Train Kinematics Simulation with Real-World Data:
Trajectory Data Generation

Authors:

Andrea Vivai
Enea Dragoni

Supervisors:

Giorgio Buttazzo
Federico Nesti

Steps:

1. Used OpenStreetMaps (OSM) to extract railway tracks and maximum speed data.

2. Calculated the desired speed profile while including maximum acceleration and jerk constraints.

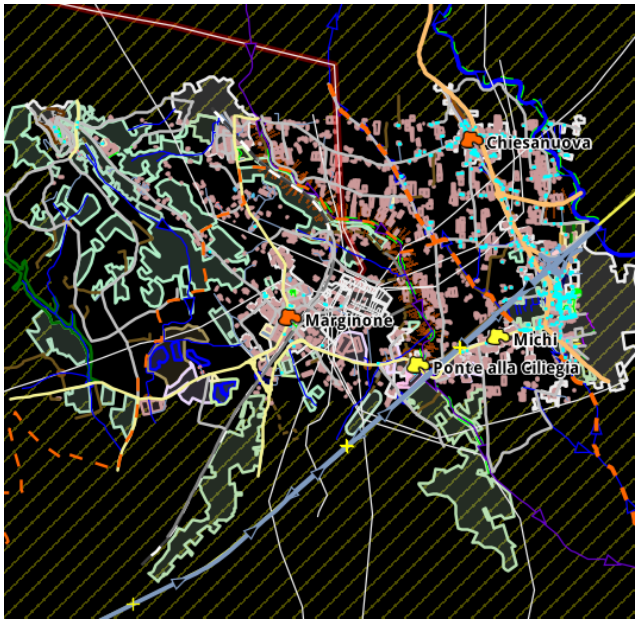
3. Modeled the train with a double carriage dynamic system, obtaining positions velocities, accelerations, orientation, and angular velocity ground-truth data.

4. Simulated IMU readings (accelerometer and gyroscope) based on the ground-truth values.

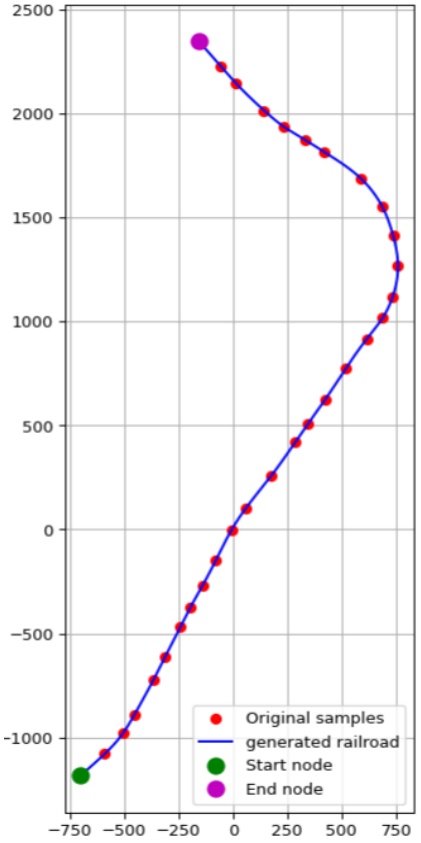
Railroad Generation:

Choosed the desired area

Parsed the corresponding XML file



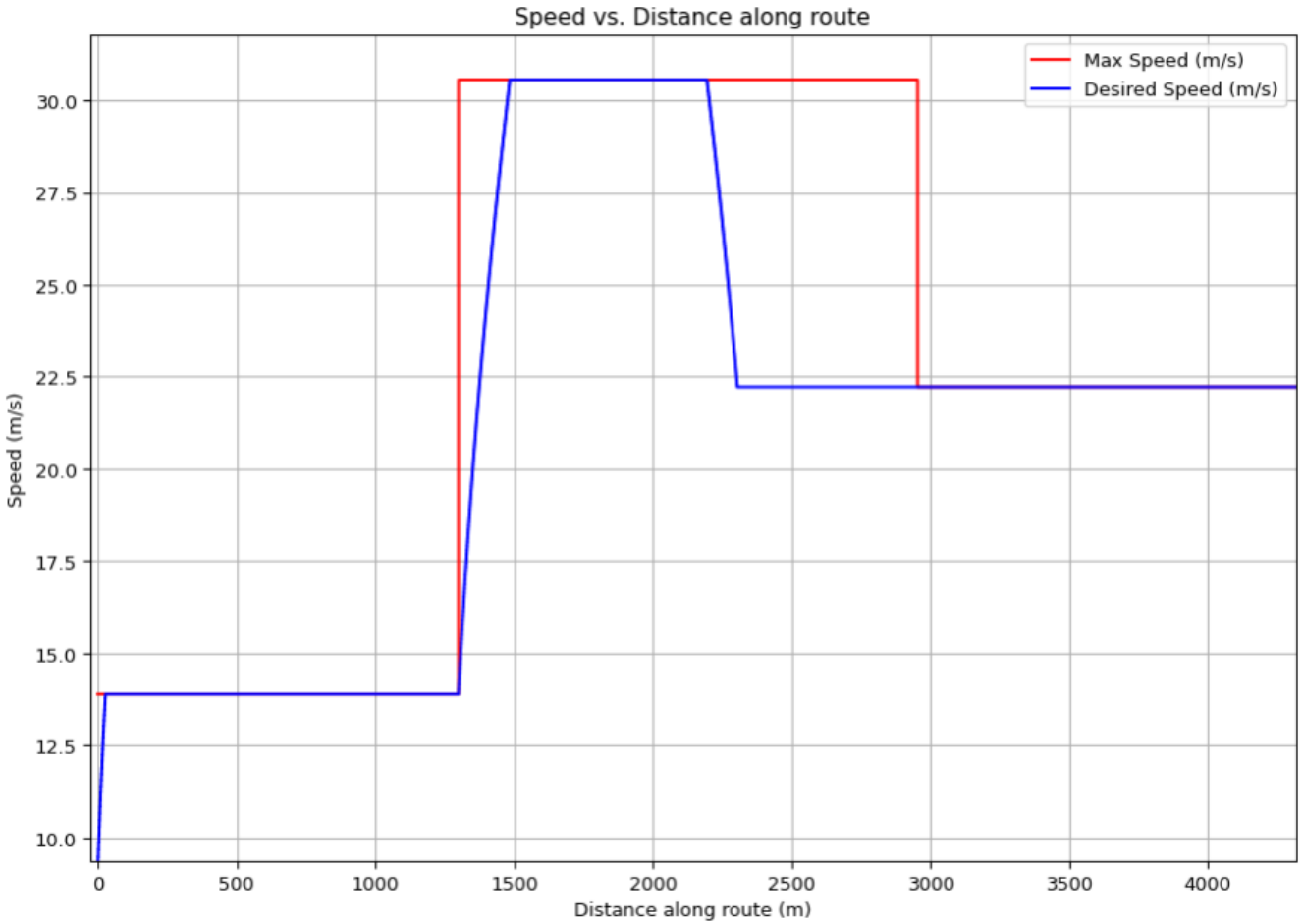
Generated the railroad



https://www.openstreetmap.org.

OpenStreetMap

Desired speed setpoint:



Train constraints:

$$|A_{max}| = 2m/s^2$$

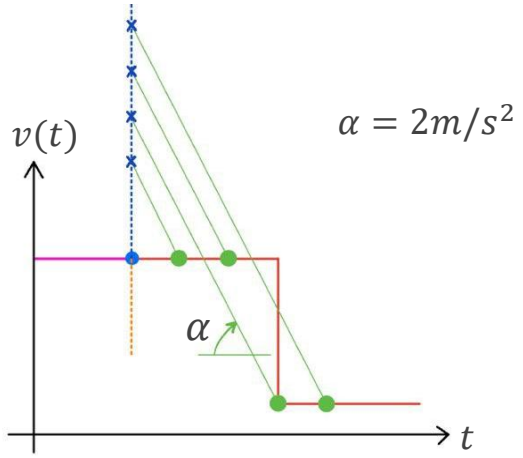
$$|J_{max}| = 1m/s^3$$

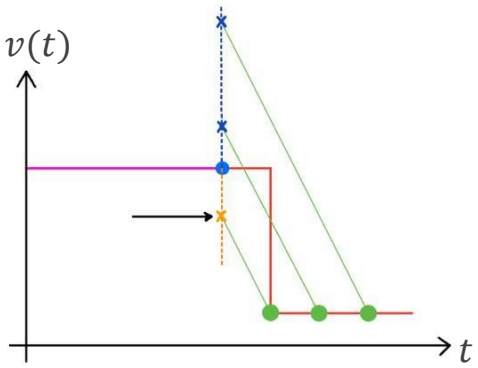
In the real world, train dynamics follow a desired speed profile created by a human, not the maximum speed profile. The profile generator also provides safety margins by decelerating automatically beforehand to compensate for any train delays.

How to know when to decelerate?

There are many ideas, to see the full explanation see the documentation in the notebook.

As example, consider these 2 situations:





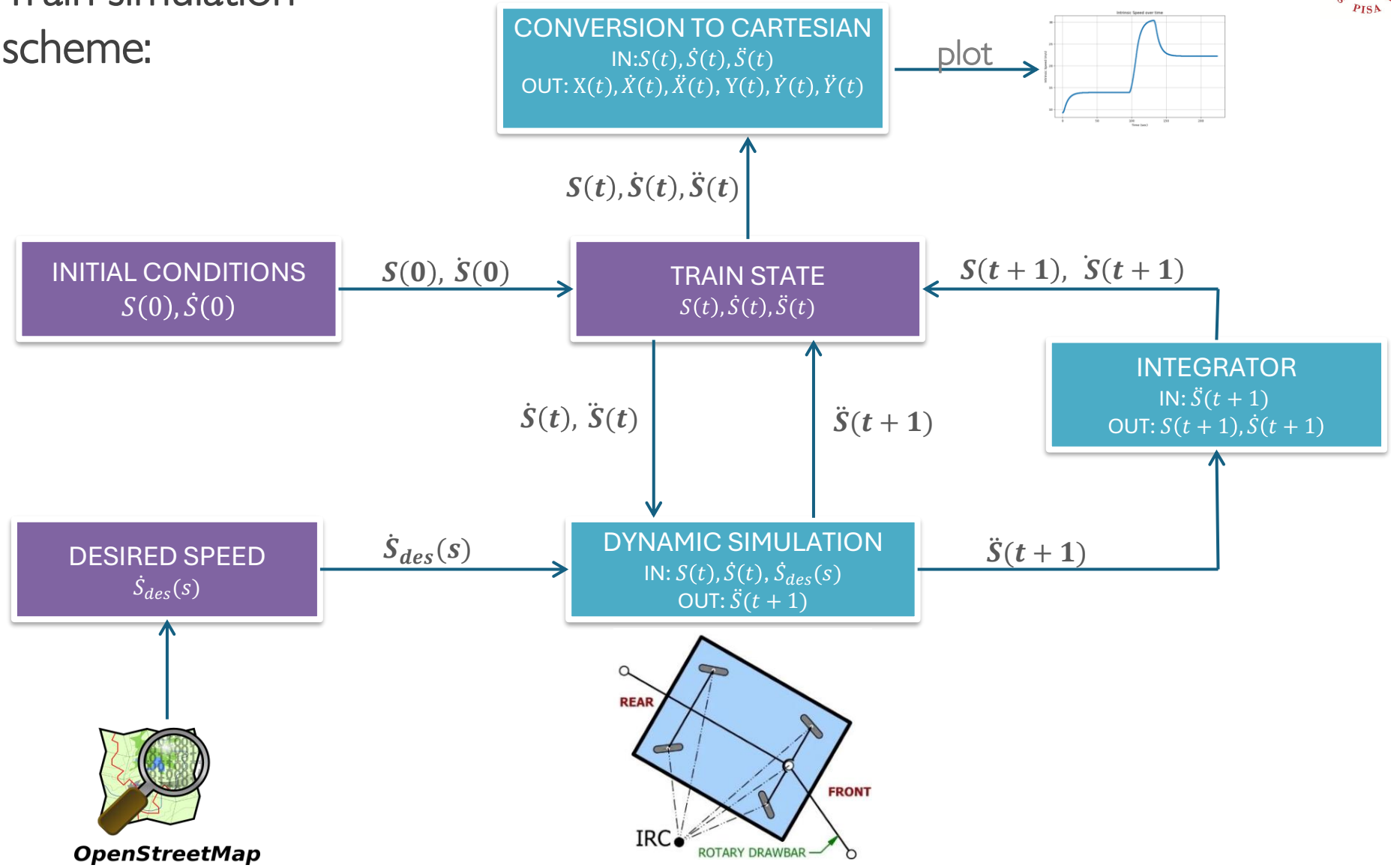
Calculate the intersection (x, x) between the vertical line passing through the current maximum speed and the line with a deceleration of 2 m/s^2 passing through the considered speed point.

Set the current speed limit to be the minimum between the current maximum speed and the intersection point with the minimum value (in this specific case, the current speed coincide with the current maximum speed).

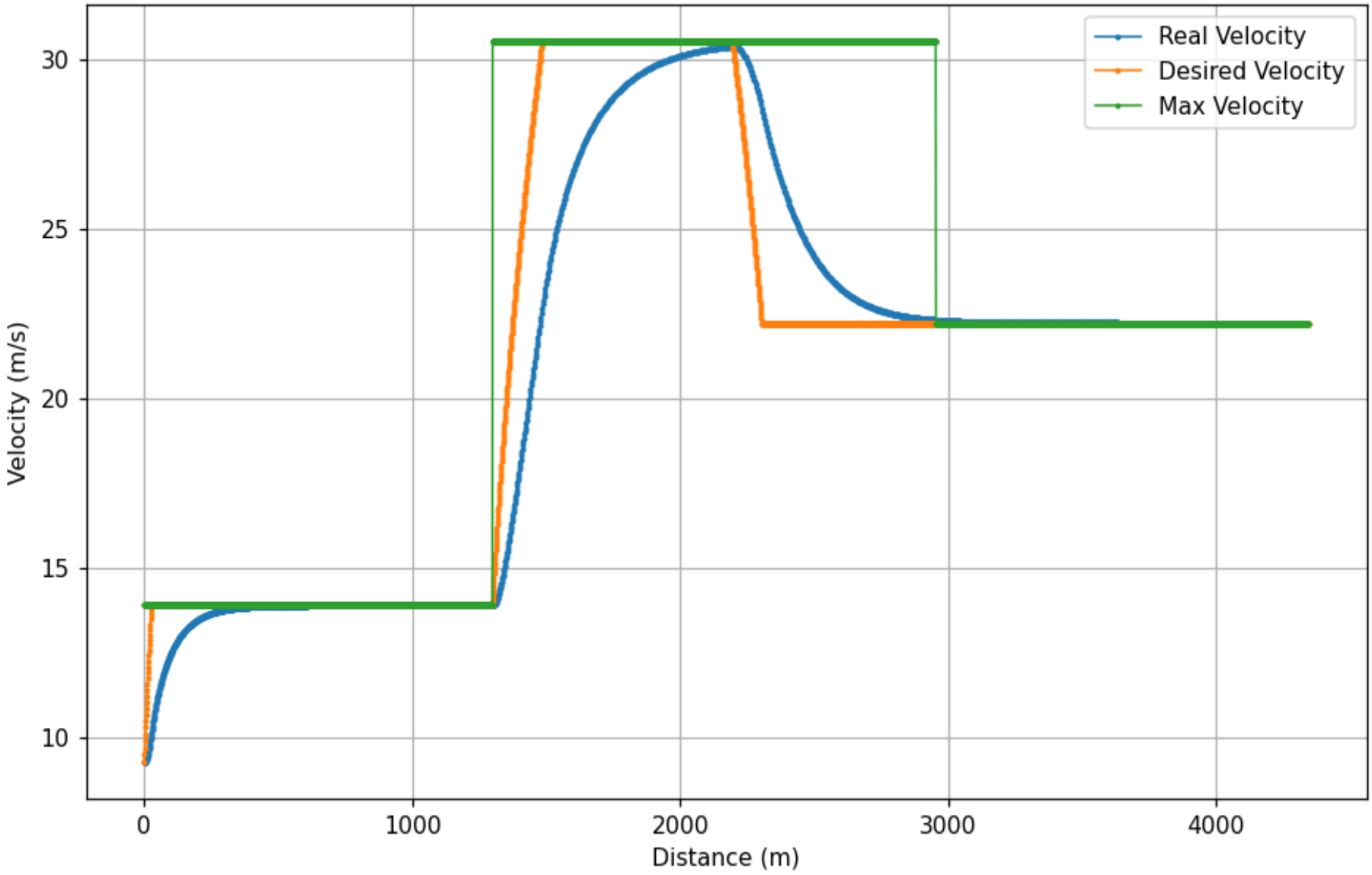
If the current speed limit is less than the current speed, the velocity will be decreased. The check is repeated for many points forward

- PAST MAX SPEED
- CURRENT MAX SPEED
- FUTURE MAX SPEED
- FUTURE SPEED CHECK
- NEED TO DECELERATE
- DON'T NEED TO DECELERATE

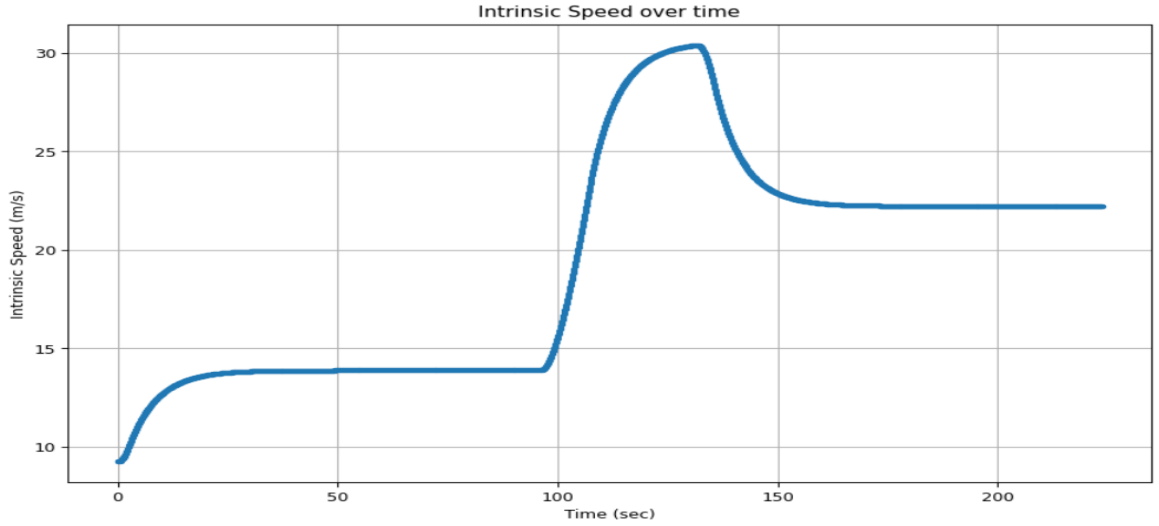
Train simulation scheme:

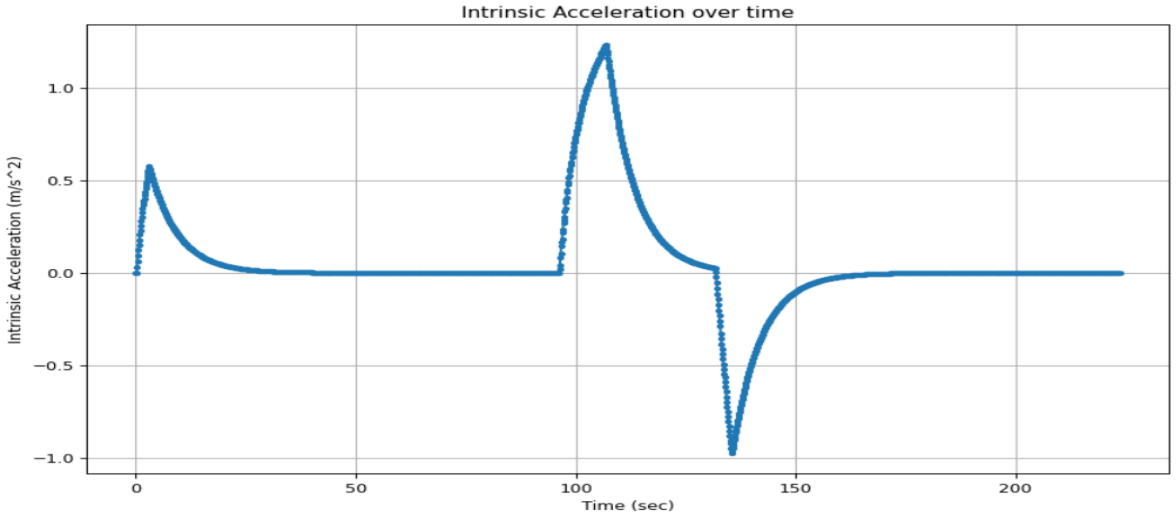


Dynamic Behaviour of the Train:

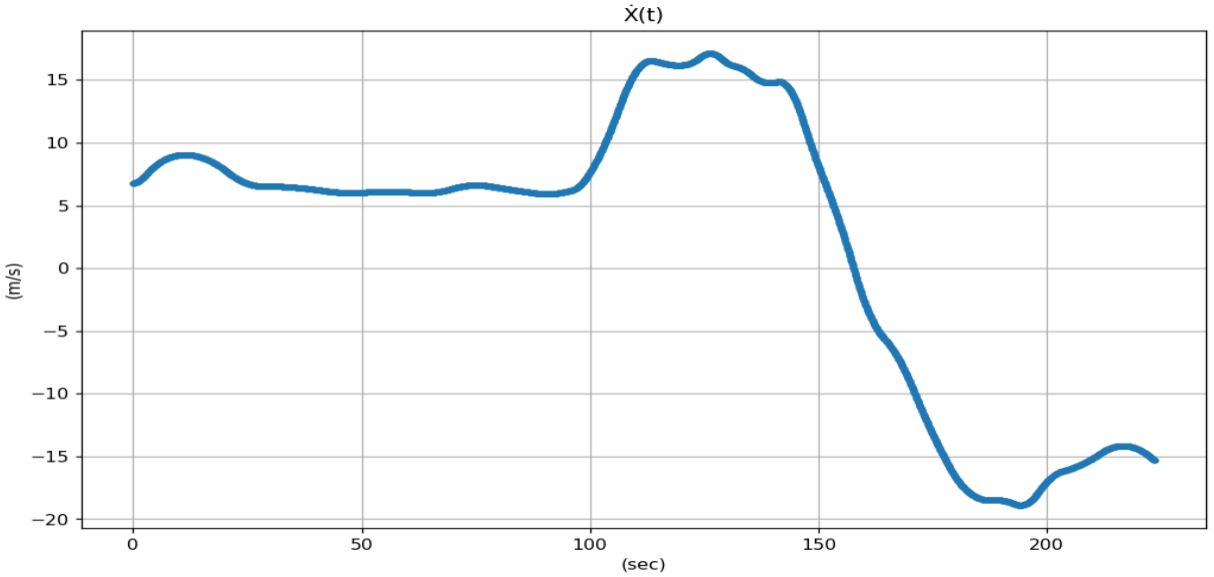


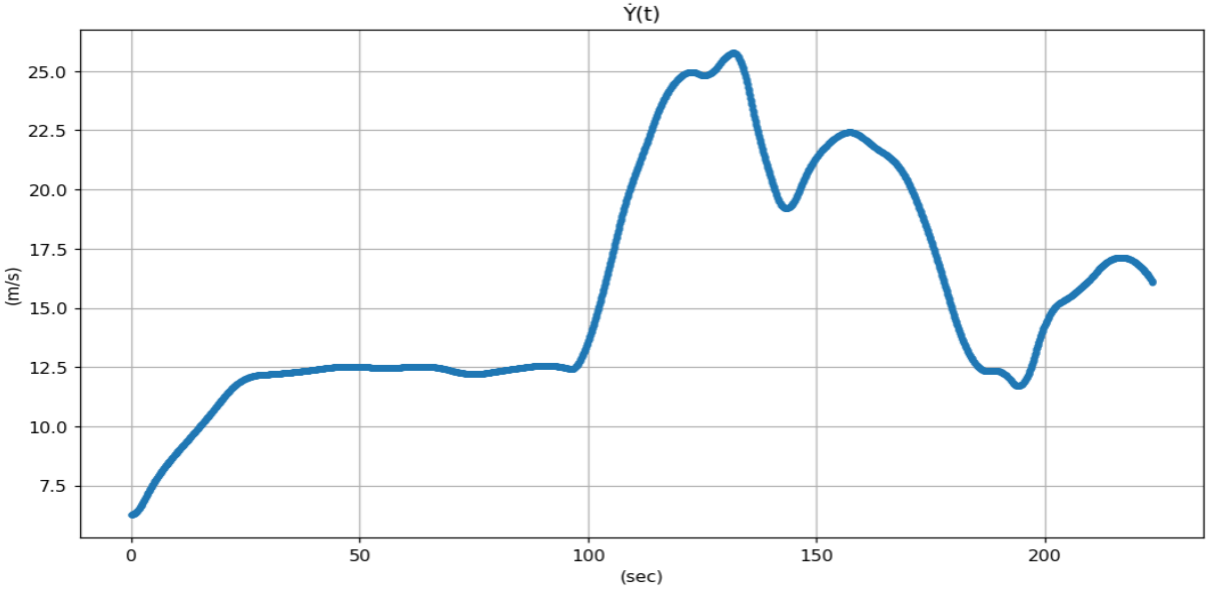
Intrinsic Data - Front Bogie





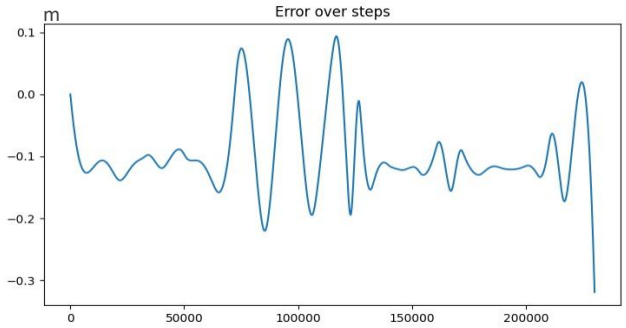
Cartesian Data - Front Bogie





How to model the back bogie kinematics?

One simple way would be to assume the same speed module for both the front and the back bogie, but this would cause errors during curves, as shown here:

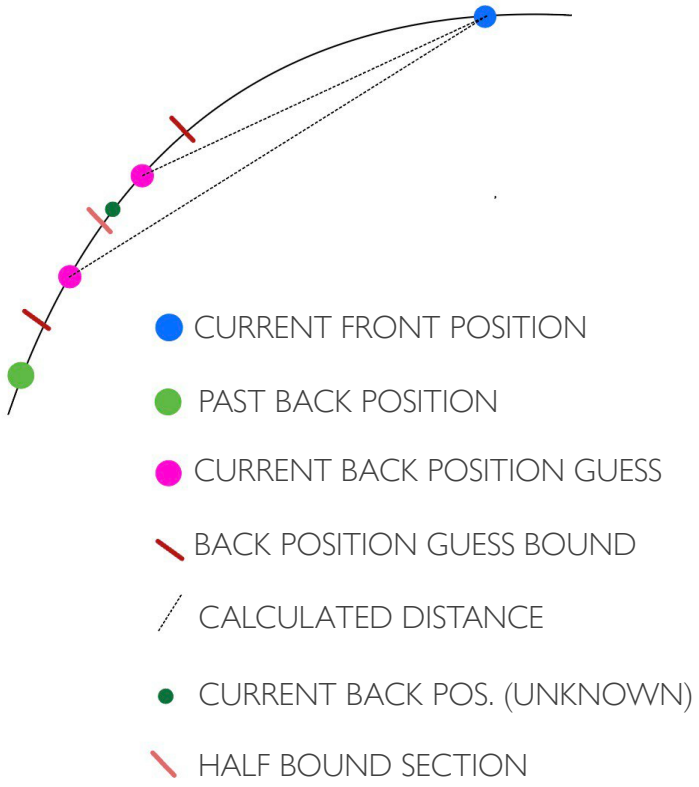


To eliminate this error, the following algorithm was implemented:

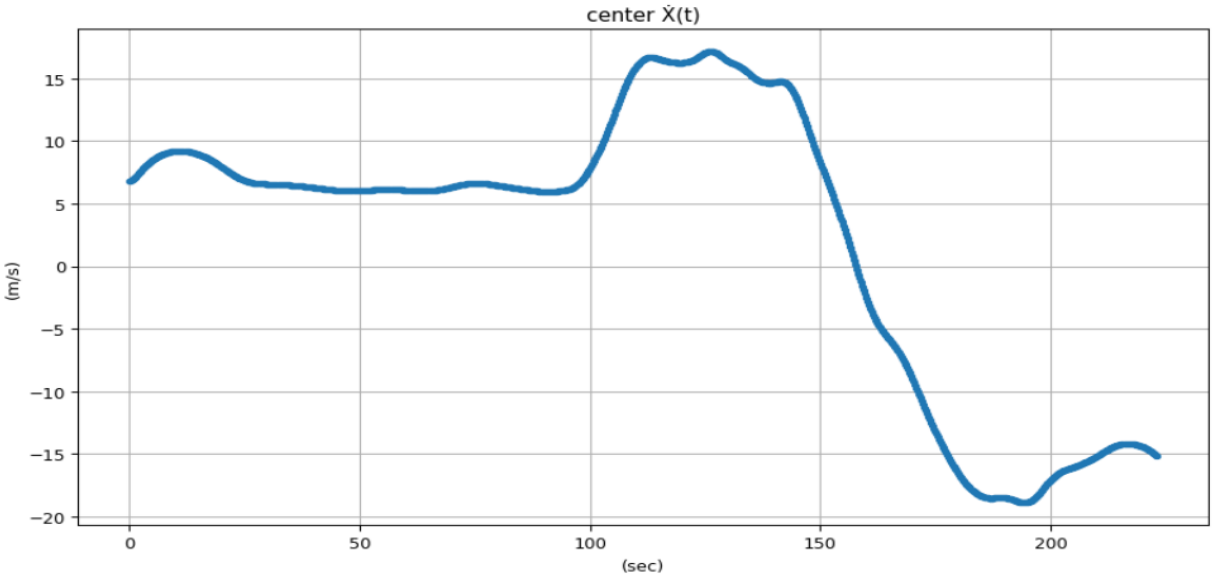
- Starting from the current position of the front bogie and the past position of the back bogie, we assume an advancement upper and lower bound for the back bogie, defining a railroad section.

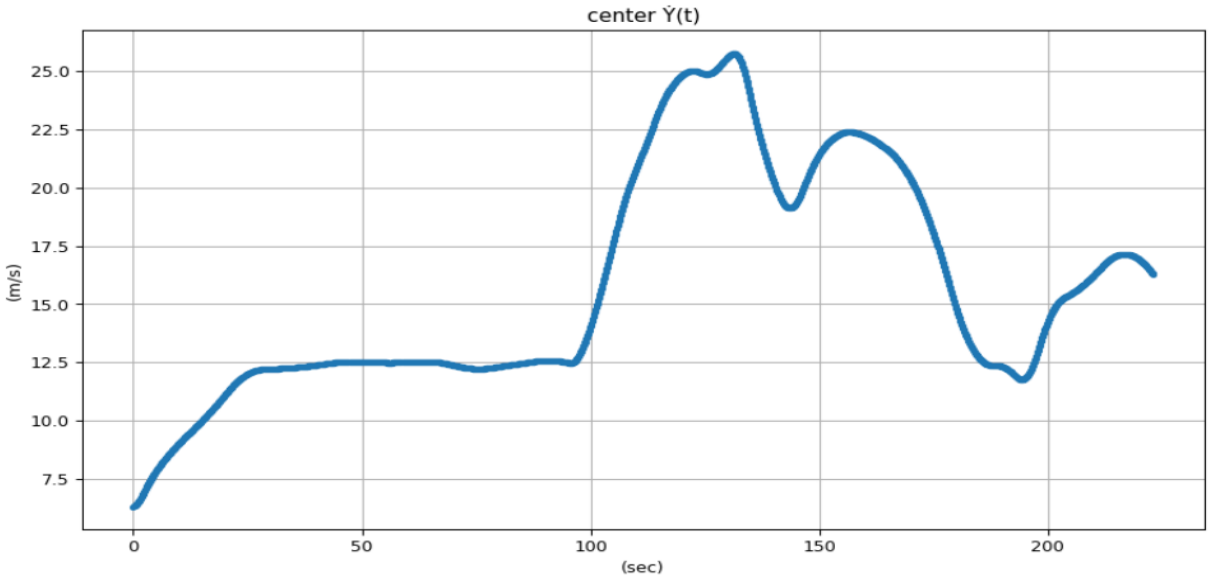
- The chosen railroad section is iteratively halved, and the distance between the centroid of each half and the front bogie position is calculated: the distance closer to the real cart length is chosen to be halved again until a stop criterion is met, and the centroid of the current section is returned as the back bogie position.

- The algorithm is then repeated with a new front bogie position.



Cartesian Data - Cart Center





Train Animations