Meeting 08/13/2020

Shuo Zhang

Fixed issues of PPO training (Gazebo Hand 0.1% model)

1) Smaller learning rate
2) Sparse reward setting (-1 and 0)
3) Goal location separate training
4) Only use the best policy within 10M steps training for best seed

Datasize for Dynamics Model	A* planning +Rollout	PPO from model + Offline planning + Rollout	Online vanilla PPO	AIP
100%	Done + Done	Done + Done + Done	Not yet	Not yet
0.1%	Done + Done	Done + Done + Not yet	Not yet	Not yet

GPS	AIP (based on ideas so far)
Dynamics Model:	
- time-varying local linear models	- single global nonlinear neural network model
- iteratively trained	- just one model from the very beginning
Features:	
- the controller be updated iteratively	- the controller be never updated
- the controller itself works well	- execution in real env using the controller works not well
- linear gaussian controller (iLQR)	- A* controller deterministic, PPO controller nonlinear. Both are not applicable in the derivation equations in the GPS paper.
- force policy to exactly follow controller with KL=0	- better new policy trained in a model-free fashion
- train policy by using Dual Gradient Descent(updating Lagrange multiplier)	- train policy in traditional actor-critic RL way (for PPO, just SGD, unconstrained optim.)
Usage of KL:	
- KL between old and new controller	- KL between old and new policy
- KL between controller and policy	

GPS

- <u>Policy:</u>

    u_final=pi_theta(x)

    KL(pi_theta||controller)=0

- <u>Controller:</u>

Update

- under newly trained dynamics
- using newly collected data
- considering old controller distribution

KL(controller_new||controller_old)<epsilon

- <u>Objective:</u>

- train an arbitrary parameterized policy pi_theta under the guide of linear gaussian controller policies
- if the controller policies from the dynamics model performs not well in the real env, the parameterized policy will also work not well

AIP (based on ideas so far)

u_final=pi_theta([x, u_controller])

KL(pi_theta_new || pi_theta_old)<epsilon

???KL(pi_theta || controller)<omega??

No Update

- Just one fixed global dynamics

- Train an improved (a better) policy based on the controller policy
- We want to get a policy which works well in the real env, though the controller policy from the dynamics model might work not well

AIP against TRPO

Difference 1:
AIP: u_final=pi_theta([x, u_controller])
TRPO: u_final=pi_theta([x])

Difference 2:
AIP:
KL(pi_theta_new || pi_theta_old)<epsilon
??KL(pi_theta || controller)<omega??
TRPO:
KL(pi_theta_new || pi_theta_old)<epsilon