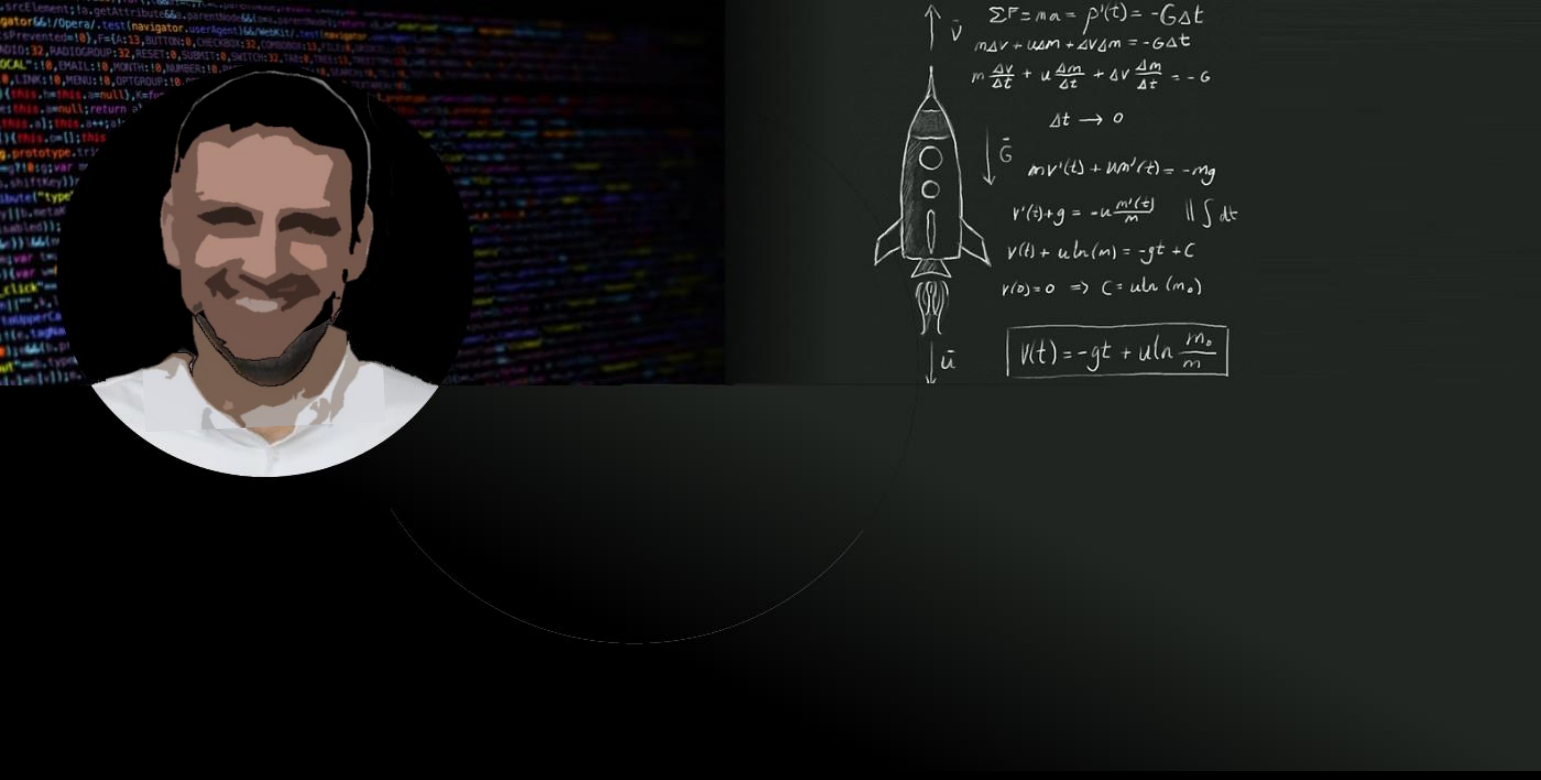


ALGORITHMS PHILOSOPHY

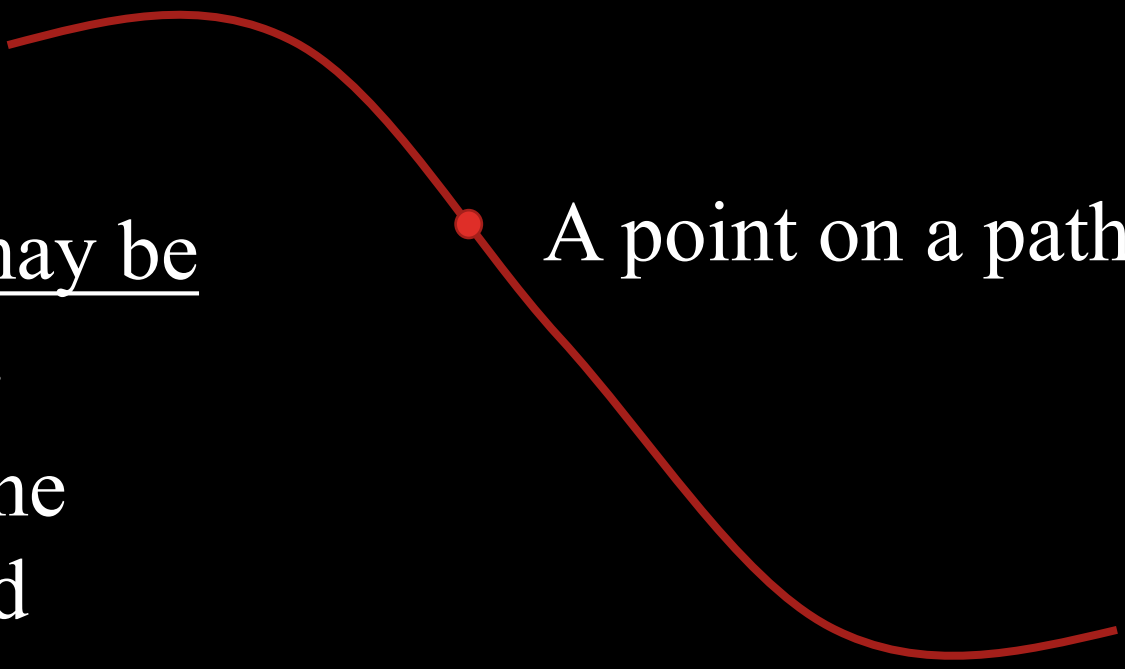
C4dynamics

Many engineers develop their algorithms detached from context.

If you are engaged with physical systems, follow this..

C4dynamics

A POINT



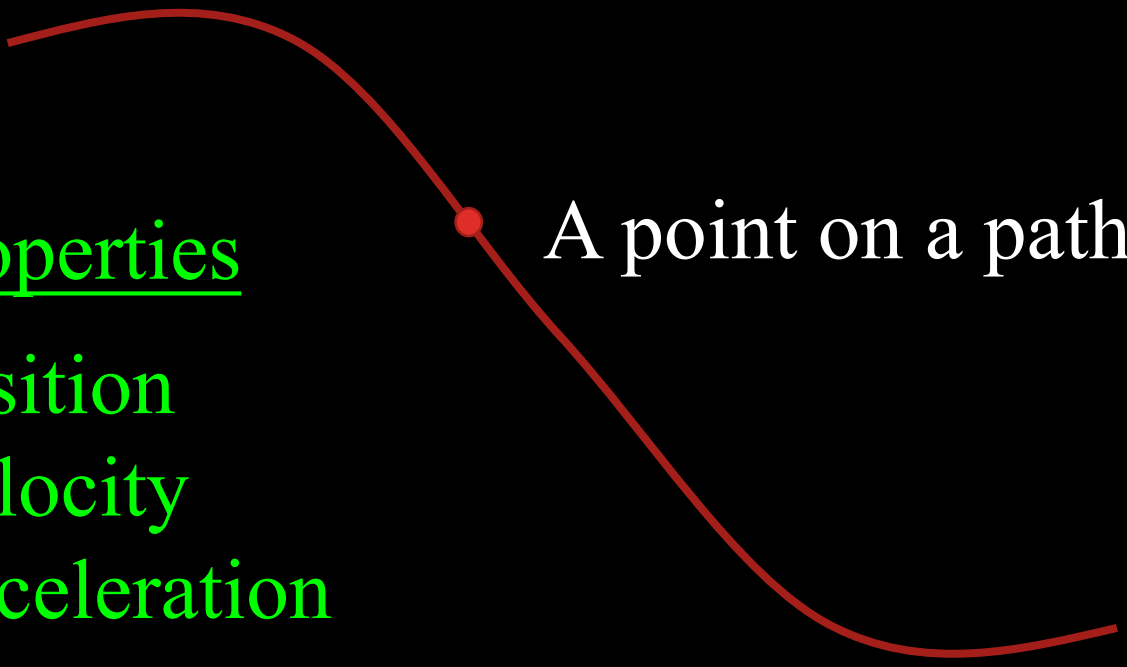
A point on a path

It may be

Car
Plane
Bird
...

C4dynamics

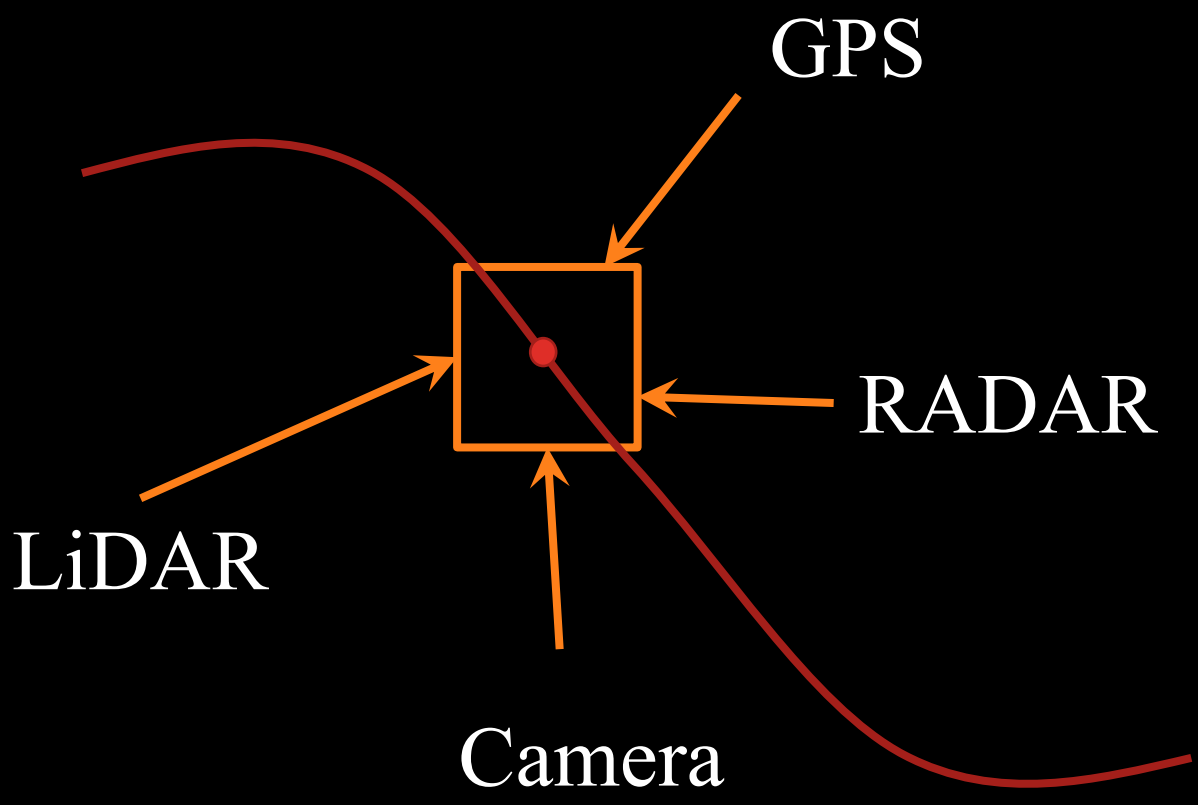
A POINT



Properties

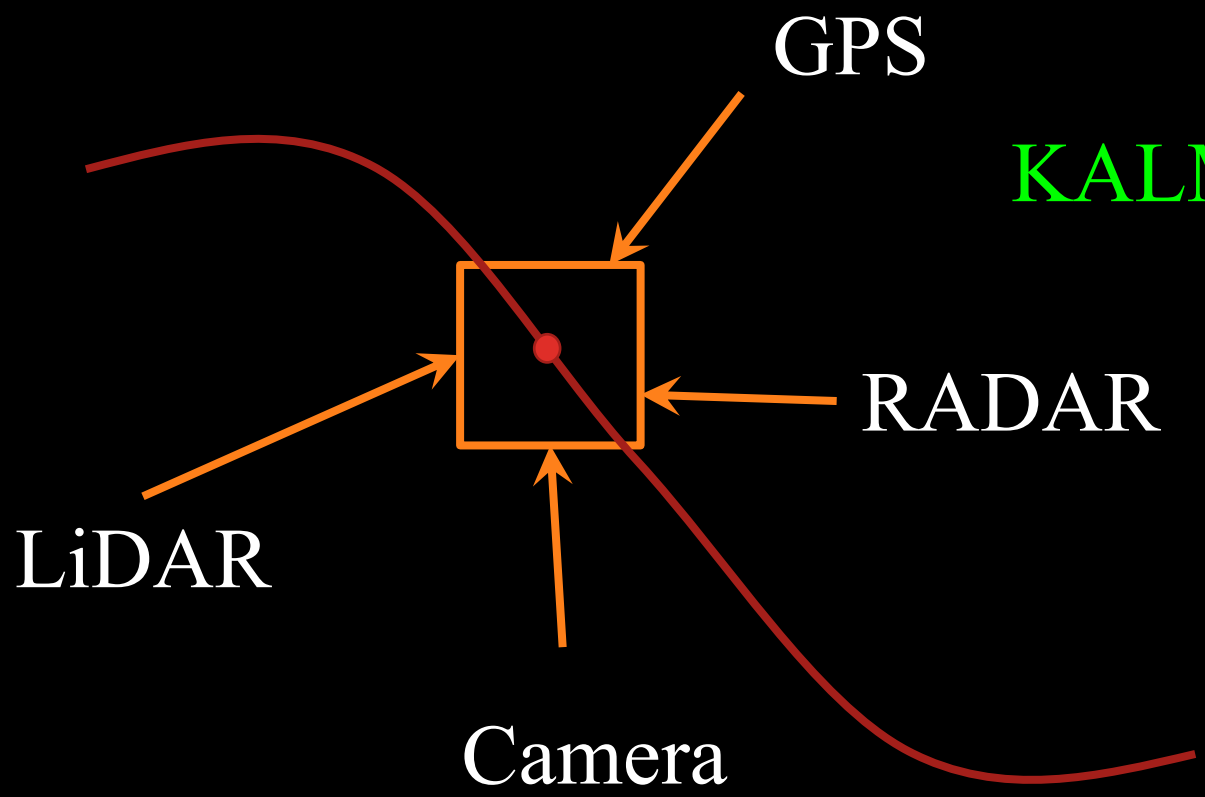
Position
Velocity
Acceleration

UPDATE FROM SENSORS



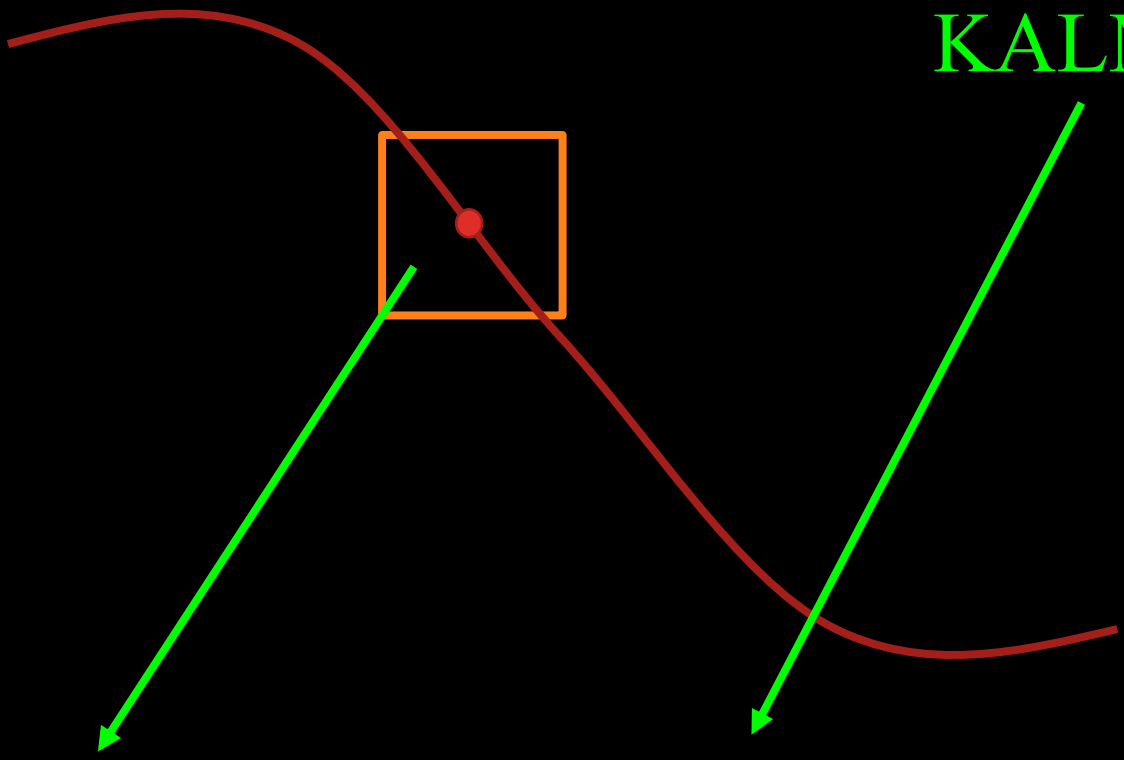
C4dynamics

DIFFERENT ALGORITHMS



C4dynamics

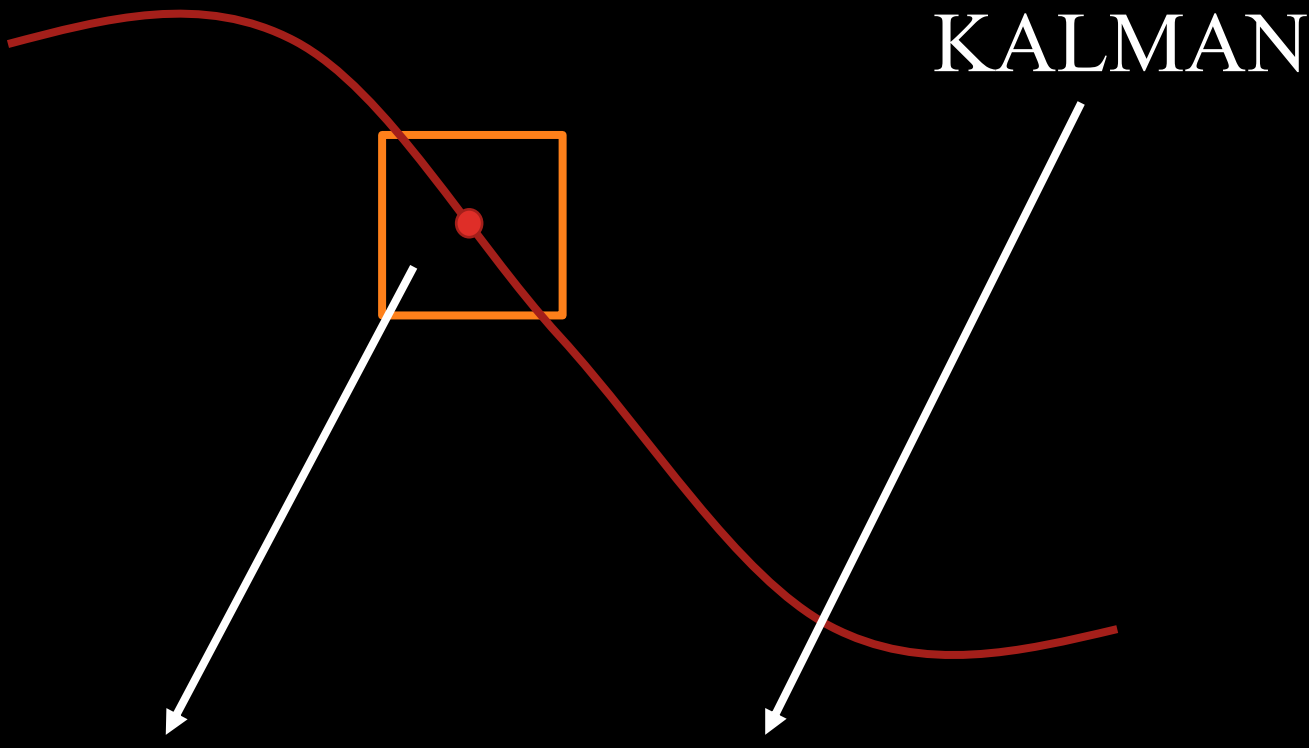
POINT'S POINT OF VIEW



data_point.update(my_algo, new_data)

C4dynamics

POINT'S POINT OF VIEW



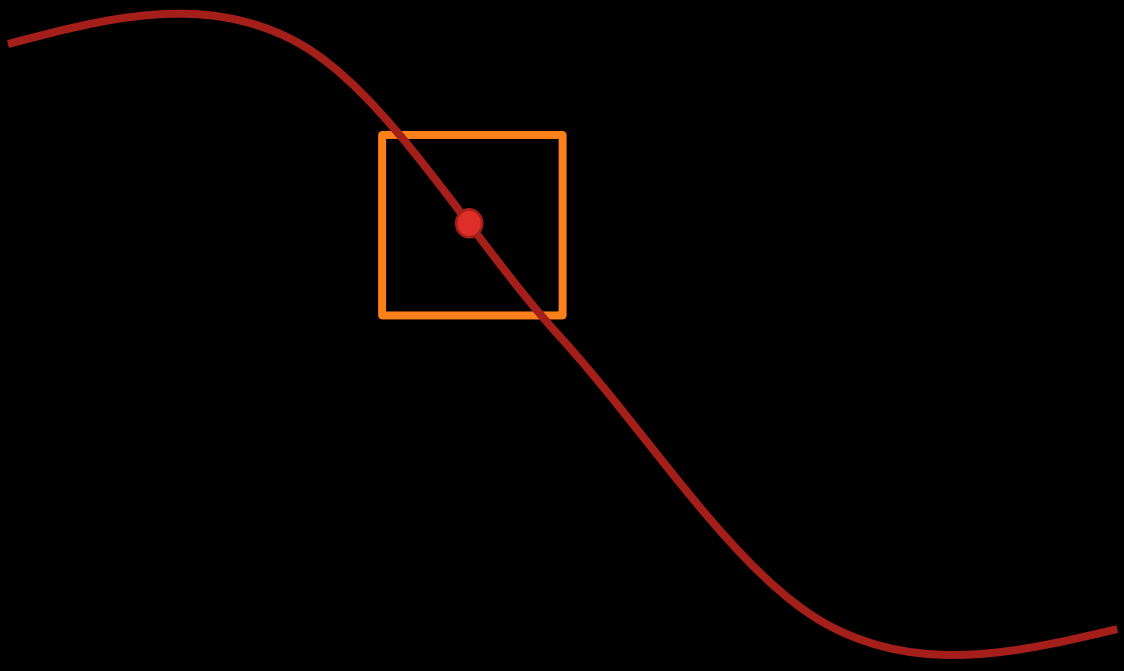
data_point.update(my_algo, new_data)

Binds the algorithm with the physical effect!

C4dynamics

So..

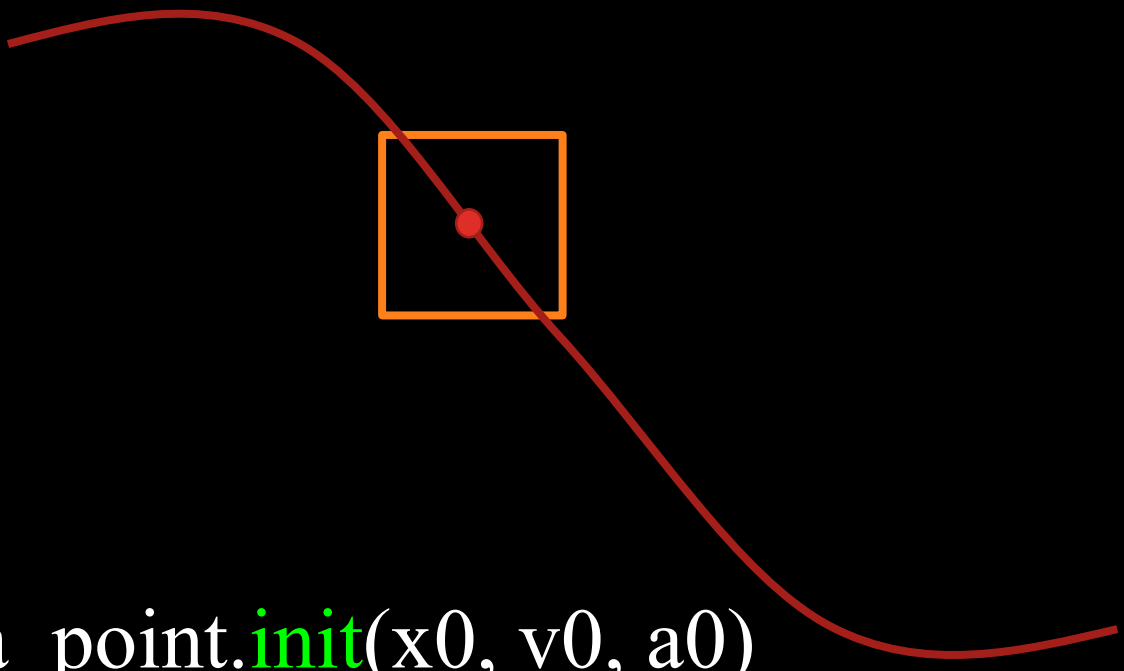
DO EVERYTHING FROM ONE PLACE!



C4dynamics

So..

DO EVERYTHING FROM ONE PLACE



```
data_point.init(x0, v0, a0)

new_data = my_sensor.measure()

data_point.update(my_algo, new_data)

data_point.plot(t0, tf)
```

Whatever your algorithm is, always look from the object's point of view.

This is Algorithm Context!

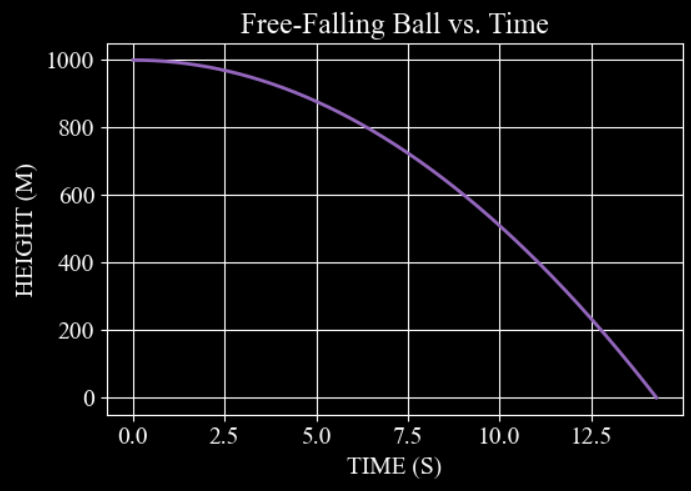
C4dynamics

Want to work with cool algorithm framework?

Download now C4dynamics and run freefall.py

Follow the instructions there:

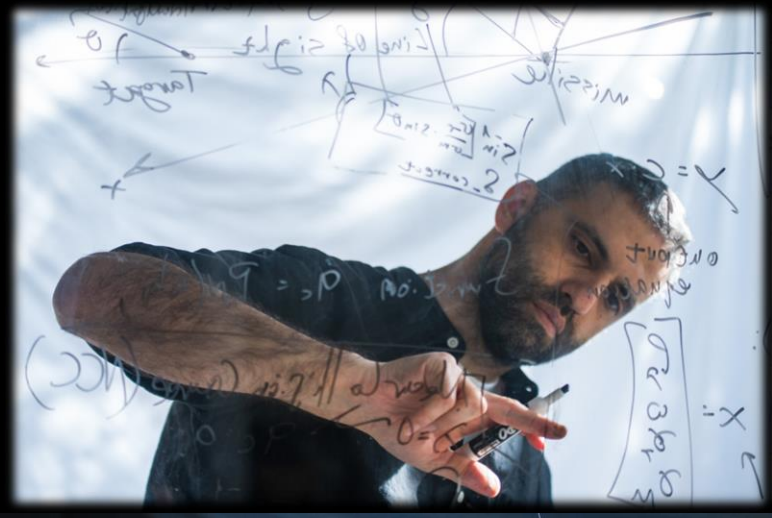
https://github.com/C4dynamics/C4dynamics/blob/main/examples/freefall.py

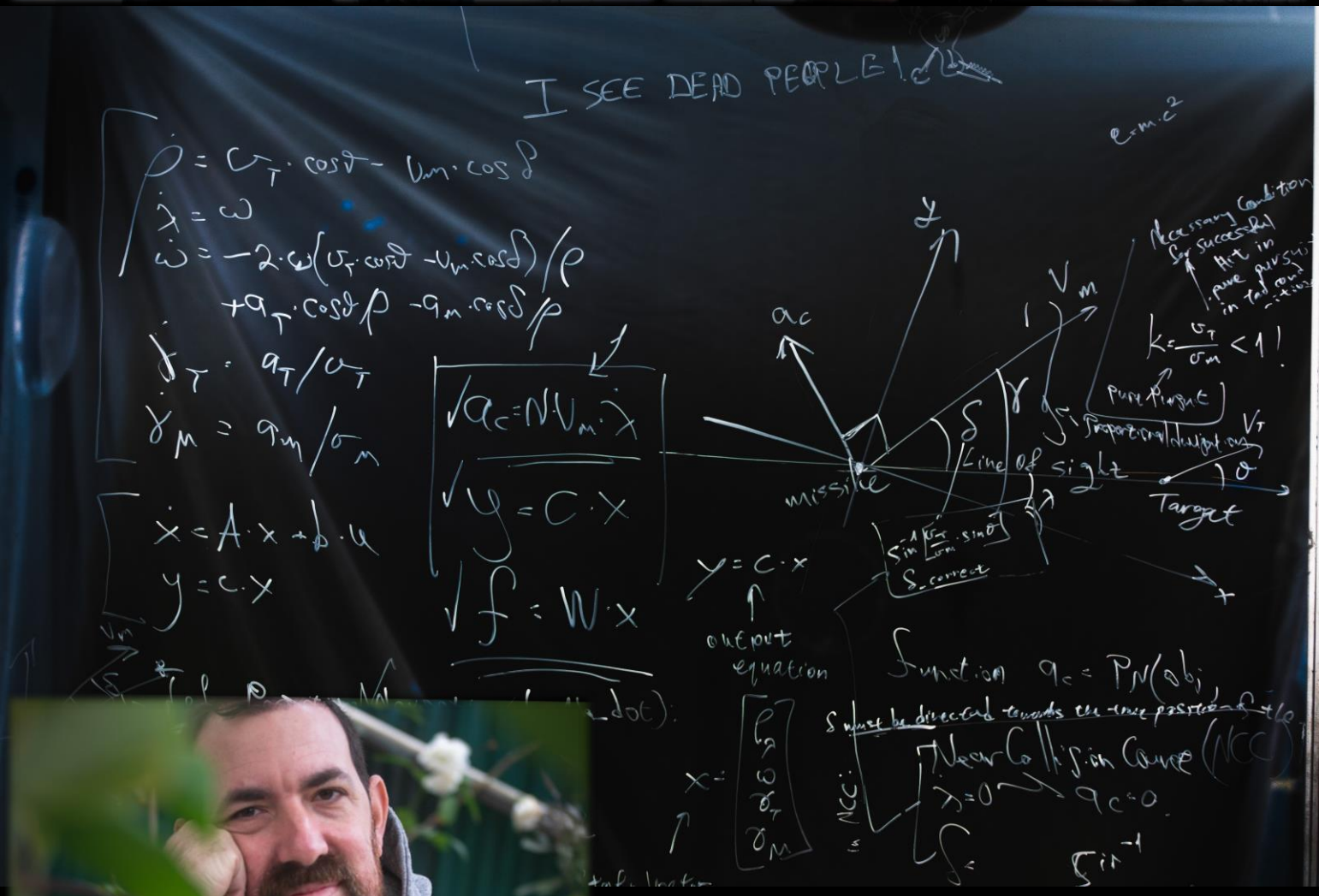


C4dynamics

A cutting-edge, high-standard algorithms development framework

C4dynamics





Gavriel Weinberger

C4dynamics