Cognitive Agents for Autonomous Robots

University of Dayton

Air Force Research Laboratory

Team Members

Air Force Research Laboratory

- Kerry Hill - Sensors Directorate
- Todd Jenkins - Sensors Directorate
- Scott Douglass - Human Effectiveness Directorate
- Trevor Bihl - Sensors Directorate
- **Ashley DeMange - Sensors Directorate**

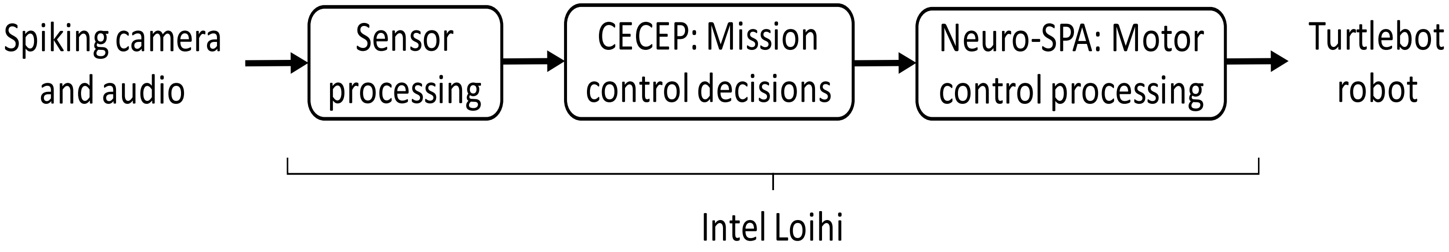
University of Dayton

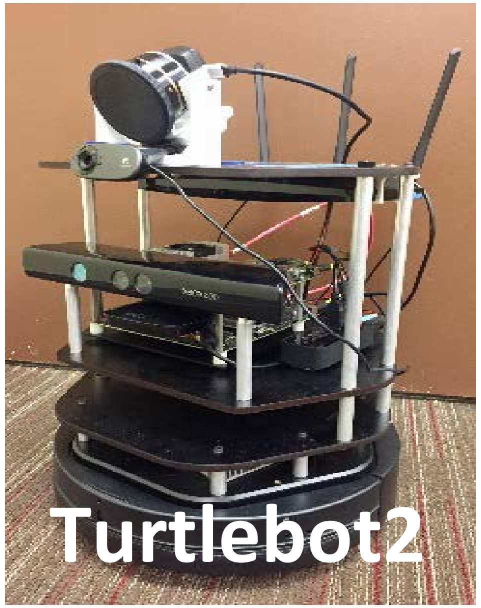
- Tarek Taha
- **Chris Yakopcic**
- David Fan (UDRI)
- Nayim Rahman
- Tanvir Atahary
- Alex Beigh (UDRI)

INRC Project Overview

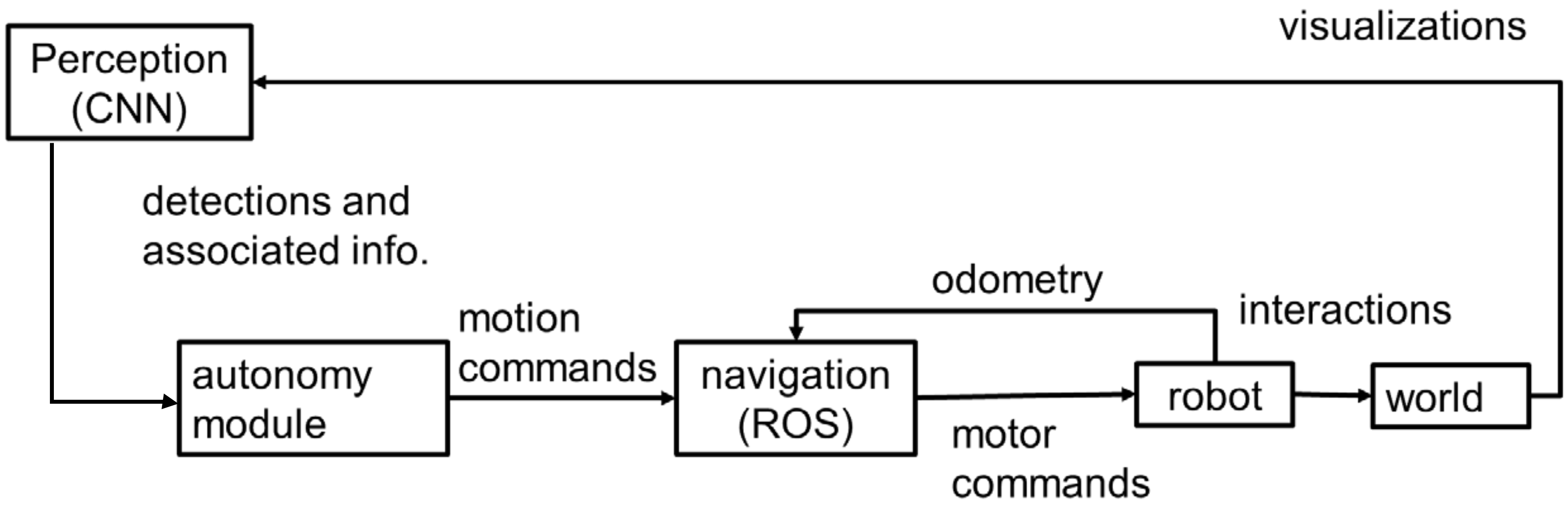
This project will develop cognitive agents for an autonomous robot. The system will take inputs from sensors (an Inilabs spiking camera and other sensors) and process these as inputs to a pair of cognitive agents:

- A high-level decision agent that takes a set of mission requirements, such as what order to complete a set of objectives based on the priorities for each objective.
- A motor control agent that accomplishes a specific set of physical tasks as directed by the high-level agent.

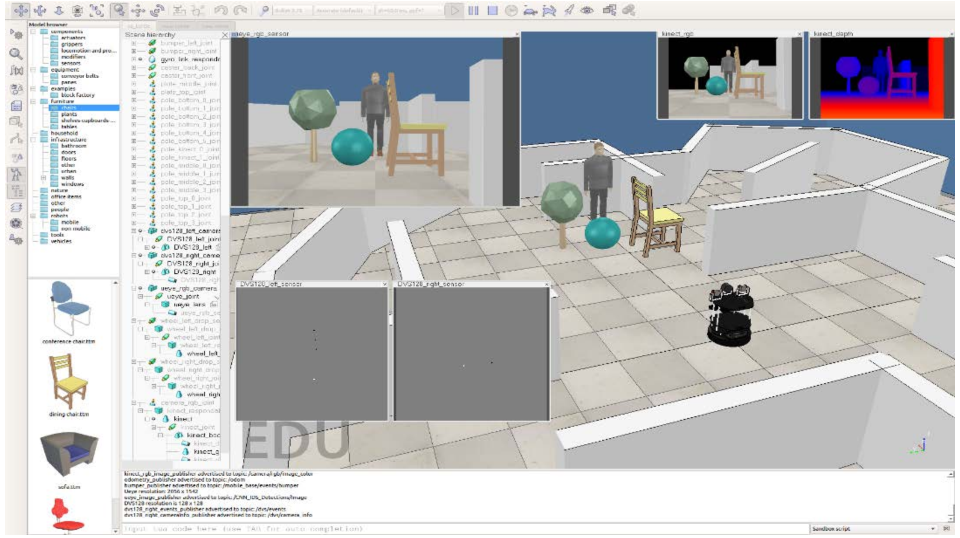




Progress: Reactive Robot (no CECEP or spiking camera)



- Begins with perception
  - CNN detections from video camera sent to Reasoner
- Reasoner then decides the platform's course of action
- Action sent to the navigation and control components
- Interaction with world (real or M&S) sensed, movement triggers changes in its perception and the loop repeats.



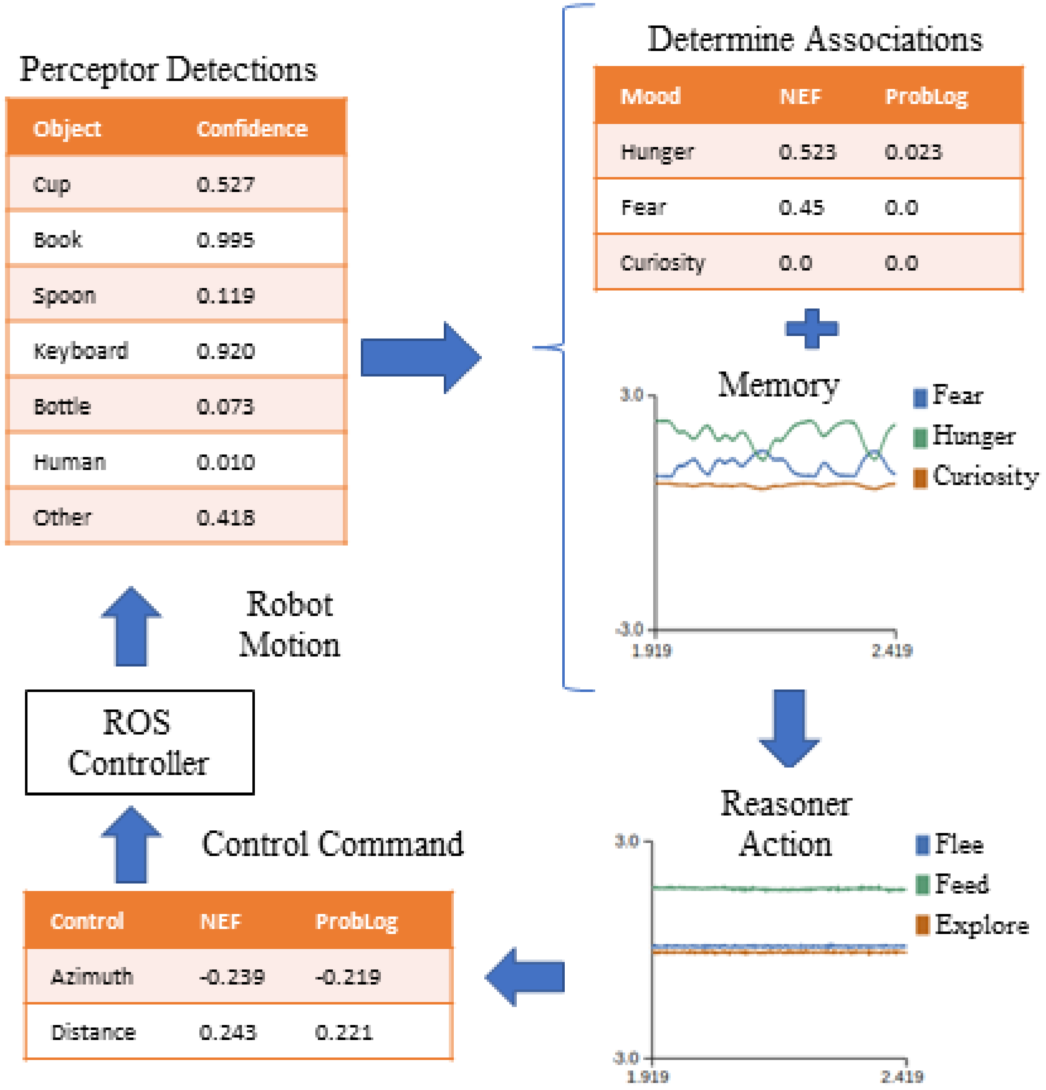
Example Results

High level intentions:

- “Feed”
- “Flee”
- “Explore”

General “Moods” for the reasoned:

- “Hunger”
- “Fear”
- “Curiosity”



Object	Confidence
Cup	0.527
Book	0.995
Spoon	0.119
Keyboard	0.920
Bottle	0.073
Human	0.010
Other	0.418

Mood	NEF	ProbLog
Hunger	0.523	0.023
Fear	0.45	0.0
Curiosity	0.0	0.0

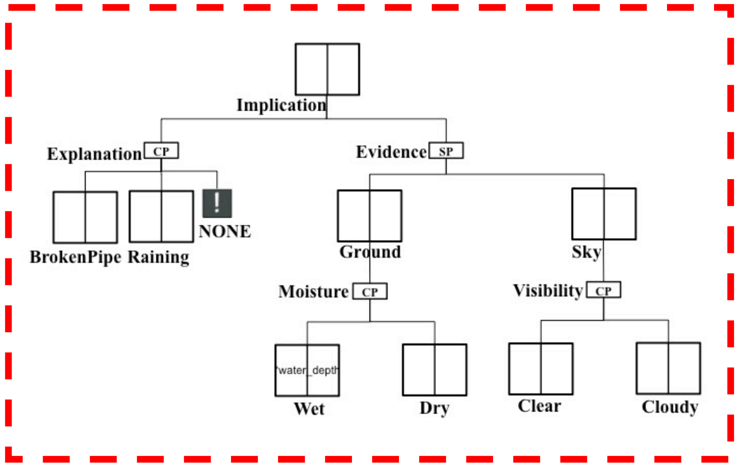
Control	NEF	ProbLog
Azimuth	-0.239	-0.219
Distance	0.243	0.221

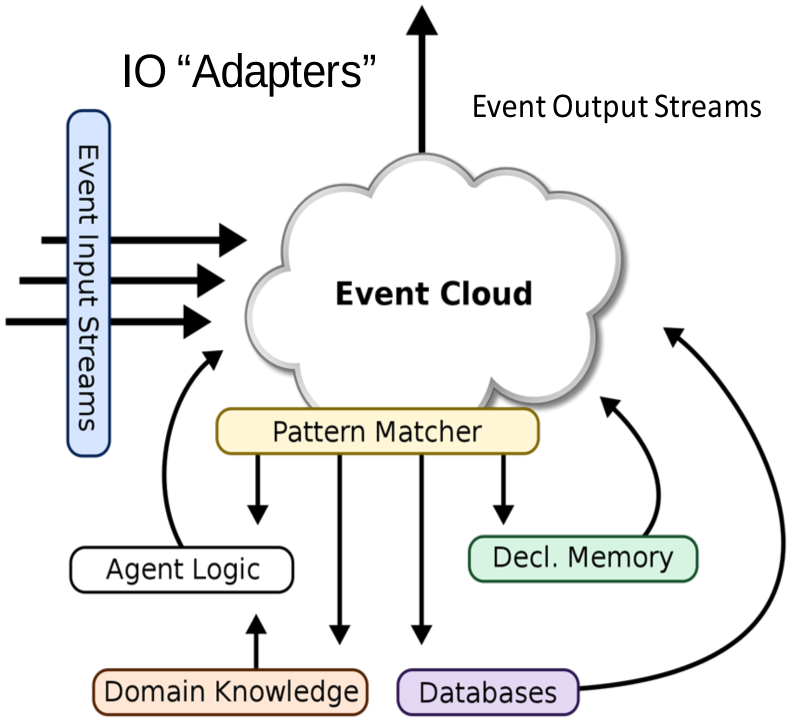
Example output from perceptor, with example object classes. Reasoner gives intents, which result in control outputs and moods (which are fed back to the Reasoner).

Bihl, T. J., Jenkins, T. R., Cox, C., DeMange, A., Hill, K., Zelnio, E. (2019) “From the lab to the internship and back again: Learning autonomous systems through creating a research and development ecosystem,” AAAI/EAAI, Honolulu, HI

High Speed Cognitive Domain Ontologies Using Loihi Spiking Neurons

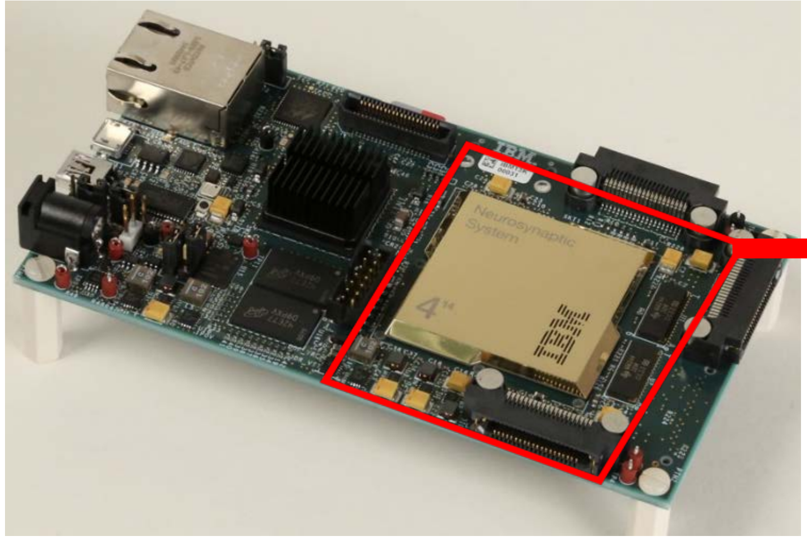
- CECEP Architecture
- The CDO is the decision making engine within the CECEP architecture
- These very simple examples quickly become very complex in realistic systems
  - Need to find best of billions of solutions

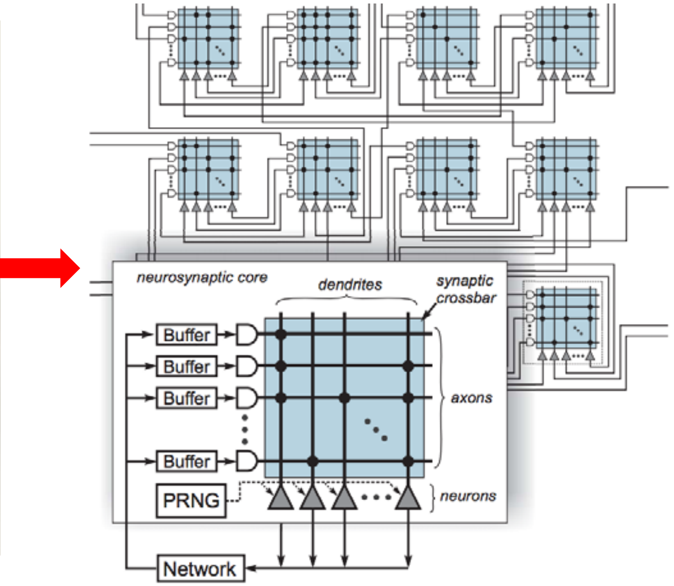




Objective

- Optimized resource allocation is extremely computationally expensive
- We need low SWaP alternatives, large problems are currently prohibitively expensive to solve.
- This is done using a series of spiking neurons that fire according to the most logical vehicle assignment options
- This work covers a MATLAB implementation of the spiking neuron based algorithm

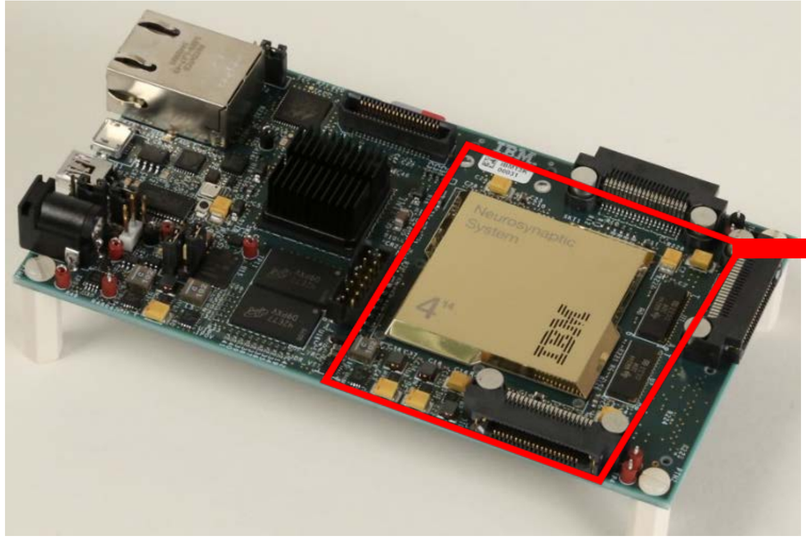


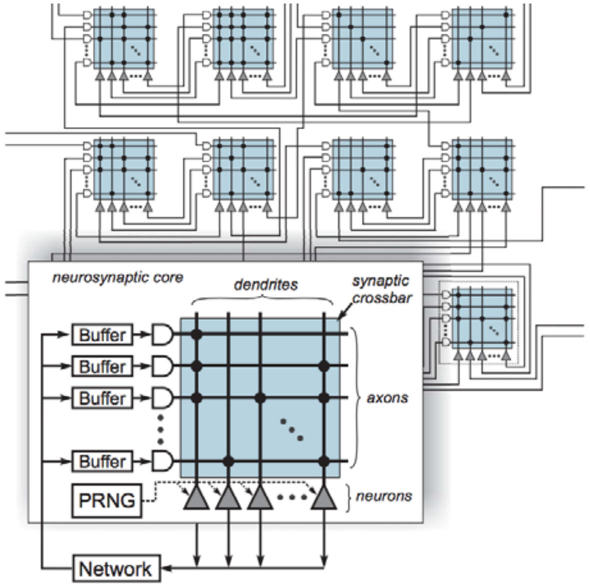


Allocation Problem Size	Number of Possible Solutions
2 × 2	9
4 × 4	625
6 × 6	117,649
8 × 8	43,046,721
10 × 10	25,937,424,601

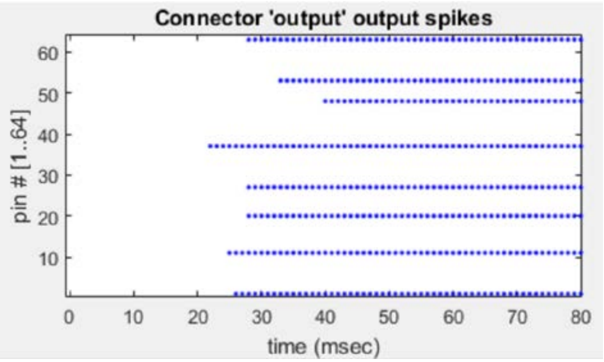
Objective

- Optimized resource allocation is extremely computationally expensive
- We need low SWaP alternatives, large problems are currently prohibitively expensive to solve.
- This is done using a series of spiking neurons that fire according to the most logical vehicle assignment options
- This work covers a MATLAB implementation of the spiking neuron based algorithm





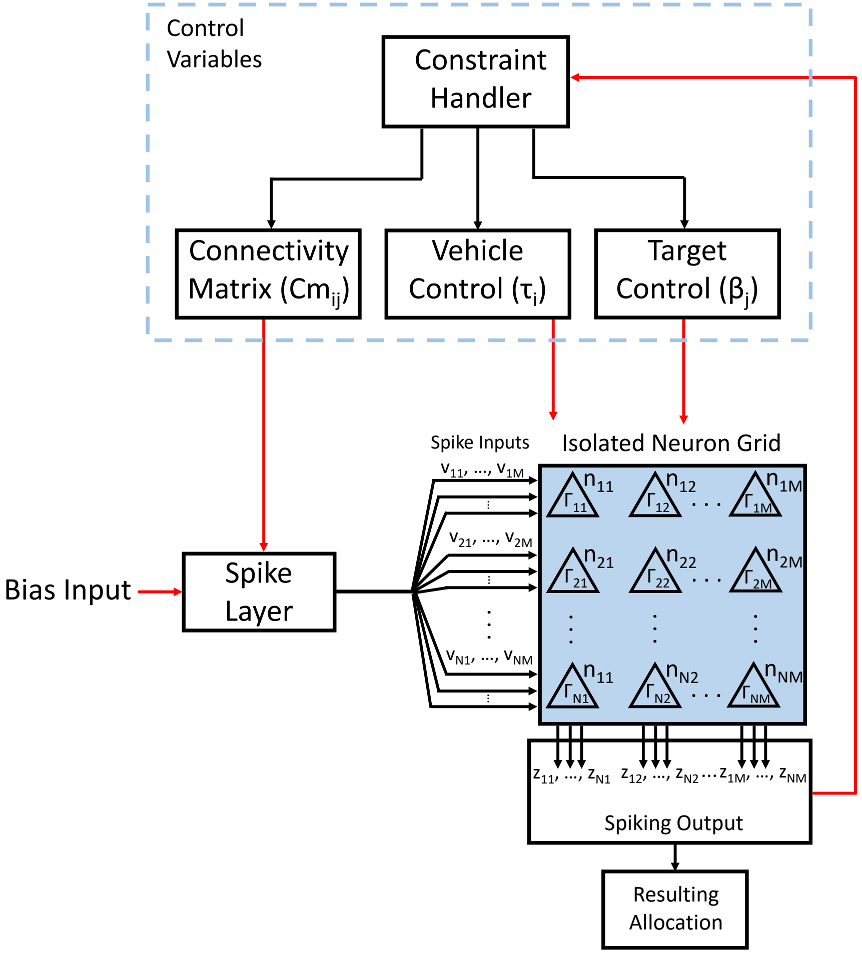
True North Spiking Output



Vehicle 1 Goes to Target 1
Vehicle 2 Goes to Target 3
Vehicle 3 Goes to Target 4
Vehicle 4 Goes to Target 3
Vehicle 5 Goes to Target 5
Vehicle 6 Goes to Target 8
Vehicle 7 Goes to Target 5
Vehicle 8 Goes to Target 7

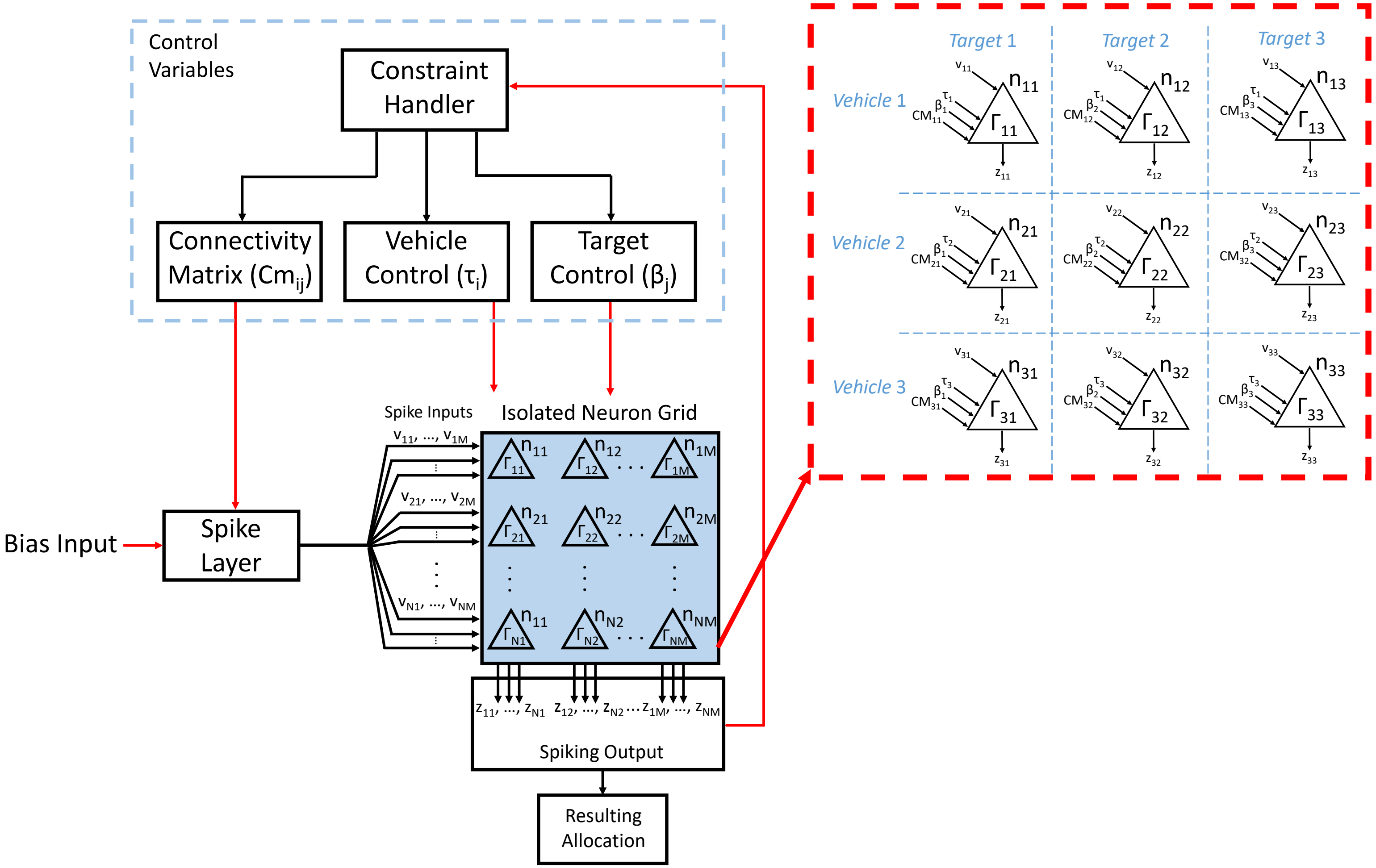
Simplified Algorithm Block Diagram

- Single neuron holds connection between one vehicle and one target
  - Capable of allocating N vehicles for M separate targets using N×M neurons
  - Spike accumulation is proportional to a weighted sum of priority, success, and time
- Weight Parameters
  - TTA: Time to the target
  - Priority: Necessity of reaching target
  - TOT: Hold time for vehicle once target is reached
  - Probability of Success: Likelihood that a target will be completed by a certain vehicle
  - TTC = TOT + TTA
- Control Parameters
  - CM: Connectivity matrix hold vehicle-target compatibility
  - τ: A vehicle can only be assigned to one target
  - β: Penalize (but do not stop) multiple vehicles from reaching a single target



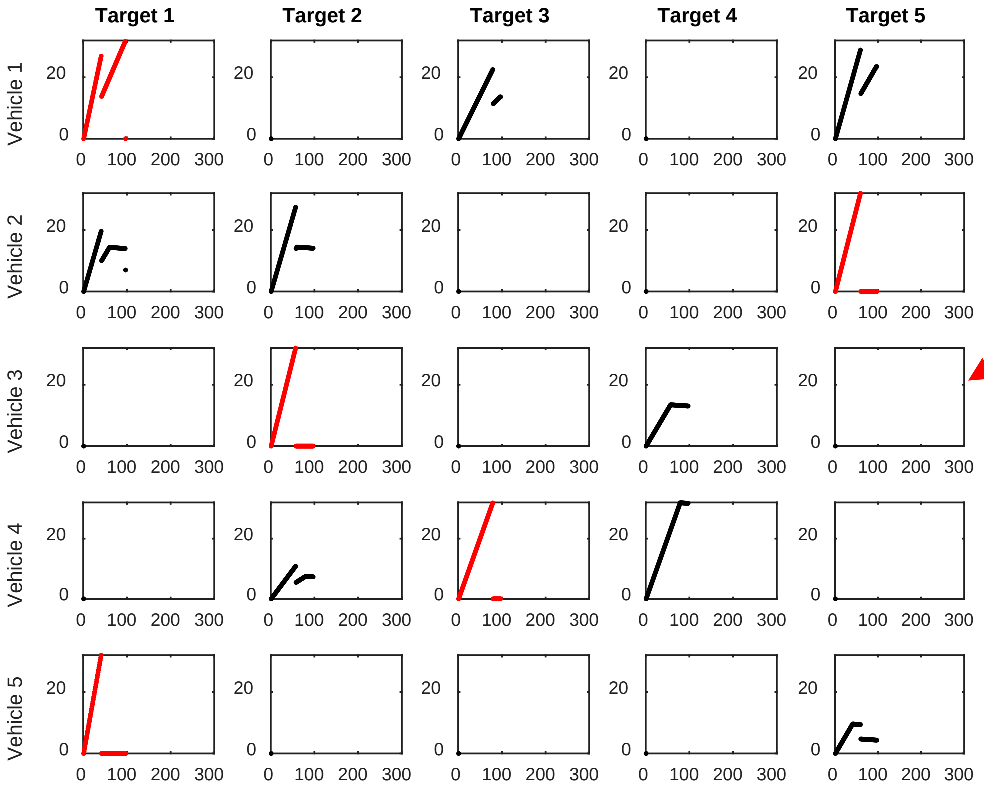
Simplified Algorithm Block Diagram

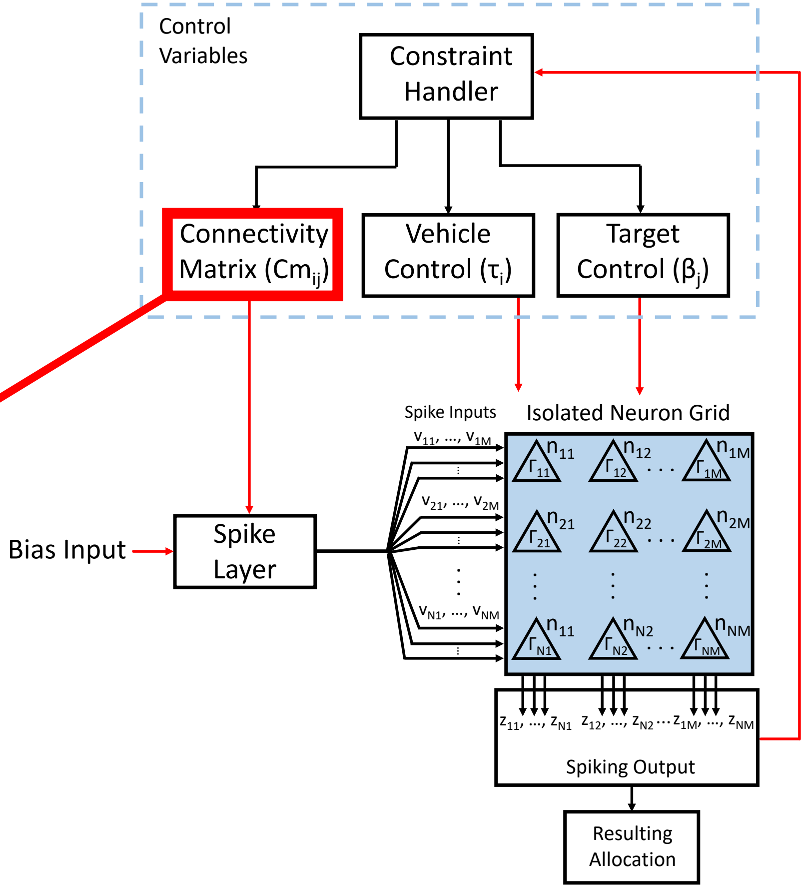
- Single neuron holds connection between one vehicle and one target
  - Capable of allocating N vehicles for M separate targets using N×M neurons
  - Spike accumulation is proportional to a weighted sum of priority, success, and time
- Weight Parameters
  - TTA: Time to the target
  - Priority: Necessity of reaching target
  - TOT: Hold time for vehicle once target is reached
  - Probability of Success: Likelihood that a target will be completed by a certain vehicle
  - TTC = TOT + TTA
- Control Parameters
  - CM: Connectivity matrix hold vehicle-target compatibility
  - τ: A vehicle can only be assigned to one target
  - β: Penalize (but do not stop) multiple vehicles from reaching a single target



Example Allocation

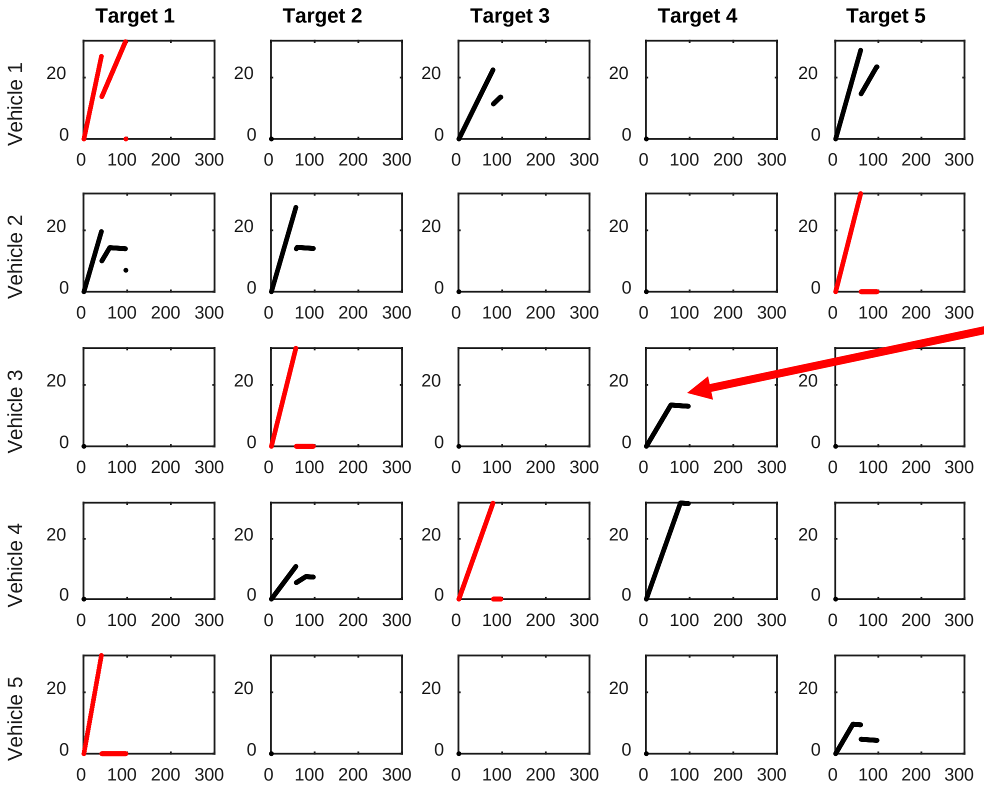
Connectivity Matrix Response

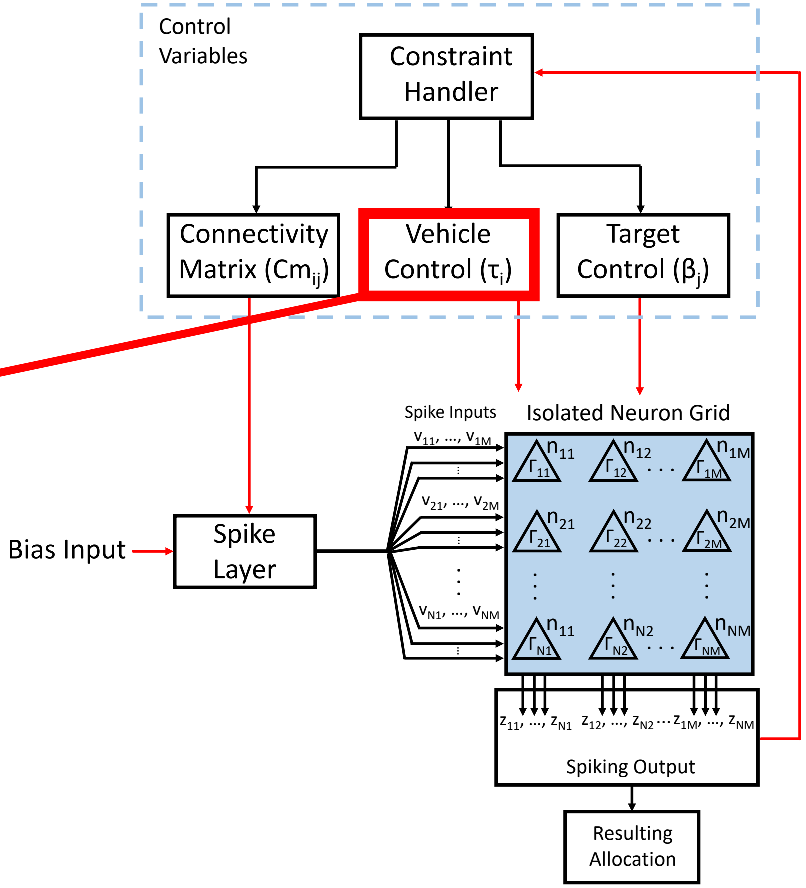




Example Allocation

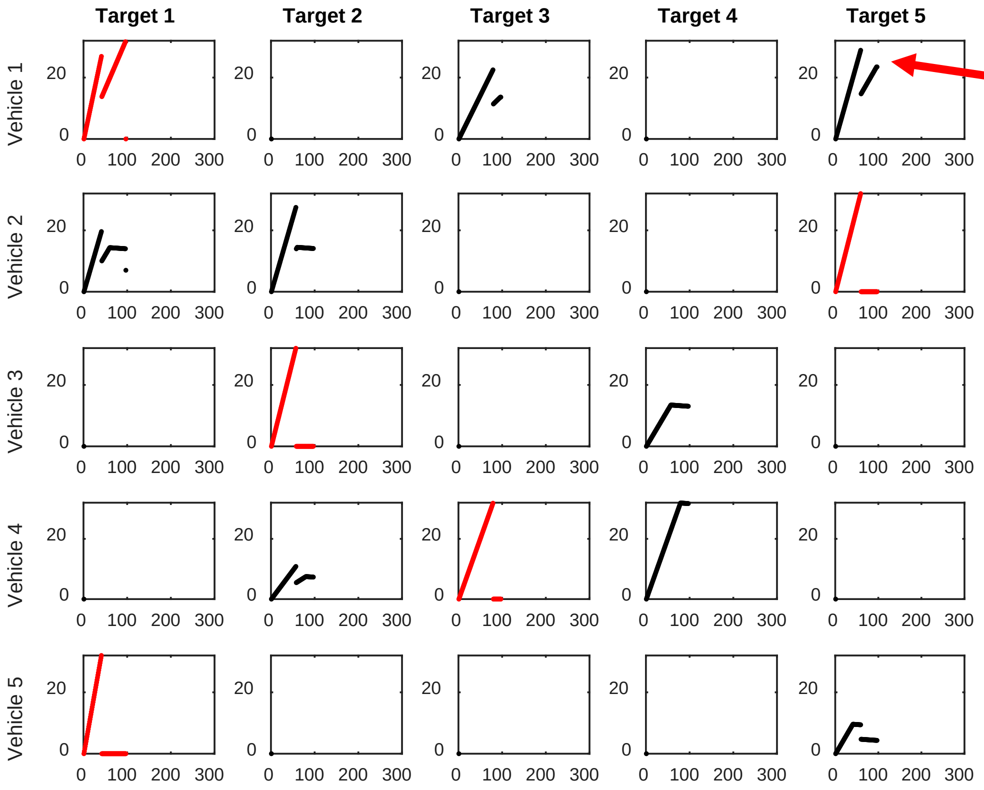
Vehicle Control Response

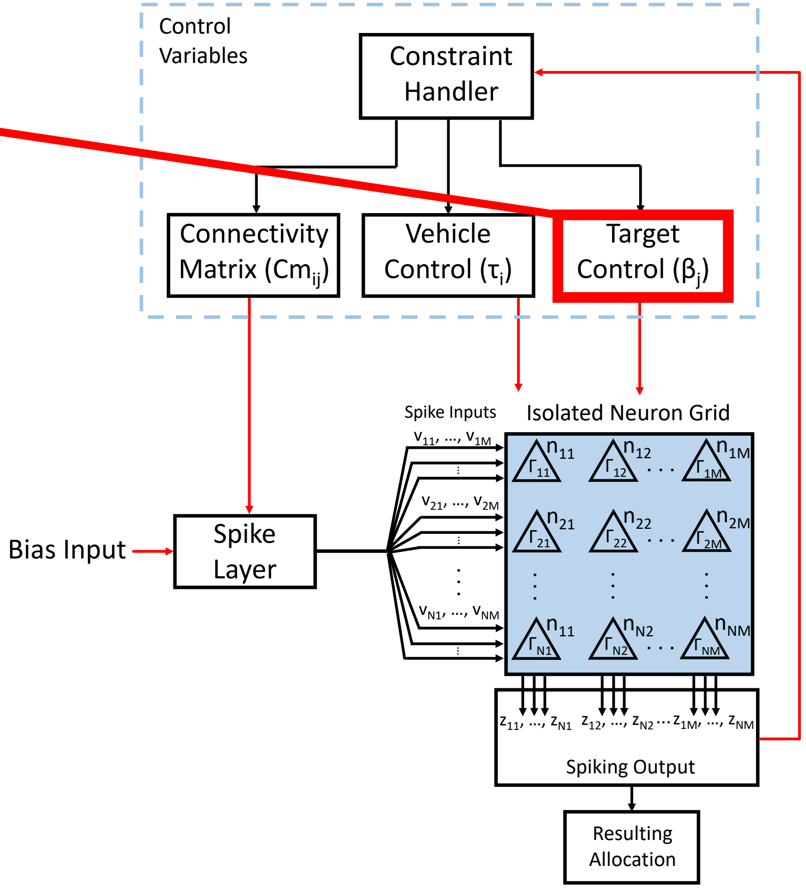




Example Allocation

Target Control Response





Algorithm Comparison

	Exhaustive Search Using CPU/GPU		Spiking System			
Allocation Size	Baseline CDO Reward	Baseline CDO Result	Effective Reward	Allocation Result	Answer Rank	Answer Percentile
3×3	18.8703	[2 1 1]	18.4807	[2 1 2]	2 of 64	98.44%
4×4	11.3777	[4 1 1 3]	10.7303	[4 2 1 3]	9 of 625	98.72%
5×5	22.6219	[1 5 2 4 1]	22.6203	[1 5 2 3 1]	2 of 7,776	99.99%
6×6	31.2628	[2 4 1 5 3 3]	29.1396	[5 4 1 6 3 3]	78 of 117,649	99.93%
7×7	48.448	[4 2 2 6 5 6 7]	43.0929	[4 2 4 6 2 6 7]	1,431 of 2.09M	99.93%
8×8	40.8782	[1 3 4 7 5 3 6 8]	40.6779	[1 3 4 8 5 3 6 7]	4 out of 43.0M	~100%

- The best allocation for each case was determined using a GPU exhaustive search
- The table compares this result to the approximate result obtained from the Loihi spiking system

Timing Comparison

- Runtime comparison between the exhaustive search and spiking system

Allocation Size	CDO Search Time (CPU)	CDO Search Time (GPU)	Spiking System Time	Spiking System Speedup
2×2	2.6 ms	-	30.6 ms	0.08×
3×3	8.9 ms	-	31.2 ms	0.29×
4×4	84.7 ms	-	35.9 ms	2.36×
5×5	13,041 ms	-	42.2 ms	309×
6×6	-	223.6 ms	53.1 ms	4.21×
7×7	-	454.8 ms	53.9 ms	8.43×
8×8	-	480,205 ms	62.4 ms	7696×

- Solution space on an asset allocation problem grows with problem size
  - Neuron utilization grows at a much smaller rate

Allocation Problem Size	Number of Possible Solutions	Number of Neurons
2 × 2	9	4
4 × 4	625	16
6 × 6	117,649	36
8 × 8	43,046,721	64
10 × 10	25,937,424,601	100

Next Steps

- Asset Allocation
  - More complete algorithm comparison
  - Study of maximum scalability
  - **Results Found in:**
    - C. Yakopcic, T. Atahary, T. M. Taha, A. Beigh, and S. Douglass, “High Speed Approximate Cognitive Domain Ontologies for Asset Allocation based on Isolated Spiking Neurons,” IEEE National Aerospace and Electronics Conference (NAECON), pp. 241-246, Dayton, OH, July, 2018.
    - C. Yakopcic, T. Atahary, N. Rahman, T. M. Taha, A. Beigh, and S. Douglass, “High Speed Approximate Cognitive Domain Ontologies for Asset Allocation Using Loihi Spiking Neurons,” IEEE/INNS International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, July, 2019 (Accepted).

- Loihi
  - Constraint Solving
  - Energy / Timing Benchmark