Craft #LikeABosch
online hackathon

Software development challenge

Introduction

Our task is to simulate the operation of an ultrasonic parking system. We are curious how ultrasonic sensors placed in the bumper of a car would detect the distance when different objects - obstacles - come into their field of view.

The diagrams below will help you in understanding.

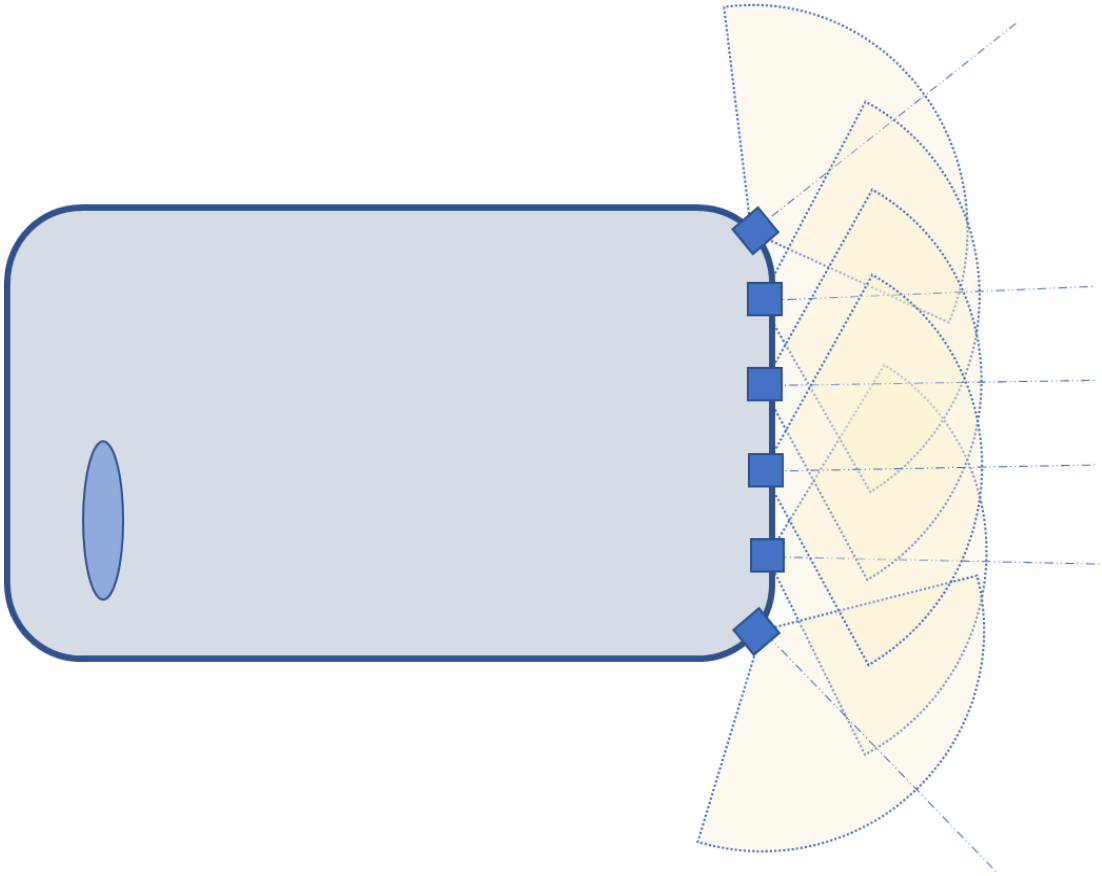

Figure 1: Schematic view of the car with sensor locations

The small blue ellipse symbolizes the steering wheel. Six sensors are mounted in the rear bumper of the car to detect obstacles behind the car.

The sensing area of the sensors can be described by a single pie-slice pattern with an opening angle of 120°. The detection distance is 3 m. It's important to see that the sensor emits a signal everywhere in this pie slice, which could hit our target objects. The question is which of the reflected signals the sensor can detect.

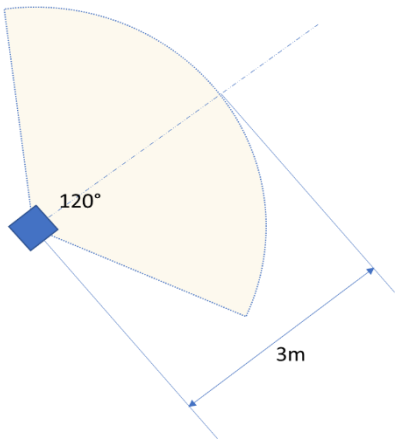

Figure 2: A sketch of the sensor with characteristics

The location and coordinates of the sensors in the car are the following:

Coordinates		
Sensor	X	Y
S1	110	95
S2	70	100
S3	30	100
S4	-30	100
S5	-70	100
S6	-110	95

The coordinate system of the car is the center of the rear axle, which can also be interpreted as the world coordinate system.

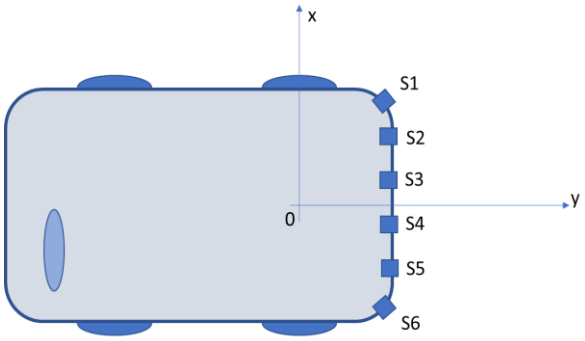

Figure 3: Schematic view of the car with coordinate system and sensors

For simplicity, sensor detection works in the following way:

We differentiate two kinds of echoes: *direct and cross*. Direct echo is always reflected from the surface of the obstacle perpendicular to the sensor. The sensor only detects it when the reflection is coming directly into the sensor.

The cross echo will always bounce back towards the adjacent sensor, only if it comes from the surface of the target (with the same angle of entry and exit) right into the adjacent sensor.

Each sensor can detect its own direct echo and the cross echo of two adjacent sensors.

The real distance measured by the sensors will always be half the distance travelled by the ultrasound.

So, the signals emitted by the sensor will hit the object everywhere within its characteristics, and out of the many signals coming back, it can only detect the signals visible in the image (green and purple lines). Practically, the task is to calculate these lines.

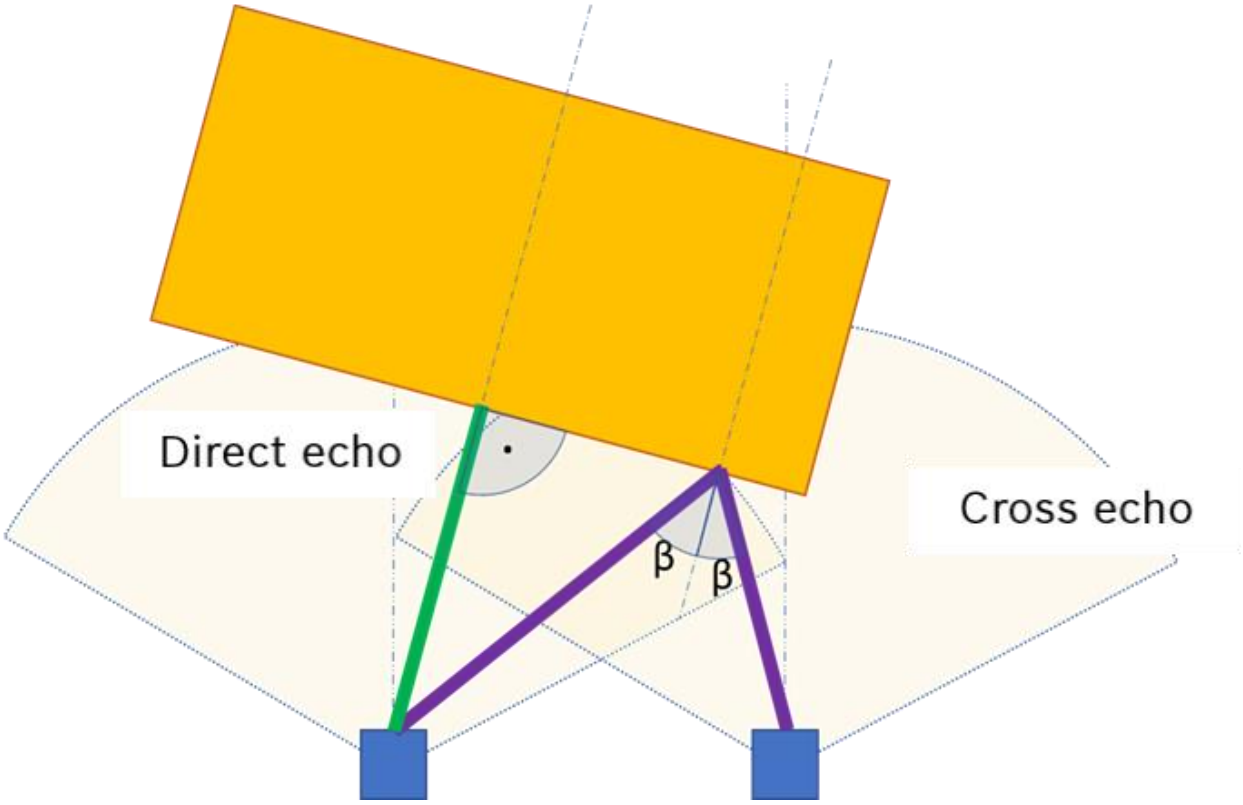

Figure 4: Schematic diagram of direct and cross echo

The challenge

The task is to simulate the real distance (direct and cross) measured by sensors for obstacles of arbitrary size, but only for "box" and "cylinder" (rectangle, circle) types of obstacles.

It is important that the simulation must be as real-time and interactive as possible, and that the obstacles can be moved and rotated freely in the field of view of the sensors. It is also essential that the obstacle should be large enough to be "seen" by several sensors.

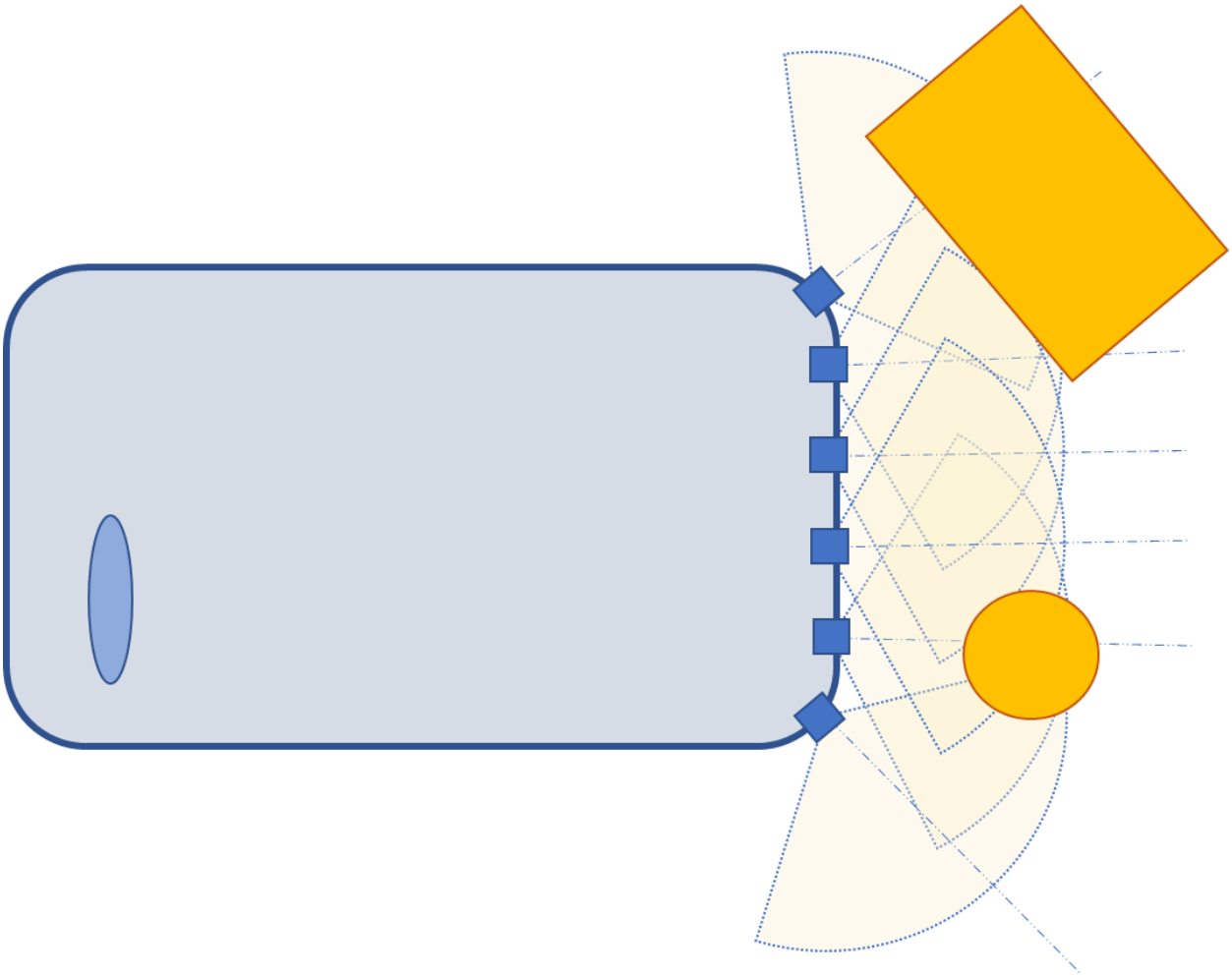

Figure 5: Example with a car and obstacles

In the second part of the exercise, you must show in a "live" demo that the radar sensor is fast and efficient. Obstacles can be adjusted, and the car can perform a reversing movement.

While reversing, the car approaches the objects we have placed, and the ultrasonic sensor measures the distance to each object. The car is automatically placed 5 to 10 meters away from the objects (not seeing anything) and a simulation method should slowly approach the car with a pushing motion to the object(s) behind it. If there are no objects behind it than drive to the end of the playing field and stop there.

If there are objects, go all the way to the nearest one (in a straight line with a smooth movement) and (also) color-code how far away it is. For example, if there is an object within 1 meter, the sensor that sees it should be orange.

Of course, this simulation can also be made more interactive, for example by moving the car by hand or by speeding up/slowing down the simulation.

Evaluation criteria

- Correctness of distance measurement algorithm, calculation method, its running time and memory requirements.
- How interactive and manageable the simulation is, how user-friendly the display of distances is.
- Quality of software code, architecture.