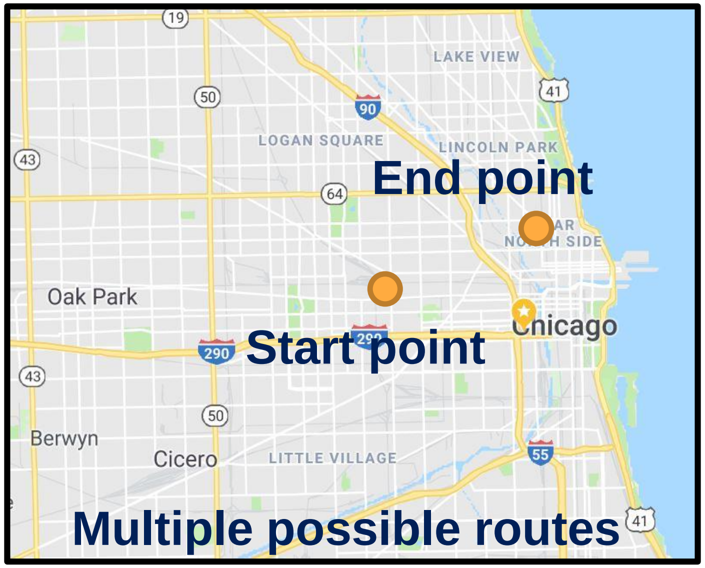
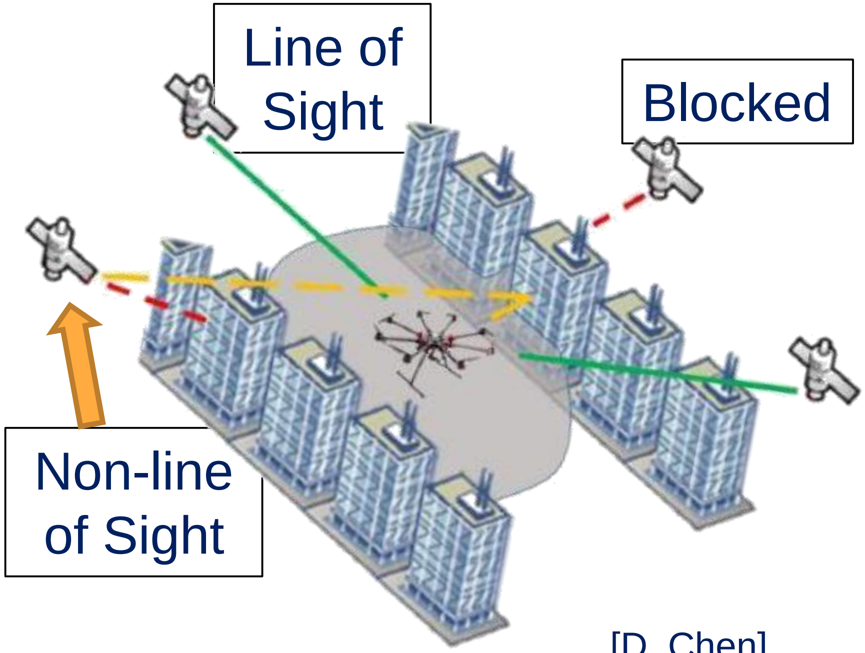
Urban Path Planning to Minimize GPS Integrity Risk Using PPO

Sriramya "Ramya" Bhamidipati

Motivation and Objectives

- Multipath due to tall buildings in urban areas
- Perform autonomous path planning to minimize the GPS integrity risk, induced due to multipath





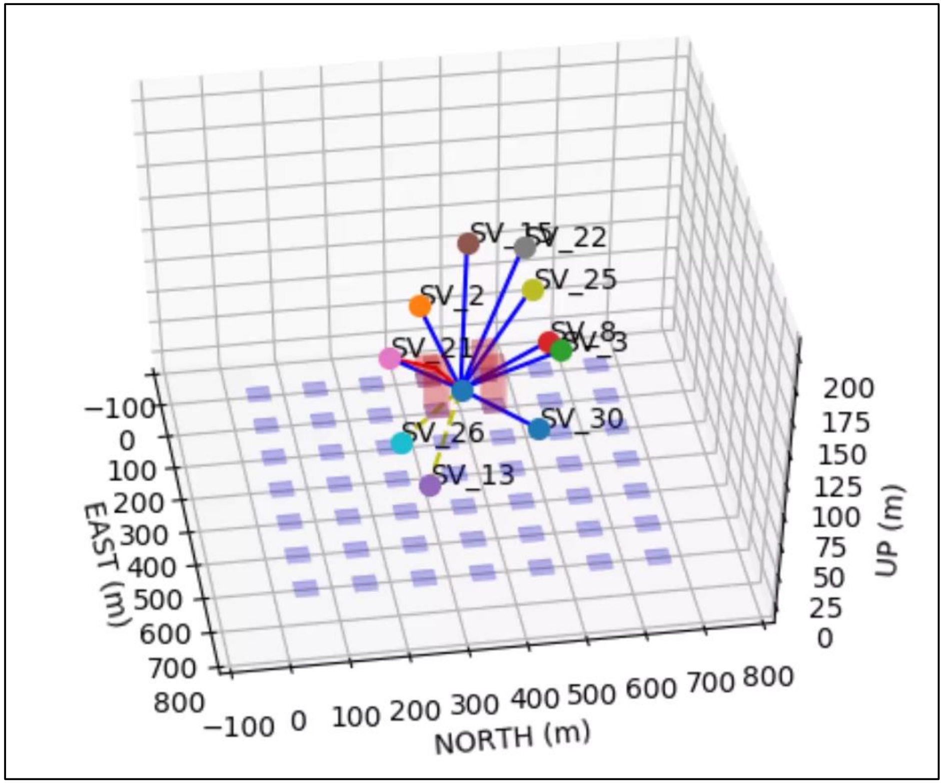
[D. Chen]

Aspects and Assumptions

- Assumptions:
  - GPS satellites are considered static and clock bias errors are ignored
  - Buildings are considered rectangle blocks and uniform distributions with heights ranging from 20-100m
- Given the grid layout of roads in urban areas,
  - Discrete control actions are considered
  - Key decision making points are at intersections
  - Along the rest of the street the same action is continued
- Minimize the GPS integrity risk metric

Simulated Environment

- Multipath effects are induced due to the surrounding 4 buildings
- Building models/LiDAR to obtain the 3D point cloud data
- Constant goal is to reach the coordinates (800, 800, 4)

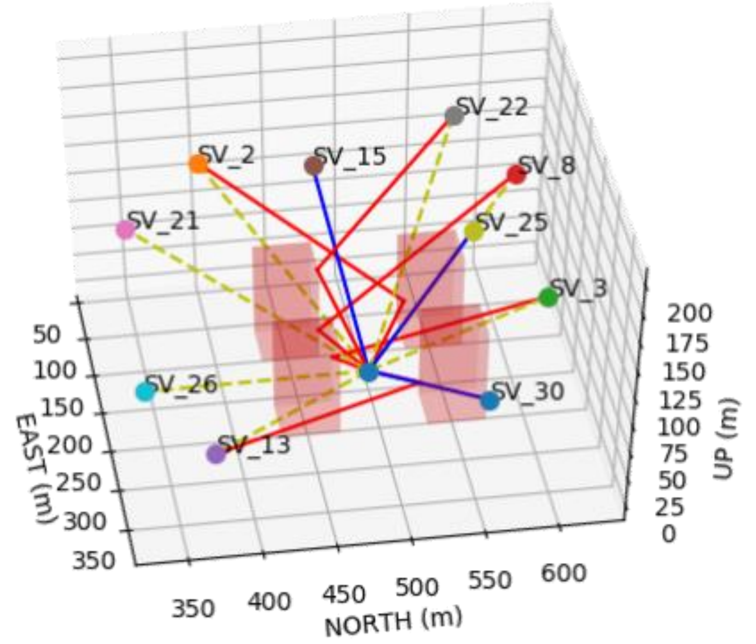


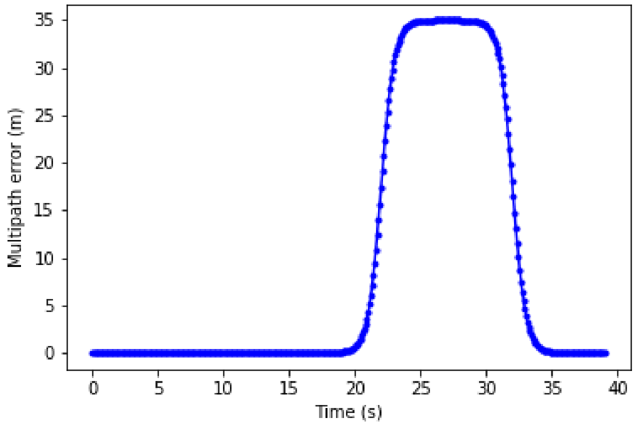
Blue are LOS GPS signals, red are the NLOS signals and dotted yellow are blocked by buildings

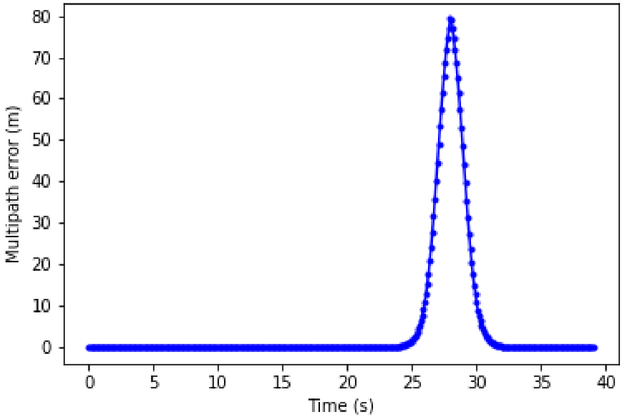
Simulated Environment Cont'd.

Added multipath profiles to NLOS satellites to simulate tracking loop errors

- Tanh profile is considered based on random parameters for start time, rate, amplitude



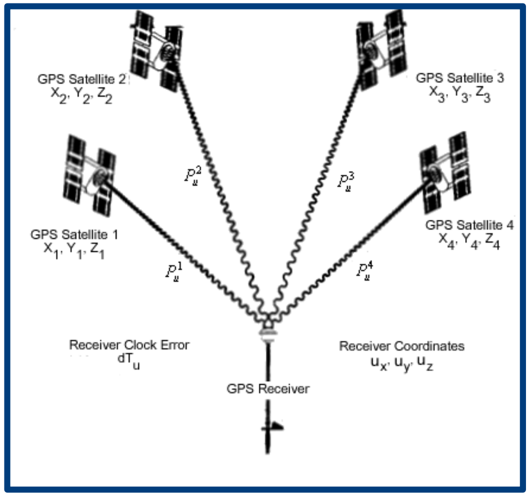




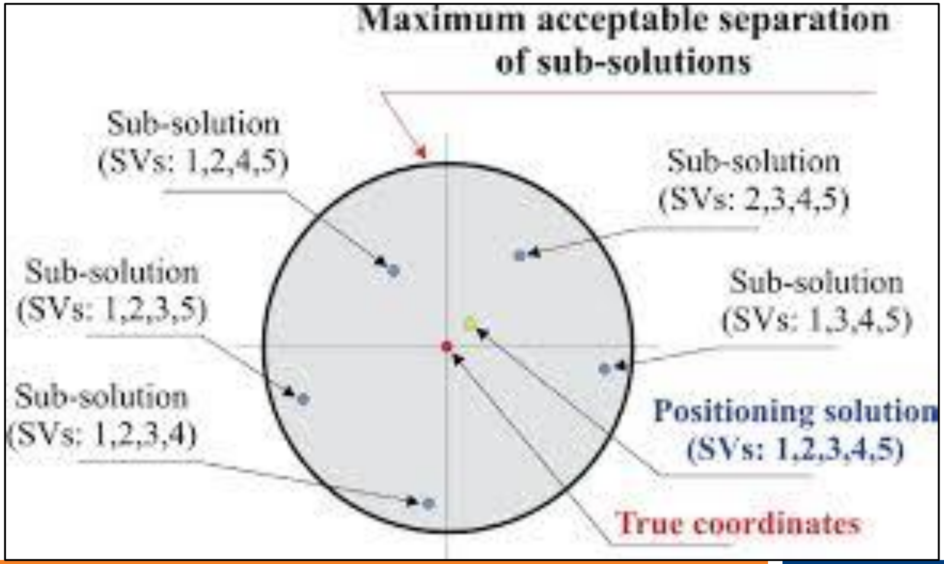
SV 15, 25, 30: Direct Line of sight
SV 2, 3, 8, 13, 22: Non-line of sight
SV 21, 26: Blocked

Reward

- Three components:
  - Integrity risk calculated via GPS RAIM-based solution separation Difference in distances between the previous and the current decision points
  - On reaching the goal



$$r = \begin{cases} r_{gps} = max_{i=1}^{\binom{N}{4}} \Delta_i = \mathrm{x_{all}} - \mathrm{x_i} \\ r_{dist} = c_r(d_{t-1} - d_t) \\ r_{goal} \qquad\qquad d_t < 1e^{-3} \end{cases}$$



Actor-Critic: Discrete PPO

- Separate NN designed for actor and critic
- ReLu activation and categorical distribution for the actor

Algorithm 1 PPO-Clip

Input: initial policy parameters θ_0, initial value function parameters ϕ_0
for $k = 0, 1, 2, ...$ **do**
  Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
  Compute rewards-to-go $\hat{R}_t$.
  Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.
  Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \;\; g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

  typically via stochastic gradient ascent with Adam.
  Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

  typically via some gradient descent algorithm.
end for

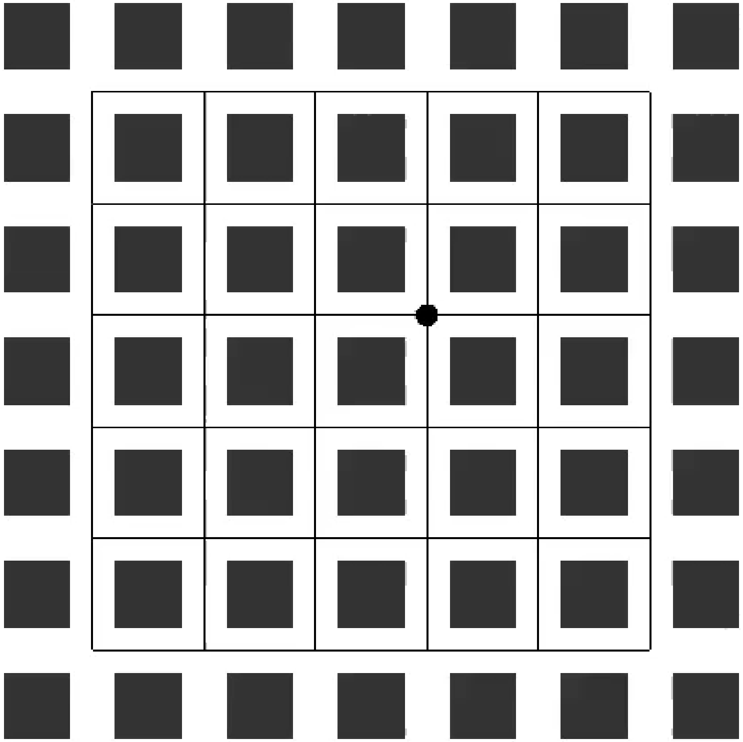
- Actor consists of 3 linear hidden layers and critic consists of 4 layers, with 20 nodes in each layer

Outline

- Motivation and Prior Work
- Algorithm Details
- Results
  - Case #1: Open-sky with 3D robot position as input
  - Case #2: Urban area with 3D robot position as input
  - Case #3: Open-sky with observations from GPS and 3D building models/LiDAR as input
  - Case #4: Urban area with observations from GPS and 3D building models/LiDAR as input

Case #1: Open Sky

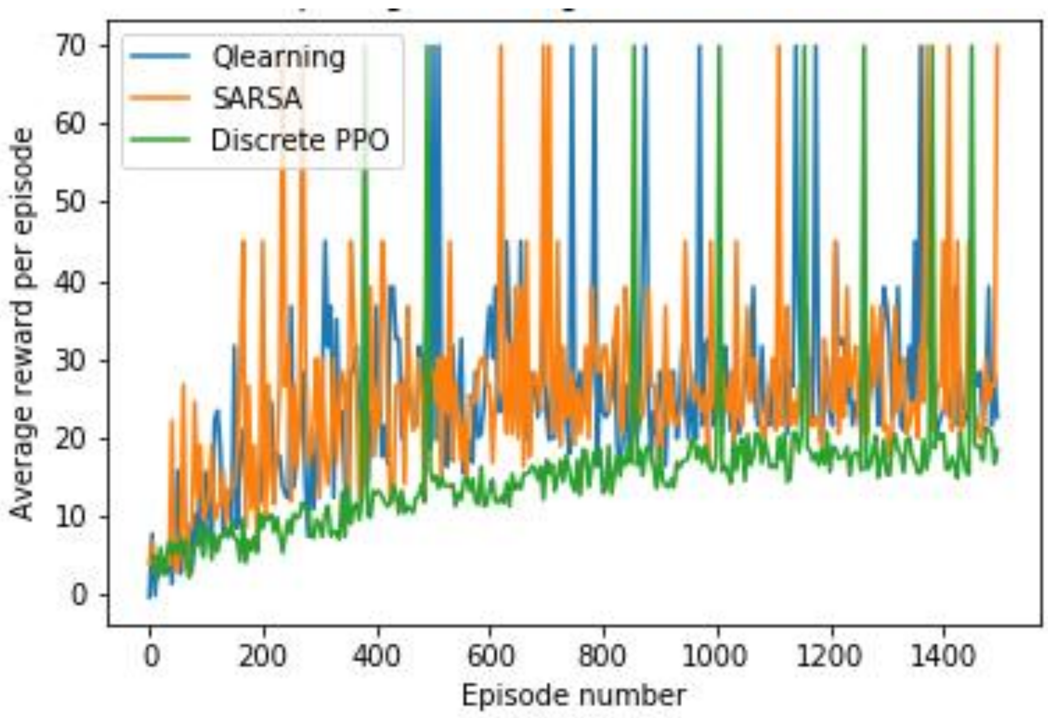
- Algorithms: Discrete PPO
- Input: Current 3D position
- Output: Discrete velocity
- Heights of buildings are all 20m indicating relatively open-sky conditions



Deep RL-based discrete PPO accurately navigates the robot from arbitrary start position to GOAL

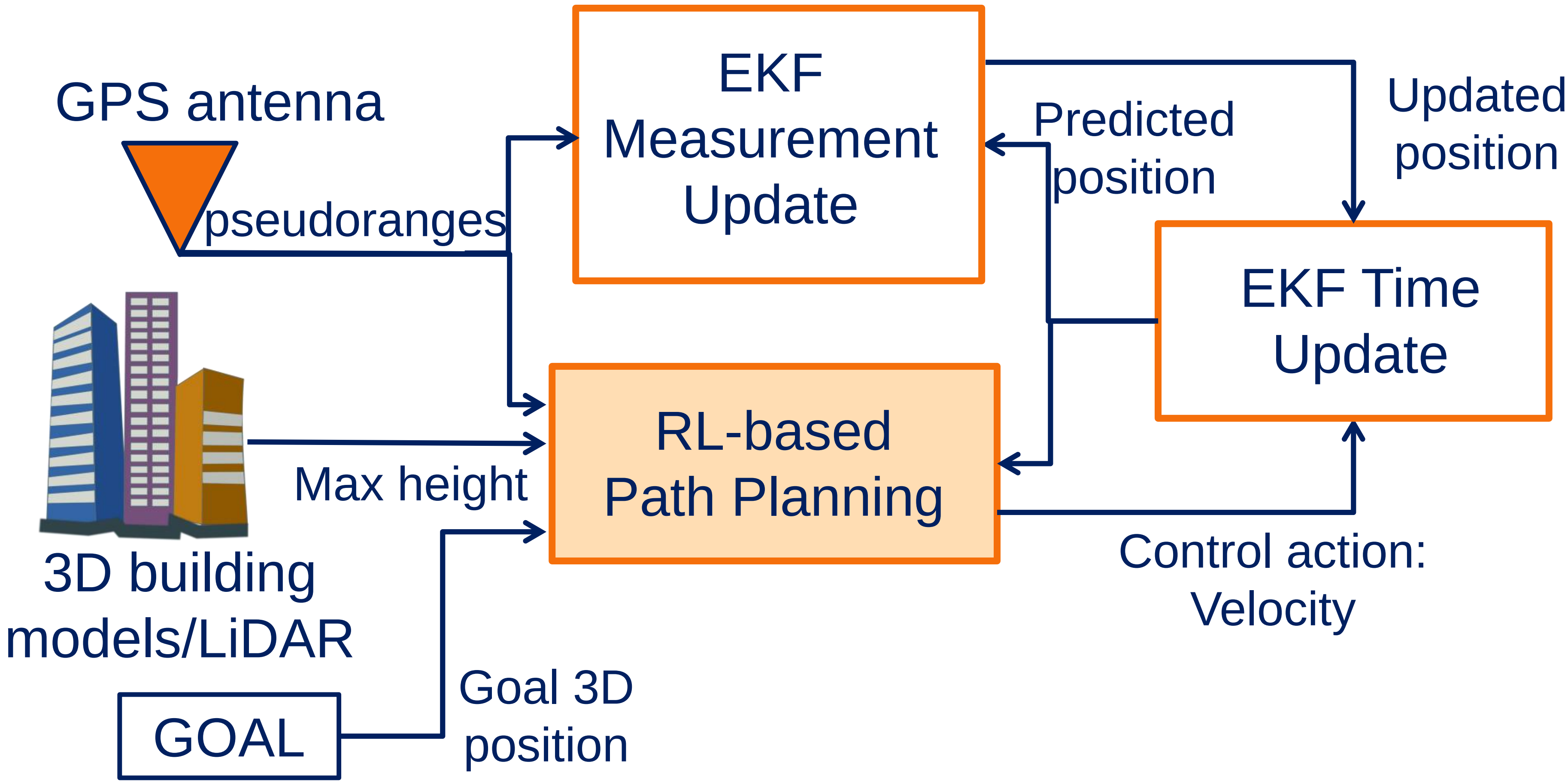
Case #2: Urban Area Results

- Input: Current 3D position
- Output: Discrete velocity
- Episode=100 steps
- Heights range between 35-100m



Deep RL-based discrete PPO also converges, but Q-learning and SARSA perform slightly better

Case #3 & #4: Architecture

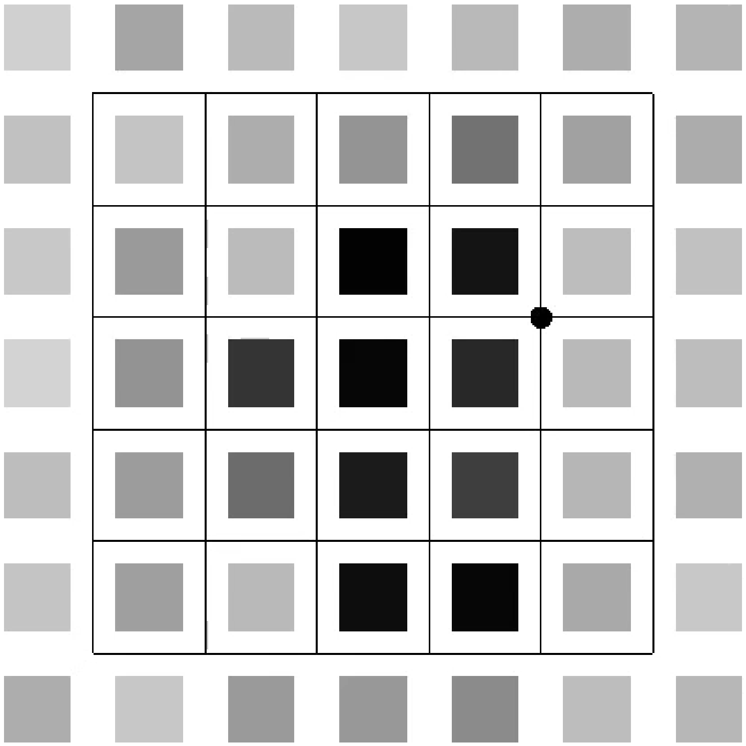


Case #3 & #4: Setup Details

- Deep-RL Algorithm: Discrete PPO
- Multi-core processing, number of workers = 8
- Input to NN: $[\Delta\mathbf{x}, \Delta\rho^1, \cdots, \Delta\rho^N, d^1, \cdots, d^M]$
  - $\Delta\mathbf{x}_\mathrm{t} = \hat{\mathbf{x}}_\mathrm{t} - \mathbf{x}_\mathrm{g}, \Delta\rho^i = \rho^i - h(\hat{\mathbf{x}}_\mathrm{t}, \mathbf{y}_t^\mathrm{i})$, d^j is the height of building available at az^j, i.e., based on max and min ranges, ρ^i is pseudoranges from i^{th} satellite, $\hat{\mathbf{x}}_\mathrm{t}, \mathbf{x}_\mathrm{g}$ denote predicted position by EKF and goal position to be reached, $h(.)$ denotes measurement model
- Output from NN: control velocity $\mathbf{v}_\mathbf{t}$
  - One of four discrete: forward, back, top or bottom

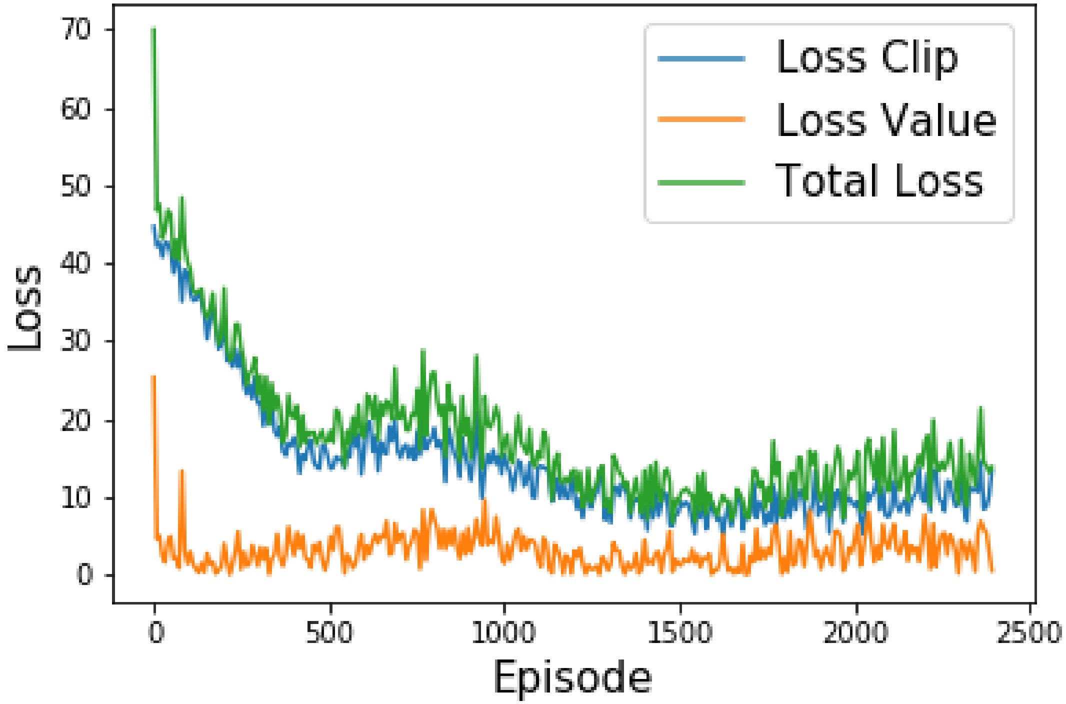
Case #4: Training

- Area: 0.25 sq km area with high-rise buildings
- Light grey and black indicates height of 20m and 100m, respectively
- Training time is ~6 hrs
- Average time to goal=69.3s



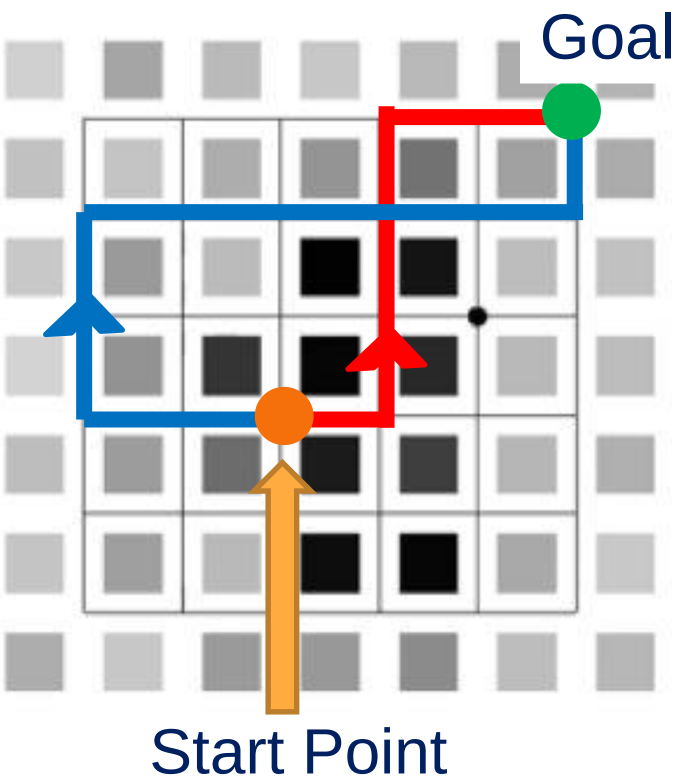
Deep RL-based discrete PPO converges even when **observation vector** is used for training

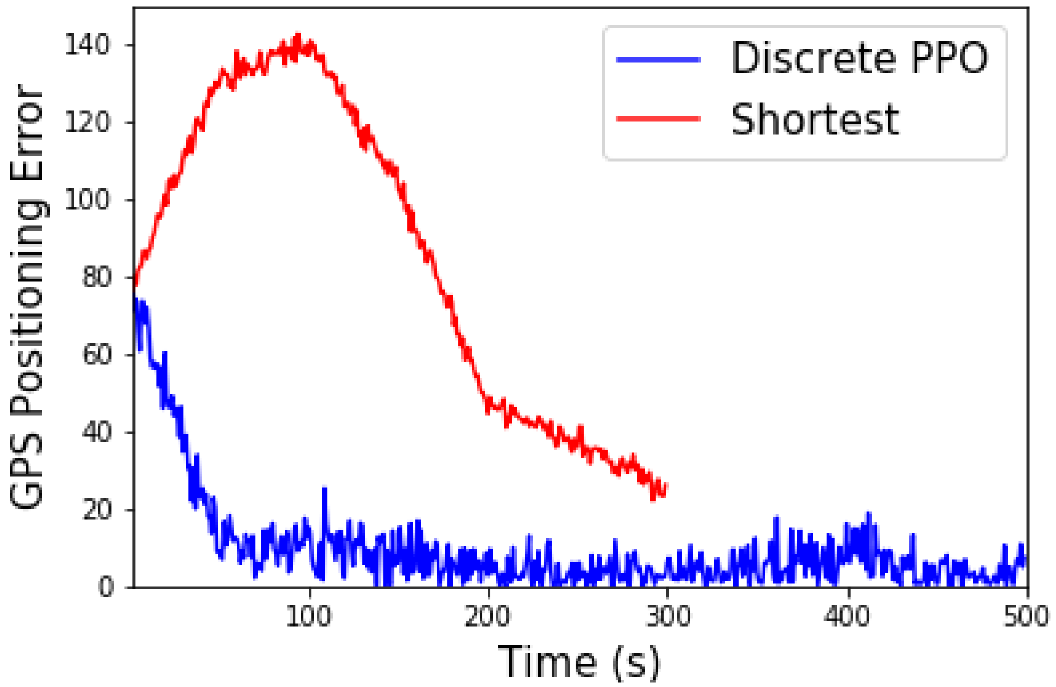
Case #4: Loss Function



Value loss of the critic quickly decreases whereas the clip loss of the actor gradually reduces

Case #4: GPS Position Error





RMS GPS position error of path estimated via PPO is $9.9 \pm 13.79m$, the shortest path is $85.19 \pm 39.93m$

Future/Ongoing Work

- Incorporate this to my research on SLAM-based integrity estimation
- Analyze its performance for a larger urban areas
- Utilize the trained policy in one urban section for navigating in another section
- Satellites motion and the receiver clock bias errors
- Arbitrary goal positions
- Extending the work to generic urban area using
  - Multiple actor and centralized critic to ensure that the robot learns both a local as well as a global policy

Thank you

Sriramya Bhamidipati
Doctoral Student
University of Illinois at Urbana Champaign
Advisor: Prof. Grace Xingxin Gao
Email: sbhamid2@illinos .edu