi-Agent Environment) Multi-Agent Reinforcement Learning (MARL) Unity Simulator
- **Some ARL research relevant to this call**
- **Questions**

AGENDA

- **Program motivation**
- Army doctrine background
- Desired outcomes/ metrics
- GFE Software
- Some ARL research relevant to this call
- Questions

MOTIVATING VISION



AGENDA

- Program motivation
- **Army doctrine background**
- Desired outcomes/ metrics
- GFE Software
- Some ARL research relevant to this call
- Questions

RECONNAISSANCE

- **Reconnaissance answers the Commander's Critical Information Requests(CCIRs).**
  - Information that the commander needs to make decisions, understand the threat and other aspects of the operational environment
- **Reconnaissance identifies:**
  - terrain characteristics
  - enemy and friendly obstacles to movement
  - the disposition of enemy forces and civilian population so the commander can make decisions to shape the battlefield.
- **Reconnaissance prior to unit movements and occupation of assembly areas is critical to protecting the force and preserving combat power.**

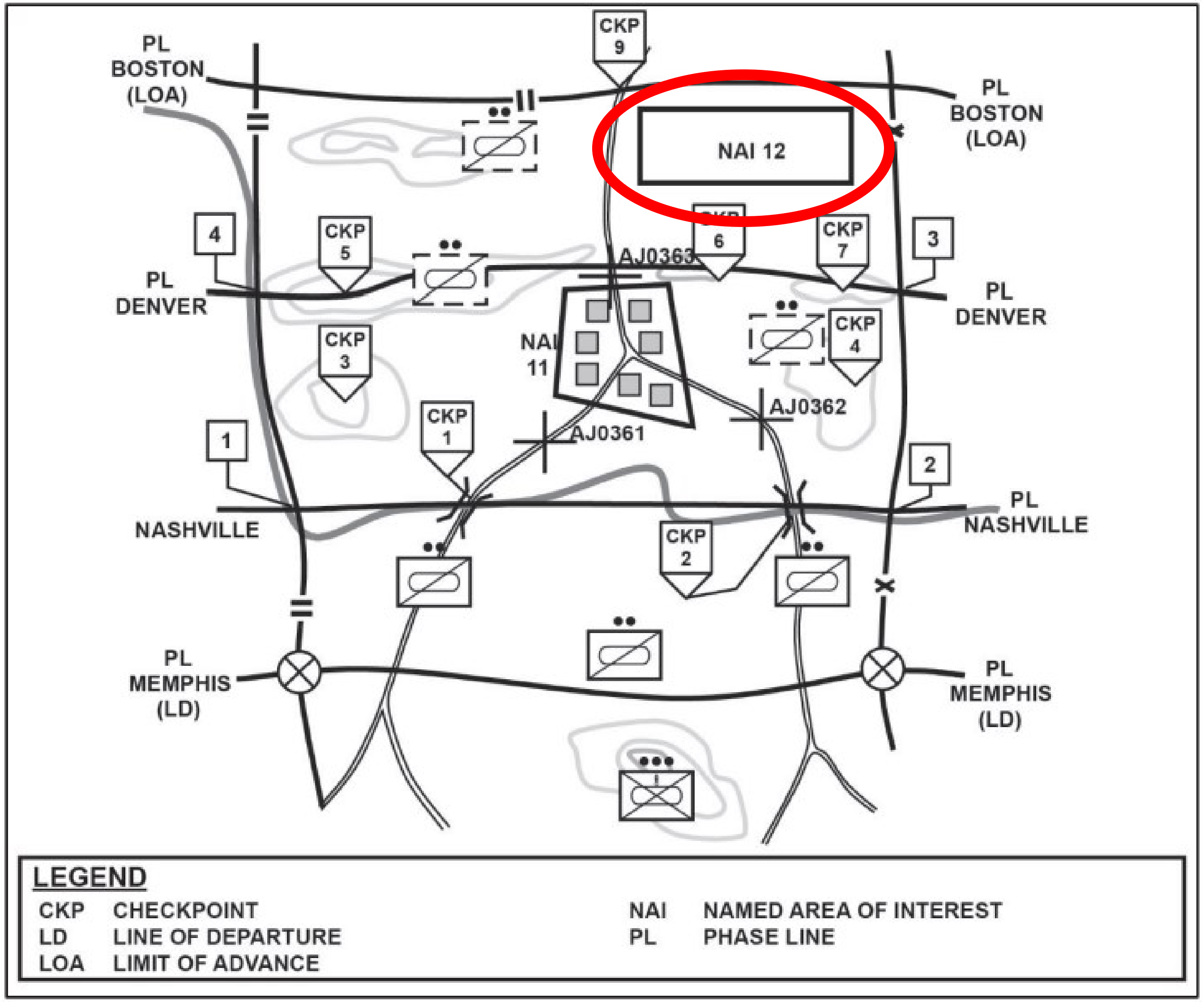
- **Reconnaissance keeps the force free from contact as long as possible so that it can concentrate on its decisive operation**

Additional Details:
ATP 3-90.5: Combined Arms Battalion:
https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN32974-ATP_3-90.5-000-WEB-1.pdf
ATP 3-20.98: Scout Platoon:
https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN19886-ATP_3-20.98-000-WEB-1.pdf

RECONNAISSANCE

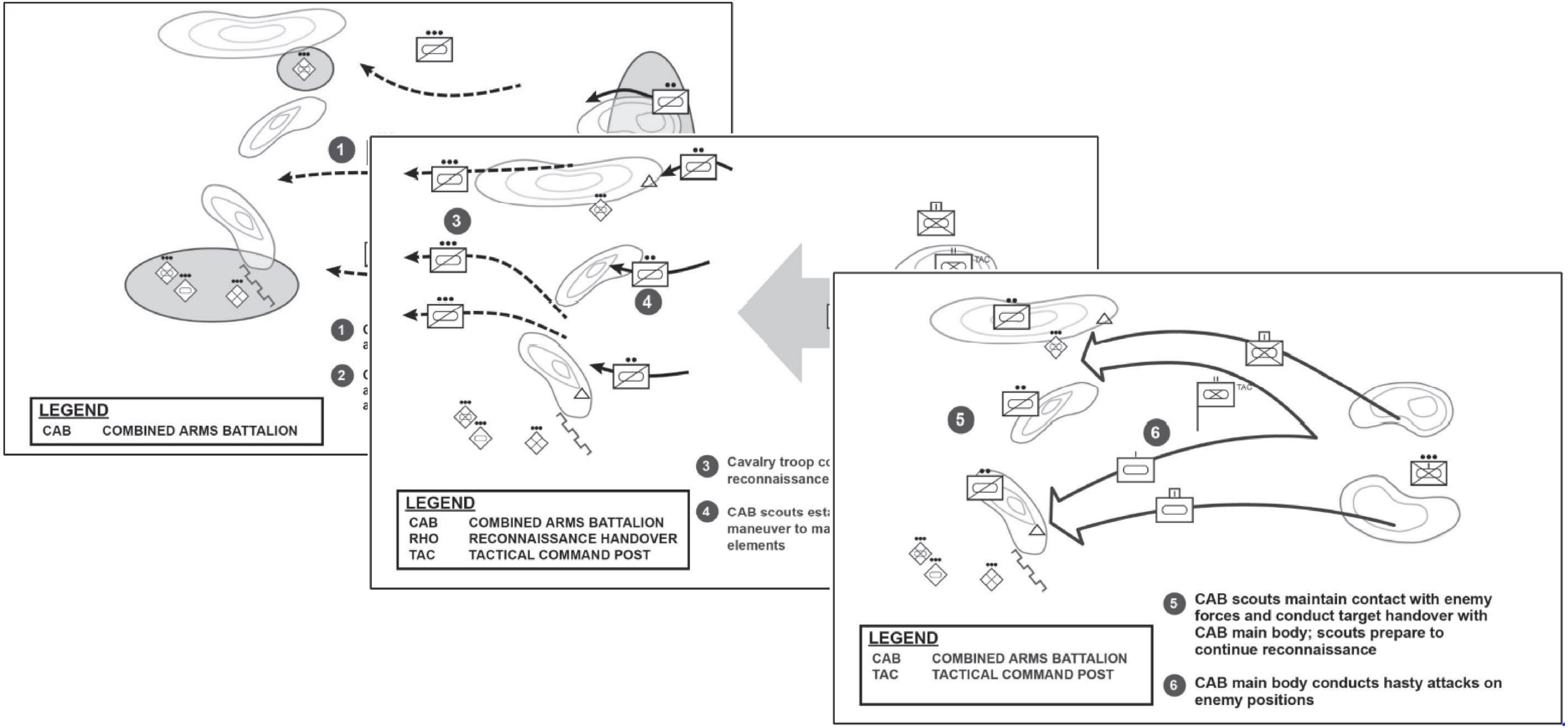
- **Fundamentals of reconnaissance**
  - Ensure continuous reconnaissance.
  - Retain freedom of maneuver.
  - Gain and maintain enemy contact.
  - Develop the situation
  - Orient on the reconnaissance objective.
  - Report all required information rapidly and accurately.

- **Types of reconnaissance**
  - Zone reconnaissance.
    - obtain detailed information on all routes, obstacles, terrain, and enemy forces within a zone defined by boundaries
  - Area reconnaissance.
    - provides information about a specified area such as a town, ridge, woods, or other feature critical to operations
  - Route reconnaissance.
    - obtain detailed information of a specified route and all terrain which the enemy could influence movement along that route



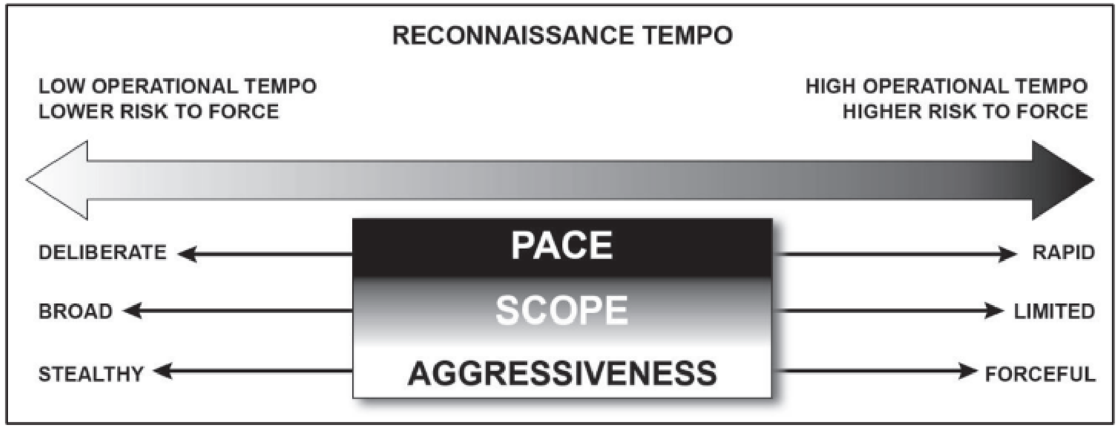
Example zone recon mission (NAI 12) with phase lines and established friendly checkpoints

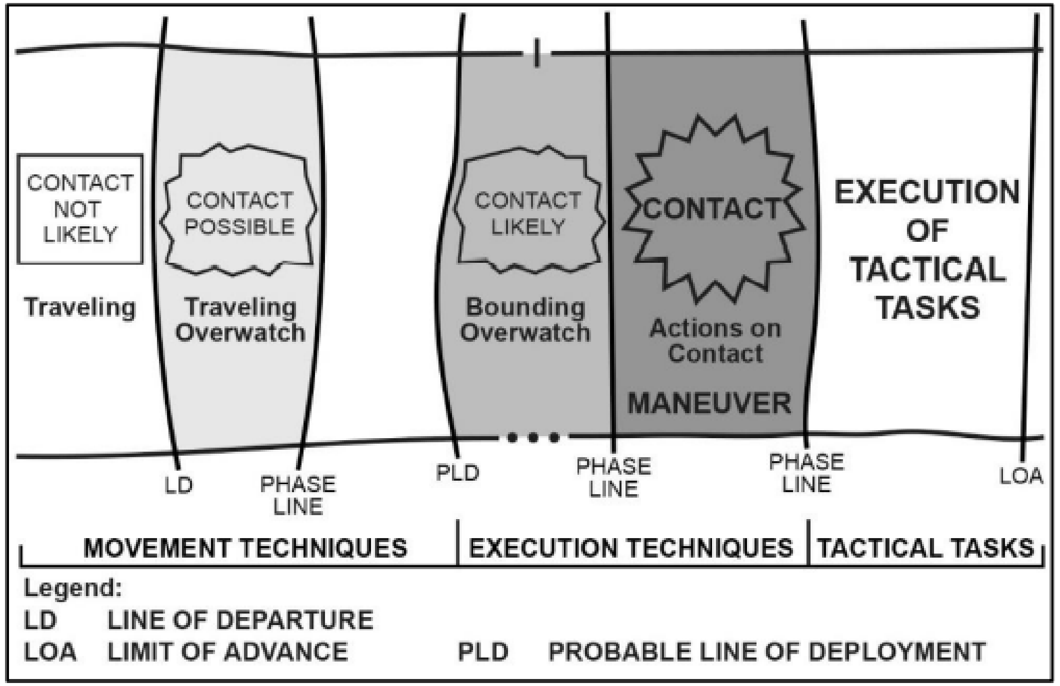
RECON ELEMENTS IN COMBINED ARMS BATTALION (EXAMPLE)



TACTICAL GROUND MANEUVER - PURPOSES

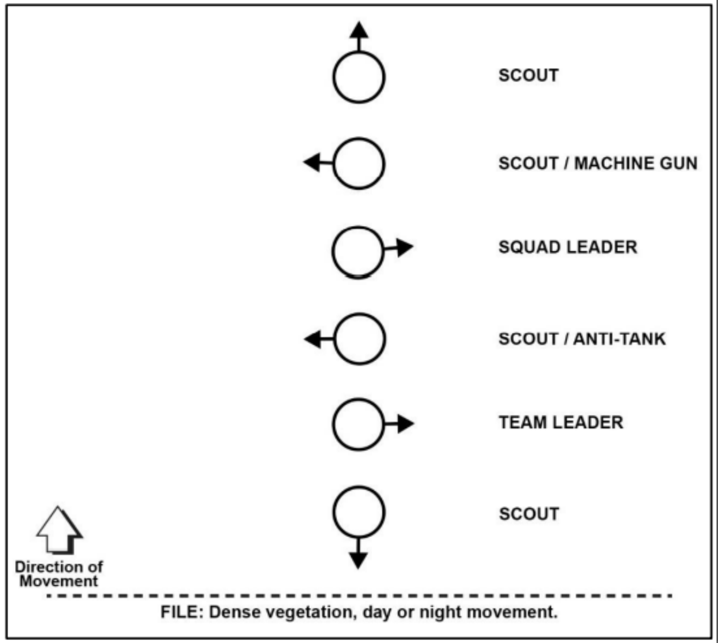
- Minimize exposure
- Maintain freedom of movement
- Maximize available tactical options
- React successfully to contact
- Find and observe threats without compromise



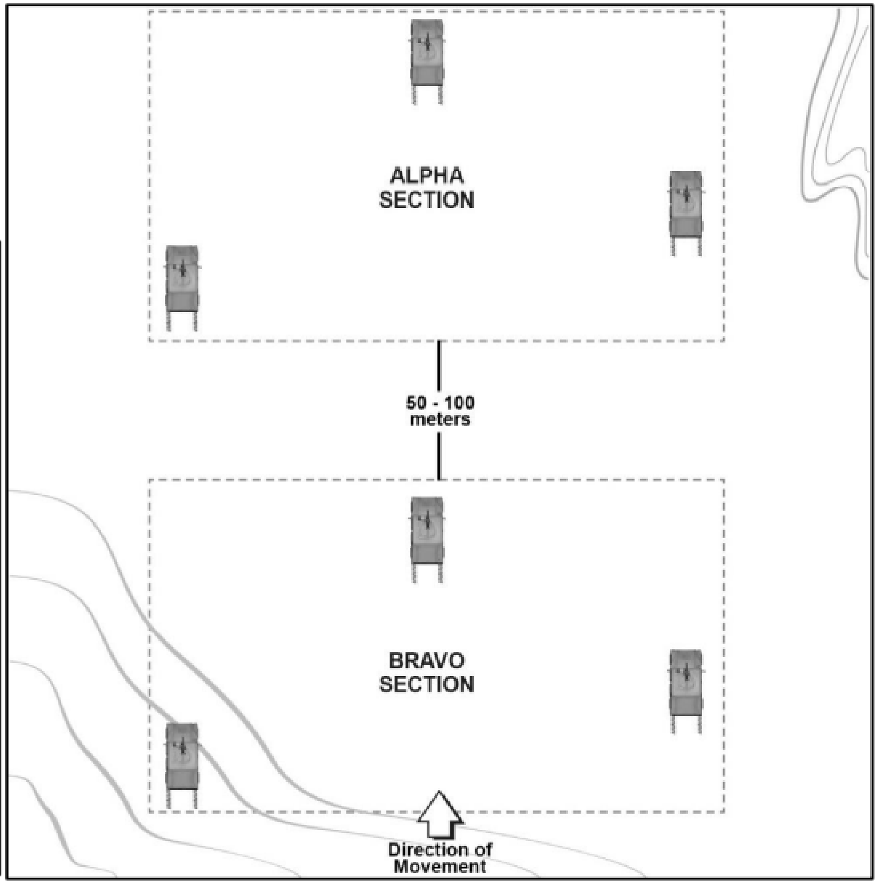


Movement Techniques	***When Normally Used***	***Characteristics***			
		Control	***Dispersion***	***Speed***	***Security***
Traveling	Contact not likely	More	Less	Fastest	Least
Traveling overwatch	Contact possible	Less	More	Slower	More
Bounding overwatch	Contact expected	Most	Most	Slowest	Most

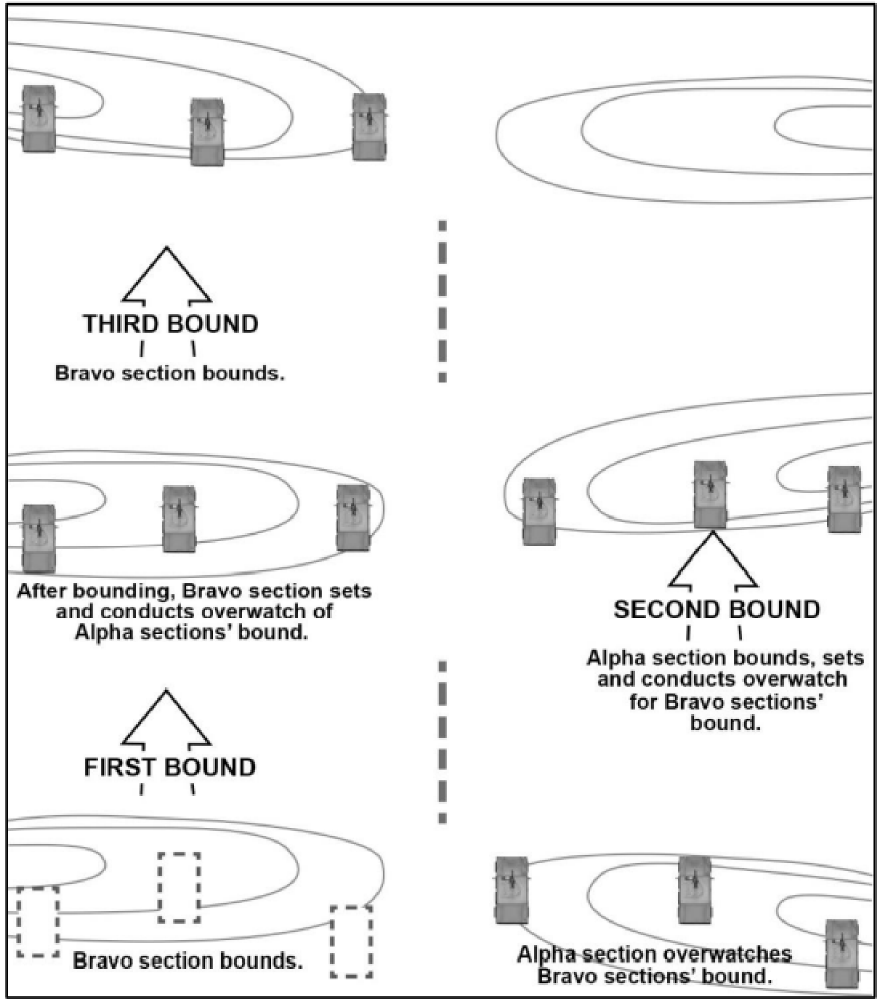
EXAMPLES OF TACTICAL MOVEMENT TECHNIQUES



Traveling



Traveling Overwatch



Bounding Overwatch

AGENDA

- Program motivation
- Army doctrine background
- **Desired outcomes/ metrics**
- GFE Software
- Some ARL research relevant to this call
- Questions

DESIRED OUTCOMES

- **Learn generalizable multi-robot coordinated maneuvers**
  - Applies tactical maneuver paradigms to terrain/cover in simulated environments for "squad scale" teams (4-10 assets)
  - Not specialized to each simulation scenario
  - Could be specialized to scenario high-level class e.g., "Forest"/ "Town" / "Rural"

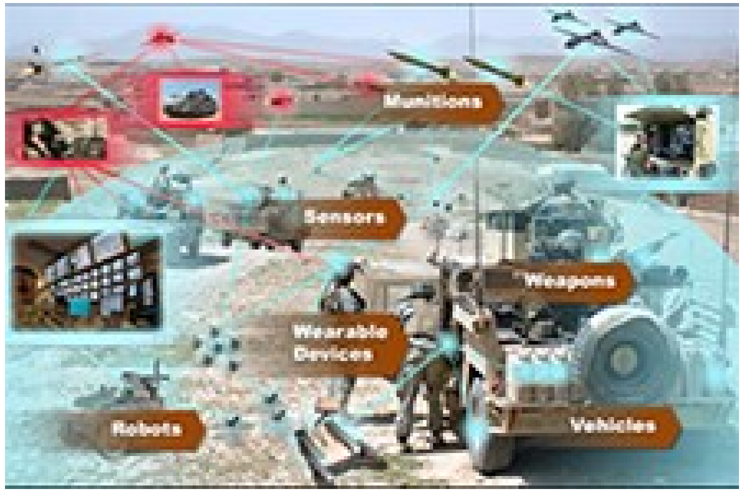
- **Learn novel as well as doctrinal maneuvers**

- **Focus on multi-agent autonomy/ control, but…**
  - Export measurements from simulator where needed ("semantic" camera, topography, object/ adversary detection)
  - Think about realism
    - incorporate realistic noise/ error/ range/ FOV where possible when adding simulator (virtual) sensors
    - Distributed algorithms / not relying on centralized single points of failure would be preferred

CONNECTIONS TO OTHER PROGRAMS

- **Leverage existing research in other ARL programs: DCIST, IoBT, SARA, others**
- **Work closely with ARL technical POC's embedded in internal essential research programs (AIMM, HAT, EOT, VICTOR, etc.)**
  - We will help make connections to technical POC's during the proposal review process
- **Leverage software components and interfaces in GFE ARL autonomy stack**
- **Evaluation on GFE simulators preferred but alternatives will be entertained**

Distributed and Collaborative Intelligent Systems and Technology – DEVCOM Army Research Laboratory



Internet of Battlefield Things – DEVCOM Army Research Laboratory

Scalable, Adaptive, and Resilient Autonomy (SARA) – DEVCOM Army Research Laboratory

SOME NOTIONAL METRICS

- **Properties of metrics for this program**
  - **Dependent on specific tasks / behaviors being utilized**
  - **Critical to describe progress evaluating maneuver "quality"**
  - **Metrics should be developed further by participants / collectively**
- **Some initial concepts for metrics**
  - Path comparison to a human baseline
  - “Red-team” observation/detection:
    - Duration observed out of cover (without counter-suppression/observation)
    - Effectiveness of the usage of cover (proportion of the vehicle exposed from cover)
    - Degree of exposure (e.g., on a ridgeline)
  - Tempo:
    - How long did it take to reach goal location?
    - Was the goal location reached on schedule?
- **Independent variables:**
  - Environmental complexity/ scale of maneuver/ heterogeny
  - Sparsity/Density
  - Diversity of tactics needed
  - Red-force disposition/tactics/strategy/ numbers

Test Site	Baseline Mean	Baseline Median	Baseline Best	IOC Mean	IOC Median	IOC Best
(P,Q)	5.201	4.457	3.422	**1.415**	1.362	1.160

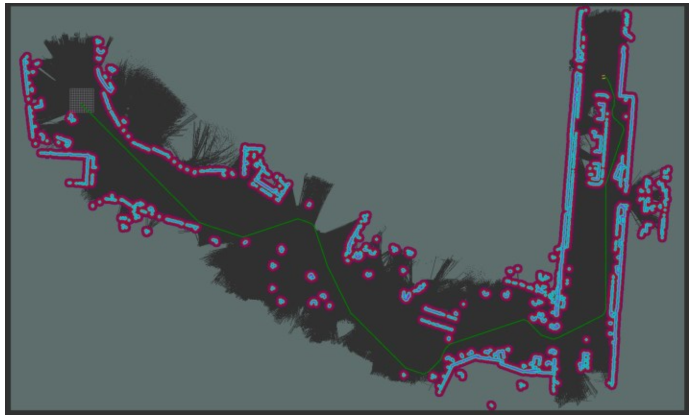
Path difference metrics: learned robot trajectories vs human-driven "ground truth" on "covert" navigation task

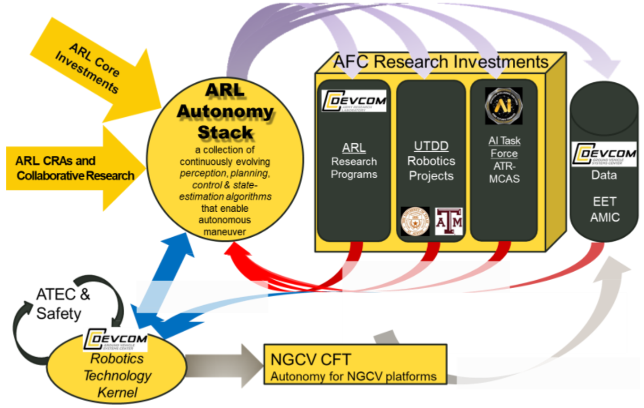
AGENDA

- Program motivation
- Army doctrine background
- Desired outcomes/ metrics
- **GFE Software**
  - ARL Autonomy Stack
  - ARL Unity Simulator
  - MMAE (MITRE Multi-Agent Environment) Multi-Agent Reinforcement Learning (MARL) Unity Simulator
- Some ARL research relevant to this call
- Questions

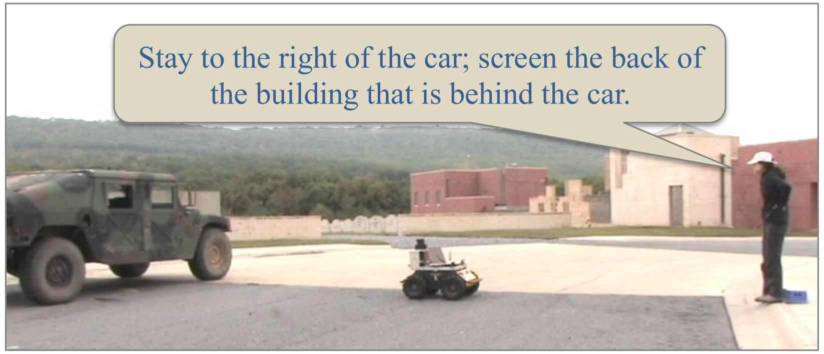
ARL GROUND AUTONOMY STACK IN ONE SLIDE

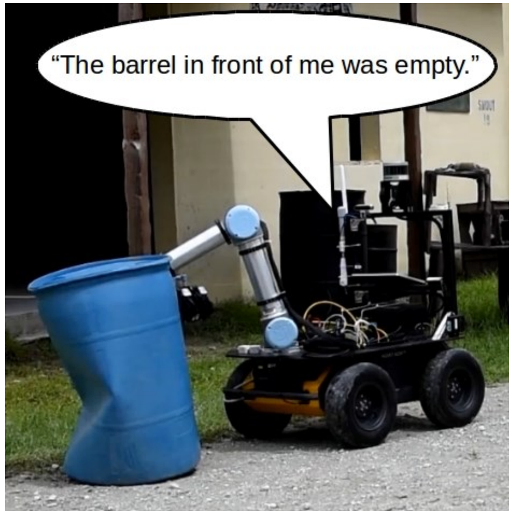
- **ARL Ground Autonomy Stack**
- **Focal point of ARL ground autonomy research**
  - ROS1 based – taking advantage of a flexible framework
  - Monolithic repository to encourage cross-functional development and collaboration with a focus on supporting reproducible research
  - Common simulation environment (Unity)
  - Feature branches documented in gitlab with merge request reviews for integration into master





STACK BACKGROUND

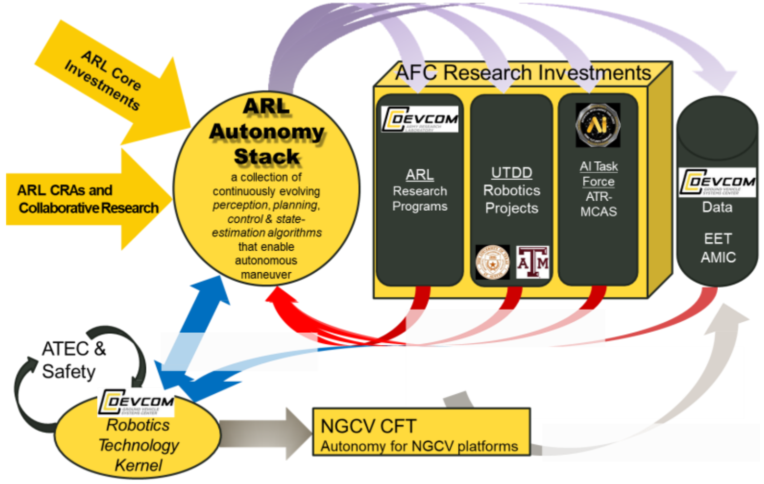




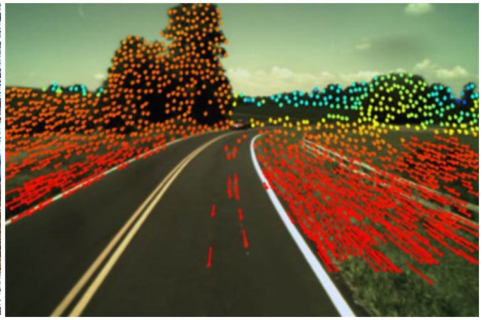
- **ARL has advanced numerous concepts of military use of small unmanned systems and robot teaming**
  - Programs: MAST, RCTA, DCIST
  - Locations: Off-road, urban, tunnels
- **Outcomes**
  - Semantic segmentation (HIM)
  - Navigation planning (SBPL)
  - SLAM (GTSAM, Omnimapper)
  - Language grounding (H2SL)
- **The stack represents work from many academic and industry collaborations**
  - Driven by ARL research consortiums

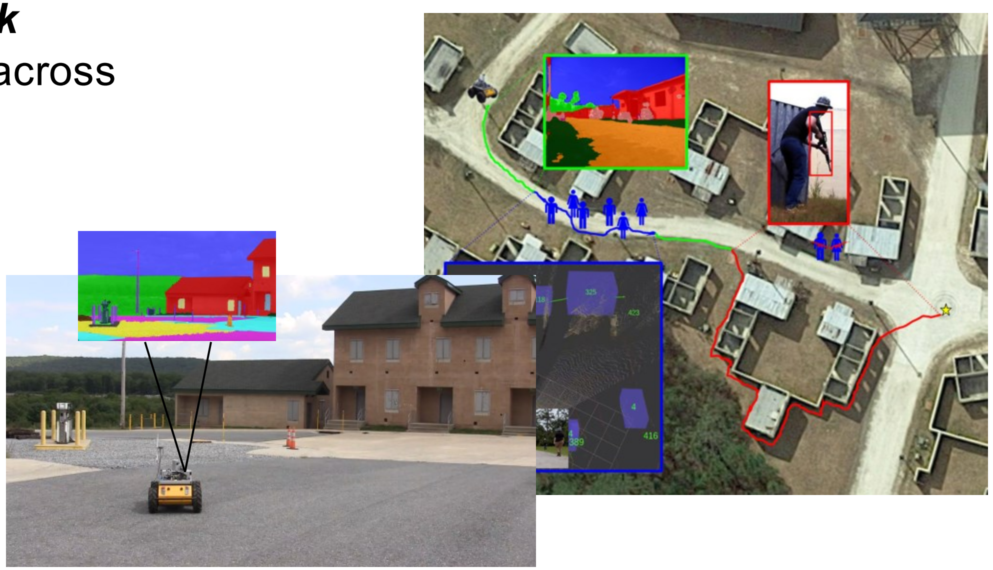
STACK BACKGROUND

- **ARL has a long history in ground autonomy**
  - Tapped pathfinding work from Europe in late 90s (Ernst Dickmanns)
  - ARL researchers were key drivers of ground autonomy work through the DEMO I – DEMO II - DEMO III series of programs
  - Demo III laid groundwork for first DARPA Grand Challenge
  - Gen-1 LIDAR was outgrowth of first ARL Robotics Consortium
  - Focus shifted to small vehicles as self-driving auto industry started



- **ARL continues to shape ground autonomy**
  - RCTA2 drove the state-of-the-art in the use of semantics for navigation
  - Early post-RCTA work evolved into current ***ARL Autonomy Stack***
  - ARL Autonomy Stack Currently actively in use and development across multiple ARL/DOD programs

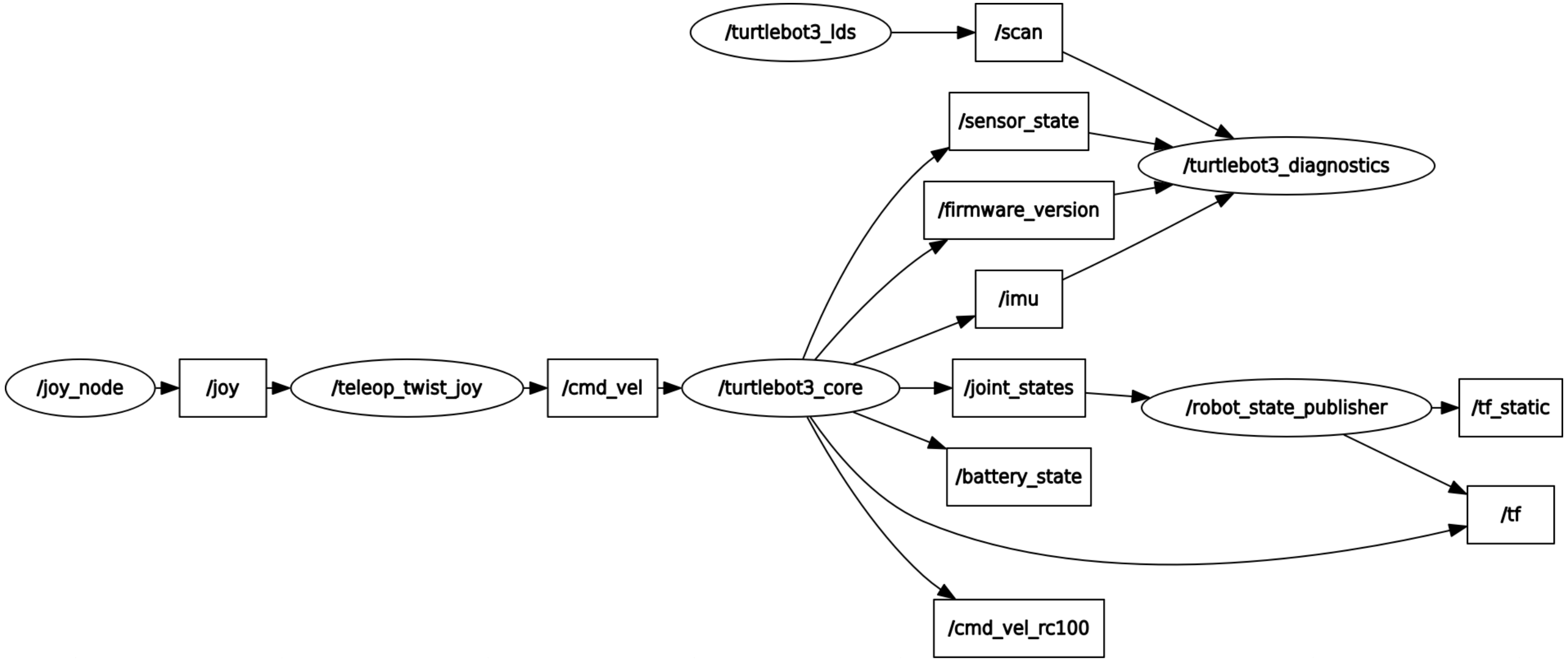




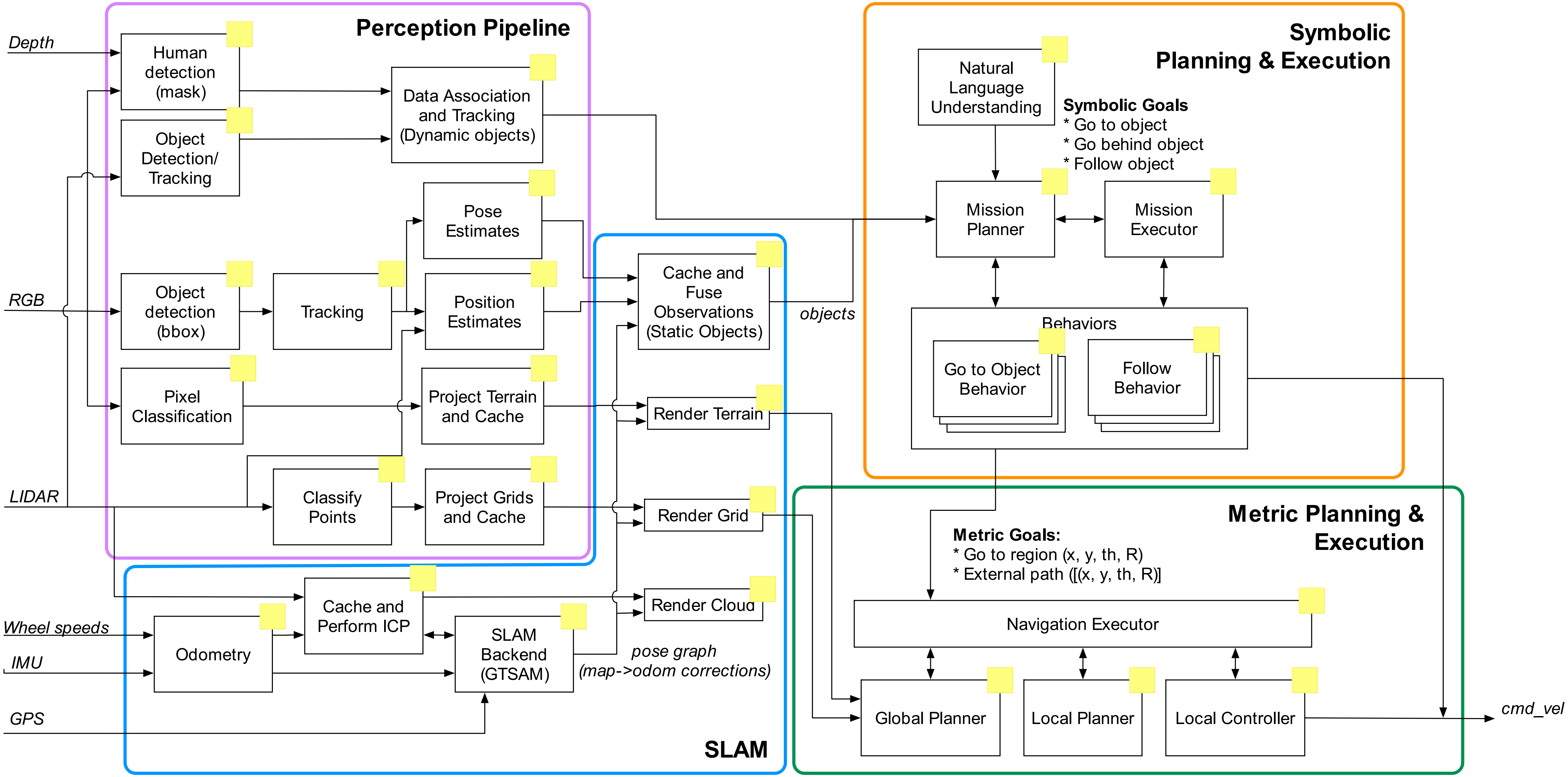
THE ROBOT OPERATING SYSTEM (ROS)

- **What is ROS**
  - A flexible framework for writing robotics software
  - Primarily supporting C++ and Python
  - Standards are tracked through ROS Enhancement Proposals (REP)
  - Designed to simplify collaborative robotics, avoid reinventing the wheel
  - Individual functions (nodes) pass information (messages) in standardized formats
  - Flexibility of algorithms using similar inputs and outputs
  - Platform agnostic design

- **Long pedigree**
  - History back into the late 90s
  - Player/Stage/Gazebo into Player2
  - Willow Garage
  - Open Source Robotics Foundation



ARL GROUND AUTONOMY REFERENCE ARCHITECTURE



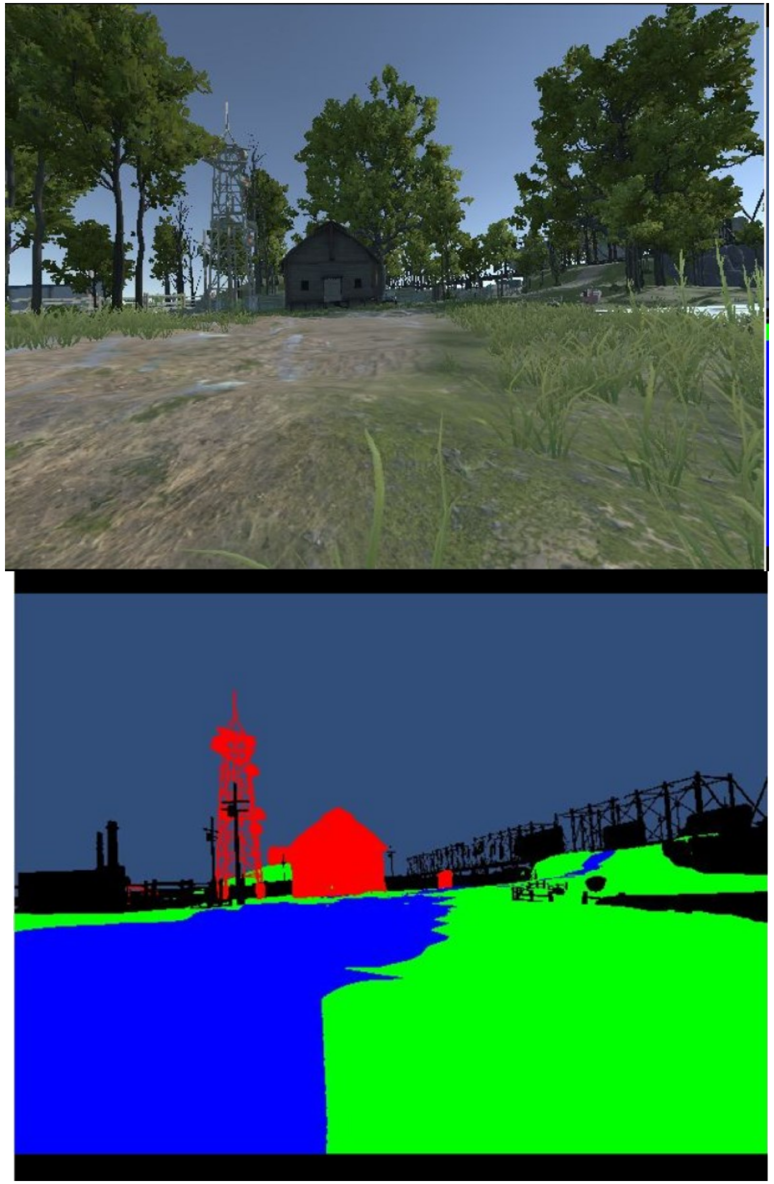
PERCEPTION

- **Object Detection**
  - Neural Net based object detection (e.g., You Only Look Once (YOLO))
  - Using trained models to identify objects in an image
  - Outputs classified bounding boxes
- **Object Tracking**
  - Determines from frame to frame if detect objects are the same
  - Maintain continuity of objects and their unique IDs for world model
  - Uses Simple Online Realtime Tracking (SORT) as default, very simple but effective
- **Object Localization**
  - Takes in LiDAR pointclouds and labelled images from detector
  - Determine pose of objects in the world by combining location in image and location in pointcloud
  - Requires accurate LiDAR-Camera calibration for estimating object location

PERCEPTION

- **Semantic Segmentation**
  - Classifies each pixel in the image (as opposed to bounding box style object detection), or each point in a point cloud

- **Multi-modal Fusion**
  - Combine complementary semantic segmentation algorithms (image, point cloud) into a single output
  - Returns SegmentedPointCloud, which is a Point Cloud that embeds labels and confidences for each point
  - Used to create costmaps based on the cost of semantic classes (e.g., for terrain traversability)



ARL GROUND AUTONOMY STACK – METRIC PLANNING

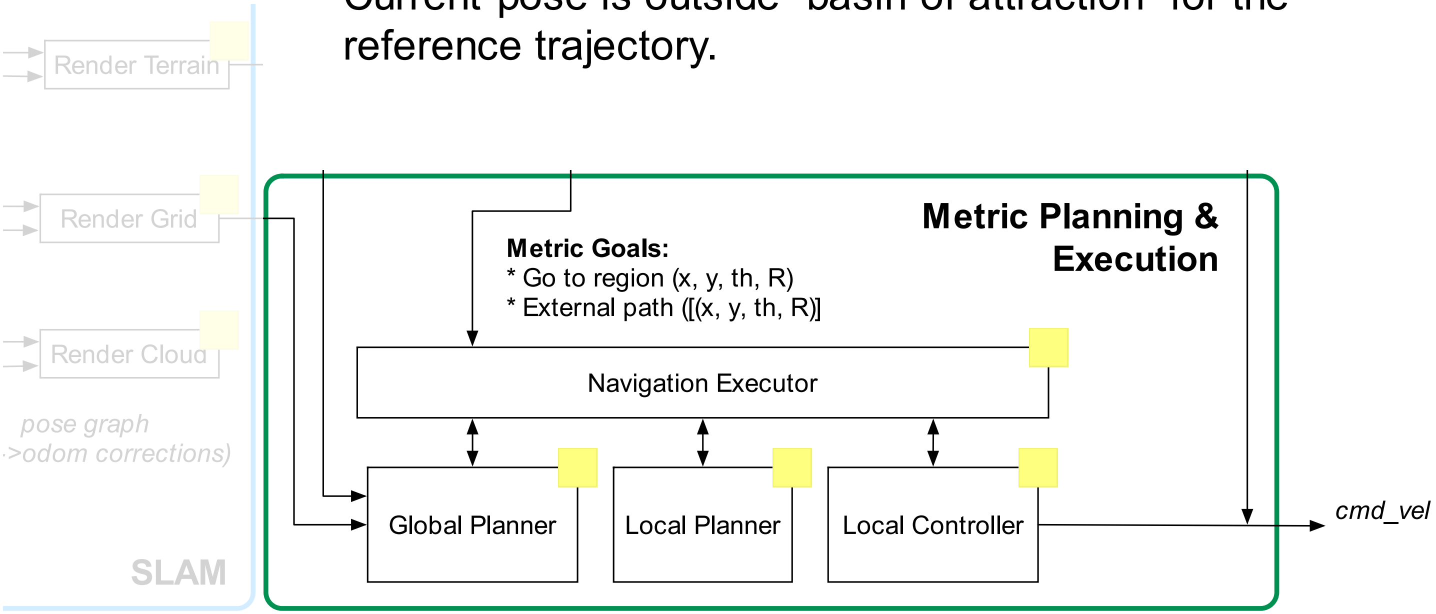
Global Planner
Compute obstacle free motion plans from start to goal location in map frame. Can be kinematically feasible, e.g., lattice planning with motion primitives.

Local Planner
Compute short-horizon trajectory to follow reference trajectory in odometry frame. Incorporate local costmap with high update rate. E.g., trajectory optimization via receding horizon MPC.

Local Controller
Achieve desired controls between updates of the local planner. Often a simple passthrough to platform.

Metric Goals
- Go to region (x, y, θ, R)
- External path [(t, x, y, θ, R)]

Navigation Executor
Accept metric goals and sequence execution of global planning, local planning, and control. Monitors task completion and triggers global replanning if
- Current plan is infeasible (due to new knowledge of obstacles)
- Current pose is outside “basin of attraction” for the reference trajectory.



ARL AUTONOMY STACK PLANNERS

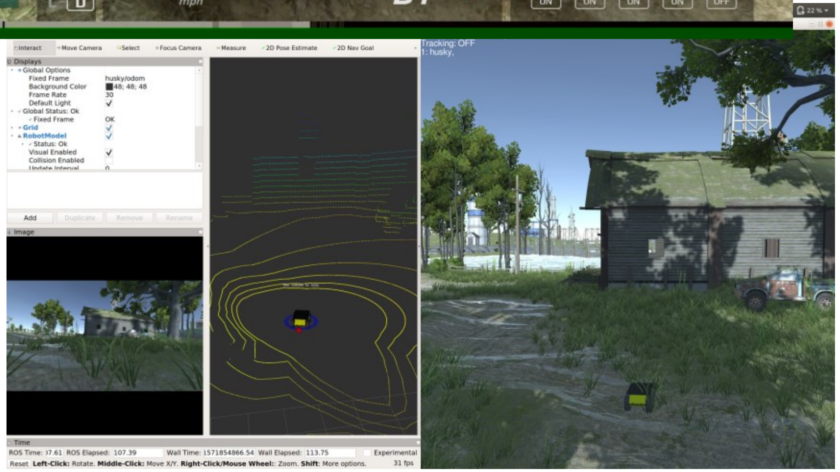
- **Global Planners**
  - Search Based Planning Library (SBPL)
    - Graph search library implementing Anytime Heuristic Search to optimize path
  - Generalized Lazy Search (GLS)
    - Switches between lazy search and edge evaluation to optimally compute shortest path
  - Efficient Adaptive State Lattice (EASL)
    - Discretizes the space of states considered during adaptation to limit search time and computation

- **Local Planners**
  - NLopt
    - Nonlinear optimization library generating optimized trajectories
  - Model Predictive Path Integral (MPPI)
    - Kinematic controller based on Model Predictive path integral control. Relies on random sampling and an exponential weighting scheme to determine controls.
  - Receding Horizon Model Predictive Control (RHMPC)
    - Algorithm for mobile robot model predictive control that utilizes the structure of a regional motion plan to effectively search the local continuum for an improved solution.

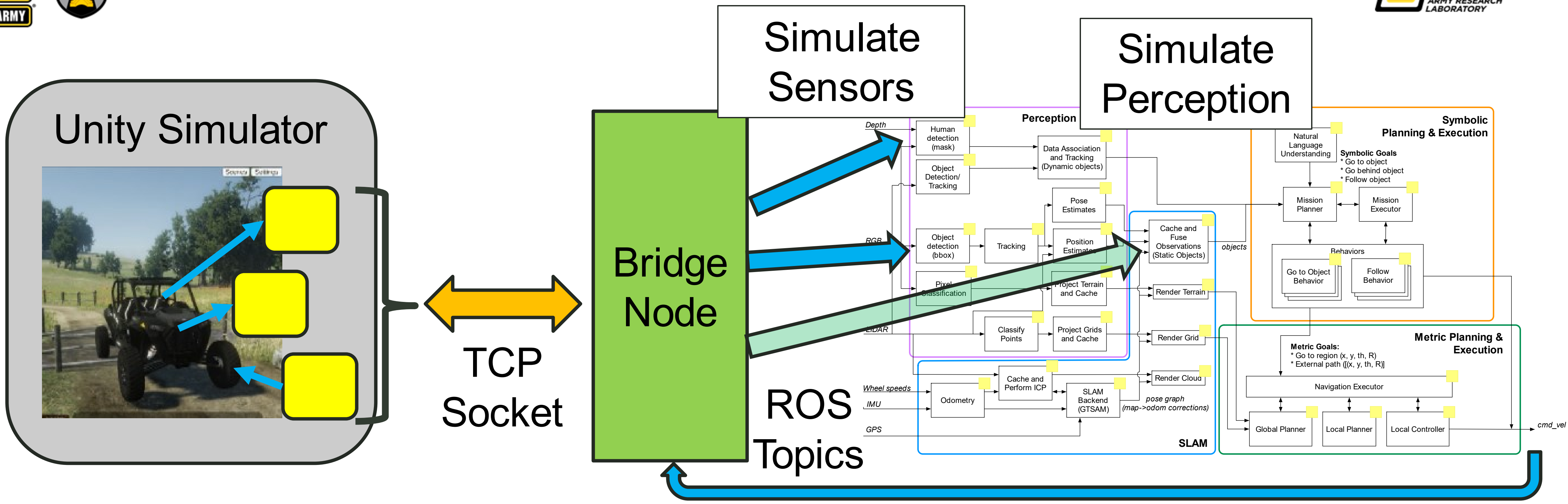
ARL UNITY SIMULATION

ARL Autonomy Stack has an “opt-in” package for a Unity-based perceptual and physics simulation

- **ROS Interface**
  - Spawn robots, sensors, environment
  - Receive sensor data on hardware-compatible topics
  - Send actuation commands to simulated platform
- **Platforms**
  - Clearpath Husky
  - Clearpath Warthog
  - Polaris MRZR
- **Sensors**
  - IMU
  - Cameras
    - RGB
    - Depth
    - Semantic segmentation
    - Object detection
  - 3D LIDAR



ARL SIMULATOR ARCHITECTURE



Unity Features:

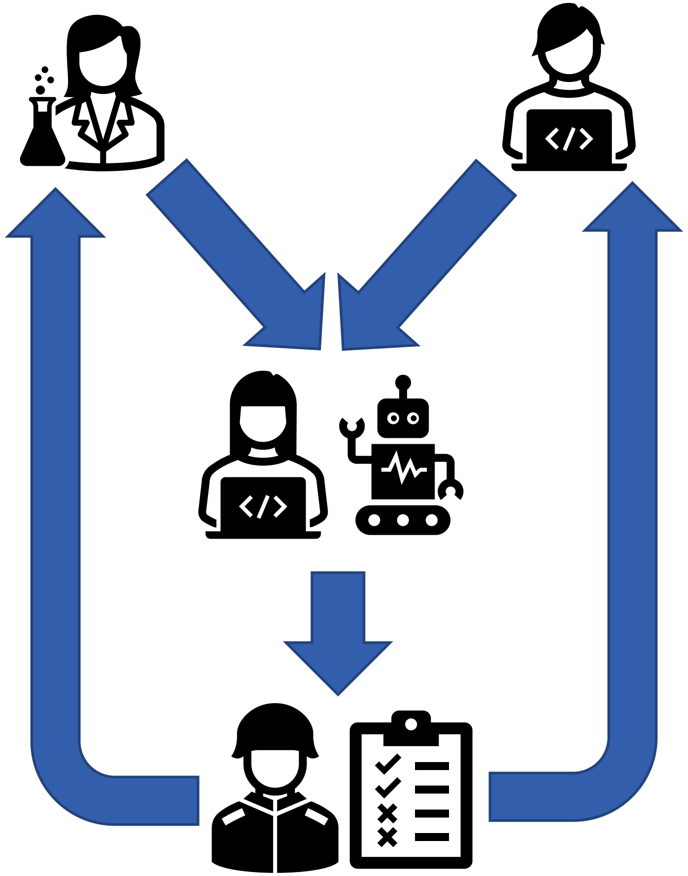
- C++ engine, C# scripting interface
- Full-featured editor
- Many commercial plugins/tools

The MITRE Multi-Agent Environment (MMAE)

MMAE offers capabilities that bridge the needs of:

- **Reinforcement learning (RL) research scientists**
- **Robotics developers**
- **Applied researchers**

MMAE offers a sandbox for RL researchers to iterate with robotics and operations experts throughout training, experimentation, and testing, shortening timelines for transitioning policies onto robotic platforms for operationally relevant demonstration and evaluation.



Origins of MMAE

MMAE was developed in support of Dr. Derrik Asher under the ARL AI for Maneuver and Mobility (AIMM) Essential Research Program (ERP):

- **Objective:** Next Generation Combat Vehicles (NGCV) with AI-enabled maneuver that rapidly learn, adapt, reason, and act in Multi-Domain Operations (MDO).
- **Challenge:** The future operational environment will be contested in all domains in an increasingly lethal and expanded battlefield, conducted in complex environments against challenged deterrence. (MDO 1.5, 6 Dec 18)

MMAE enables researchers to pursue a fundamental question of the AIMM ERP: can we develop and integrate **autonomous mobility** and **context-aware decision making** for combat environments to achieve **<u>autonomous maneuver</u>**?

MMAE Capabilities

The MMAE framework combines a reinforcement learning interface for training and testing with Unity-developed configurable multi-agent tasks

- Provides tested multi-agent Unity environments with configurable bounds, agent spawning locations, and collision geometry
- Train, resume, transfer, and test single agent reinforcement learning (RL) or multi-agent RL (MARL) policies
- Includes Combined Search and 3D Predator-Prey tasks, along with templates to easily configure and modify additional tasks
- Bash script interface and JSON based integrated framework for configuring task, algorithm, environment, agents, and training parameters
- Easy integration with RLlib and Gym reinforcement learning algorithms:
  - Single and multi-agent training
  - RLlib's Proximal Policy Optimization (PPO) and Multi-Agent Deep Deterministic Policy Gradient (MADDPG) reinforcement learning algorithms are currently available for train/test
  - Gym-style interface that provides ability for easy integration of new algorithms
- Multi-agent Unity environments provide configurable multi sensor observation capabilities

MMAE User Interface and Visualization

2D "Mini-Map" summary view

Agent Observations and Dynamic Sensor Visualizations

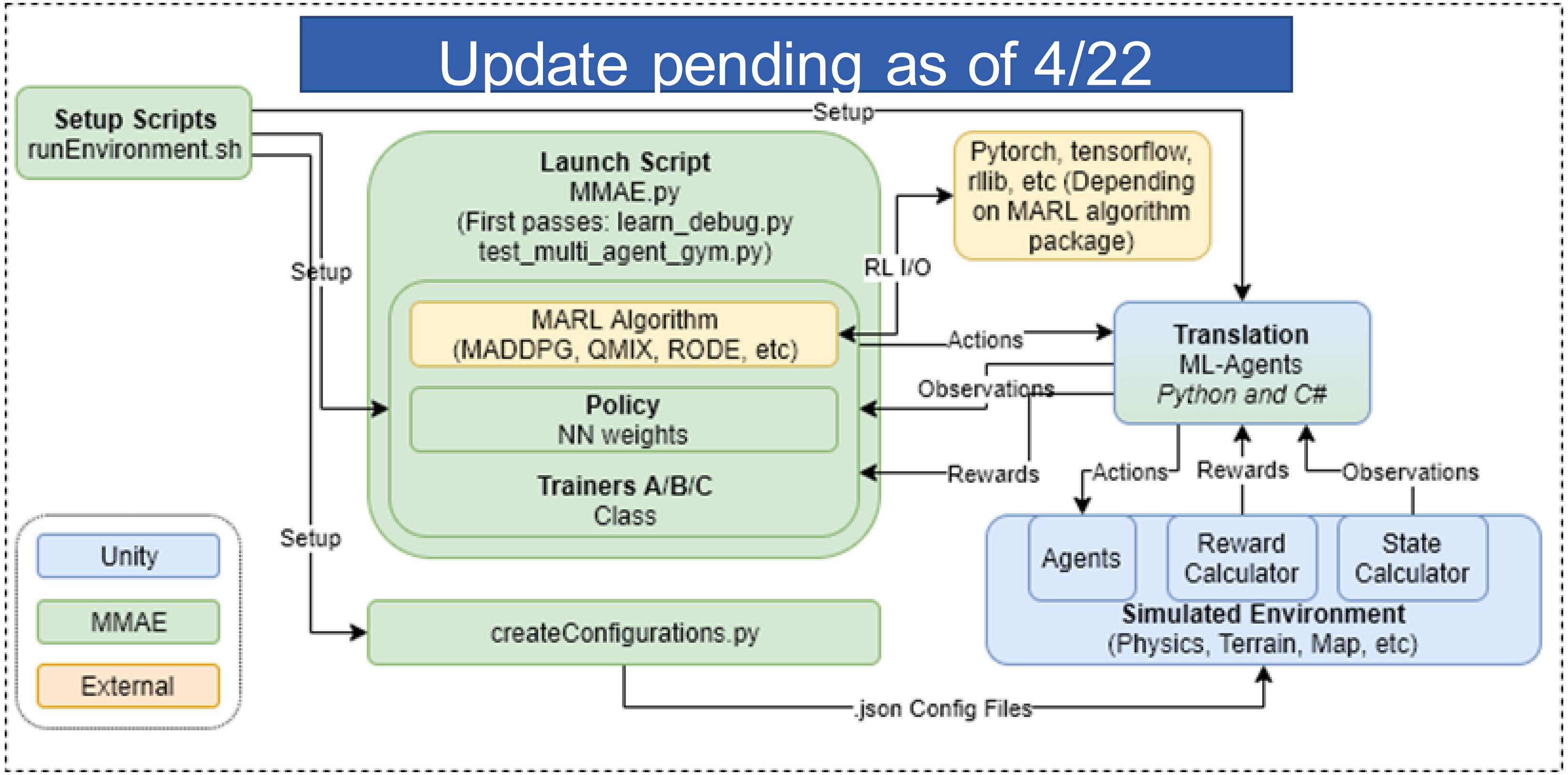
Individual Agent Ground Truth Information

MMAE Features

- **MMAE provides easy to use configuration options that offer task specific design flexibility**
  - Unity environments:
    - Map selection (Camp Lejeune, bounded geometric environments)
    - Number of agents
    - Multi sensor selection configuration (Camera, Ray perception, Near field and far field sensors for traversability and map exploration)
    - Unified reward settings configuration across all tasks
    - Physics model parameters including agent dynamics and actions
    - Selection of predefine geobounds for train/test, and agent spawn location constraints
    - Exploration state initialization options – options for setting pre-explored states
    - Logging and GUI configurations
  - Reinforcement learning algorithms:
    - Bash script for easy setting for logging, debugging, weight save/restore, train/inference mode, Headless/GUI mode, reward reconfiguration, RL library and algorithm selection settings
    - RL training hyperparameter configuration and settings scripts via JSON files

MMAE Architecture and Components

The MMAE framework architecture integrates the simulated environment and agents, RL algorithm and training process management, and a shared interface for communication and configuration of those components



AGENDA

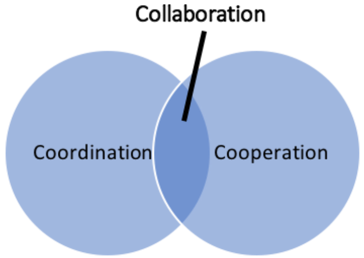
- Program motivation
- Army doctrine background
- Desired outcomes/ metrics
- GFE Software
- **Some ARL research relevant to this call**
- Questions

Current Effort in Multi-Agent Tactical Behaviors

Assumption:
The future Army will have computational entities (agents) controlling robots and executing decisions faster than humans can, which will require artificial intelligence (AI).

Solution:
Creating agents that specifically adjust (adapt) their behaviors, strategies, and actions to that of their counter-parts (human or computational) will result in the most effective and efficient multi-agent mission execution.

Agents Leveraging Learning for Intelligent Engagement with Soldiers



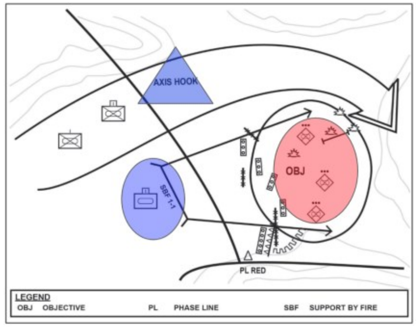
Coordination – an explicit measurable relationship between actors that is described by directly observable quantities.
* Perfectly correlated dancing agents
Cooperation – an implicit or inferred relationship between actors (aligned goal).
* Baking a cake and cooking a steak (independent sub-goals)
Collaboration - the intersection of both coordination and cooperation.
* Predator agents working together to capture a prey agent

ALLIES Objectives

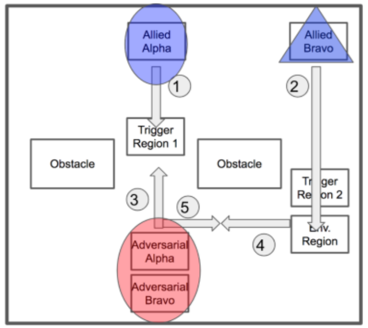
Identify Collaboration: develop, evaluate, and validate measures to quantify agent-agent, human-agent, and human-human collaboration.

Implement Collaboration: Use validated collaboration measures as teaching signals to improve efficiency in training AI agents and robustness in agents' behaviors in cooperative domains with human and non-human partners.

Operationalize Collaboration: Algorithmic and methodological modifications for optimal teaming.



Envelopment Maneuver from Doctrine

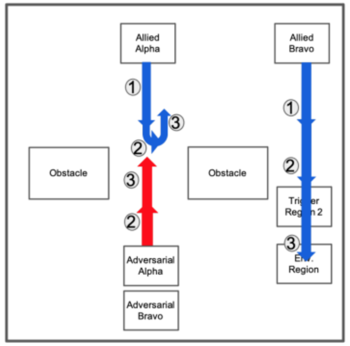


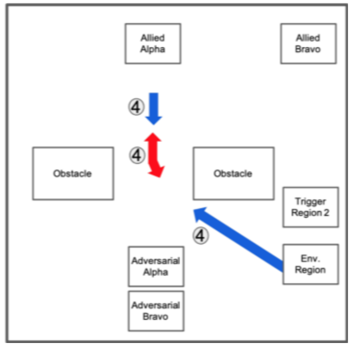
Implementation of Doctrine in Simulation

How can we operationalize collaboration in multi-agent systems?

Utilize Multi-Agent Reinforcement Learning (MARL) in combination with Army doctrine (MARDOC) to:

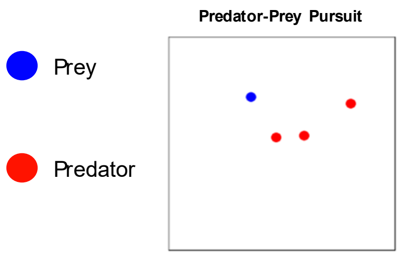
- Include doctrine into the agent training process to facilitate emergent militarily relevant behaviors in large/complex state spaces.
- Test the efficacy of Army doctrine in MARL tasks/scenarios towards an implementation of trained policies in physical robotic systems.





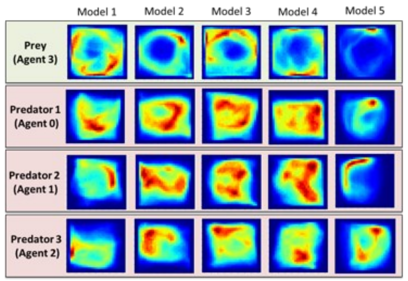
Emergent Militarily Relevant MARDOC Behavior

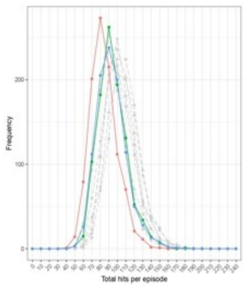
Emergent Coordination
in Multi-Agent Systems (MAS)

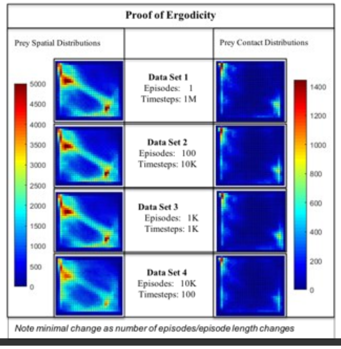


Measured Coordination in MAS

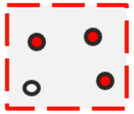
Ergodic Spatial Distribution Measurements



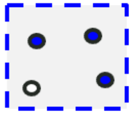




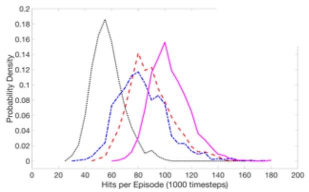
Policy Duplication



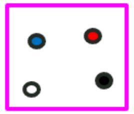
Homogeneous team – replicated Agent 0



Homogeneous team – replicated Agent 1

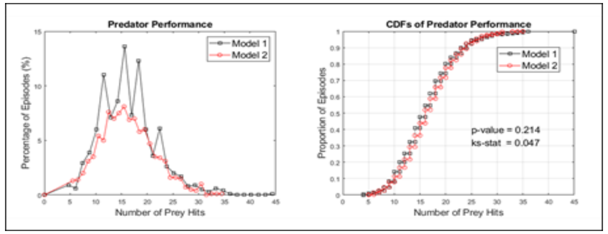


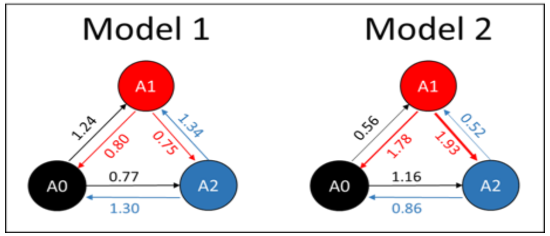
Homogeneous team – replicated Agent 2



Heterogeneous team - Agent 0, Agent 1, & Agent 2

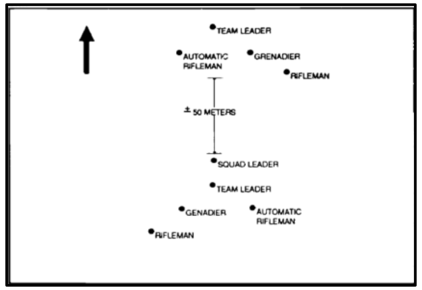
State Space Perturbations



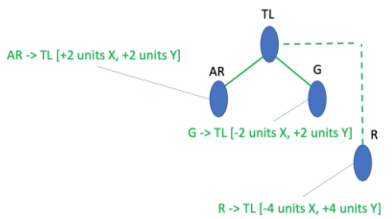


Current Projects

Formation Control for Coordinated Maneuver



The goal is to maintain the length of the line segments

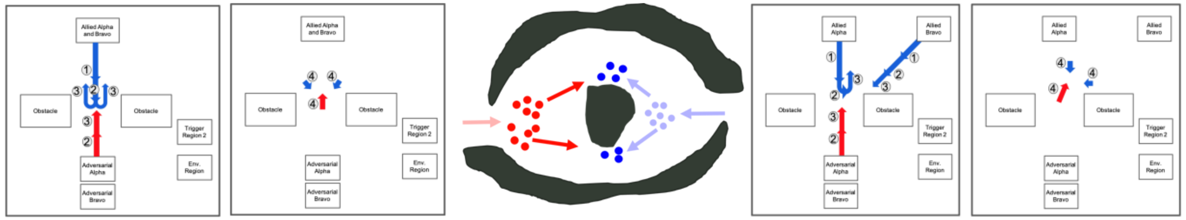


Treat TL (Team Leader) as the origin for distance calculation

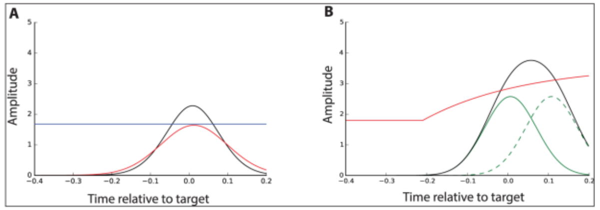
Need GPS of team lead and yourself (per robot)

Need to know the labels for every robot (i.e., AR, TL, G, R)
- Each robot needs to know their role/label, and the roles/labels of all other robots

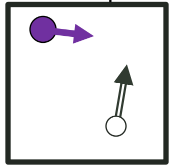
Policy Interpretations towards Robotic Implementation



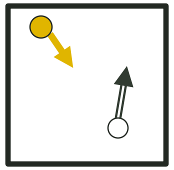
Meta-Reasoning for Fixed Policy Selection



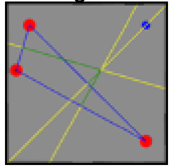
Interceptor



Chaser



Delaunay Triangulation



Questions?

usarmy.adelphi.devcom-arl.mbx.tbam-crp-questions@army.mil