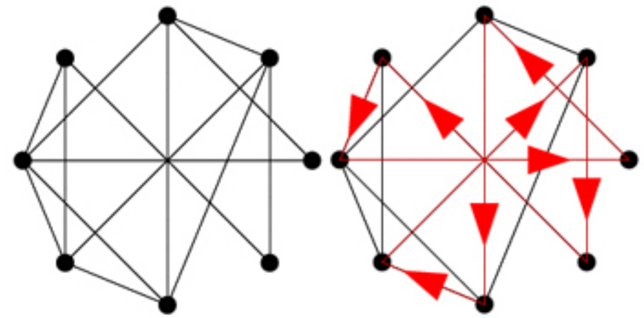
UAS Collision Avoidance, Navigation, and Target Assignment in a Congested Airspace Using Fuzzy Logic



Brandon Cook, Tim Arnett, & Brett Rich

Undergraduate Students, Department of Aerospace Engineering,

AIAA Student Members

Dr. Elad Kivelevitch

Assistant Professor Educator, Department of Aerospace Engineering,

AIAA Senior Member

Outline

- Project Objectives & Motivation
- Simulation Environment Description
- Assumptions
- Algorithms
- Results
- Future Work

UNIVERSITY OF Cincinnati

Project Objectives

- Develop a robust collision avoidance algorithm that allows safe operation of several UAVs in congested airspace
- Assign targets to UAVs using bidding techniques
- Address path planning and navigation issues
- Provide robust solutions in dynamic environments

Motivation

- Finding solutions to these problems allows a better understanding of how multiple UAVs can interact in congested environment and still accomplish their tasks
- Important as UAV applications proliferate (civilian and military)

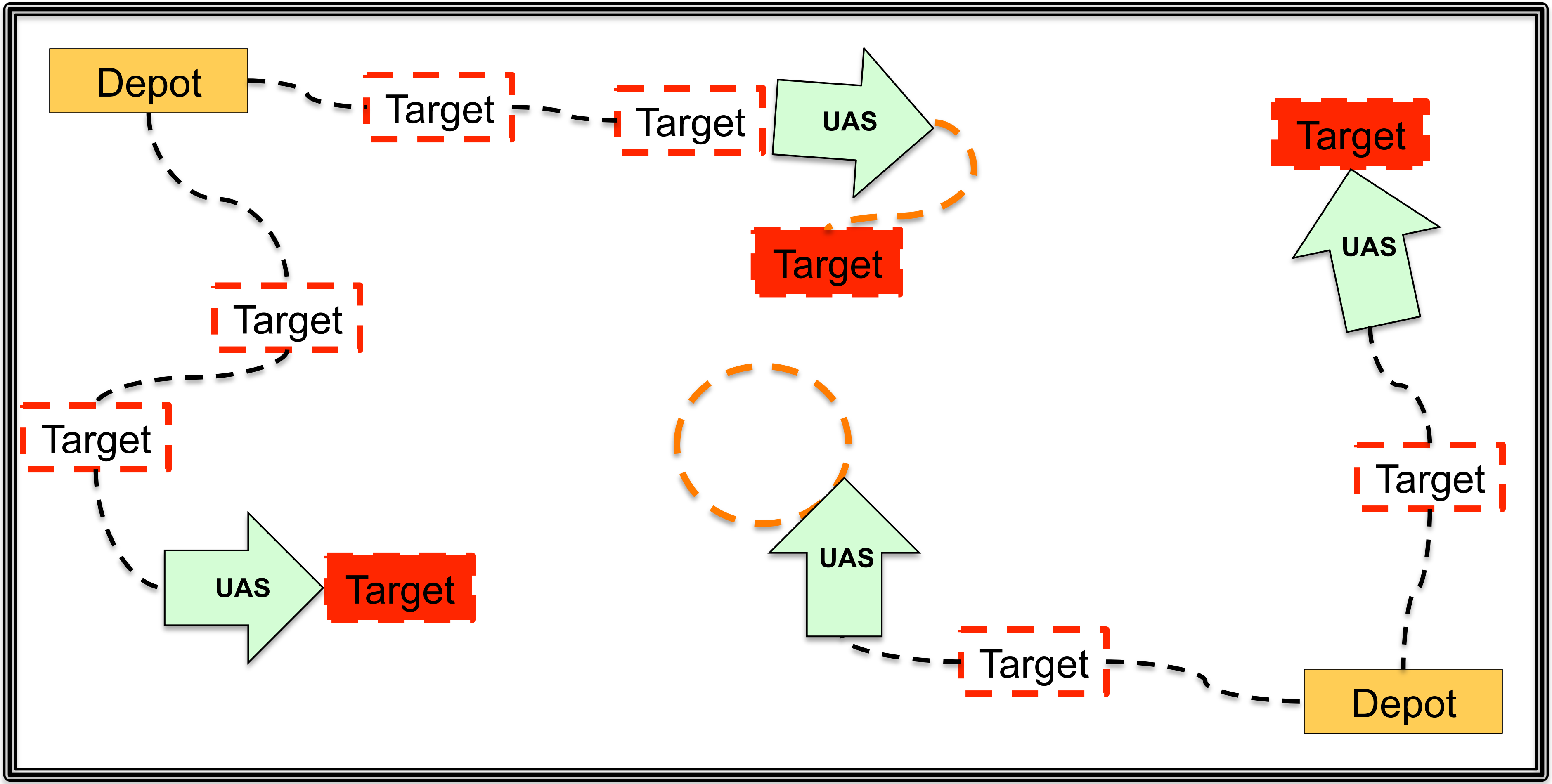
Simulation Environment

- Operational Area: $96\ km^2\ (8 \times 12\ km)$
- Operational depots
  - 2 for non-clustered targets
  - 1 for clustered targets
- 6 UAVs (3 starting at each depot)
- 40 Targets
- UAV Speed: 100 m/s

Mission Objectives

- Acquire and allocate targets to vehicles
- Control vehicle flight paths to targets
- Avoid obstacles throughout simulation
- Visit all targets in airspace

Simulation Layout



Assumptions

- Level, 2D flight – All UAVs at same altitude
- Constant velocity (100 m/s)
- Maximum load factor of $n=3.5$ for all UAVs
- Vehicles limited to
  - Turn radius: $R_{\min} = V^2 / g\sqrt{n^2-1} = 100^2\ m^2/s^2 \,/\, 9.81\ m/s^2\ \sqrt{3.5^2-1} = 300\ m$
  - $\Psi_{max} = g/V\ \sqrt{n^2-1}$
- Target "visited" when UAV within 150 m
- Targets are stationary

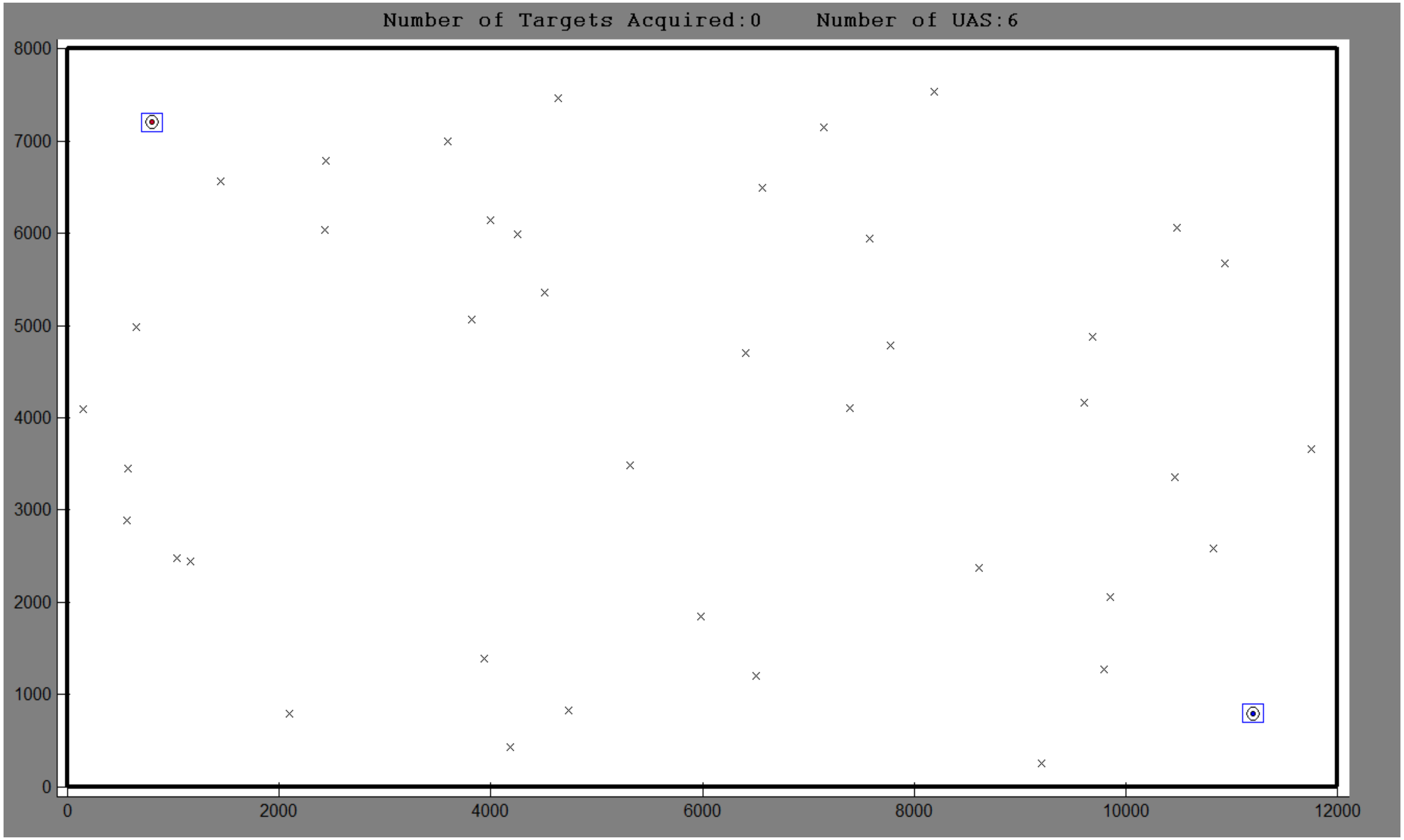
Simulation Cases

- Static Environment
- Cluster-First Static Environment
- Dynamic Environment

- Priority vs. Non-Priority for each environment

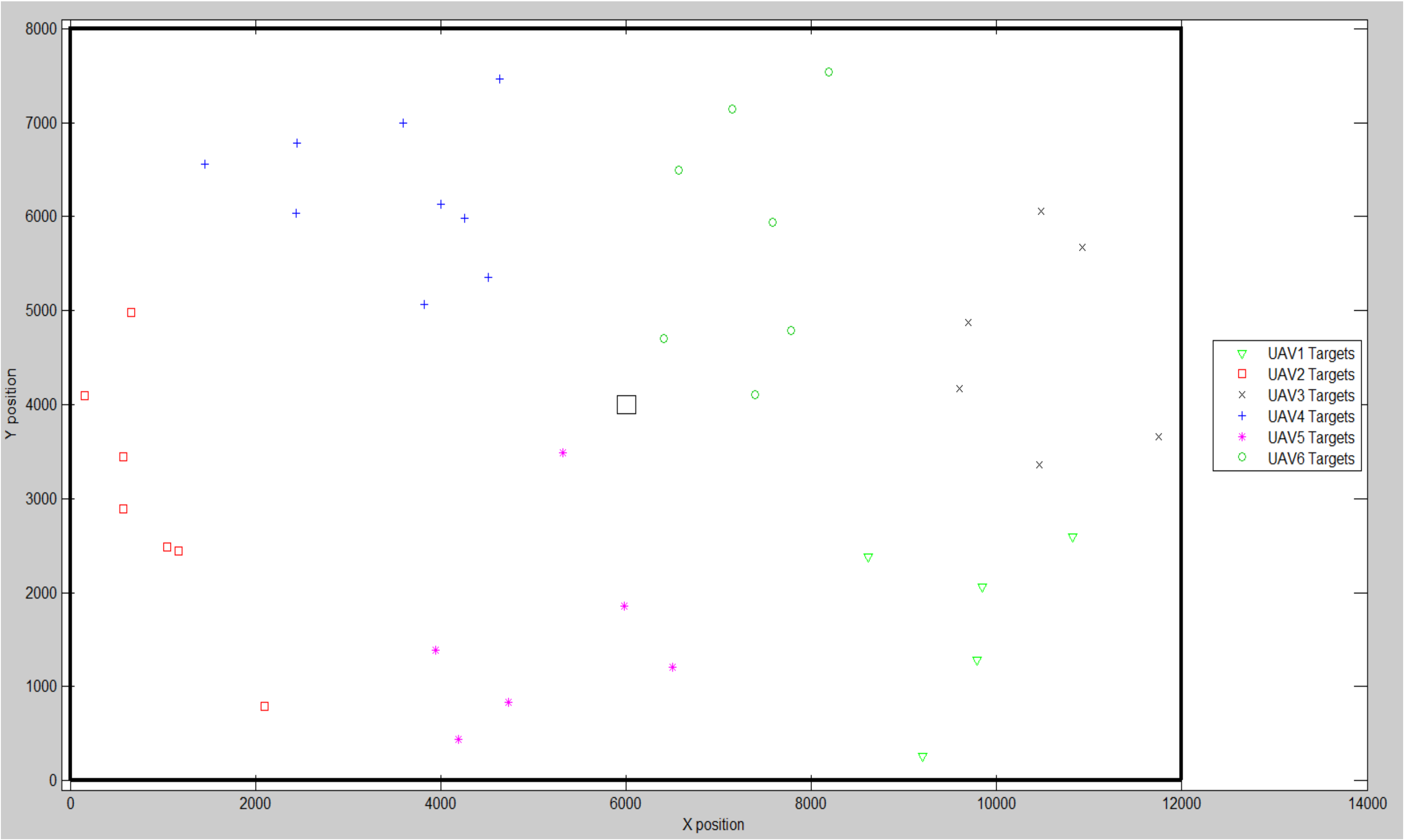
Simulation Cases

Static Environment



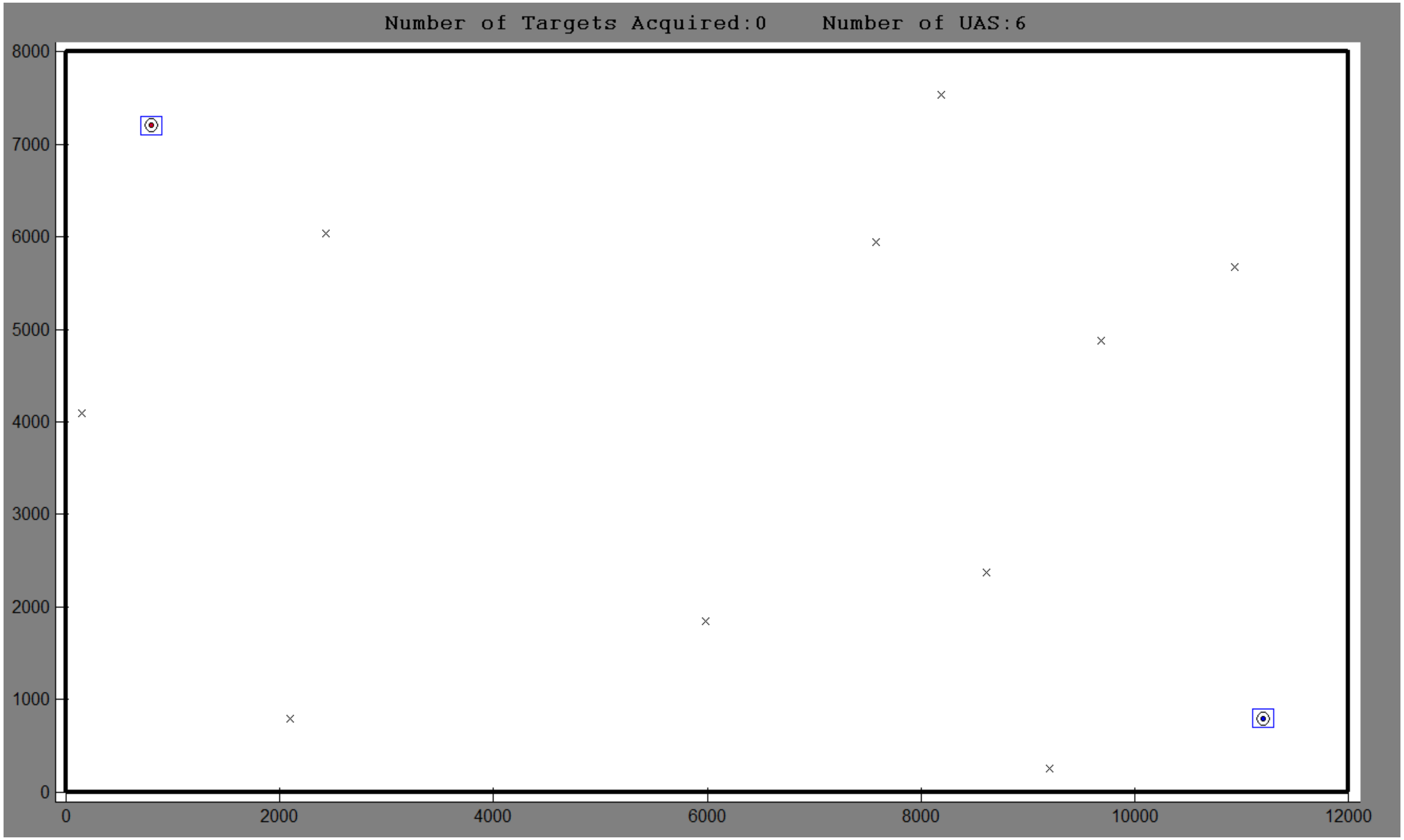
Simulation Cases

Cluster-First Static Environment



Simulation Cases

Dynamic environment



Algorithms

- Navigation
- Target Assignment
  - Cluster-First Techniques
- Vehicle Prioritization
- Collision Avoidance

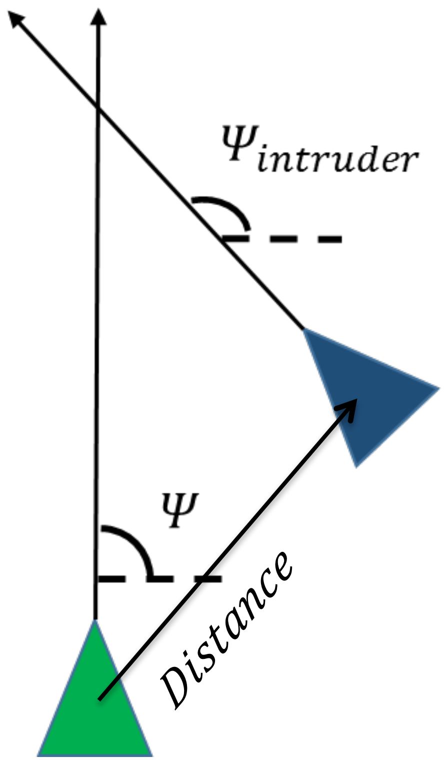
Algorithms

Collision Detection

Step 1	• Populate distance matrix
Step 2	• Move UAVs
Step 3	• Update distance matrix
Step 4	• Check if distance is lower than threshold
Step 5	• If threshold was met, check if distance increased/decreased
Step 6	• If distance decreased check priority ranking
Step 7	• If priority ranking is lower, run obstacle avoidance FIS
Step 8	• Return to step 1

Algorithms

Collision Avoidance



$$\Psi_{rel} = \Psi_{intruder} - \Psi$$

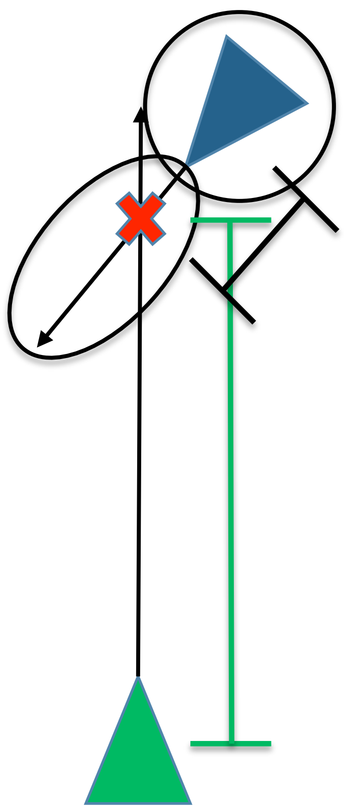
- Provides heading control for vehicles when in range of obstacles (180⁰ Field of View)
  - Overrides target navigation
- Fuzzy system developed based on two inputs:
  - $\Psi{\downarrow}rel$
  - $Distance$
- Outputs Ψ for vehicle
- If multiple intruders exist, average Ψ output contributions

Algorithms
Collision Avoidance – Filtering Techniques

- Developed two Fuzzy Inference Systems
  - Go In Front
  - Go Behind
- Both have identical inputs, outputs, and membership functions
- Use different rule sets (each have 2 rules)
- Need to decide which FIS to use using filtering techniques

Algorithms

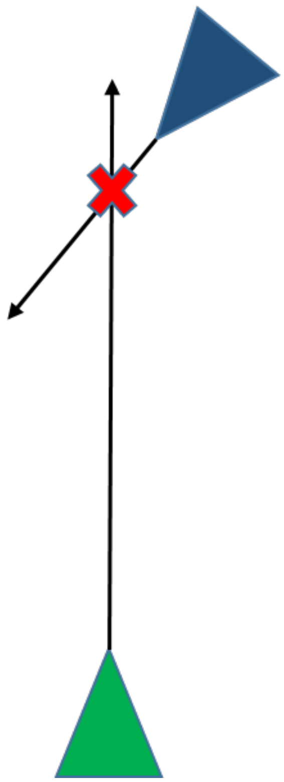
Collision Avoidance – Filtering Techniques



- **Conflict Scenario Parameters**
  - Location of intruder
  - Direction of travel relative to “ownship”
  - Distance to heading intersection point

Algorithms

Collision Avoidance – Filtering Techniques

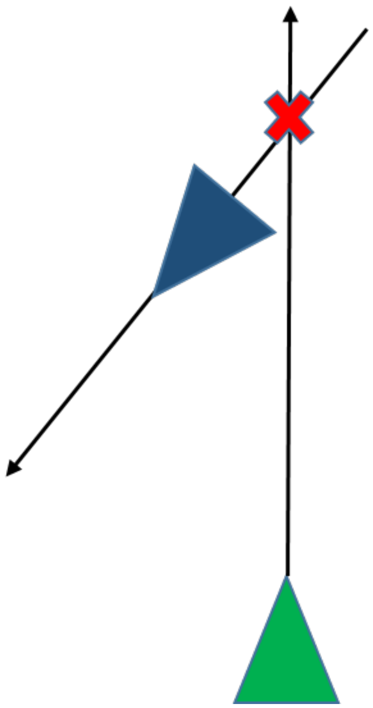


- **Conflict Scenario #1**
  - On Right
  - Going Left
  - Closer to Him

Resolution: Go Behind

Algorithms

Collision Avoidance – Filtering Techniques

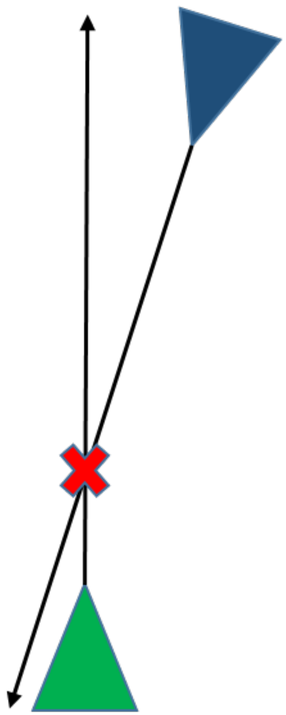


- **Conflict Scenario #2**
  - On Left
  - Going Left
  - Closer to Him

Resolution: Go Behind

Algorithms

Collision Avoidance – Filtering Techniques

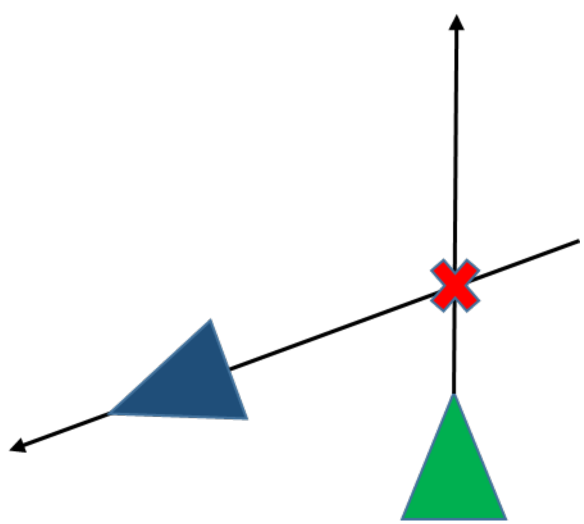


- **Conflict Scenario #3**
  - On Right
  - Going Left
  - Closer to Me

Resolution: Go In Front

Algorithms

Collision Avoidance – Filtering Techniques

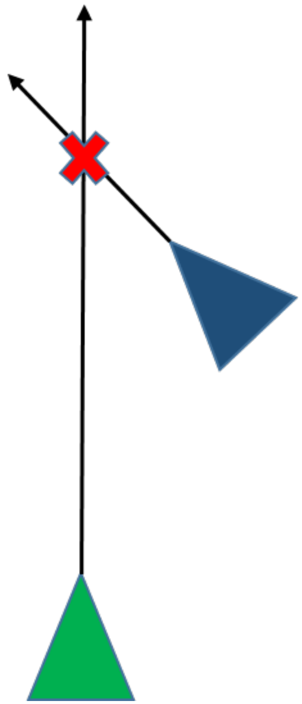


- **Conflict Scenario #4**
  - On Left
  - Going Left
  - Closer to Me

Resolution: Go Behind

Algorithms

Collision Avoidance – Filtering Techniques

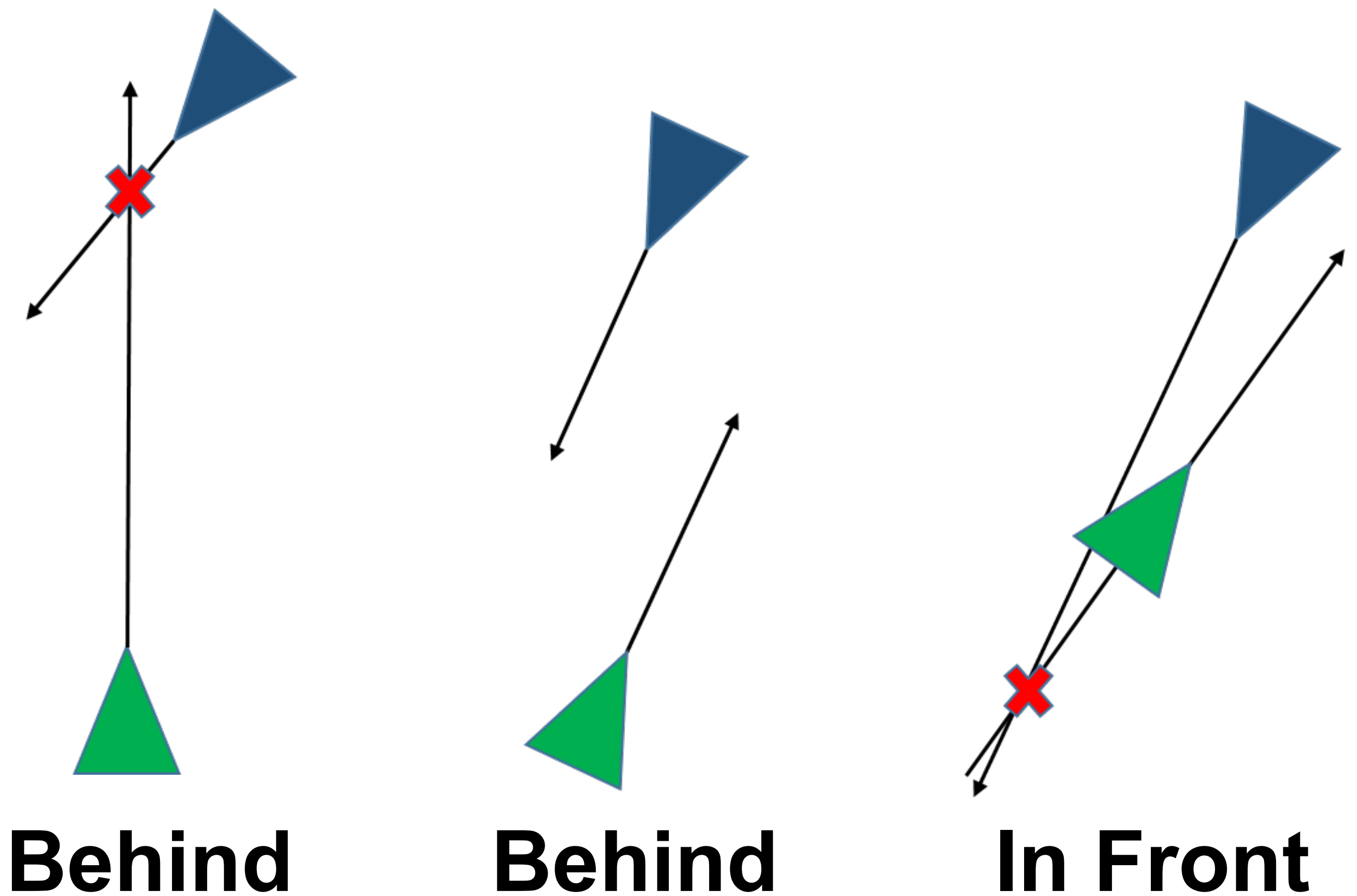


- **Conflict Scenario #5**
  - On Right
  - Going Left
  - Closer to Him

NOTE: Similar Heading

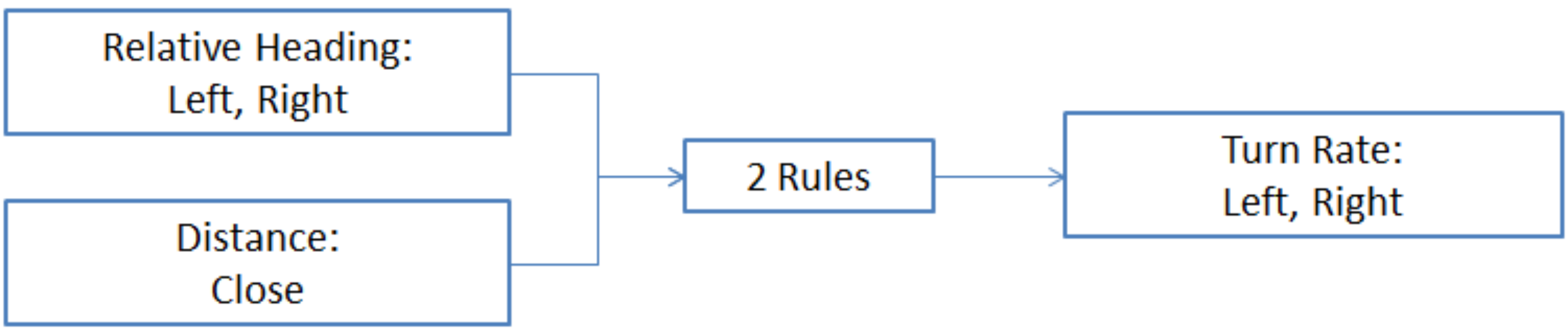
Resolution: Go Behind

Algorithms
Collision Avoidance – Filtering Techniques



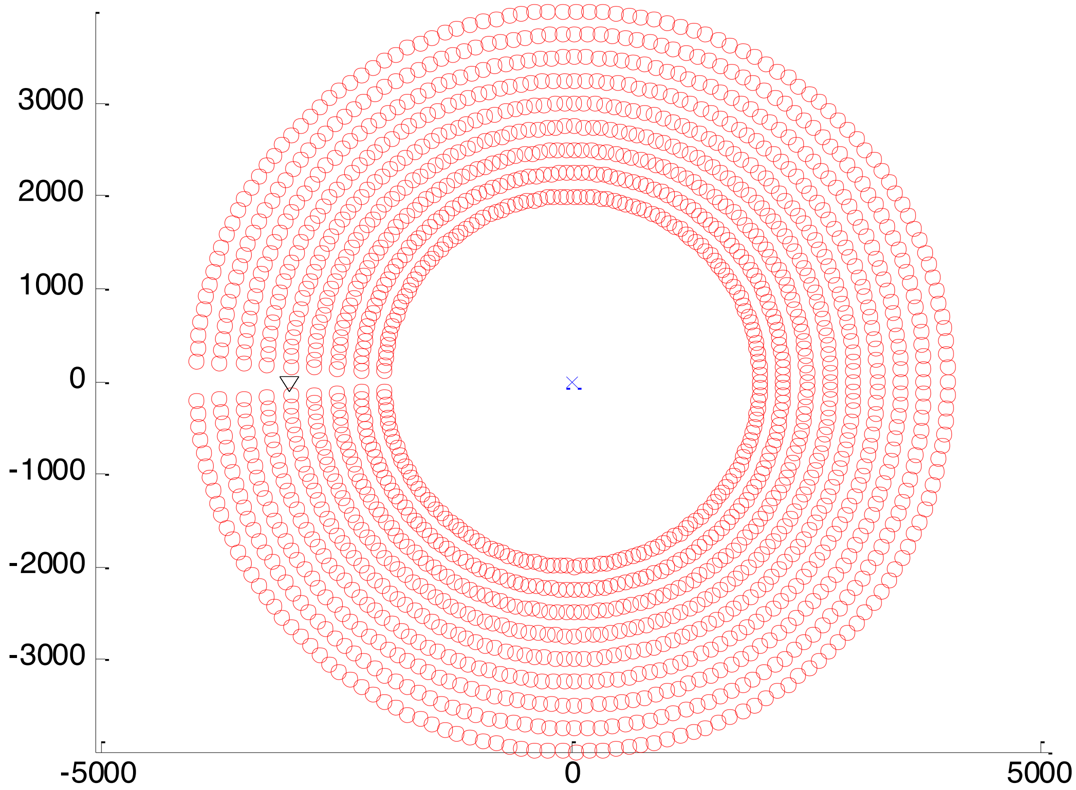
Algorithms

Collision Avoidance – FIS Logic

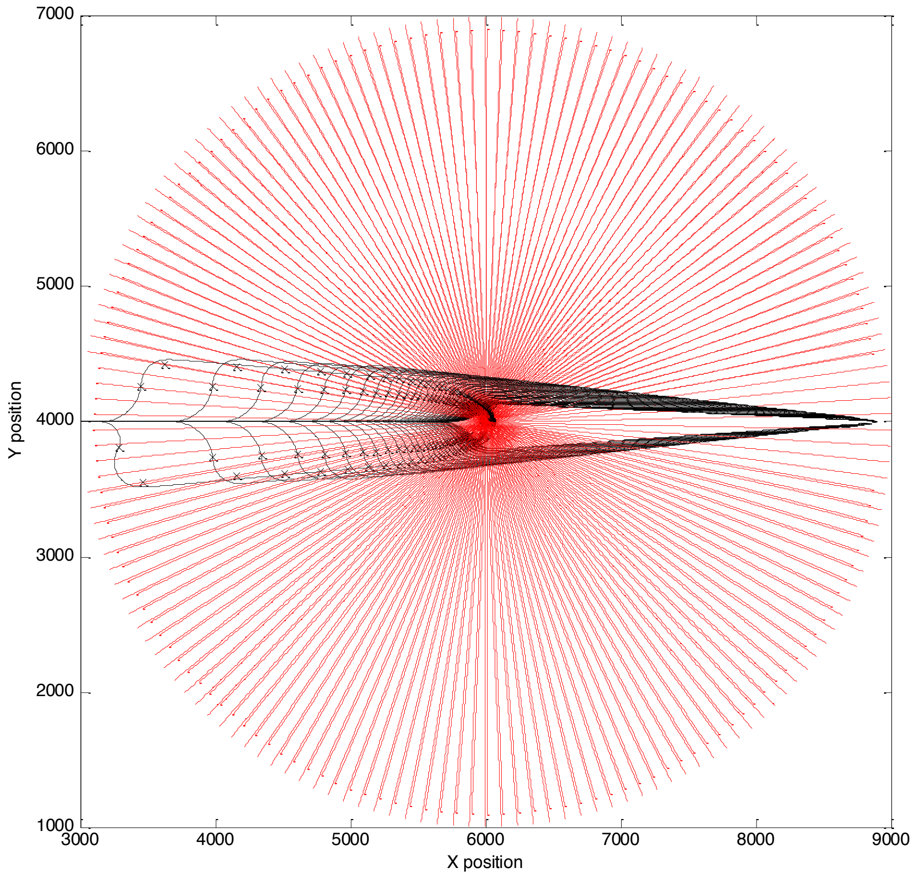


Algorithm Testing

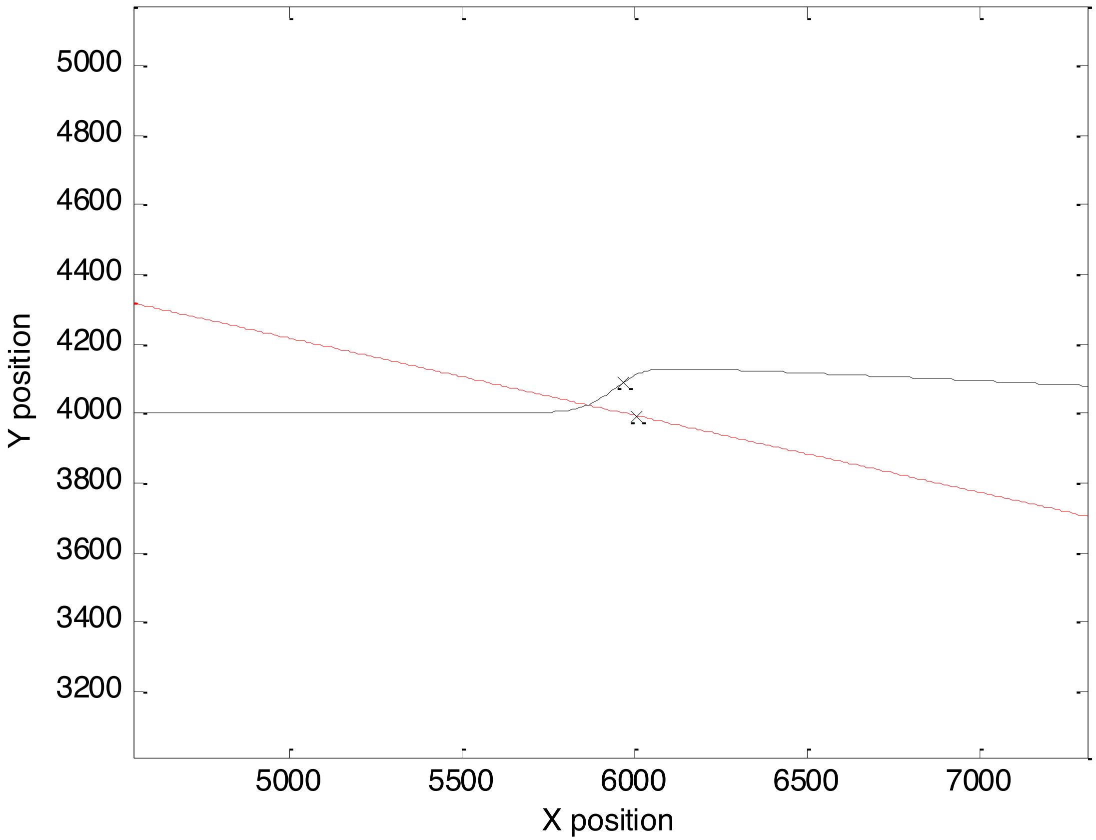
Collision Avoidance



Algorithm Testing
Collision Avoidance – Priority

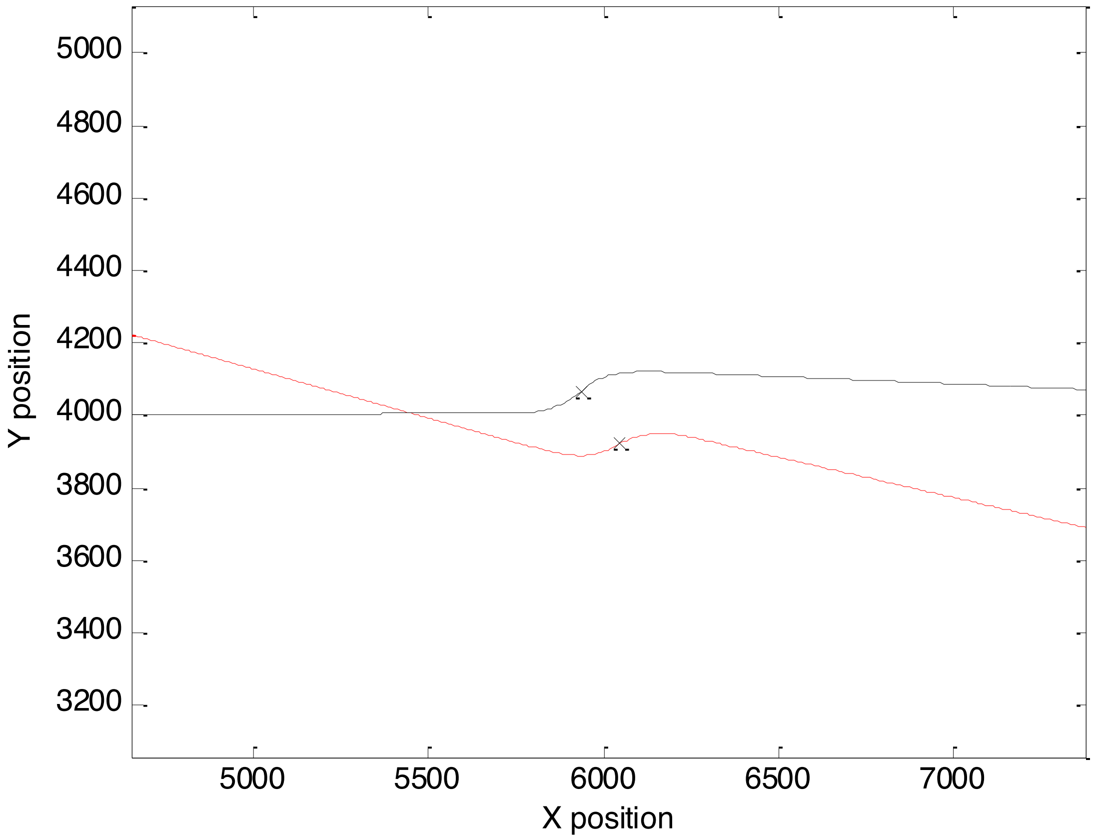


Algorithm Testing
Collision Avoidance – Priority



Algorithm Testing

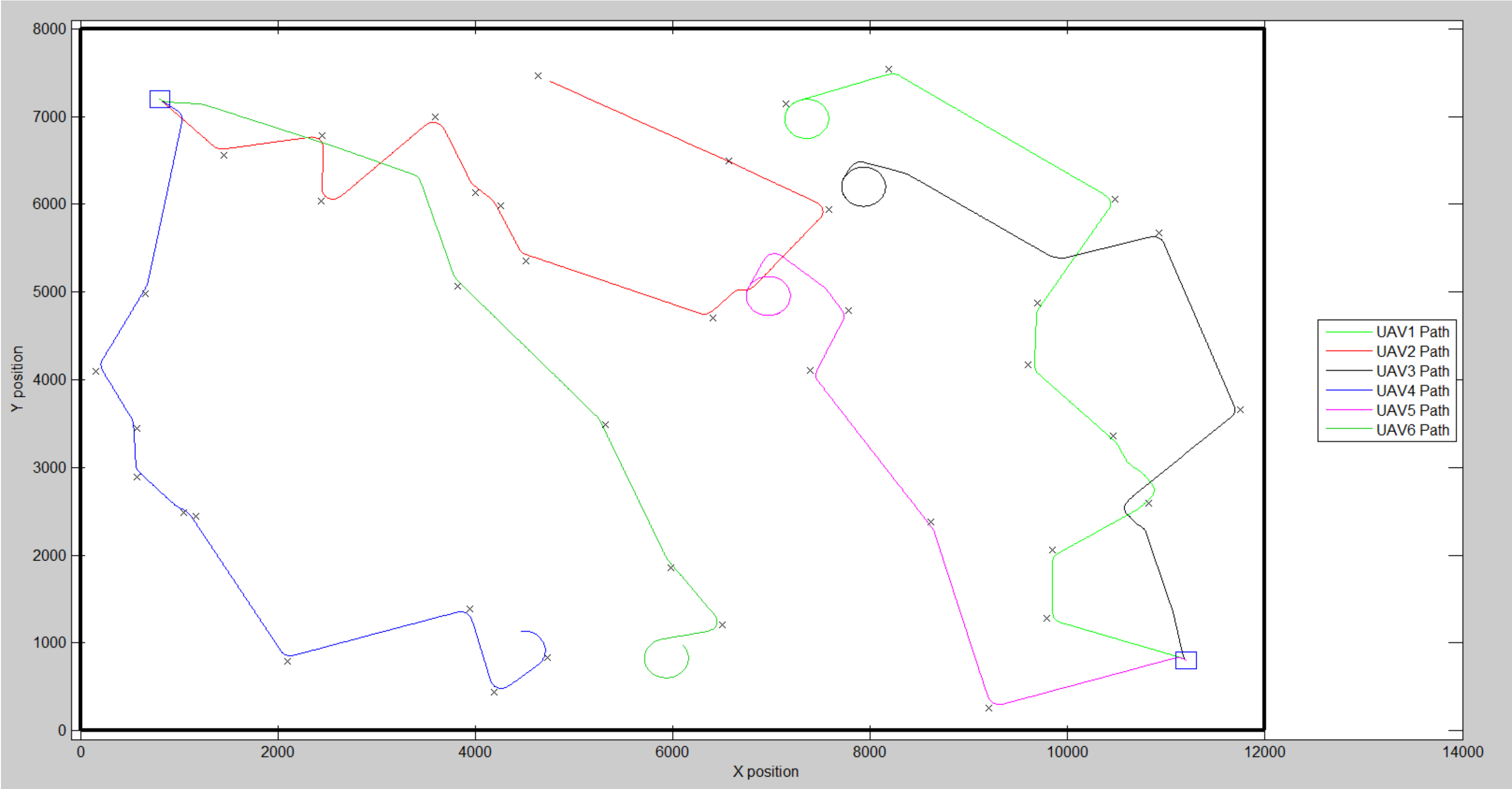
Collision Avoidance – No Priority



Results

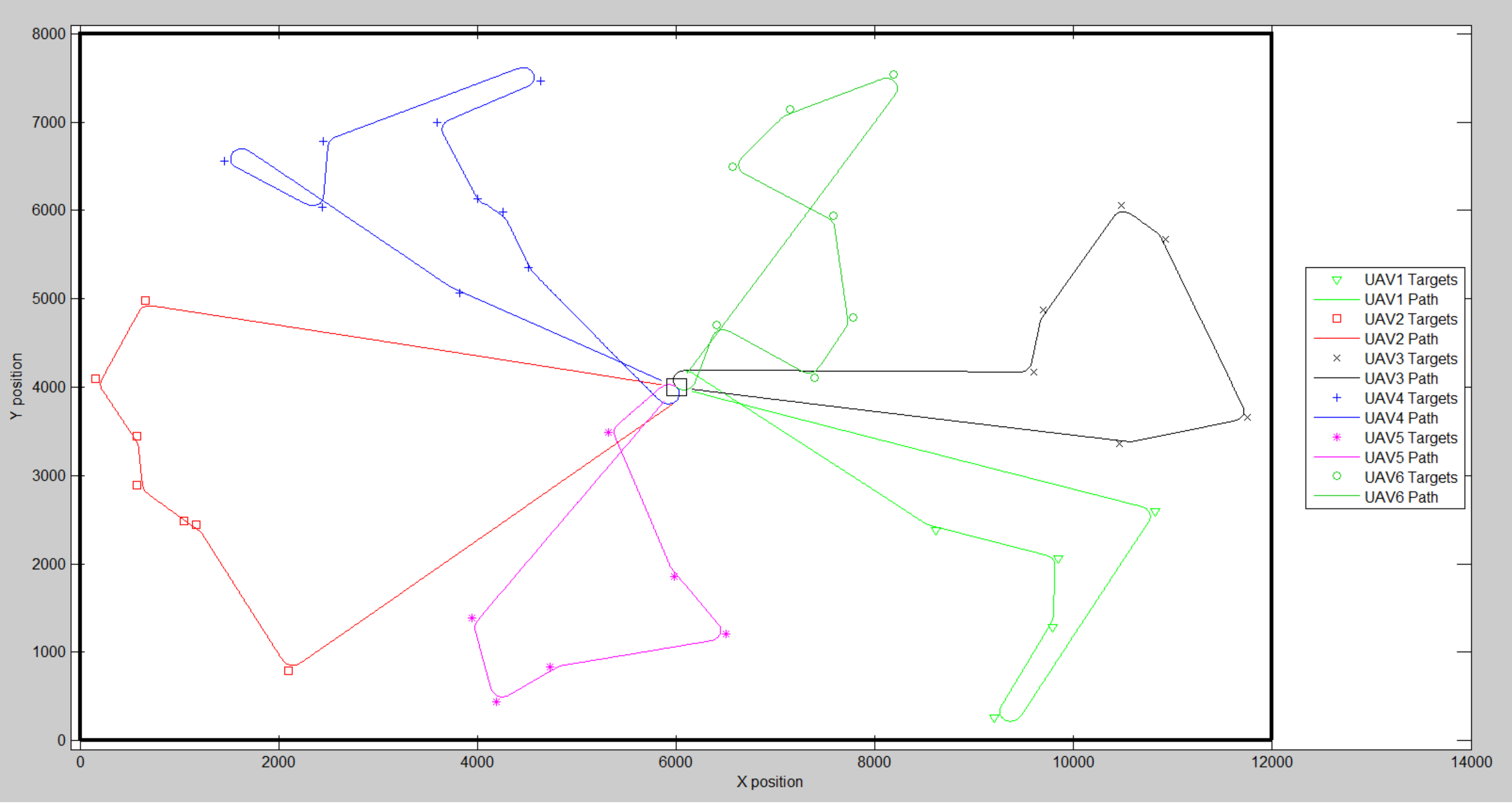
Results

Static Environment



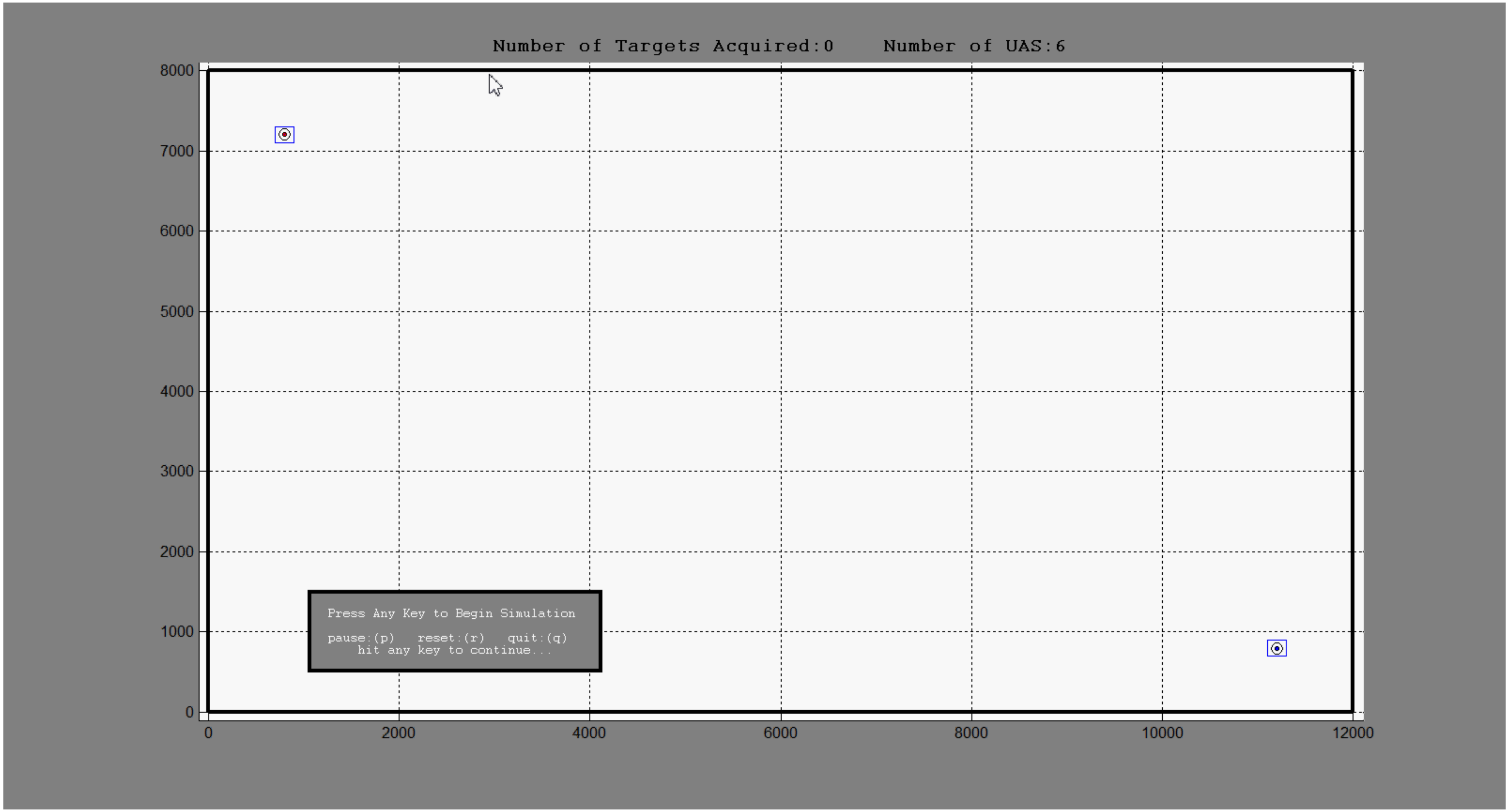
Results

Cluster First



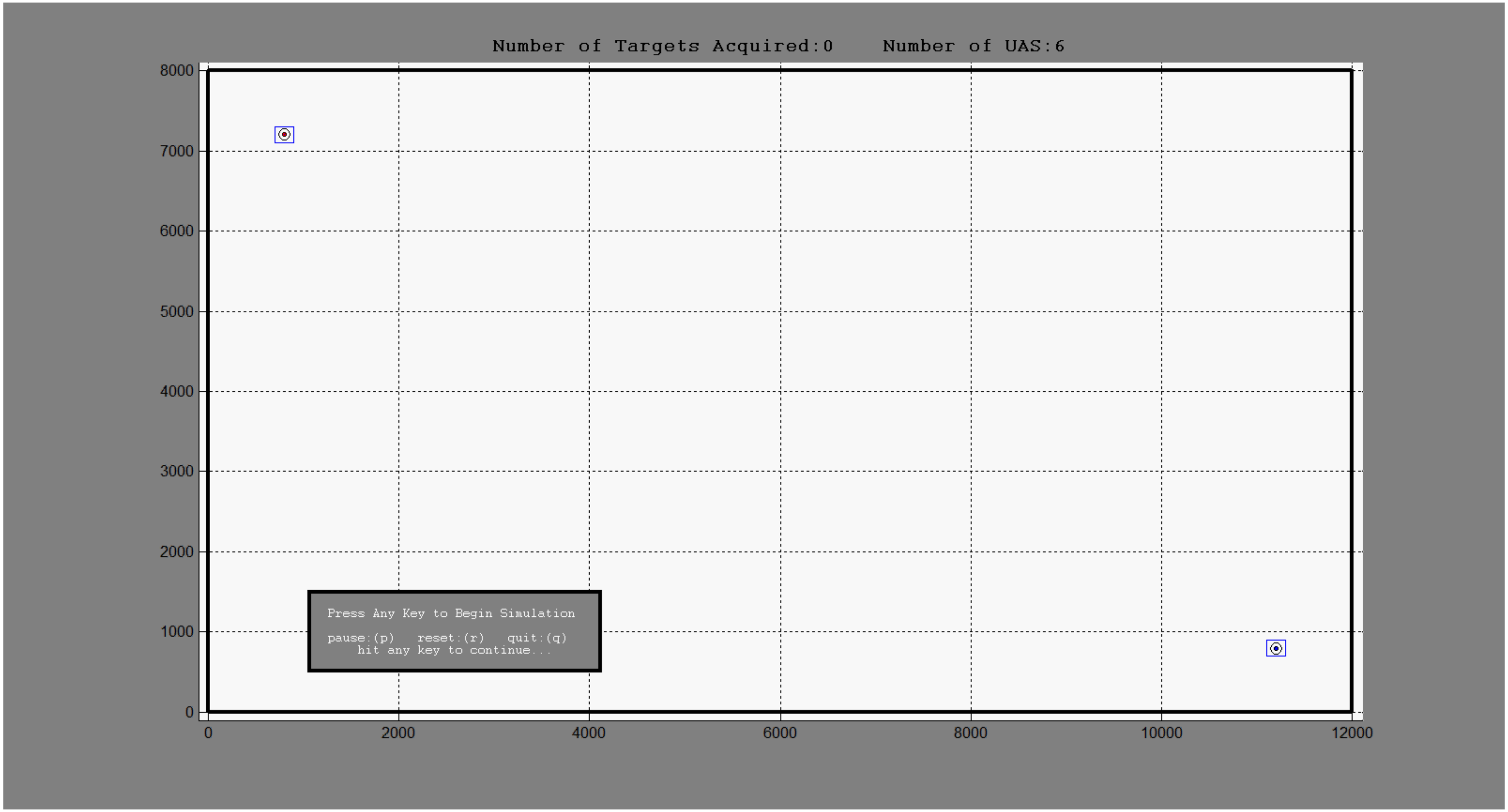
Results

Static Environment



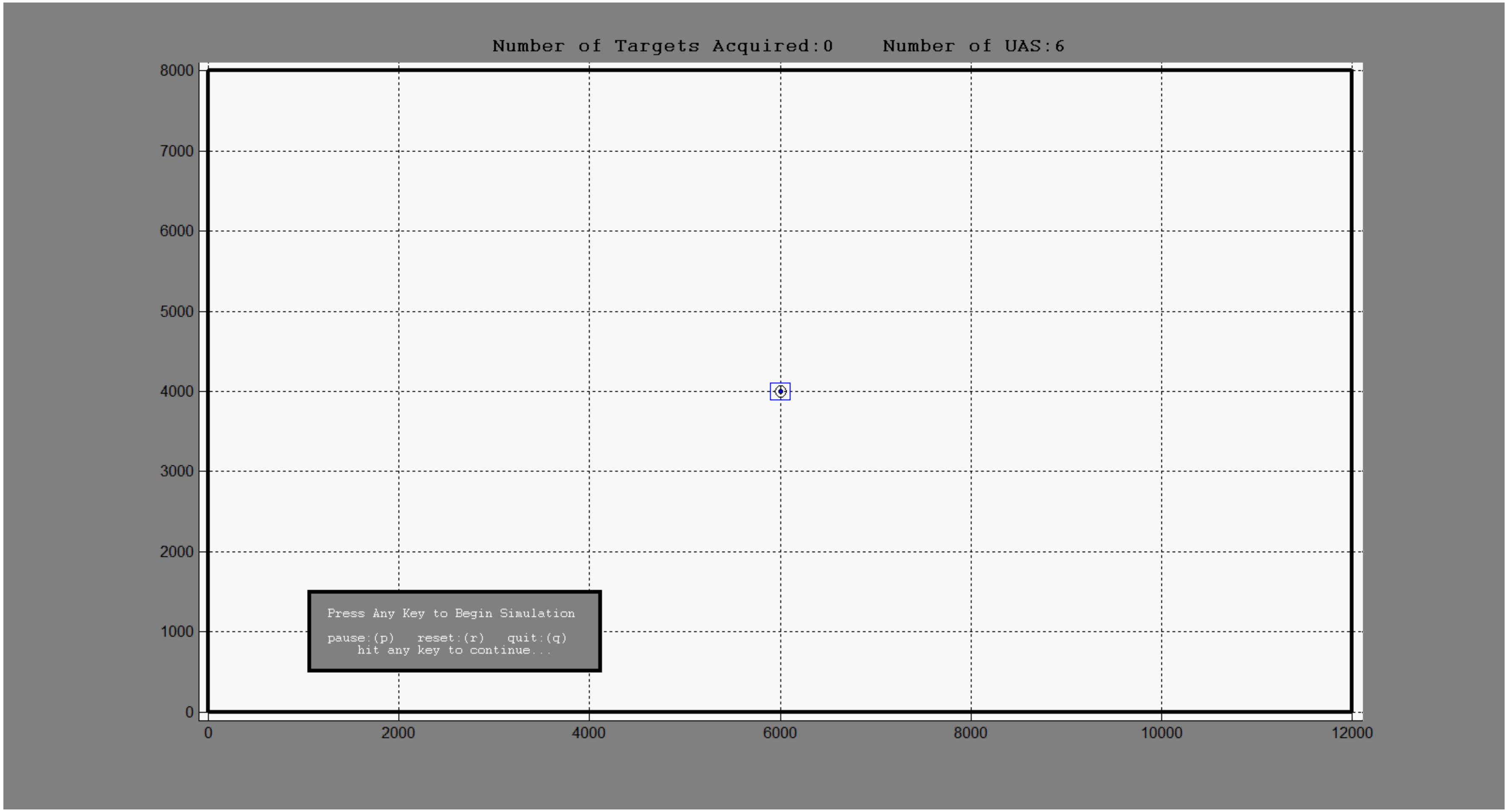
Results

Dynamic Environment



Results

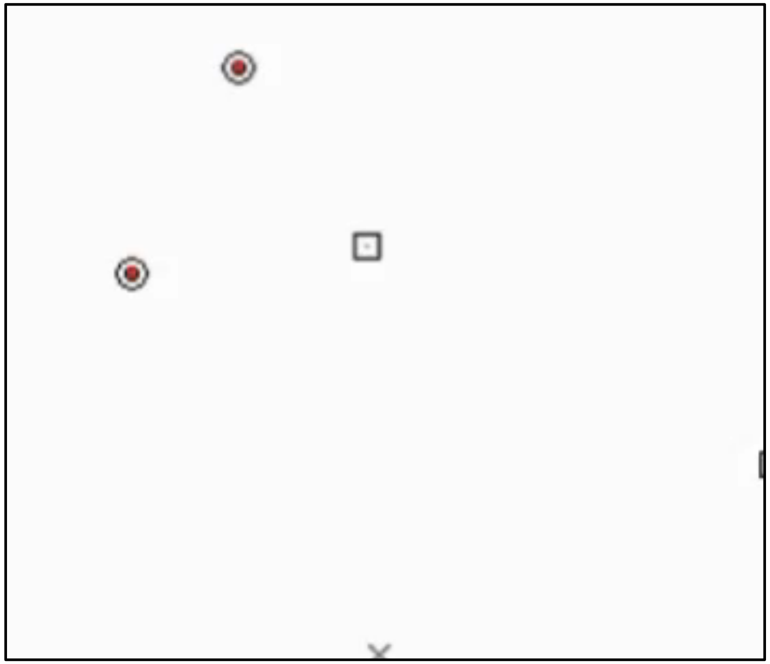
Cluster First



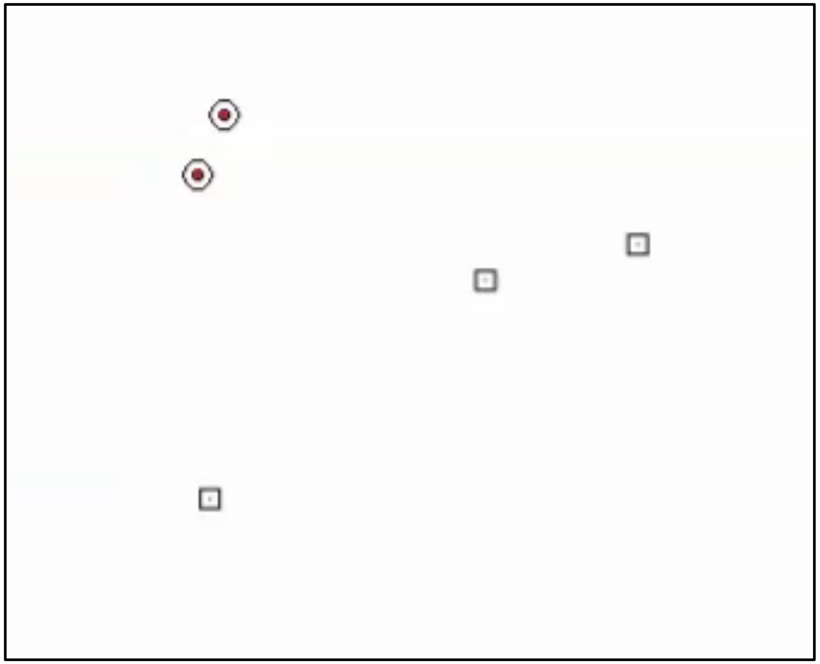
Results

Avoidance

Non-Prioritized Vehicles

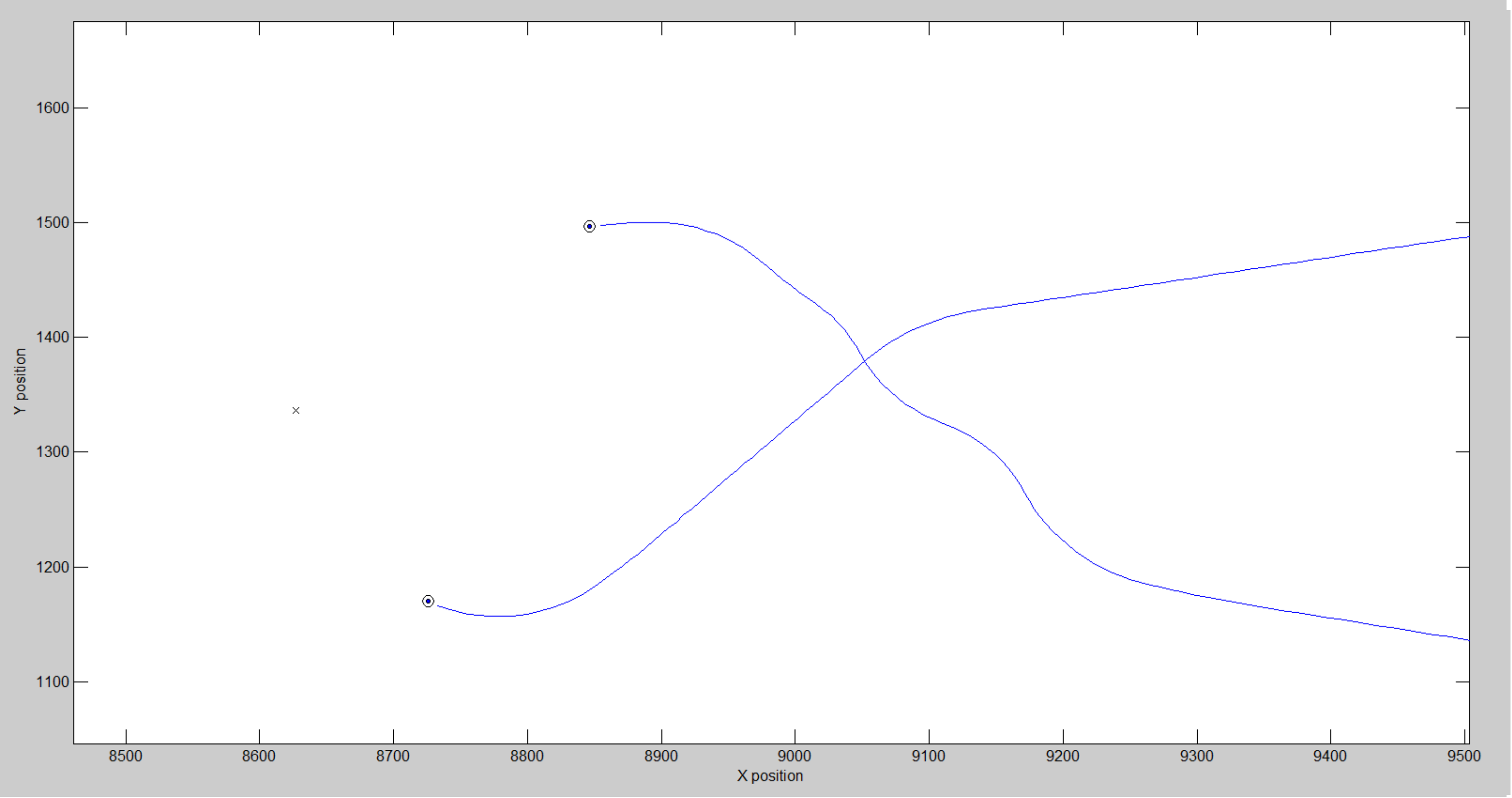


Prioritized Vehicles



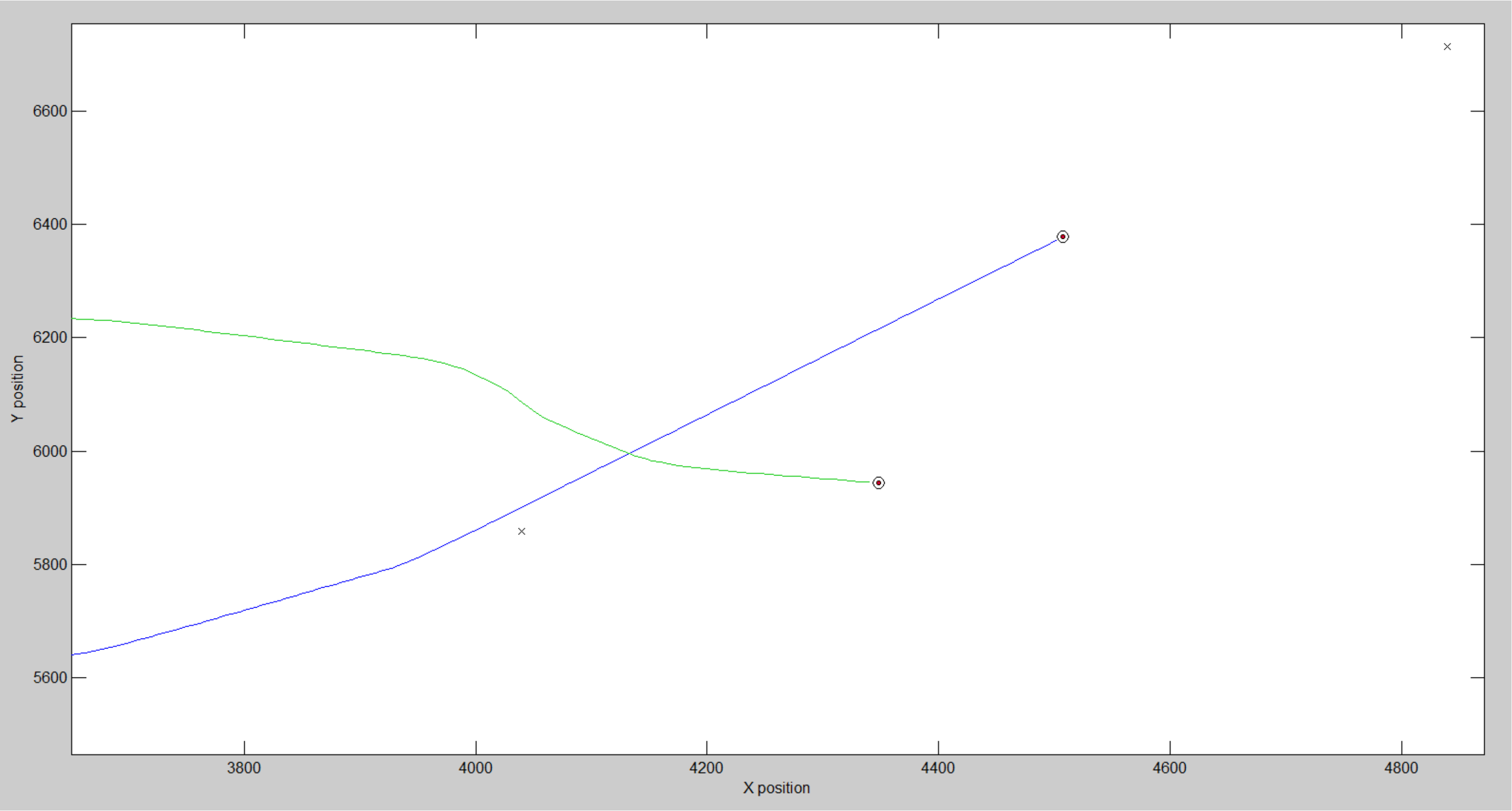
Results

Avoidance – Non-Prioritized Vehicles



Results

Avoidance – Prioritized Vehicles



Results

Case	Total Flight Hours	Collisions	Collisions/ Flight Hour
Static – Priority	837.6	0	0
Static – No Priority	843.7	0	0
Dynamic – Priority	1599.9	0	0
Dynamic – No Priority	1637.8	0	0
Cluster – Priority	1297.9	0	0
Cluster – No Priority	1291.8	0	0

Conclusions

- Navigation FIS drives error to zero with little overshoot and smooth output
- Bidding based purely on distance not the best solution overall
  - Robust to changing number of targets such as the dynamic case shown
  - Purposefully chosen to increase probability of collisions
- Verified that using Fuzzy Logic for collision avoidance provides a robust solution to dynamic environments
  - Has few and simple inputs and rules
  - Had 100% success rate for all simulation environments and testing (7,509 Flight Hours)
  - Ability to adapt to a dynamic environment

Conclusions

- Fuzzy logic is an effective tool for collaboration between autonomous agents in a time-critical spatio-temporal environment
- A completely autonomous, robotic, intelligent swarm would be useful in applications including:
  - Safe Integration of UAVs into the National Airspace
  - Space robotics
  - Celestial body exploration and colonization
  - Homeland security
  - Disaster relief programs

Future Work

- Navigation
  - Already robust but could explore taking subsequent targets into account
- Target Assignment/Route Planning
  - Better route planning options (2 opt, Concorde) for within cluster areas and in dynamic environments
- Avoidance
  - Fuzzy avoidance system can be ramped up for heterogeneous systems (different closure rates and vehicle turning radii)
  - Verify for several-agent conflict scenarios (3 or more)
  - Test algorithms on real flight paths
- Environment
  - Add third dimension (altitude)
  - Introduce static obstacles (no fly zones)
  - Communication limitations/dropouts
  - Human Interaction

Acknowledgements

This presentation was made possible in part by a Travel Grant from the Gerald A. Soffen Fund for the Advancement of Space Science Education

For more information, visit
http://SoffenFund.org

Gerald Soffen
Viking Project Scientist

References

1. Mitchell, S., Cook, B., and Cohen, K., “Fuzzy Logic Inferencing for Pong”, *Logic Programming: Theory, Practices and Challenges*. Nova Science, 2014.
2. Sameh, A., "Clustering," *CS500 - Data Mining* [online lecture], URL: http://info.psu.edu.sa/psu/cis/asameh/cs-500/dm13-clustering.ppt [cited 19 April 2014]

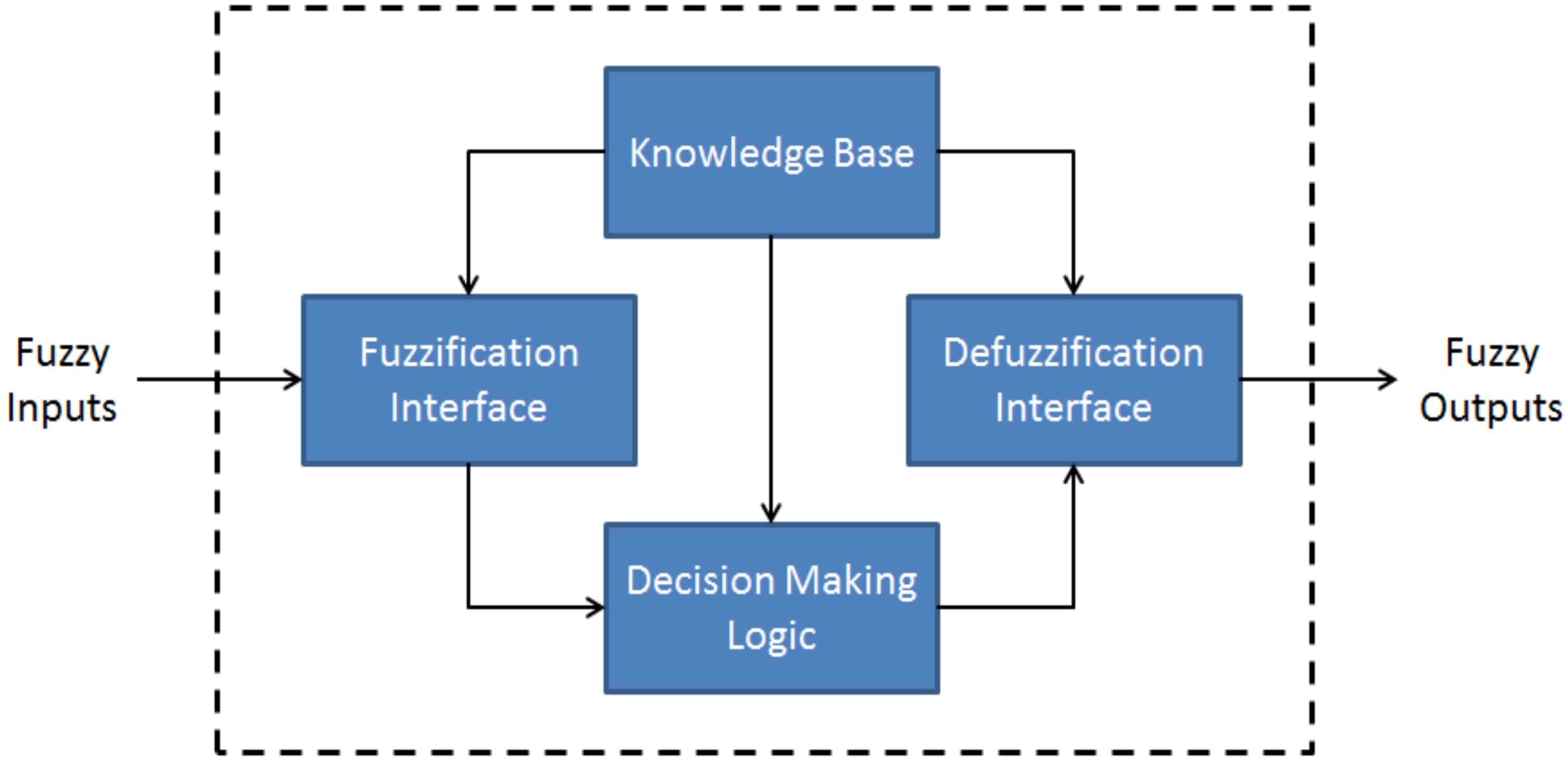
Questions?

Backup Slides

Fuzzy Logic

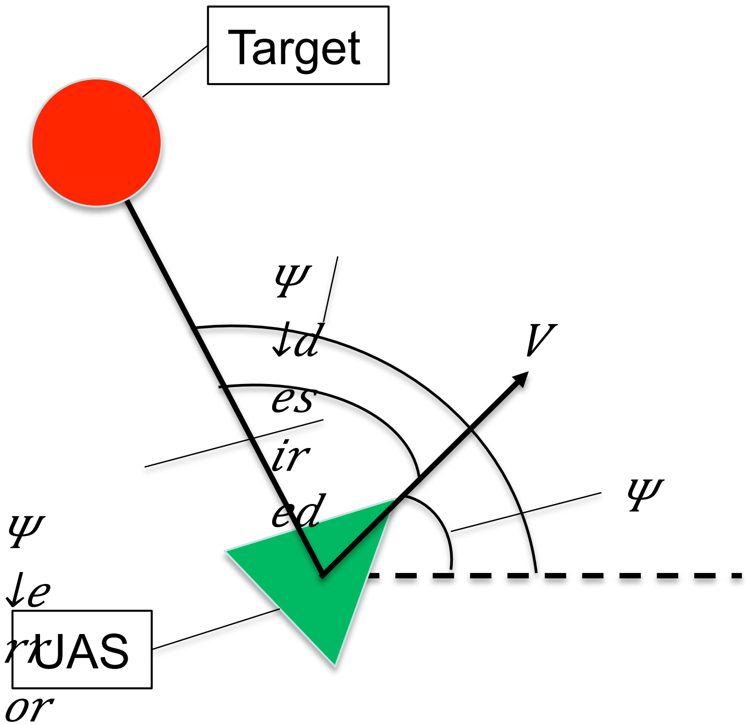
- Allows for decision making in a manner which resembles human-like reasoning
- Utilizes linguistic reasoning
- Common terms
  - Inputs
  - Rules
  - Outputs
  - Membership Function
  - Fuzzy Inference System (FIS)
  - Fuzzy Set

Fuzzy Inference System



[1]

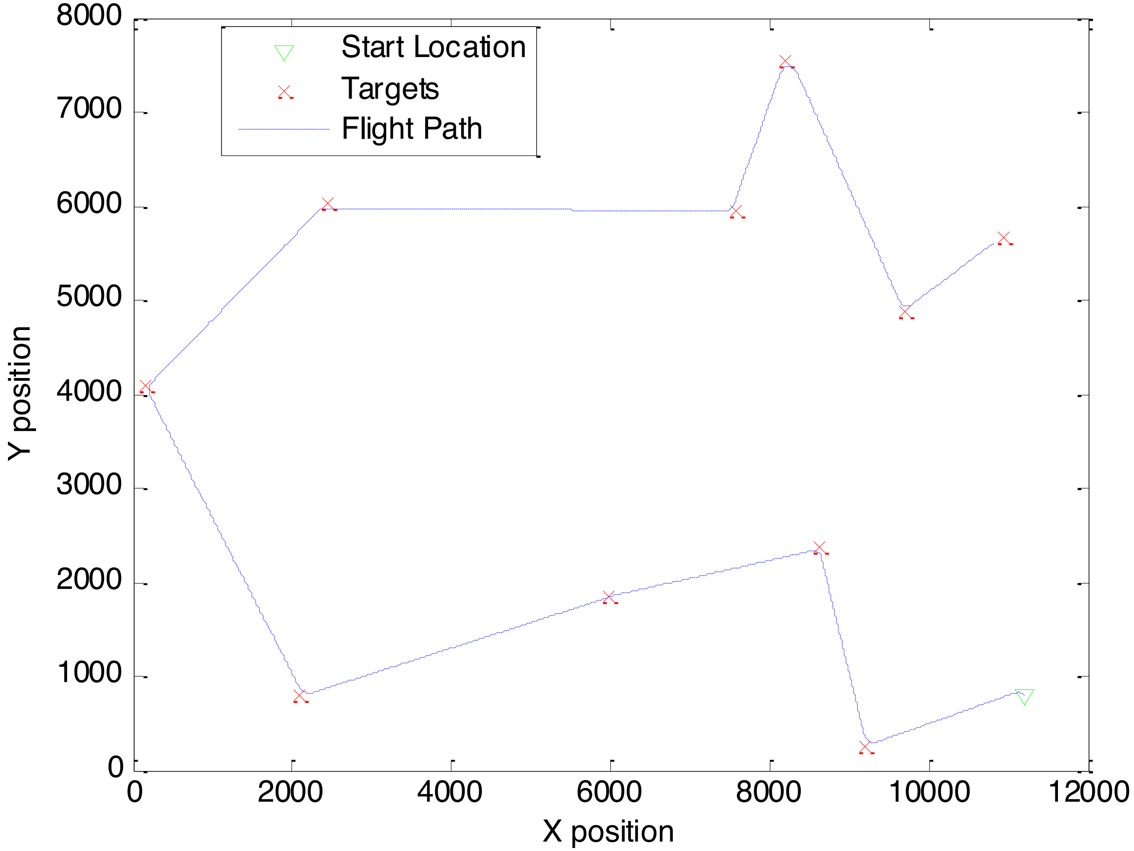
Algorithms
Navigation



- Provides control for vehicle heading towards current target
- Proportional and fuzzy controllers developed
  - Both very effective, chose Fuzzy for sim
  - Based on Ψ_{error} input
- Drives Ψ_{error} to zero

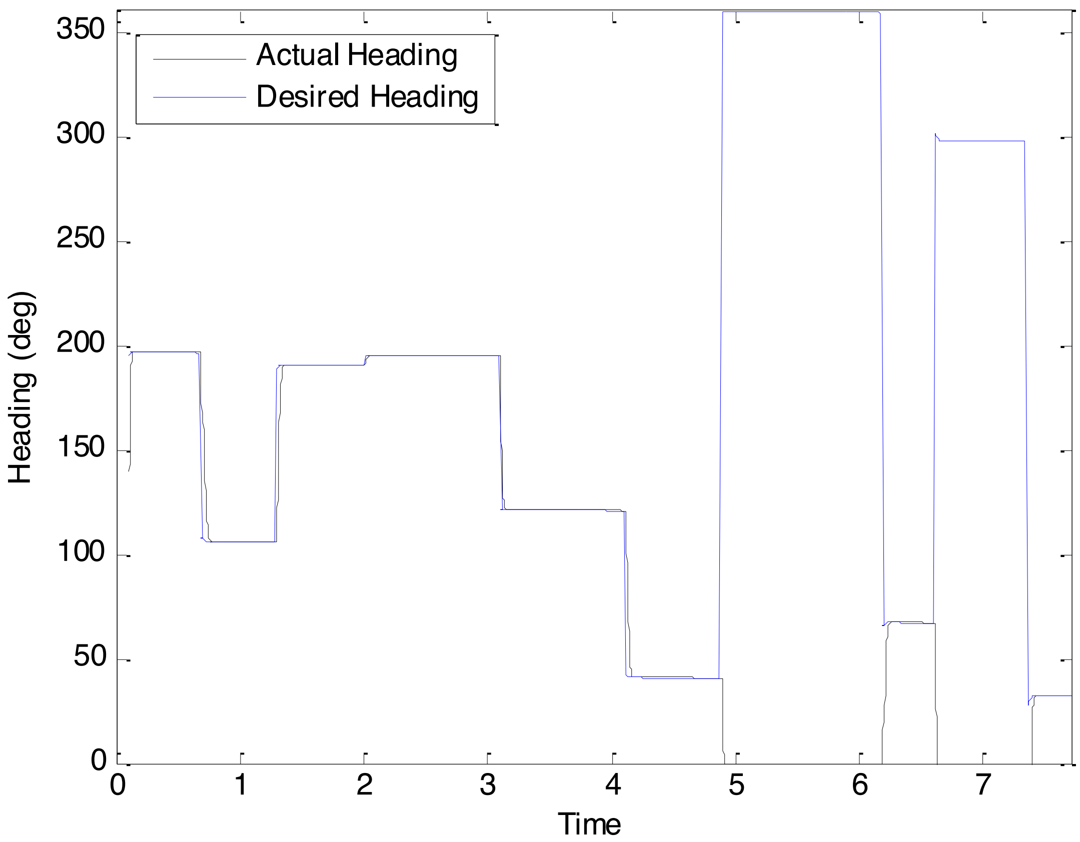
Algorithm Testing

Navigation



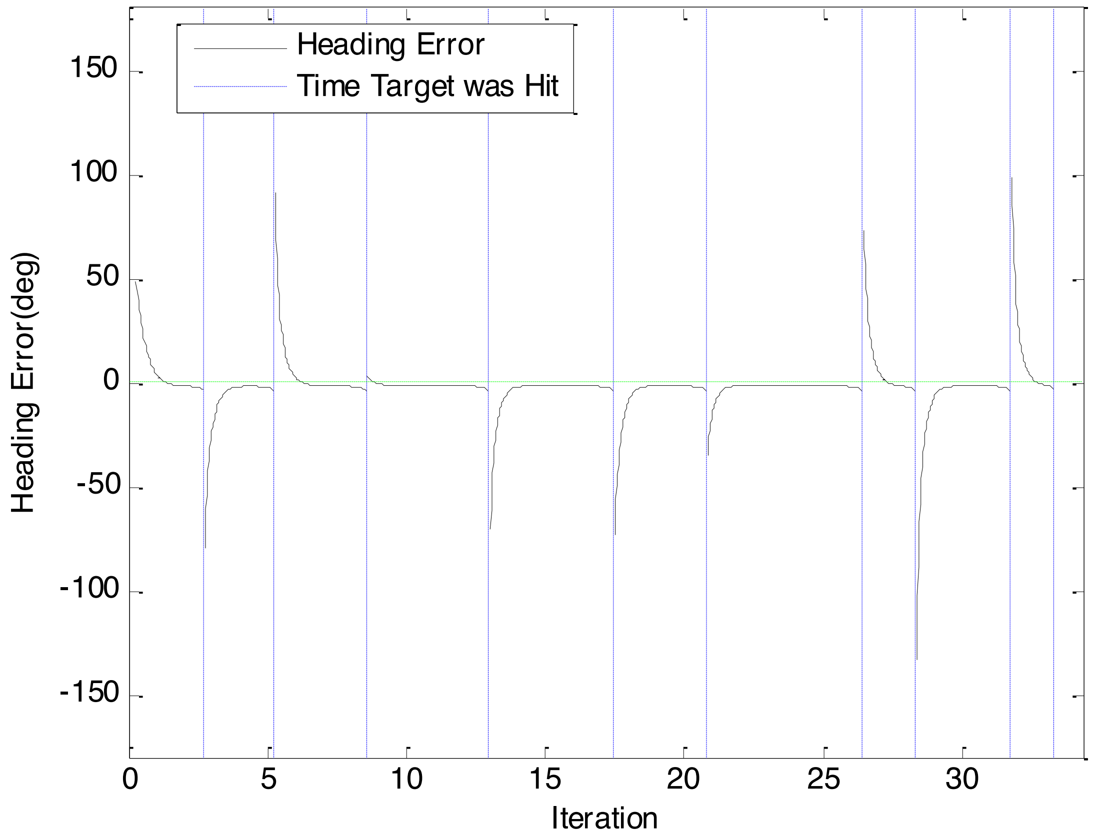
Algorithm Testing

Navigation

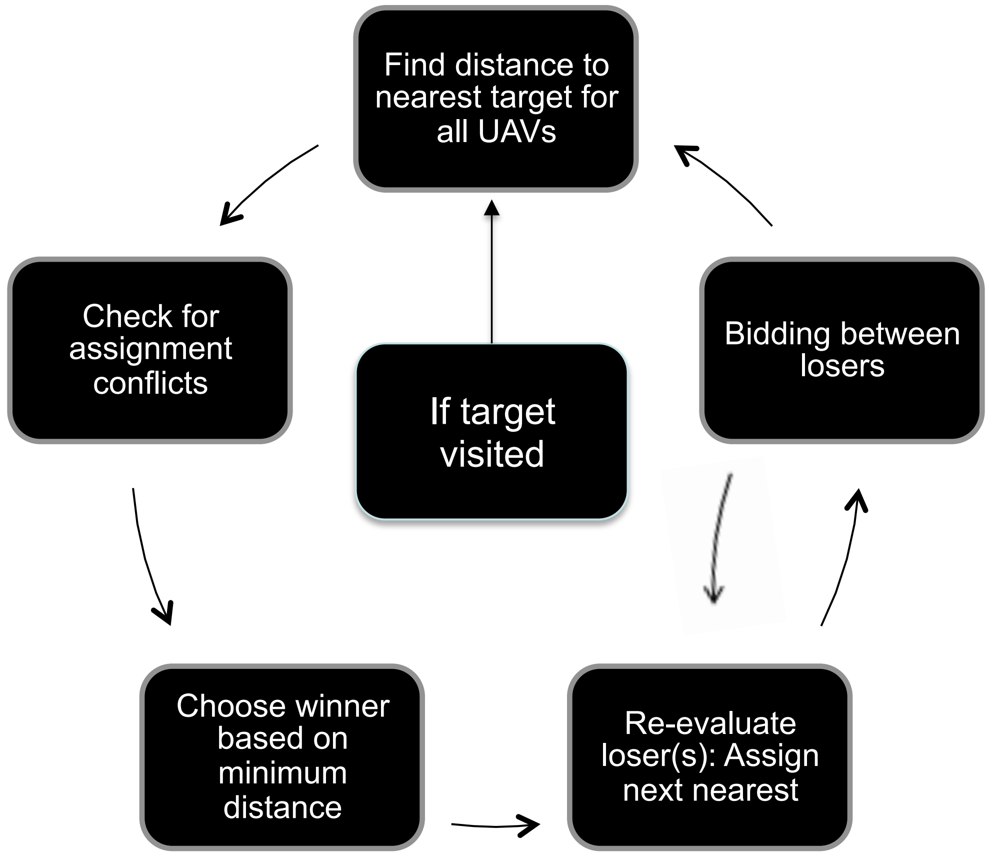


Algorithm Testing

Navigation

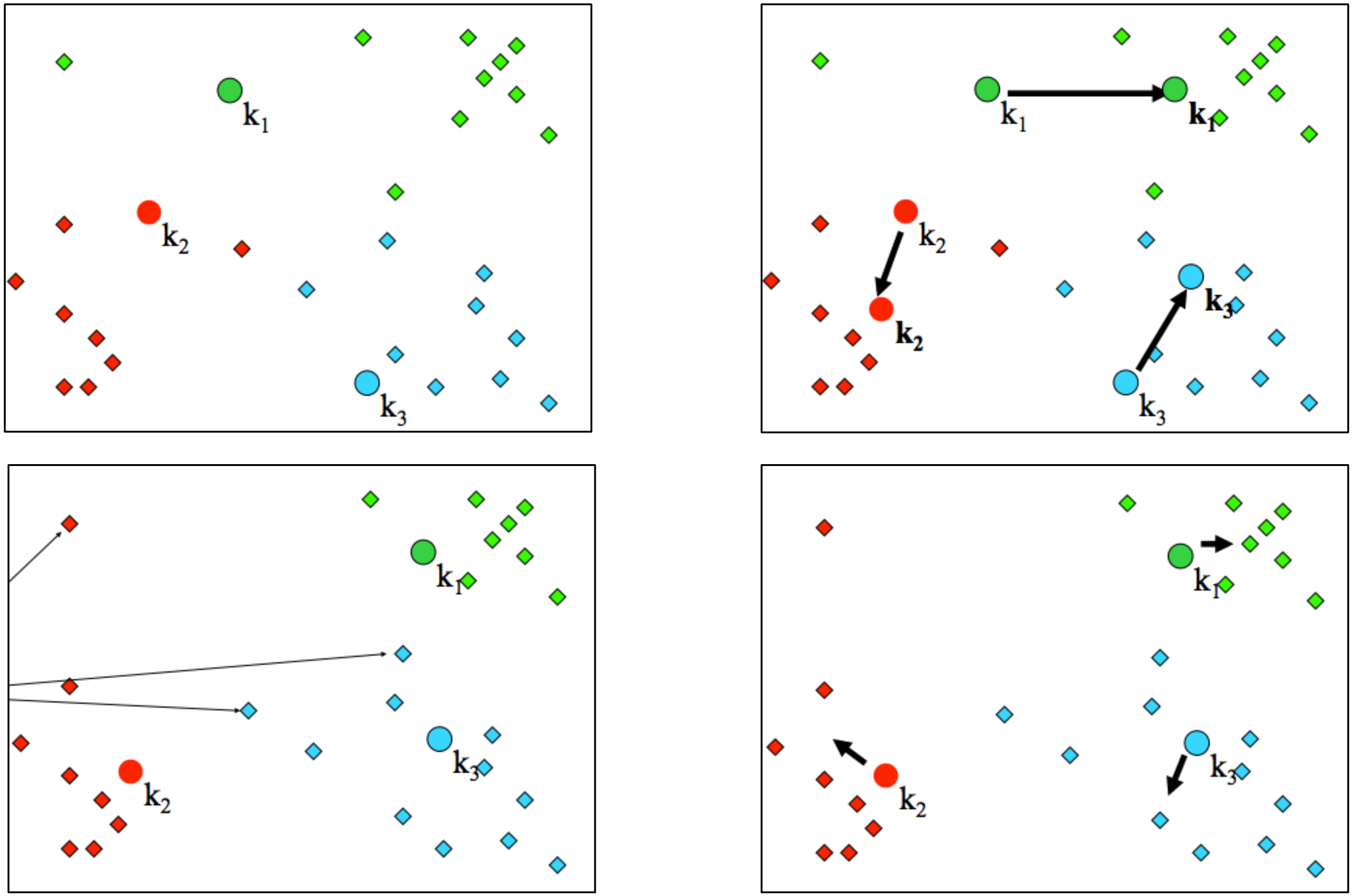


Algorithms
Target Assignment



Algorithms

Cluster-First Technique - KMEANS



[2]

Algorithms

Priority Assignment

- UAV closest to its target gets assigned highest priority
  - Does not ever deviate from its flight path
- Next closest UAV is assigned second highest priority (etc.)
- Lowest priority avoids ALL other UAVs

Algorithms

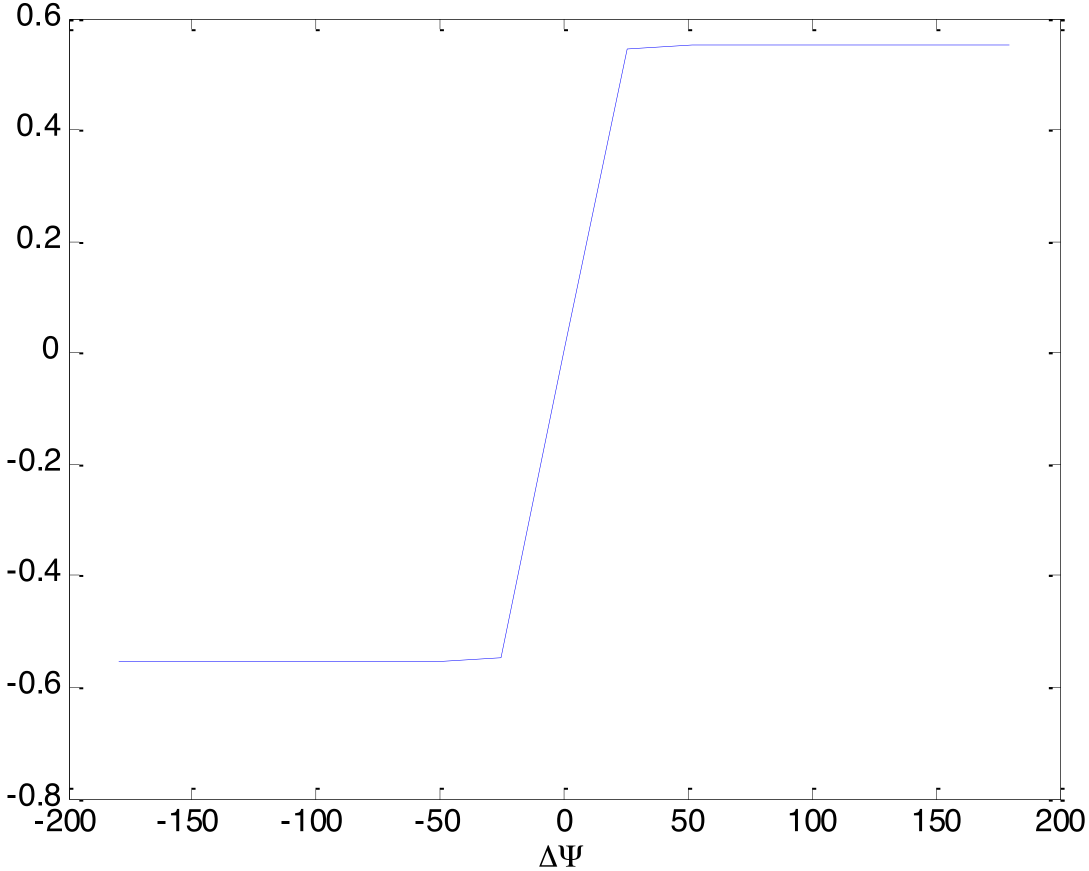
Navigation– FIS Logic

Heading Error:
Far Right, Right, None, Left, Far Left

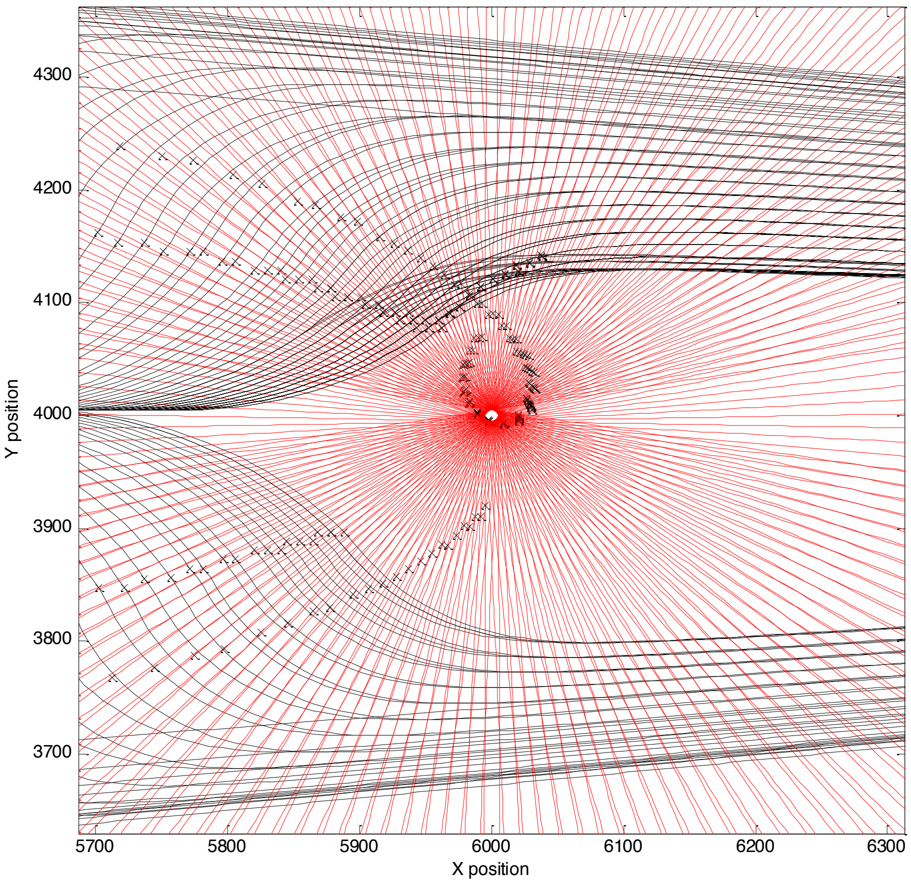
→ 5 Rules →

Turn Rate:
Hard Left, Left, None, Right, Hard Right

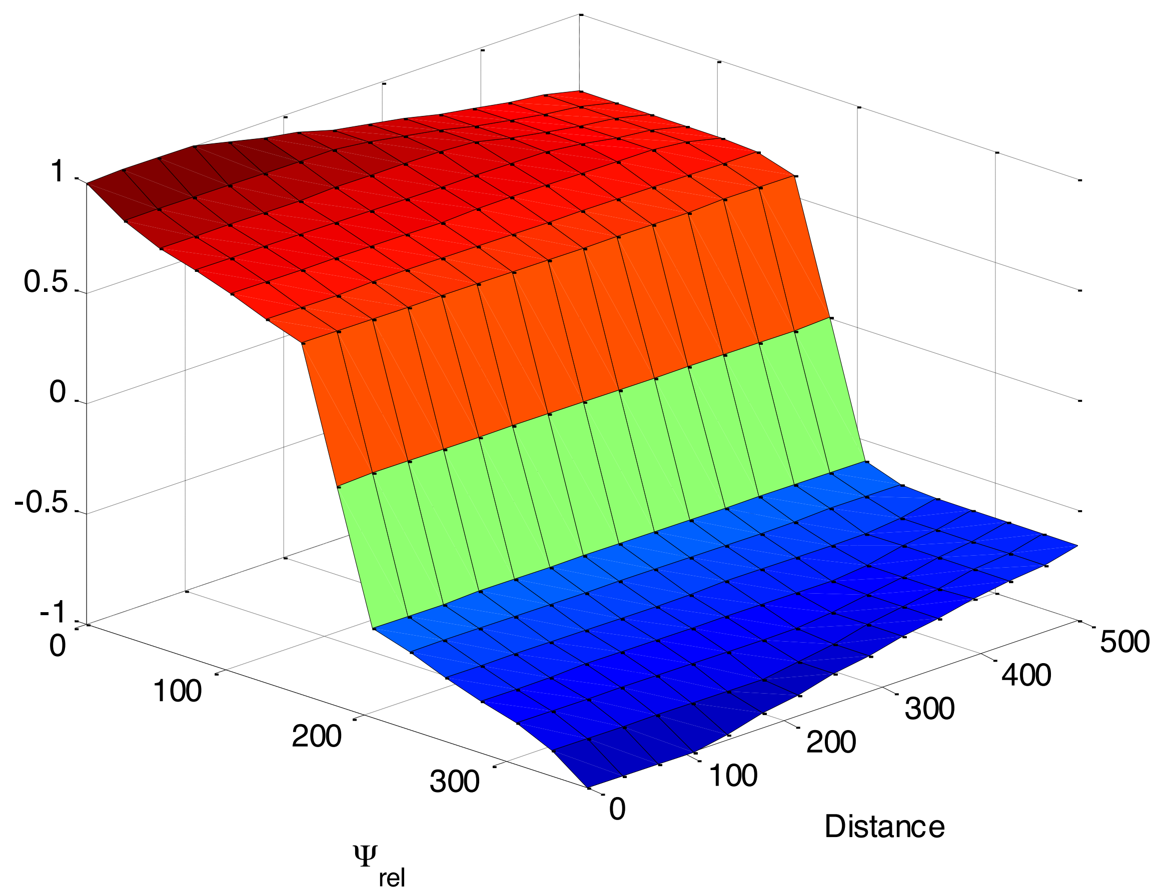
Algorithms
Navigation – FIS Surface



Algorithm Testing
Collision Avoidance – Priority

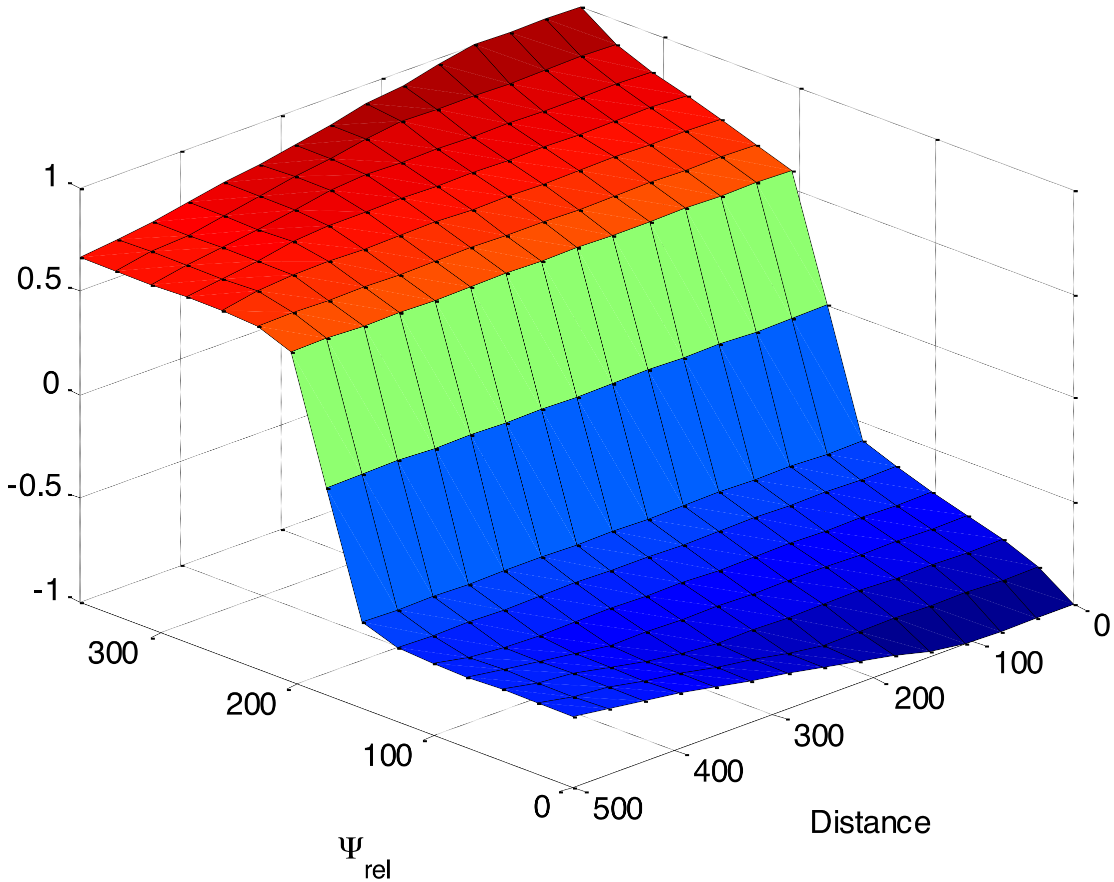


Algorithms
Collision Avoidance – Go Behind Surface



Algorithms

Collision Avoidance – Go In Front Surface



Results

Static Environment & Prioritized Vehicles