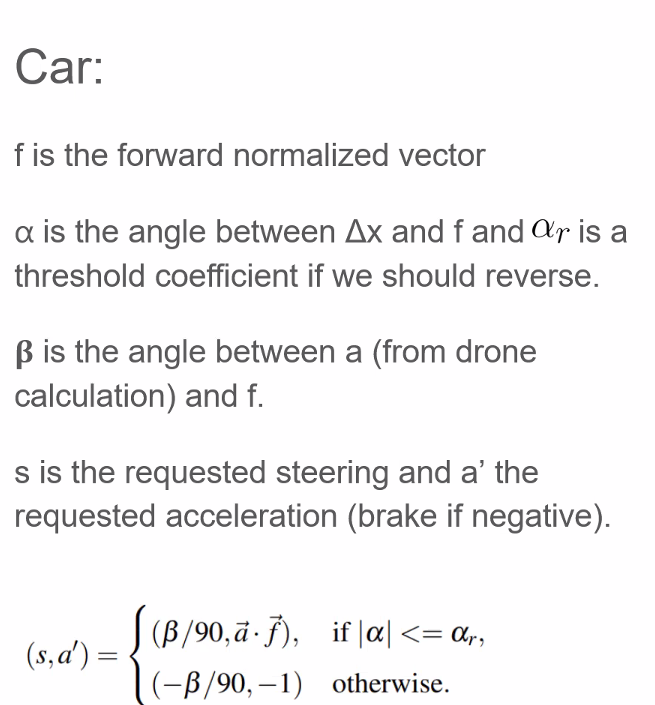
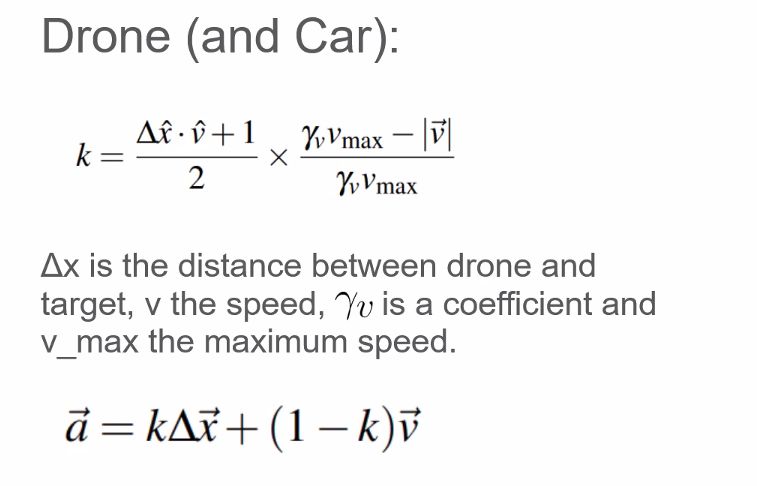
**Gruppo 3:** facevano decelerare la macchina quando il punto era nelle vicinanze del waypoint.

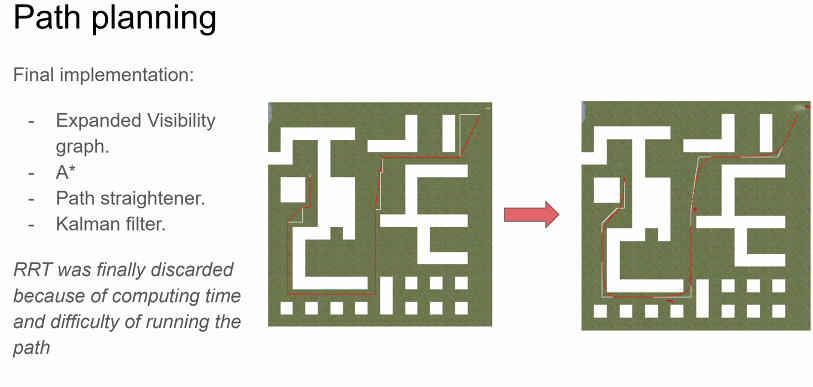
**Group 9**: collision grid over the transactable[i[j]

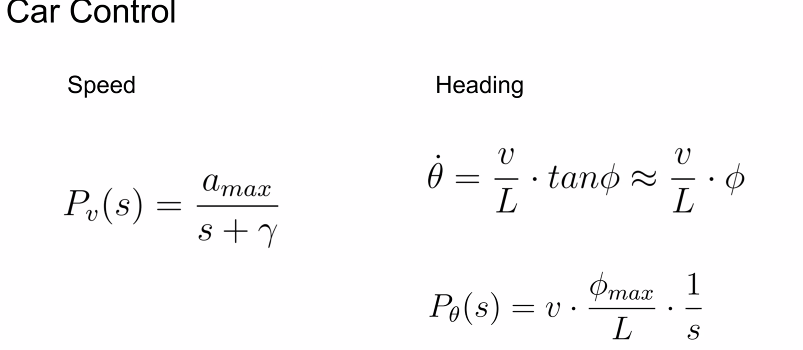
Calcolano un coefficiente e poi calcolano acceleazione con un termine negativo

Prodotto scalare: se positivo la macchine accellera, altrimenti diminuisce.

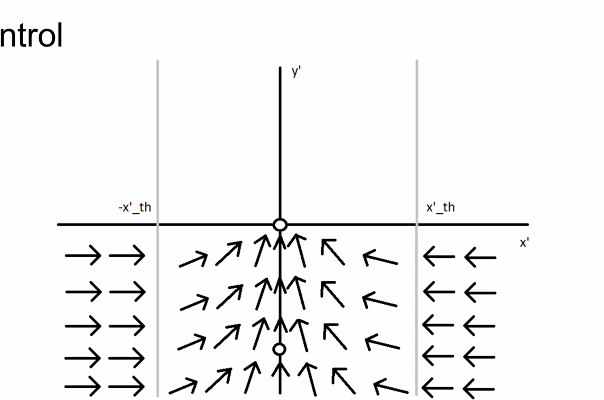


**Group 7 -** Car Specialists:





Per angoli piccoli l’angolo non variava olto, hanno usato un PID più semplice.



Transform the plane such that goal is center 0.

X is how far you deviated from the path.

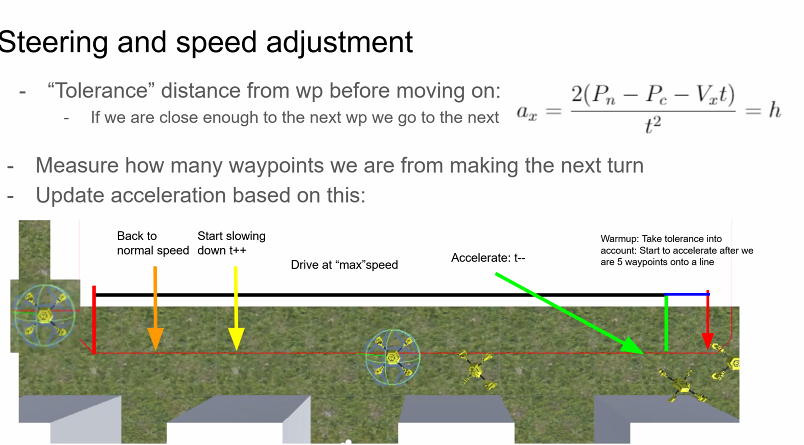
Adattato il car model al drone

Adapted Velocity

Good steering, easily amnage short turns.

Group 13:

Safety chec- padding to avoid collision.



Tolerance = green.

You have a step size which will make a grid, but not grid defined by the beginning.

Use acceleration.

Hybrid a\* for the car

Discretized steeing angles -pi/8,

5 states per grid.

Small gid size. Penalised taking turns

1 velocity for straight line

1 velocity for curving

Looks 3 points ahead in the future.

Euclidean distance cost:

Steps costs:  
brances adds more,

Can your car steering handle reversing: no.

No error recovery.

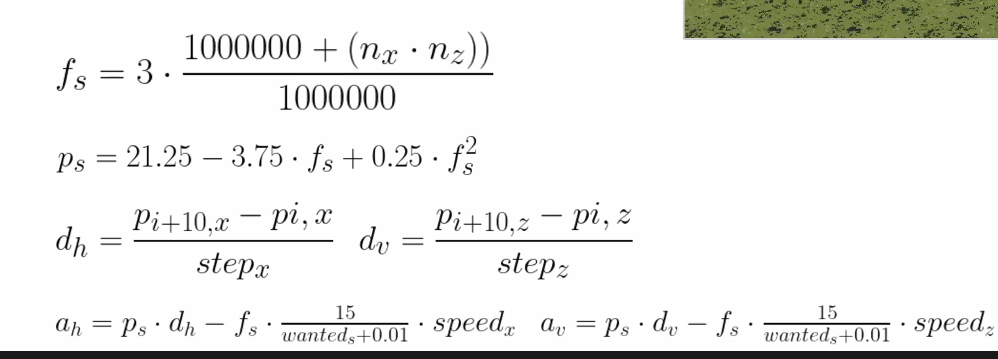
How long does it take to converge? Max 15000 iteations. We allowed 3 tates per grid, and 5 added more steering angle.

**Group1:**

Share a lot with previous group: we used A\*.

Tells you how fast to go for each

Computes maximum speed.



They used 3 nodes for steering angles.

To ajust acceleration

They look 2 nodes ahead. And then if steering is needed, they break, otherwise they accelerate.

**Group 10:**

Rrt\* didn’t work for terrain A and B. find to try nearest point.

For terrain

Drone model: slow down the model by calculating the dot product