自駕車系統專題
Self-driving System Project

Unit 1-1：AI-DO Introduction

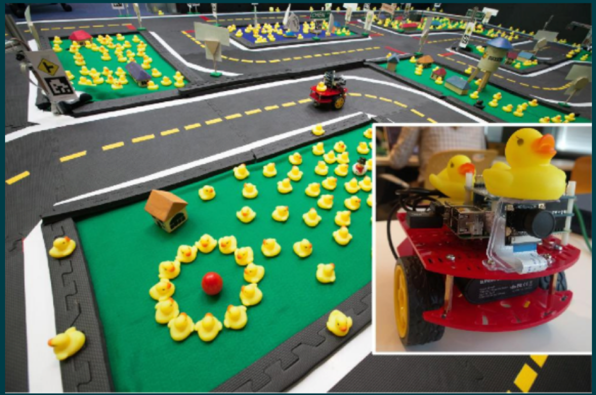
Lecturer：Yuan-Hsiang Lin (林淵翔)

Outline

- Overview
- AI-DO Challenges
- AI-DO Rules
- Getting Started in AI-DO

Overview

- **What are the AI Driving Olympics(AI-DO) ?**
  - **The AI Driving Olympics (AIDO) are a set of robotics challenges designed to exemplify the unique characteristics of data science in the context of autonomous driving**



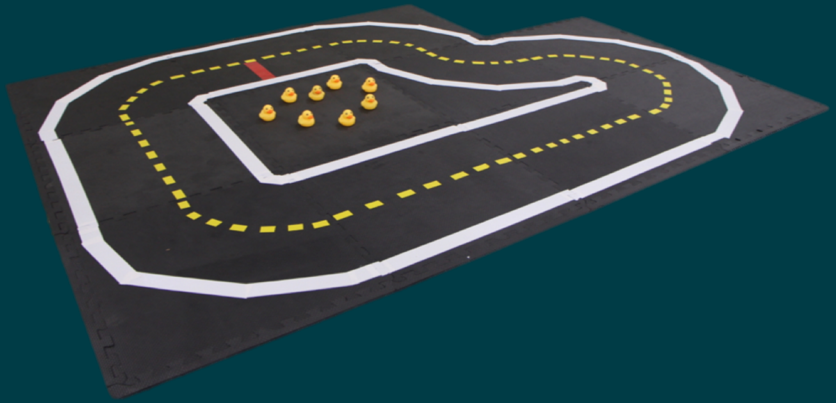
http://docs.duckietown.org/DT19/AIDO/out/aido_overview.html

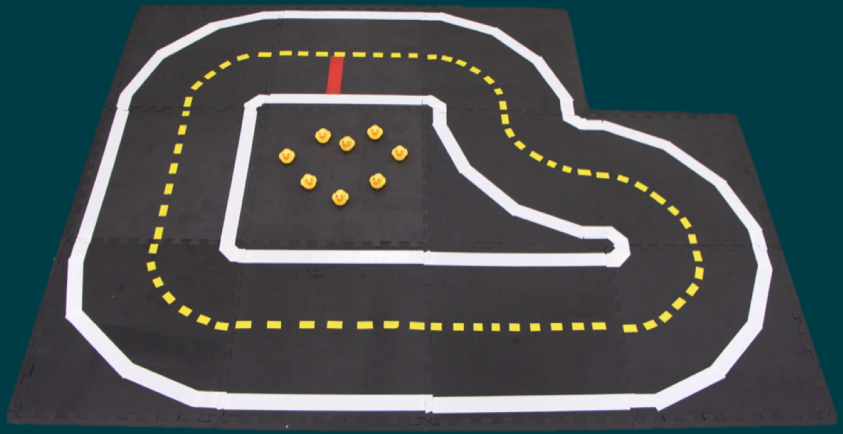
Overview

- The Duckietown platform
  - Simulation and training environment : Allows to test in simulation before trying on the real robots.
  - Remote robotariums : Try the code in controlled and reproducible conditions.
  - Physical Duckietown platform : Miniature vision-based vehicles and cities in which the vehicles drive. The robot hardware and environment are rigorously specified, which makes the development extremely repeatable.

Overview

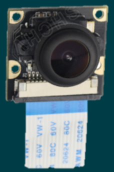
- **The Duckietown**



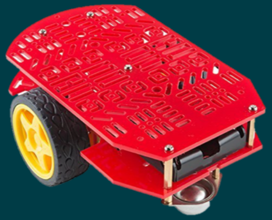


Overview

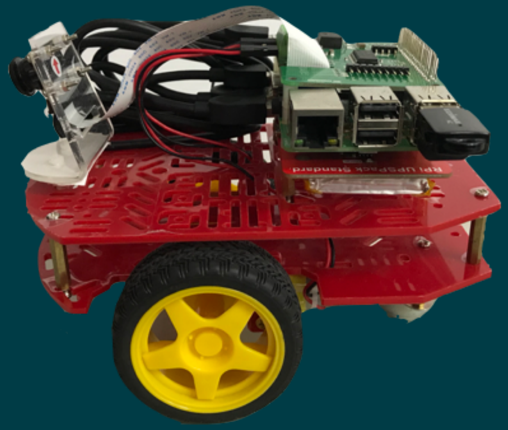
- The Duckiebot

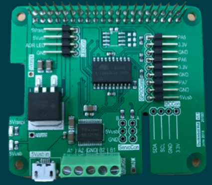


Fisheye Camera

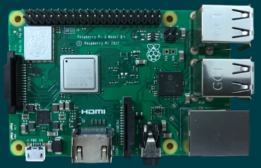


Chassis

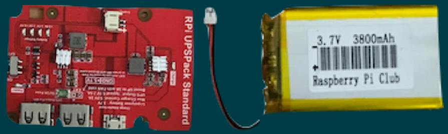




Duckietown Hut



Raspberry Pi 3B+



Battery

Overview

- Course overview

Lab 1
Environment Setup

Theory
- AI-DO Introduction
- Raspberry Pi Introduction
- ROS Introduction

Experiment
- Laptop Setup
- Duckiebot Assembly
- Duckiebot Setup

Lab 2
Motor Control

Theory
- Motor Control

Experiment
- Wheels Calibration

Lab 3
Lane Following

Theory
- Camera Calibration
- Line Detection
- Line Filter

Experiment
- Camera Calibration
- Lane Following

Lab 4
Object Detection

Theory
- Object Detection

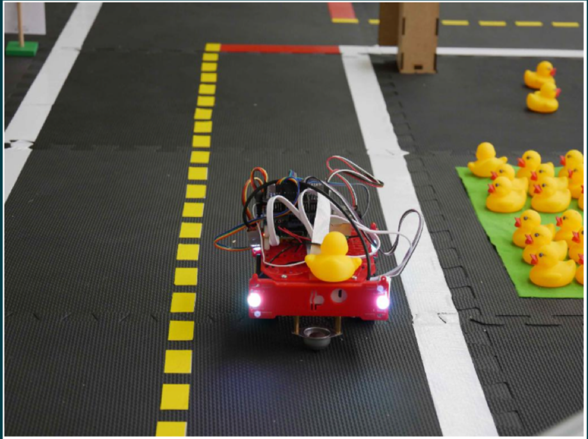
Experiment
- Object Detection

AI-DO Challenges

- **Overview of challenges**
  - **Lane Following (LF)**
  - **Lane Following with Vehicles (LFV)**
  - **Lane Following with Vehicles and Intersections (LFVI)**

AI-DO Challenges

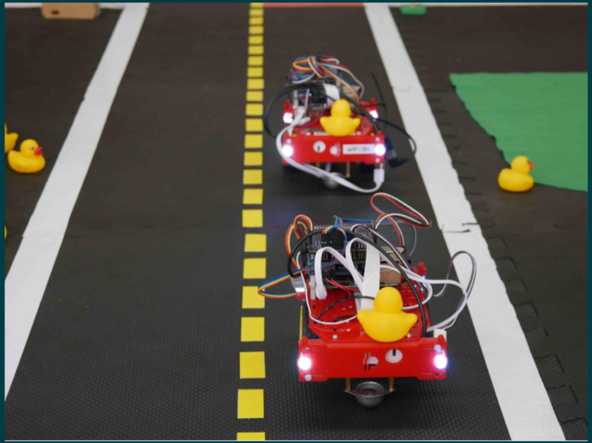
- Lane Following (LF)
  - Control of a Duckiebot to drive on the right lane on streets within Duckietown without other moving Duckiebots present



http://docs.duckietown.org/DT19/AIDO/out/lf.html

AI-DO Challenges

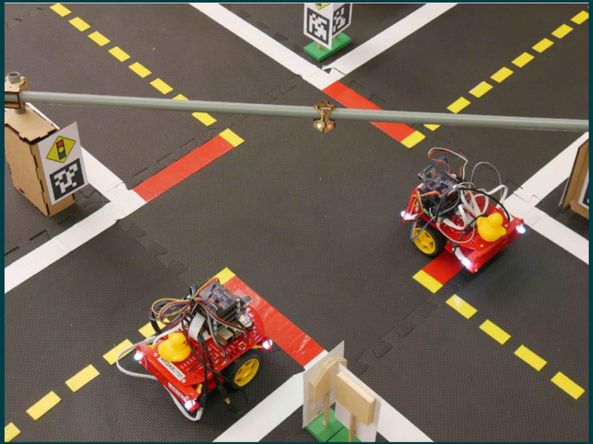
- **Lane Following with Vehicles(LFV)**
  - Control of a Duckiebot to drive on the right lane on streets within Duckietown with other moving Duckiebots and static obstacles present



http://docs.duckietown.org/DT19/AIDO/out/lf_v.html

AI-DO Challenges

- **Lane Following with Vehicles and Intersections(LFVI)**
  - Control of a Duckiebot to drive on the right lane and **through intersections** on streets with Duckietown with other moving Duckiebots and static obstacles



http://docs.duckietown.org/DT19/AIDO/out/lf_v_i.html

AI-DO Rules

- **Performance Objective**
  - **Measures the traveled distance by the integral of speed**
- **Traffic Law Objective**
  - **Follow the traffic laws in Duckietown**
- **Comfort Objective**
  - **To achieve smoother driving in Duckietown**

AI-DO Rules

- **Performance Objective**
  - We choose the integrated speed $v(t)$ along the road (not perpendicular to it) over time of the Duckiebot. This measures the moved distance along the road per episode, where we fix the time length of an episode. This encourages both faster driving as well as algorithms with lower latency. An episode is used to mean running the code from a particular initial configuration

$$\mathcal{J}_{P-LF(V)}(t) = \int_0^t -v(t)dt$$

AI-DO Rules

- **Performance Objective**
  - **The integral of speed is defined over the traveled distance of an episode up to time $t = T\_eps$, where T_eps is the length of an episode**
  - **The way we measure this is in units of "tiles traveled":**

$$\mathcal{J}_{P-LF(V)}(t) = \#\text{ of tiles traveled}$$

AI-DO Rules

- **Traffic Law Objective**
  - **Quantification of "Staying in the lane"**
  - **Quantification of "Keep safety distance"**
  - **Quantification of "Avoiding collisions"**
  - **Hierarchy of rules**

AI-DO Rules

- **Traffic Law Objective : Quantification of "Staying in the lane"**
  - **The Duckietown traffic laws say : The vehicle must stay at all times in the right lane, and ideally near the center of the right lane**
  - **We quantify this as follows: let $d(t)$ be the absolute perpendicular distance of the center of mass the Duckiebot body from the middle of the right lane, such that $d(t)$=0 corresponds to the robot being in the center of the right lane at a given instant. While $d(t)$ stays within an acceptable range no cost is incurred**

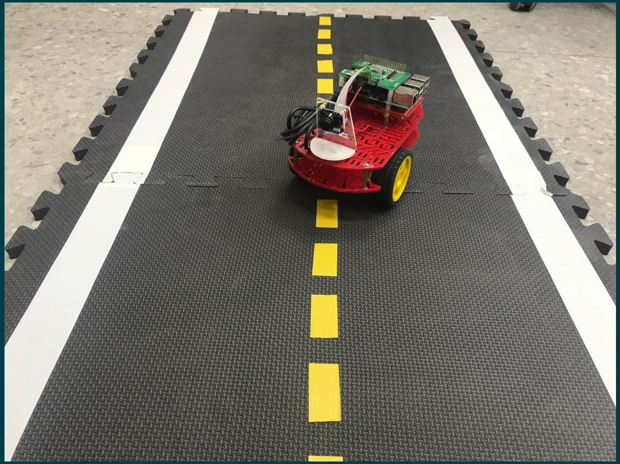
AI-DO Rules

- **Traffic Law Objective : Quantification of "Staying in the lane"**
  - **When the safety margin d_{safe} is violated, cost starts accumulating proportionally to the square of $d(t)$ up to an upper bound d_{max}. If this bound is violated a lump penalty α is incurred**
  - **The "stay-in-lane" cost function is therefore defined as :**

$$\mathcal{J}_{T-LF}(t) = \int_0^{T_{eps}} \begin{cases} 0 & d(t) < d_{safe} \\ \beta d(t)^2 & d_{safe} \leq d(t) \leq d_{max} \\ \alpha & d(t) > d_{max} \end{cases}$$

AI-DO Rules

- **Traffic Law Objective : Quantification of "Staying in the lane"**
  - **An example situation where a Duckiebot does not stay in the lane is shown in figure**



AI-DO Rules

- **Traffic Law Objective : Quantification of "Keep safety distance"**
  - **The Duckietown traffic laws say : Each Duckiebot should stay at an adequate distance from the Duckiebot in front of it, on the same lane, at all times**

AI-DO Rules

- **Traffic Law Objective : Quantification of "Keep safety distance"**
  - **We quantify this rule as follows: Let $b(t)$ denote the distance between the center of mass of the Duckiebot and the center of mass of the closest Duckiebot in front of it which is also in the same lane. Furthermore let b_safe denote a cut-off distance after which a Duckiebot is deemed "far away". Let δ denote a scalar positive weighting factor**

$$\mathcal{J}_{\mathrm{T}-SD}(t) = \int_0^t \delta \cdot \max(0, b(t) - b_{safe})^2$$

AI-DO Rules

- **Traffic Law Objective : Quantification of “Avoid collisions”**
  - **The Duckietown traffic laws say: At any time a Duckiebot shall not collide with a duckie, Duckiebot or object**

AI-DO Rules

- Traffic Law Objective : Quantification of "Avoid collisions"
  - The vehicle is penalized by ν if within a time interval of length t_k $t \in [t, t+t\_k)$ the distance $\ell(t)$ between the vehicle and a nearby duckie, object or other vehicle is zero or near zero. $\ell(t)$ denotes the perpendicular distance between any object and the Duckiebot rectangular surface. The collision cost objective therefore is

$$\mathcal{J}_{T-AC}(t) = \sum_{t_k} v \mathbb{I}_{\exists t \in [t-t_k, t) \ell(t) < \epsilon}$$

AI-DO Rules

- **Traffic Law Objective : Quantification of "Avoid collisions"**
  - **where ν is the penalty constant of the collision**
  - **Time intervals are chosen to allow for maneuvering after collisions without incurring further costs**
  - **An illustration of a collision is displayed in figure**

AI-DO Rules

- **Traffic Law Objective : Hierarchy of Rules**
  - **To account for the relative importance of rules, the factors $\alpha,\beta,\gamma,\delta,\nu$ of the introduced rules will be weighted relatively to each other**
  - **Letting > here denote "more important than", we define the following rule hierarchy :**

$$\mathcal{J}_{T-AC} > \mathcal{J}_{T-SI} > \mathcal{J}_{T-SD} > \mathcal{J}_{T-LF}$$

I.e.:
Collision avoidance > Stop line > Safety distance > Staying in the lane

AI-DO Rules

- **Comfort Objective**
  - **In the single robot setting, we encourage "comfortable" driving solutions. We therefore penalize large angular deviations from the forward lane direction to achieve smoother driving. This is quantified through changes in Duckiebot angular orientation $\theta\_(bot(t))$ with respect to the lane driving direction**

AI-DO Rules

- **Comfort Objective**
  - **Good angle metric : As a comfort objective, we measure the average absolute squared changes in angular orientation of $\theta_{bot(t)}$ over time**

$$\mathcal{J}_{c-LF/LFV}(t) = \frac{1}{t}\int_0^t \left|\theta_{bot(t)}\right|^2 dt$$

AI-DO Rules

- **Comfort Objective**
  - **Valid direction metric : As an additional pointer we calculate the fraction of times the Duckiebot has a "good" angular heading or valid direction. Where θ_{good} corresponds to an angle of 20 degrees (converted to radians)**

$$J_{VD-\frac{LF}{LFV}}(t) = \frac{1}{t}\int_0^t \mathbb{I}_{|\theta_{bot(t)}|<\theta_{good}}\, dt$$

Getting Started in AI-DO

- Get the needed accounts
  - Docker hub account. Create an account here
  - Duckietown account. Create an account here
- Software requirements
  - Python 3.6 or higher
  - Ubuntu 18.04
  - Docker
  - Git
  - Duckietown Shell

Getting Started in AI-DO

- Make a submission :
  - Check out the competition template challenge-prediction :

    - $ git clone https://github.com/duckietown/challenge-prediction

  - Submit

    - $ cd challenge-prediction/predictor_last
    - $ dts challenges submit

  - Monitor the submission

    - $ dts challenges follow --submission submission ID

  - Look at the Leaderboard here

References

- AI-DO Challenges
  - http://docs.duckietown.org/DT19/AIDO/out/lf.html
- AI-DO Rules
  - http://docs.duckietown.org/DT19/AIDO/out/measuring_performance.html
- Getting Started in AI-DO
  - http://docs.duckietown.org/DT19/AIDO/out/quickstart.html

Thank You