Reinforcement Learning-Based End-to-End Parking for Automatic Parking System

CS885 – Reinforcement Learning

Paper by: P. Zhang, L. Xiong, Z. Yu, P. Fang, S. Yan, J. Yao, and Y. Zhou
(Sensors 2019)

Presented by: Neel Bhatt

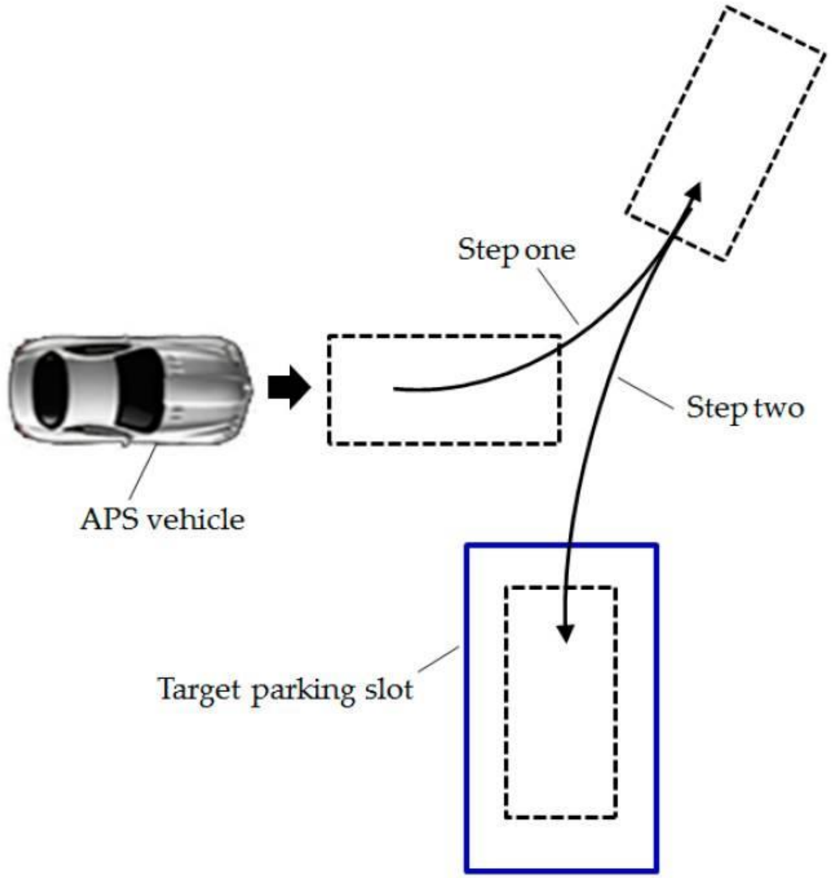
UNIVERSITY OF WATERLOO

Context and Motivation

- High density urban parking facilities can benefit from an **automated parking system** (APS):
  - Increase parking safety
  - Enhance utilization rate and convenience

- BS ISO 16787-2016 stipulates parking inclination angle to be confined within ±3°

- <u>This paper</u> focuses on a **DDPG** based end-to-end automated parking algorithm



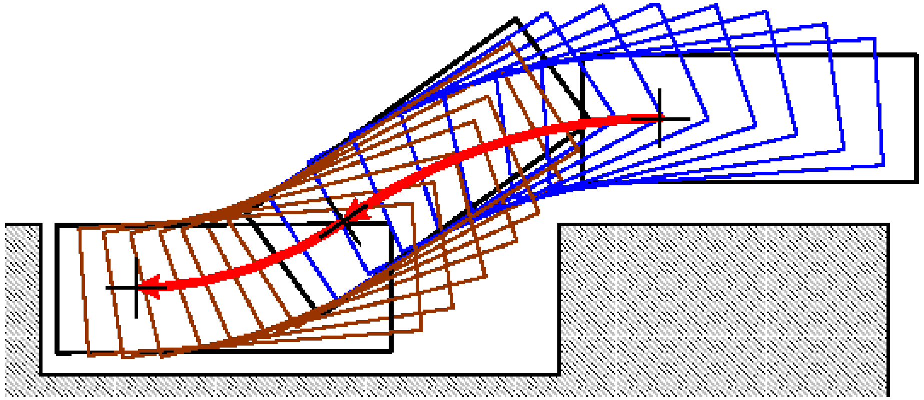
Related Work

Path Planning

- Consists of predefined trajectory functions: B-splines, η^3-splines, Reeds-Shepp curves
- Involves geometric numerical optimization of the curve parameters subject to vehicle non-holonomic constraints

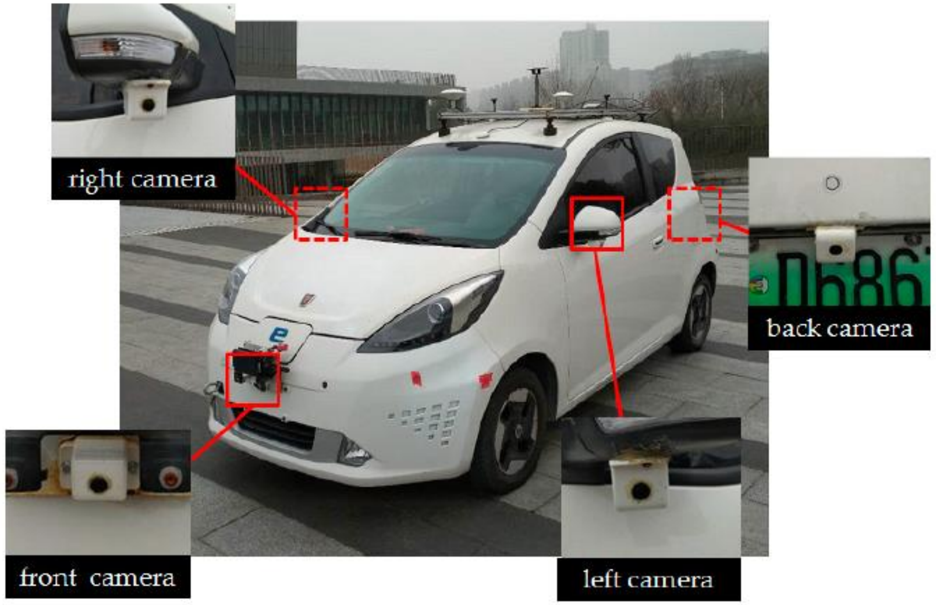
Path Tracking

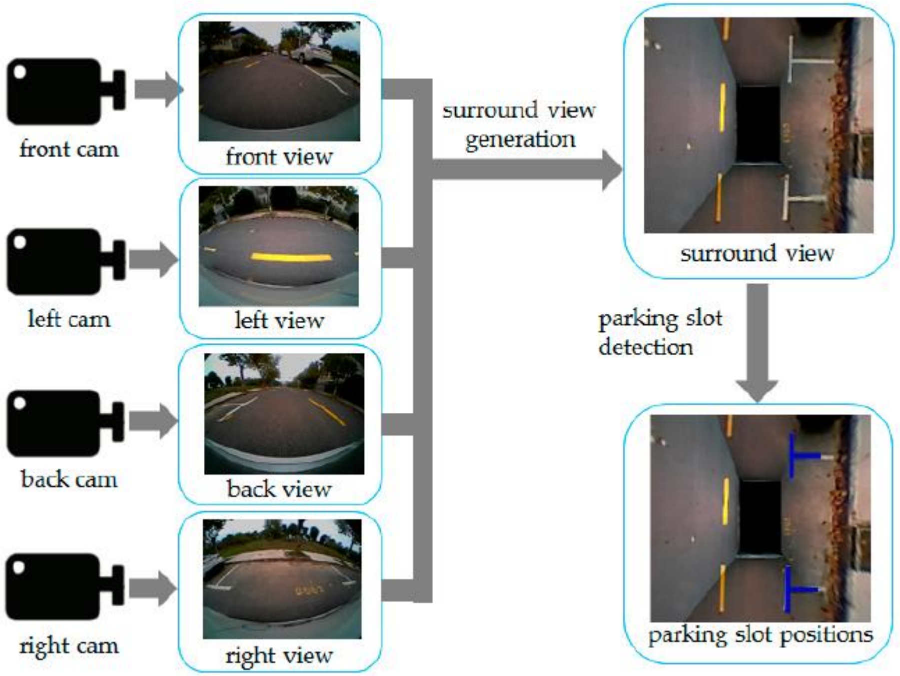
- Often accomplished through feedforward control using 2DOF vehicle dynamics model
  - Proportional-Integral-Differential (PID) Control
  - Sliding Mode Control (SMC)



Problem Background and MDP Formulation

- The features of the parking spot include T and L shaped markings
- In an end-to-end scheme, these features are identified and represented internally
- In this paper, a separate vision based detection module (with tracking) is used



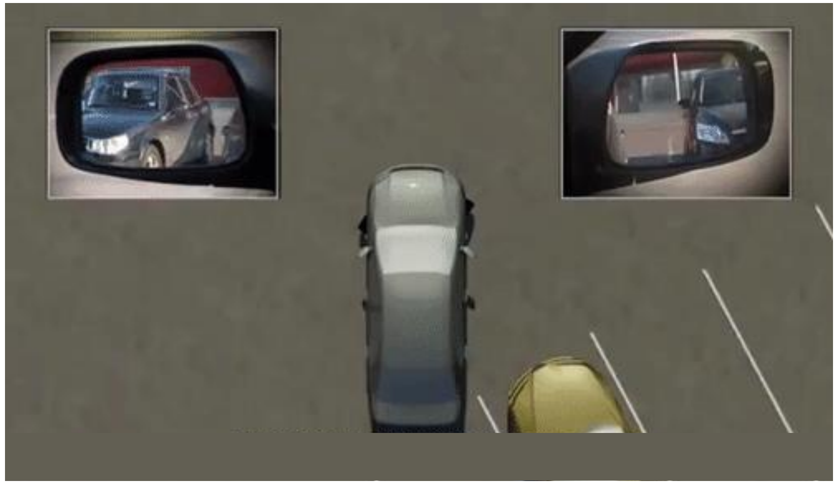


Problem Background and MDP Formulation

- The state, s, consists of features that correspond to coordinates of the 4 corners of the desired parking spot

- The action, a, refers to the continuous space of steering angle provided by the APS

- The state transition function, T, is unknown and not modelled explicitly



Problem Background and MDP Formulation

- The reward, r, is formulated as: $r = R_{cp} + R_l + R_d$

  Deviation from the center of the parking spot and attitude error:

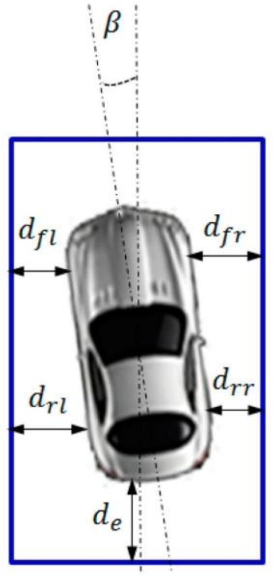
  - $R_{cp} = \left(5 - 5\left(\frac{1}{2}abs\left(Y_{p0} + Y_{p1}\right) + \frac{1}{2}abs\left(Y_{p2} + Y_{p3}\right)\right)\right) + \left(5 - 5abs\left(\frac{Y_{p0} - Y_{p3}}{X_{p0} - X_{p3}}\right)\right)$

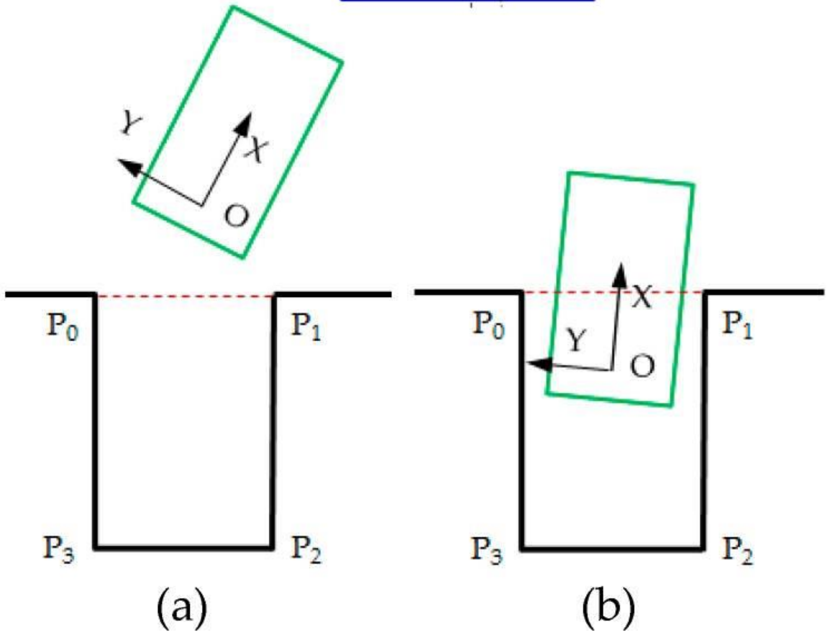
  Line Pressing:

  - $R_l = -10$

  Lateral Bias:

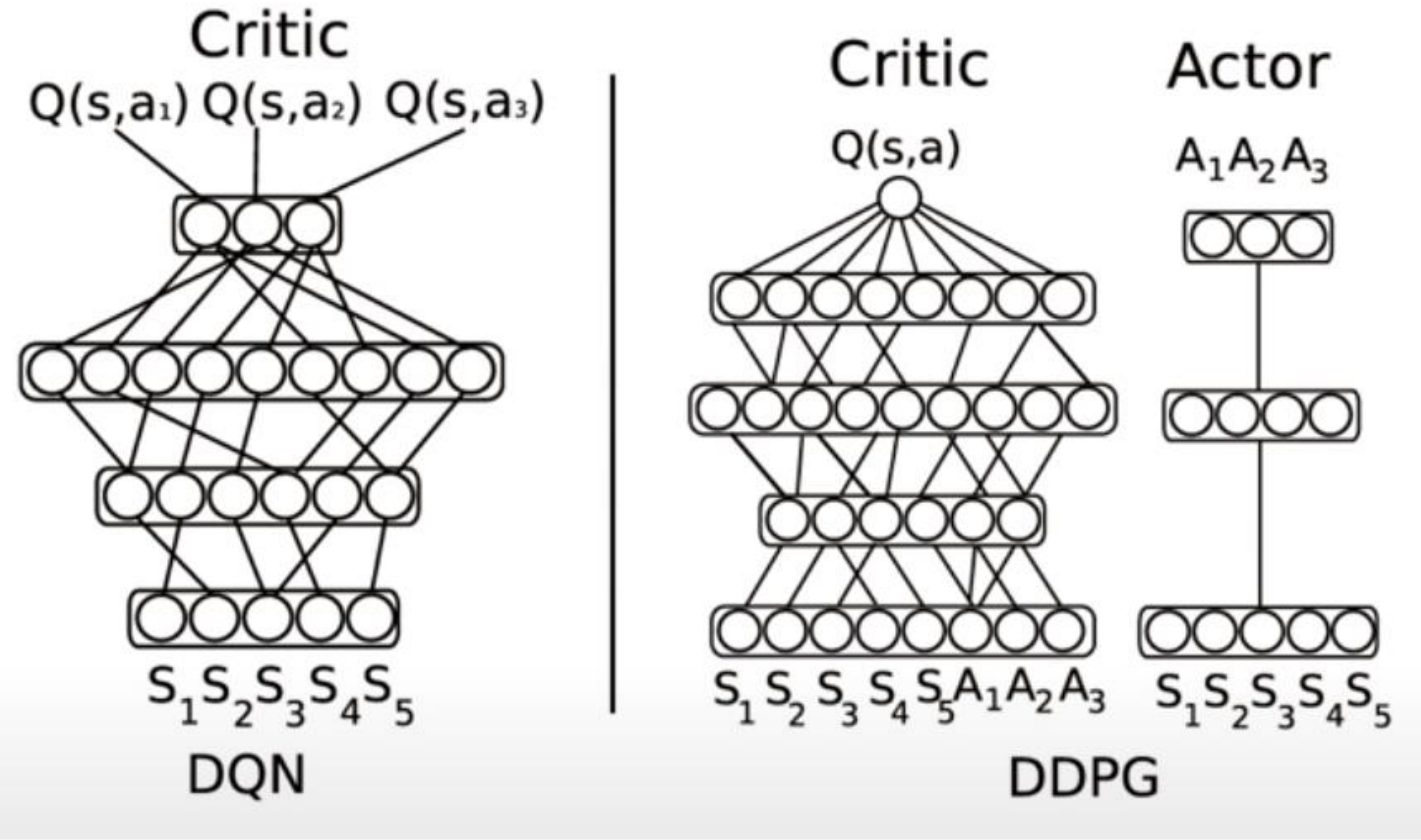
  - $R_d = -10$





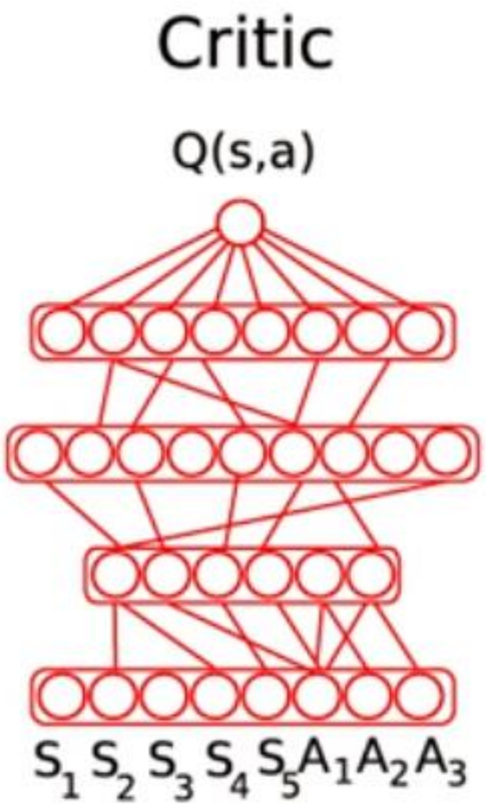
Deep Deterministic Policy Gradient (DDPG)

- DDPG is a model-free, off-policy actor-critic algorithm based on DPG



DDPG – Training Process

- Note that the action features are included as network inputs
- A target Q network is updated based on the hyperparameter $\tau < 1$
- The temporal difference between the target and Q network are used perform gradient updates
  - The parameters of the Q network are updated by minimizing the MSE loss function as in DQN

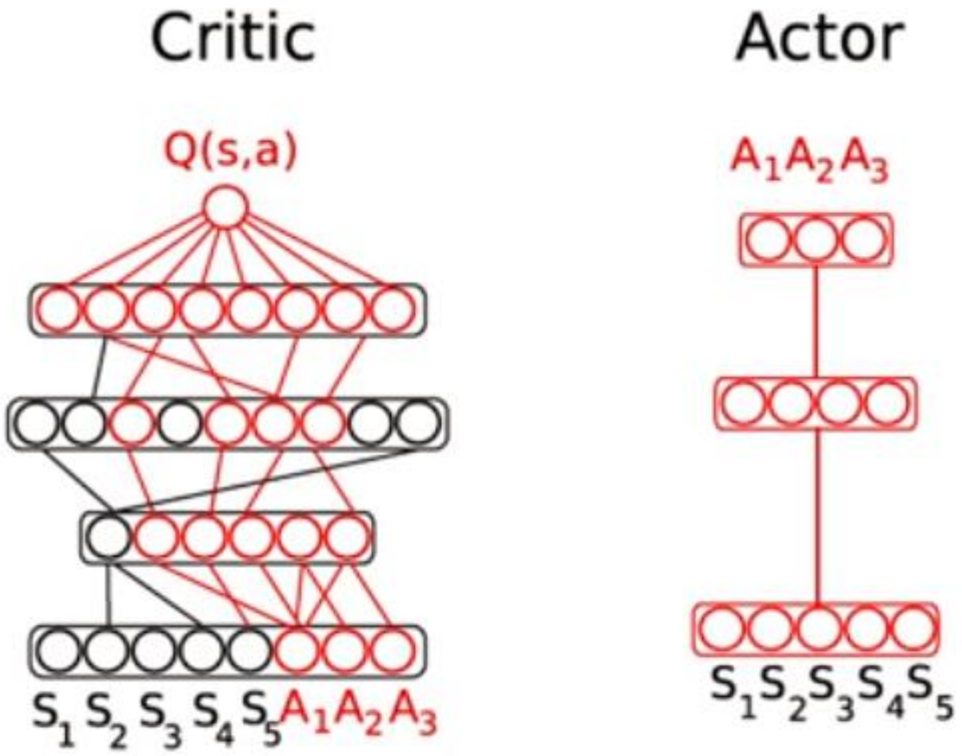


DDPG – Training Process

- The actor is trained using the DPG theorem:

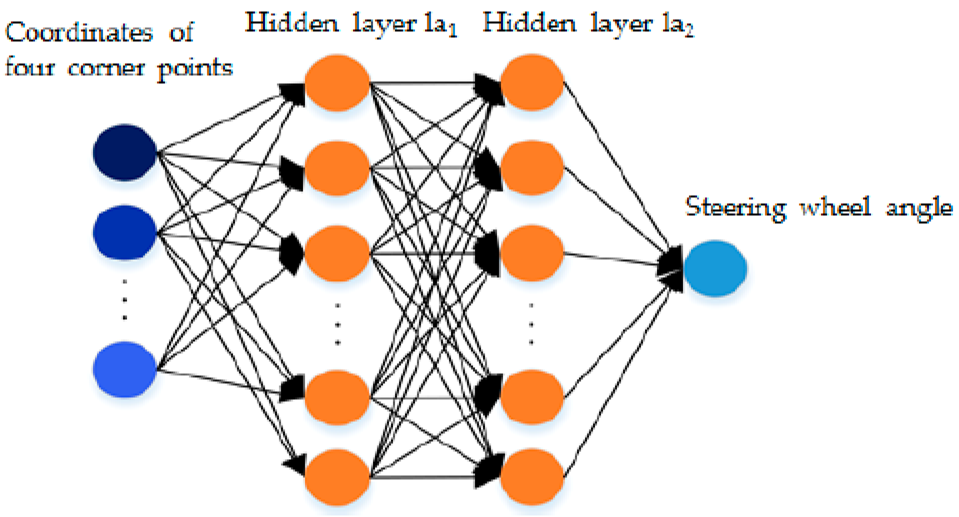
$$\nabla V_\theta(s_0) \propto E_{s\sim\mu_\theta(s)}\left[\nabla_\theta \pi_\theta(s) \nabla_{\mathrm{a}} Q_\theta(s,a)\Big|_{a=\pi_\theta(s)}\right]$$

- A target π network is updated based on the hyperparameter $\tau < 1$
- The presence of the Q function gradient over actions points to utilizing this Q function gradient as an error signal to update actor parameters

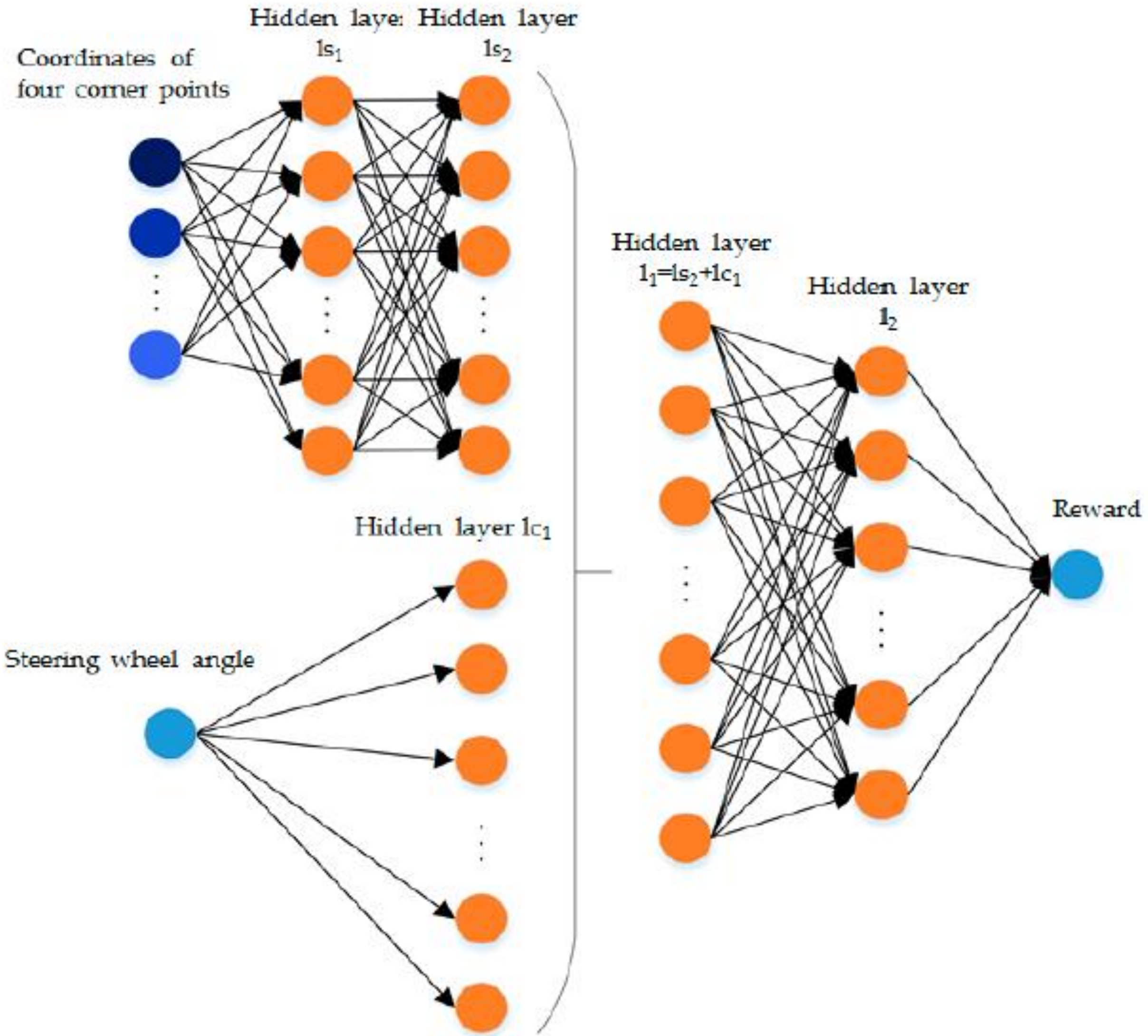


Network Architecture

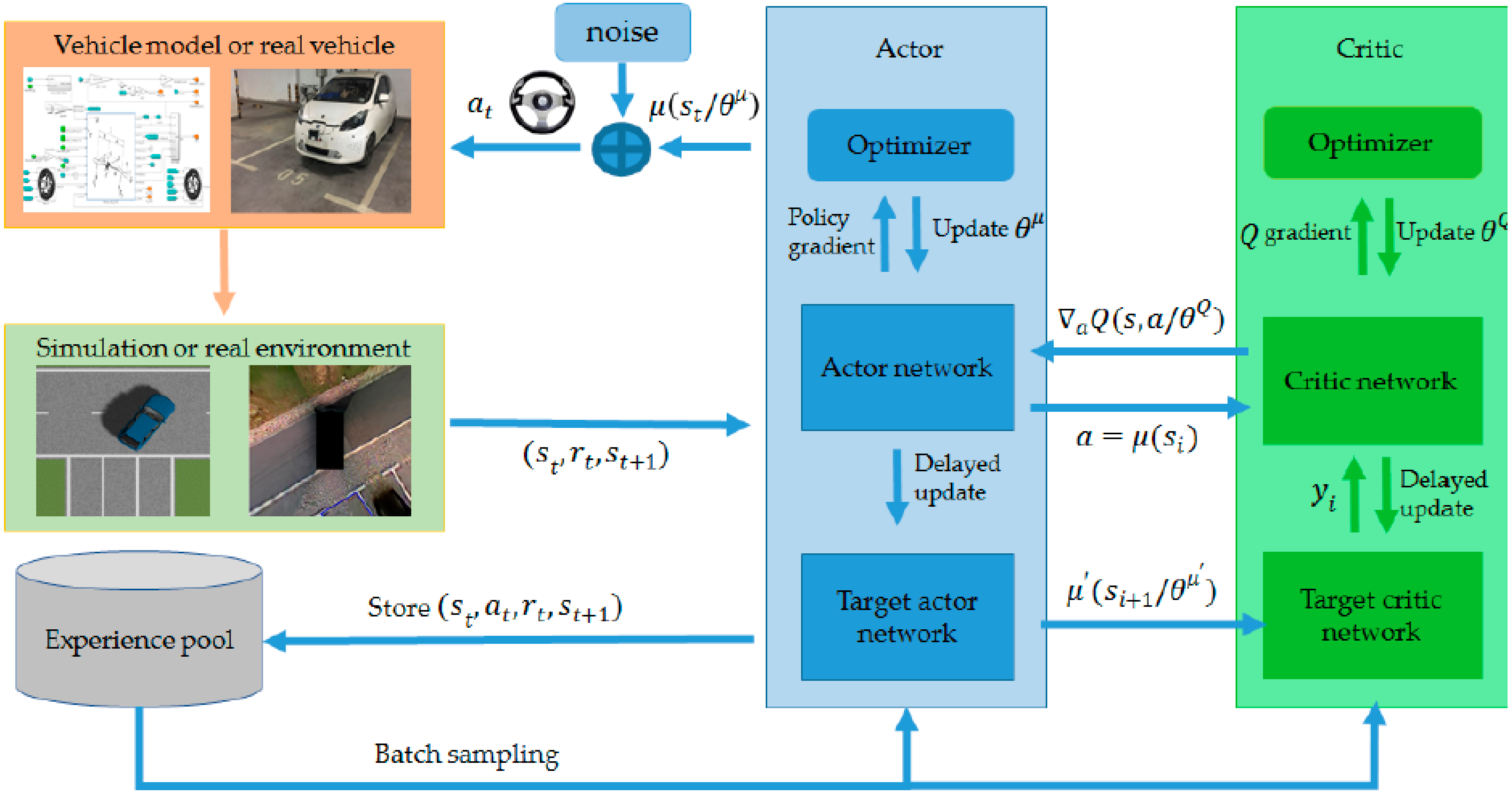
Actor



Critic

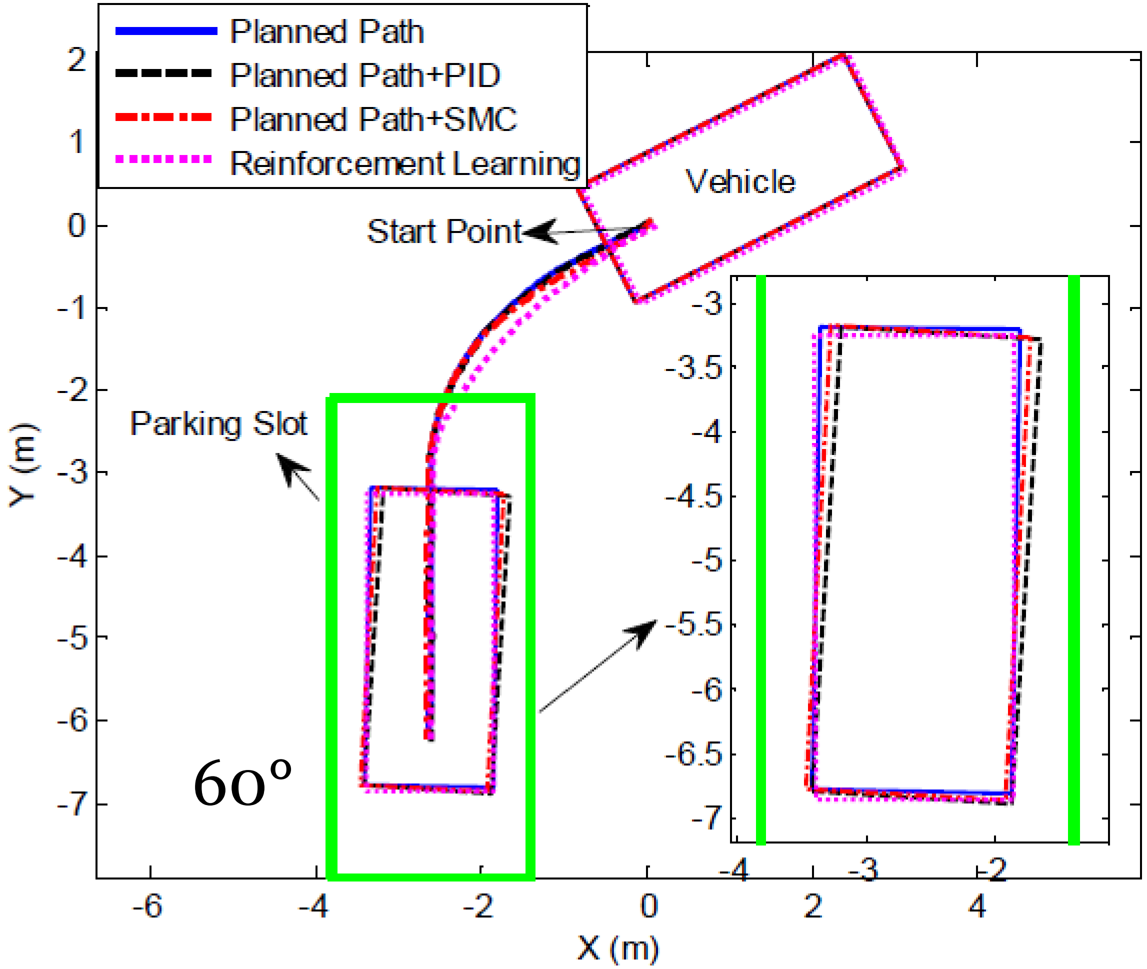


Overall Scheme



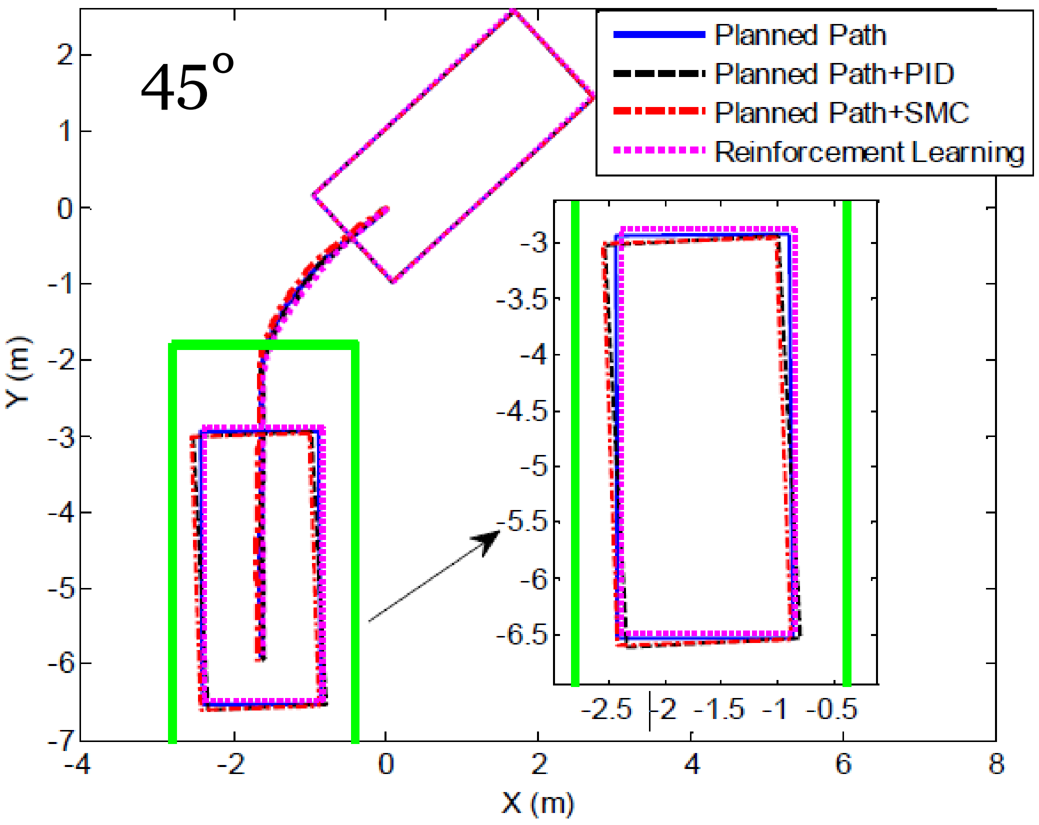
Experimental Evaluation – 60°

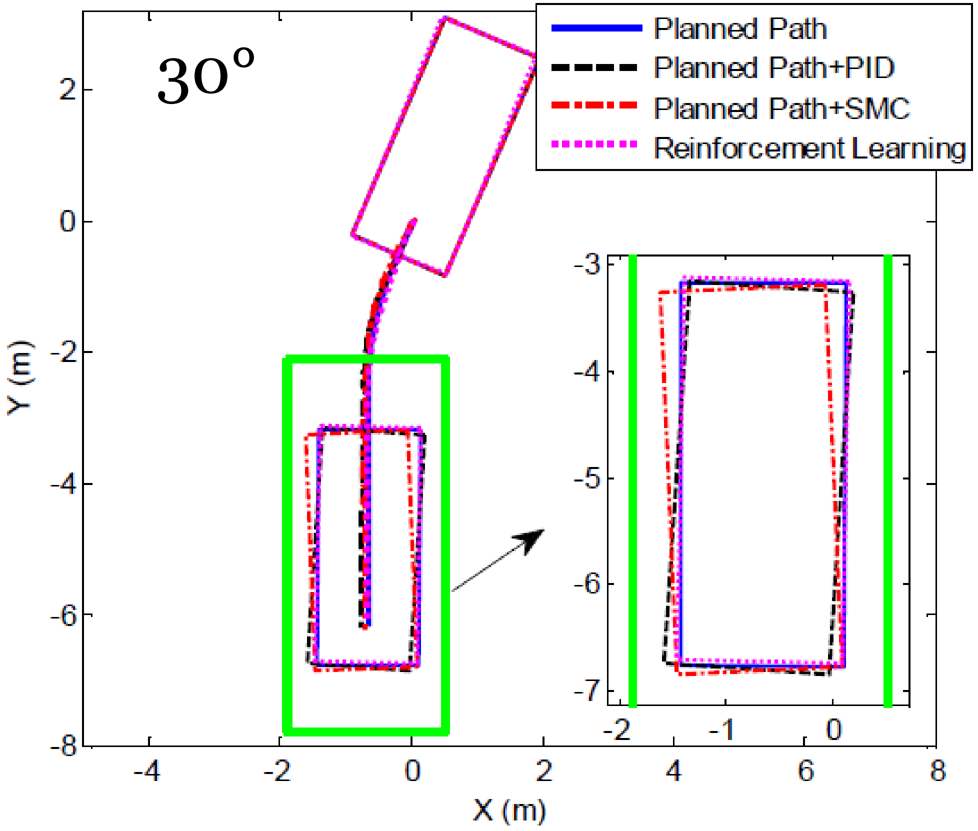
- Initial approach angles: 60,45, and 30°
- Attitude inclination error: -0.747°
- Path planning and tracking approaches such as PID and SMC show > 3° attitude error



Experimental Evaluation – 45 and 30°

- The attitude error remain < 1° for initial attitude angles of 45 and 30°





Discussion and Critique

- Significant improvement in inclination error
- Path Planning vs RL generated path: tracking issues
  - Tracking cannot be customized in unseen scenarios
  - Cases where approach angle is 90°
- Is the claim of the approach being "end-to-end" valid?
  - DDPG can learn policies end-to-end based on original paper
- Future directions: Inverse RL to mitigate sub-optimal reward convergence due to handcrafted reward scheme