Cooperative and Collaborative Distributed Autonomy

Dr. Scott Boskovich

smboskovich@cpp.edu

Electromechanical Engineering Department
College of Engineering
Cal Poly Pomona

Outline

- Distributed Autonomy
  - Homogeneous vs Heterogeneous
  - Vehicle Interaction
  - Cooperative & Collaborative
- Rule Based & Equation Based Approaches
- Vehicles and Architectures
- Intended Applications

What is Distributive Autonomy?

- Multiple Vehicles
- Vehicles perform autonomous behaviors using localized processing (on vehicle)
- Vehicles are "connected"
  - Share local state to neighboring (or farther)
  - Individual vehicle to Infrastructure (V2I)
  - Individual vehicle to Individual vehicle(V2V)
- Composition Types:
  - Homogeneous Compositions
  - Heterogeneous Compositions
  - Mixed

Composition Types

- Homogeneous Compositions
  - Swarms
  - Same vehicle type
  - Same vehicle purpose
  - Redundancy through replacement

- Heterogeneous Compositions
  - Teams/Clusters
  - Different vehicle type
  - Different purpose
  - Redundancy via similar capabilities using different methods of implementation
    - More resilient to common error

Distributive Use Cases

- Mapping unknown Environments
  - Quickly
  - Hostile
- Search and Rescue / C4-ISR Applications

- Logistics & Transportation of goods
  - Connected Vehicles "Transportation Trains"
  - Warehouse Automation

- Dynamic Vehicle Routing
  - Incident rerouting
  - Control for vehicle flow without devices i.e. signals

Distributed Processing

- Pure Centralized
  - Significant I/O bandwidth required
  - Significant Processor Bandwidth required
  - State for each vehicle is always known if connected
    - Is this necessary?
- Pure Decentralized
  - Dynamic neighbor
  - 100% of all vehicle state is unknown at any time is difficult
    - Data can be propagated throughout network
    - It 100 % of all state data from
- Hybrid -> Combination of both[1]

Vehicle Behavior

- Fundamental to any structure is vehicle behavior
- Moreover Vehicle Interaction!
- Vehicles can "do their own thing" regardless of other vehicles
- Consequences: even with avoidance – depending on vehicle density
- The Higher vehicle density - the lower the ability to perform task

Describing Vehicle Interaction

- Rule Based
- Equation Based

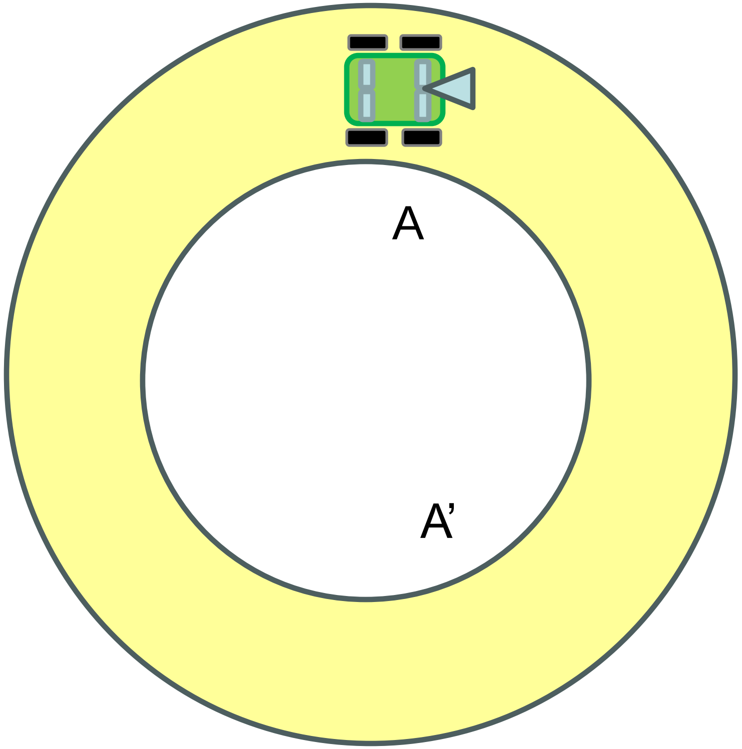
- Systems are Very Complicated & Complex
  - Homogeneous to a lesser degree
  - Heterogeneous to a larger degree
  - Subject to Initial Conditions

- How to ensure individual objects are met based on the code of the individual?

  - Individually vehicles algorithms can be complete, and computable
  - Consider more than one vehicle…

Cooperative Autonomy^2

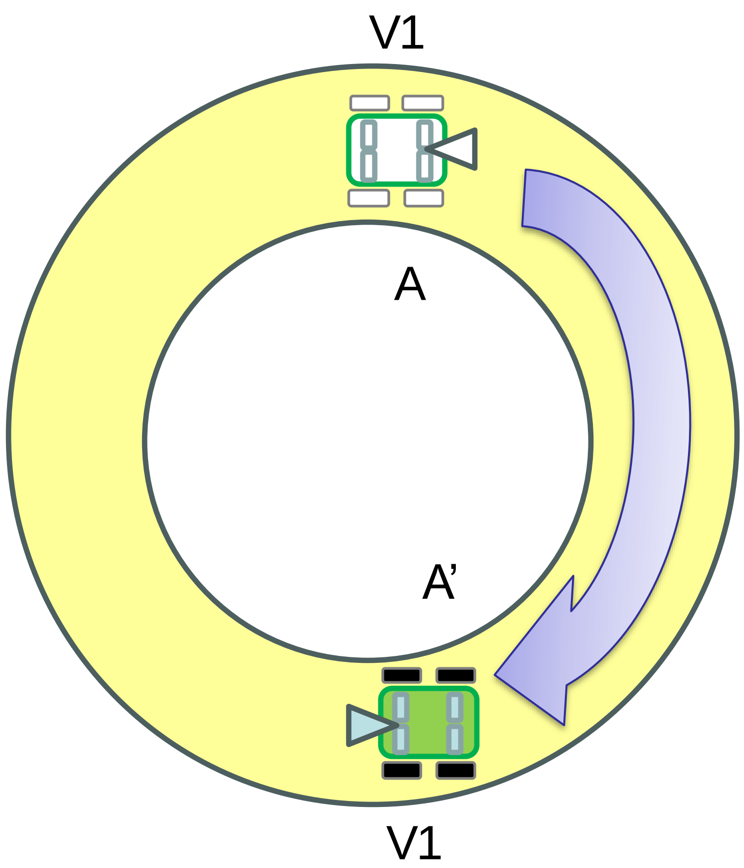
- Simple Scenario
- Given ***one*** Vehicle
- Path from Point A to Point A'
  - A*, D*
  - Wall Following
  - Bug Algorithms
  - Potential Fields
  - :



Cooperative Autonomy?

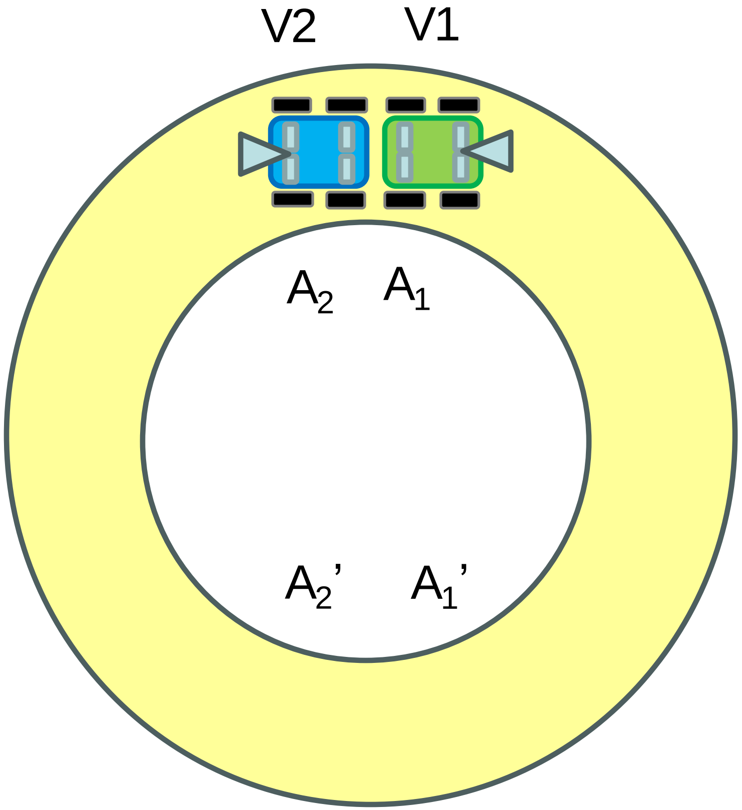
- Simple Scenario
- Given ***one*** Vehicle
- Path from Point A to Point A'
  - A*, D*
  - Wall Following
  - Bug Algorithms
  - Potential Fields
  - :

Vehicle V1 can perform the task while cooperating with itself



Cooperative Autonomy?

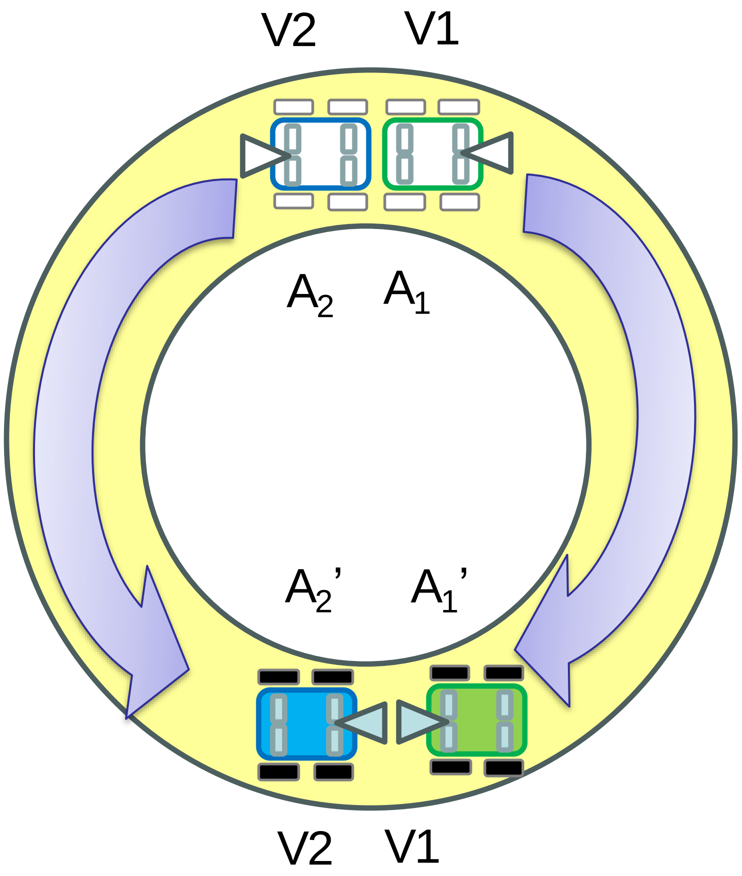
- Simple Scenario
- Given ***Two*** Vehicles
  - Homogeneous
- Are they really Cooperative???



Vehicle V1 and Vehicle V2 can both perform the task while cooperating with one another

Cooperative Autonomy?

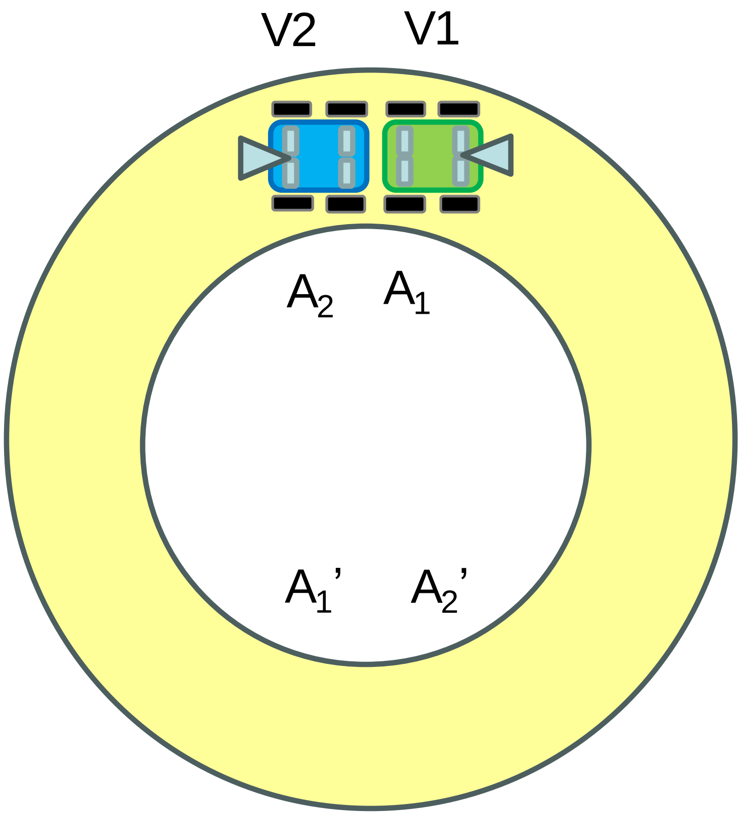
- Simple Scenario
- Given *Two* Vehicles
  - Homogeneous
- Are they really Cooperative???



Vehicle V1 and Vehicle V2 can both perform the task while cooperating with one another

Cooperative Autonomy?

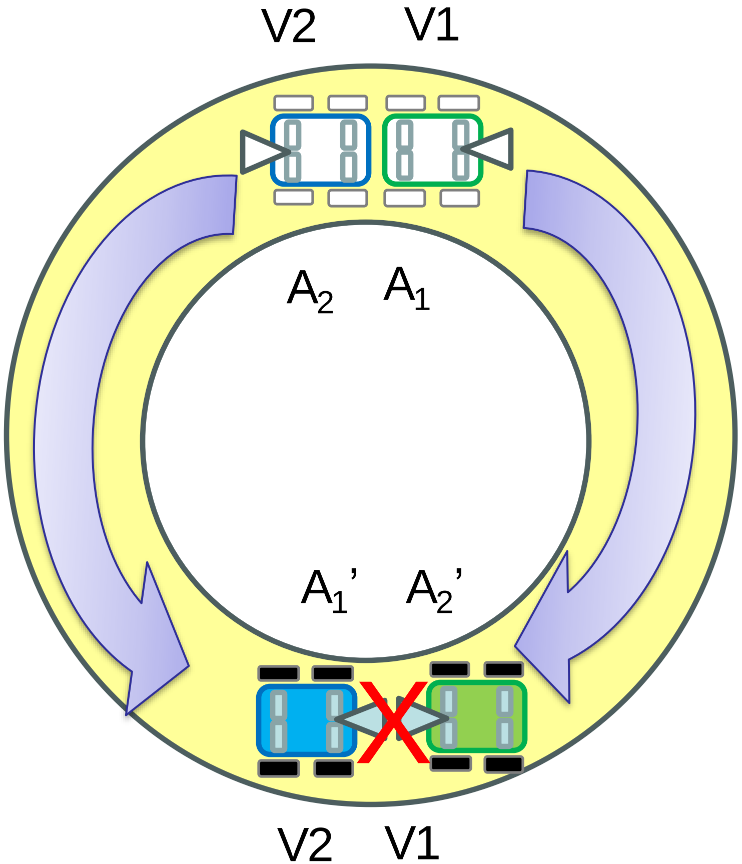
- Simple Scenario
- Given *Two* Vehicles
  - Homogeneous
  - Change Destinations…
- Are they really Cooperative???



Vehicle V1 and Vehicle V2 can both perform the task while cooperating with one another

Cooperative Autonomy?

- Simple Scenario
- Given *Two* Vehicles – Homogeneous
- By Looking at code only and knowing each codebase worked – was this predictable?

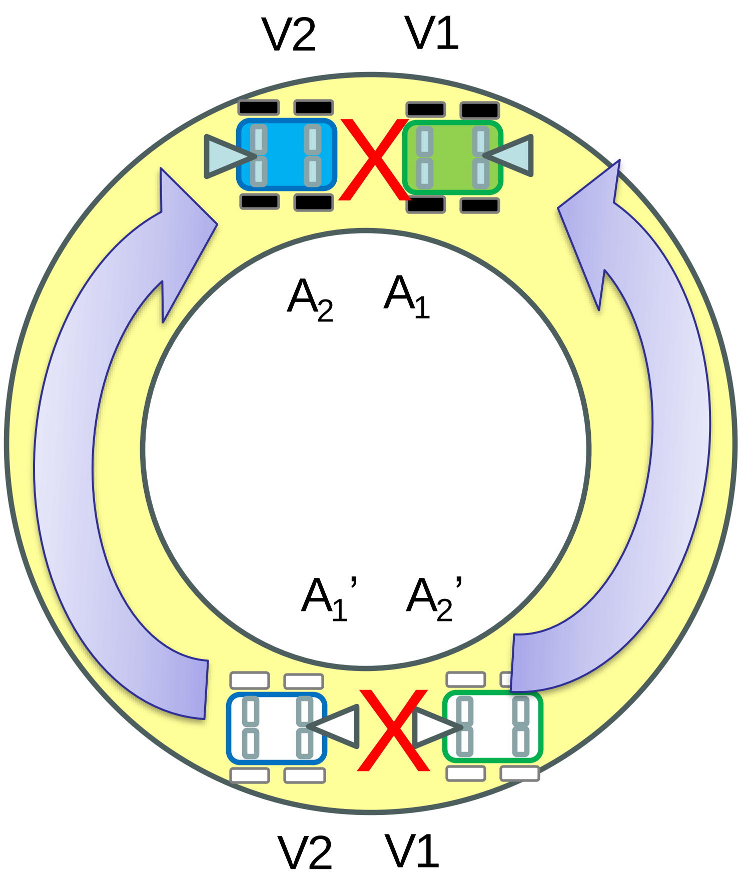


Vehicles are not cooperable

Cooperative Autonomy?

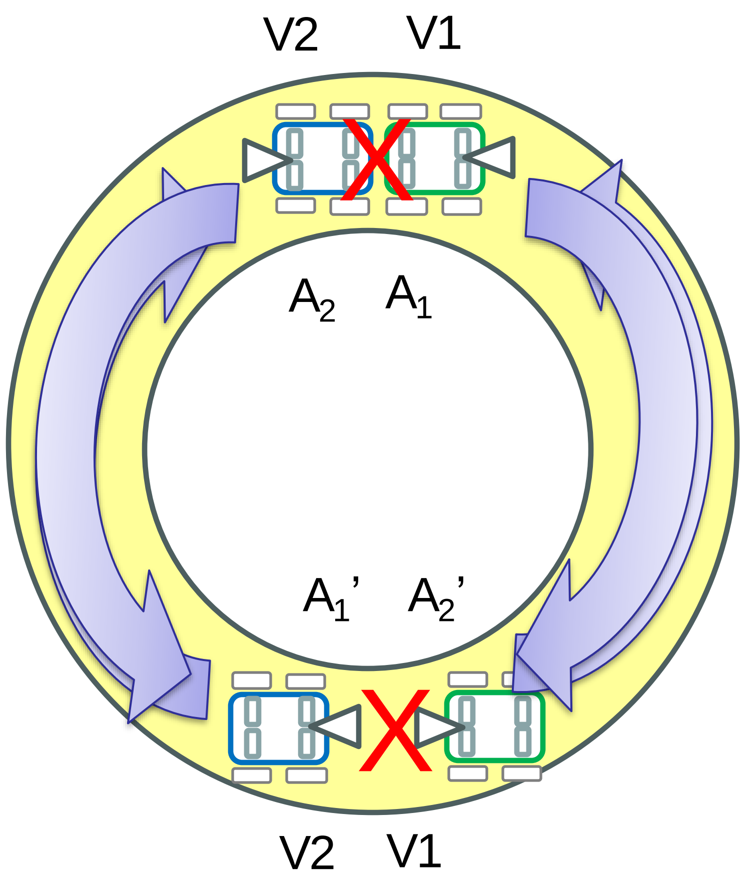
- Simple Scenario
- Given *Two* Vehicles – Homogeneous
- By Looking at code only and knowing each codebase worked – was this predictable?

Vehicle V1 and Vehicle V2 impede each other unable to reach their destination



Cooperative Autonomy?

- Simple Scenario
- Given *Two* Vehicles – Homogeneous
- By Looking at code only and knowing each codebase worked – was this predictable?

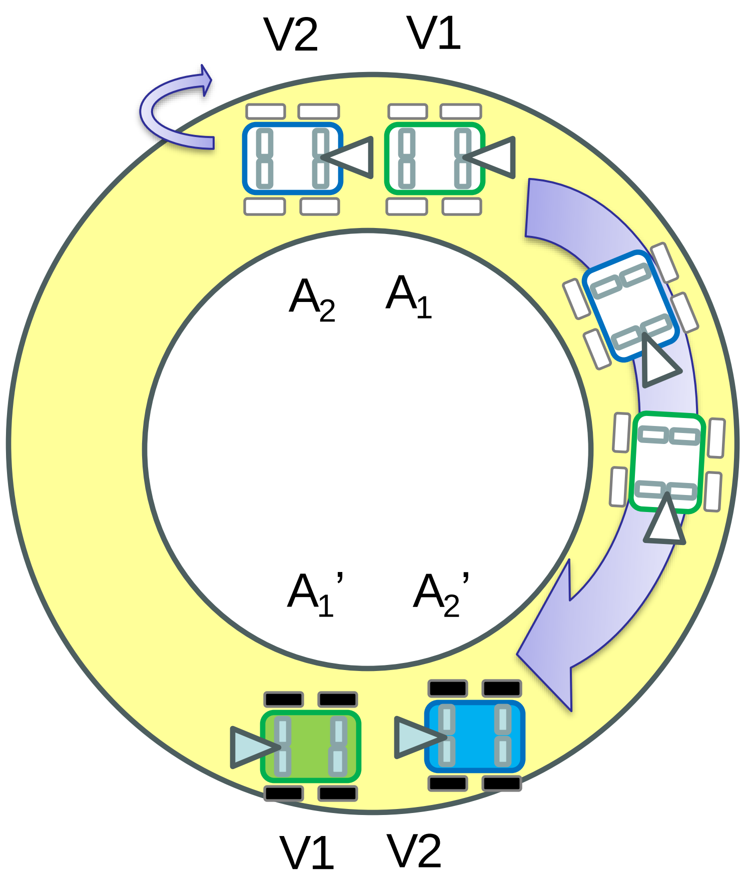


The system would oscillate back and forth unable to reach the desired goal state => unstable unless checked for oscillation. System unable to attain goal

Cooperative Autonomy?

- Simple Scenario
- Given *Two* Vehicles
  - Heterogeneous
  - V2 rotates 180
  - Follows V1
  - Rotates back 180 to desired pose
  - Method to detect cooperative before testing?

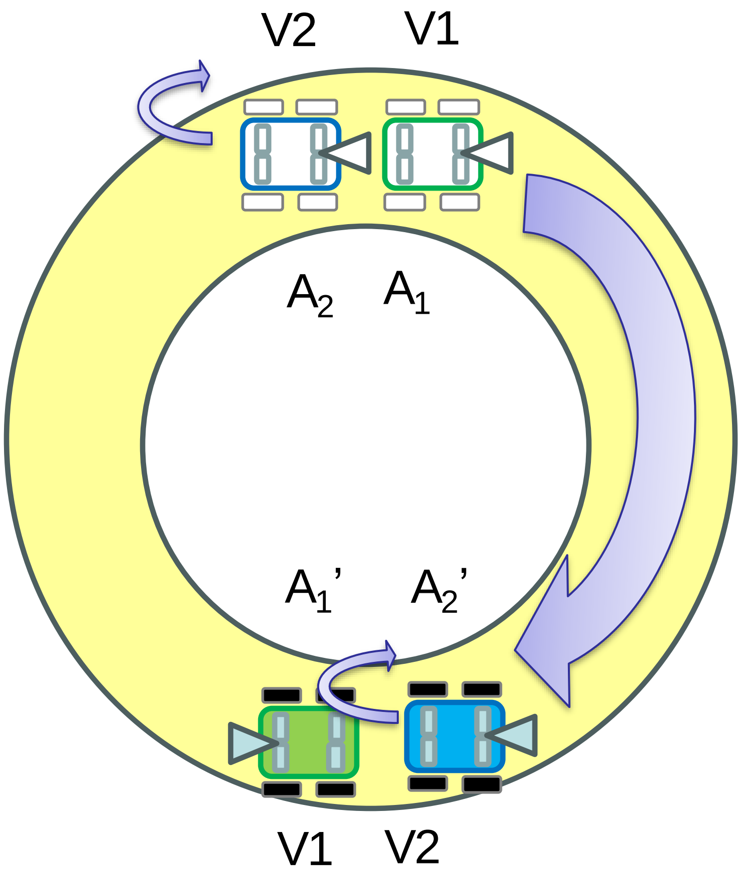
Vehicle 1 and Vehicle 2 are cooperative



Cooperative Autonomy

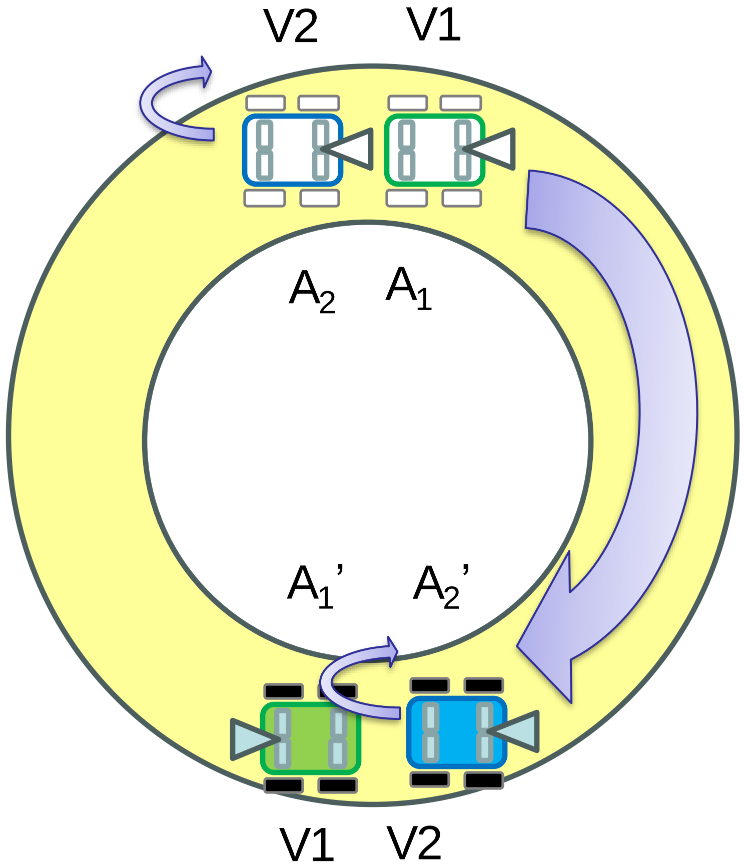
- Given *Two* Vehicles
  - Heterogeneous
  - V2 rotates 180
  - Follows V1
  - Rotates back 180 to desired pose
  - Method to detect cooperative before testing?

Vehicles are now different types, or one has evolved its behavior to become different



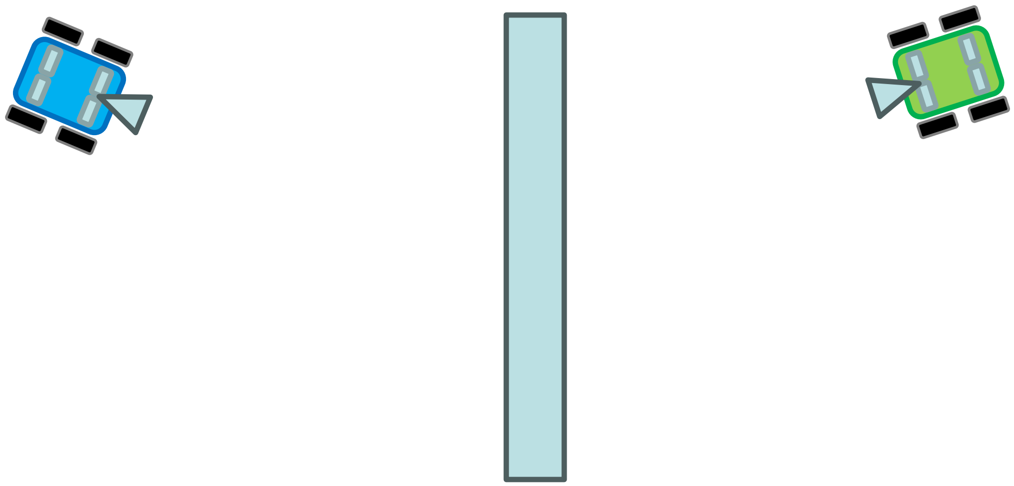
Cooperative Autonomy

- This was using a rule-based approach
- If detected that oscillation exists then only one can updates its behavior
- The approach is ad-hoc and require testing to ensure closure
- System is sensitive to initial conditions



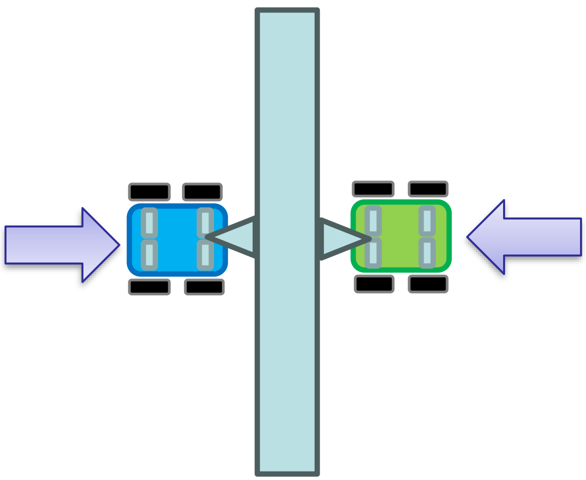
Collaborative Autonomy

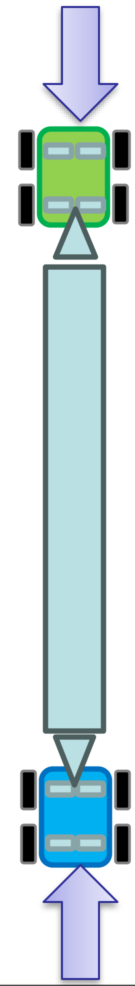
- Given two vehicles must find and push a pipe



Collaborative Autonomy

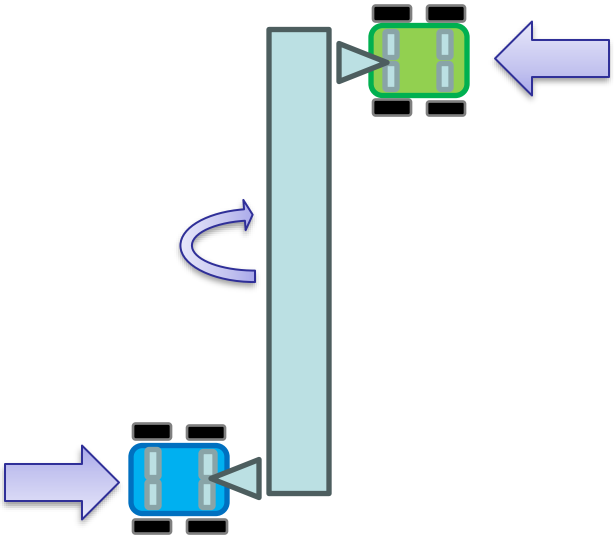
- If avoiding each other & at Midpoint push -> uncooperative





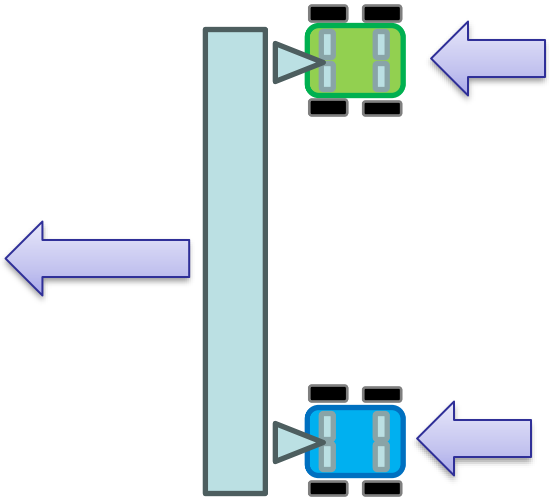
Collaborative Autonomy

- Selecting an end symmetric about each end
  -> Cooperative, not Collaborative



Collaborative Autonomy

- Selecting an end symmetric about each end
  -> Cooperative and Collaborative



Vehicle Interactions are Complex

- Large complex vehicle systems are Complex Systems
- A Complex System Exhibits[3] :
  - Large Number of Interacting Agents
  - Exhibit Emergent Self Organizing Behavior
    - This can be difficult to anticipate from individual behavior
  - Their Behavior does not result from the existence of a centralized controller
- Complex Systems are non-linear[4]
- Consider Defining System Stability as that of a Complex System

Rule Based Vehicle Behaviors

- Vehicle Behaviors can be described using Rule Based Methods
  - BOIDS
  - Predator Prey
  - Formation Flow -> Follow The Leader etc...
  - Conway's Life

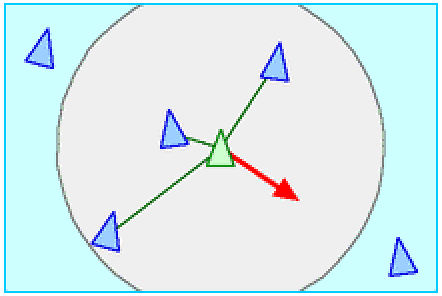
- Simple Rules can exhibit complex interactions

- Using a set of Rules how does the designer know if the system will perform its intended purpose?

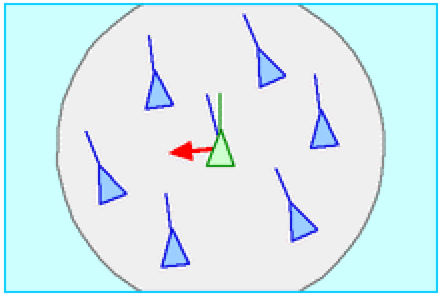
  - Rule Based Systems are Ad-Hoc
  - What was the reasoning for the original rules

BOIDS[5,6]

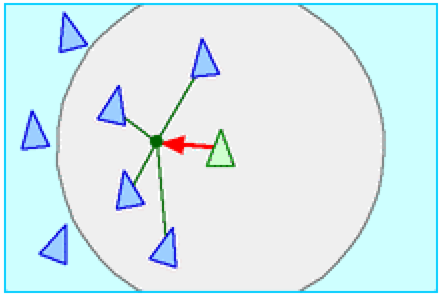
- First published in 1986 by Craig Reynolds
  - Three simple rules exhibit flocking



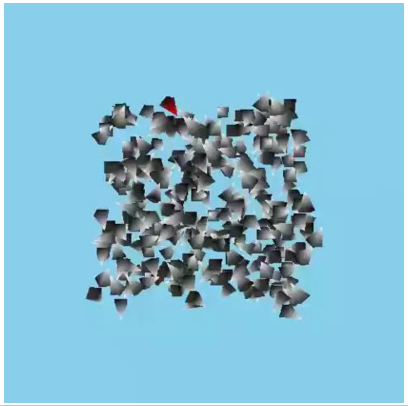
Separation



Alignment

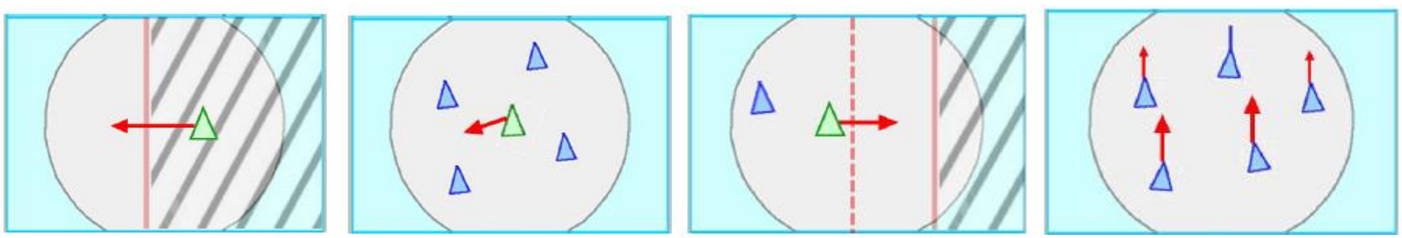


Cohesion

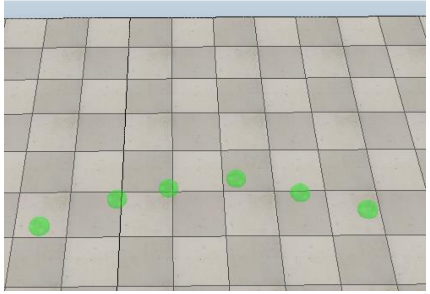


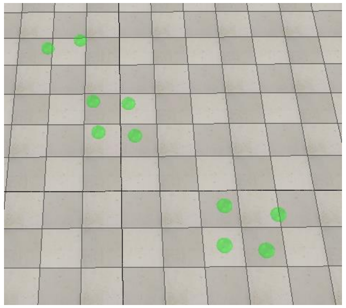
Extensions and Use of Boids

- Boids was extended to perform Distributed Mapping by adding additional rules[7]
- Efficient to map a given region by use of Distributed Autonomy
- Implemented in Python and Simulated via Coppelia Simulation Software



Boundary Avoidance | Eccentricity | Boundary Cohesion | Velocity Gradient





- Boids is found to be a stable system[7]
- Boids has been realized in both Rule Based and Expression based models[8]

Extensions and Use of BOIDS

Boundary Avoidance: Boids Steer towards the mapped area when reading a predefined border. Achieved by modifying the steering vector. As the boid travels further away from the map, the steering vector increases

Eccentricity: Due to aggregation causing concentration of boids forcing mapping inefficiencies, **'time-in-group'** is used to allow boid to steer away from neighbor group into new direction.

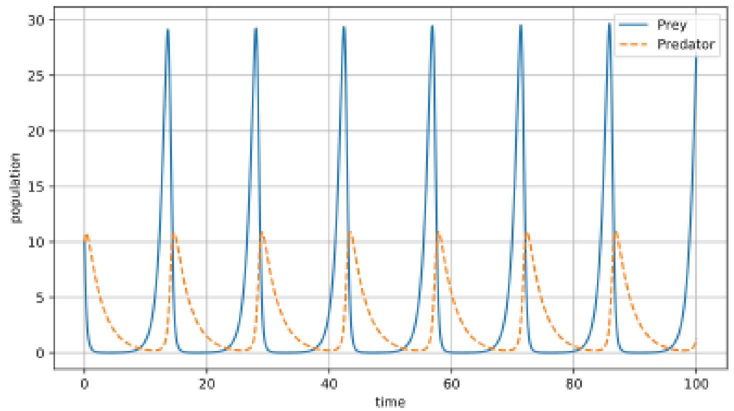
Boundary Cohesion: To prevent missing regions to map, due to perpendicular avoidance, this allows the boid to stay within a distance from an edge.

Velocity Gradient: The further away a boid is from the leading edge, the velocity of the boid is increased. This results in creating fronts vs bounded clusters

Describing Dist. Complex Systems with Expressions

- Formation Flow -> Follow the Leader
- Predator–Prey[10] (Lotka-Voltera equations)

$$\frac{dx}{dt} = \alpha x - \beta xy,$$
$$\frac{dy}{dt} = \delta xy - \gamma y,$$



X	population density of prey
Y	population density of predator
dx/dt	growth rate of the prey population
dy/dt	growth rate of the predator population
α, β	prey growth rate and effect of predation on prey
γ, δ	predator death rate and effect of presence prey on predation growth rate

where

Predator–Prey Implications

- Coupled non-linear system
- Predator Prey can be decomposed into Lyapunov Functions to demonstrate Stability[10,11]
- Effort to decompose General Rule Based Systems into Expressions
- Apply Lyapunov Function Composition to rule decompositions
- Provide ability to assess stability for generalized Dist. Systems vs Ad-Hoc approaches

Implementation of Dist. Autonomy

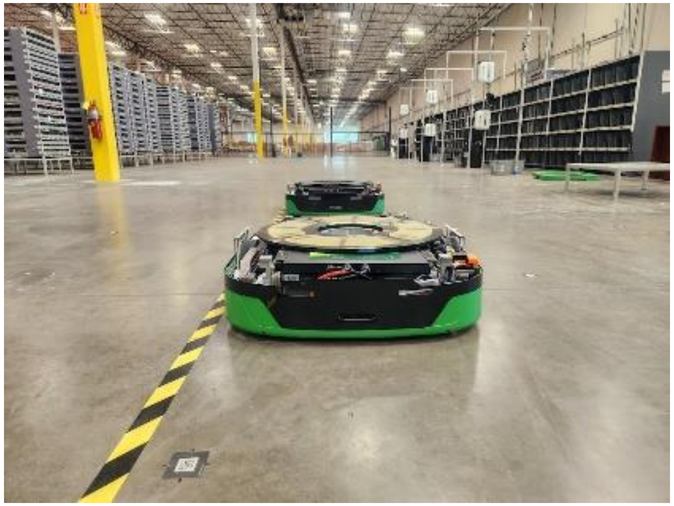
- Ideally a system is known to be stable prior implementation
- Simulation of complex dynamical systems with complicated hardware is difficult to simulate
  - Computational resources, Sim time, etc…
- Utilize autonomous mobile robots exhibiting behaviors as described in existing and TBD algorithms
- Develop method using Lyapunov Functions to decompose rules into expressions to determine distributed system stability

Using Mobile Robots

- Received a donation of 60 autonomous robots from iHerb Corp.
- Originally developed for logistic warehouse applications
- Based upon hybrid Centralized System Arch.
- Robots were designed / developed inhouse with advanced engineering teams including:
  - Mechanical
  - Electrical
  - Firmware
  - Embedded Software
  - Vehicle Management Software

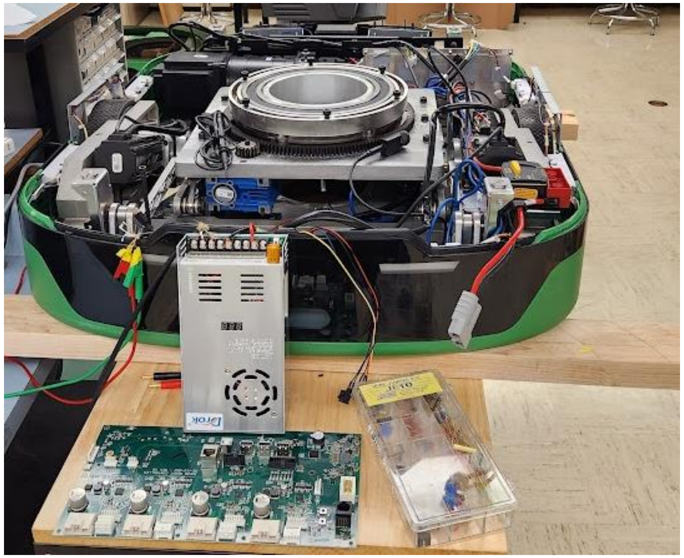
iHerb Robots – Baseline

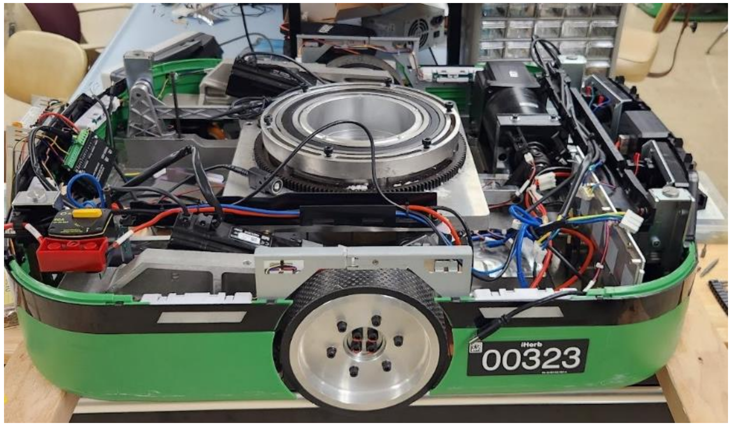
- Carrying Capacity of 500Ks @ up to 5m/s
- Motors & Drive
  - Differential Drive => 2 BLDC Motors
  - 1 Platform BLDC Motor
  - 1 Lift BLDC Motor
- Sensors
  - 1024 CPR quadrature encoder per motor
  - Intel Realsense Depth Sensors
  - Top & Bottom Integrated Cameras



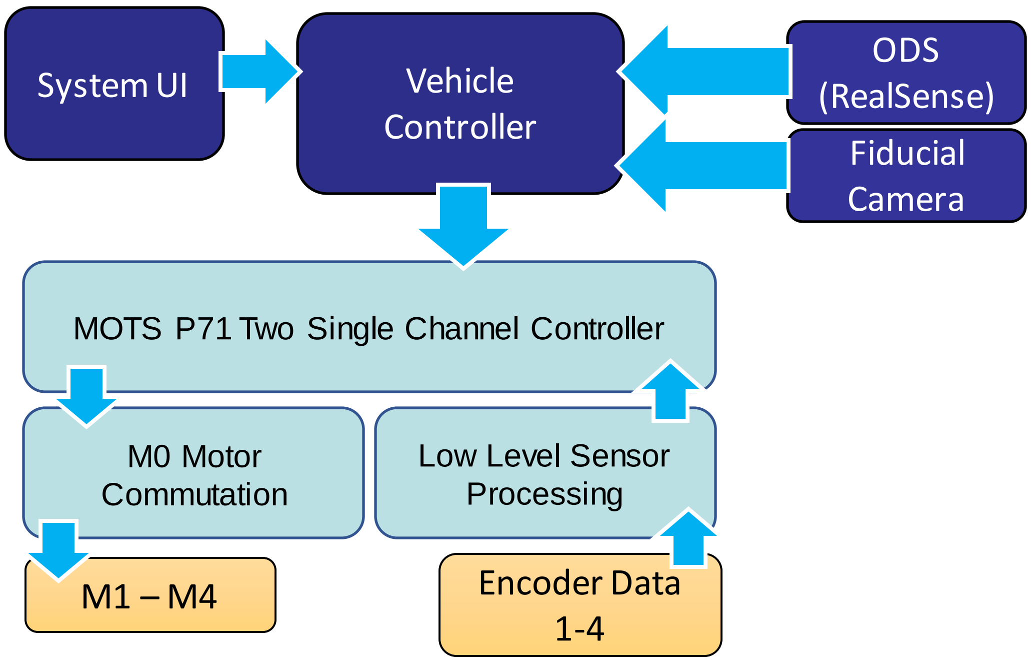
iHerb Robot Processing

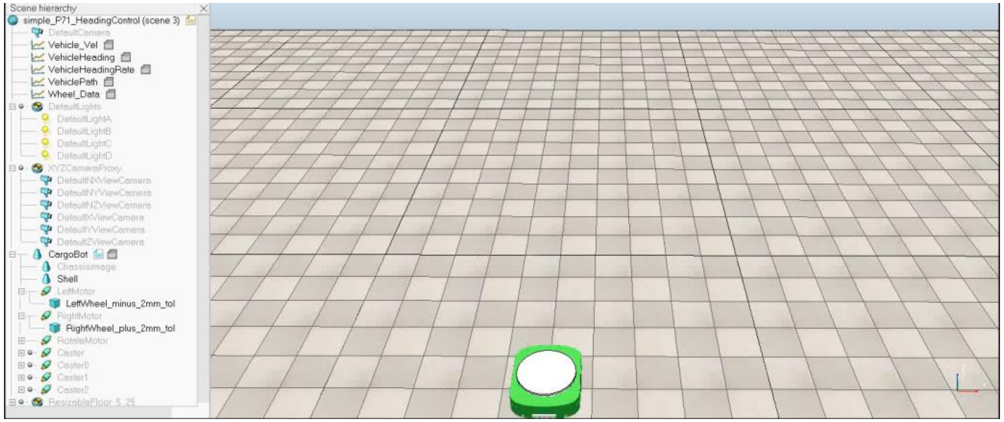
- Vehicles Received Commands from Central pub/sub
- Used QR Codes on ground for localization
- Internal Processing include:
  - iMX7 Dual Core ARM Application Processor w/ M4 core
  - Embedded Linux
  - 4 M0 ARM Processor for Commutation Control
  - Xilinx Zynq 7020 FPGA (Fabric + 2 ARM Processors)





Baseline Architecture / Controls





New Processing Architecture

- Implement a ROS2 DDS Pub/Sub-Based Approach with Ethernet Backplane

- Multi-Processor Design
  - iMX7 => nVidia Orin
    - Sensor Fusion, Image Processing and Behavior Processing

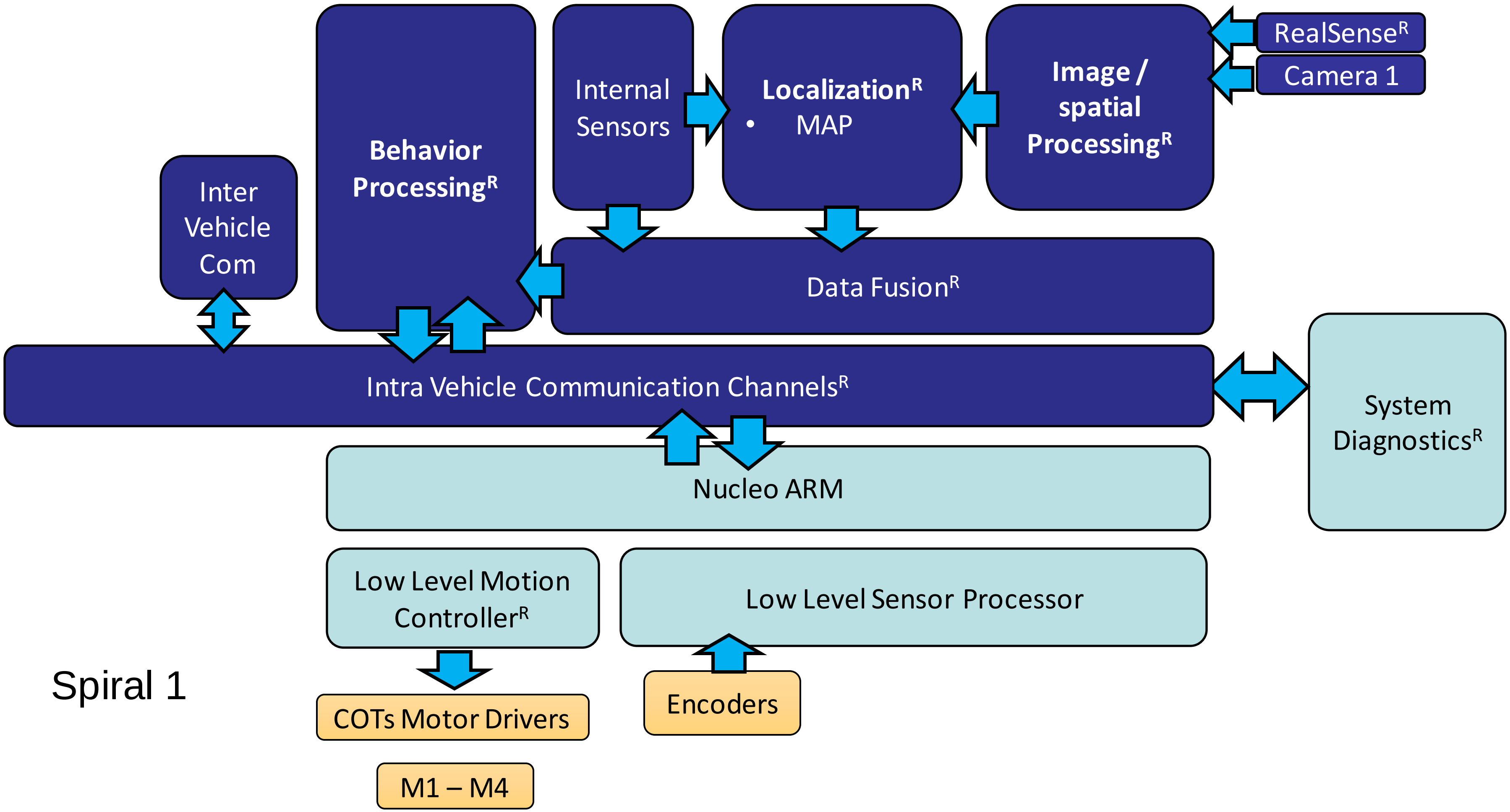
- BLDC Commutation M0 ARM => FPGA Fabric

- P71 Motion Control => FPGA ARM Processors

- Sensors to be Added:
  - Velodyne 3D Lidar, IMU, Additional Camera(s), UWB Localization via Qorvo Modules

- 3 Spirals
  - Each Spiral adds new feature while updating existing

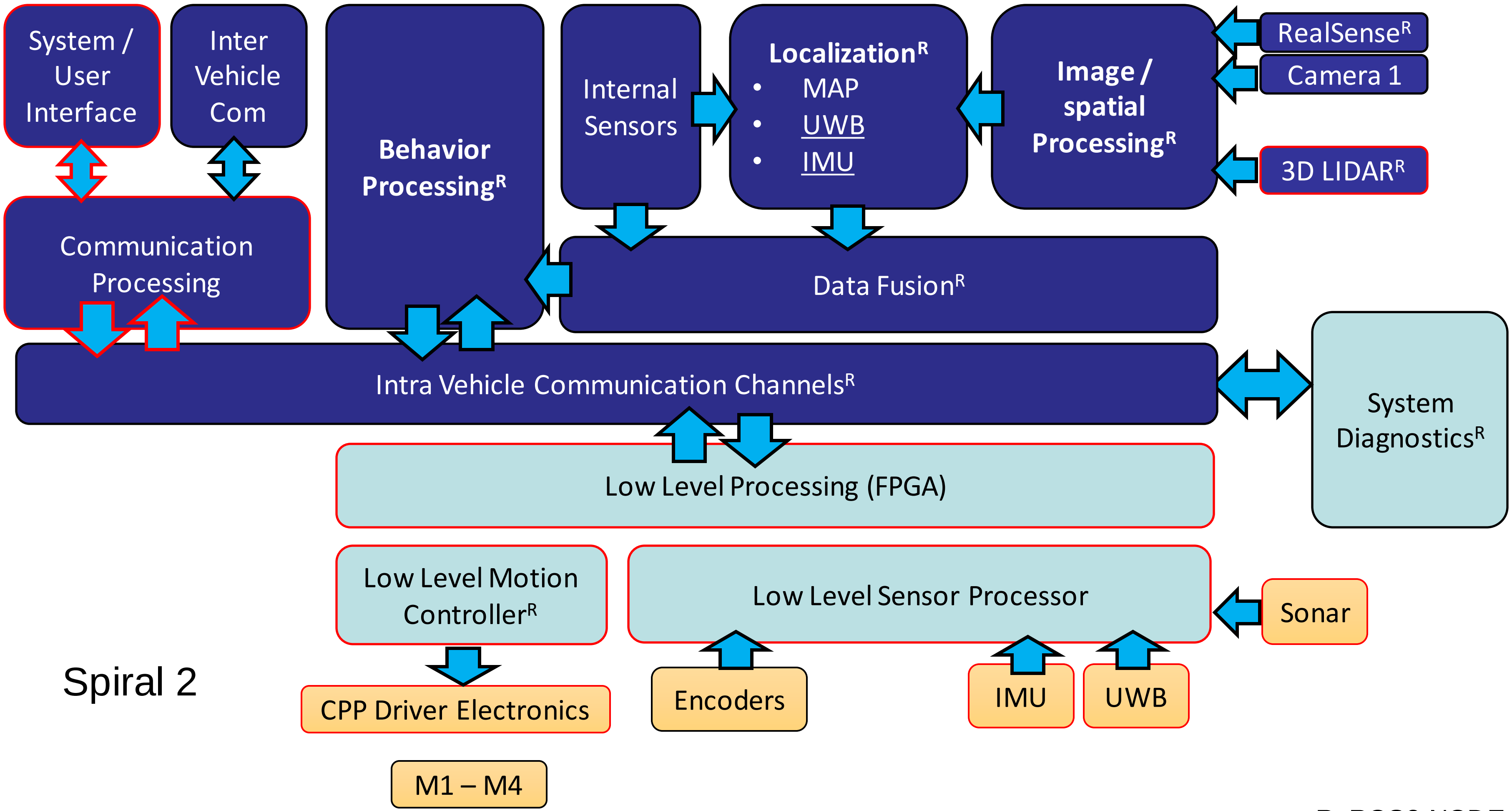
CPP Vehicle Processing Architecture



Spiral 1

R: ROS2 NODE

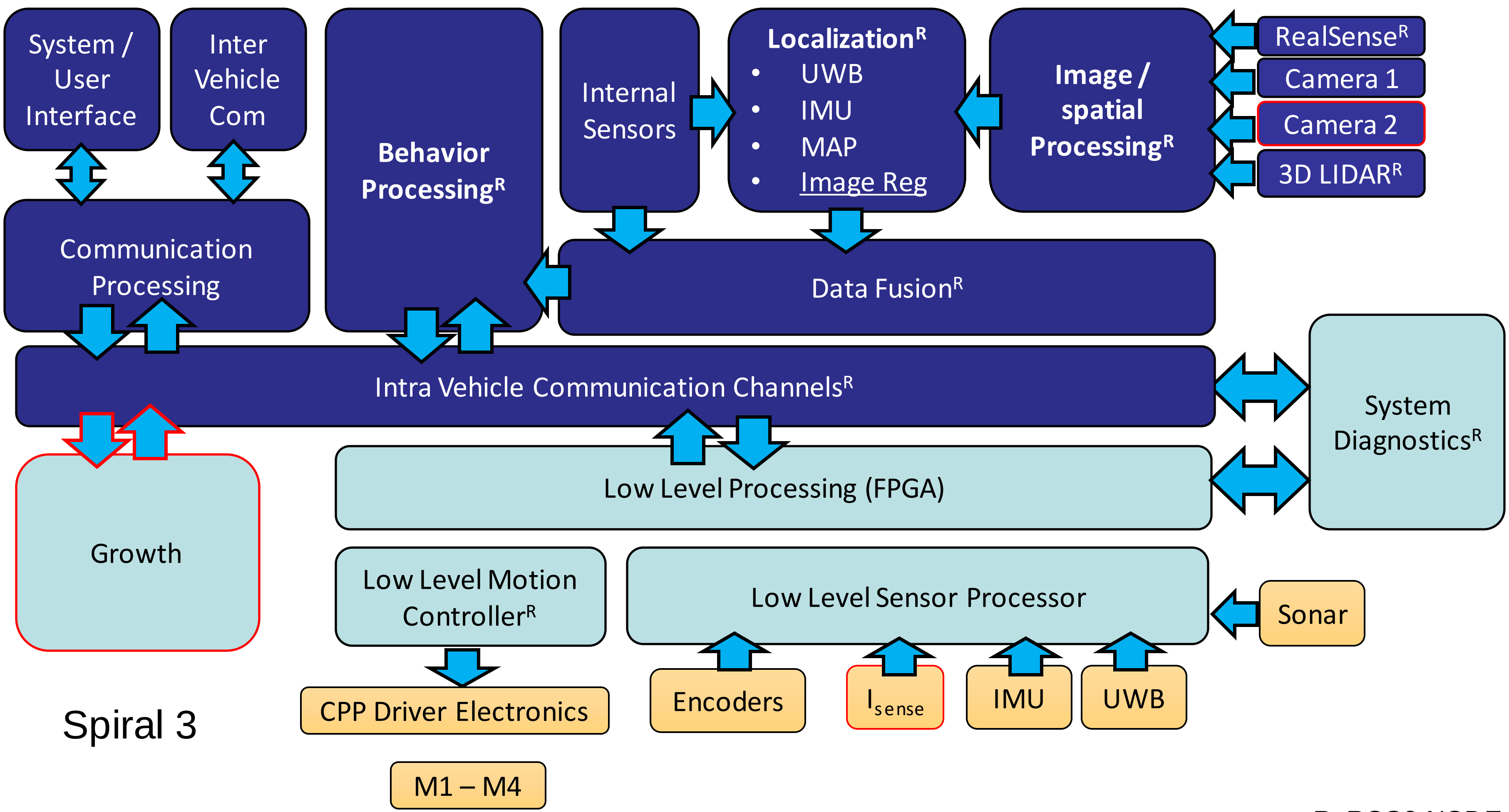
CPP Vehicle Processing Architecture



Spiral 2

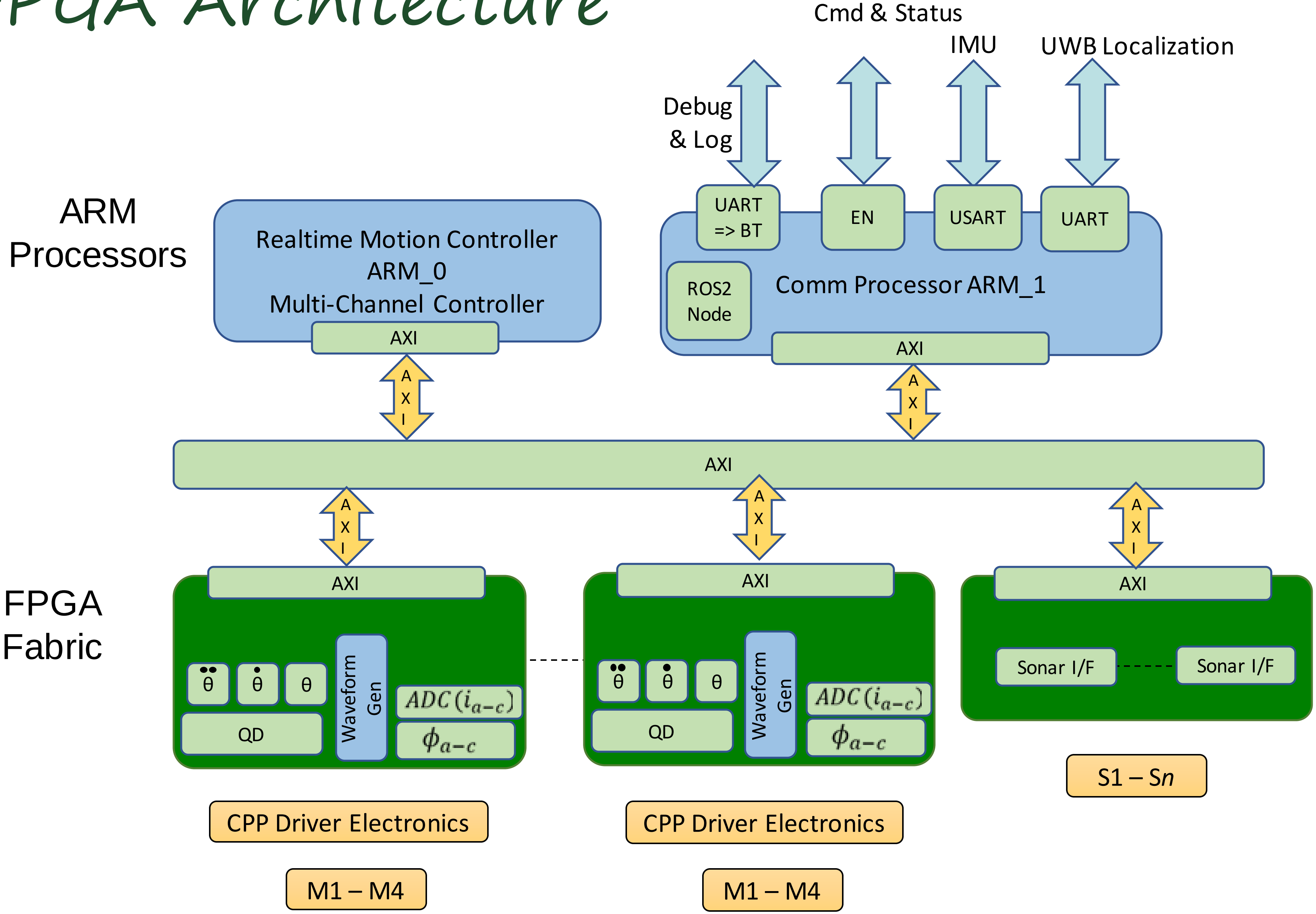
R: ROS2 NODE

CPP Vehicle Processing Architecture



R: ROS2 NODE

FPGA Architecture



Vehicle Use Cases

- Use CPP Pathway network as road network for ITS scenarios using Distributed Behavior Control Methods
  - Rule Based (BOIDS etc.) / Expression Based => Rule Closure
- Realtime Road-Network Image Segmentation[1,2] (recently completed work)
- Dynamic Mapping
- V2V, V2I => Dynamic Routing / Rerouting, Non-infrastructure Control
- Physically Connected Vehicle load sharing / Virtual Train






Thank You

Questions???

References

1. Boskovich, Barth " Vehicular network rerouting autonomy with a V2V, I2V and V2I communication matrix classification*", 16th International IEEE Conference on Intelligent Transportation Systems,* Oct, 2013.
2. Boskovich, S., *A Distributed Approach to Dynamic Routing Algorithms Based on Vehicle-to-Vehicle Interaction*, University of California, Riverside, March 2015.
3. Newman, M. E. J. (2011). Resource letter. cs-1: Complex systems. *American Journal of Physics, 79*, 800–810.
4. Bianconi, G., Arenas, A., Biamonte, J., Carr, L. D., Kahng, B., Kertesz, B., Kurths, J., Lü, L., Masoller, C., Motter, A. E., et al. (2023). Complex systems in the spotlight: Next steps after the 2021 nobel prize in physics. *Journal of Physics: Complexity, 4*(1), 010201.
5. Reynolds, Craig (1987). "Flocks, herds and schools: A distributed behavioral model". Proceedings of the 14th annual conference on Computer graphics and interactive techniques. Association for Computing Machinery. pp. 25–34.
6. Wikipedia, access 12/4/2023, BOIDS
7. Boskovich, Sherman, "Application of a Modified Boids Algorithm for Navigation and Mapping with a System of Small UAVs", *American Institute of Astronautics and Aeronautics Scitech Forum*, San Diego CA, Jan., 2019
8. Tanner, Jadbabai, Pappas, "Stability of Flocking Motion*" Technical Report No. MS-CIS-03-03*, University of Pennsylvania
9. Sharma, Vanualailai, Rag, "Obstacle and Collision Avoidance Control Laws of a Swarm of Boids*", World Academy of Science, Engineering and Technology, International Journal of Mathematical, Computational, Physical and Quantum Engineering,* Vol. 8, No. 2, 2014.
10. Wikipedia Access 12/4/2023 Predatory-Prey
11. Boukal, Krivan, "Lyapunov functions for Lotka-Volterra predator prey models with optimal foraging behavior", *Journal of Mathmatical Biology*, pp 493-517, Springer, 1999
12. Weigel, Cameron, "LiDAR and Camera Scene Understanding Methods and the Construction of a Small-scale Autonomous Vehicle", Thesis, Cal Poly Pomona, 2023