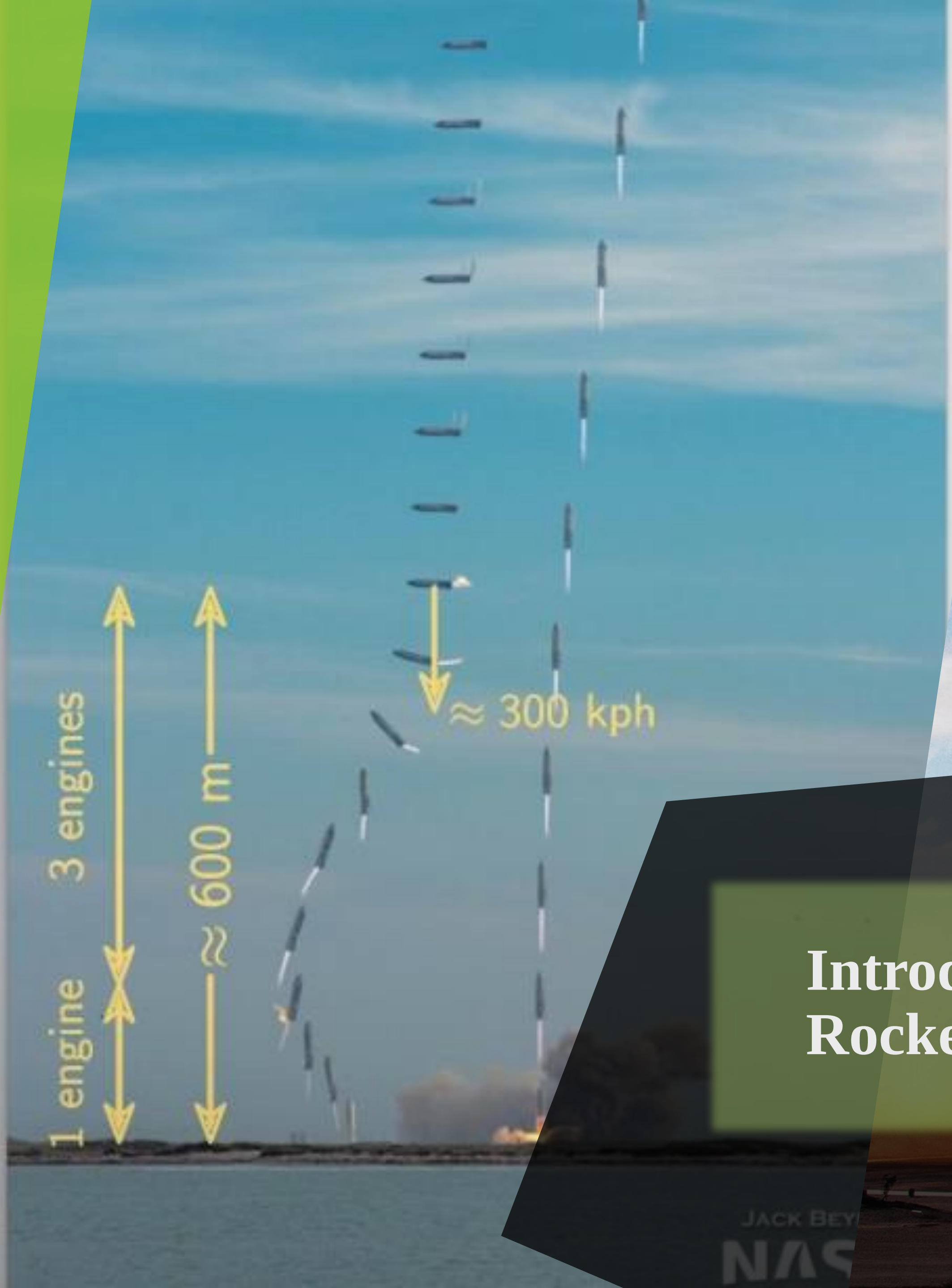
LEARNING TO GENERATE INITIAL GUESSES FOR TRAJECTORY OPTIMIZATION PROBLEM OF ROCKET LANDING

Prayag Sharma, Youssef Nsouli, Yuxuan Zhu

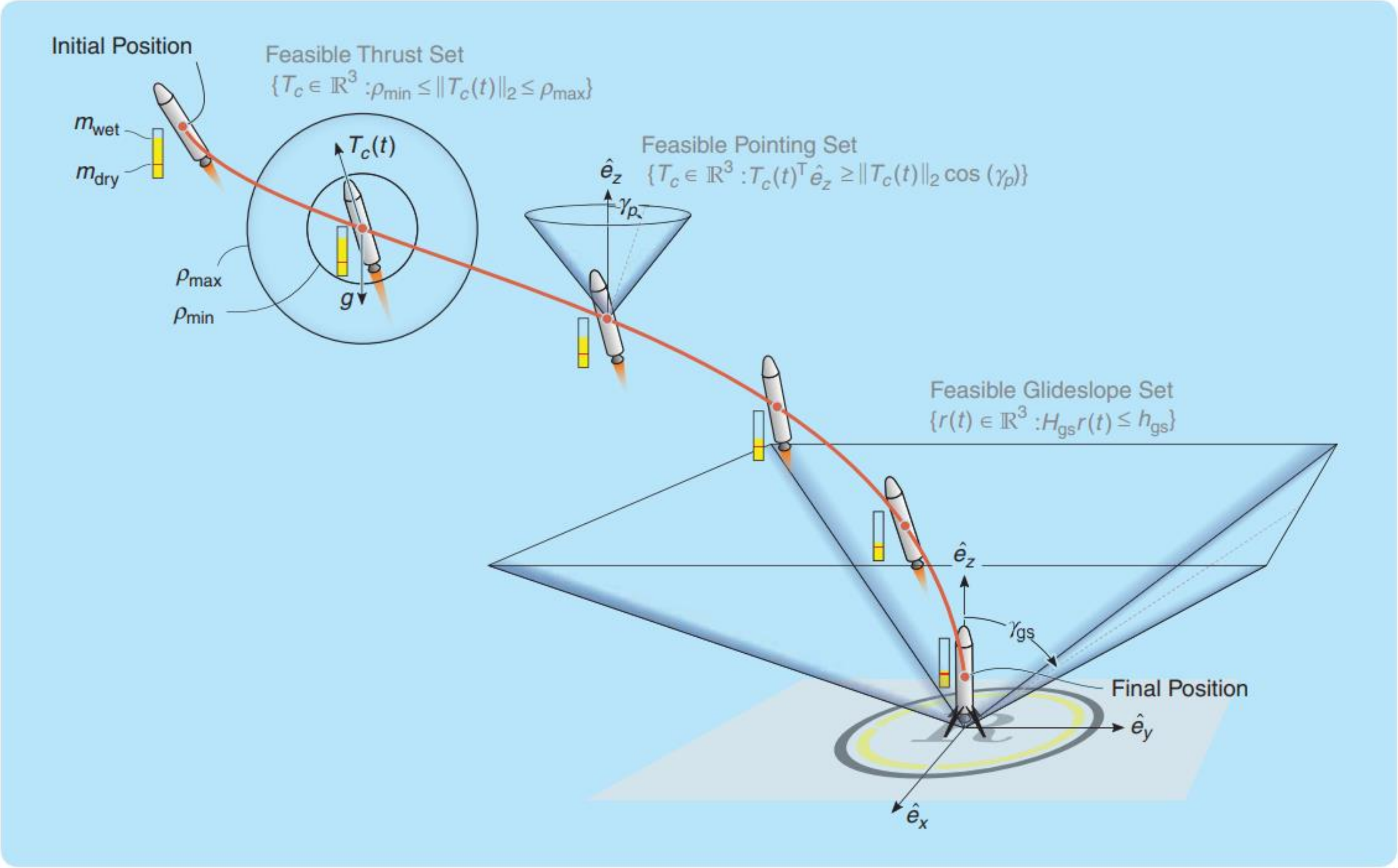


Introduction:
Rocket Landing Problem


T+05:34


T+ 00:07:22
STAGE 2 TELEMETRY
SPEED
ALTITUDE
19747 km/h
173 km
FALCON HEAVY TEST FLIGHT
STARTUP
MAX-Q
MAIN ENGINE CUTOFF
BOOSTERS LAND
CORE LANDS
LIFTOFF
BOOSTER ENGINE CUTOFF
FAIRING DEPLOY
STAGE 2 SHUTDOWN

Powered Descent 3-DOF Trajectory Optimization



Powered Descent 3-DOF Trajectory Optimization

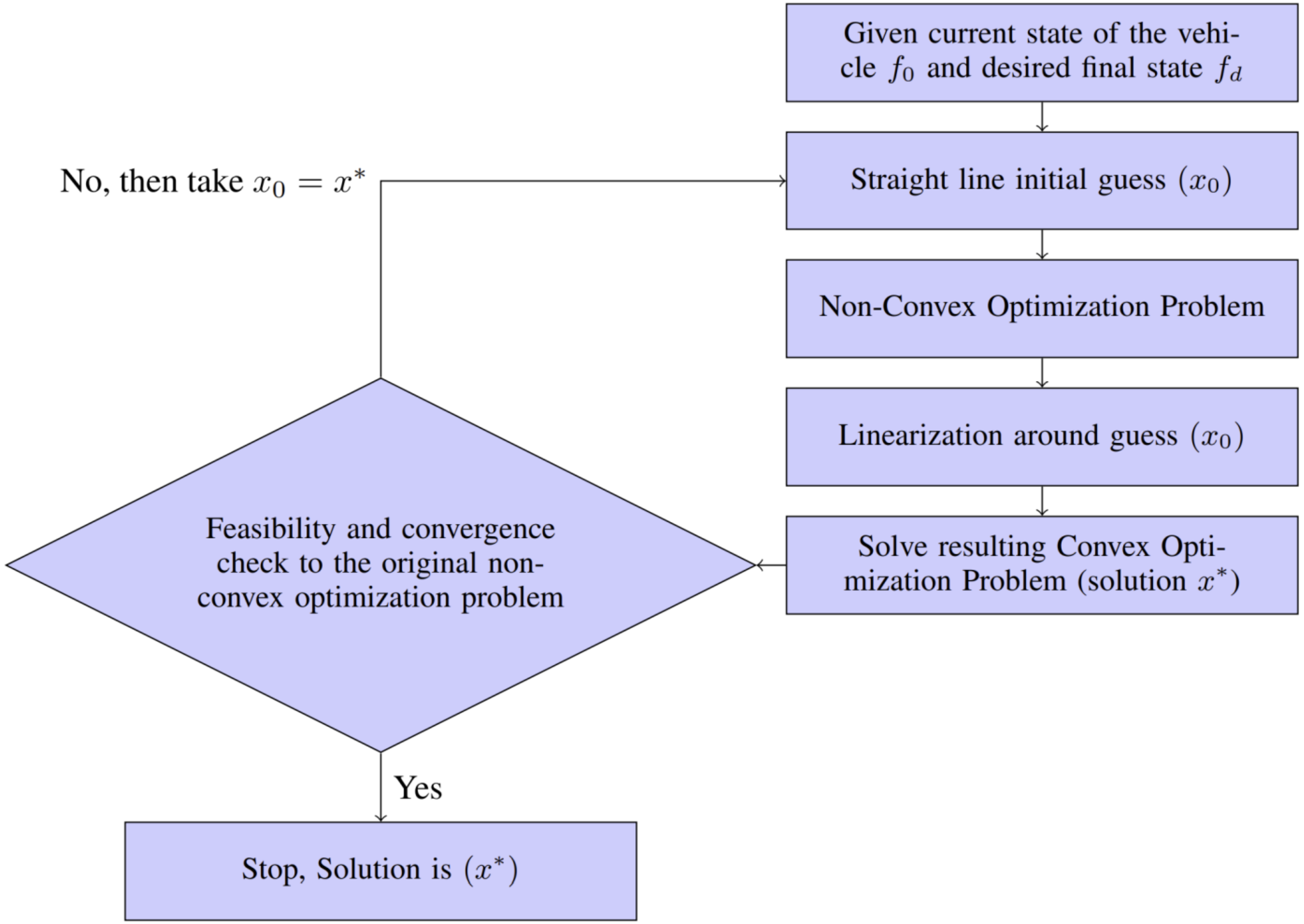
- Major Challenges:
  - Feasible and if possible optimal solution in less than 1 sec
  - In real-time
  - Onboard based on the initial condition
  - Limited computation capacity
  - Space grade processors

Optimal Control/Optimization Problem

$$\left[\begin{array}{ccc} & \max\limits_{t_f, T_c(\cdot)} m(t_f) = \min\limits_{t_f, T_c(\cdot)} \int_0^{t_f} \|T_c(t)\| \, \mathrm{d}t & \\ \textbf{State Dynamics:} & & \textbf{Boundary Conditions:} \\ \dot{r}(t) = V(t) & \textbf{Path Constraints:} & m(0) = m_{\mathrm{wet}},\ r(0) = r_0,\ \dot{r}(0) = \dot{r}_0 \\ \ddot{r}(t) = g + \frac{T_c(t)}{m(t)} & 0 < \rho_1 \leq \|T_c(t)\| \leq \rho_2 & r(t_f) = 0,\ \dot{r}(t_f) = 0 \\ \dot{m}(t) = -\alpha \|T_c(t)\| & \|Sx(t) - v\| + c^T x + a \leq 0 & T_c(0) = \|T_c(0)\| \hat{n}_0,\ T_c(t_f) = \|T_c(t_f)\| \hat{n}_f \\ r, V, T, g \in \mathbb{R}^3,\ m \in \mathbb{R} & & \end{array}\right]$$

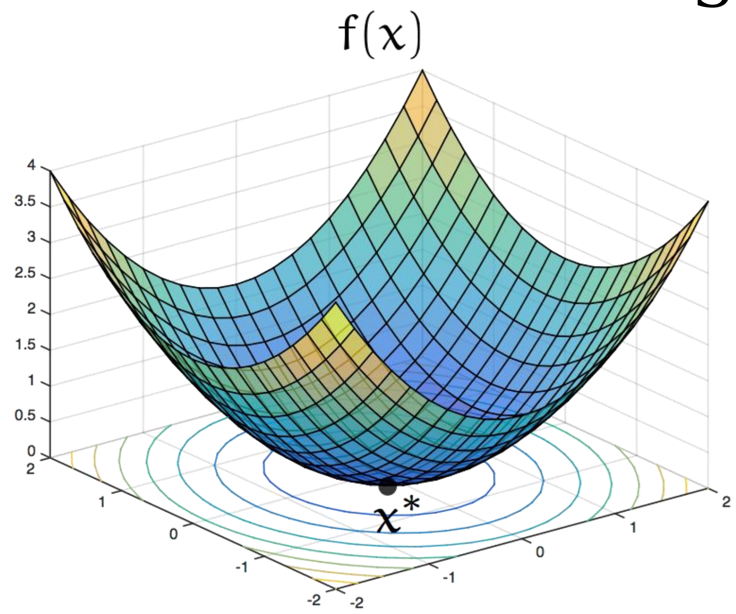
- There has been a lot of work on proving and formulating these optimal control problems towards guarantee feasible solutions when solved using certain optimization techniques like Trust region methods

Solved Using Successive Convexification of the original non-convex optimization problem



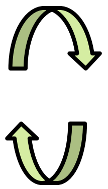
How to Deal with Non-Convexity:

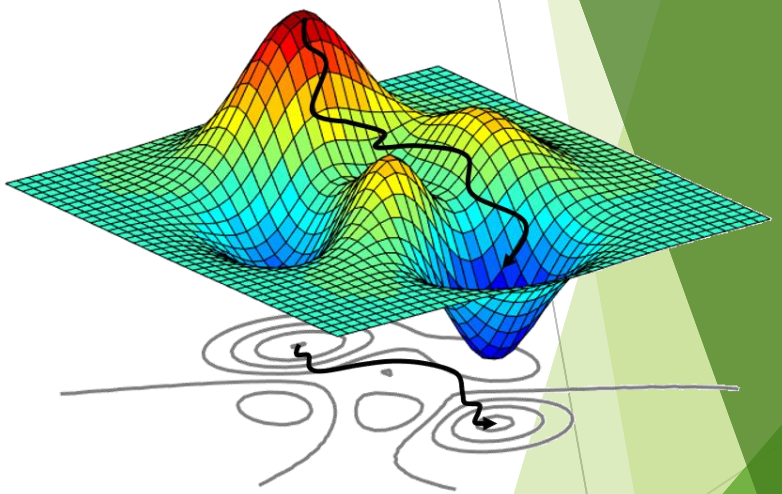
- Methods like penalized trust region and discretization techniques can guarantee local minima

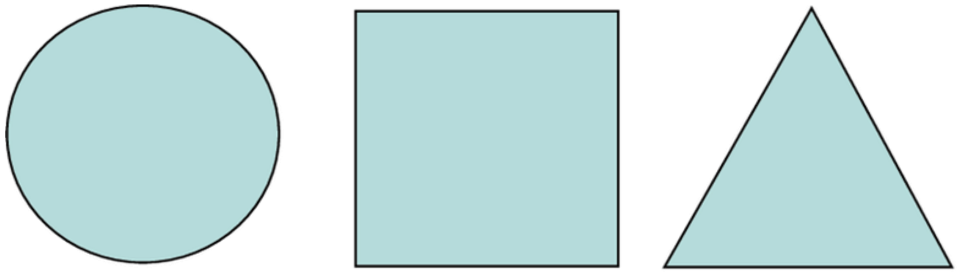


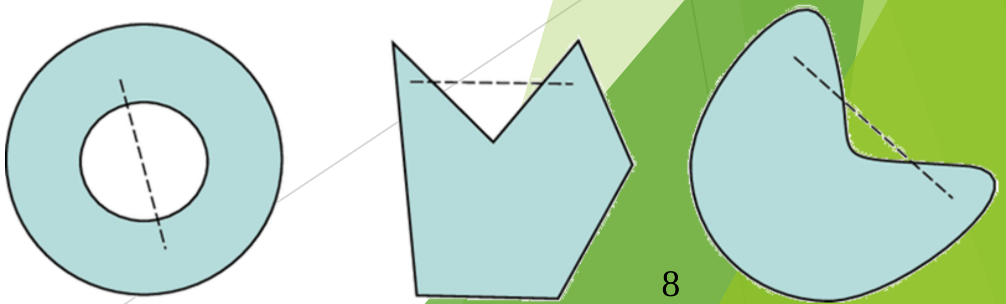
Sequential Convexification

Linearization
at each Iteration

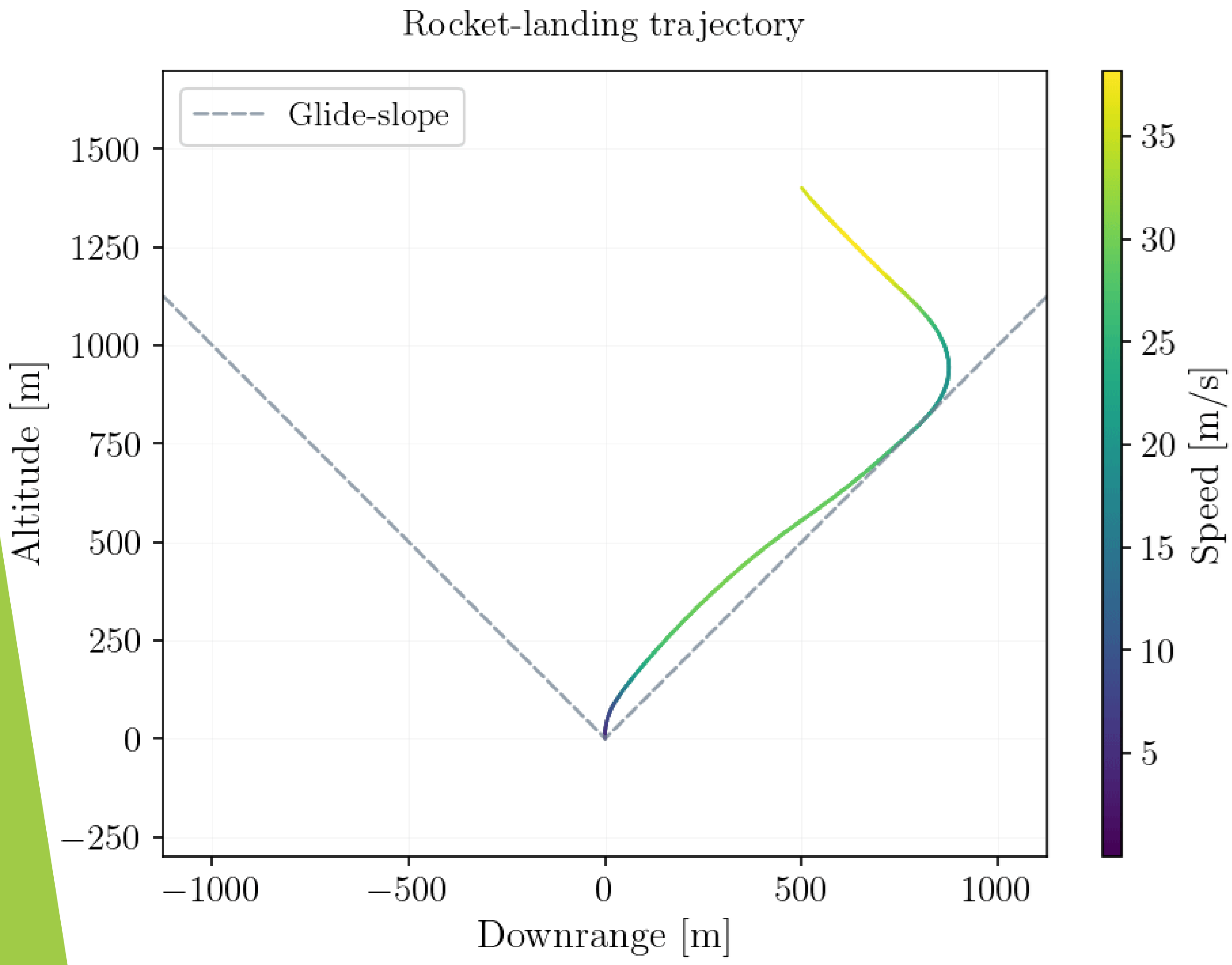




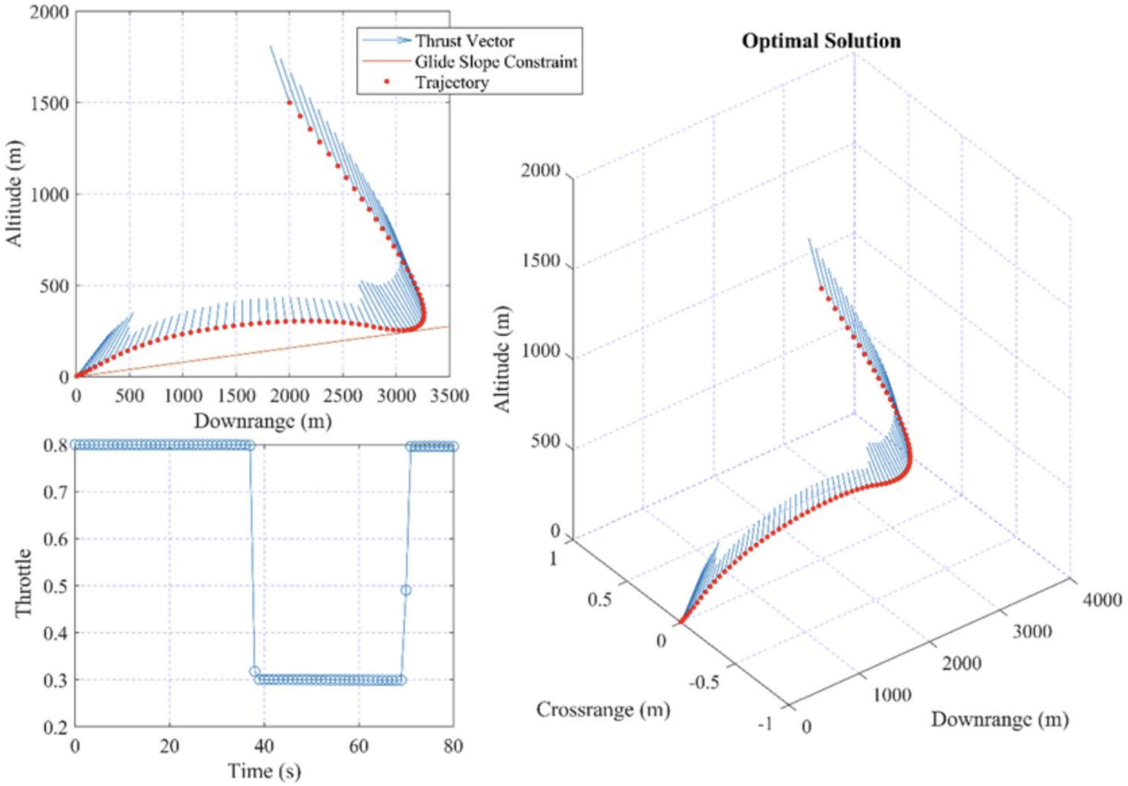




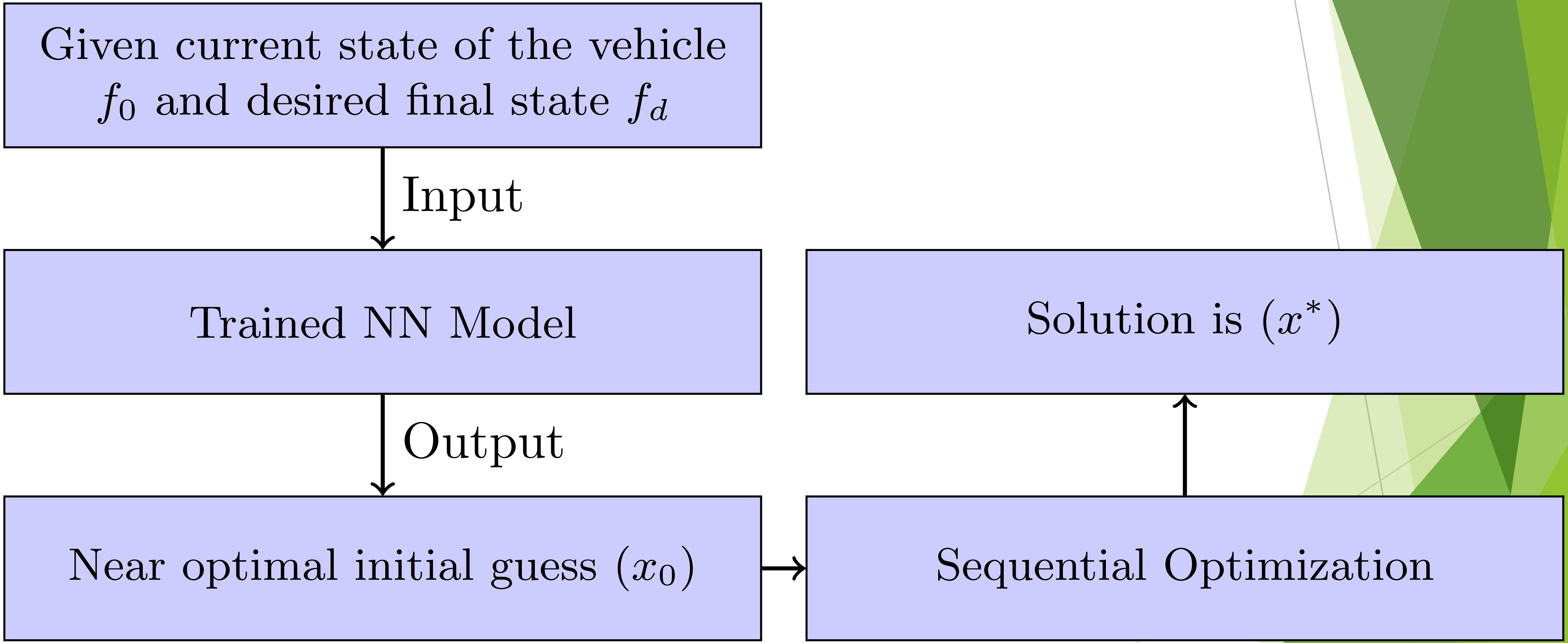
Optimal Solution Example



Optimal Solution Example



Proposed Solution: Learning



Motivation

Challenges in real-time computation:

Computationally heavy

Hard to get the solution in real-time

Neural Networks:

Pre-trained, give the input, get the output in real-time

Forward pass is computationally cheap

Use NN to assist in decreasing the computational time of the optimization problem

Goals and Incentives

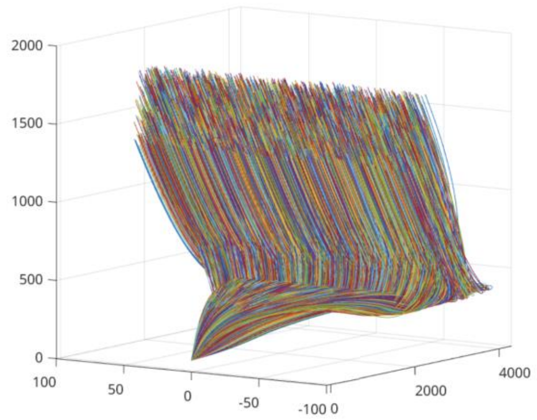
Objective: Improve time-sensitive performance in rocket trajectory optimization that minimizes fuel consumption

Approach: Model and train two types of Neural Networks

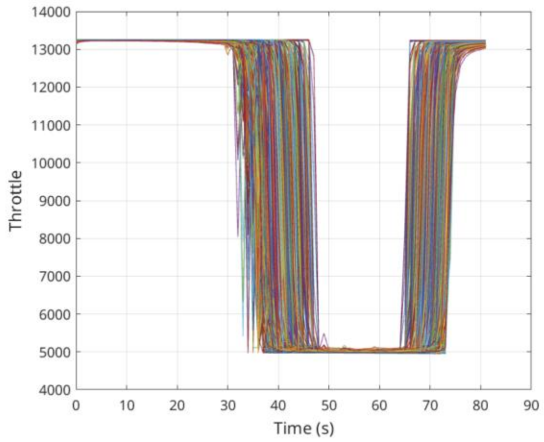
Primary Goal: Approximate the optimal solution for the convexified optimization problem

Training Data:

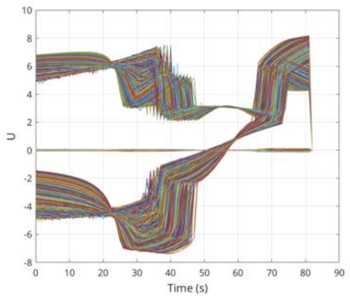
- 6000 generated trajectories
- Initial conditions based on normal distribution from pseudo-real data



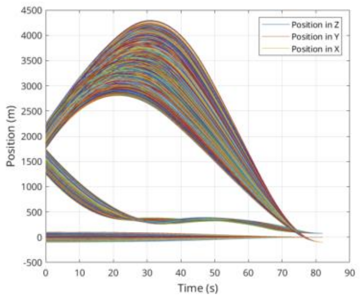
3D Trajectories for 6000 runs.



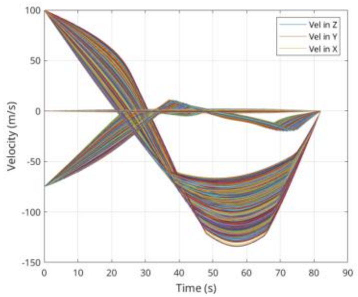
Norm of Thrust vector for 6000 runs.



Thrust Components plotted separately for 6000 runs.



Position plotted separately for 6000 runs.



Norm of Thrust vector for 6000 runs.

Training Data

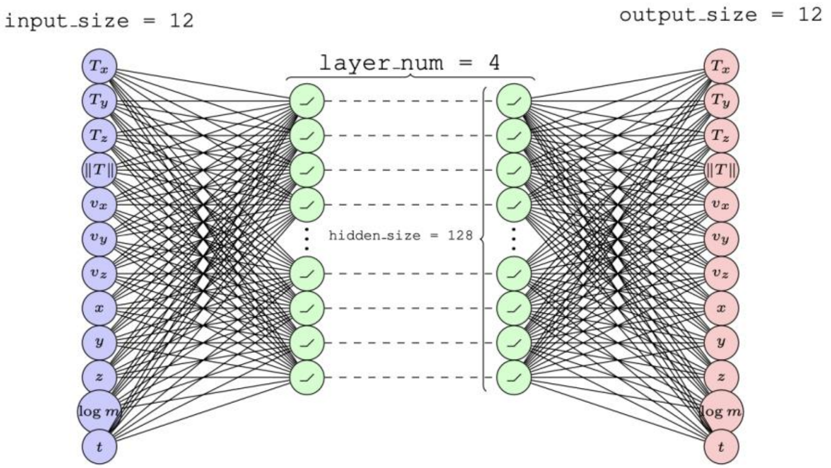
- **Input (**6,000 Trajectories; 83 time-stamps each**)**

T_x	T_y	T_z	$\|T\|$	v_x	v_y	v_z	x	y	z	$\log m$	t
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

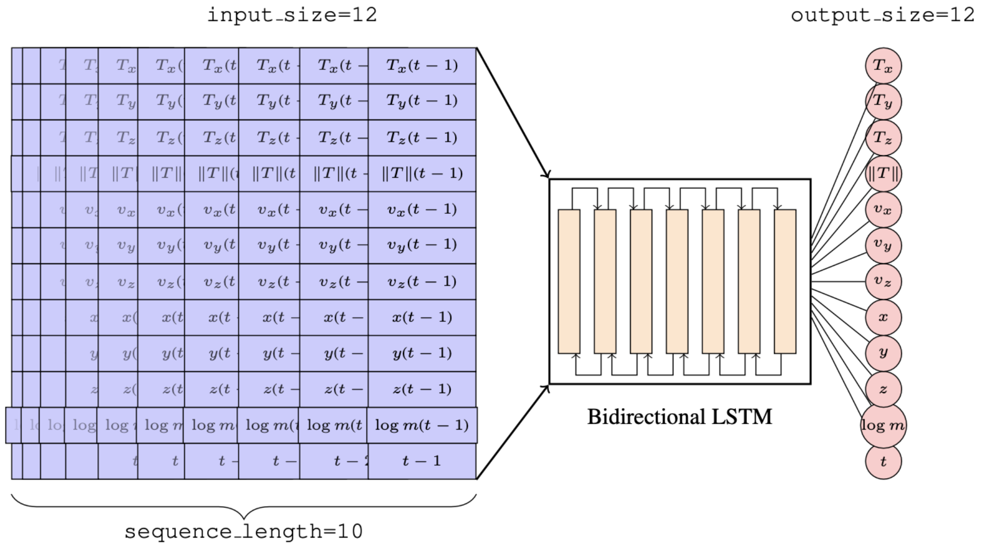
- T_x, T_y, T_z : Coordinate components of the thrust vector, T.
- $||T||$: The norm of the thrust vector.
- v_x, v_y, v_z : Coordinate components of the velocity vector, v.
- x, y, z: Coordinate components of the position vector.
- $log\ m$: The logarithmic value of the rocket's mass.

Model Architecture

DNN



LSTM



Model Training

- **DNN**
  - Each time step is treated independently
  - No explicit modeling of temporal dependencies.
  - Input features for each time step are passed directly to the fully connected network.

- **RNN**
  - Sequential data is provided as input
  - With a fixed sequence length used to capture temporal dependencies.
  - Each sequence comprises multiple time steps, allowing the model to learn contextual relationships.

$$\mathcal{L}(\mathbf{X}^t, \mathbf{Y}^t) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i^t - \mathbf{y}_i^t\|_2^2$$

$$\mathbf{X}^t = \begin{bmatrix} \mathbf{x}_1^{t\top} \\ \mathbf{x}_2^{t\top} \\ \vdots \\ \mathbf{x}_n^{t\top} \end{bmatrix}, \mathbf{Y}^t = \begin{bmatrix} \mathbf{y}_1^{t\top} \\ \mathbf{y}_2^{t\top} \\ \vdots \\ \mathbf{y}_n^{t\top} \end{bmatrix}$$

Training Environment

Training Environment

PyTorch.DataLoader

$\downarrow$

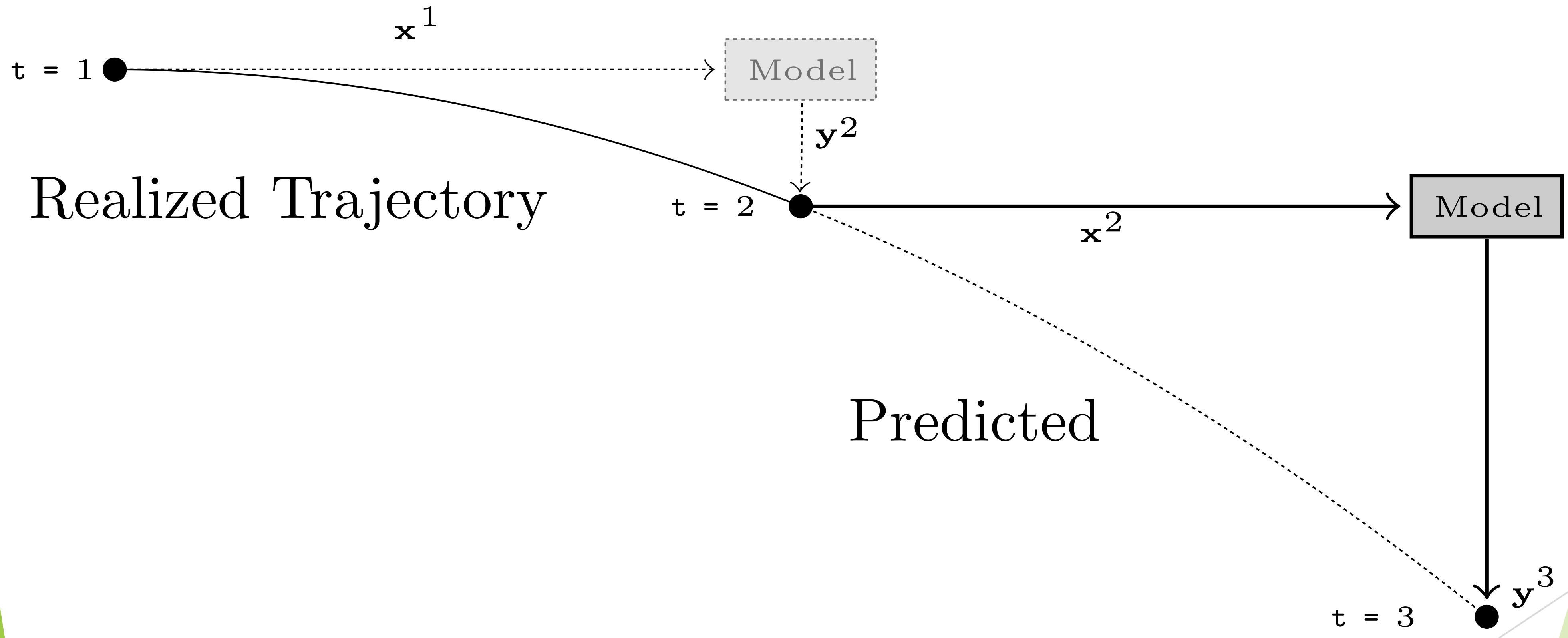
batch_size=32

$\mathbf{X}^{t_1}, \mathbf{X}^{t_2}, \dots \mathbf{X}^{t_{\max}}$

Model Prediction

- **DNN**
  - Give the initial guess to the model.
  - Pass the output of the model iteratively.
  - After 82 passes, we get the final state.
- **RNN**
  - Give the initial sequential guesses to the model.
  - Pass the output of the model iteratively.
  - After (82 – `initial_sequence_length`) passes, we get the final state.

Model Prediction



Testing

- **Targets**
  - The ground truth final state (12 features)
- **Predictions**
  - The predicted final state (12 features)
- **Metrics**
  - Mean Squared Error (MSE)
  - Mean Absolute Error (MAE)
  - Root Mean Squared Error (RMSE)

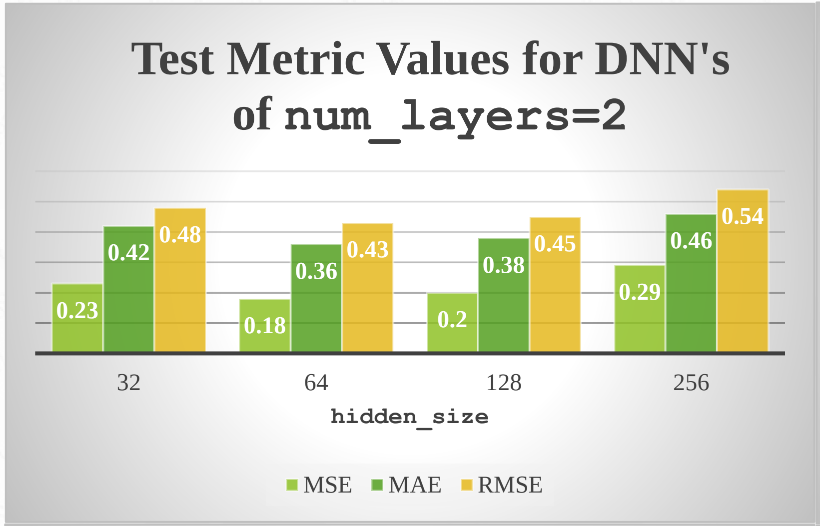
Testing

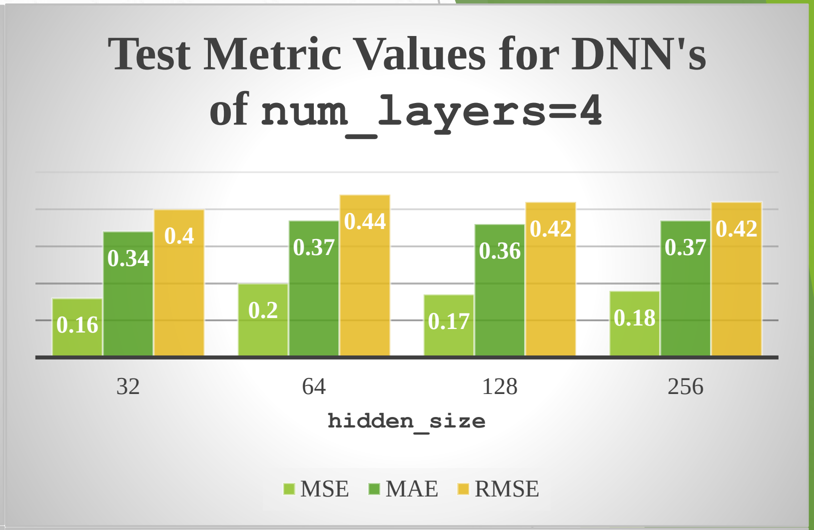
$$\mathrm{MSE} = \frac{1}{\mathrm{n\_trajectories}} \sum_{i=1}^{\mathrm{n\_trajectories}} (\mathbf{x}_i^\top - \mathbf{y}_i^\top)^2$$

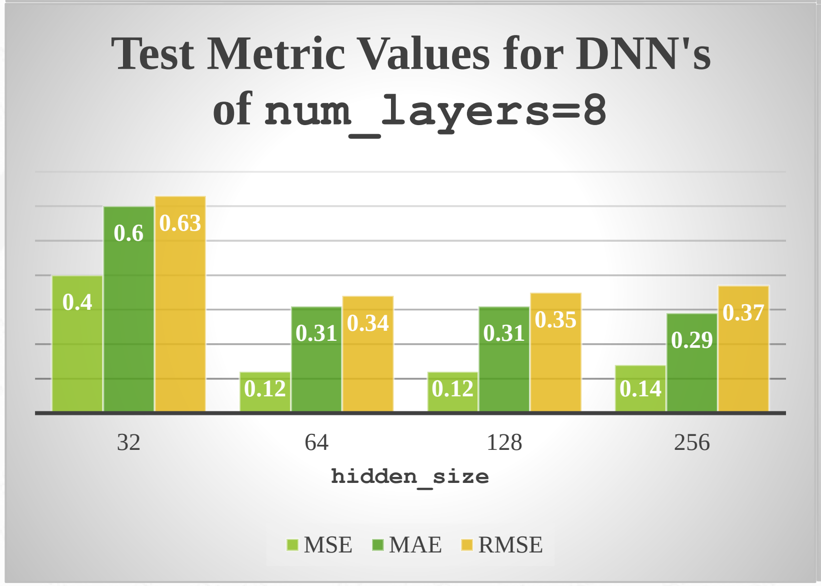
$$\mathrm{MAE} = \frac{1}{\mathrm{n\_trajectories}} \sum_{i=1}^{\mathrm{n\_trajectories}} |\mathbf{x}_i^\top - \mathbf{y}_i^\top|$$

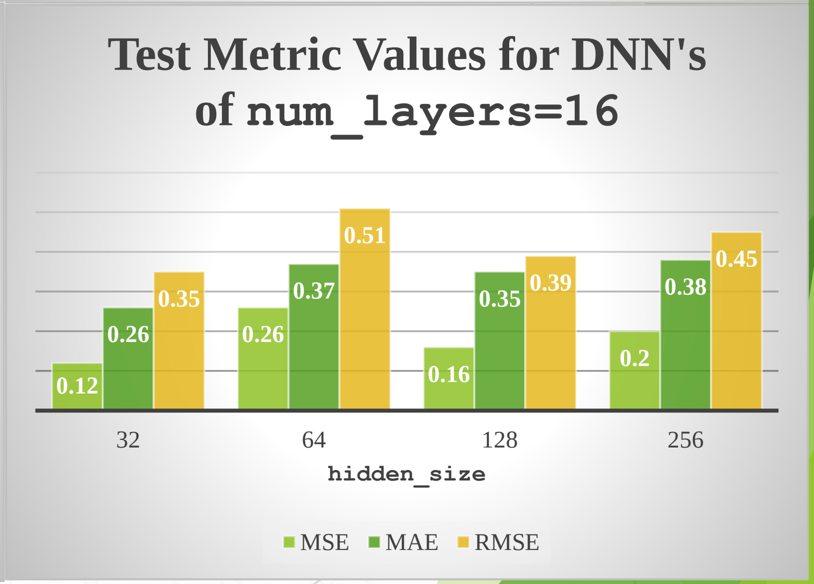
$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

DNN Results

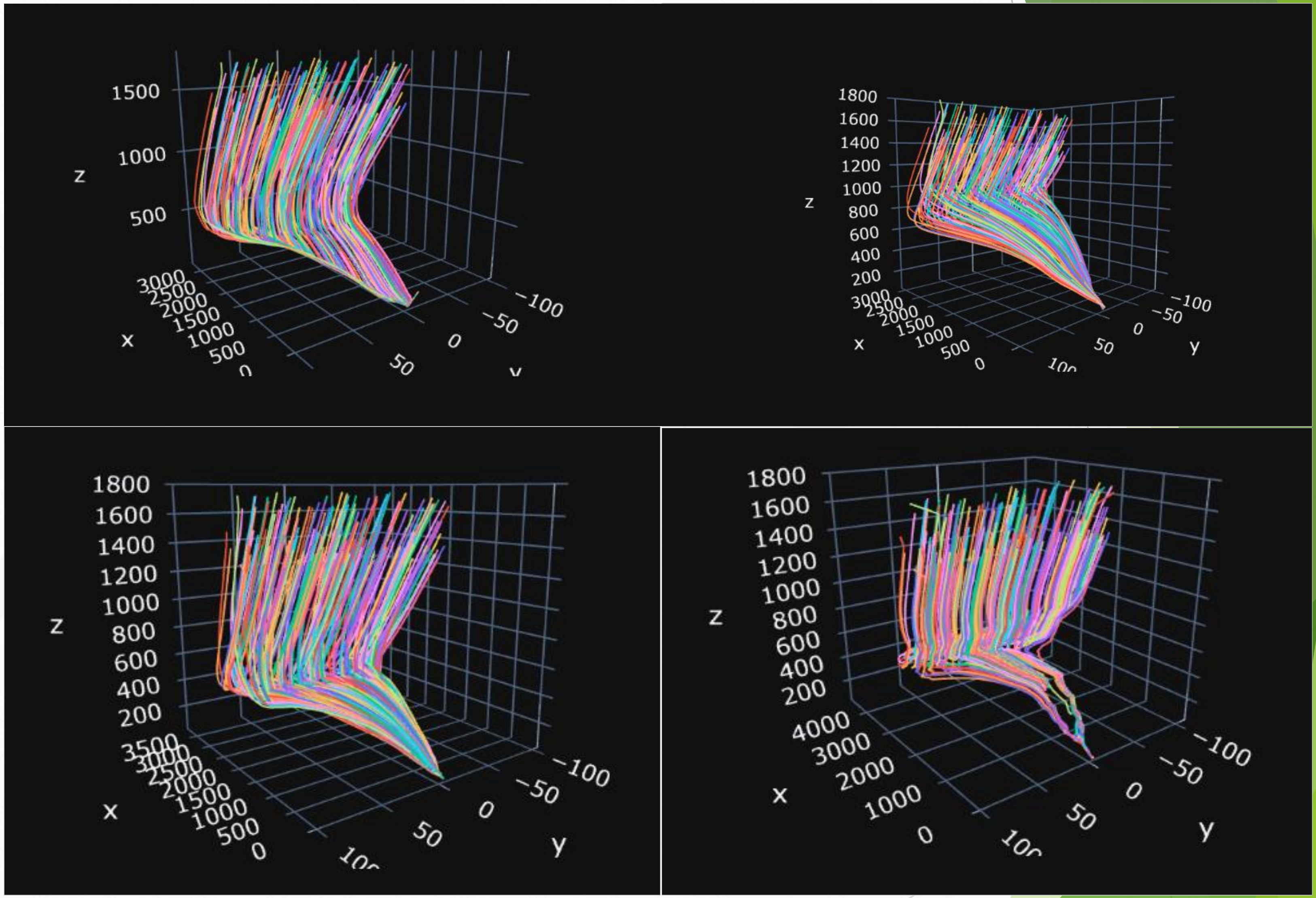

Test Metric Values for DNN's of num_layers=2
0.23
0.42
0.48
0.18
0.36
0.43
0.2
0.38
0.45
0.29
0.46
0.54
32
64
128
256
hidden_size
MSE
MAE
RMSE


Test Metric Values for DNN's of num_layers=4
0.16
0.34
0.4
0.2
0.37
0.44
0.17
0.36
0.42
0.18
0.37
0.42
32
64
128
256
hidden_size
MSE
MAE
RMSE

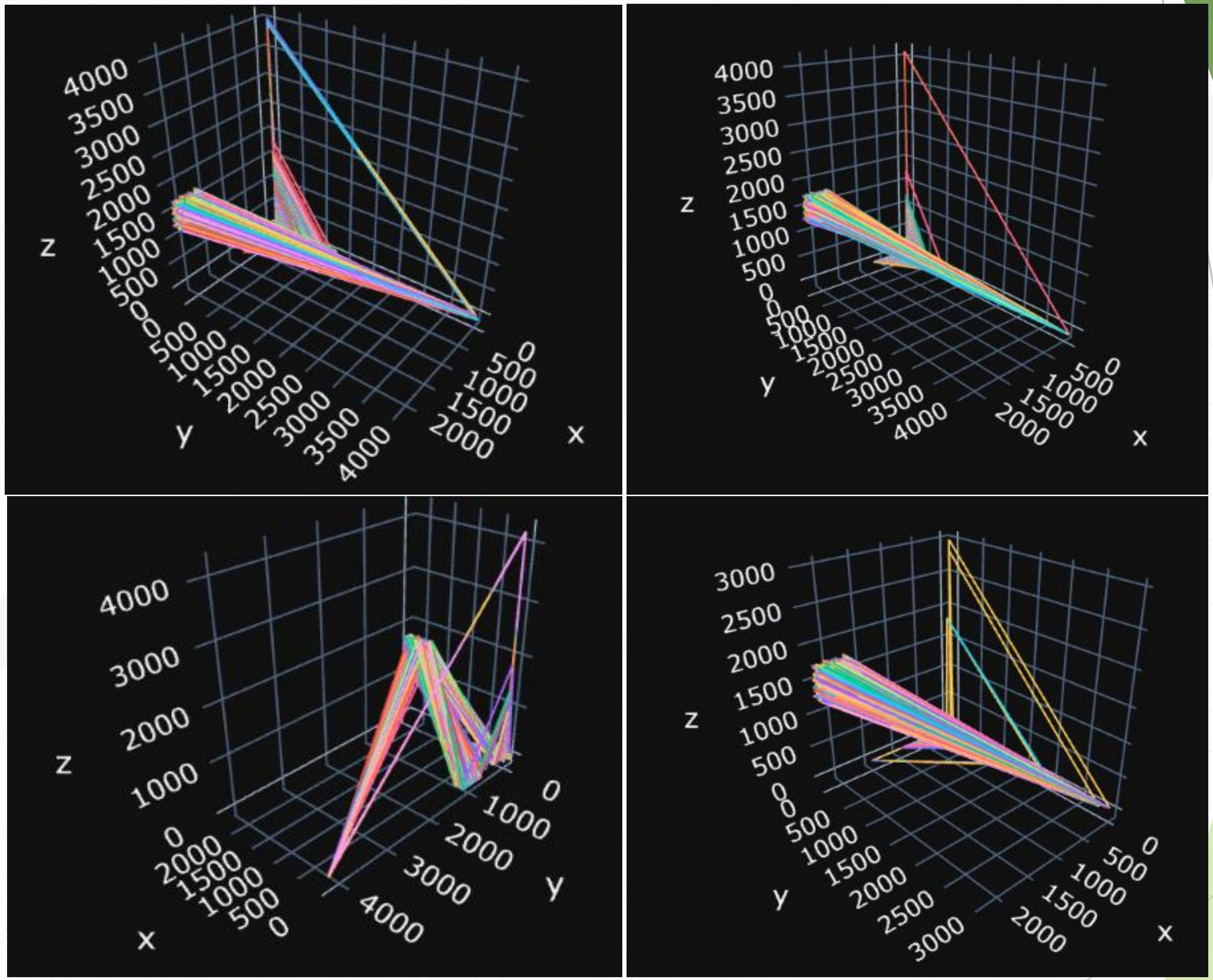

Test Metric Values for DNN's of num_layers=8
0.4
0.6
0.63
0.12
0.31
0.34
0.12
0.31
0.35
0.14
0.29
0.37
32
64
128
256
hidden_size
MSE
MAE
RMSE


Test Metric Values for DNN's of num_layers=16
0.12
0.26
0.35
0.26
0.37
0.51
0.16
0.35
0.39
0.2
0.38
0.45
32
64
128
256
hidden_size
MSE
MAE
RMSE

DNN Visualization



Results


4000
3500
3000
2500
2000
1500
1000
500
0
z
y
x
500
1000
1500
2000
2500
3000
3500
4000
0
500
1000
1500
2000

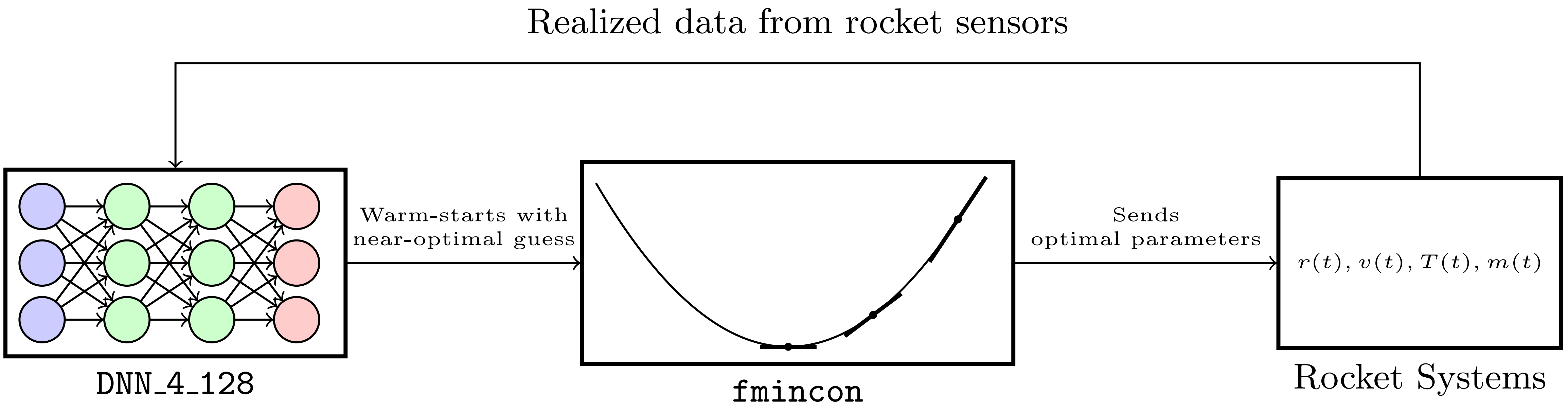
Possible Reasons for the Bad Performance of LSTM’s

Data Characteristics:

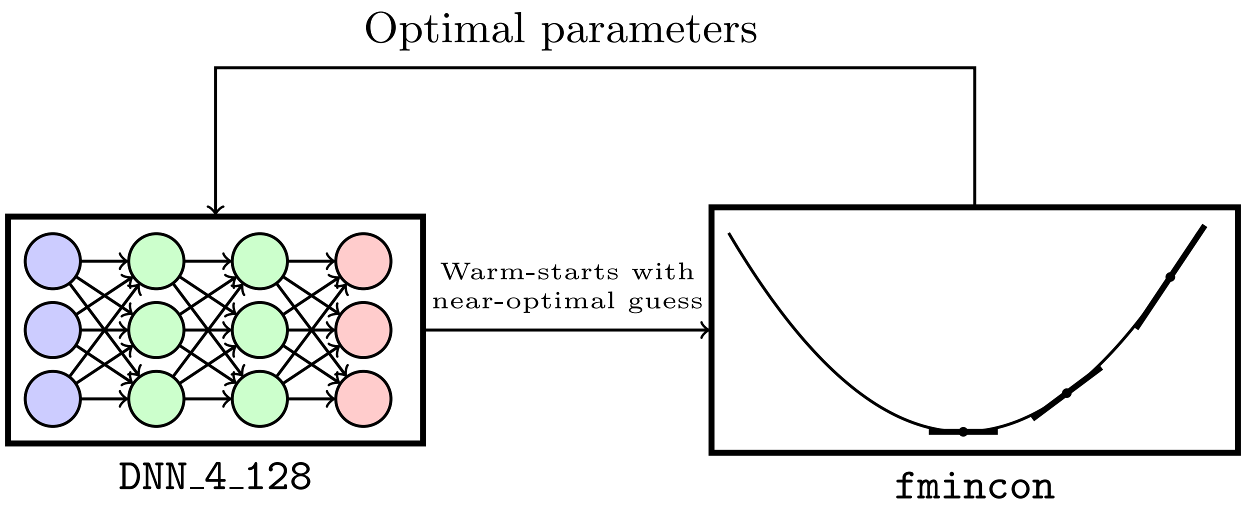
The dataset might not exhibit complex temporal dependencies that LSTMs are designed to capture.

Model Complexity:

The added complexity of LSTMs requires more computational resources and may struggle in scenarios where temporal dependencies are not dominant.

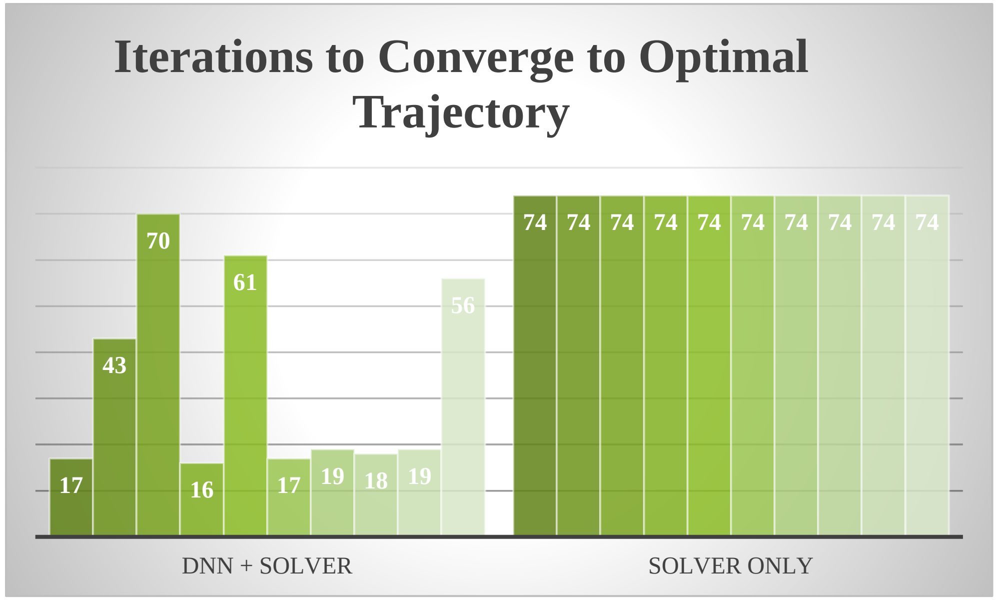


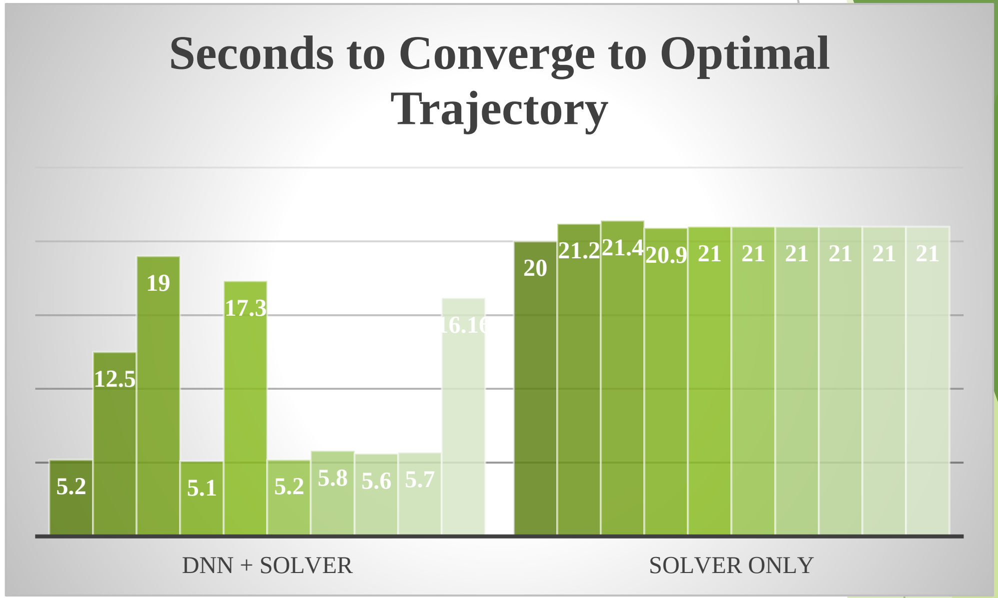
Warm-Starting the Mathematical Solver



Warm-Starting the Mathematical Solver

Time Gains





- On average, twice as fast!

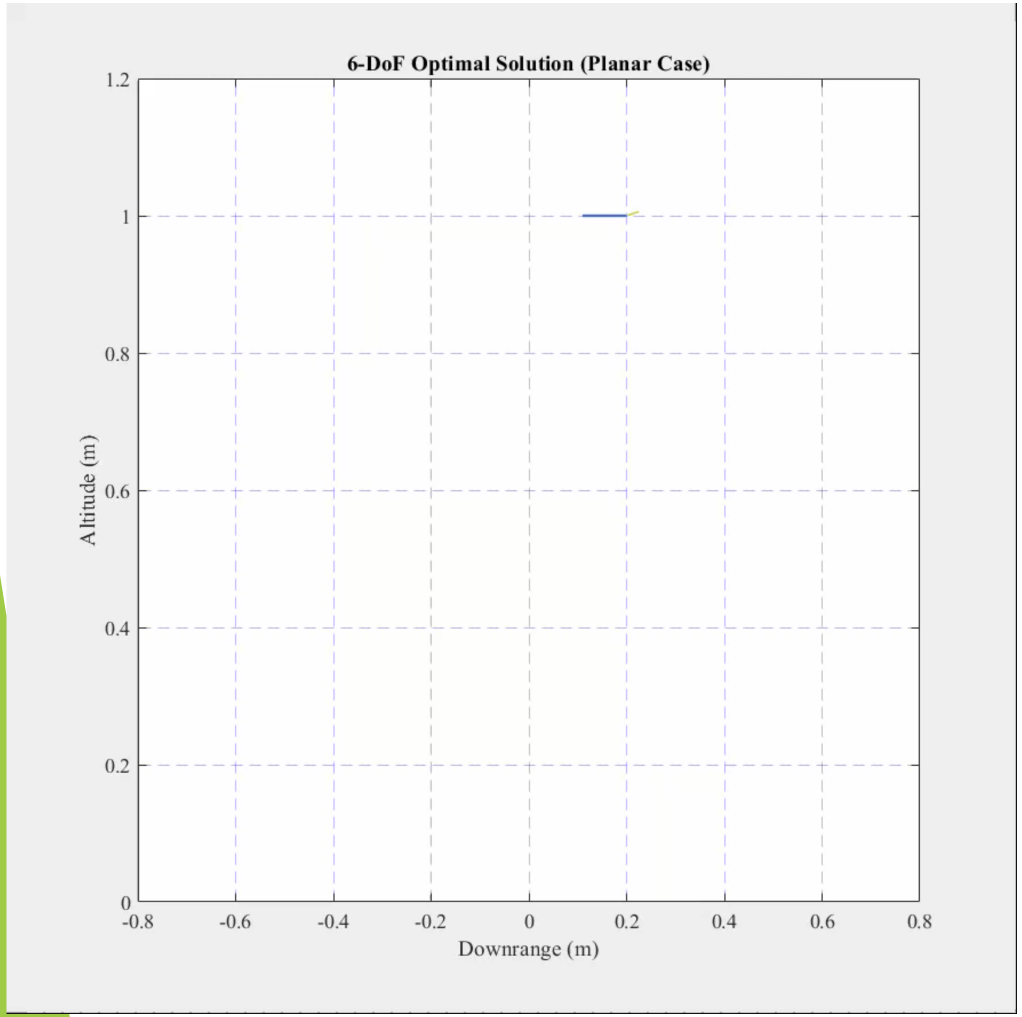
Future Work:

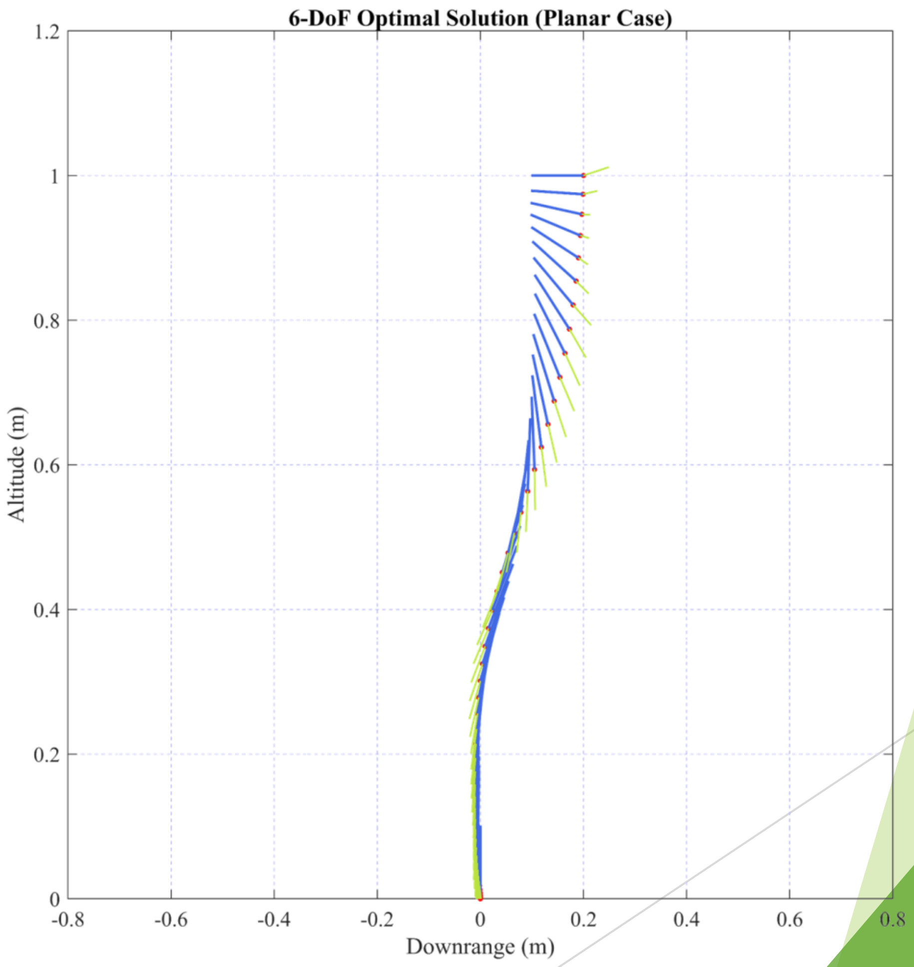
- Recent research papers show better results using Transforms but require data of the order of 50,000 trajectories to train.

Future missions to Mars and Moon to Replicate Similar Landings

Future Work

- Complete 6-DOF modelling for more challenging space missions.
- (Open problem <1 Sec)





Q/A

References

- Behcet Acikmese and Scott R. Ploen. Convex programming approach to powered descent guidance for mars landing. Journal of Guidance, Control, and Dynamics, 30(5):1353–1366, 2007. doi:10.2514/1.27553. https://doi.org/10.2514/1.27553
- Lars Blackmore, Behcet Acikmese, and Daniel P. Scharf. Minimum-landing-error powered-descent guidance for mars landing using convex optimization. Journal of Guidance, Control, and Dynamics, 33(4):1161–1171, 2010. doi: 10.2514/1.47202. https://doi.org/10.2514/1.47202
- Julia Briden, Trey Gurga, Breanna J. Johnson, Abhishek Cauligi, and Richard Linares. Improving Computational Efficiency for Powered Descent Guidance via Transformer-based Tight Constraint Prediction. 2024. doi: 10.2514/6.2024-1760. https://arc.aiaa.org/doi/abs/10.2514/6.2024-1760
- John M. Carson, Michelle M. Munk, Ronald R. Sostaric, Jay N. Estes, Farzin Amzajerdian, James B.Blair, David K. Rutishauser, Carolina I. Restrepo, Alicia M. Dwyer-Cianciolo, George Chen, and Teming Tse. The SPLICE Project: Continuing NASA Development of GN&C Technologies for Safe and Precise Landing. doi: 10.2514/6.2019-0660. https://arc.aiaa.org/doi/abs/10.2514/6.2019-0660
- Matthew Fritz, Javier Doll, Kari C. Ward, Gavin Mendeck, Ronald R. Sostaric, Sam Pedrotty, ChrisKuhl, Behcet Acikmese, Stefan R. Bieniawski, Lloyd Strohl, and Andrew W. Berning. Post-Flight Performance Analysis of Navigation and Advanced Guidance Algorithms on a Terrestrial Suborbital Rocket Flight. doi: 10.2514/6.2022-0765. https://arc.aiaa.org/doi/abs/10.2514/6.2022-0765

References

- Abhinav G. Kamath, Purnanand Elango, Yue Yu, Skye Mceowen, Govind M. Chari, John M.Carson III, and Behcet Acıkmese. Real-time sequential conic optimization for multi-phaserocket landing guidance. IFAC-Papers OnLine, 56(2):3118–3125, 2023. ISSN 2405-8963. doi:https://doi.org/10.1016/j.ifacol.2023.10.1444. https://www.sciencedirect.com/science/article/pii/S2405896323018529 22nd IFAC World Congress.
- Ping Lu. Propellant-optimal powered descent guidance. Journal of Guidance, Control, and Dynamics, 41(4):813–826, 2018. doi: 10.2514/1.G003243. https://doi.org/10.2514/1.G003243
- National Academy of Engineering. Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium. National Academies Press, 2017. ISBN 9780309450393. https://books.google.com/books?id=tCFCDgAAQBAJ
- Taylor P. Reynolds, Michael Szmuk, Danylo Malyuta, Mehran Mesbahi, Behcet Acıkmese, and John M. Carson. Dual quaternion-based powered descent guidance with state-triggered constraints. Journal of Guidance, Control, and Dynamics, 43(9):1584–1599, 2020. doi: 10.2514/1.G004536. https://doi.org/10.2514/1.G004536
- Daniel P. Scharf, Behcet Acıkmese, Daniel Dueri, Joel Benito, and Jordi Casoliva. Implementation and experimental demonstration of onboard powered-descent guidance. Journal of Guidance, Control, and Dynamics, 40(2):213–229, 2017. doi: 10.2514/1.G000399. https://doi.org/10.2514/1.G000399
- Zhipeng Shen, Shiyu Zhou, and Jianglong Yu. Real-time computational powered landing guidance using convex optimization and neural networks. arXiv preprint arXiv:2210.07480, 2022.