

# HANDSFREE VALET TECHNOLOGY (HFVT)

## Wireless sensor networks

Joshua Matt Abraham, Third Year, Electronics and Communication Engineering(ECE) Department, St.Joseph's College of Engineering, Old Mahabalipuram Road(OMR), Chennai-600119. [joshabraham101@gmail.com](mailto:joshabraham101@gmail.com) 9884547759

Joseph Jos K, Third Year, Electronics and Communication Engineering(ECE) Department, St.Joseph's College of Engineering, Old Mahabalipuram Road(OMR), Chennai-600119.

[josephjosk11594@gmail.com](mailto:josephjosk11594@gmail.com) 9445327328

### Abstract:

In a world striving for excellence in the field of automobiles, auto industries have been competing with each other to satisfy the users need for comfort, luxury and convenience. But one thing they seemed to have ignored until recently is a self-sufficient automated parking system. Parking as we know has been a dilemma for our drivers ever since dearth of parking space became a major issue. The systems at present are not completely autonomous and are severely limited. Therefore we propose Hands Free Valet Technology (HFVT): a technology that can identify a parking space and simultaneously perform a parallel or head-in park by itself thus providing the user an effective car parking system at his disposal. The goal of HFVT is to model real world dynamics of vehicular parking as accurately as possible regardless of parking lot structure (parallel or head-in) within the confines of the hardware and software while retaining the potential for technological advancement, scalability, and deployment of technology. The HFVT contains four primary components integrated neatly into a single package: a vehicle chassis, four sensors (two ultrasonic and two bumps), two motors, and a goal-based reflex algorithm. Firstly, a car is driven down a street searching for a parking space to its right using a distance sensor. When the car identifies a space, it checks to see whether it is large enough for the car to be parked in. If it determines that there is sufficient space, the car will begin parking into that space completely autonomously. It uses information from sensors placed on the front, right, left and rear of the car to direct the car into the parking space. Once the car is parked, it will remain in that position until it is reset.

### Introduction:

The design for a Hands Free Valet Technology (HFVT) took into account several considerations.

- Model real-time real-world dynamics (friction, velocity, turning radius, etc.)
- Function as an embedded system (use preceptors and actuators)
- Perform concurrent actions robustly (move, sense, think, and speak at the same time)
- Autonomously reach a goal state (correctly park the car in a valid parking spot)

### Keywords:

Actor Model:

It helps to translate the inputs and outputs of the sensors and motors into relevant mathematical signals. For example, angles and distances could be computed to determine arc length, velocity etc..

Concurrency Model:

It helps to separate and abstract away functionality between low level motor and sensor control from the high-level parking algorithm.

State Machine:

Finite State Machines(FSM) were heavily used to accurately define the transitions between goal states.

Valet:

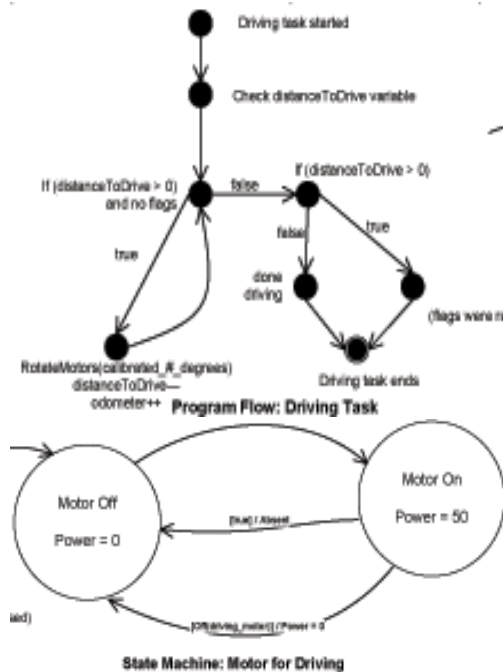
A service employee who parks cars for guests.

### Algorithms:

#### Driving Control Algorithm:

Requirements:

- drive specified distances
- drive continuously while tracking distance travelled
- stop at anytime and keep no state of previous driving commands
- drive concurrently to other tasks



**Sensor Algorithm:**

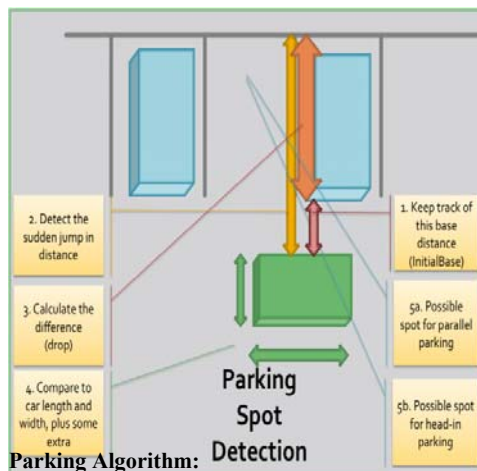
The sensor algorithm is a function that obtains, computes, and then repackages the two ultrasonic and two bump sensor raw output data in a manner that allows easy comprehension. It utilizes four global variables for the four respective sensors, which are constantly updated once the sensor algorithm computes the new raw data. The sensor algorithm is comprised of two methods, Sensys for ultrasonic and Touchsys for bump.

Sensys:

A while loop that continually polls. Sensys keeps a buffer of past ultrasonic sensor readings for each of the two sensors. At each poll, update the global variable for the particular ultrasonic sensor with the median of the buffered values.

Touchsys:

Continually check to see if there's a true reading on each of the two bump sensors. If there is, set the particular global bump variable to be true, else, to false.



The Parking algorithm must perform robustly within the model of a real world environment to keep the vehicle and passengers safe. Therefore, several criteria must remain satisfied throughout the entire parking process.

- The parking spot must be deep enough to position the car either for head-in or parallel parking.
- The width of the spot must also correspond to the specified depth. A spot that is deep enough to fit the car lengthwise, but is not wide enough should not be considered a spot at all. Similarly, spots that are not very deep must be long enough to fit a car for parallel parking.
- The spot must be empty.

The Parking algorithm uses a high-level finite state machine which consists of six states that must be traversed for each successful park:

Looking:

The vehicle begins with the state Looking. However, it is already performing an important action.

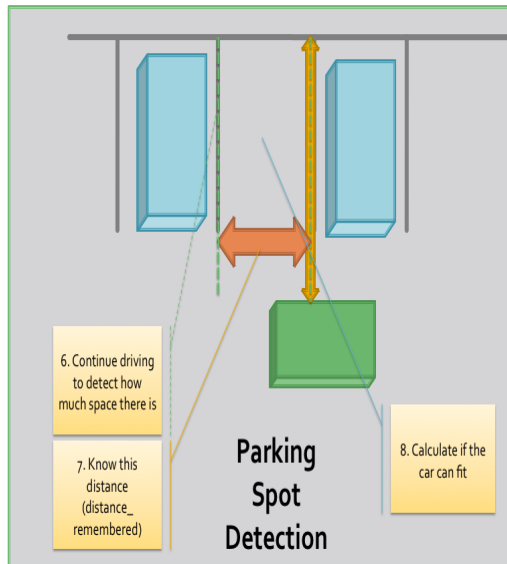
- HVFT polls the ultrasonic sensors to determine the initial distance from the right side of the car and sets it as the InitialBase.
- At this time, the algorithm has also initiated concurrent tasks: sensors (excluding secondary ultrasonic), driving, and a progress indicator.
- Assuming that the default action of HVFT is to drive straight, it will continue to be in that state until a desired transition condition (specific spike in sensor readings or a jump in distance for the depth between the right side of the car and a wall or open space) is met.
- The vehicle sets a flag isParallel based on the sensor reading, sets the distance to Drop, and transitions into the state Detecting.

Detecting:

HFVT looks to satisfy two conditions based on the type of spot for parking that is Parallel denotes.

The space maintains a depth as previously indicated by Drop or else an obstacle has appeared in the spot.

- The car has travelled a specific distance, distance\_remembered, which it increments after driving at each time step.
- This ensures that the space is wide enough for the head-in or parallel parking. If either of these conditions is violated, the vehicle will go back into looking.

**Alignment:**

The Alignment state is more dichotomous than Detecting depending on what isParallelindicates.

**Parallel Parking:**

The vehicle pulls forward to align its back wheels with the beginning of the parking spot, which accounts for the arc length computed by the turning radius for an S-turn and then proceeds to the state Park.

**Head-in Parking:**

The vehicle needs to align itself to the desired position which is to reverse back and away from the parking spot, similar to a half S-turn and then proceeds to the state Park.

**Park:**

- If isParallelset for head-in parking, the vehicle drives forward at a desired angle for the distance of the parking spot's width and depth or more specifically the distance\_remembered. Also, the vehicle must account for the InitialBasedistance that it was initially at.
- If isParallelset for parallel parking, the car must travel in a similar manner except in reverse. To accomplish an S-turn, the vehicle turns its wheels in one direction and travels half the distance required and turns its wheels in the opposite direction for the rest of the distance.

Thus, the vehicle successfully parked inside a spot, achieved its goal state, and transitioned into the state Stop.

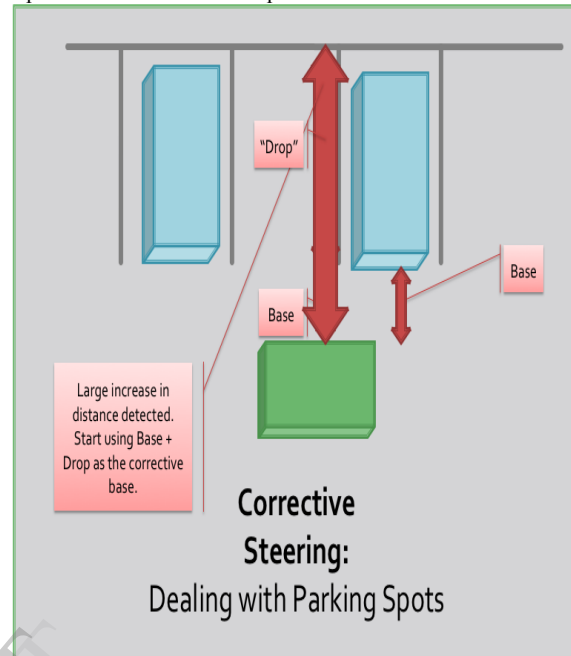
**Corrective Steering(Stop):**

In the Stop state, based on both the ultrasonic sensor readings, the vehicle now determines whether or not it is parallel to the surface on the right, as the differential determines the angle of misalignment and the direction to realign.

HFVT can also now maintain an optimal distance

- to the curb if parking parallel (or)

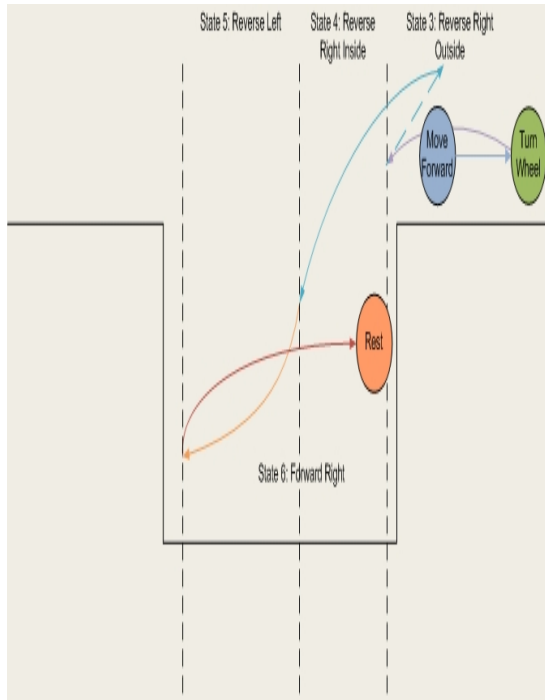
- to the neighbouring cars if parking head-in. Using the information from the two touch sensors, the vehicle will slowly align itself for the parking spot's defined width and depth.

**Design and Implementation:****Realistic Conditions:**

- It is better to use integer calculations rather than floating point to calculate distances. The increased accuracy would not significantly improve the parking ability because controlling the car with that degree of accuracy is difficult.
- The significant challenge in implementation is the concurrency design, centered on the shared usage of sensor data, which needed to be extracted, filtered, and processed.

**Sensor Choice and Placement:**

There are two touch sensors and two ultrasonic sensors. The two touch sensors were selected because they were accurate and reliable in short range and they are mounted on the front and rear bumpers. This placement was chosen because the algorithm did not need front or rear distance values, except at extremely close ranges where the ultrasonic was most inaccurate. Two ultrasonic sensors were selected because they provided distance readings in a range of distances, and both are mounted on the right side of the car. The sensors transmit signals, which bounce off objects around the car and reflect back to them. The HFVT then uses the amount of time that it takes those signals to return to calculate the location of the objects. From these four sensor inputs, the car collects all the data from the physical world and makes a judgment on the state of itself in relation to the world.

**Conclusion:**

Our final prototype and software system is able to detect and park into both head-in and parallel parking spots. This is done without hard-coding the path the car should take, but instead, interfacing with the environment with sensors, and moving in the environment adaptively. The successful prototype drives continuously but its behaviour is not always successful. It drives a short distance, stops, collects sensor data, “thinks,” and then repeats this process. However, in course of time with due alterations there is no doubt that HFVT will serve as an effective tool for automated car parking.

**Advantages:**

- HFVT can fit cars into smaller spaces than most drivers can manage on their own. This makes it easier for people to find parking spaces, and allows the same number of cars to take up fewer spaces.
- This technology would prevent mishaps such as minor dents and scratches.
- HFVT would function of very cheap sensors and be easily integrated into new situations.

