TRAINING QUADCOPTERS TO RACE

TEAM FYRA

AMY PUENTE MISHARI ALIESA PUSHPREET SINGH HANSPAL SHILPA THOMAS

OBJECTIVE

Create a *robust* and *generalized* quadrotor control policy which will allow a simulated quadrotor to follow a trajectory in a near-optimal manner.

Utilize an *OpenAI Gym* environment as the simulation and train using *Reinforcement Learning*.

RELATED WORK

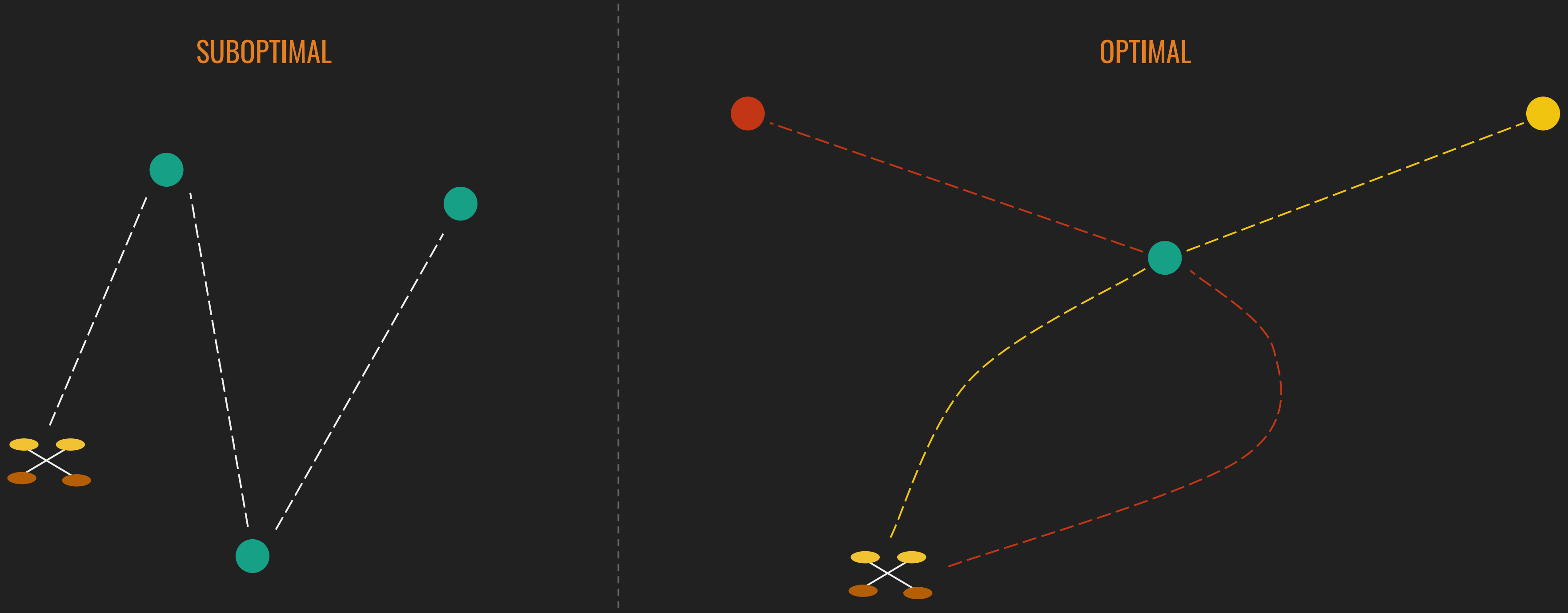
Our work is an extension of research being done in the *Robotic Embedded Systems Laboratory* (RESL) at the University of Southern California.

Molchanov et al. learned a unified quadrotor control network to stabilize a quadrotor that *hovers* at a specified point.

Our project *extends* this work by focusing on *path following*.

THE PROBLEM AND SOME IDEAS

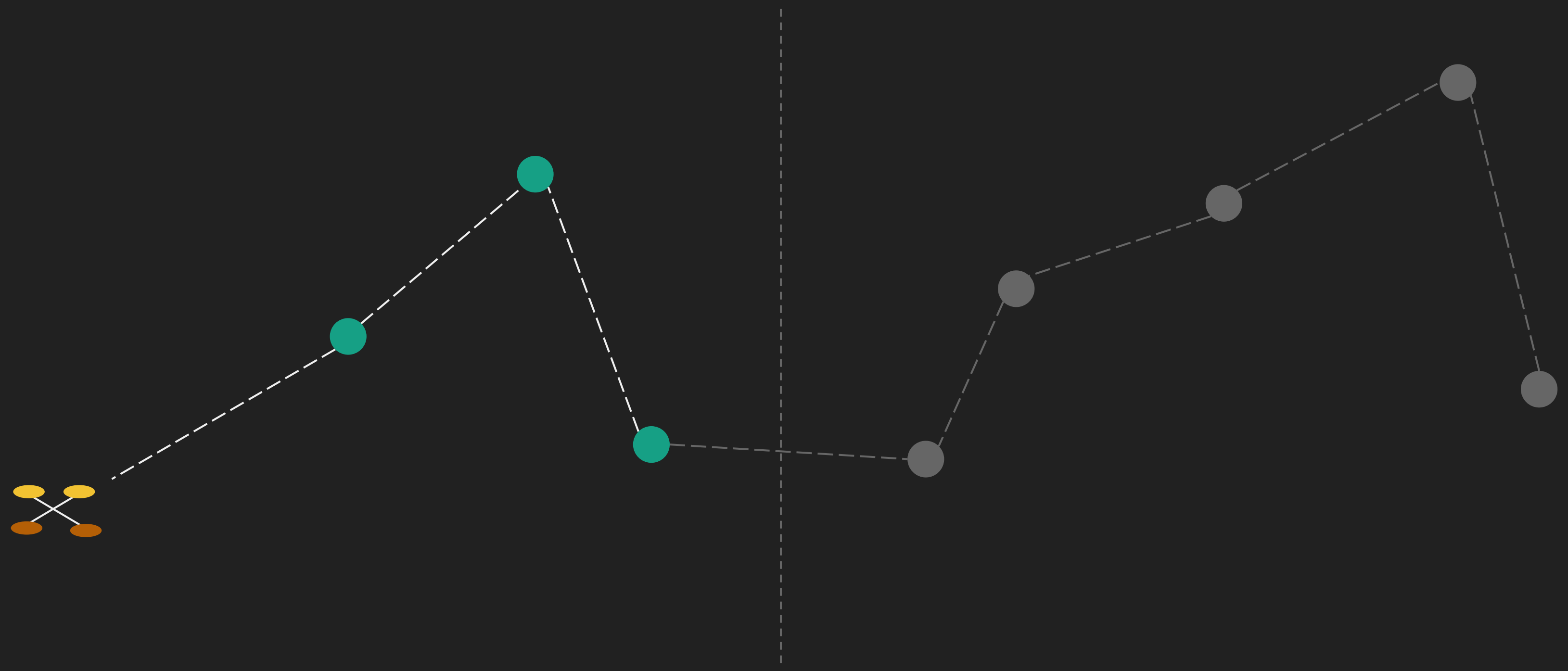
TRAJECTORY PLANNING?



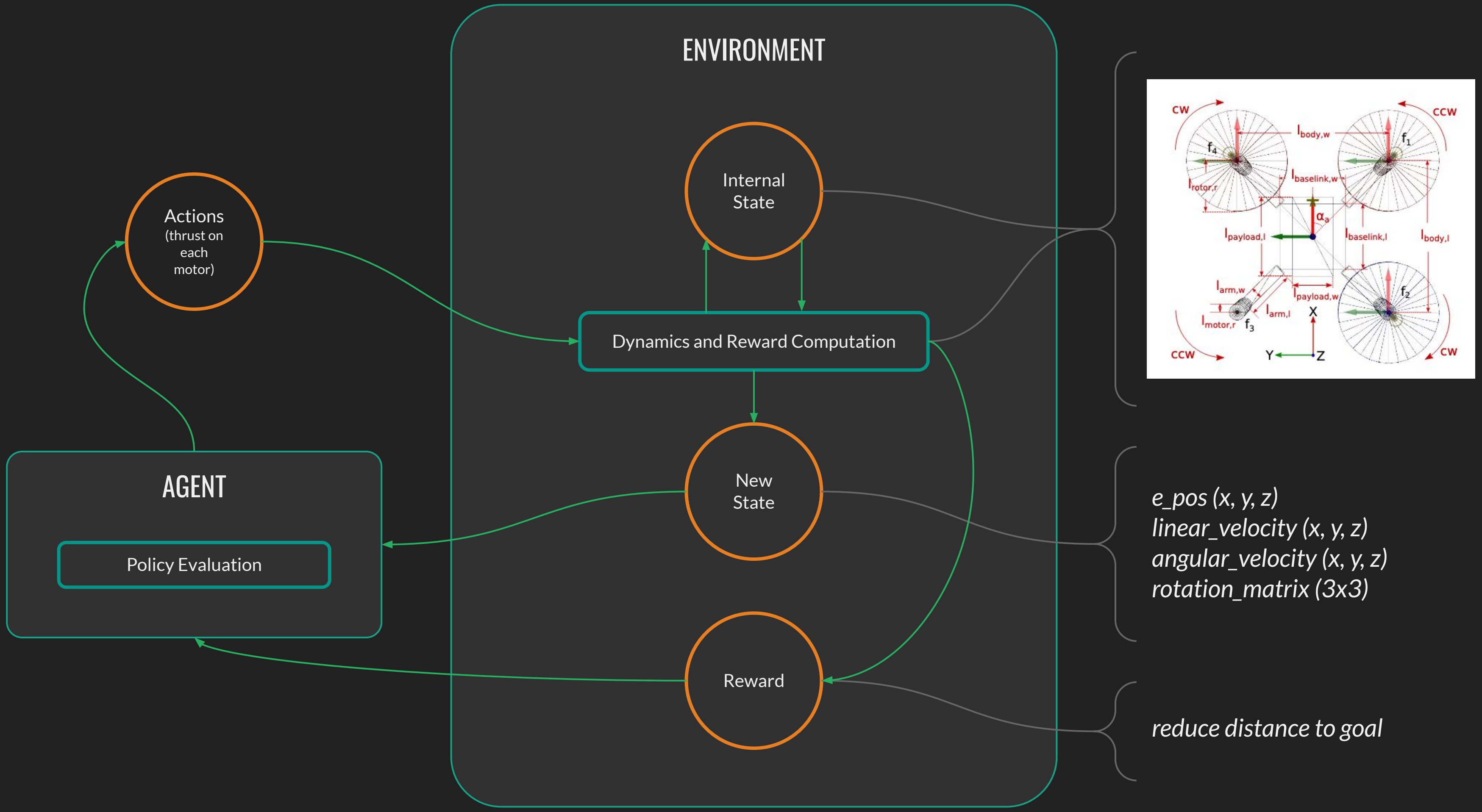
An optimal policy should strike a balance between tracking accuracy and the time-to-completion.

TRAJECTORY PLANNING?

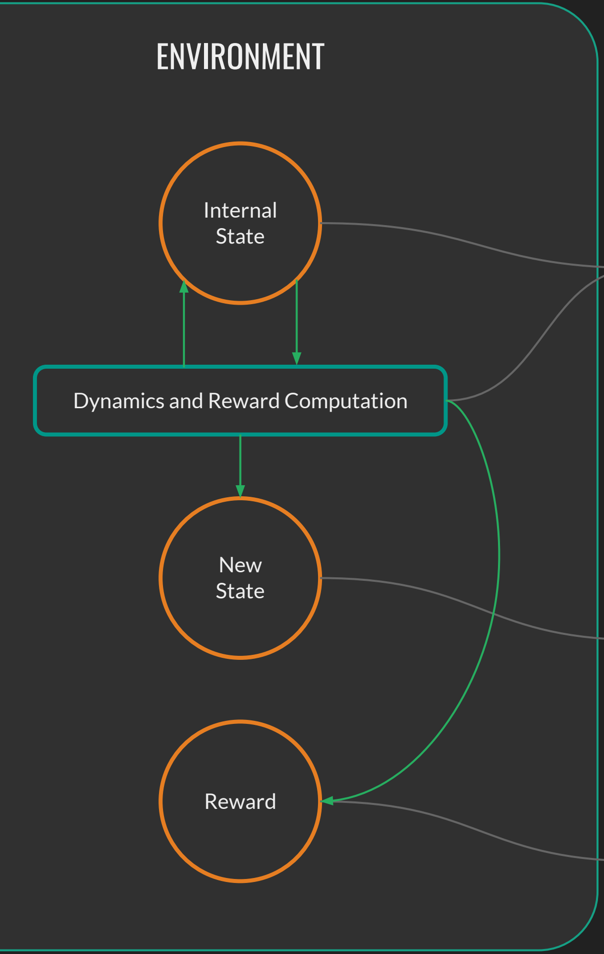
HOW FAR DO WE NEED TO LOOK?

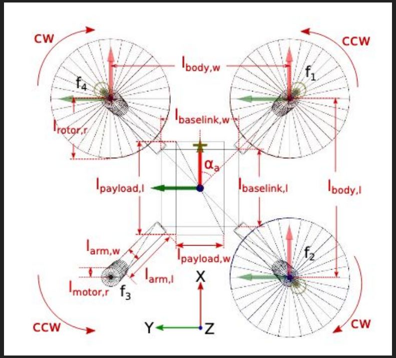


THE ENVIRONMENT



ENVIRONMENT
Internal State
Dynamics and Reward Computation
New State
Reward
Actions (thrust on each motor)
AGENT
Policy Evaluation
CW
CCW
CCW
CW
X
Y
Z
e_pos (x, y, z)
linear_velocity (x, y, z)
angular_velocity (x, y, z)
rotation_matrix (3x3)
reduce distance to goal





e_pos (x, y, z)
linear_velocity (x, y, z)
angular_velocity (x, y, z)
rotation_matrix (3x3)

reduce distance to goal

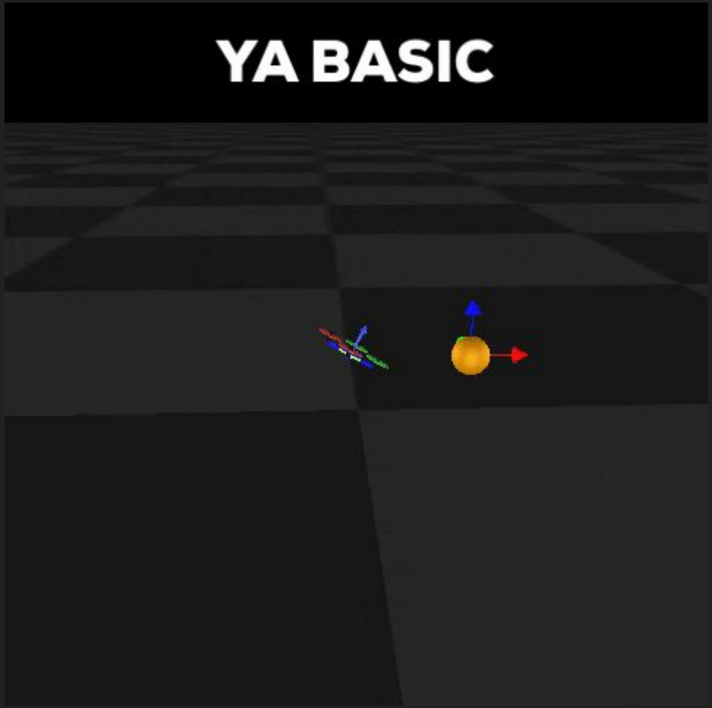
New state representation

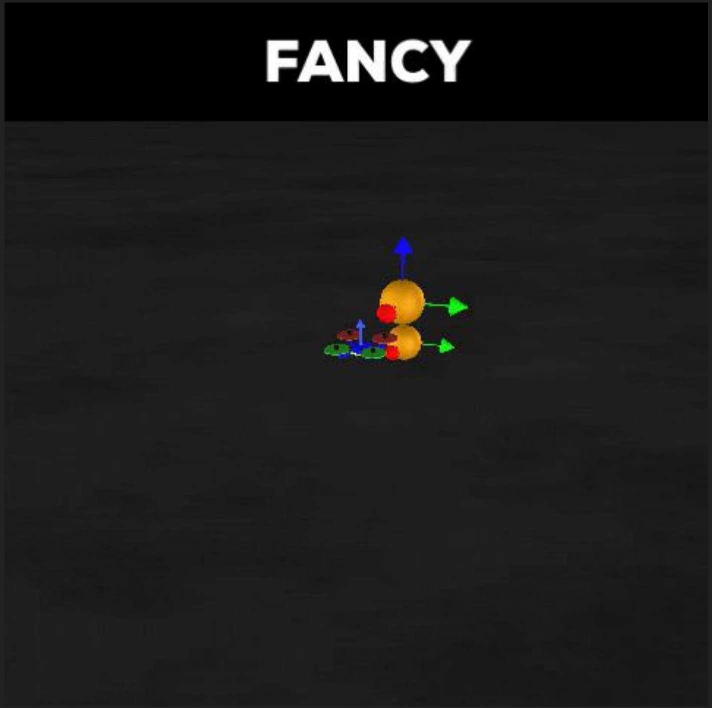
e_pos_goal1 (x, y, z)
e_pos_goal2 (x, y, z)
...
e_pos_goaln (x, y, z)
linear_velocity (x, y, z)
angular_velocity (x, y, z)
rotation_matrix (3x3)
reached_goal1 (bool)
reached_goal2 (bool)
...
reached_goaln (bool)

New reward functions

reduce dist to all goals in turn
only care about dist to current goal
reduce min_dist to goal
negative reward for not reaching goal
...

BETTER VISUALIZATION



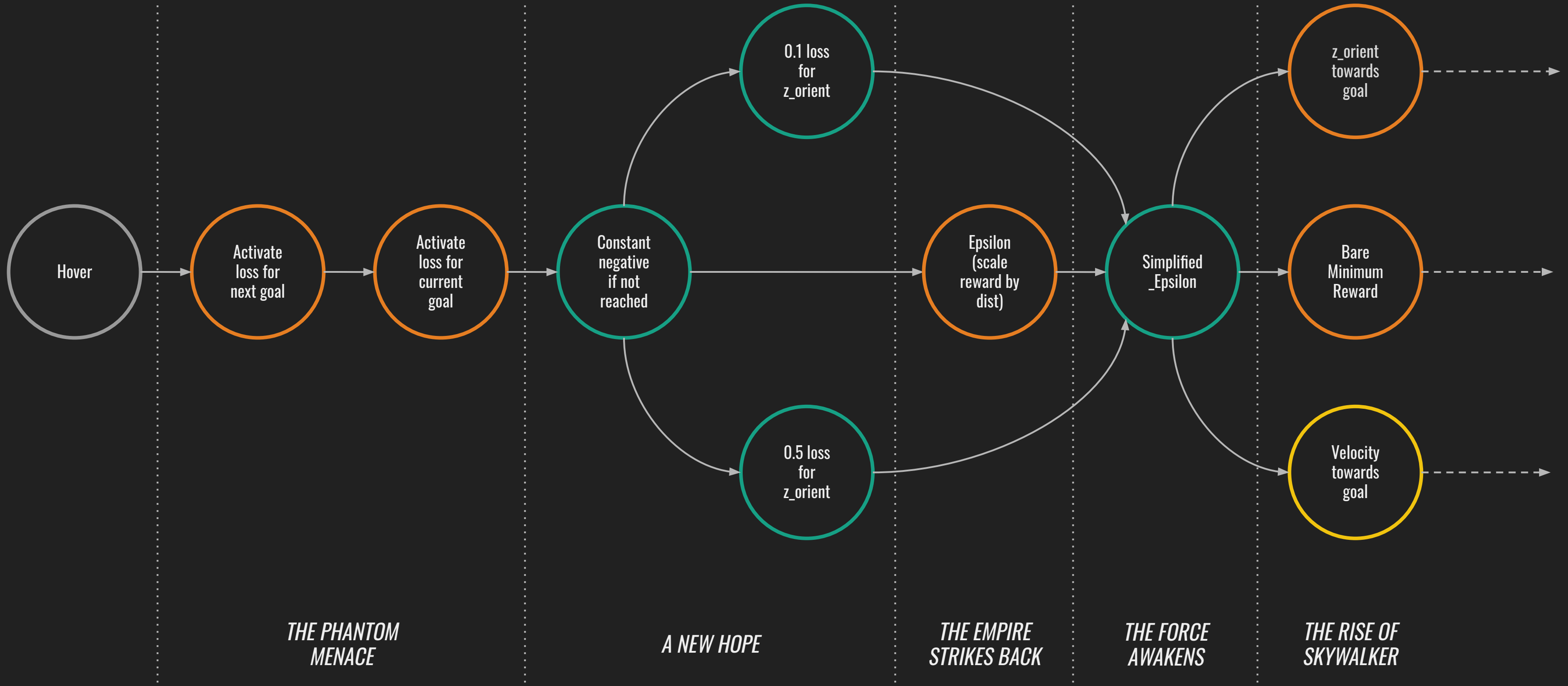


METHODOLOGY

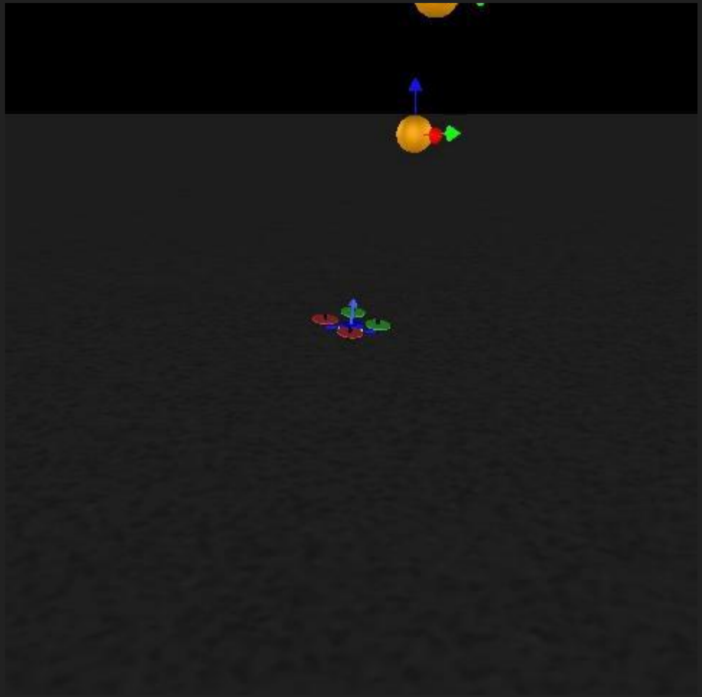
Explore PPO through *reward shaping*.

Try other RL algorithms like *A2C*, *DDPG* etc.

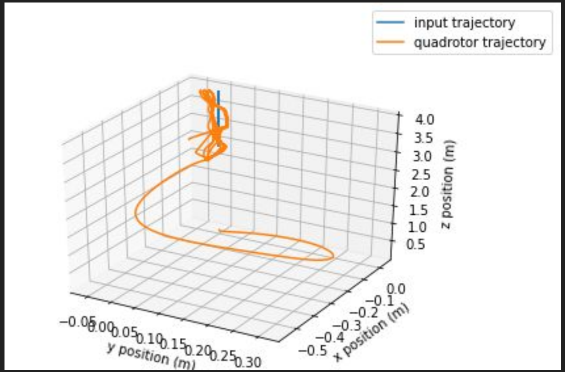
REWARD SHAPING

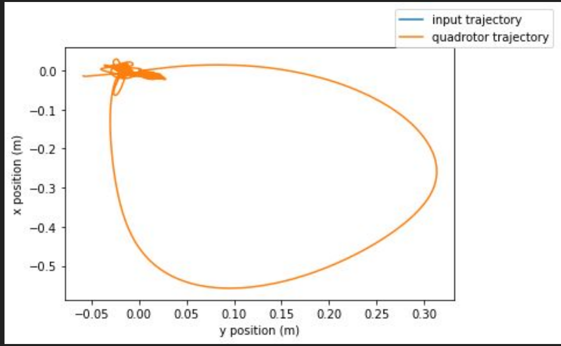


TRAINING



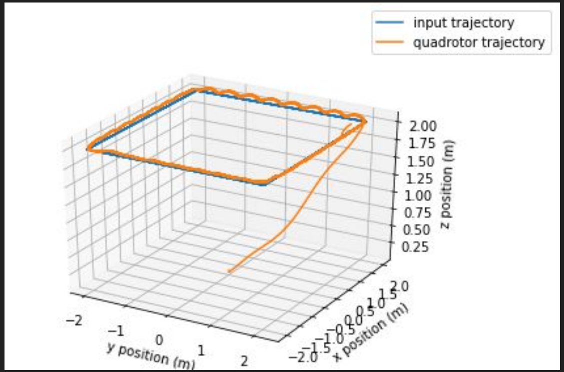
TESTING WITH TRAJECTORIES

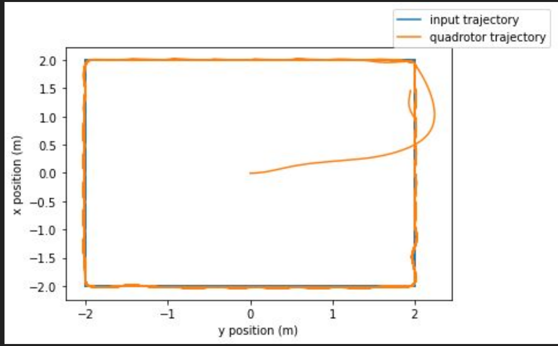




VERTICAL

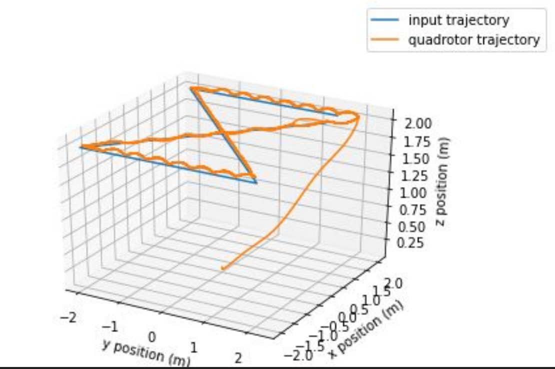
TESTING WITH TRAJECTORIES

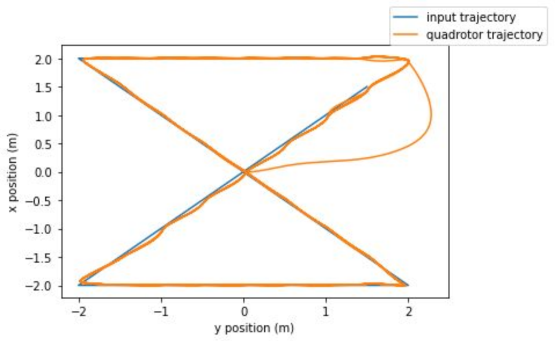




SQUARE

TESTING WITH TRAJECTORIES





SQUARE WITH DIAGONALS

FUTURE WORK

Curriculum learning to learn easier trajectories and move on to harder ones.

There is still scope for further *hyperparameter tuning* to yield better performance.

The quad currently tries to stop at each point, with a *longer horizon* it might perform better.

WHAT WORKED

Quadcopter can go from *one goal to another*.

Generalises well to *different initial positions* and *goal positions* (with constant distance between goals)

Training is now *streamlined* with *docker containers*.

It is still *fun*!

WHAT DIDN'T WORK

Lack of generality (models did not generalise to goals with varying distances).

Going straight to each point might still be better than what our controller can currently achieve.

DDPG model could not successfully hover - likely needs more hyperparameter tuning.

Supervised learning using a PS4 controller - flying the quad manually required a low-level controller.

THAT'S ALL FOLKS