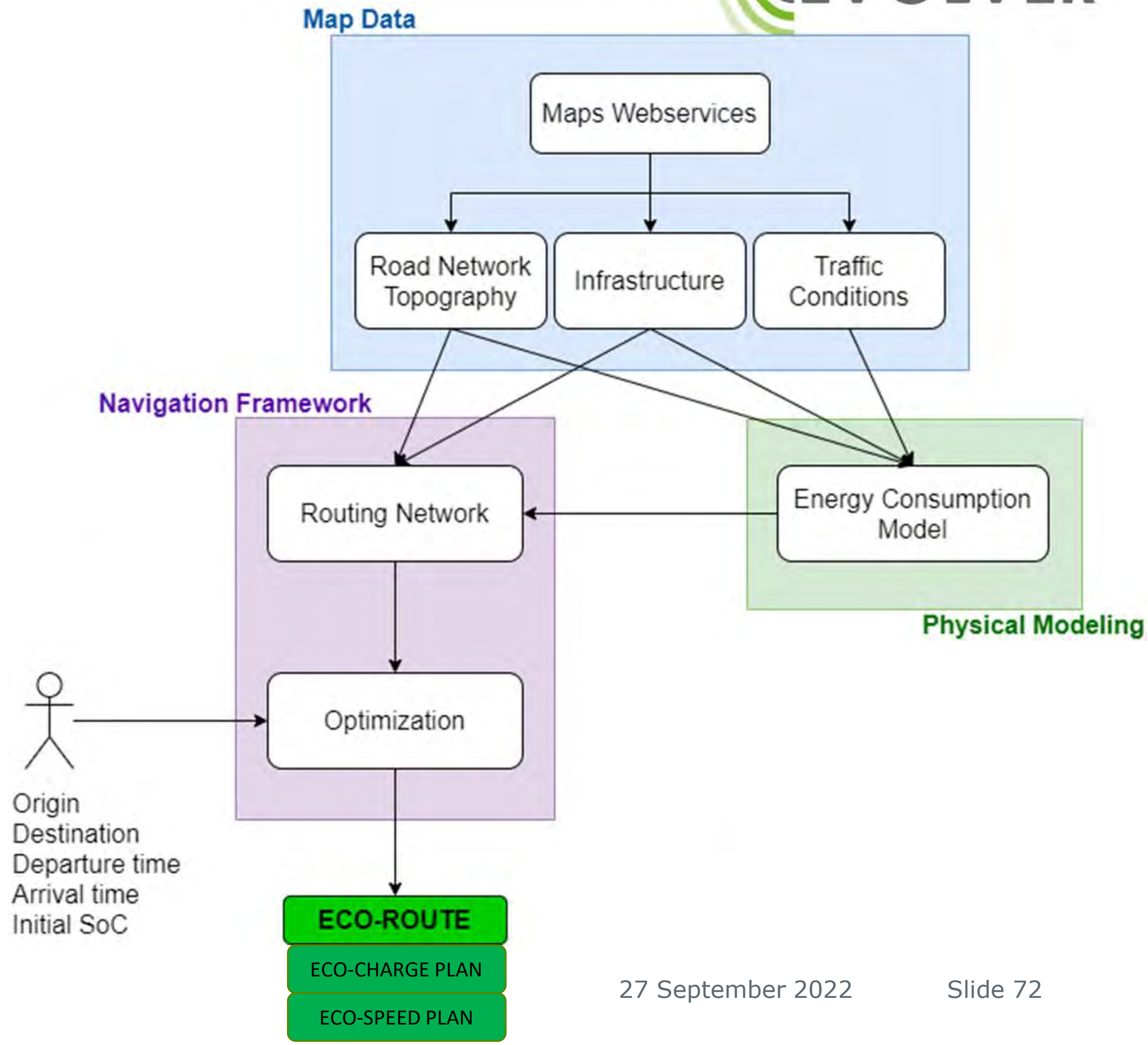
CEVOLVER

ECO-CHARGING

For a given origin and destination, finds the optimal route, charging plan, and average speed plan

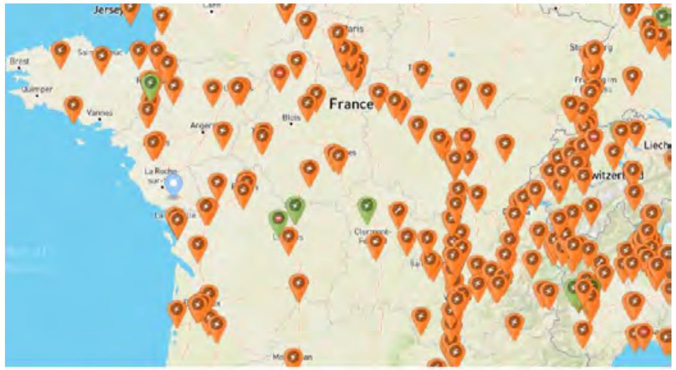
ECO-CHARGING Concept

- Map data
  - Road network topography (connectivity, altitude, etc.)
  - Infrastructure (signalization, charging stations, etc.)
  - Real-time traffic conditions
- Physical Modeling
  - Consider different powertrains
  - Predict energy consumption on each road segment
  - Predict travel and charging time
- Navigation framework
  - Model the road network as a directed graph
  - Find the optimal routing solution
  - Account for user's constraints



ECO-CHARGING Data

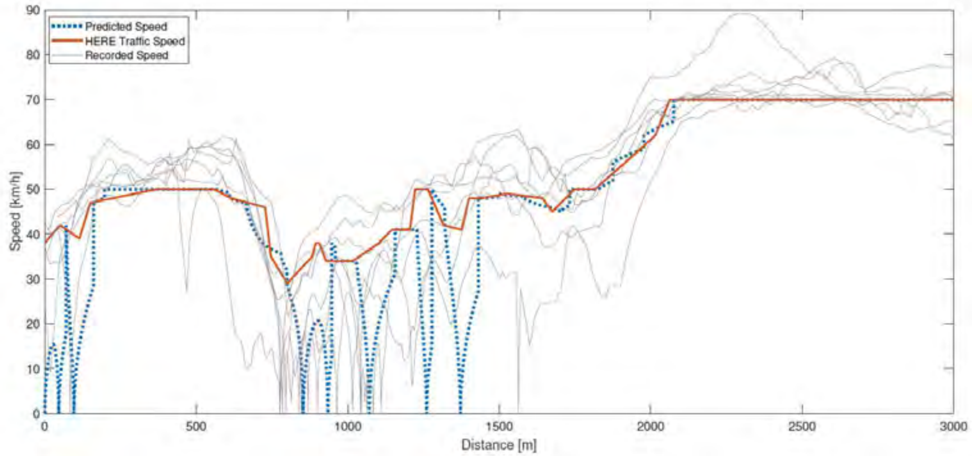
- Road topography (offline)
  - Length
  - Importance class
  - Geometry (lat, long, altitude)
  - Signalization
- Charging infrastructure (offline)
  - Location (lat, long)
  - Charger power (DC, ≥ 50kW)
- Traffic data (online)
  - Avg. travel time
  - Avg. traffic speed
- Ambient data (online)
  - Temperature
  - Weather conditions

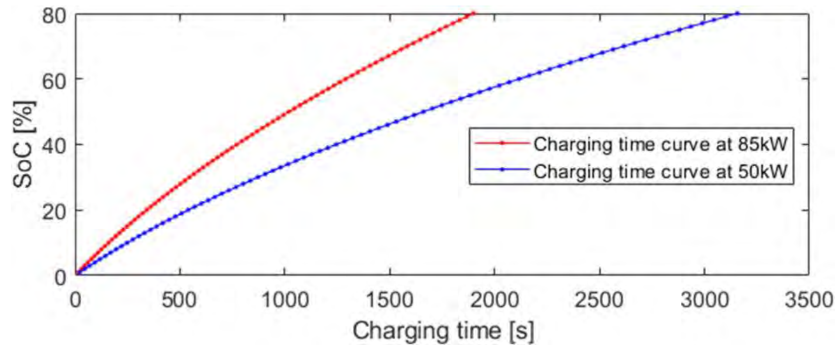


Source: https://openchargemap.org/site

ECO-CHARGING Model

- Each graph link i is attributed an energy weight (e_i) and a time weight (t_i)
- A « synthetic speed profile » is constructed that respects the average speed (v_i) chosen and the traffic speed ($TRAF$), with speed transients expectedly induced by signalization ($INFR$)
- Estimation of link time:
  - Travel time depending on the speed profile
  - Recharge time depending on Δ_i if nonzero
- Estimation of the energy consumption per link:
  - A vehicle model (VEH) produces a load profile, based on the speed profile, the link slope ($ROUT$), the weather conditions ($WEAT$), then energy by integration over the link





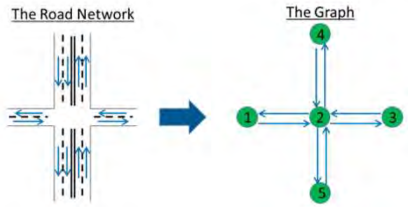
ECO-CHARGING Optimization Problem

Objective
- Minimize a weighted sum (weight = λ) of total travel time (t) and energy consumption (e)
- By choosing the route segments (i) taken ($x_i = 1$ if segment i is taken, $x_i = 0$ otherwise)
- Additionally choosing the average speed (v_i) and the amount of recharge (Δ_i) on each segment

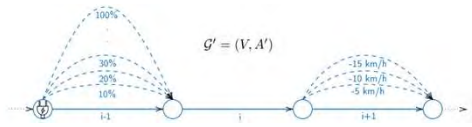
Constraints
- Topology: segments i must be connected, they must go from origin O to destination D
- Physics: amounts of recharge Δ_i are limited by the capacity of the battery; net energy consumption must never exceed battery capacity
- Speed limits: average speeds v_i must respect the speed limits on segments i

$$\min_{x_i} \sum_{i \in A_{\mathcal{L}}} [\lambda t_i(v_i, \Delta_i) + (1-\lambda) e_i(v_i, \Delta_i)] \cdot x_i$$

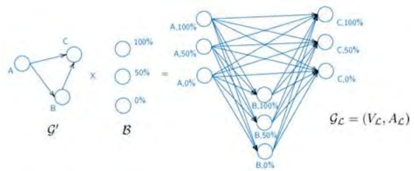
Directed graph



Multigraph Expansion



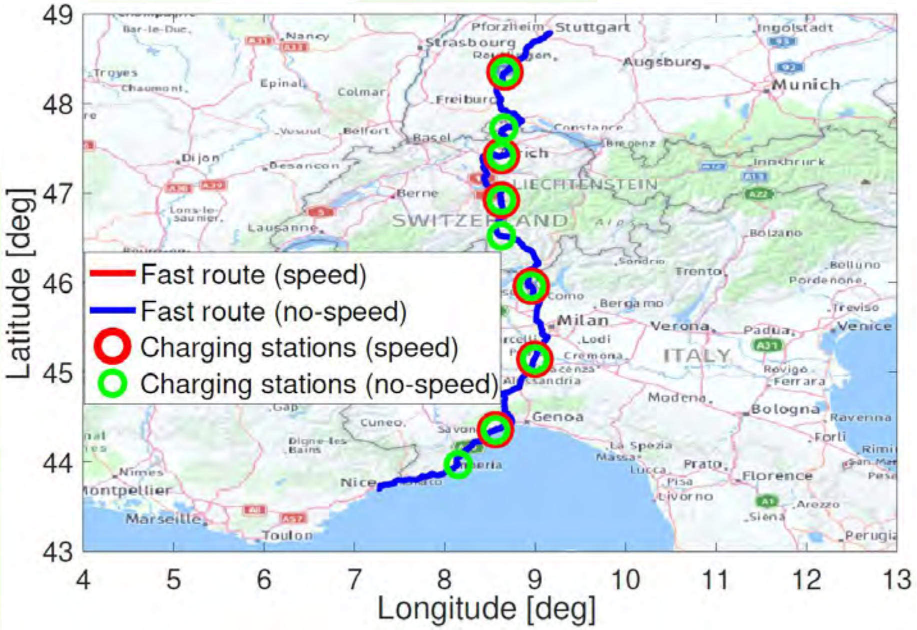
Lexicographic Product



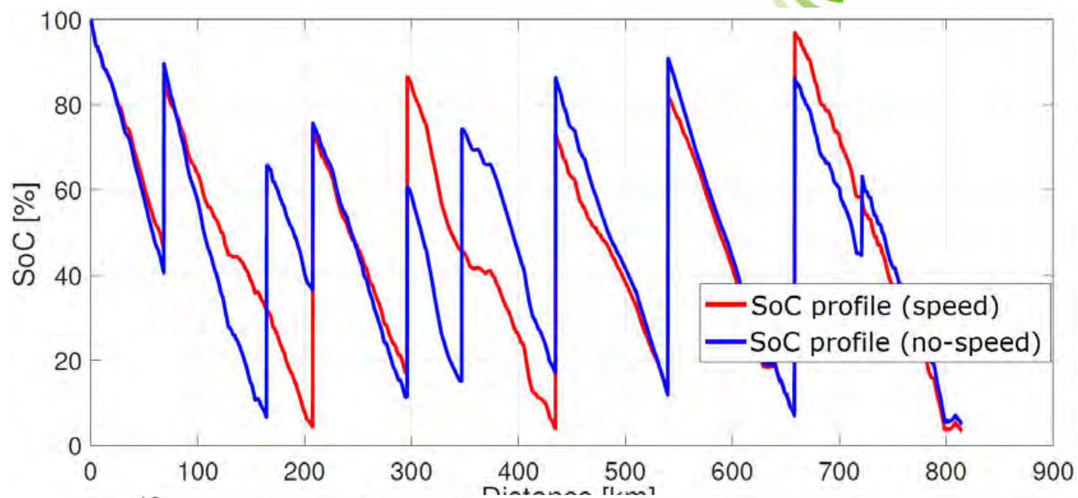
The constraints are embedded in the graph construction

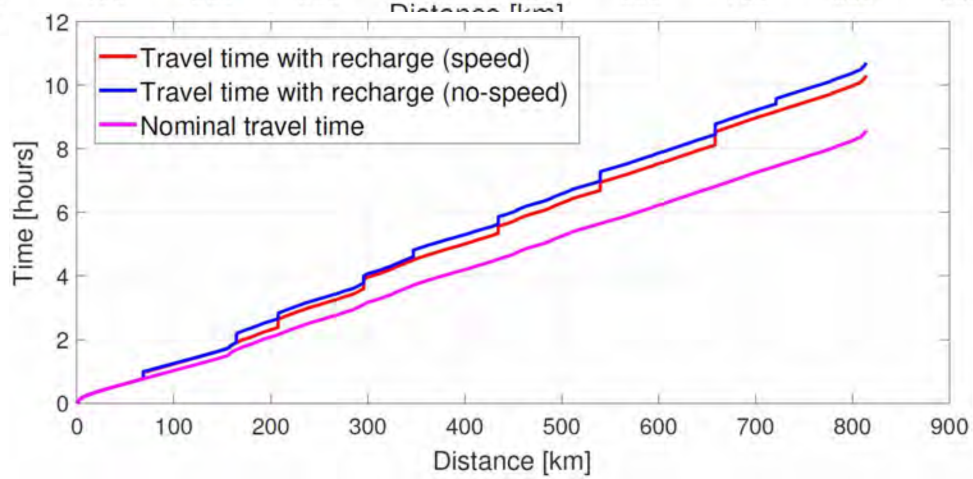
The solution is more accurate compared to other methods

ECO-CHARGING Results (1/2)

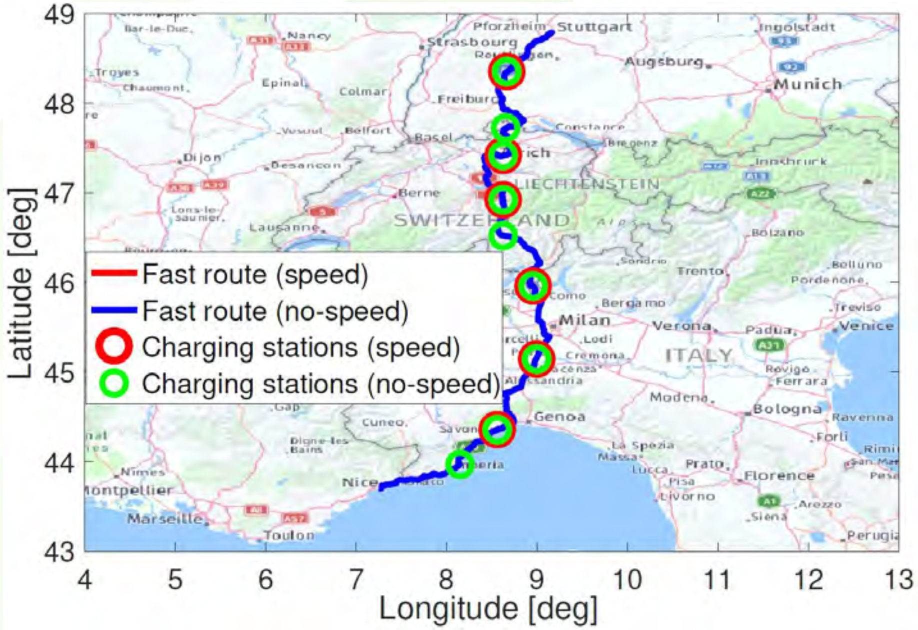


	No-speed	Speed
Graph size (arcs)	3 167 837	217 617 481
Avg. CPU Time [s]	0.7	104
Recharged energy [kWh]	153	123
Energy consumption [kWh]	181.5	152
Trip time [h]	10.7	10.3
Number of charging stops	9	6

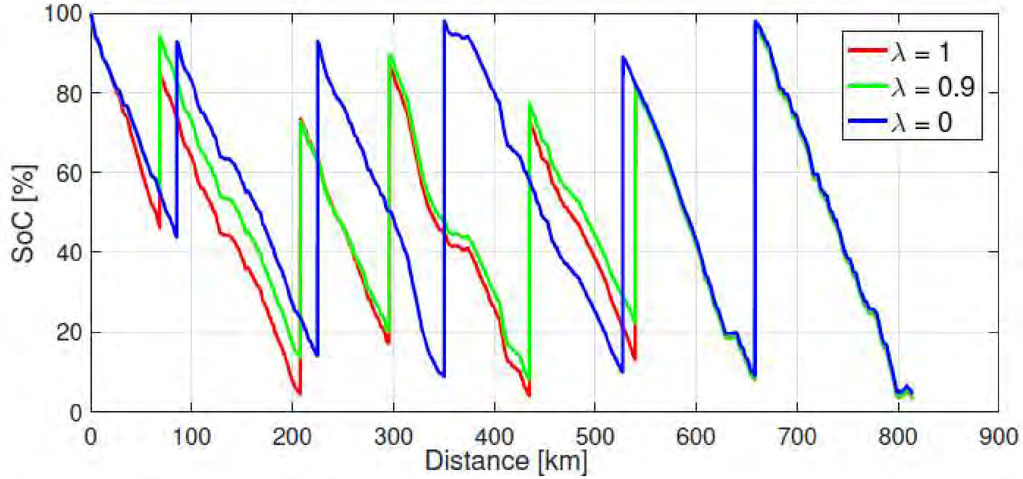


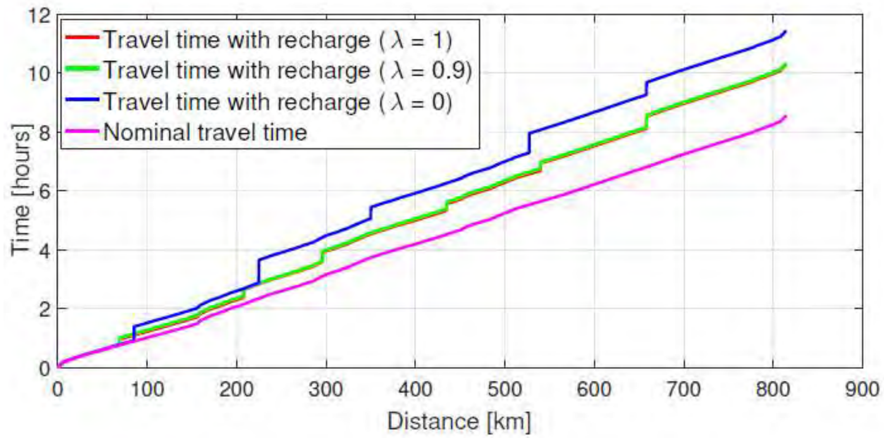


ECO-CHARGING Results (2/2)

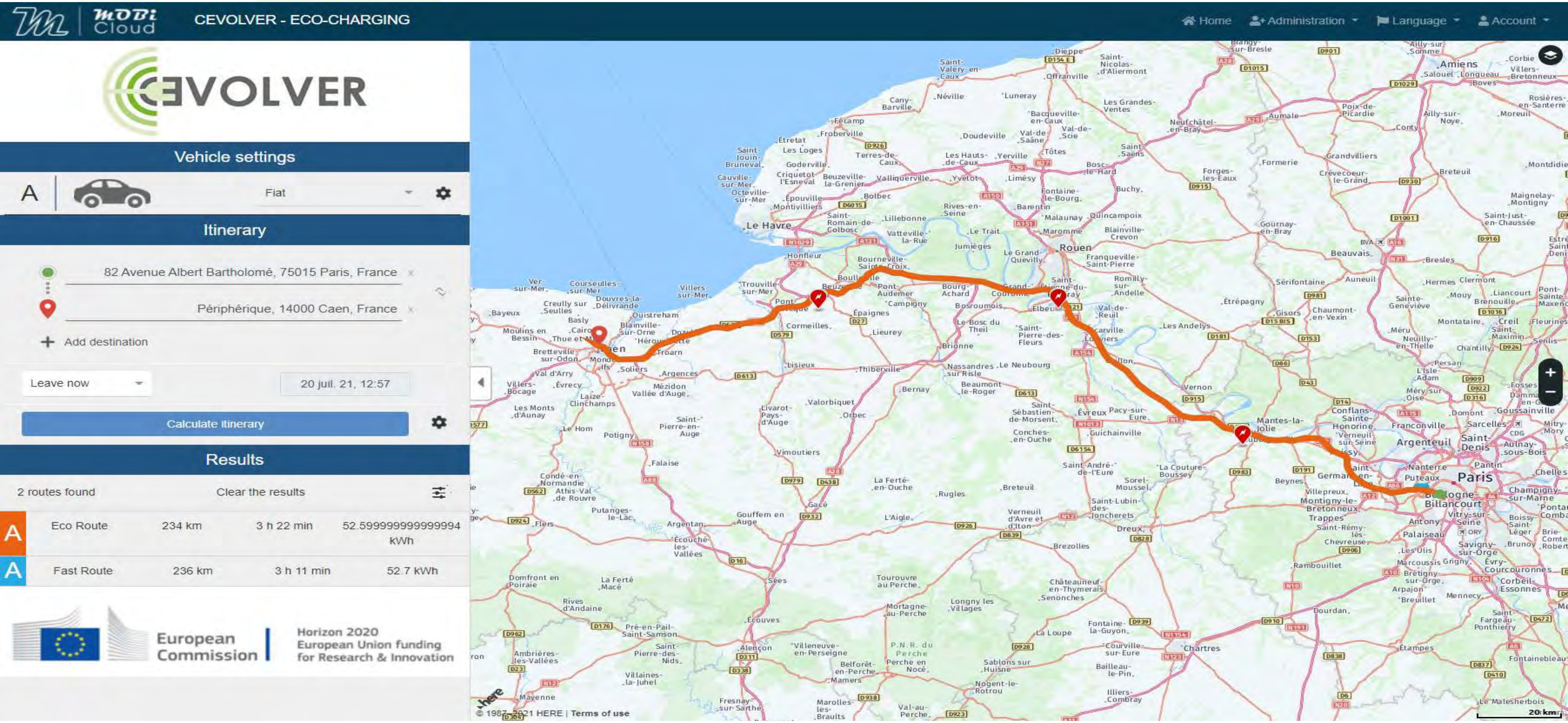


	$\lambda = 0$ min-ene	$\lambda = 0.9$	$\lambda = 1$ min-time
Recharged energy [kWh]	117	117	123
Energy consumption [kWh]	145.6	145.9	152
Trip time [h]	11.44	10.34	10.29
Number of charging stops	5	6	6





ECO-CHARGING Web-service

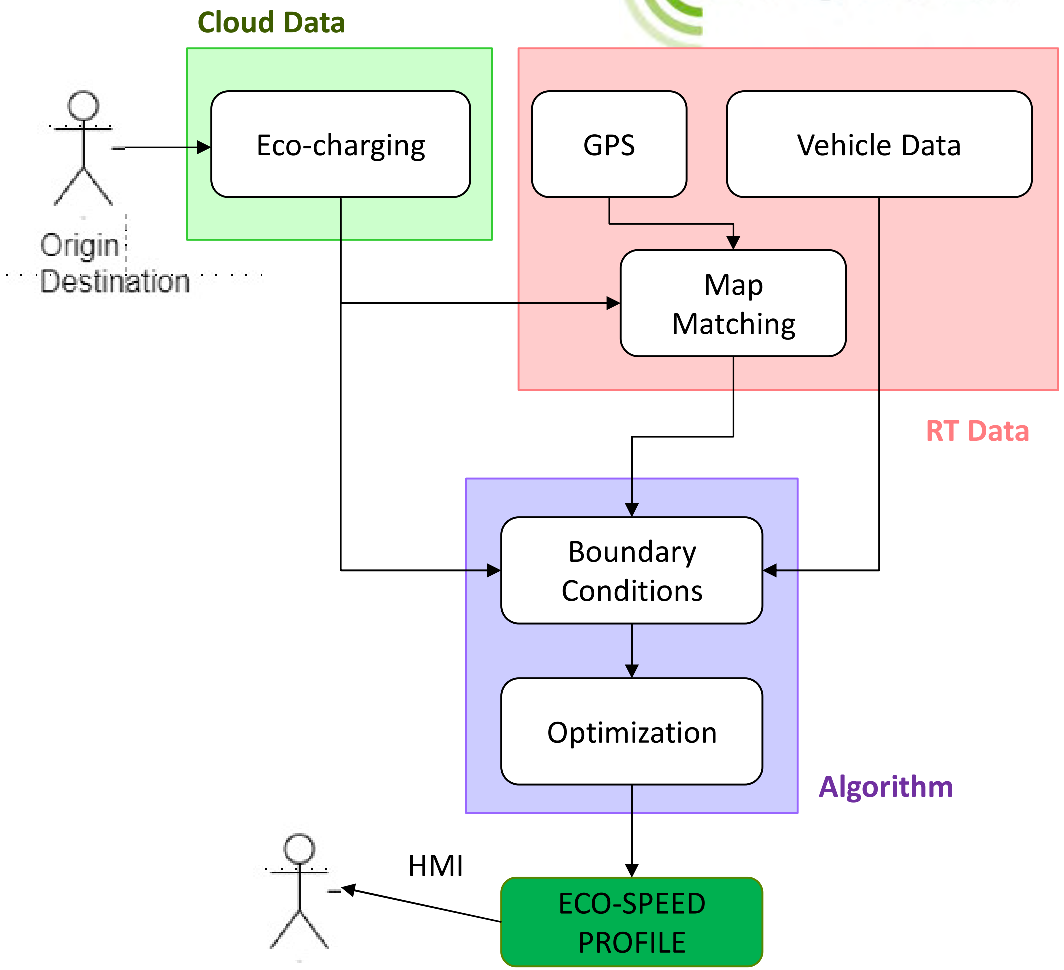


ECO-DRIVING

Along a given route, finds the optimal speed profiles to follow and advises the driver in real time

ECO-DRIVING Concept

- Cloud Data
  - Eco-Charging webservice (eco-route with all its attributes)
  - External webservices (traffic, weather)
- Real-Time Data
  - ADAS sensors (leading vehicle detection, traffic light) (optional)
  - In-vehicle sensors (SoC, speed, etc.)
  - GPS + Map Matching (current position, current segment on route)
- Algorithm
  - Update boundary conditions (shrinking horizon, remaining distance and time, target speed)
  - Update constraints (on speed, position)
  - Find the optimal speed profile



ECO-DRIVING Optimization Problem

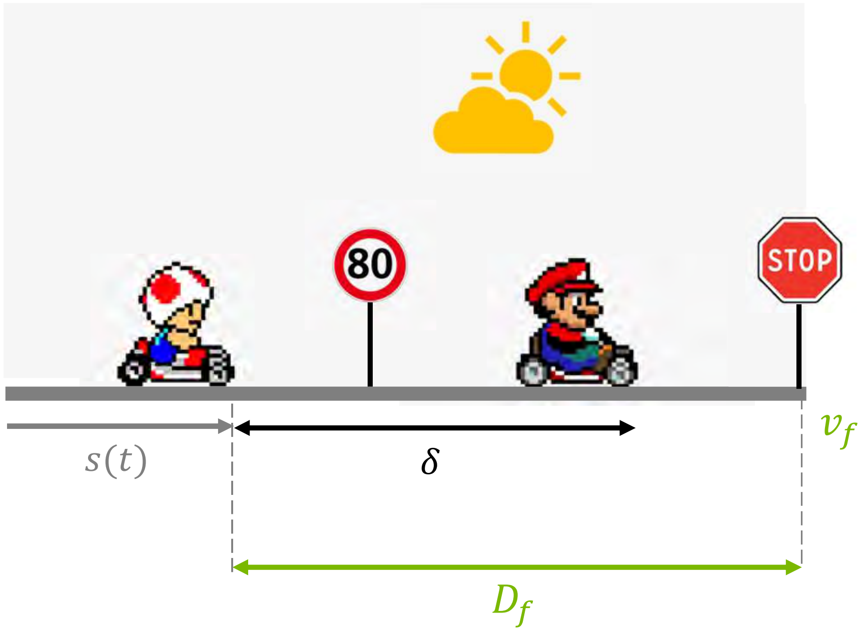
Objective

- Minimize the cumulated battery energy consumption (P_b) over an horizon
- Defined as the remaining portion of the current road segment, i.e., a given distance (D_f) and time (T_f)
- By acting on the instantaneous acceleration ($u(t)$)

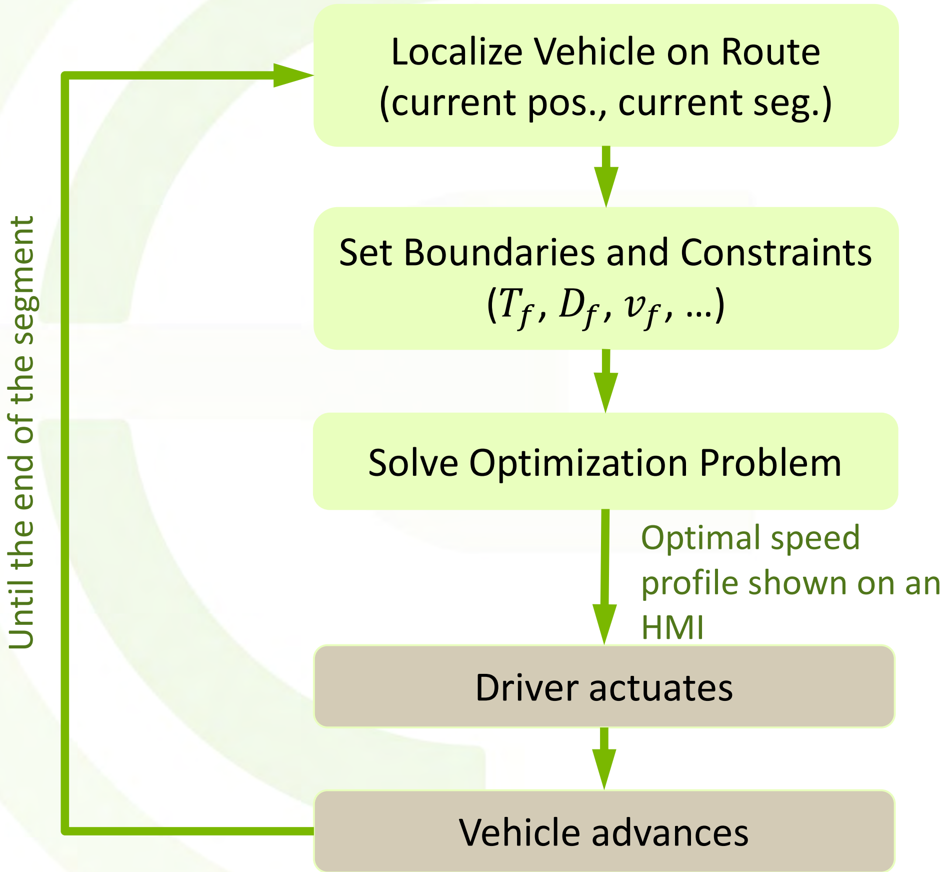
Constraints and Boundary Conditions

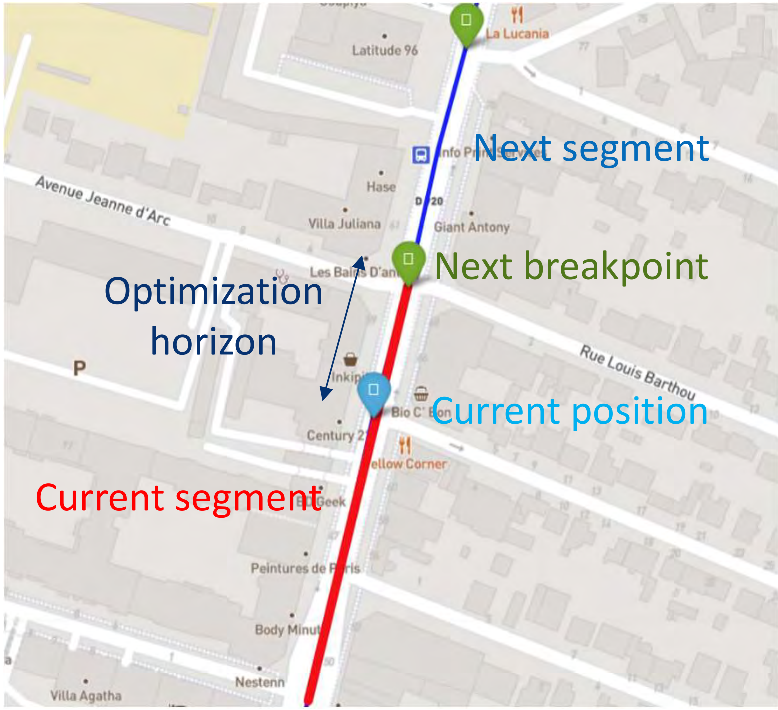
- Physics: Vehicle dynamics (position $s(t)$, speed $v(t)$)
- Drivability: Accelerations and decelerations are limited by design
- Terminal: Initial speed (v_i), end-of-segment speed (v_f)
- State: speed is limited by max speed, gap to preceding vehicle (δ) must be larger than a safe gap

$$\min_{u(t)} \int_0^{T_f} P_b(u,v)dt$$



ECO-DRIVING Algorithm



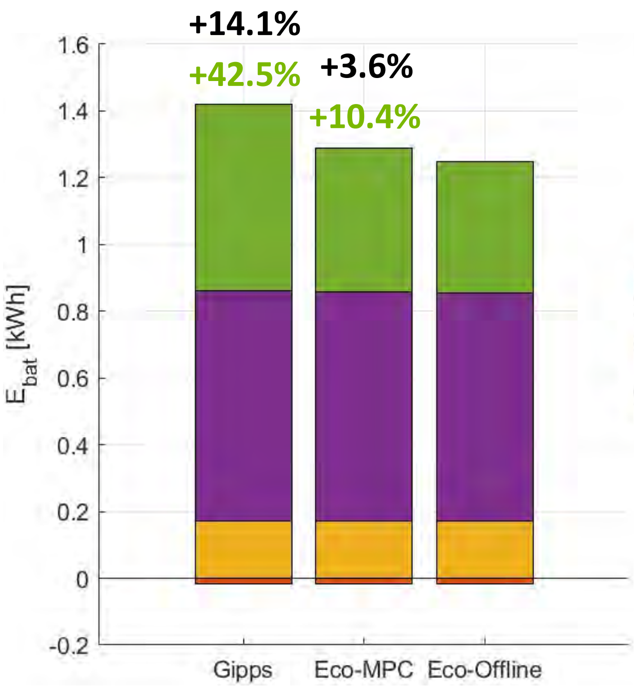


ECO-DRIVING Results (1/3)

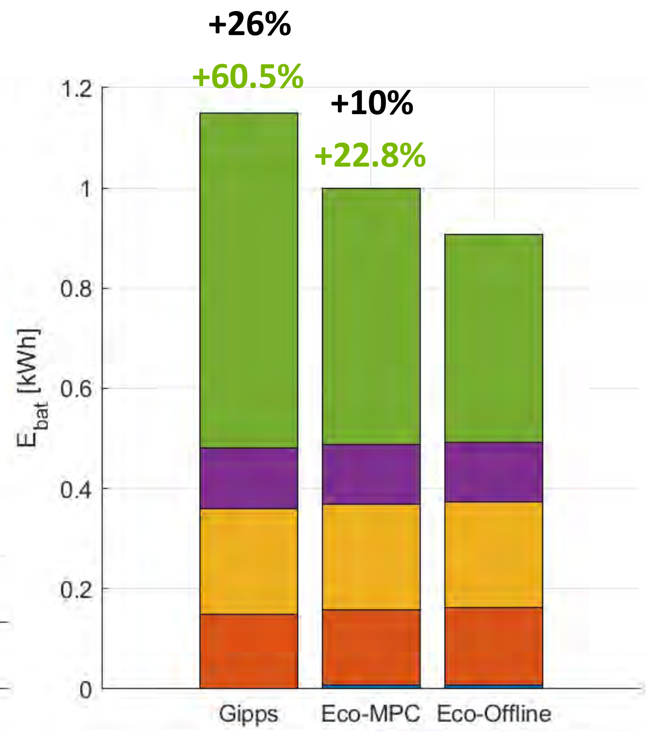
TWO TRIP SIMULATIONS

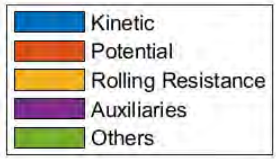
- Battery energy reduction achieved by the simulated eco-driving is of 9% and 13% compared to a standard driver model « Gipps »
- When considering only effective part (excluding noncompressible energies), reduction is of 22-23%
- Eco-driving is also reasonably close to an offline optimization (« eco-offline »), done using interior-point optimization (4% and 10%)

Trip 1



Trip 2

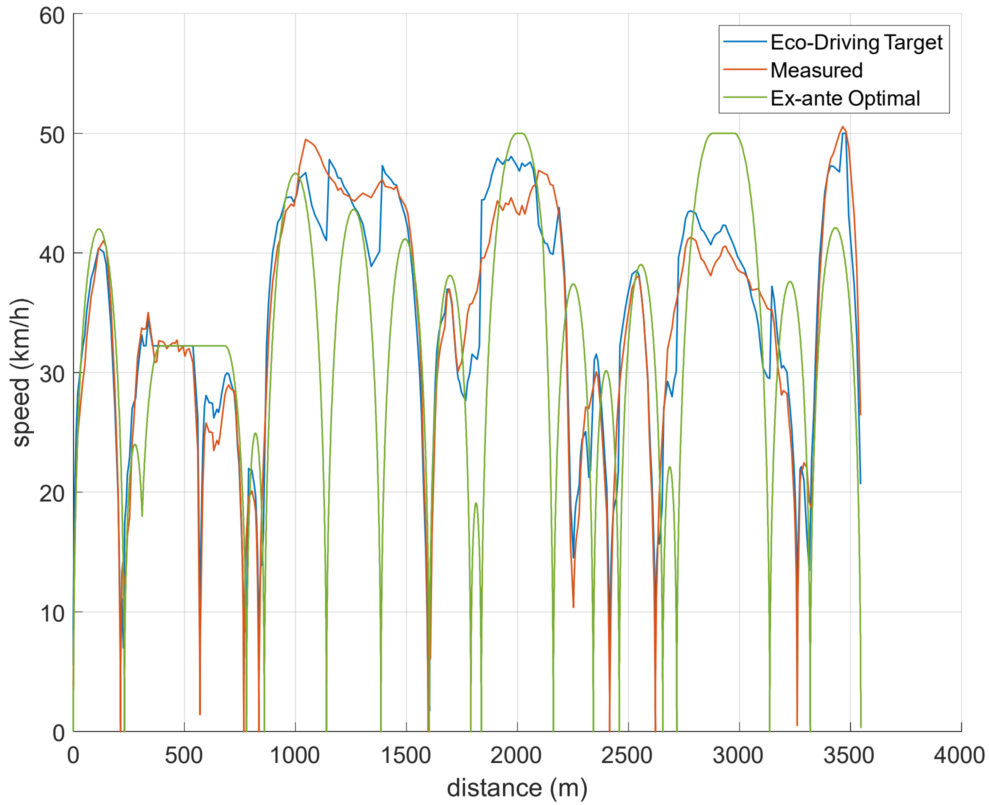




ECO-DRIVING Results (2/3)

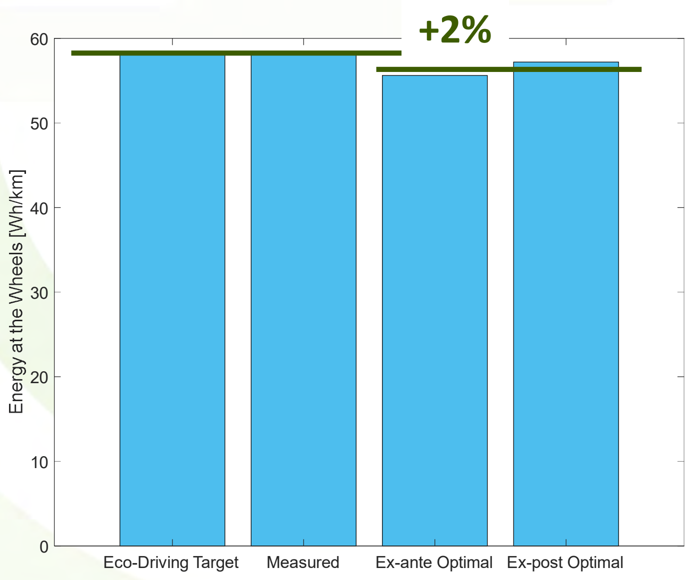
- Besides tests conducted by OEMs, IFPEN has run several driving tests with the eco-driving system, however
  - Without camera/radar
  - Without traffic light information

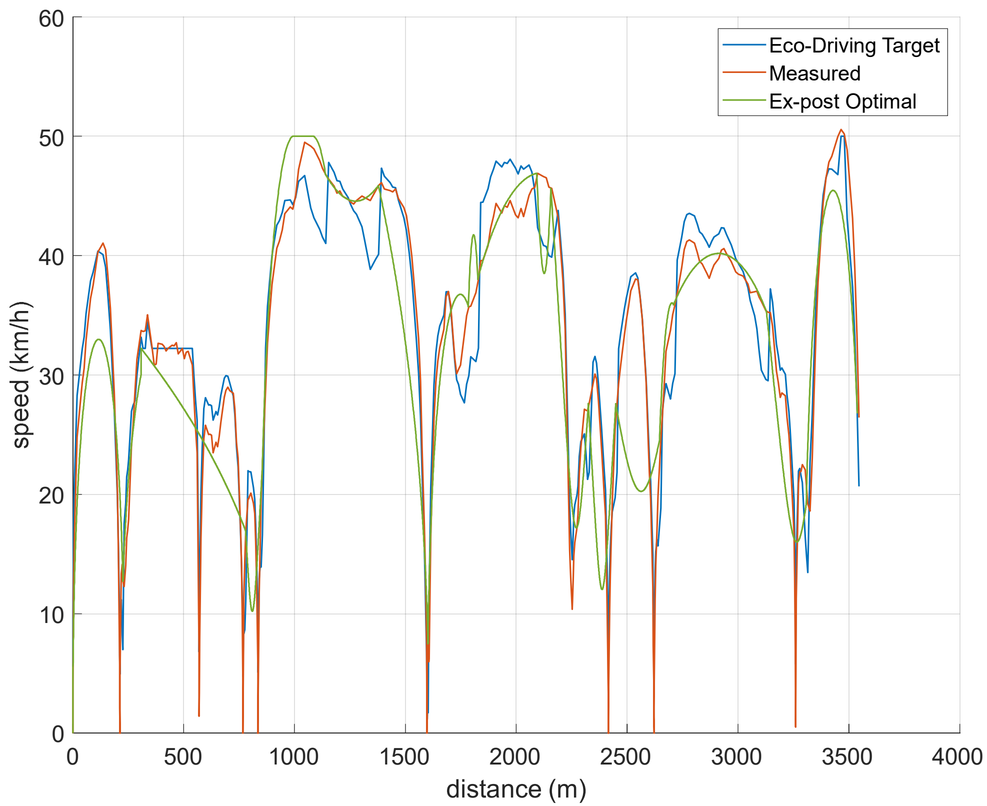
- Advised speeds (blue) are effective in anticipating stops and other events, i.e., lead the driver toward the estimated optimum (green).



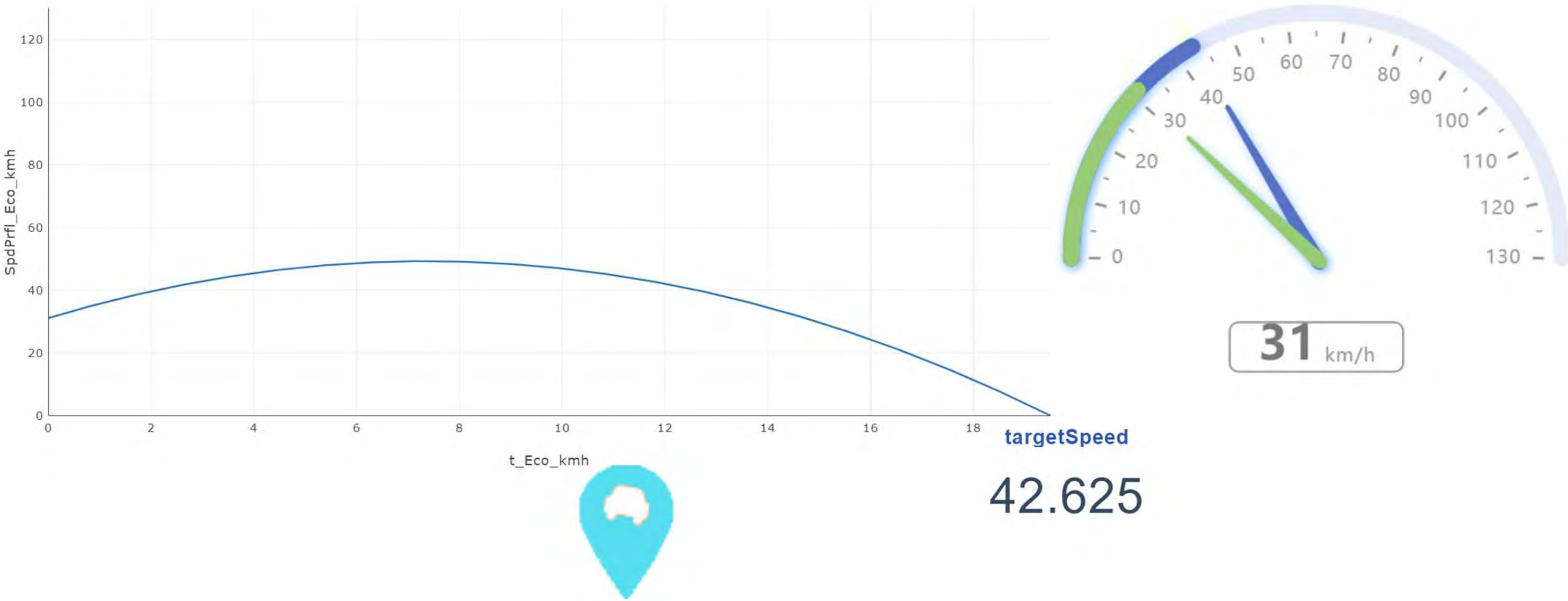
ECO-DRIVING Results (3/3)

- Actual constraints experienced during the drive can be post-processed, and an ex-post optimum calculated
- This analysis shows only small deviations between actual energy consumption with eco-driving and the predicted optimum





ECO-DRIVING HMI



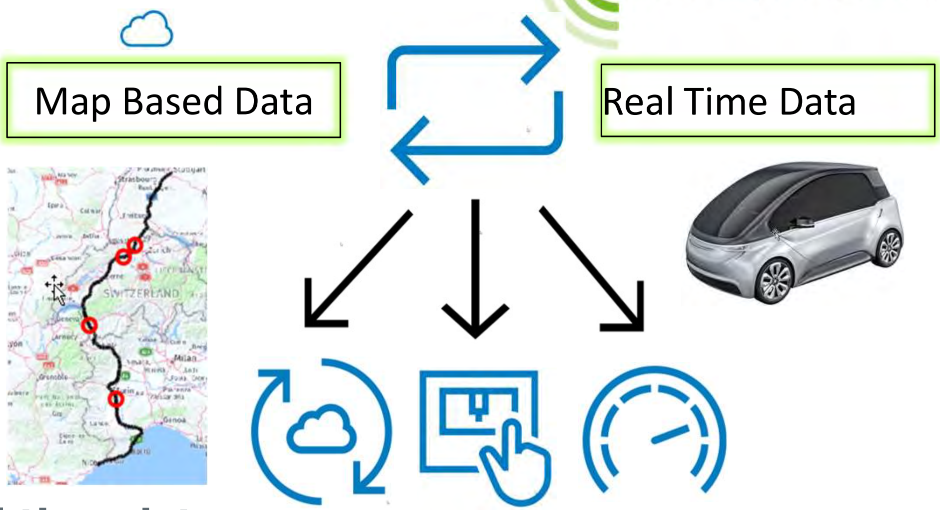
Trip Itinerary Manager
Enables a dynamic update of the trip and keeps the estimated arrival time up to date

Trip Itinerary – “ready to use”

… the most important factor when traveling from A to B is:

- Arriving at the destination at the scheduled (pre-calculated) arrival time.
- The newly developed **Trip Itinerary (TI)** function enables a dynamic update of the trip and keeps the estimated arrival time up to date.
    - TI is the central interface between off-board trip optimizer and vehicle application software.
    - TI compares actual vehicle state with trip plan and updates accordingly.

Trip Itinerary - Main Functions

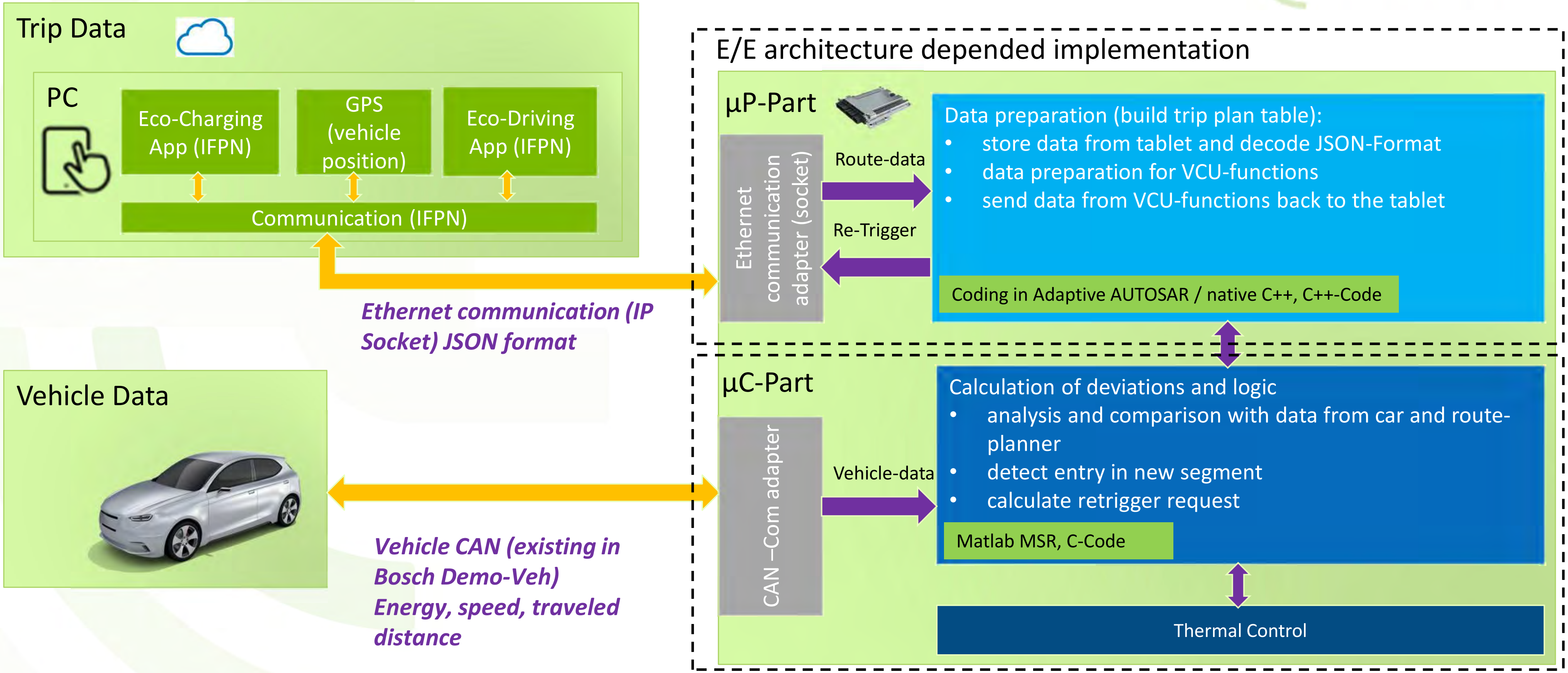


- **Receiving and processing data from eco-Charging:**
  - Store data from tablet and decode JSON-Format
  - Determination of current position as starting point
  - Data preparation for monitoring-functions
  - Transmit data back to the tablet – (re-trigger, speed advice, changed conditions (SOC))

- **Monitoring of deviations between predicted and real time data:**
  - Actual position vs. planned route
  - Predicted energy deviation of high voltage battery between planned and actual energy
  - Deviation between planned and actual travel time

- **Central coordination of reactions - based on calculated deviations, which include:**
  - Determination of trigger reason for re-calculation of the route
  - Start-trigger for battery pre-conditioning before charging event

Trip Itinerary – Functional Architecture



Trip Itinerary – Objectives

- **User centric solution:**
  - Increases convenience and provide support for long distance traveling (more than one charging stop).
  - Ensures that the vehicle arrives the charging points according to initial trip plan from routing-App.
  - Avoid uncertainty with belong to booked charging slot and charging duration.
- **Energy efficiency:**
  - Combination of eco-Charging, eco-Driving and Trip Itinerary ensures energy optimized traveling.
  - Displaying deviations between actual and predicted supports energy optimized driving (in best case feedback loop with ACC).
- **Long distance traveling:**
  - Re-trigger function enabling dynamic planning of charging stops to enhance long distance traveling for EVs with small battery capacity.
  - Increased usability of predicted data.

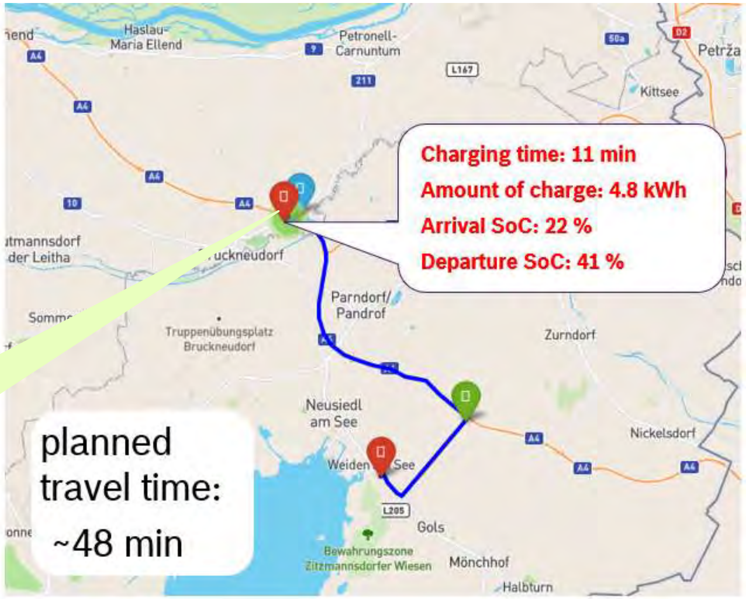
Trip Itinerary – Results

Example: Data sampled with real test vehicle driving near Vienna

Initial planned route at trip start

Re-trigger due to a higher energy consumption as predicted

New calculated route with additional charging stop



Additional charging-stop

Conclusions

- New connected functions have been developed within CEVOLVER
- Developments range from methodological research (optimal control, graph optimization…) to cloud/onboard implementation
  - In particular, one of the first (if not the first) demonstrations of a complete predictive ecodriving system based on a model-based optimization process and taking into account various driving situations and constraints
- These functions succesfully contribute to CEVOLVER objectives, as demonstrated experimentally → See Technical Targets