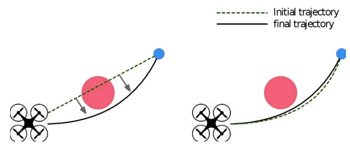
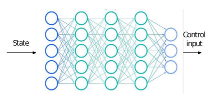
Guided Policy Search using Sequential Convex Programming For Initialization of Trajectory Optimization Algorithms

STUDENTS: Taewan Kim with Purnanand Elango and Behcet Acikmese

Motivation



- Initial trajectory guess is a key input of trajectory optimization algorithms
- It can impact the speed of convergence and reliability of final trajectory
- We generate the initial trajectory guess by training neural net policy



- Once train the policy, we can generate trajectories with different initial points

Problem Formulation

- Optimal control problem. Input "u" comes from neural net.

$$\begin{aligned}\min_{\substack{\theta, x^i(\cdot), u^i(\cdot),\\ i=1,\dots,N}} \quad & \sum_{i=1}^{N}\int_0^{t_f} J(t, x^i(t), u^i(t))\mathrm{d}t \\ \text{s.t.} \quad & \dot{x}^i(t) = f(t, x^i(t), u^i(t)), \\ & x^i(t) \in \mathcal{X}(t), \quad u^i(t) \in \mathcal{U}(t), \\ & s(t, x^i(t), u^i(t)) \le 0, \\ & \qquad \forall t \in [0, t_f], \\ & u^i(t) = \pi_\theta(x^i(t_k)), \quad \forall t \in [t_k, t_{k+1}), \\ & \qquad k = 0, \dots, K-1, \\ & x^i(0) = x^i_{\text{init}}, \quad x^i(t_f) = x^i_{\text{final}},\end{aligned}$$

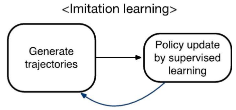
- After optimizing, we can generate a new trajectory by propagating dynamics with the trained policy

$$\begin{aligned} u(t) &= \pi_\theta(x(t_k)) \quad \forall t \in [t_k, t_{k+1}), \\ \dot{x}(t) &= f(t, x(t), u(t)).\end{aligned}$$

How to train neural net policy?: Guided policy search

- **Imitation learning (IL)** by trajectory optimization (TO) algorithms
- Generate multiple trajectories by TO as data, then train neural net policy
- However, sometimes the trajectory data is too hard to learn directly
- **Guided policy search (GPS)** is motivated from the problem of IL

<Imitation learning>

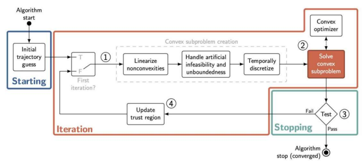


<Guided Policy Search>

Levine, Sergey, and Pieter Abbeel. "Learning neural network policies with guided policy search under unknown dynamics." *Advances in neural information processing systems* 27 (2014).

Background: Sequential Convex Programming (SCP)

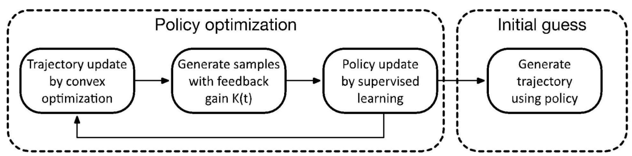
- SCP is trajectory optimization based on convex optimization
- Through linearization and discretization, it approximates the problem to convex sub-problem
- It repeats to this convex optimization problem



[ref] Malyuta, Danylo, et al. "Convex optimization for trajectory generation." *arXiv preprint arXiv:2106.09125* (2021).

Guided Policy Search via SCP

- We develop a new GPS method by employing the idea of SCP
- First, we solve single iteration of SCP
- Second, around the solution of the first step, we generate trajectories with LQR gain
- Third, the trajectory data in second step is used to train the policy
- Repeat the above steps until the convergence of neural net policy



Detail in trajectory update via convex optimization

- Problem formulation

$$\begin{aligned}\min_{\substack{x^i_k, u^i_k, \nu^i_k,\\ i=1,\dots,N,\\ k=0,\dots,K-1}} \quad & \sum_{i=1}^{N}\sum_{k=0}^{K-1} J(t_k, x^i_k, u^i_k) + J_{vc}(\nu) + J_{trp}(u^i_k) \\ \text{s.t.} \quad & x^i_{k+1} = A^i_k x^i_k + B^i_k u^i_k + z^i_k + \nu^i_k, \\ & x^i_k \in \mathcal{X}(t_k), \quad u^i_k \in \mathcal{U}(t_k), \\ & C^i_k x^i_k + D^i_k u^i_k + r^i_k \le 0, \\ & x^i_0 = x^i_{\text{init}}, \quad x^i_{K-1} = x^i_{\text{final}},\end{aligned}$$

- Similar with the sub-problem of SCP
- But, it has an additional penalty J_trp

$$J_{trp}(u) = w_{trp}\|u^i_k - \pi_\theta(\bar{x}^i_k)\|_2^2.$$

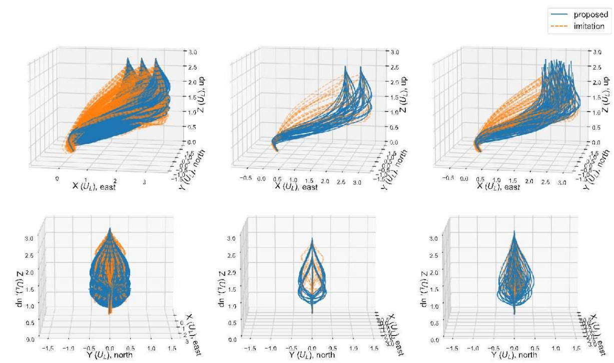
- This penalty enforces the solution is closed to the policy
- So, trajectory update adapts to the neural network

Detail in policy update via supervised learning

- Policy training step
- With data, the policy is trained by supervised learning
- Any neural network optimizer can be used

$$J_p(\theta) = \sum_{s=1}^{S}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\|u^i_{k,s} - \pi_\theta(x^i_{k,s})\|.$$

Application to minimum-fuel powered descent guidance



- 6-dof powered descent guidance problem for a reusable rocket
- Has the following constraints:
- 6-dof rocket dynamics, mass, glide slope angle, angular velocity, tilt angle, thrust, gimbal angle

- Performance comparison

Evaluation of Cost, Constraints and Boundary Condition

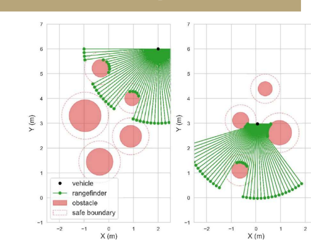
	Validation		Test	
Property (Unit)	Proposed	Imitation	Proposed	Imitation
$-m(t_f)$ (U_M)	-1.8591	-1.8648	-1.8610	-1.8670
$\max_k \|c_k\|_\infty$ (-)	0.0031	0.0615	0.0361	0.2112
$\|r_\mathcal{I}(t_f) - r_{\mathcal{I},f}\|_2$ (U_L)	0.0050	0.0240	0.0081	0.0315
$\|v_\mathcal{I}(t_f) - v_{\mathcal{I},f}\|_2$ (U_L/U_T)	0.0059	0.0567	0.0102	0.0440
$\|\theta_e(t_f) - \theta_{e,f}\|_2$ (°)	0.3074	0.7400	0.5536	0.726
$\|\omega_\mathcal{B}(t_f) - \omega_{\mathcal{B},f}\|_2$ $(°/U_T)$	0.3748	4.9455	0.7976	4.072

Initialization Performance Comparison

Property	Proposed	Straight-line
Convergence success rate	100% (100/100)	93% (93/100)
Mean	2.20	8.55
Median	2	5
Standard deviation	0.40	7.90

Future Work, References, and Acknowledgments

- Future works
- Obstacle avoidance problems
- Final time-free problem
- Policy with partial information of state



- Faculty : Behcet Acikmese
- Students : Taewan Kim, Purnanand Elango

WILLIAM E. BOEING
DEPARTMENT OF AERONAUTICS & ASTRONAUTICS

ADVISERS: ADD THEM HERE

SPONSORS: ADD OR DELETE