Trajectory Optimization with PyBullet and CasADi

Chau Huynh
Department of Electrical and Computer Engineering
University of Washington
Seattle, U.S.A
cmdh@uw.edu

***Abstract*—This paper focuses on an application of trajectory optimization to maneuver a race car to a predetermined target. This application is demonstrated through a project using PyBullet as an environment for physical simulation and CasADi as a frame work to solve nonlinear optimization problems.**

***Index Terms*—trajectory optimization, optimization, nonlinear program**

I. Introduction

Trajectory optimization has numerous applications, especially in robotics. It assists in robot manipulation for robot arms movement, robots walking, and robots path-planning. With robot manipulation and path-planning, trajectory optimization can be used in industry production for factories, robot control, and in aerospace. To demonstrate an application of trajectory optimization, this paper focuses on a project that maneuver a simulated race car to a randomly determined target. The project outlined in this paper is based on researches done at University of California, Santa Barbara on the benefits of combining trajectory optimization and deep reinforcement learning [1]. However, the project described in this paper only focuses on implementing the trajectory optimization aspect of that research. The motivation behind this project is to reaffirm basic knowledge on trajectory optimization and to give hands-on experience applying trajectory optimization to a more complex physical system.

II. Overview

This project can simulate movements of a race car on a soccer field with a grid. The project is currently stuck at solving the nonlinear optimization problem because of errors with variable type casting from CasADi to Numpy. However, with more time and code debugging to resolve this issue, this project can display the behavior of the car to go towards the generated target.

III. Background

The dynamics for a robotic system can be written as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + A(q)^\top \lambda = B(q)u + F \quad (1)$$

where $M(q)$ is the inertial matrix, $C(q,\dot{q})\dot{q}$ is the centrifugal and Coriolis forces, $G(q)$ is the potentials (gravity), $A(q)^\top\lambda$ are constraint forces with λ as unknown multiplier a priori, $B(q)$ maps control inputs u into generalized forces, and F contains non-conservative forces such as friction.

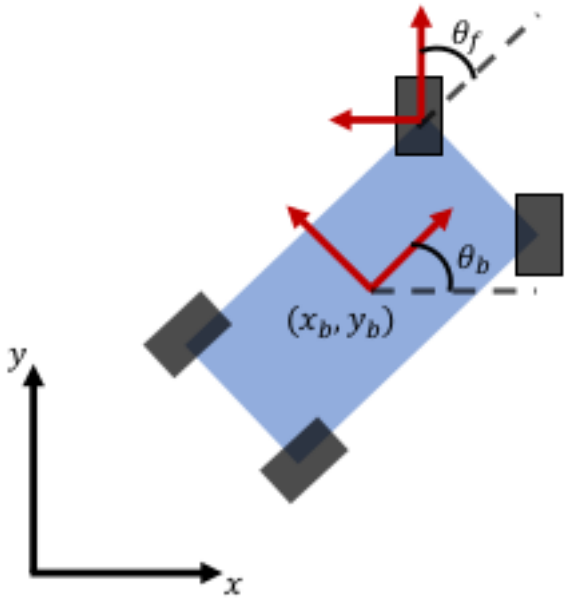

Fig. 1. Car model used for trajectory optimization.

For the race car, the state consists of the position variables q and velocity variables $\dot{q}$:

$$q = [x, y, \theta, \theta_w], \dot{q} = [\dot{x}, \dot{y}, \dot{\theta}, \dot{\theta}_w] \quad (2)$$

where (x, y) is the center of mass coordinates of the body of the car in the world frame, θ is the yaw angle of the body with respect to the global x-axis, and θ_w is the steering angle of the front wheels with respect to the body's x-axis [1].

The car's dynamics matrices M,C,G,A,B, and F can be obtained from reference [2]. Section 6 of reference [2] covers in detail the modeling of a four-wheel robot, using dynamics of a bicycle model, that can be directly applied to this project's race car.

IV. Optimization algorithm

Trajectory optimization problems are formulated in nonlinear programs in the general form of

$$\min_u J(u)$$

subject to dynamics constraints

where $J(u)$ is the cost function. For this project, the objective is to minimize the distance between the center of mass of the car and the target. So the cost function is

$$J = (x_g - x)^2 + (y_g - y)^2 \tag{3}$$

With the dynamics of the systems established, the optimization problem for the race car can be formulated as

$$\min_{q,\dot{q},u} \quad (x_g - x)^2 + (y_g - y)^2$$

$$\text{subject to } M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + A(q)^\top \lambda = B(q)u + F$$

$$A(q)\dot{q} = 0$$

The condition $A(q)\dot{q} = 0$ is to ensure that the system does not allow slipping and skidding. Slipping is free rolling in the direction the wheel is pointing and skidding is non-zero velocity along the wheel's rotation axis perpendicular to the free rolling direction. To solve this nonlinear problem, the framework CasADi is used with initial states as current the positions and velocities of the car and the final states as the location of the target. This project uses the solver "IPOPT" included in CasADi.

V. Results

The desired results is similar to results outlined in reference [1] for the pure trajectory optimization approach. The race car should move towards the target with optimal velocity and steering angle. However, this project is stuck at the solving the nonlinear program part because of bugs arising in the process of integrating CasADi with the dynamics.

VI. Discussion

This project is on-going to resolve the errors mentioned above. The future work of the project is to incorporate deep reinforcement learning that allow for skid to cut down the time the car took to arrive at the target.

References

[1] G. Bellegarda and K. Byl, "Combining Benefits from Trajectory Optimization and Deep Reinforcement Learning," https://arxiv.org/abs/1910.09667, 2019.

[2] G.D. Bellegarda,"Leveraging Trajectory Optimization to Improve Deep Reinforcement Learning, with Application to Agile Wheeled Robot Locomotion," University of California, Santa Barbara, https://escholarship.org/uc/item/6zs0h9nq, 2019.

[3] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," Mathematical Programming Computation, vol. 11, no. 1, pp. 1–36, 2019.

[4] E. Coumans and Y. Bai,"Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.