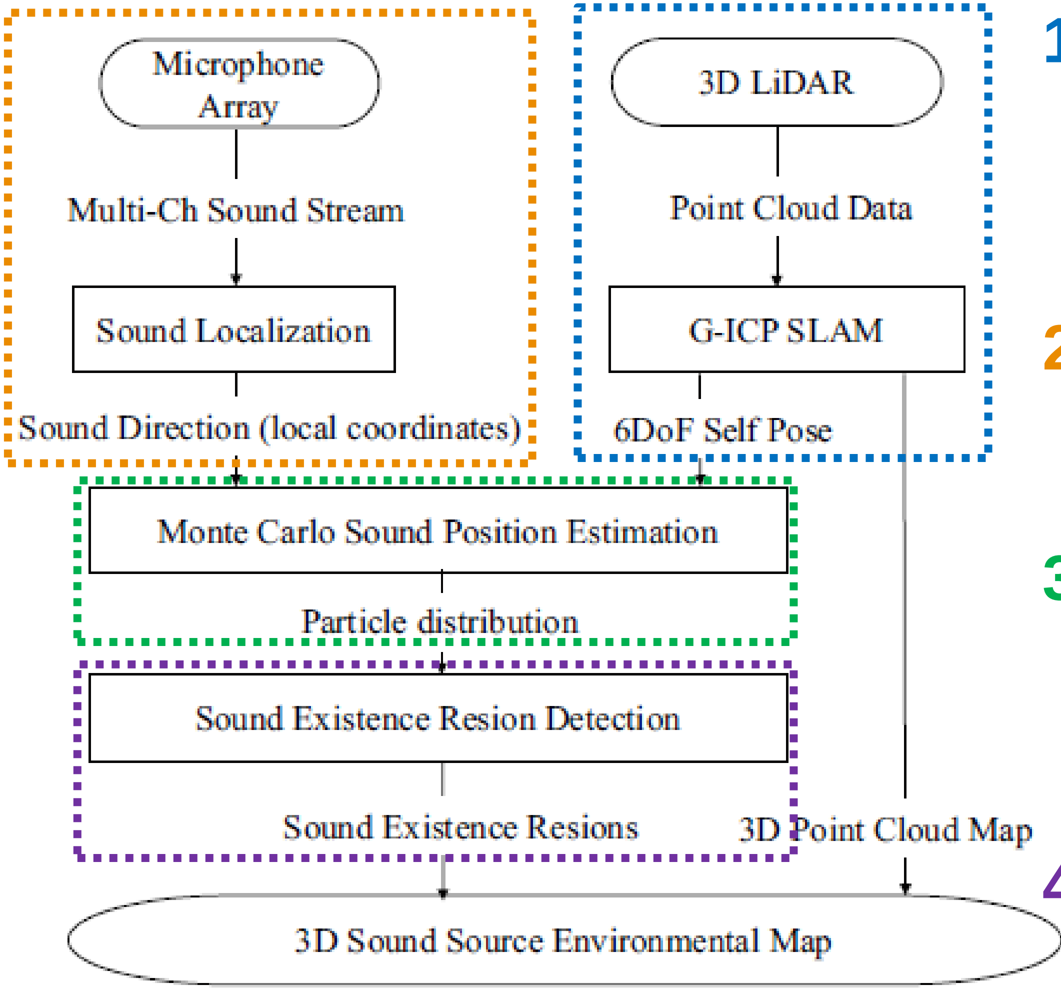
PLATO : Policy Learning using Adaptive Trajectory Optimization

Gregory Kahn et al., ICRA 2017

SeungWoon Kim

KAIST

Probabilistic 3D Sound Source Mapping using Moving Microphone Array / IROS 2016



1. **SLAM**
   → **Find the hardware's location in the 3D map**
2. **Sound Localization**
   → **Detect the directions of sound**
3. **Particle Filter**
   → **Calculate the conversion region of directions**
4. **Sound Source Region Detection**

Contents

- ☐ Motivation
- ☐ Background
- ☐ Main Contribution
- ☐ Results
- ☐ Discussion
- ☐ Summary and Q&A

Motivation (1)

- **Policy search**(via optimization or RL) **is used in many robotic tasks**
  - **Manipulation**
  - **Self-driving vehicles**

https://am.is.tuebingen.mpg.de/uploads/research_project/image/45/unmounting_wheel.jpg

http://iranjavan.net/wp-content/uploads/2016/08/wdd2.jpg

Motivation (2)

- What is **Policy search**?
  - **Strategy for finding optimal control** for robots and autonomous system
  - **Strategy that combines perception and control**
- Two **obstacles when using RL** in the real world
  - RL is difficult to apply to large non-linear function approximators.
  - A partially trained policy can perform unreasonable and even unsafe actions.

→ To select optimal learning method is important!

KAIST

Background

□ Method comparison

- ○ **DAgger method**
  - Selects between teacher and current policy during training with some probability

- ○ **MPC-guided policy search**
  - Seeks to minimize KL-divergence between the teacher and policy distributions.

    * KL divergence is a measure (but not a metric) of the non-symmetric difference between two probability distributions

Main Idea (1)

PLATO

- **Trains neural networks policies using an adaptive MPC**
- **Teacher : adaptive MPC (Model-Predictive Control)**

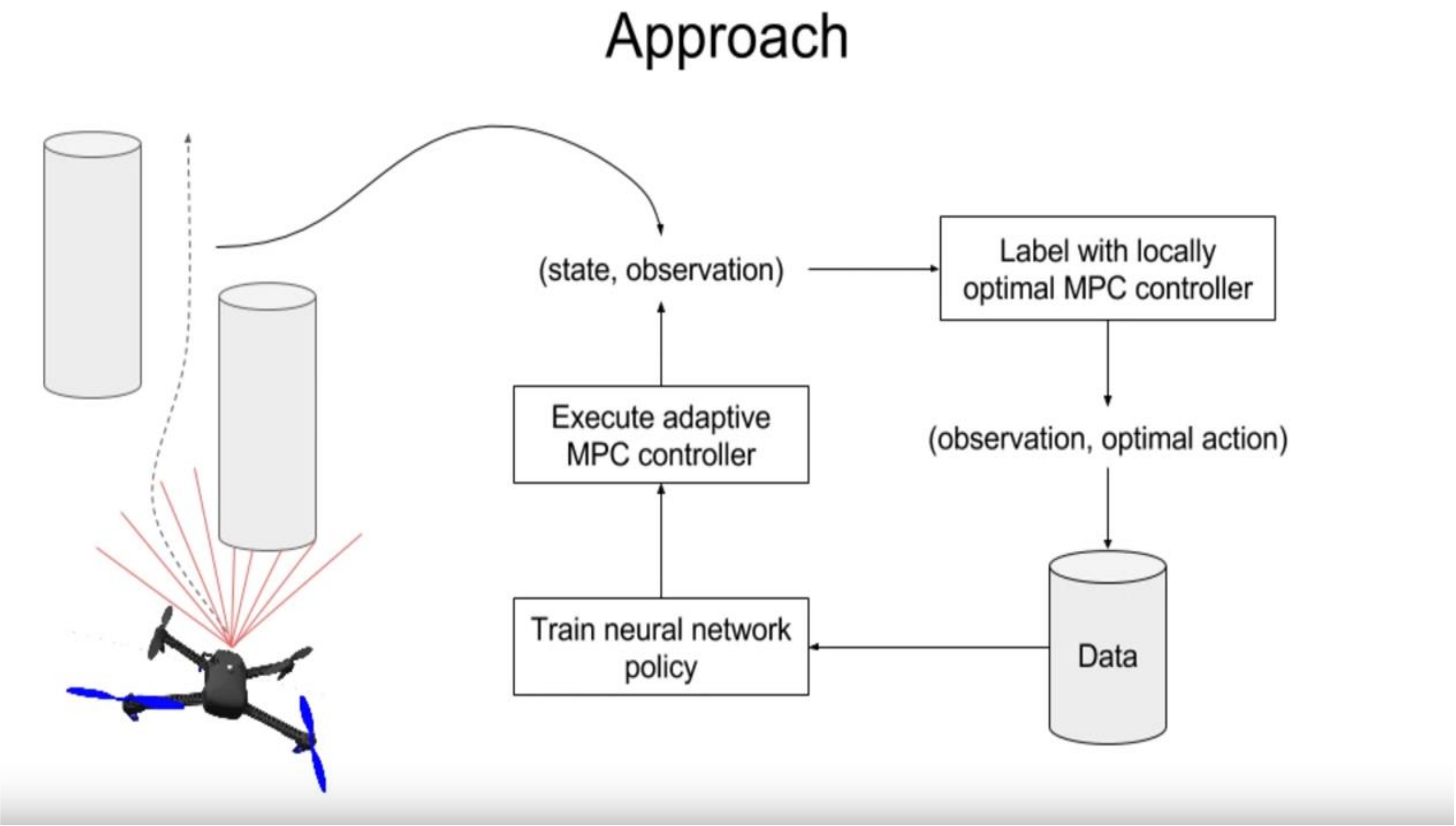
  *** MPC is a traditional optimal control algorithm**
- **Algorithm**

Initialize data $\mathcal{D} \leftarrow \emptyset$
for $i = 1$ **to** N **do**
  for $t = 1$ **to** T **do**
    Optimize π_λ^t with respect to Equation — Optimize with respect to KL-divergence
    Sample $\mathbf{u}_t \sim \pi_\lambda^t(\mathbf{u}|\mathbf{x}_t, \theta)$
    Optimize π^* with respect to Equation — Optimize with respect to teacher
    Sample $\mathbf{u}_t^* \sim \pi^*(\mathbf{u}|\mathbf{x}_t)$
    Append $(\mathbf{o}_t, \mathbf{u}_t^*)$ to the dataset $\mathcal{D}$
    State evolves $\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$
  end for
  Train $\pi_{\theta_{i+1}}$ on $\mathcal{D}$
end for

Main Idea (2)

- The advantages of this approach
  - The teacher can exploit the true state, while the policy is only trained on the observations
  - We can choose a teacher that will remain safe and stable, avoiding dangerous actions during training
  - We can train the final policy using standard and robust supervised learning algorithms

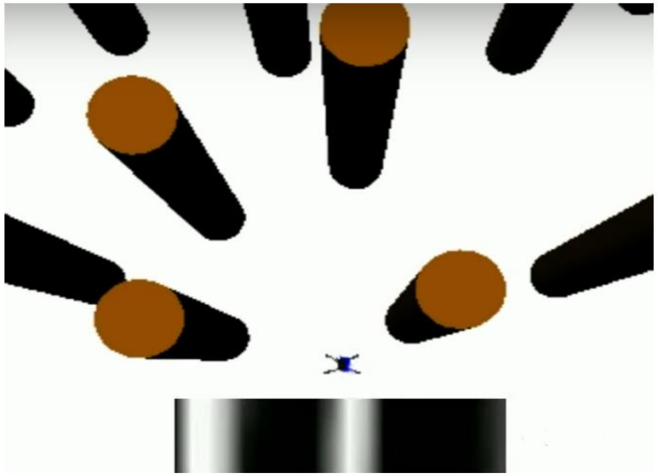
Results (1)


Approach
(state, observation)
Label with locally optimal MPC controller
Execute adaptive MPC controller
(observation, optimal action)
Train neural network policy
Data

Results (2)

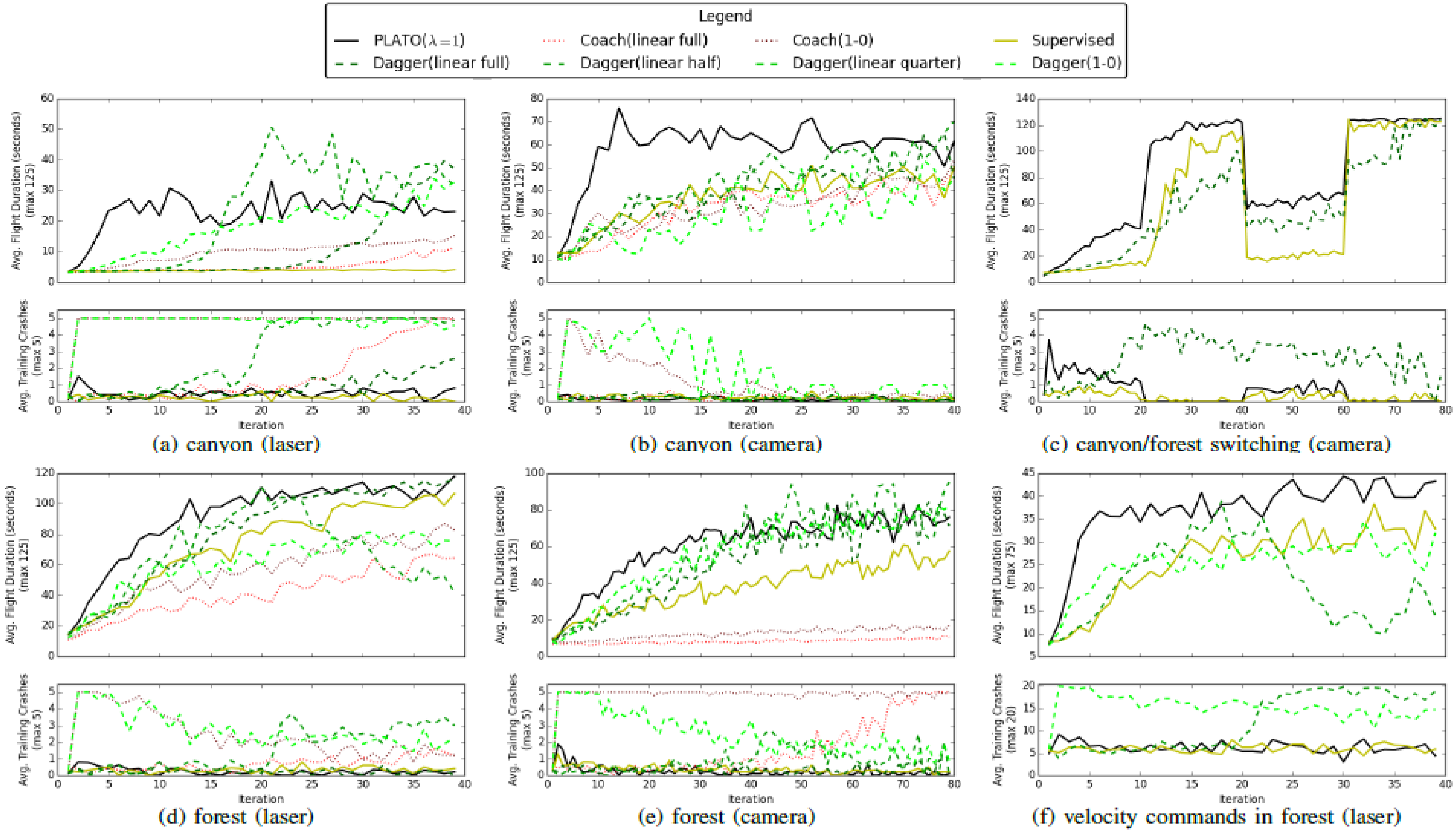
☐ Approach

- Task : A series of simulated quadrotor navigation tasks (with laser, camera)



- Comparison methods
  - DAgger
  - Coaching algorithm
  - MPC-GPS
  - Standard supervised learning
- Environments : winding canyon with randomized turns, dense forest of cylindrical trees
  - Canyon : changes direction up to $\pi/4$ radians every 0.5m
  - Forest : composed of 0.5m radius cylinders with an average spacing of 2.5m

Results (3)



(a) canyon (laser)
(b) canyon (camera)
(c) canyon/forest switching (camera)
(d) forest (laser)
(e) forest (camera)
(f) velocity commands in forest (laser)

Results (4)

- **Evaluation**(centered by PLATO)
  - Can learn effective policies faster, and converges to a solution that is better than other methods.
  - Experiences less than one crash per episode.
  - Successfully learn polices, outperforming prior methods and minimizing the number of crashes.

KAIST

Results (5)

Neural Network Policies
Learned by PLATO

KAIST

Discussion

- **The advantages**
  - **Benefits from the robustness of MPC**
    - * minimizing catastrophic failures at training time
  - **Use a different set of observations than MPC**
    - * the policy can be directly on raw input from onboard sensors, forcing it to perform both perception and control
- **The advantages**
  - **Difficult to apply in most real-world scenarios**
    - * requires full state knowledge to train
- **Outlook**
  - **Possibility of acquiring real-world network policies that directly use rich sensory inputs**
  - **Apply PLATO on real physical platforms**

KAIST

Summary and Q&A

- Any Question?