Gym Torcs Exp

Updated 36 Days Ago All Users

Actions

Last Author huyong
Subscribers *None*

Table of Contents

Targets
LKA
ACC
Related Works
DeepDriving: Simulating ...
LKA in Torcs
PID control to improve st...
Results
Problems
Possible Solutions
Create special tracks with...
Learn single lane cruise w...
Learn changing lanes
ACC in Torcs
Create a straight lane in ...
Learn ACC with DDPG
Reward functions
Network
Results
Stability
Network optimization
Problems
Work Plan
implementation in real enviro...

Targets

This document elaborates the learning process of **motion plannig** in Torcs. We focused on two simple subjects: Lane Keeping Assistance System (LKAS) and Adaptive Cruise Control System (ACC).

LKA

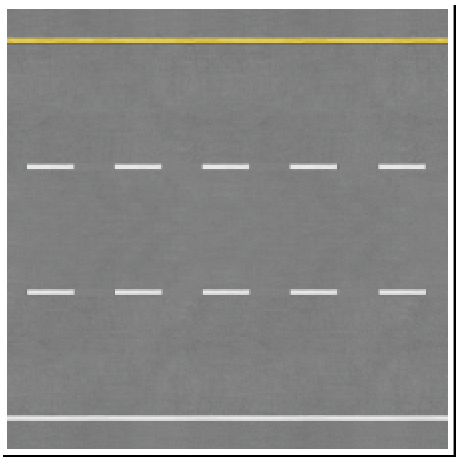
Lane Keeping Assist is a driver assist system that works in conjunction with LDW to alert the driver if the vehicle begins to drift out of a detected lane. LKAS goes a step further by actively applying steering torque to help keep the vehicle within the lane.

ACC

Autonomous cruise control is an optional cruise control system for road vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead.

Related Works

DeepDriving: Simulating Multi-lanes in Torcs



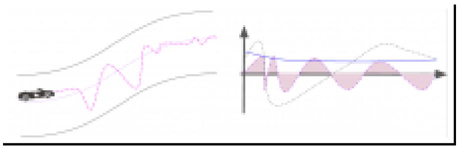
Torcs supports user-defined background for lanes.

DeepDriving use CNN to estimate depth.

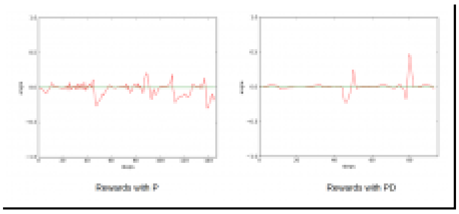
LKA in Torcs

PID control to improve stability in speeding and steering

Add the proportional part of error to the rewards is not enough. Derivative terms are required for better stable runs. More details in Torcs_DDPG_LKA.slice



Results with P and D terms in Rewards:



Results

ucar@10.110.0.214:/home/yonghu/Reports/lka$/home/yonghu/Reports/lka/RL_lka.mkv

Problems

1. **Unstable training** Loss of Critic model doesn't decrease during training

   A. Speed vs keeping
   B. Sensitive Torcs environment
   C. Lack of knowledges about the road ahead

2. **Implementing on real vehicles**

   A. Training on RL trainning history experiments
   B. Experts driving history

Possible Solutions

1. Pre-load the map

Read the XML define map in advanced. Providing the current position in the track, return the curvatures of the current and next track segement and the distance to this change.

2. Pretenship RL learning

To accelerate the training process, rewrite a rule based controller.
This rule based controller takes into account the information of whole map, thus can accelerate and brake more continuously.
But it can not drive exactly in lane especially when turning.

3. Path planning with the TORCS map

Load the TORCS map, and do the path planning with OMPL (or something else).

Create special tracks with multiple-lanes and speed limits

Learn single lane cruise with changing speed limits

To learn driving under speed limit, the reward related to speed is

```
ratio = target_speed / (speed_limit - target_speed)
sp' = min(sp, ratio * (speed_limit -  sp))
```

Learn changing lanes

ACC in Torcs

Create a straight lane in Torcs with inf length.

```
cd /usr/local/share/games/torcs/tracks/oval/
cp ./longstr inflong; cd inflong
# edit length of the straight path, e.g : change 500 to 50000
trackgen -c oval -n inflong
```

Learn ACC with DDPG

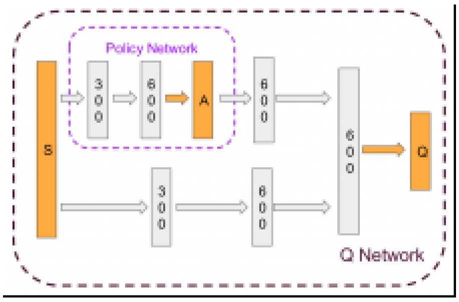
Reward functions

```
tc = dist / speed  #time from collision if front car stops abruptly
if tc <= 0:
  restart
if  tc < 1:
  r = 2.0  - 1 / tc
else:
  r = 1 / tc
r *= speed*angle
```

Network

Input: Speed, AccDist

Output: accel, brake



Results

Learned to follow a robot car with random behaviour.

```
steering = 10*angle/PI - 0.1*trackPos
target_speed = 70 + random(-20, 20)
if speed < target_speed - 50*steering:
  accel  += 0.01
else:
  accel -= 0.01
```

Stability

1. Change reward function, take the steering angle difference into account to prevent it from steering abruptly.

After 1600 continuous episodes, it behaves better.

ucar@10.110.0.214:/home/yonghu/Reports/ddpg_front_70-+10.mkv
ucar@10.110.0.214:/home/yonghu/Reports/ddpg_front_70-+20.mkv

Network optimization

- Halve the NN size.
- Increase learning rate

Problems

- The robot drives not like a human
- The car will run out of the track after a certain distance (a very small angle exists in the startup maybe)

Work Plan

https://docs.google.com/document/d/1_hoTFYbX7YO-Ul_ow_a3KAfIHM8Rar_HkqqeiwKw2zM/edit?usp=sharing

implementation in real environment

Refer to paper, Abbeel, P., Coates, A., Quigley, M., & Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight.
First study the environment model from the human playing history.