1 Mapping Controls and States

Given: A car model which has state:

$$x_{car} = \begin{bmatrix} p_x \\ p_y \\ \delta \\ v \\ \psi \\ \dot{\psi} \\ \beta \end{bmatrix}$$

p_x Global Position, p_y Global Postion, δ Steering Angle, v Velocity, ψ Orientation, $\dot{\psi}$ Yaw Rate, β Slip Angle.
and control inputs:

$$u_{car} = \begin{bmatrix} \dot{\delta} \\ a \end{bmatrix}$$

$\dot{\delta}$ Steering Angle Rate, a Longitudinal Acceleration

The used planning model has its own state definition $x_{planning}$ and control inputs $u_{planning}$.

A planning model has to provide a mapping from the car states to its states:

$$x_{car} \mapsto x_{planning}$$

and a mapping from its control inputs to the control inputs of the car:

$$u_{planning} \mapsto u_{car}$$

1.1 Single Track Planning Model

$$x_{planning} = \begin{bmatrix} s \\ n \\ \xi \\ v \\ \delta \end{bmatrix}$$

$(s,\, n)$ Frenet Frame Coordinates along the road, ξ alignment error to the road, v velocity, δ steering angl.

$$u_{planning} = \begin{bmatrix} a \\ \dot{\delta} \end{bmatrix}$$

a Longitudinal Acceleration, $\dot{\delta}$ Steering Angle Rate

$$x_{car} = \begin{bmatrix} p_x \\ p_y \\ \delta \\ v \\ \psi \\ \dot{\psi} \\ \beta \end{bmatrix} \mapsto \begin{bmatrix} self.road.get\_road\_position(x, y)[0] \\ self.road.get\_road\_position(x, y)[1] \\ \psi - \theta(s) \\ v \\ \delta \end{bmatrix}$$

$$u_{planning} = \begin{bmatrix} a \\ \dot{\delta} \end{bmatrix} \mapsto \begin{bmatrix} \dot{\delta} \\ a \end{bmatrix} = \begin{bmatrix} \dot{\delta} \\ a \end{bmatrix}$$

1.2 Double Integrator Planning Model

$$x_{planning} = \begin{bmatrix} s \\ n \\ \dot{s} \\ \dot{n} \end{bmatrix}$$

$(s,\, n)$ Frenet Frame Coordinates along the road, $(\dot{s},\, \dot{n})$ Change of the Frenet Frame Coordinates along the road,

$$u_{planning} = \begin{bmatrix} u_t \\ u_n \end{bmatrix}$$

u_t, u_n artificial control inputs, which can be mapped (a_s, a_n) Acceleration of the Frenet Frame Coordinates along the road using g.

$$x_{car} = \begin{bmatrix} p_x \\ p_y \\ \delta \\ v \\ \psi \\ \dot{\psi} \\ \beta \end{bmatrix} \mapsto \begin{bmatrix} self.road.get\_road\_position(x,y)[0] \\ self.road.get\_road\_position(x,y)[1] \\ v\cos(\psi - \theta(s)) \\ v\sin(\psi - \theta(s)) \end{bmatrix}$$

For control input we additional require the current state $x_0 = [s_0, n_0, \dot{s}_0, \dot{n}_0]^T$ of the planning model and we need to store the current steering angle of the car δ_{cur}:

$$x_1 = \begin{bmatrix} s_1 \\ n_1 \\ \dot{s}_1 \\ \dot{n}_1 \end{bmatrix} = x_0 + \begin{bmatrix} \dot{s}_0 \\ \dot{n}_0 \\ u_t \\ u_n \end{bmatrix} dt$$

$$\begin{bmatrix} a_s \\ a_n \end{bmatrix} = g(u_t, u_n)$$

$$\xi_0 = \arctan\left(\frac{\dot{n}_0}{\dot{s}_0}\right)$$
$$\xi_1 = \arctan\left(\frac{\dot{n}_1}{\dot{s}_1}\right)$$

$$v_0 = \sqrt{\dot{s}_0^2 + \dot{n}_0^2}$$
$$v_1 = \sqrt{\dot{s}_1^2 + \dot{n}_1^2}$$

$$\psi_0 = \xi_0 + \theta(s_0)$$
$$\psi_1 = \xi_1 + \theta(s_1)$$

$$\dot{\psi} = \frac{\psi_1 - \psi_0}{dt}$$

$$u_{planning}, x_{planning} \mapsto \frac{1}{dt} \begin{bmatrix} \arctan(l_{wb}\frac{\dot{\psi}}{v_1}) - \delta_{cur} \\ v_1 - v_0 \end{bmatrix} = \begin{bmatrix} \dot{\delta} \\ a \end{bmatrix}$$

The result for this mapping shows that a is quite accurate. However, $\frac{l_{wb}}{v_x}\dot{\psi}$ does not account for the forces involved and represents a simplification. Consequently, $\dot{\delta}$ fails to produce the desired orientation change.