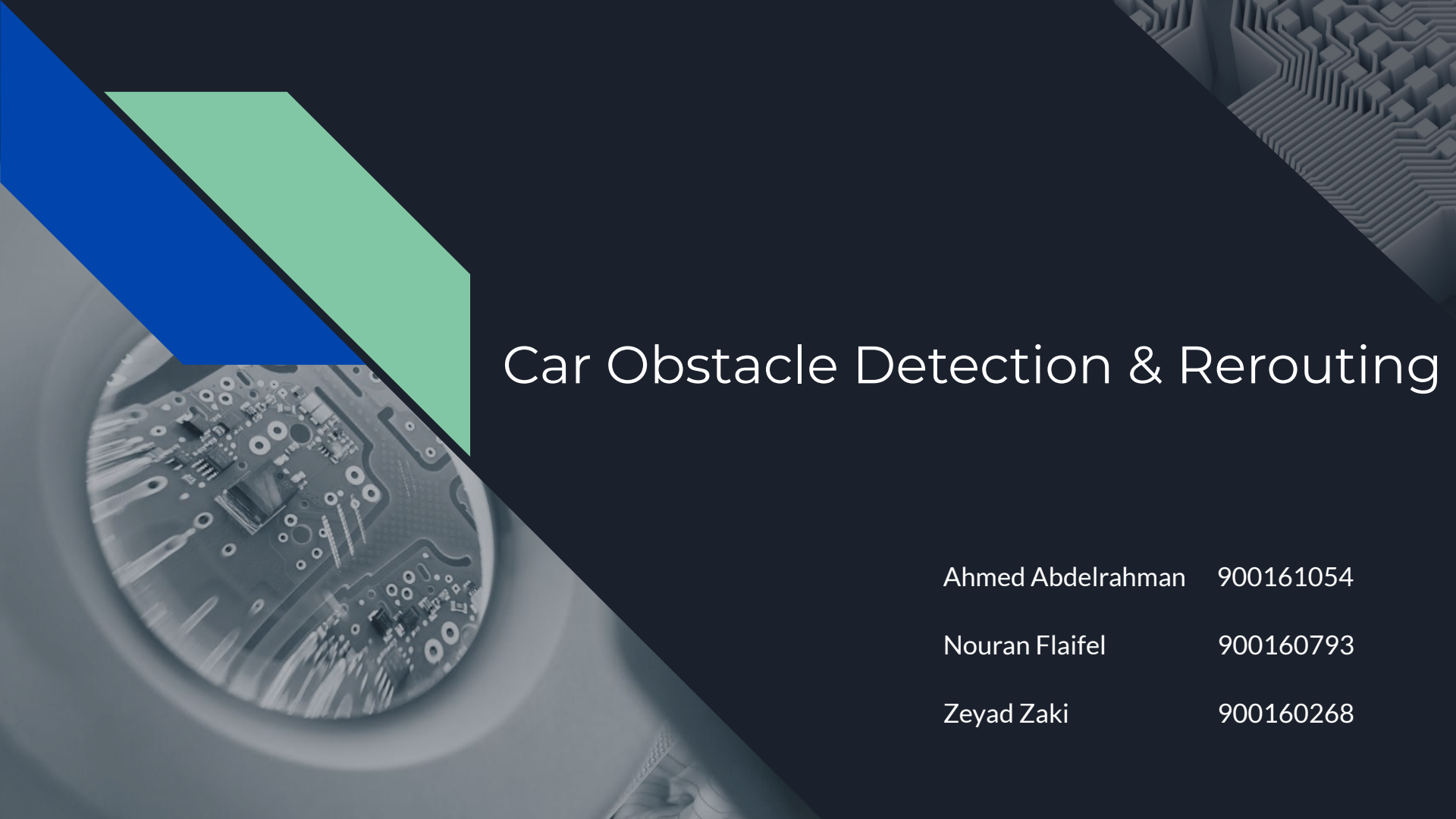


Car Obstacle Detection & Rerouting

Ahmed Abdelrahman	900161054
Nouran Flaifel	900160793
Zeyad Zaki	900160268

Agenda

Recap

Localization using Accelerometer

New Proposed Solution

Speed Calculation

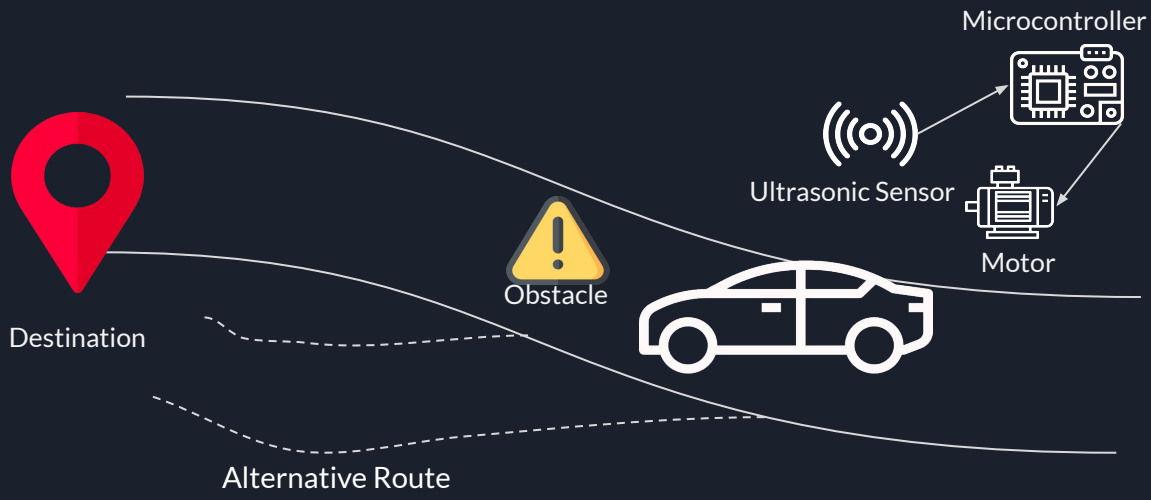
Mapping Function

Prototype Demo

What’s Next

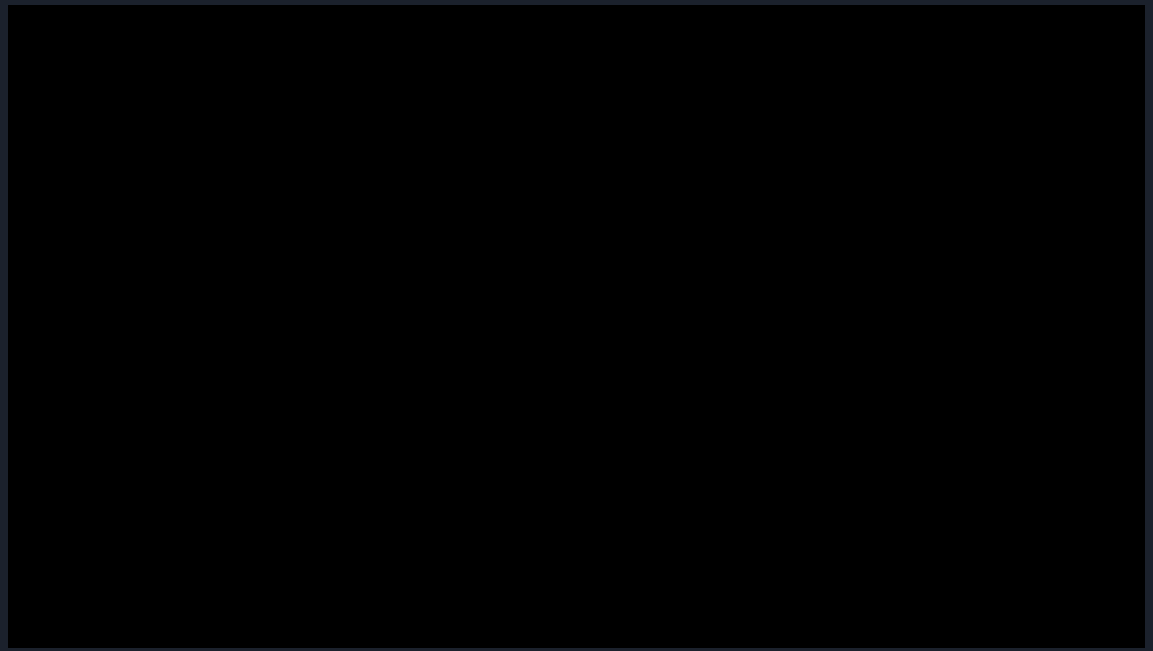
Recap

The smart car will be moving in a specific track to reach a final destination. It should be able to detect any obstacles that it faces and be able to reroute in order to avoid them while still aiming to reach the destination.



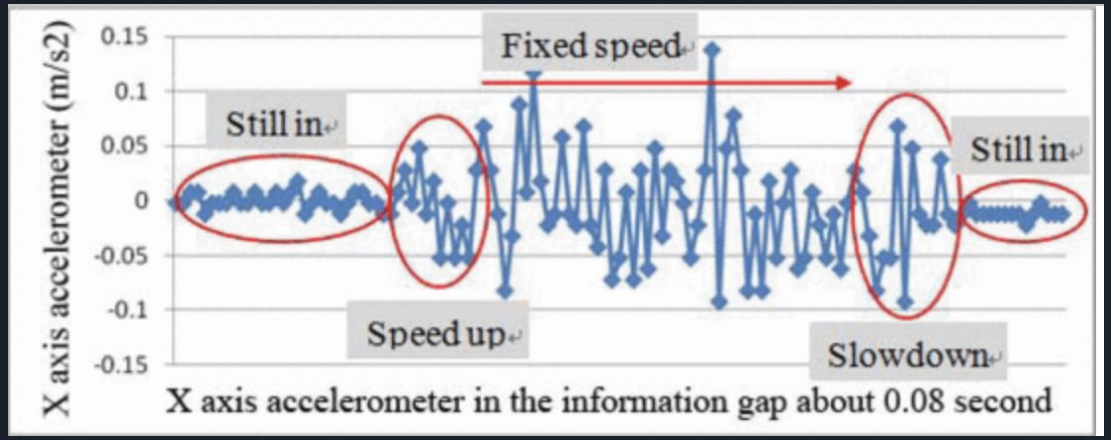
Recap

Last Prototype Progress:



Localization using Accelerometer

- Accelerometer readings are too sensitive as seen in the graph, fluctuating even when standing still
- This leads to us having inaccurate speed readings
- Having inaccurate speed readings will throw off our distance calculations
- So in conclusion the whole system will be faulty if we use the accelerometer



New Proposed Solution

Since the speed given to the Dagu is constant, we can use that directly to calculate the the location of the car in our map

- First we needed to calculate the speed the car was moving with (keeping in mind the time it takes to reach that speed)
- After that we created a map with relative positioning, setting each direction to be one of the 4 axes
- Using this information we can calculate the exact location by just having a timer that calculates the elapsed time

Speed Calculation

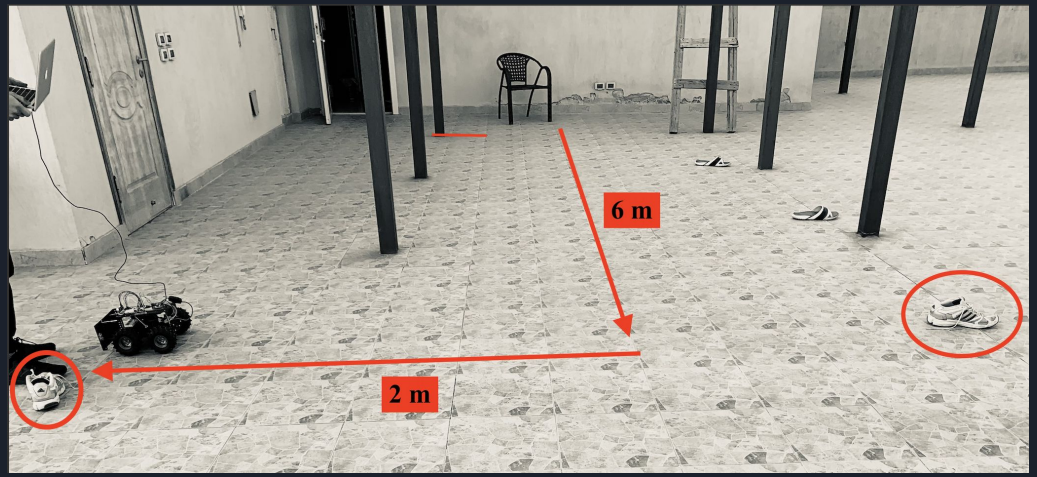
We did 3 trials to calculate the average velocity to be able to do the mapping and calculate the distance.

- Divided the area into 6 2m sections
- Recorded the time taken to reach each section
- Calculated the average velocity

Trials	Meters	Time (SW1)	Offset 1.5s	Time (SW2)	Offset 1.5s	Avg Time	Speed (m/s)	Average Speed
1	2	5.34	3.84	5.26	3.76	3.8	0.526315789	0.562046475
	4	3.47		3.43		3.45	0.579710145	
	6	3.49		3.61		3.55	0.563380282	
	8	3.37		3.57		3.47	0.576368876	
	10	3.63		3.63		3.63	0.550964187	
	12	3.61		3.34		3.475	0.575539568	
2	2	5.43	3.93	4.89	3.39	3.66	0.546448087	0.557936005
	4	3.55		3.65		3.6	0.555555556	
	6	3.5		3.46		3.48	0.574712644	
	8	3.59		3.6		3.595	0.556328234	
	10	3.81		3.5		3.655	0.547195622	
	12	3.31		3.74		3.525	0.567375887	
3	2	5.33	3.83	5	3.5	3.665	0.545702592	0.563986527
	4	3.62		3.58		3.6	0.555555556	
	6	3.37		3.35		3.36	0.595238095	
	8	3.83		3.41		3.62	0.552486188	
	10	3.29		3.67		3.48	0.574712644	
	12	3.56		3.58		3.57	0.56022409	

Mapping Function

- We created a function that receives the final destination coordinates (x,y) in meters.
- The function moves the car in the x direction until destination is reached, then starts moving in the y direction.
- The location is calculated every 100ms



Demo

What’s next

- Integrating the obstacle detection into the maps
- Designing several maps with different obstacle locations
- Could experiment using a wheel encoder to get better real-time speed estimates
- Finalizing the project and posting it on GH

Thank you!