Virtual Research Presentation Conference 2020

Scalable and Distributed Swarm Motion Planning via Integrated Optimization and Machine Learning

Principal Investigator: Changrak Choi (347)
Co-Is: Shahrouz Ryan Alimo (393), Kyongsik Yun (349), Amir Rahmani (347), Anthony Davis (329)
External Co-I: Ramtin Madani (University of Texas at Arlington)
Program: RTD Topic

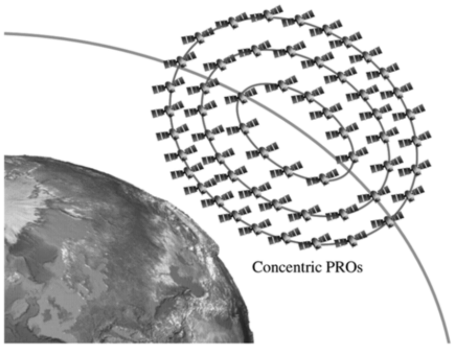
Jet Propulsion Laboratory
California Institute of Technology

Assigned Presentation # RPC-275

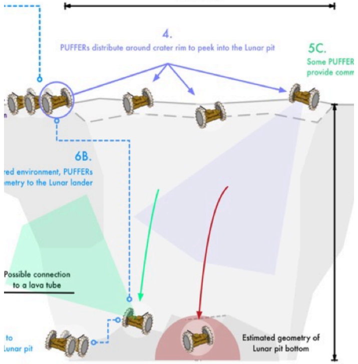
Abstract

In this task, we aim to develop a distributed and real-time motion planning algorithmic framework and software for multi-agent dynamical systems that can scale up to more than hundreds of agents. Motion planning and trajectory optimization are central to future NASA missions involving swarms of autonomous space vehicles. In response, we seek to fundamentally advance the existing practices for swarm motion planning, which lack scalability and are not compatible with highly distributed computational platforms. To this end, we take a hybrid machine learning and optimization theoretical approach with two objectives in mind: i) Tackling the inherent computational complexity of motion planning, the so-called NP-hardness, to reduce trajectory optimization computation time for large swarm by orders-of-magnitude, and ii) Enabling distributed decision-making by swarms of agents with very limited computational capability. Specifically, we use advances and techniques in artificial intelligence, convex optimization, and distributed control to build a swarm motion planner that can be tailored to JPL’s future space exploration problems.

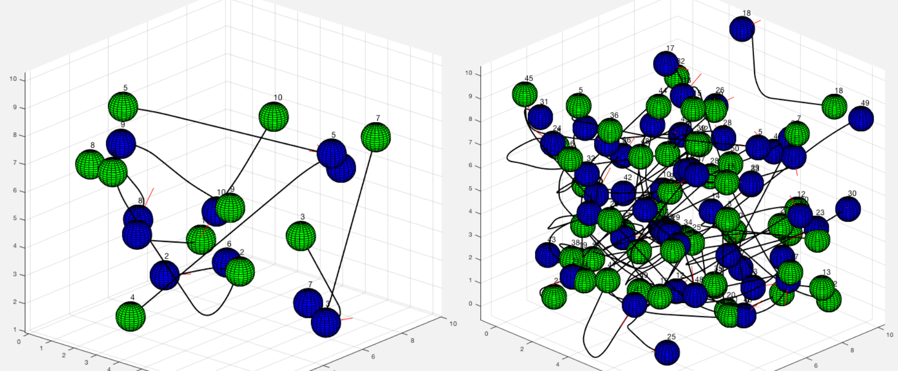
Globally Optimal Swarm Coordination



Spacecraft Swarm [Morgan et. al]



Multiple PUFFERs in Exploration



Example of Fuel Optimal Trajectories for Reconfiguration of CubeSat Swarm in Extremely Clustered Environment

Problem Statement:

How to plan motion trajectories of large fleet of space vehicles that is globally optimal w.r.t. mission objectives with requirement that it must be computed
1) ***on-board***, 2) ***real-time***, and 3) ***highly scalable***

Innovation over SoA:

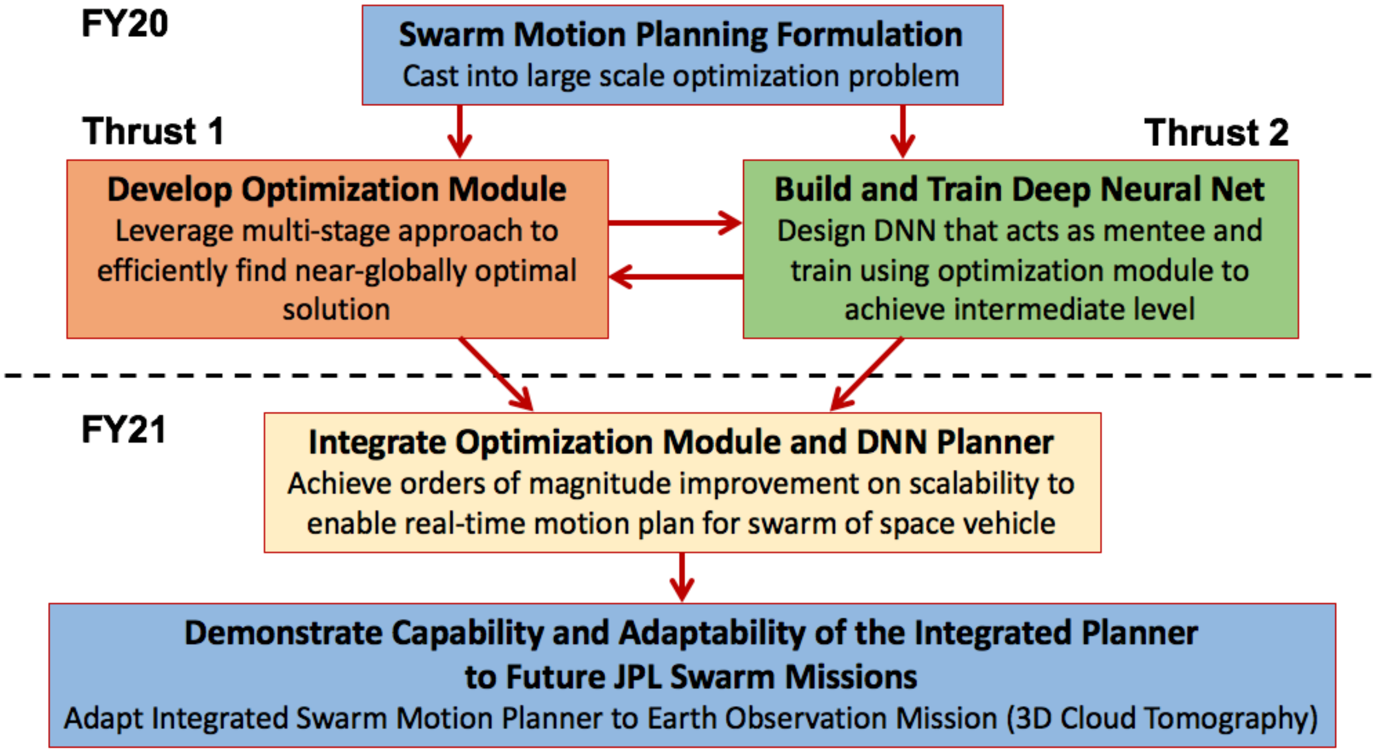
SoA	Our Approach	Benefit
Poor Scalability	Highly Scalable	Applicability to large swarm missions
Offline	Online/Real-time	Robustness to uncertainty
Centralized	Distributed	Enable on-board computation

Importance to JPL:

Enable swarm missions that cannot rely on the ground-in-the-loop guidance to continuously adjust trajectories on its own, to guarantee mission success and be robust to uncertainties that cannot be pre-determined

Technical Approach

Synergistic integration of recent advances in *Optimization* and *Machine Learning*
To achieve required scalability and computational efficiency for real-time on-board swarm motion planning



Technical Approach

Synergistic integration of recent advances in *Optimization* and *Machine Learning*
To achieve required scalability and computational efficiency for real-time on-board swarm motion planning

FY20

Swarm Motion Planning Formulation
Cast into large scale optimization problem

Thrust 1

Develop Optimization Module
Leverage multi-stage approach to efficiently find near-globally optimal solution

Thrust 2

Build and Train Deep Neural Net
Design DNN that acts as mentee and train using optimization module to achieve intermediate level

---

Key Technical Questions

1. Can we advance the SoA in multi-agent planning via optimization? (RA-L)
2. Can we replicate optimal solutions via training neural net? (ASCEND 2020)
3. Can we make synergetic integration of optimization and NN-based planner to tackle the curse of dimensionality of multi-agent planning? (SciTech 2021)

Technical Approach – Optimization

Thrust (1) Massively Scalable and Distributed Optimization for Motion Planning

Aims orders-of-magnitude scalability improvements with global optimality guarantee by tackling inherent non-convexity via novel *Parabolic Relaxation* combined with Feasibility Enforcement and Numerical Search

High-Level Approach

1. Formulate multi-agent motion planning into large-scale QCQP problem (original problem)
2. Lift original problem into higher dimensional space to linearize objective ftn & collision avoidance constr
3. Convexify lifted formulation via computationally efficient parabolic relaxation
4. Enforce feasibility of solution via penalty and further computational efficiency through numerical search

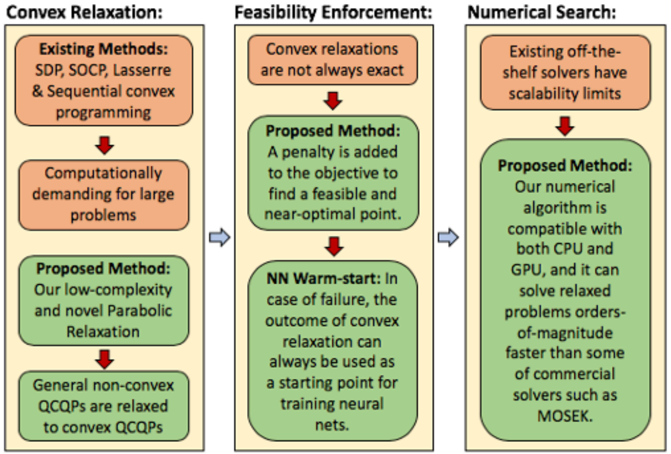


Figure 1. Proposed Optimization-theoretical Approach

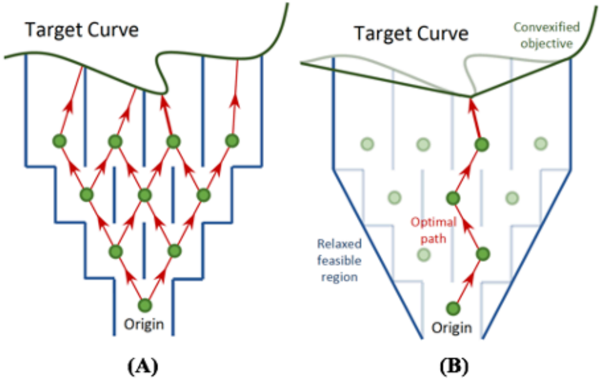
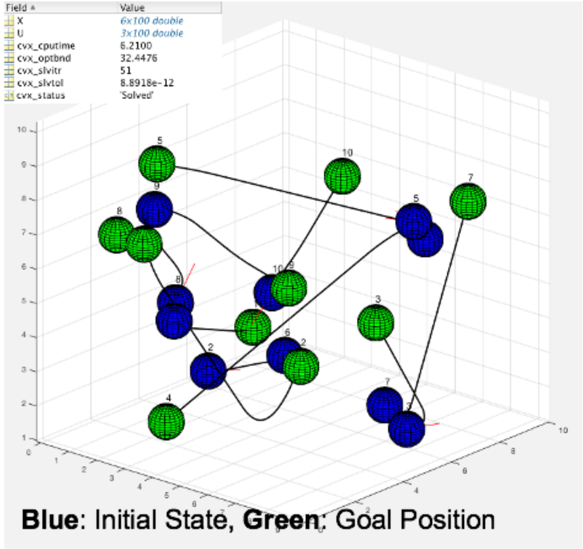


Figure 2. (A) Inherent complexity of motion planning due to non-convexity (B) Convex relaxation and feasibility enforcement to tackle the complexity

Technical Approach – Optimization

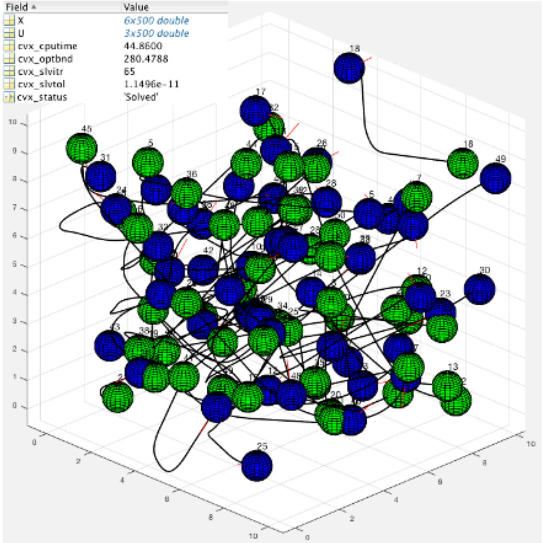
Simulation Results (CubeSat with 3-axis Thrusters in Deep Space)

- 1U CubeSats (with 3-axis thrusters) are placed randomly inside 10x10x10 arena
- Each CubeSat must simultaneously travel to its assigned goal position (randomly created
- Optimal trajectories that minimizes total delta-V expenditure while ensuring collision avoidance among agents and dynamic feasibility must be computed



Blue: Initial State, **Green**: Goal Position

10 CubeSats: f*=32.45, computation time=6.21s



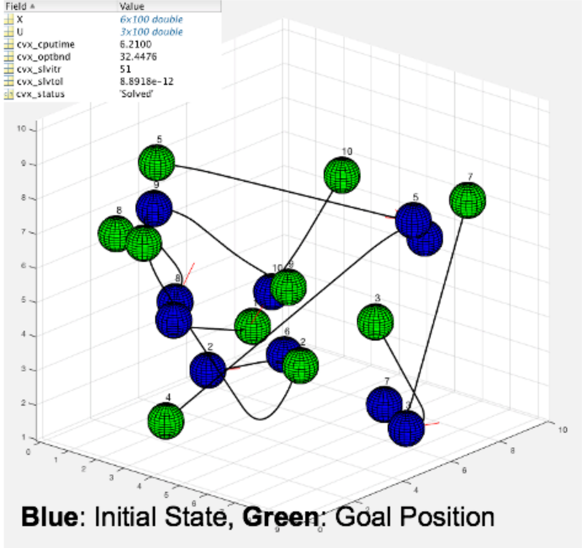
50 CubeSats: f*=280.48, computation time=44.86s

SoA methods cannot solve this problem instance

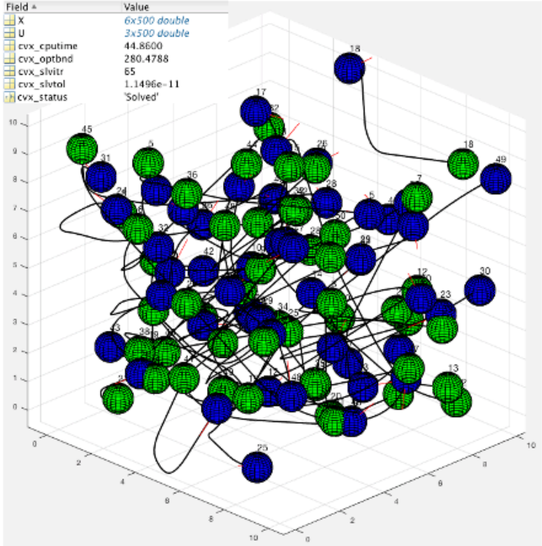
Technical Approach – Optimization

Our optimization approach (overcomes limitations of SoA)

- Does not rely on local approximation of the problem
- Convergence does not requires having good initial seed and is fast
- Guarantees global optimality and finite step convergence to feasibility
- Can handle arbitrary objective, constraints that can be expressed quadratically



10 CubeSats: f*=32.45, computation time=6.21s



50 CubeSats: f*=280.48, computation time=44.86s

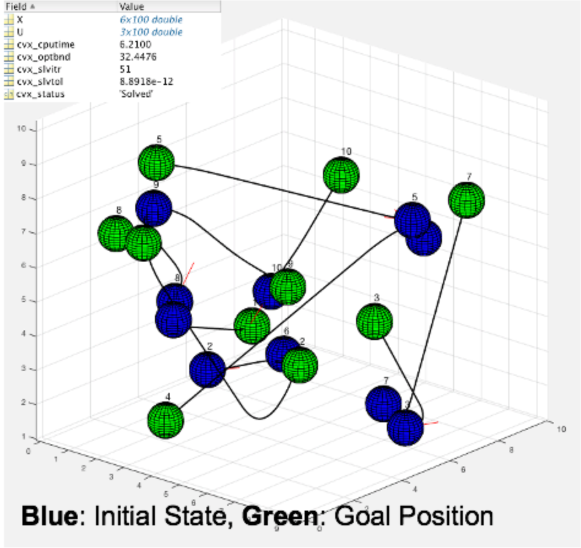
SoA methods cannot solve this problem instance

Technical Approach – Optimization

We are working to further extend our approach to reduce computation time by

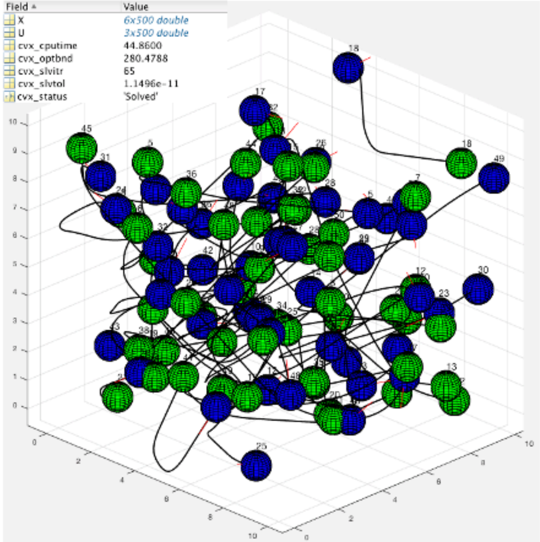
- Enabling distributed computation among agents (of centralized policy) through ADMM
- Utilizing potential GPUs in base station (e.g. mothership) through massive parallelization

However, optimization eventually suffers from the curse of dimensionality that is inherent in the multi-agent trajectory planning problem -> **we seek to leverage advances in ML**



Blue: Initial State, **Green**: Goal Position

10 CubeSats: f*=32.45, computation time=6.21s



50 CubeSats: f*=280.48, computation time=44.86s

We want to further reduce this computation time by 1/10

Technical Approach

Synergistic integration of recent advances in *Optimization* and *Machine Learning*
To achieve required scalability and computational efficiency for real-time on-board swarm motion planning

FY20

Swarm Motion Planning Formulation
Cast into large scale optimization problem

Thrust 1

Develop Optimization Module
Leverage multi-stage approach to efficiently find near-globally optimal solution

Thrust 2

Build and Train Deep Neural Net
Design DNN that acts as mentee and train using optimization module to achieve intermediate level

---

Key Technical Questions

1. Can we advance the SoA in multi-agent planning via optimization? (RA-L)
2. Can we replicate optimal solutions via training neural net? (ASCEND 2020)
3. Can we make synergetic integration of optimization and NN-based planner to tackle the curse of dimensionality of multi-agent planning? (SciTech 2021)

Technical Approach – Machine Learning

Thrust (2) Intelligent Initialization and Optimization-free Planning via Deep Learning

a) provides high-quality initial seed for optimization solver to significantly reduce the convergence time

b) optimization-free planner that can provide instant solutions via a Deep Neural Network (DNN).

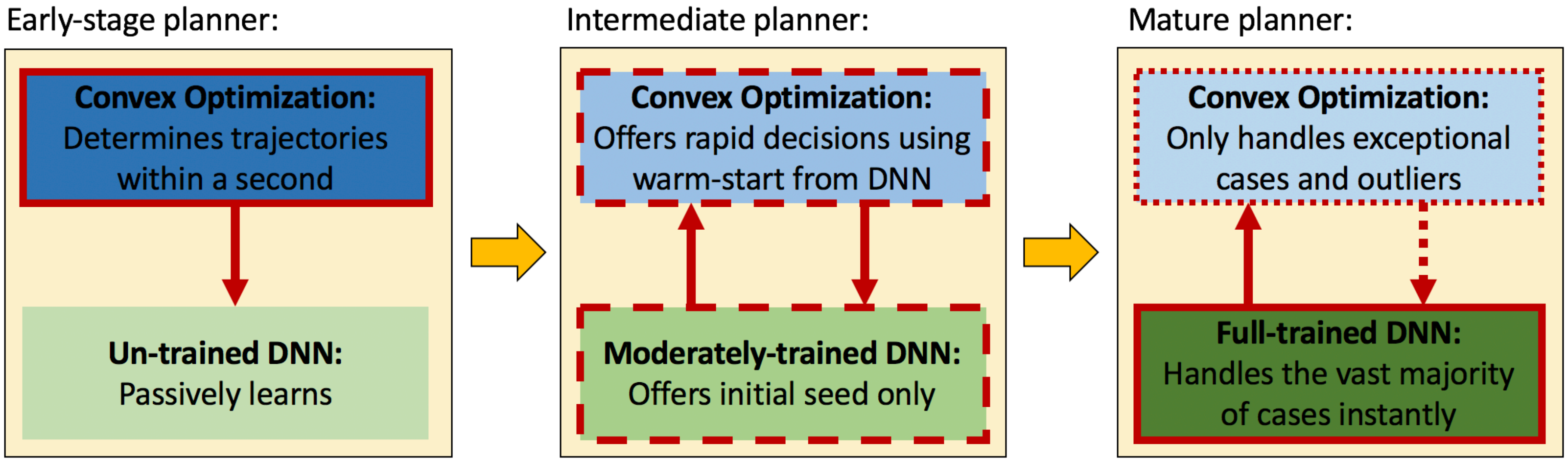
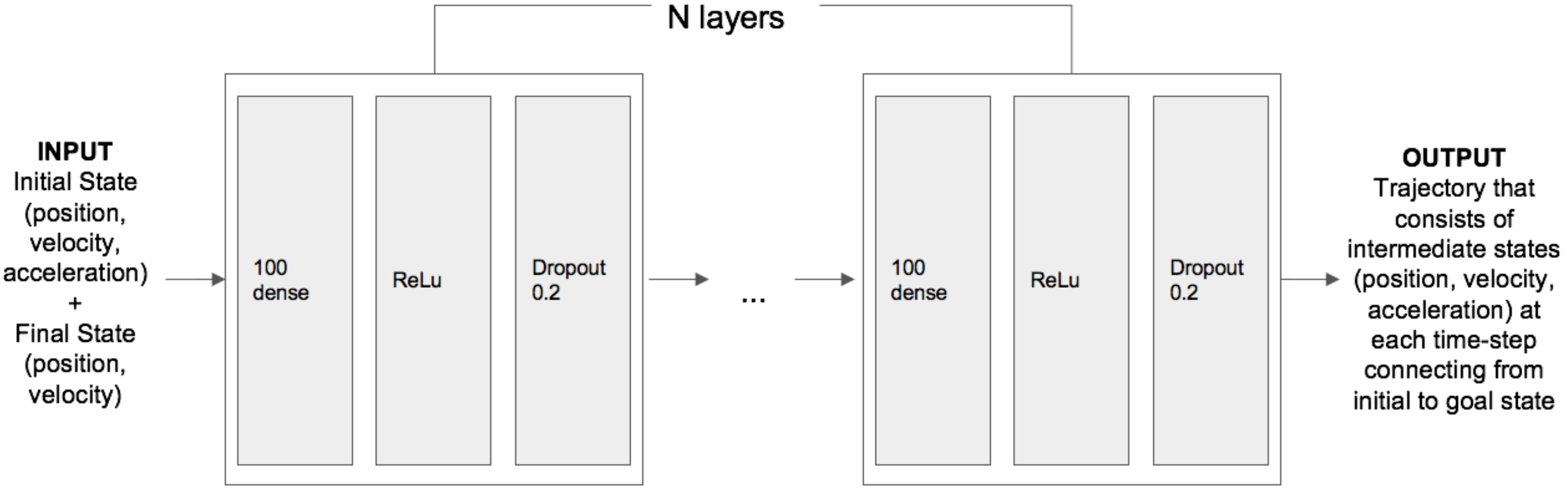


Figure 3. The Mentor/Mentee relationship between the optimization and AI technologies to be developed. Matured DNN will provide instantaneous computation-free plan for the vast majority of cases while Reliable but slower and more expensive optimization module handles exceptional cases for robustness

Technical Approach – Machine Learning

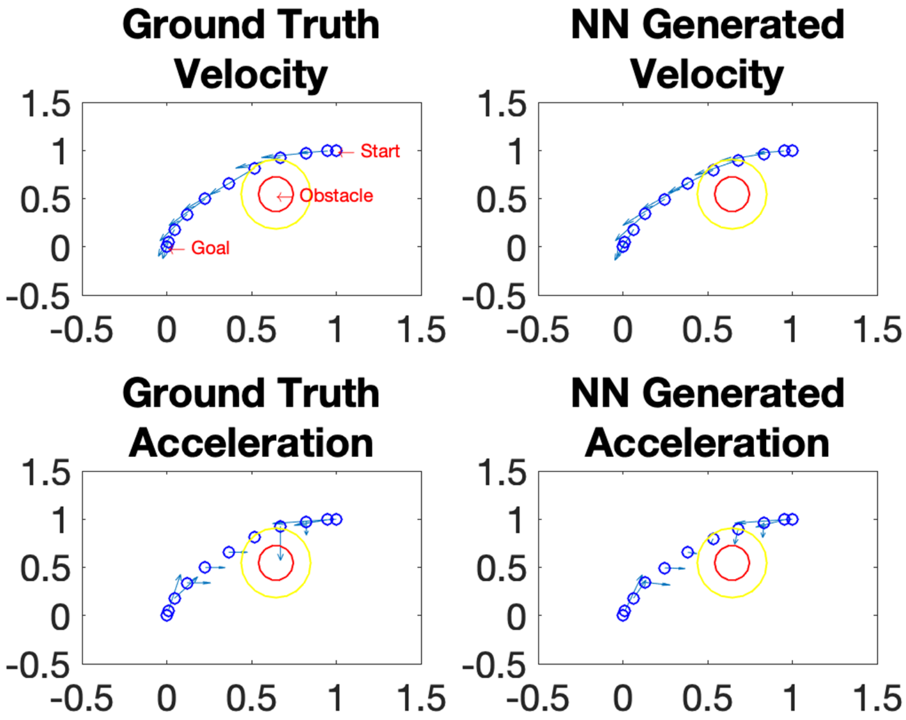
Current Neural Network Architecture (Feed-Forward NN)



Hyperparameter tuning is used to determine the number layer and nodes that are optimal for a physical system under consideration

Technical Approach – Machine Learning

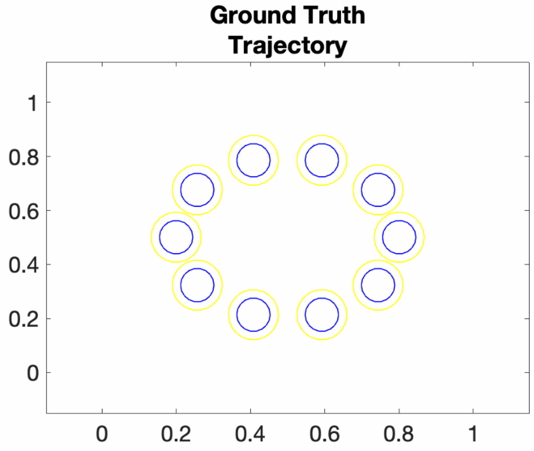
Physical System – 2D Double Integrator, Single Agent + Single Obstacle (5k dataset)

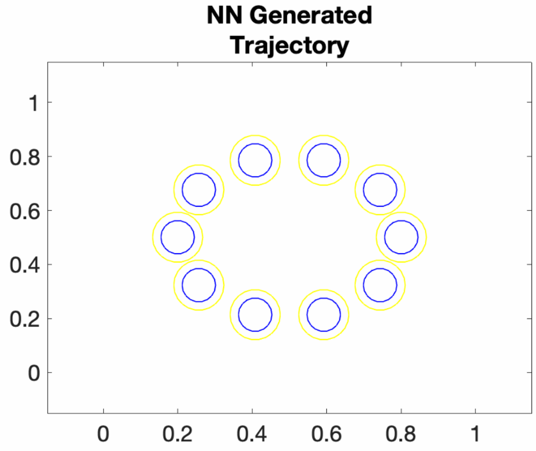


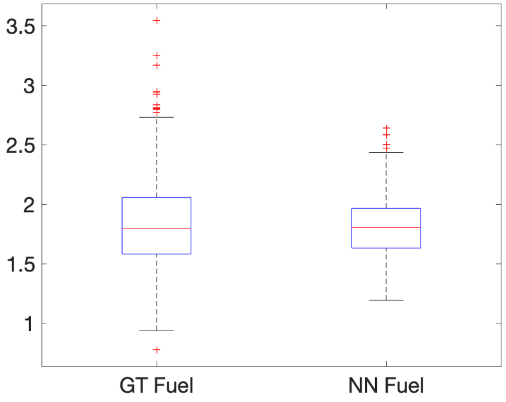
Our preliminary results show accurate position and velocity estimation (RMSE = 0.0129 ± 0.0088) using a deep learning based numerical model, given the simplest single agent and single obstacle.

Technical Approach – Machine Learning

Physical System – 2D Double Integrator, 10 Agent (30k dataset)



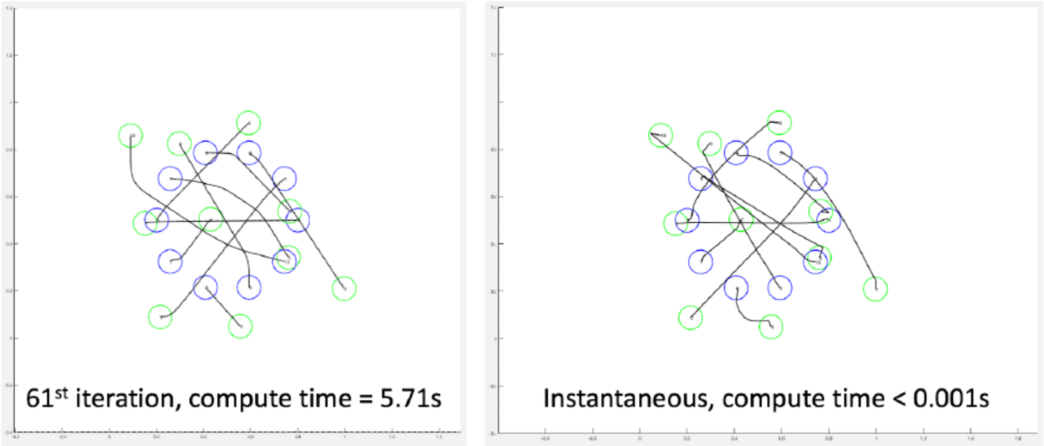




- NN was trained on the 10 agents neural net model using 30k dataset.
- Fuel consumption of NN model was comparable to that of the ground truth (optimal sol)
- NN generated trajectories still showed some collisions but quantum jumps are nearly resolved (compared to 1k dataset)
- Trained NN takes only 0.5ms to generate solution (compared to ~5s via optimization)

Technical Approach – Machine Learning

Physical System – 2D Double Integrator, 10 Agent (30k dataset)



Global Optimal Solution, $f^*_{opt} = 2.2545$ ML Solution, $f^*_{ML} = 2.2916$

- NN was trained on the 10 agents neural net model using 30k dataset.
- Fuel consumption of NN model was comparable to that of the ground truth (optimal sol)
- NN generated trajectories still showed some collisions but quantum jumps are nearly resolved (compared to 1k dataset)
- Trained NN takes only 0.5ms to generate solution (compared to ~5s via optimization)

Summary

SOA Comparison

Multi-Agent Planning SoA	mRRT*	ORCA	MILP	SCP	Parabolic
Methodology	Sampling -based	Velocity Obstacle based	Optimization	Optimization	Optimization
Scalability	▲	✓	X	▲	▲
Optimality	▲	X	✓	▲	✓
Dynamical Feasibility	X	X	▲	✓	✓
Computational Efficiency	▲	✓	X	▲	▲

Key Technical Questions

1. Can we advance the SoA in multi-agent planning via optimization? (RA-L)
2. Can we replicate optimal solutions via training neural net? (ASCEND 2020)
3. Can we make synergetic integration of optimization and NN-based planner to tackle the curse of dimensionality of multi-agent planning? (SciTech 2021)

Summary

SOA Comparison

Multi-Agent Planning SoA	mRRT*	ORCA	MILP	SCP	Parabolic	DNN
Methodology	Sampling -based	Velocity Obstacle based	Optimization	Optimization	Optimization	Machine Learning
Scalability	▲	✓	X	▲	▲	✓
Optimality	▲	X	✓	▲	✓	▲
Dynamical Feasibility	X	X	▲	✓	✓	▲
Computational Efficiency	▲	✓	X	▲	▲	✓

Key Technical Questions

1. Can we advance the SoA in multi-agent planning via optimization? (RA-L)
2. Can we replicate optimal solutions via training neural net? (ASCEND 2020)
3. Can we make synergetic integration of optimization and NN-based planner to tackle the curse of dimensionality of multi-agent planning? (SciTech 2021)

Summary

SOA Comparison

Multi-Agent Planning SoA	mRRT*	ORCA	MILP	SCP	Parabolic	DNN	Integrated Approach
Methodology	Sampling -based	Velocity Obstacle based	Optimization	Optimization	Optimization	Machine Learning	Opt + ML
Scalability	▲	✓	X	▲	▲	✓	✓
Optimality	▲	X	✓	▲	✓	▲	✓
Dynamical Feasibility	X	X	▲	✓	✓	▲	✓
Computational Efficiency	▲	✓	X	▲	▲	✓	✓

Key Technical Questions

1. Can we advance the SoA in multi-agent planning via optimization? (RA-L)
2. Can we replicate optimal solutions via training neural net? (ASCEND 2020)
3. Can we make synergetic integration of optimization and NN-based planner to tackle the curse of dimensionality of multi-agent planning? (SciTech 2021)

Publications and References

1. K. Yun, C. Choi, S. Alimo, A. Davis, L. Forster, A. Rahmani, M. Adil, and R. Madani, “Swarm Motion Planning using Deep Learning for Space Applications,” ASCEND 2020 (abstract accepted, full paper submitted)

2. S. Alimo, K. Yun, C. Choi, A. Rahmani, M. Adil, F. Kamangar, and R. Madani, “Scalable Swarm Trajectory Planning via Integrated Optimization and Machine Learning,” AIAA SciTech 2021 (extended abstract accepted)

3. C. Choi, A. Rahmani, and R. Madani, “Scalable Swarm Trajectory Planning via Parabolic Relaxation,” IEEE RA-L (to be submitted)

4. N. Pugh, C. Choi, and H. Park, “Multi-agent Optimal Planetary Exploration via Multi-stage Path and Motion Planner,” IEEE Aerospace Conference 2021 (abstract accepted)

New Technology Reports

5. NTR #51586, “Scalable and Distributed Swarm Motion and Trajectory Planning for Large Scale Multi Agent Systems”

6. NTR #51654, “Scalable and Distributed Swarm Motion Planning using Machine Learning”

7. NTR #51655, “Scalable and Distributed Swarm Motion Planning via Integrated Optimization and Machine Learning”

Thanks to Our Team Members