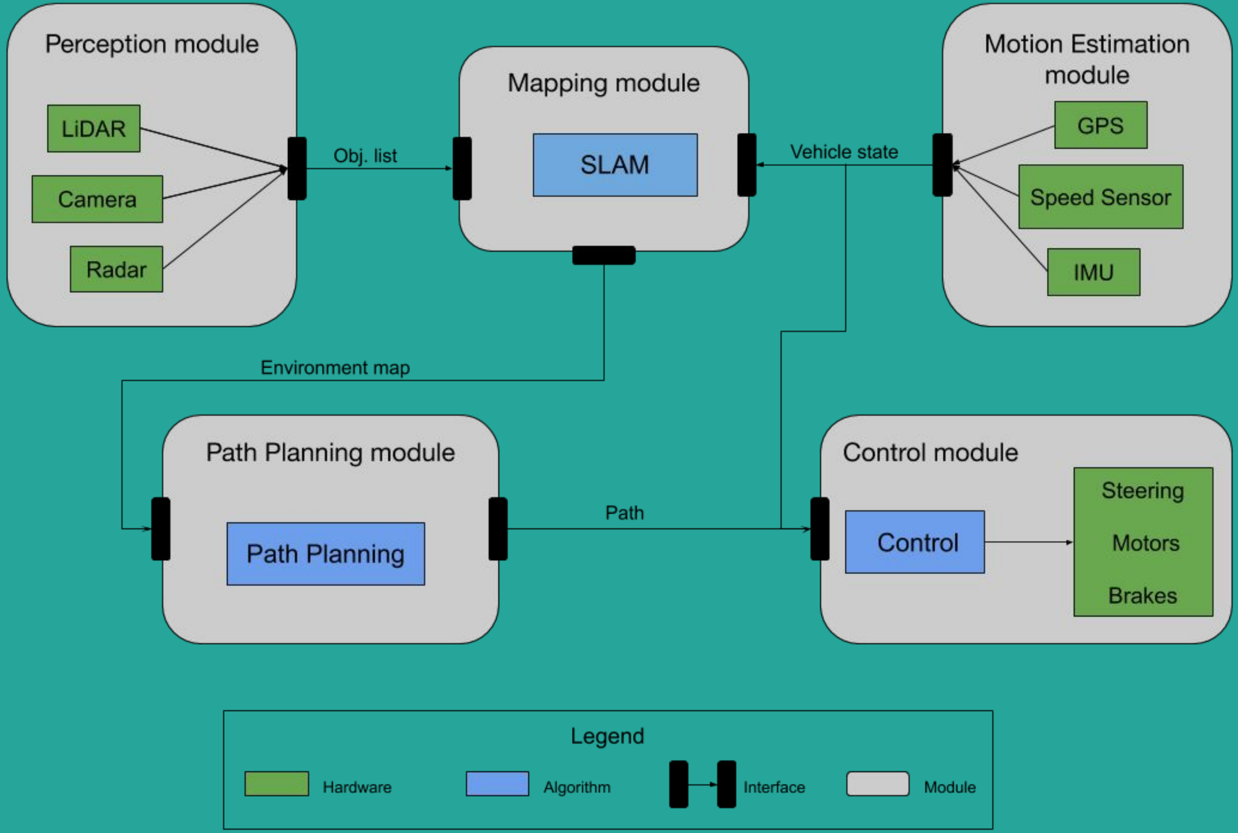
Bachelor project : Control module

Eloi Garandel, IC

Objectives

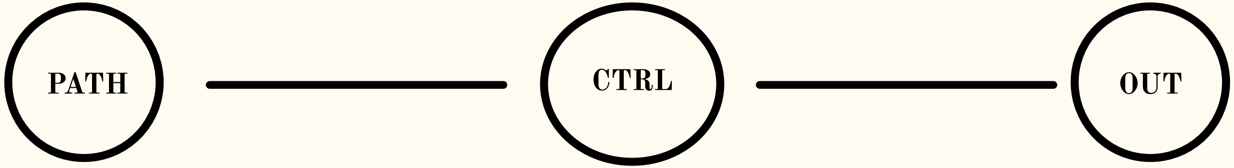
- Create a control module for a 2-wheel robot
- Adapt it for the EPFL Racing Team

What’s a Control module ?



Simplified architecture of a self-driving vehicle HW/SW

Control module operation



Path to follow

- Output of the path planning algorithm
- Output of the motion estimator

Control algorithm

Computes the next state of the car given the point to follow, the current state of the vehicle and the chosen model

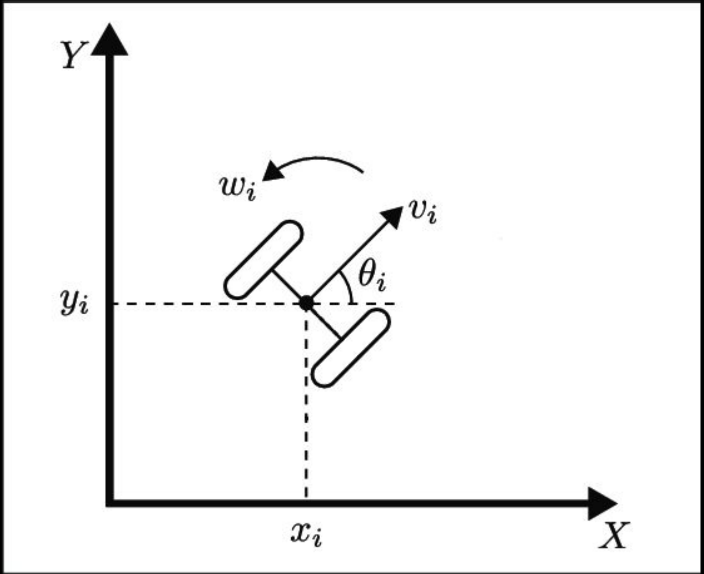
State of the vehicle

Outputs the new next optimal state of the vehicle

Robot model and motion equations

$$F_{robot}(\mathbf{x},\ \mathbf{u}) = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$x_{k+1} = f_{robot}(x_k, u_k) = x_k + \Delta t * F_{robot}(x_k, u_k)$$



Error function

$$Error(x_k, \mathbf{u_k}) = q_c * E(x_k) + q_v * C(\mathbf{\Delta u_k})$$

$$= \mathrm{q}_c * \frac{\frac{(X - X_{ref})^2}{maxErrorX^2} + \frac{(Y - Y_{ref})^2}{maxErrorY^2}}{2} + q_v * \begin{bmatrix} \Delta v & \Delta\omega \end{bmatrix} \begin{bmatrix} 0.5/(v_{max})^2 & 0.0 \\ 0.0 & 0.5/(\omega_{max})^2 \end{bmatrix} \begin{bmatrix} \Delta v & \Delta\omega \end{bmatrix}^T$$

MPC formulation

$$\min_{\mathbf{u}} \quad \sum_{k=0}^{N} q_c * E(x_k) + q_v * C(\mathbf{u_k})$$

$$\begin{aligned} s.t.\ & x(0) = x_0 \\ & x_{k+1} = f_{robot}(x_k, u_k) \\ & u_{min} \leq \mathbf{u} \leq u_{max} \\ & \Delta u_{min} \leq \mathbf{\Delta u} \leq \Delta u_{max} \end{aligned}$$

Demo and implementation explanation

EPFL Racing Team Adaptation

Similarities

Overall software structure

Error function

Differences

Motion equation and vehicle model

New vehicle model

$$F_{car}(\mathbf{x},\ \mathbf{u}) = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \\ \dot{v_x} \\ \dot{v_y} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} v_x cos\varphi - v_y sin\varphi \\ v_x sin\varphi - v_y cos\varphi \\ r \\ \frac{F_x}{m} \\ (\dot{\delta} v_x + \delta \dot{v_x}) \frac{l_R}{l_R + l_F} \\ (\dot{\delta} v_x + \delta \dot{v_x}) \frac{1}{l_R + l_F} \end{bmatrix}$$

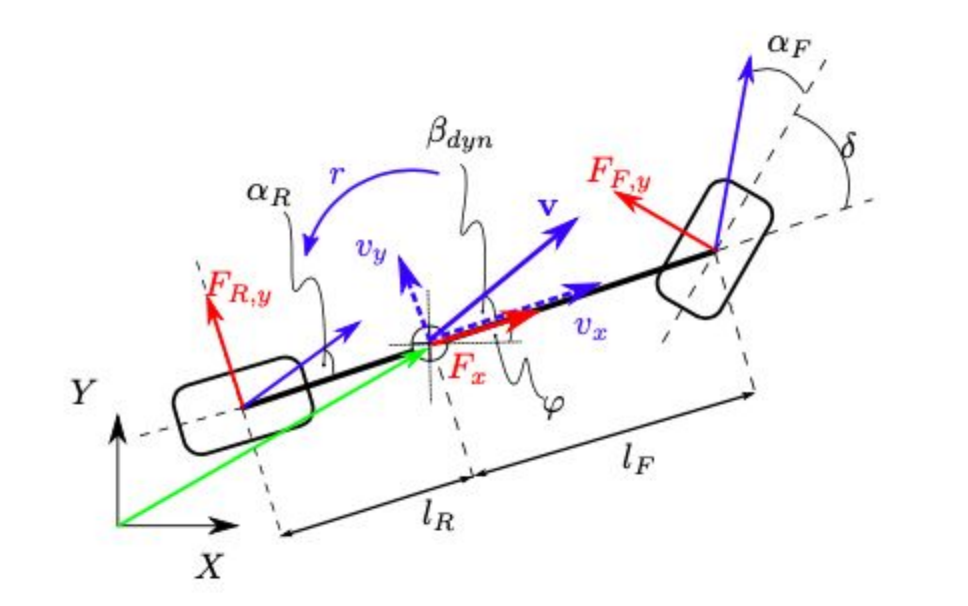


Image from AMZ 2019 report

Demo and implementation explanation

Next Semester objectives with EPFL Racing Team

- Test on real car
- Extend car model

Thank you for your attention
Feel free to ask any questions