Physics-Inspired Deep Learning Techniques for Modeling and Control of Complex Aircraft

Student: Ethan Fitz

Student Email: fitzed@mail.uc.edu

Faculty: Manish Kumar and Prashant Khare

Faculty Email: kumarmu@ucmail.uc.edu and kharept@ucmail.uc.edu

AFRL Sponsor: Michael Bolender and David Casbeer

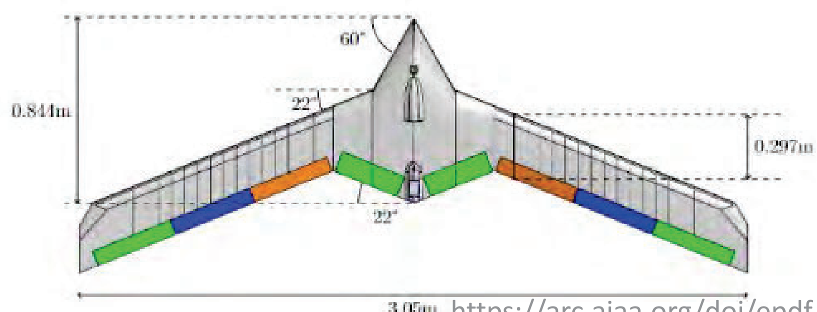
AFRL Directorate: AFRL/RQ

Approved for public release: distribution unlimited

PA #: AFRL-2023-5282

Motivation

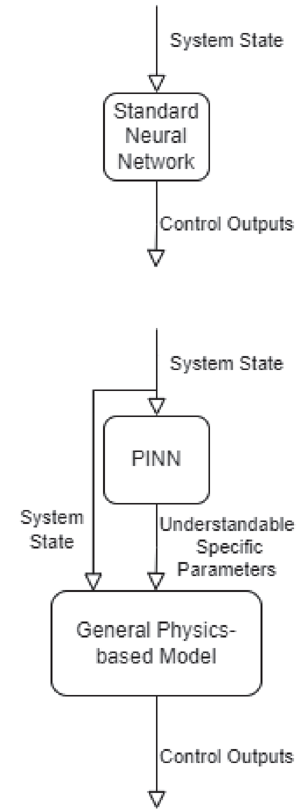
- Traditional model based control methods may not work well for certain complex aircraft or extreme changes in vehicle parameters.
  - However, traditional controllers can be verified and do not require training data
- AI-based control methods can handle these cases
  - However, AI-based controllers are black boxes and require a large training data set
- A controller that combines both strategies could potentially outperform either alone, while mitigating some of the disadvantages



https://arc.aiaa.org/doi/epdf/10.2514/6.2016-1753

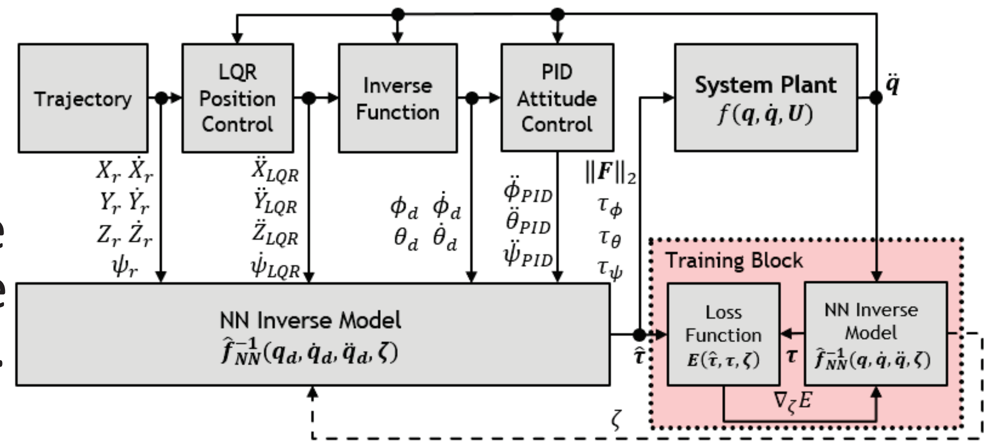
Introduction

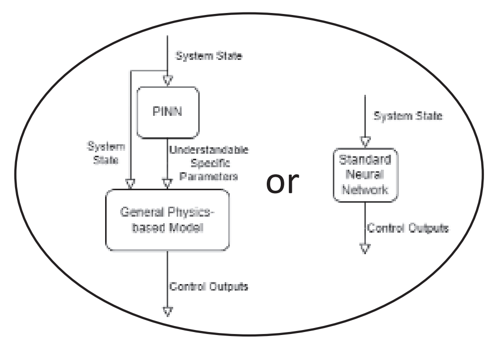
- A Physics-Informed Neural Network (PINN) is a neural network that uses prior physics knowledge as part of its structure or loss function.
- Integrating neural networks and physics models like this combines their advantages and can also result in
  - Reduced training data requirements
  - Increased generalizability
  - Human-readable neural network output
- A type of PINN known as a Deep Lagrangian Neural Network (DeLaN) has been developed to control a simulated quadrotor UAV and is being tested against more established controller types.
- Currently, its performance is being compared to that of a Feed Forward Neural Network (FFNN)-based controller.
- This presentation details the FFNN controller's development and comparison to the DeLaN controller.



General Structure

- Simulink is used to simulate a quadcopter performing a trajectory following task.
- LQR and PID controllers generate desired acceleration and attitude states from trajectory waypoints.
- Both controllers use an inverse model-based structure where the neural network serves as the inverse model of the vehicle, transforming the desired and current system states into actuation forces and torques to be applied to the plant.

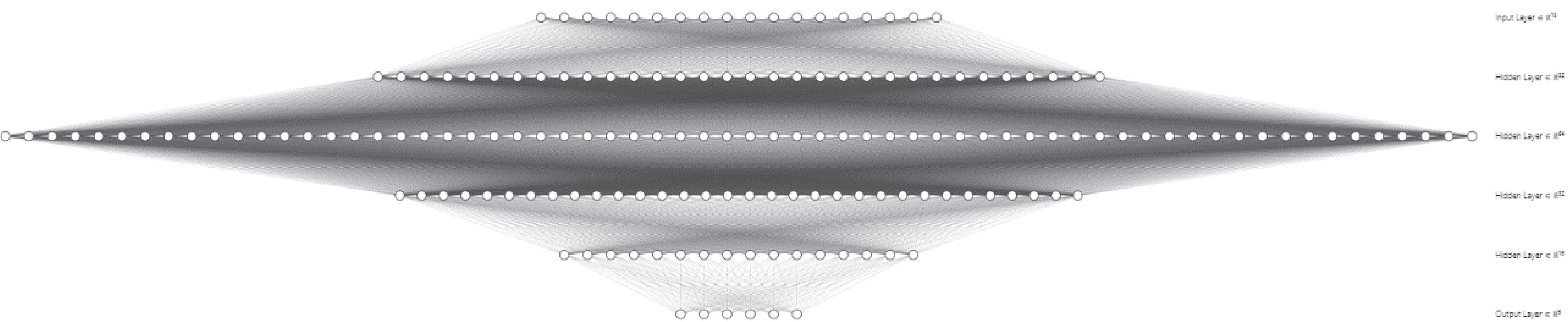




Bennet Breese

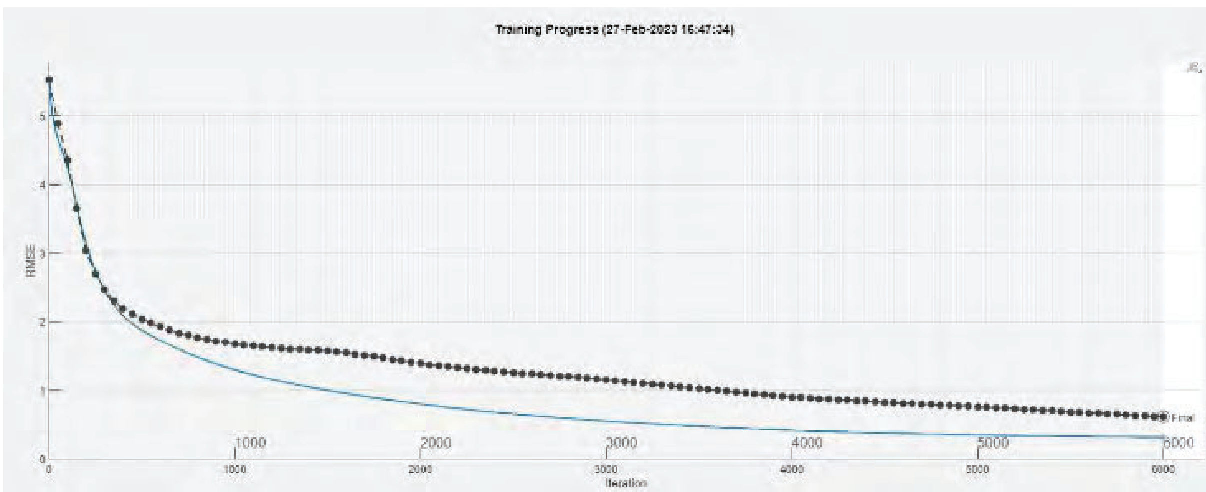
FFNN Structure

- Six fully connected layers
- Maximum width of 64 neurons
- Input is an 18-value system state vector
- Output is a vector of six component forces and torques



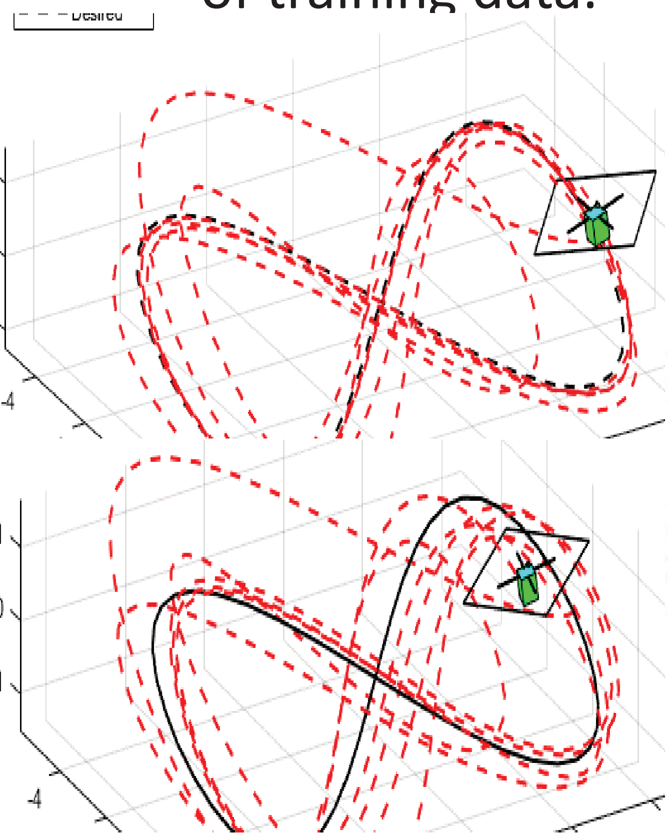
FFNN Training

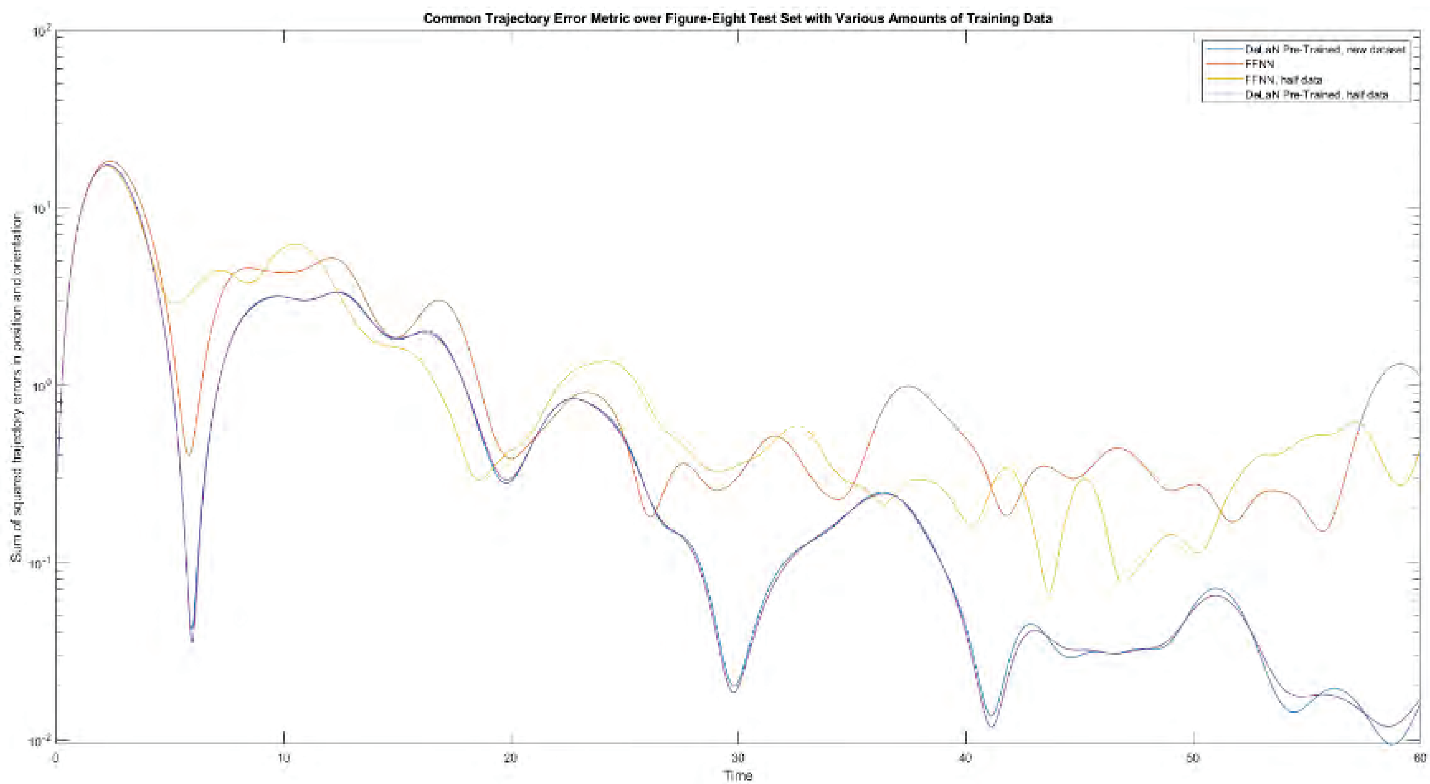
- The training dataset for both the DeLaN and the FFNN was generated by subjecting the simulated quadcopter to random forces and torques and collecting the resulting system states.
- The dataset contains pairs of states and actuation forces that were the inputs and desired outputs for the network.



Offline Learning Results

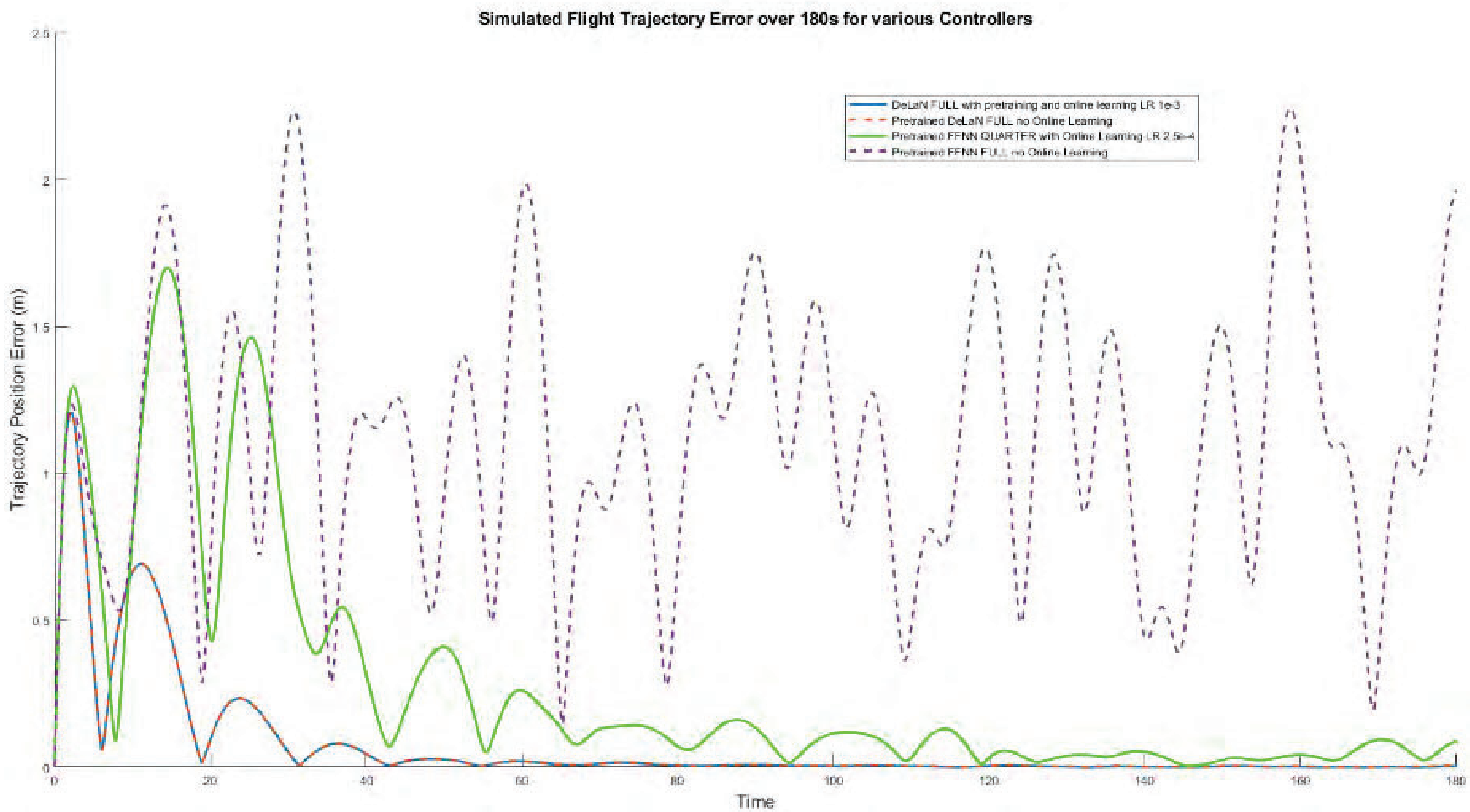
- The Pre-trained FFNN controller was tested against the DeLaN controller in a trajectory-following simulation, with varying amounts of training data.





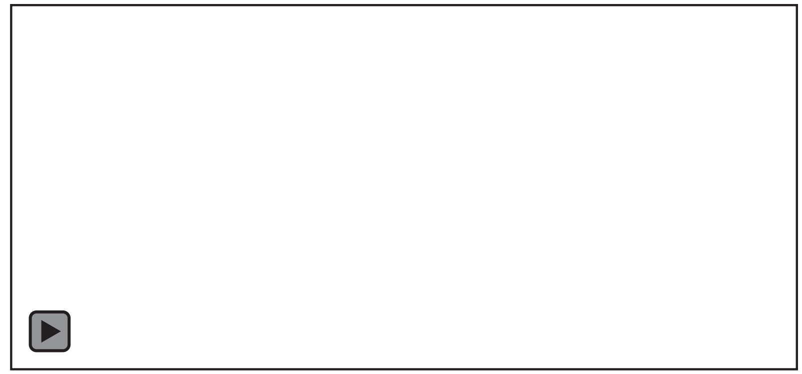
Online Learning Results

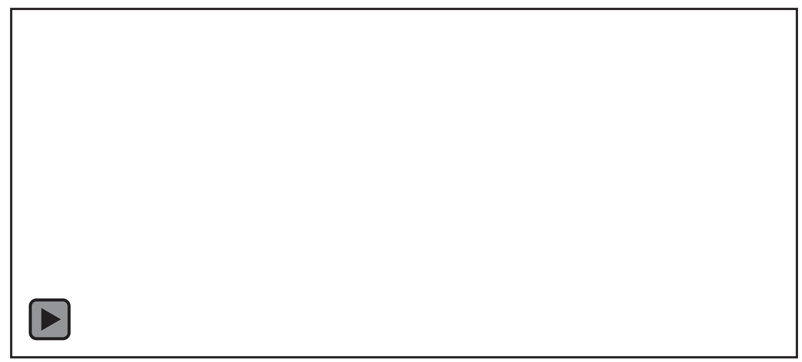
- Online Learning was implemented for the FFNN and compared against the DeLaN controller's online learning performance



Conclusions

- The Physics-Informed DeLaN controller shows superior performance at trajectory following compared to the ‘black box’ FFNN controller.
  - Lower trajectory error
  - Less training data required to achieve peak performance





Next Steps

- Now: Compare responses to sudden vehicle mass change during flight
- Soon: Real-time hardware test (Crazyflie UAV platform)
- Later: Implement DeLaN controller for more complex aircraft (flexible wing)

https://www.bitcraze.io/products/crazyflie-2-1/