CS 748: Project Update

Learning to Fly

Guide: Prof. Shivaram Kalyanakrishnan

Yash Gadhia (180100130)
Andrews George Varghese(180070005)

Tuning of Bias Parameters

Earlier, the bias parameters of the neural network were not being tuned in policy search and were kept fixed to the random initialisation. This is fixed now and they are also a part of the parameter space of our black box optimisation algorithm (Hill Climbing for now).

Updates to Objective Function

Old Objective Function:

Algorithm Our objective function for hill climbing

---

if $RunTime < 150s$ **then**

$\quad Objective \leftarrow 0$

else if $RunTime$ is $150s$ **then**

$\quad Objective \leftarrow \frac{1}{|DeltaHeading|+1} \times \frac{1}{|DeltaAltitude|+1}$

else

$\quad Objective \leftarrow EpisodeReward$

end if

Updates to Objective Function

Issues in old objective function:

- Minimising delta heading and altitude only at the 1st checkpoint(150 s)
  - What if policy gets stuck at some other checkpoint?
  - Need to include minimisation of delta heading and altitude at any checkpoint
- At checkpoint, the algorithm looks at points where the simulation time is greater than current time & chooses a policy which gives maximum reward out of those
  - But it doesn’t check whether that maximum is greater than the reward of the current policy
  - Even if time is greater, we only want to move there if the “path” is better

Updates to Objective Function

New objective function:

#at_checkpoint refers to whether the current policy is at a checkpoint
#dha refers to delta heading and altitude
If at_checkpoint:
  If ∃ policies with time>current_time and reward>current_time:
    Take the policy which has maximum reward out of those
  Else if ∃ policies stuck at same checkpoint with dha<current_dha:
    Take the policy with minimum delta heading and altitude
Else if not at_checkpoint:
  Maximise Reward

Random Seeds and Radius Annealing

- Ran hill climbing for multiple initial seeds (4 in total)
  - Tested 100,000 random seeds for initialisation
  - Filtered ones which at least reach the first checkpoint (150 s)
  - Choose ones that give high reward/low delta heading & altitude

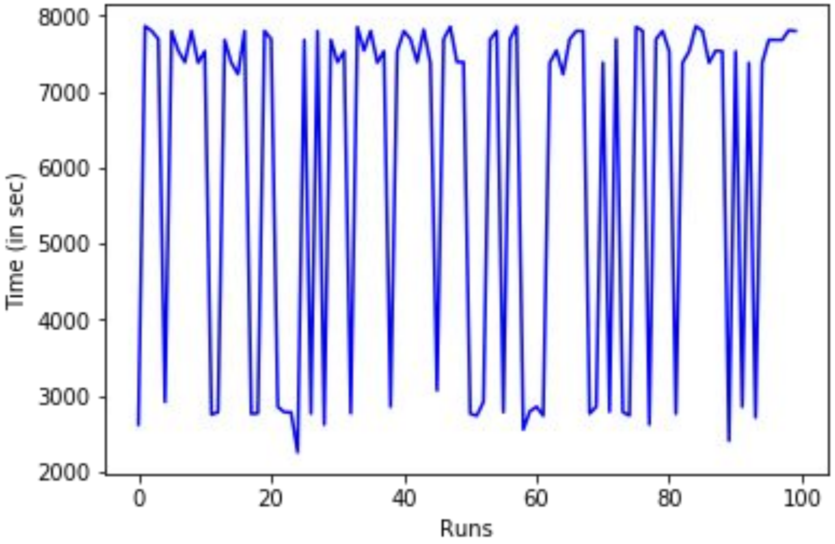
- Annealed radius of hill climbing and ran it with improved policy of the previous run
  - Started with a radius of 1
  - Then 0.5, 0.25 & 0.1

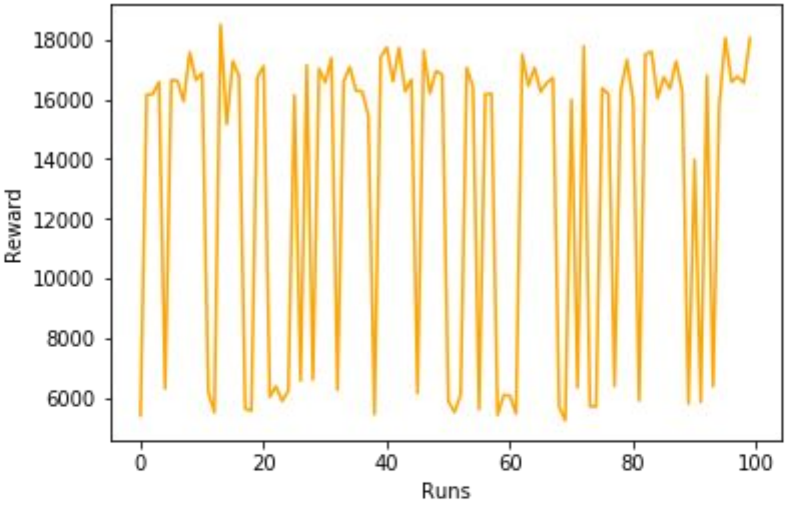
Results

- Obtained a policy that flies the aircraft for 100 mins (40 checkpoints) on average with an average reward of 13144.39 (Average over 100 runs)

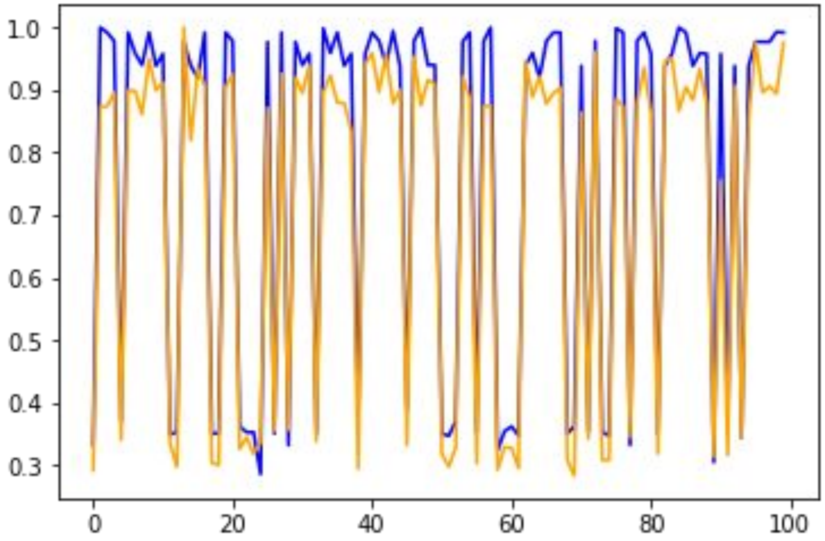
- Previously, the maximum we had achieved in a single run was 10 mins (4 checkpoints) of flying time with a reward of 415.79

Visualisation of multiple test runs



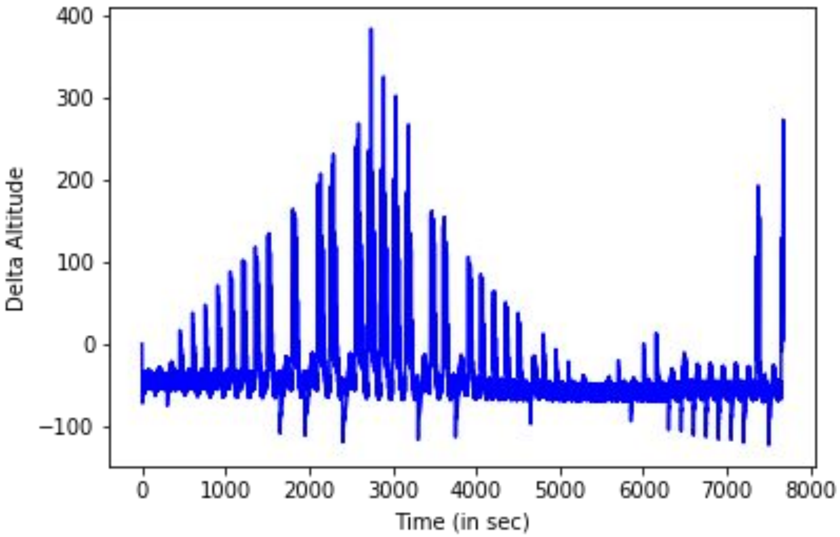


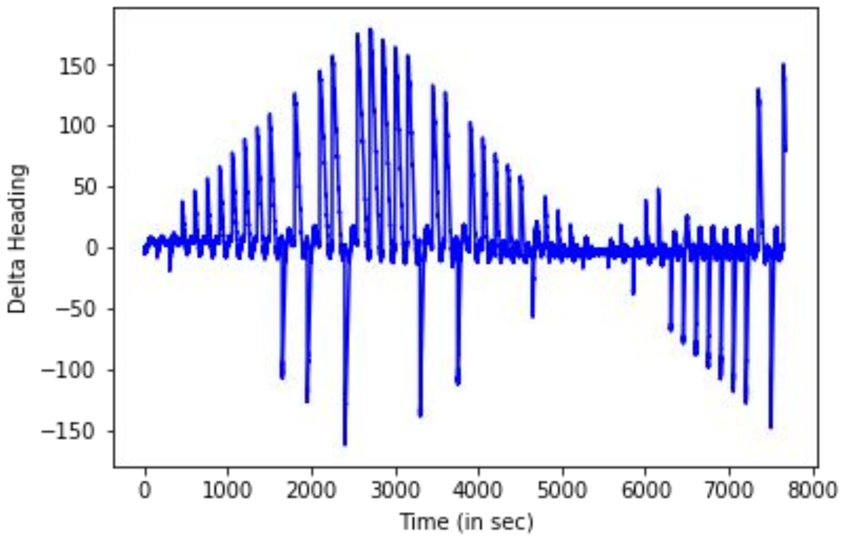
Visualisation of multiple test runs



High rewards are aligned with high run times

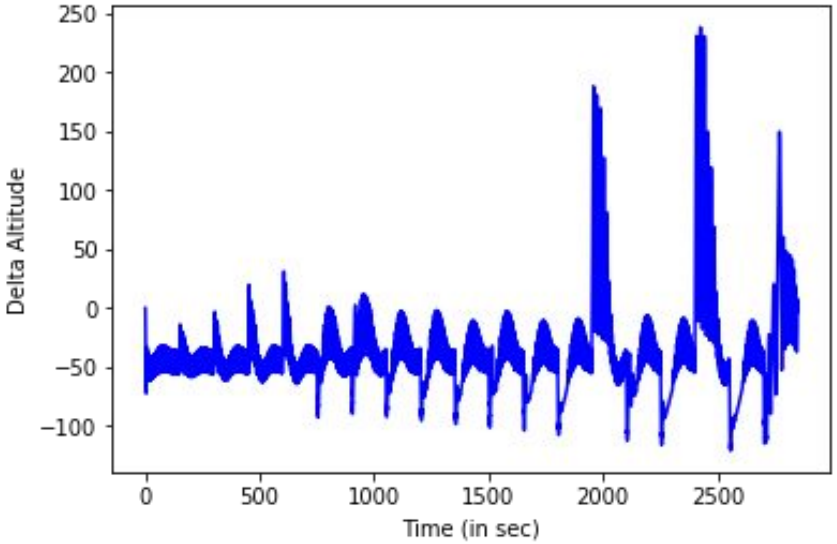
Qualitative Analysis of the Best Run

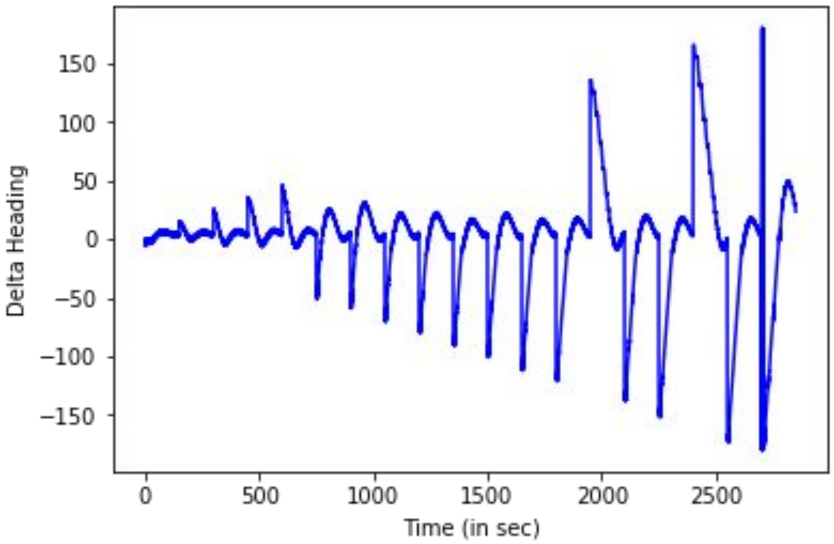




Run Time: 128 mins (51 checkpoints), Reward: 18499.41

Qualitative Analysis of the Worst Run





Run Time: 47.5 mins (19 checkpoints), Reward: 5249.76

Possible Next Steps

- Getting a better estimate of reward for each policy
  - Currently, due to lack of compute and parallelization in code, the policy corresponding to each point in the search space is run only once to get the estimate of its reward
  - Ideally, this should be averaged over multiple runs
  - This is also the reason that the final policy that is returned doesn't actually give the best on-average performance

- CMA-ES/ Other strategies for policy search

Thank You