Autonomous Parking System

Youzhe Dou

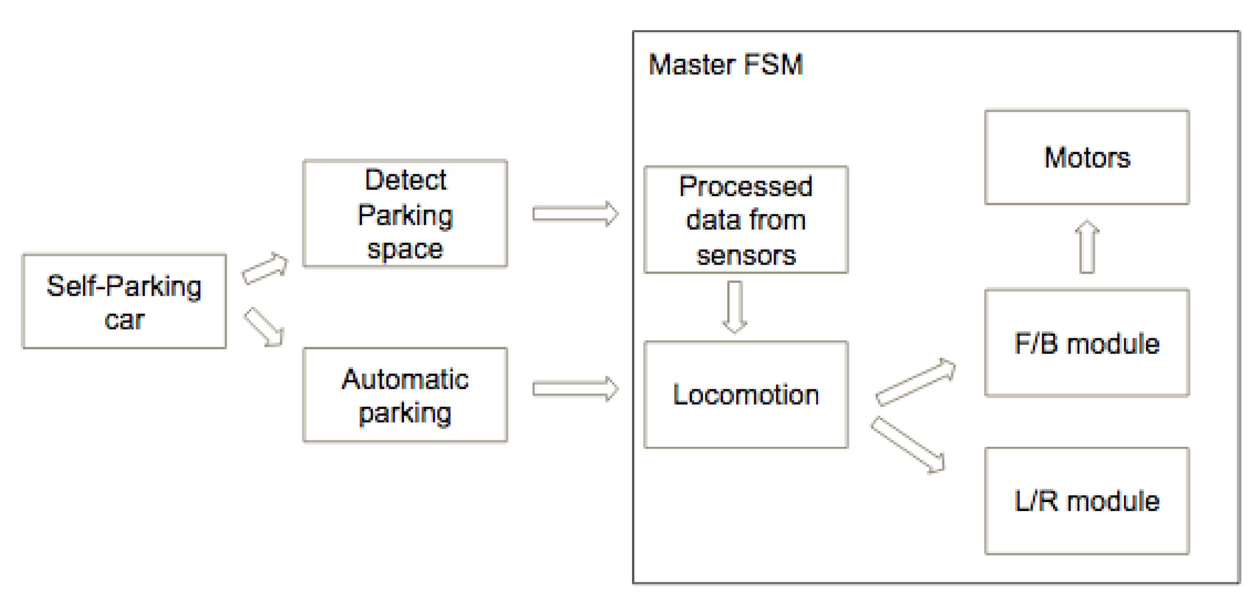
Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, Maryland, 21218, USA

INTRODUCTION and ABSTRACT

This autonomous parking system project aims to develop an intelligent system that will be an essential part of autonomous vehicles. The system provides general algorithms and standard procedures for a car to find an empty parking slot and park itself. The system is implemented using a Finite State Machine (FSM) because of its step-by-step nature. An Arduino controlled robotic chassis is used to mimic real-life vehicles. Sensors such as ultrasonic range finders, magnetometer and photonic wheel encoder are used to extract information and parameters that are important for motor control and decision making at each state. This system is scalable and can be tuned to cope with vehicles of any size and parking facilities of any structures accordingly.

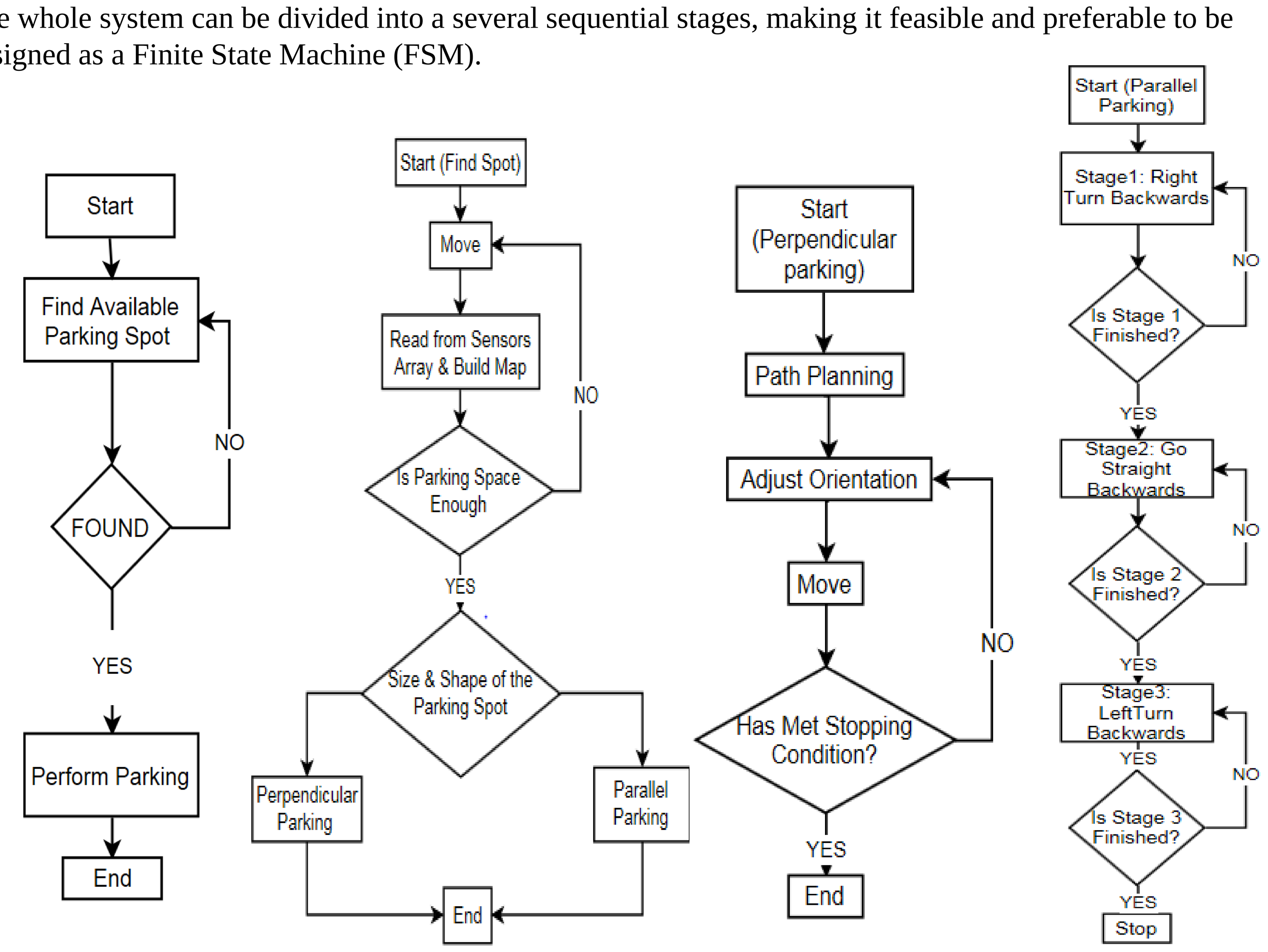
Driverless driving is becoming a trend in the motor vehicle industry. During the CES 2017, all the major car manufactures such as BMW, Volkswagen and Audi demonstrated their prototype for self-driving vehicles. As one of the essential part of self-driving, self-parking will be more frequently used since it is safer and the related technology is well developed. Different from traditional parking guidance tools such as rear camera and parking alarm, the self-parking technology aims to perform fully autonomous parking without any help needed from the driver. Self-parking significantly helps drivers who are not confident in their parking skills. Other than increasing the efficiency for the parking facilities, this hassle-free process also eliminates any safety concern for inexperienced drivers.

Such a complex system consists of many small tasks which will be carried out step by step. There are many researches focus on each task and solve the problems using different approaches. One way proposed to construct the environment is to use range finders arrays to build a 3D occupancy grids. Path planning models are created using both differential wheels and Ackermann steering.



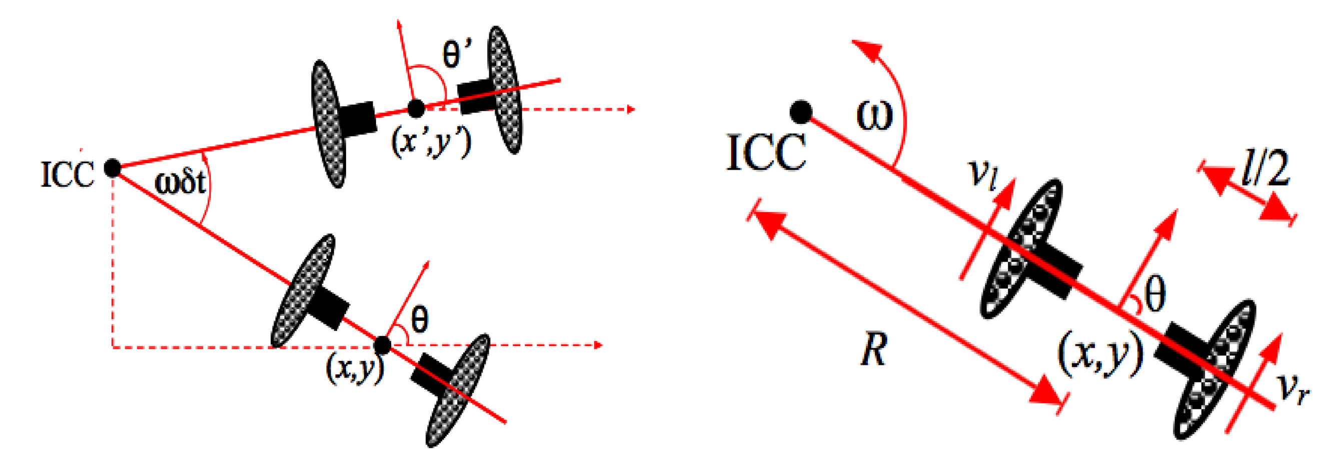
SYSTEM DESIGN AND ALGORITHM

The whole system can be divided into a several sequential stages, making it feasible and preferable to be designed as a Finite State Machine (FSM).



DIFFERENTIAL WHEEL DRIVE LOCOMOTION

The robotic chassis used in this project is equipped with differential drive wheels and the direction of motion is controlled by separately controlling speeds $v_l$and v_r of the left and right wheels respectively. Although differential-drive robot is able to turn "on the spot" by setting $v_l$and v_r same magnitude and different signs, this property is not used in this project because it will be irrelevant and impossible to apply the motion algorithm to real-life cars with Ackermann steering.

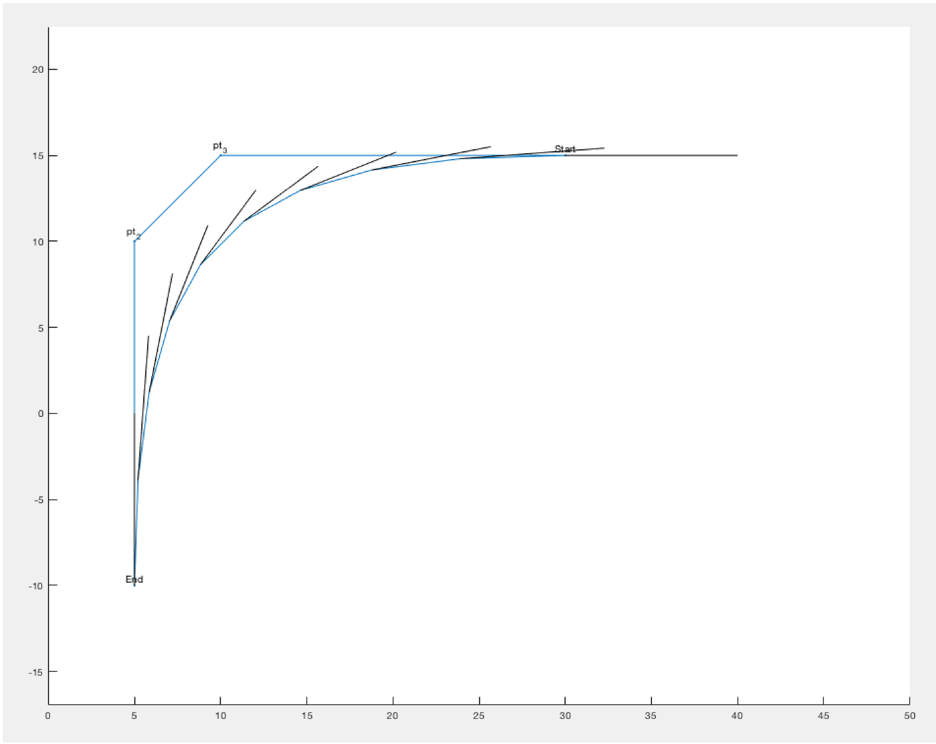


Forward kinematic equations:

$$ICC = [ICC_x, ICC_y] = [x - R\sin\theta, y + R\cos\theta]$$

$$\begin{bmatrix} x' \\ y' \\ \theta \end{bmatrix} = \begin{bmatrix} \cos(\omega\Delta t) & -\sin(\omega\Delta t) & 0 \\ \sin(\omega\Delta t) & \cos(\omega\Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\Delta t \end{bmatrix}$$

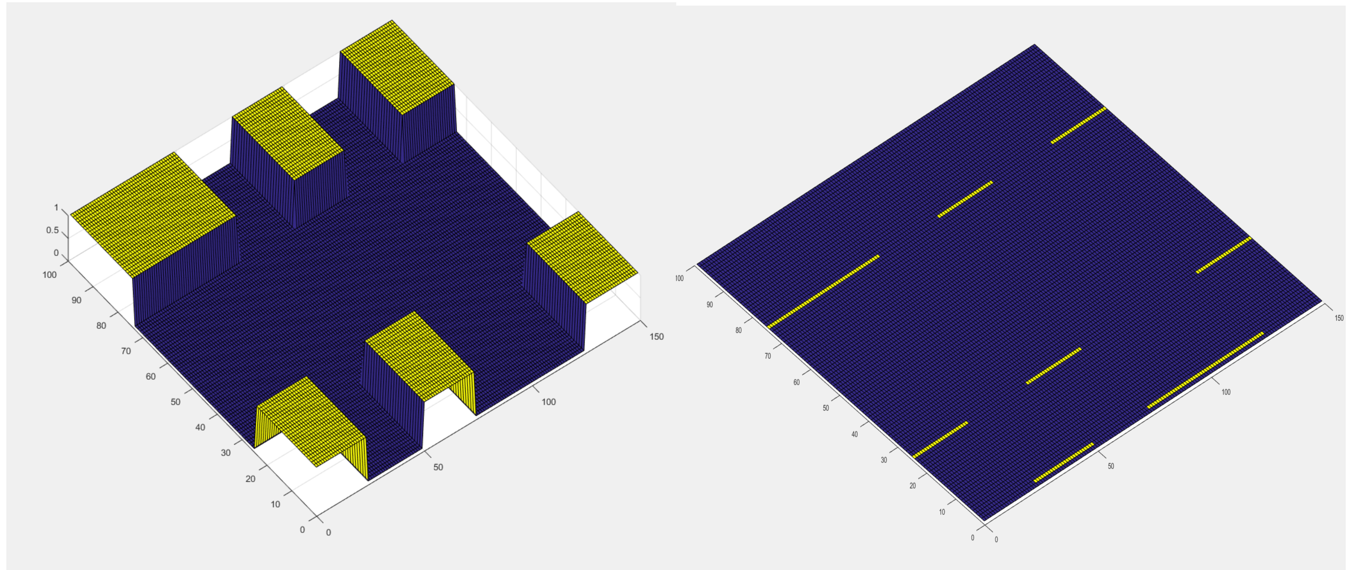
The paths for both perpendicular and parallel parking are generated using cubic Bezier curve and they are similar to real-life scenarios. The smoothness of this curve is dependent on the amount of increment of 't' and there will be a heading value assigned with each point on the curve.



Cubic Bezier curve : $B(t) = (1-t)^3Pt0 + (1-t)^2tPt1 + (1-t)t^2Pt2 + t^3Pt3, 0 \le t \le 1$

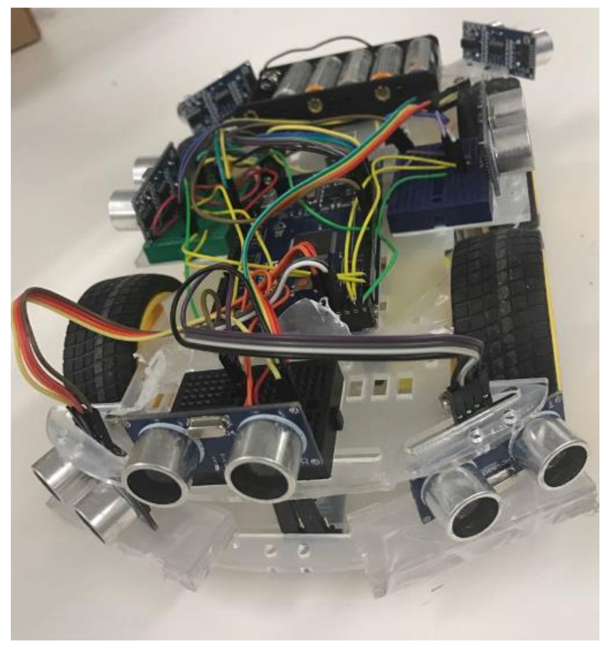
SURROUNDING ENVIRONMENT MAPPING

All the sensors are active and keep collecting information about the environment around the vehicle until the searching process is terminated. The entire space of the parking lot is represented by a 2D array of cells. Higher resolution yields better results with longer processing time which could be disastrous for such real-time system. Therefore, tuning of the size should be carried out during experiments



MATERIALS and PROTOTYPE

- Arduino Mega2560 with ATmega1280 Controller
  - 54 Digital I/O pins (15 of them are PWM output)
  - 16 Analog pins
  - 128 kB flash memory and 8 kB SRAM
  - 16 MHz Clock speed
- Ultrasonic Range Finder HC-SR04
  - 5V DC power supply
  - 40 kHz ultrasonic frequency
  - 1 cm resolution
  - 2 cm – 500 cm working range
- Triple Axis Magnetometer HMC5883L
  - I2C interface
  - 1-2 degree heading accuracy
  - Integrated 12 bits ADC
  - 160 Hz max data rate
- Photoelectric Codec Wheel Speed Encoder HC-020K
  - Working voltage: 4.5~5.5V
  - Launch tube current: If<20mA
- L298N Motor Drive Controller
  - 2-phase stepper motor, one 4-phase stepper motor or two DC motors
  - Dual-channel H-bridge driver working mode creates higher working efficiency.
  - High working power to 46V,large current can reach 3A MAX and continue current is 2A, power to 25W.
- Prototype parameters:



Size	9.6*6*3 inches
Weight	4.1 pounds
Power supply	9V
Time (searching for parking space)	Depends on parking facility
Time (perpendicular parking)	2.5 sec
Time (parallel parking)	4 sec
Space margin (perpendicular parking)	3 cm on left and right side
Space margin (parallel parking)	3 cm on left and right side 5 cm on front and back side

The prototype works perfectly for parking spot finding and mapping and perpendicular parking. It only works sometimes for parallel parking. This proves the concept and the algorithms are correctly implemented. However, parallel parking requires more precise control of the path and it is not feasible with these cheap motors due to the budget constrain. A possible future improvement will be using chassis with more steady control or even Ackermann steering.

REFERENCES and ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Prof. Ralph Etienne-Cummings, Prof. Pedro M. Julian , TAs: Johns Rattray, Tao Xiong and my fellow classmates for their guidance and suggestions. Without their support, it is not possible for me to learn and accomplish this mush.

References:

Morris, William, Ivan Dryanovski, and Jizhong Xiao. "3d indoor mapping for micro-uavs using hybrid range finders and multi-volume occupancy grids." *In RSS 2010 workshop on RGB-D: Advanced Reasoning with Depth Cameras*. 2010.

Wang, Daobin, et al. "Research on self-parking path planning algorithms."*Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference on*. IEEE, 2011.

DeSantis, Romano M. "Modeling and path-tracking control of a mobile wheeled robot with a differential drive." *Robotica* 13.04 (1995): 401-410.

Simionescu, P. A., and D. Beale. "Optimum synthesis of the four-bar function generator in its symmetric embodiment: the Ackermann steering linkage." *Mechanism and Machine Theory* 37.12 (2002): 1487-1504.