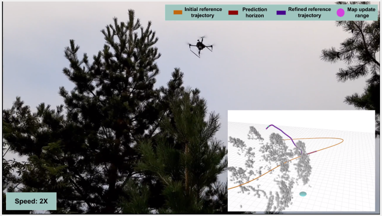
Motion Planning for Autonomous Vehicles

Model Predictive Control (MPC)

Geesara Kulathunga

April 21, 2023

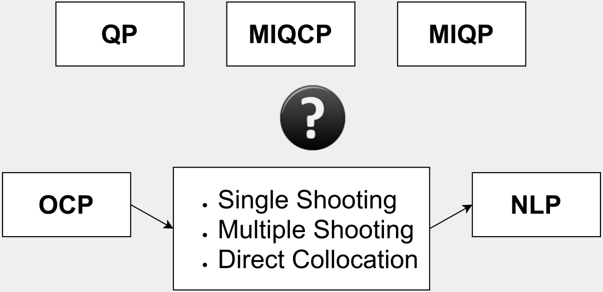


Model Predictive Control (MPC)

Contents

Ways to solve Optimal Control (OCP) Problems

An OCP problem can be transformed into an NLP problem. A problem is always solved better in a **nonlinear manner** as opposed to a **linearizing motion model** at every time since the motion model is nonlinear.



A **OCP** problem can **transform** into **NLP** in various ways, including MS (**Multiple-Shooting**) and DC (**Direct-Collocation**).

OCP Using Nonlinear Programming Problem (NLP)

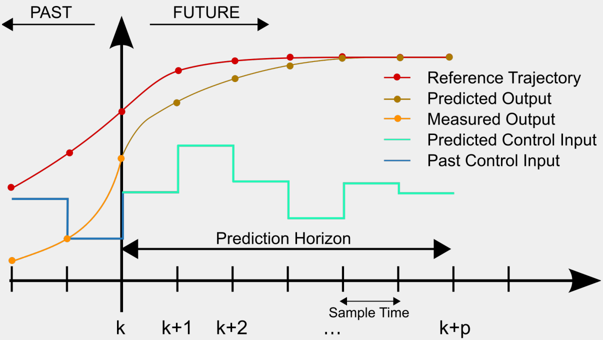
OCP

$$
\begin{aligned}
\min_{u} \quad & J_n(x_0, u) = \sum_{k=0}^{n-1} c(x(k), u(k)) \\
\text{s.t.} \quad & x_{k+1} = f(x(k), u(k)) \\
& x(0) = x_0 \\
& u(k) \in U \,\forall k \in [0, n-1],\ u \in U \subseteq \mathbf{R}^{n_u} \\
& x(k) \in X \,\forall k \in [0, n],\ x \in X \subseteq \mathbf{R}^{n_x}
\end{aligned}
$$

NLP

$$
\begin{aligned}
\min_{w} \quad & \varphi(F(w, x_0, t_k), w) \\
\text{s.t.} \quad & x_{k+1} = f(x(k), u(k)) \\
& g_1(F(w, x_0, t_k), w) \le 0 \\
& g_2(F(w, x_0, t_k), w) = 0
\end{aligned}
$$

In general, use the specified **model** to **predict** the motion of the system, **generate** a **locally optimal or feasible trajectory**, and **repeat** the procedure



Simon, D. (2014). Model Predictive Control in Flight Control Design: Stability and Reference Tracking (Doctoral dissertation, Linköping University Electronic Press).

Prediction Simulate states forward in time up to a defined horizon, **prediction horizon**, N_e from the **current state**

$$\underbrace{\mathbf{u}_k = \begin{bmatrix} u_{0|k} \\ u_{1|k} \\ \vdots \\ u_{N_e-1|k} \end{bmatrix}}_{\text{control inputs}} \xrightarrow{\text{estimate or calculate}} \underbrace{\mathbf{x}_k = \begin{bmatrix} x_{1|k} \\ x_{2|k} \\ \vdots \\ x_{N_e|k} \end{bmatrix}}_{\text{state vector}} \tag{1}$$

where $x_{i|k}$ denoted, current state x_k and $x_{i|k}$ denoted, i steps into the future, same for control as well

The **prediction** by **minimizing** a **stage cost**

$$J_{N_e}(x_k, \mathbf{u}_k) = \sum_{h=0}^{N_e} \left\| \mathbf{x}_{k+h} - \mathbf{x}_{k+h}^{ref} \right\|_Q^2 + \left\| \mathbf{u}_{k+h} - \mathbf{u}_{k+h}^{ref} \right\|_R^2$$

This can be solved numerically to estimate optimal $\mathbf{u}_k^*$

$$\mathbf{u}_k^* = \min_{\mathbf{u}} \quad J_{N_e}(x_k, \mathbf{u}_k), \quad Q \in \mathbb{R}^{n_x \times n_x} \succeq 0, \quad R \in \mathbb{R}^{n_u \times n_u} \succ 0$$

$$\begin{aligned} \text{s.t.} \quad & g_1(\mathbf{u}) = 0, \quad g_2(\mathbf{u}) \le 0 \\ & p^{lower} \le \mathbf{x}_{k+h} \le p^{upper} \quad \forall 0 \le h \le N_e \\ & u^{lower} \le \mathbf{u}_{k+h} \le u^{upper} \quad \forall 0 \le h \le N_e - 1, \end{aligned}$$

Apply the **first** element of $\mathbf{u}_k^*$ on the **system** and **repeat** the optimization

Model Predictive Control

- The **model** can be defined in **various ways**, multivariable, linear, or nonlinear, deterministic, stochastic or fuzzy
- Can handle **different types of constraints**, e.g., linear, quadratic, and nonlinear
- **Near-optimal** control inputs
- , however, requires **online optimization** that may be costly

Model Predictive Control: Prediction model

- If the plant **model** is linear, the model's **state depends** linearly on **control** inputs, i.e., $x_{k+1} = f(x_k, u_k)$. Hence, **cost**, in general, is quadratic in u_k subject to linear **constraints**. Such problems can be formulated as a convex quadratic program and **guaranteed** to have **global optimal solution** all the time.

$$\min_{\mathbf{u}} \mathbf{u}^\top R \mathbf{u} + 2r^\top \mathbf{u} \quad s.t\; A\mathbf{u} \leq b$$

$$\underset{x}{\text{minimize}} \quad f(x)$$

where $f(x) = \frac{1}{2}x^\top Q x + b^\top x + c$, where $c \in \mathbb{R}, x \in \mathbb{R}^2$, and Q is 2×2 matrix. First order necessary condition $\Delta f(x) = 0$

$$\begin{aligned} df &= \frac{1}{2}x^\top Q^\top dx + \frac{1}{2}x^\top Q dx + b^\top dx \\ &= \underbrace{\left(x^\top \frac{Q^\top + Q}{2} + b^\top\right)}_{d\dot{f}(x) = \Delta f(x)} dx \end{aligned} \tag{2}$$

Since $Q^\top = Q$, $\Delta f(x) = Qx + b$. Hence, the critical point: $Qx = -b$
Second order necessary condition $\Delta^2 f(x) = Q$. It can be either **minimum**, **maximum**, saddle point, or **singular** point, i.e., **at least one eigenvalue becomes zero**.

- If the plant **model** is nonlinear, the model's **state depends** non-linearly on **control** inputs, i.e., $x_{k+1} = Ax_k + Bu_k$. Hence, **cost**, in general, is nonconvex in u_k subject to convex and nonconvex **constraints**. Such problems are formulated as a nonlinear program and **does not guarantee** to have **global optimal solution** all the time. Therefore, the solution can have local minima, locally optimal, and may not be solved efficiently or reliably

$$\min_{\mathbf{u}} J(x_k, \mathbf{u}) \quad s.t\ g(x_k, \mathbf{u}) \leq 0$$

- **Discrete-time** necessary to have sampling interval δ, piecewise optimization is carried out
- **Continuous time** not necessary to have sampling interval δ, nor piece wise optimization is carried out. Can be linearized, good for nonlinear continuous-time systems

Model Predictive Control: Constraints

- **Hard** constraints are **satisfied** all the time, it is **not possible** to satisfy, the **problem** is **infeasible**
  - Box constraints

$$
\begin{aligned}
p^{lower} \leq \mathbf{x}_{k+h} \leq p^{upper} \quad & \forall 0 \leq h \leq N_e \\
u^{lower} \leq \mathbf{u}_{k+h} \leq u^{upper} \quad & \forall 0 \leq h \leq N_e - 1,
\end{aligned} \tag{3}
$$

  - System dynamics constraints

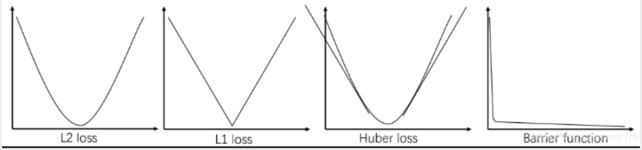
$$
g_1(\mathbf{w}) = \begin{bmatrix} \bar{\mathbf{x}}_k - \mathbf{x}_k \\ \vdots \end{bmatrix}. \tag{4}
$$

- **Soft** constraints may be **violated** to **avoid infeasibility**
  Consider the following **hard-constraints** optimization problem

$$\min_{x} \quad f(x)$$
$$\text{s.t.} \quad g_1(x) = c, \quad g_2(x) \leq d$$

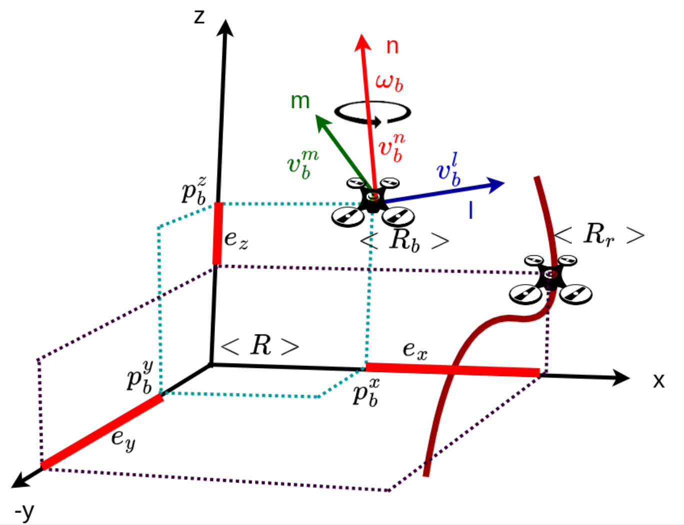
  Can be **converted** into a **soft constraints optimization problem**

$$\min_{x} \quad f(x) + \lambda_1 g_1(x) + \lambda_2 g_2(x)$$



  by using penalty terms or loss functions

Reference trajectory tracking



The system states $\mathbf{x}_k = [p_k^x, p_k^y, p_k^z, \alpha_k^z]^T \in \mathbb{R}^{n_x}$ and control inputs $\mathbf{u}_k = [v_k^x, v_k^y, v_k^z, \omega_k^z]^T \in \mathbb{R}^{n_u}$, where p_k^i and $v_k^i, i \in \{x, y, z\}$ denotes the quadrotor center position(m) and velocity (m/s) in each direction, i.e., x,y,z, at time t = k in the world coordinate frame; α_k^z and ω_k^z denote the yaw angle (rad) and yaw rate (rad/s) around the z-axis, respectively.
The simplified motion model is expressed by $\dot{\mathbf{x}}_k = \mathbf{f}_c(\mathbf{x}_k, \mathbf{u}_k)$

$$\dot{\mathbf{x}}_k = \begin{bmatrix} \dot{p}_k^x \\ \dot{p}_k^y \\ \dot{p}_k^z \\ \dot{\alpha}_k^z \end{bmatrix} = \begin{bmatrix} v_k^x cos(\alpha_k^z) - v_k^y sin(\alpha_k^z) \\ v_k^x sin(\alpha_k^z) + v_k^y cos(\alpha_k^z) \\ v_k^z \\ \omega_k^z \end{bmatrix}, \tag{5}$$

where $\mathbf{f}_c(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $n_x = n_u = 4$

Simplified Motion Model

Forward Euler discretization, $\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k)$ is introduced for a given sampling period in seconds, $\delta \in \mathbb{R} > 0$, e.g., $\delta = 0.1s$

$$\mathbf{x}_{k+1} = \begin{bmatrix} p_k^x \\ p_k^y \\ p_k^z \\ \alpha_k^z \end{bmatrix} + \delta \begin{bmatrix} v_k^x cos(\alpha_k^z) - v_k^y sin(\alpha_k^z) \\ v_k^x sin(\alpha_k^z) + v_k^y cos(\alpha_k^z) \\ v_k^z \\ \omega_k^z \end{bmatrix}, \tag{6}$$

where $\mathbf{f}_d(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$.

With Multiple Shooting

$$
\begin{aligned}
J_{N_e}(\mathbf{x}_k,\mathbf{u}_k) &= \sum_{h=0}^{N_e} \left\| \mathbf{x}_{k+h} - \mathbf{x}_{k+h}^{ref} \right\|_Q^2 + \left\| \mathbf{u}_{k+h} - \mathbf{u}_{k+h}^{ref} \right\|_R^2 \\
\min_{\mathbf{w}} \quad & J_{N_e}(\mathbf{x}_k,\mathbf{u}_k), \quad Q \in \mathbb{R}^{n_x \times n_x} \succeq 0, \quad R \in \mathbb{R}^{n_u \times n_u} \succ 0 \\
\text{s.t.} \quad & g_1(\mathbf{w}) = 0, \quad g_2(\mathbf{w}) \leq 0 \\
& p^{lower} \leq \mathbf{x}_{k+h} \leq p^{upper} \quad \forall 0 \leq h \leq N_e \\
& u^{lower} \leq \mathbf{u}_{k+h} \leq u^{upper} \quad \forall 0 \leq h \leq N_e - 1,
\end{aligned}
\tag{7}
$$

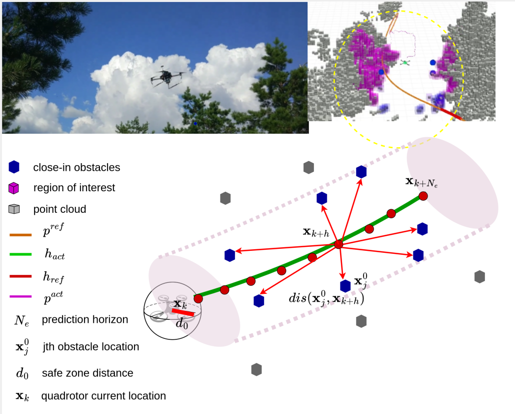
where $\mathbf{w} = [\mathbf{u}_k, \ldots, \mathbf{u}_{k+N_e-1}, \mathbf{x}_k, \ldots, \mathbf{x}_{k+N_e}]$ denotes the decision variables set to be minimized.

Notations u^{lower} and u^{upper} define the minimum and maximum linear and angular velocities allowed

Term $g_1(\mathbf{w})$ depicts the constraints that system dynamics imposes as follows:

$$g_1(\mathbf{w}) = \begin{bmatrix} \bar{\mathbf{x}}_k - \mathbf{x}_k \\ \vdots \\ f_d(\mathbf{x}_{k+h}, \mathbf{u}_{k+h}) - \mathbf{x}_{k+h+1} \\ \vdots \\ f_d(\mathbf{x}_{k+N_e-1}, \mathbf{u}_{k+N_e-1}) - \mathbf{x}_{k+N_e} \end{bmatrix}. \tag{8}$$

Reconstructing obstacle constraints in each iteration is necessary to **incorporate the dynamic environment changes** into the trajectory tracker



Term $g_2(\mathbf{w})$ describes the constraints imposed by obstacles.

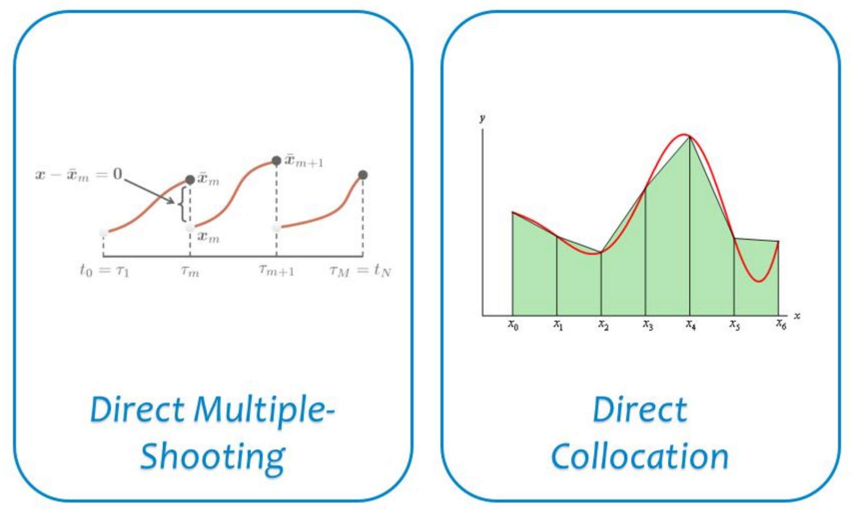
$$g_2(\mathbf{w}) = \begin{bmatrix} dis(\mathbf{x}_j^o, \mathbf{x}_k) \\ \vdots \\ dis(\mathbf{x}_j^o, \mathbf{x}_{k+h}) \\ \vdots \\ dis(\mathbf{x}_j^o, \mathbf{x}_{k+N_e}) \end{bmatrix}, \; j = 1, \ldots, N_o, \tag{9}$$

where $\bar{\mathbf{x}}_k$ is the initial position and N_o is the number of obstacles, and $dis(\mathbf{x}_j^o, \mathbf{x}_{k+h})$ can be calculated as follows:

$$-\sqrt{(x_j^o - x_{k+h})^2 + (y_j^o - y_{k+h})^2 + (z_j^o - z_{k+h})^2} + d^o$$

where d^o is the safe zone distance between the robot and close-in obstacles

Multiple shooting vs Direct Collocation

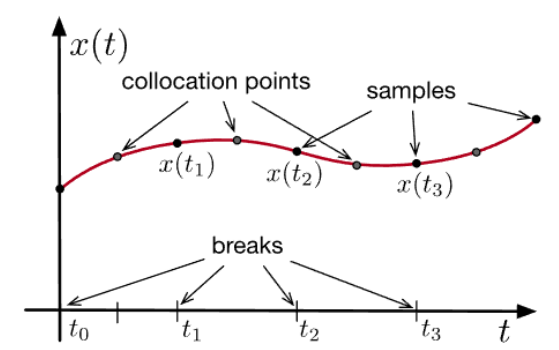


http://www.ee.ic.ac.uk/ICLOCS/Overview.html

With Direct Collocation (DC)

- **Multiple shooting**: **nonlinearity** with a sparsity structure to reduce the nonlinearity
- **Direct collocation**: add more degrees of freedom. Thus, exploits even more, but computation power increases dramatically
- **Collocation points** with respect to a **chosen polynomial**: Lagrangian 3rd order (N_d) polynomial, B-spline or Bézier
- **Fixed time interval** in multiple-shooting, but in DC, it has more **freedom** to determine how should **define points between two consecutive time interval**

- Kept the same discretization as in the multiple-shooting, i.e., $u(t) = u_k$, for $t \in [t_k, t_{k+1}], \quad k = 0, .., N_e - 1$, where N_e is the prediction horizon length

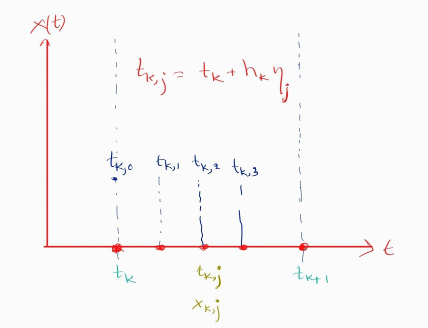


https://underactuated.csail.mit.edu/trajopt.html

- Consecutive time interval (t_k and t_{k+1}) is divided into small sub-intervals

$$t_{k,j} := t_k + h_k \eta_j, \quad k = 0, \ldots, N_e - 1,\ j = 0, \ldots, N_d$$

where Legendre points of order $N_d = 3$
$\eta = [0, 0.112, 0.500, 0.888]$ and $h_k = t_{k+1} - t_k$ and $x_{k,j}$ denote the states at $t_{k,j}$



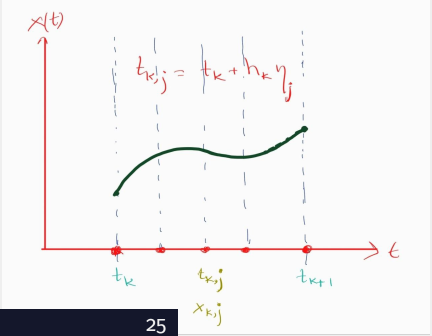
With Direct Collocation (DC)

- In each control interval, the Langrangian polynomial is defined as

$$L_j(\eta) = \prod_{r=0, r\neq j}^{N_d} \frac{\eta - \eta_r}{\eta_j - \eta_r}$$

with property

$$L_j(\eta) = \begin{cases} 1, & \text{if j =r} \\ 0, & \text{otherwise} \end{cases}$$



- State trajectory can be approximated using these basis functions

$$\bar{x}_k(t) = \Sigma_{r=0}^{N_d} L_r\left(\frac{t - t_k}{h_k}\right) x_{k,r}$$

Also, state at the end of the control interval

$$\bar{x}_{k+1}(t) = \Sigma_{r=0}^{N_d} L_r\left(1\right) x_{k,r}$$

And state time derivative at each collocation point except η_0

$$\dot{\bar{x}}_k(t) = \frac{1}{h_k} \Sigma_{r=0}^{N_d} \dot{L}_r\left(\eta_j\right) x_{k,r} := \frac{1}{h_k} \Sigma_{r=0}^{N_d} C_{r,j} x_{k,r}$$

With Direct Collocation (DC)

- Hence, these collocation equations that necessary to satisfy every state at every collocation point

$$h_k f_c(x_{k,j}, u_k) - \Sigma_{r=0}^{N_d} C_{r,j} x_{k,r} = 0, \quad k = 0, \ldots, N_e - 1, \quad j = 0, \ldots, N_d$$

  And the approximation of the end state

$$x_{k+1}(t) - \Sigma_{r=0}^{N_d} L_r(1) x_{k,r} = 0 \quad k = 0, \ldots, N_e - 1$$

Continuous nonlinear system linearization

- Consider a continuous-time nonlinear system

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}_c(\mathbf{x}(t), \mathbf{u}(t)),$$

where $\boldsymbol{f}_c : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, and reference trajectory

$$\dot{\mathbf{x}}^r(t) = \boldsymbol{f}_c(\mathbf{x}^r(t), \mathbf{u}^r(t)).$$

- Consider a continuous-time nonlinear system

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}_c(\mathbf{x}(t), \mathbf{u}(t)),$$

where $\boldsymbol{f}_c : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, and reference trajectory

$$\dot{\mathbf{x}}^r(t) = \boldsymbol{f}_c(\mathbf{x}^r(t), \mathbf{u}^r(t)).$$

- If $\mathbf{x}^r(t)$ is “close to” $\mathbf{x}(t)$,

$$\boldsymbol{f}(\mathbf{x},\mathbf{u}) \approx \boldsymbol{f}(\mathbf{x}^r, \mathbf{u}^r) + \frac{\partial}{\partial \mathbf{x}} \boldsymbol{f}(\mathbf{x}^r, \mathbf{u}^r)(\mathbf{x} - \mathbf{x}^r) + \frac{\partial}{\partial \mathbf{u}} \boldsymbol{f}(\mathbf{x}^r, \mathbf{u}^r)(\mathbf{u} - \mathbf{u}^r),$$

- Consider a continuous-time nonlinear system

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}_c(\mathbf{x}(t), \mathbf{u}(t)),$$

where $\boldsymbol{f}_c : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, and reference trajectory

$$\dot{\mathbf{x}}^r(t) = \boldsymbol{f}_c(\mathbf{x}^r(t), \mathbf{u}^r(t)).$$

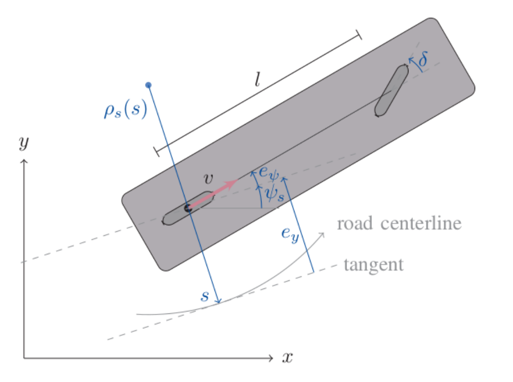
- If $\mathbf{x}^r(t)$ is “close to” $\mathbf{x}(t)$,

$$\boldsymbol{f}(\mathbf{x},\mathbf{u}) \approx \boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r) + \frac{\partial}{\partial \mathbf{x}}\boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r)(\mathbf{x}-\mathbf{x}^r) + \frac{\partial}{\partial \mathbf{u}}\boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r)(\mathbf{u}-\mathbf{u}^r),$$

- Hence,

$$\dot{\mathbf{x}} - \dot{\mathbf{x}}^r = \boldsymbol{f}(\mathbf{x},\mathbf{u}) - \boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r) \approx \frac{\partial}{\partial \mathbf{x}}\boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r)(\mathbf{x}-\mathbf{x}^r) + \frac{\partial}{\partial \mathbf{u}}\boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r)(\mathbf{u}-\mathbf{u}^r).$$

Continuous nonlinear system linearization



$$\dot{x} = \frac{dx}{dt} = v cos(\psi), \quad \dot{y} = \frac{dy}{dt} = v sin(\psi), \quad \dot{\psi} = \frac{d\psi}{dt} = \frac{v}{l} tan(\delta) \qquad (10)$$

Lima, P. F., Mårtensson, J., Wahlberg, B. (2017, December). Stability conditions for linear time-varying model predictive control in autonomous driving. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC) (pp. 2775-2782). IEEE.

Consider a car-like robot $\mathbf{x}^r(t) = (x^r(t), y^r(t), \theta^r(t), \delta^r(t))$ and $\mathbf{u}^r(t) = (v^r(t), \omega^r(t))$, hence

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{x}} \boldsymbol{f}(\mathbf{x}^r, \mathbf{u}^r) &= \begin{bmatrix} 0 & 0 & -\sin(\theta^r)v^r & 0 \\ 0 & 0 & \cos(\theta^r)v^r & 0 \\ 0 & 0 & 0 & \frac{v^r}{l\cos(\delta^r)^2} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
\frac{\partial}{\partial \mathbf{u}} \boldsymbol{f}(\mathbf{x}^r, \mathbf{u}^r) &= \begin{bmatrix} \cos(\theta^r) & 0 \\ \sin(\theta^r) & 0 \\ \frac{\tan(\delta^r)}{l} & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}
\tag{11}
$$

Continuous nonlinear system linearization

Forward Euler discretization leads to

$$\frac{\mathbf{x}_{k+1}-\mathbf{x}_k}{\Delta T}-\frac{\mathbf{x}^r_{k+1}-\mathbf{x}^r_k}{\Delta T}\approx\frac{\partial}{\partial\mathbf{x}}\boldsymbol{f}(\mathbf{x}^r_k,\mathbf{u}^r_k)(\mathbf{x}_k-\mathbf{x}^r_k)+\frac{\partial}{\partial\mathbf{u}}\boldsymbol{f}(\mathbf{x}^r_k,\mathbf{u}^r_k)(\mathbf{u}_k-\mathbf{u}^r_k).$$

$$\mathbf{x}_{k+1}-\mathbf{x}^r_{k+1}+\mathbf{x}^r_k-\mathbf{x}_k\approx\Delta T\frac{\partial}{\partial\mathbf{x}}\boldsymbol{f}(\mathbf{x}^r_k,\mathbf{u}^r_k)(\mathbf{x}_k-\mathbf{x}^r_k)+\Delta T\frac{\partial}{\partial\mathbf{u}}\boldsymbol{f}(\mathbf{x}^r_k,\mathbf{u}^r_k)(\mathbf{u}_k-\mathbf{u}^r_k).$$

$$\underbrace{\mathbf{x}_{k+1}-\mathbf{x}^r_{k+1}}_{\boldsymbol{x}_{k+1}}\approx\underbrace{\left(\Delta T\frac{\partial}{\partial\mathbf{x}}\boldsymbol{f}(\mathbf{x}^r_k,\mathbf{u}^r_k)+\boldsymbol{I}\right)}_{\boldsymbol{A}_k}\underbrace{(\mathbf{x}_k-\mathbf{x}^r_k)}_{\boldsymbol{x}_k}+\underbrace{\Delta T\frac{\partial}{\partial\mathbf{u}}\boldsymbol{f}(\mathbf{x}^r_k,\mathbf{u}^r_k)}_{\boldsymbol{B}_k}\underbrace{(\mathbf{u}_k-\mathbf{u}^r_k)}_{\boldsymbol{u}_k}.$$

This produces a linear time-varying dynamical system

$$\boldsymbol{x}_{k+1}=\boldsymbol{A}_k\boldsymbol{x}_k+\boldsymbol{B}_k\boldsymbol{u}_k,\quad k=0,\dots,N_e-1.$$

However, these type of problem formulation could lead into a **infeasible solution** time to time, i.e., no control inputs that satisfied given constraints. Additionally, **feasibility** does **not ensure** the **asymptotically stability**.

$$
\begin{aligned}
\min_{\mathbf{u}} \quad & \sum_{k=0}^{N_e-1} \mathbf{x}_k^\top \mathbf{Q_k} \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R_k} \mathbf{u}_k + \mathbf{x}_{N_e}^\top \mathbf{Q_{N_e}} \mathbf{x}_{N_e}, \\
\text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k, \quad k = 0, \dots, N_e - 1 \\
& \boldsymbol{x}_0 \text{ is given}, \\
& \boldsymbol{G}_k \boldsymbol{x}_k \leq \boldsymbol{g}_k - \boldsymbol{G}_k \mathbf{x}_{\boldsymbol{k}}^{\boldsymbol{r}}, \quad k = 1, \dots, N_e \\
& \boldsymbol{H}_k \boldsymbol{u}_k \leq \boldsymbol{h}_k - \boldsymbol{H}_k \mathbf{u}_{\boldsymbol{k}}^{\boldsymbol{r}}, \quad k = 0, \dots, N_e - 1
\end{aligned}
\tag{12}
$$

In **receding-horizon control, stability** and **feasibility** are not ensured all the time.

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sum_{k=0}^{N_e-1} \mathbf{x}_k^\top \mathbf{Q_k} \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R_k} \mathbf{u}_k + \mathbf{x}_{N_e}^\top \mathbf{Q_{N_e}} \mathbf{x}_{N_e}, \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k, \quad k = 0, \ldots, N_e - 1 \\ & \mathbf{x}_0 \text{ is given} \end{aligned}$$

where $\mathbf{Q_k} = \mathbf{Q_k}^\top \succeq 0, \mathbf{R_k} = \mathbf{R_k}^\top \succ 0.$ For a **linearaized** system, the state prediction sequence can be written in a compact sequence as follows:

$$\mathbf{X} = M x_0 + C\mathbf{U},$$

$$M = \begin{bmatrix} A_0 \\ A_1 A_0 \\ \vdots \\ A_{N_e-1} \ldots A_0 \end{bmatrix}, C = \begin{bmatrix} B_0 & & & \\ A_1 B_0 & B_1 & & \\ \vdots & \vdots & \ddots & \\ A_{N_e-1} \ldots A_1 B_0 & A_{N_e-1} \ldots A_2 B_1 & \ldots & B_{N_e-1} \end{bmatrix} \tag{13}$$

The defined quadratic cost (13) can be written in terms of $\mathbf{x}$ and $\mathbf{u}$ as

$$J = \mathbf{X}^\top \tilde{Q}\mathbf{X} + \mathbf{U}^\top \tilde{R}\mathbf{U} = \mathbf{U}^\top H\mathbf{U} + 2x_0^\top F^\top \mathbf{U} + x_0^\top G x_0 \tag{14}$$

where $H = C^\top \tilde{Q}C + \tilde{R}$, $F = C^\top \tilde{Q}M$, and $G = M^\top \tilde{Q}M$

$$\begin{aligned} \min_{\mathbf{U}} \quad & \mathbf{U}^\top H\mathbf{U} + 2x_0^\top F^\top \mathbf{U}, \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k, \quad k = 0, \ldots, N_e - 1 \\ & \boldsymbol{x}_0 \text{ is given}, \\ & \boldsymbol{G}_k \boldsymbol{x}_k \le \boldsymbol{g}_k - \boldsymbol{G}_k \mathbf{x}^{\boldsymbol{r}}_{\boldsymbol{k}}, \quad k = 1, \ldots, N_e \\ & \boldsymbol{H}_k \boldsymbol{u}_k \le \boldsymbol{h}_k - \boldsymbol{H}_k \mathbf{u}^{\boldsymbol{r}}_{\boldsymbol{k}}, \quad k = 0, \ldots, N_e - 1 \end{aligned}$$

Since $x_0^\top G x_0$ is a constant value, can be discarded from the objective function

The constraints $\mathbf{GX} \leq \mathbf{g}$ becomes $\mathbf{GCU} \leq \mathbf{g} - \mathbf{GM}x_0$

$$\begin{aligned} \min_{\mathbf{U}} \quad & \mathbf{U}^\top H\mathbf{U} + 2x_0^\top F^\top \mathbf{U}, \\ \text{s.t.} \quad & \begin{bmatrix} \boldsymbol{GC} \\ \boldsymbol{H} \end{bmatrix} \boldsymbol{U} \leq \begin{bmatrix} \boldsymbol{g} - \boldsymbol{GM}x_0 \\ \boldsymbol{h} \end{bmatrix}. \end{aligned}$$

- Boundary conditions on the velocities
  Let $v_{min} \leq v \leq v_{max}$ be allowed velocity range. Since the model was linearized

$$v_{min} - v_k^r \leq v_k - v_k^r \leq v_{max} - v_k^r, \quad k = 0, \ldots, N_e - 1 \tag{15}$$

- Boundary conditions on the steering angle
  Let $\delta_{min} \leq \delta \leq \delta_{max}$ be allowed steering limits.

$$\mathbf{1}\delta_{min} - \begin{bmatrix} \delta_1^r \\ \vdots \\ \delta_D^r \end{bmatrix} \leq \boldsymbol{GX} \leq \mathbf{1}\delta_{max} - \begin{bmatrix} \delta_1^r \\ \vdots \\ \delta_{N_e}^r \end{bmatrix}, \tag{16}$$

where $\boldsymbol{G} = \begin{bmatrix} 0 & 0 & 0 & 1 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & & & & & & & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & 0 & 0 & 0 & 1 \end{bmatrix}$.

- Term $\boldsymbol{GC}$ is constant, depends only on the sampling interval ΔT due the structure of the $\mathbf{C}$, i.e., check eq.(13) and eq.(11)

$$\boldsymbol{GC} = \begin{bmatrix} 0 & \Delta T & 0 & 0 & \dots & 0 & 0 \\ 0 & \Delta T & 0 & \Delta T & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & 0 & 0 \\ 0 & \Delta T & 0 & \Delta T & & 0 & \Delta T \end{bmatrix}. \tag{17}$$

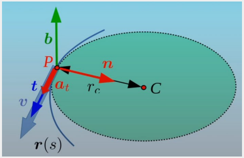
Velocity and acceleration in Frenent-Serret frame

- Derivative of $\mathbf{r}(s)$ with respect to time:

$$\dot{\mathbf{r}} = \dot{s}\mathbf{r}', \quad v = \dot{s}, \quad \mathbf{t} = \mathbf{r}' \rightarrow \mathbf{v} = v\mathbf{t}$$

$$\ddot{\mathbf{r}} = \dot{s}^2\mathbf{r}'' + \ddot{s}\mathbf{r}', \quad a = \ddot{s} = \dot{v}, \quad \mathbf{t}' = \mathbf{r}''$$

$$\rightarrow \mathbf{a} = a\mathbf{t} + v^2\mathbf{t}'$$



- Applying the first Frenent-Serret formula: $\mathbf{t}' = k\mathbf{n}$

$$\mathbf{a} = a\mathbf{t} + v^2 k\mathbf{n} = a\mathbf{t} + (v^2/r_c)\mathbf{n}$$

https://www.youtube.com/watch?v=aFCMIt63pgc

- Constraints on the tangential acceleration

$$a_k = (v_k - v_{k-1})/\Delta T$$

- Constraints on the tangential acceleration

$$a_k = (v_k - v_{k-1})/\Delta T$$

- Considering the linearized system,

$$a_{min} - a_k^r \le a_k - a_k^r \le a_{max} - a_k^r, \quad k = 0, \ldots, N_e - 1 \qquad (18)$$

Continuous nonlinear system linearization

- Constraints on the tangential acceleration

$$a_k = (v_k - v_{k-1})/\Delta T$$

- Considering the linearized system,

$$a_{min} - a_k^r \leq a_k - a_k^r \leq a_{max} - a_k^r, \quad k = 0, \dots, N_e - 1 \tag{18}$$

-

$$\Delta T a_{min} - v_k^r + v_{k-1}^r \leq (v_k - v_k^r) - (v_{k-1} - v_{k-1}^r) \leq \Delta T a_{max} - v_k^r + v_{k-1}^r. \tag{19}$$

- Constraints on the tangential acceleration

$$a_k = (v_k - v_{k-1})/\Delta T$$

- Considering the linearized system,

$$a_{min} - a^r_k \leq a_k - a^r_k \leq a_{max} - a^r_k, \quad k = 0, \dots, N_e - 1 \tag{18}$$

-

$$\Delta T a_{min} - v^r_k + v^r_{k-1} \leq (v_k - v^r_k) - (v_{k-1} - v^r_{k-1}) \leq \Delta T a_{max} - v^r_k + v^r_{k-1}. \tag{19}$$

- The first constraint, where $k = 0$ is

$$\Delta T a_{min} - v^r_0 + v_{-1} \leq \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{u}_0 \leq \Delta T a_{max} - v^r_0 + v_{-1}. \tag{20}$$

- The remaining constraints $(v_k - v_k^r) - (v_{k-1} - v_{k-1}^r) := \mathbf{HU}$

$$\mathbf{1}\Delta T a_{min} - \begin{bmatrix} v_1^r \\ \vdots \\ v_{N_e-1}^r \end{bmatrix} + \begin{bmatrix} v_0^r \\ \vdots \\ v_{N_e-2}^r \end{bmatrix} \leq \boldsymbol{HU} \leq \mathbf{1}\Delta T a_{max} - \begin{bmatrix} v_1^r \\ \vdots \\ v_{N_e-1}^r \end{bmatrix} + \begin{bmatrix} v_0^r \\ \vdots \\ v_{N_e-2}^r \end{bmatrix}, \tag{21}$$

where

$$\boldsymbol{H} = \begin{bmatrix} -1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 & 1 & 0 \end{bmatrix}. \tag{22}$$

- The magnitude of centripetal acceleration

$$a^c = \frac{v^2}{R} = |v||\omega|, \tag{23}$$

where $R = |v^r|/|\omega^r|$ is the radius of the curve, which can be assumed to be independent from changes of the control inputs and ω is the angular velocity

- The magnitude of centripetal acceleration

$$a^c = \frac{v^2}{R} = |v||\omega|, \tag{23}$$

where $R = |v^r|/|\omega^r|$ is the radius of the curve, which can be assumed to be independent from changes of the control inputs and ω is the angular velocity

- The bounds on tangential velocity

$$v_{max} = \sqrt{\frac{a^c_{max}|v^r|}{|\omega^r|}}, \quad v_{min} = -\sqrt{\frac{a^c_{max}|v^r|}{|\omega^r|}}, \tag{24}$$

where $a^c_{max} = \frac{v^2|\omega^r|}{|v^r|}$

- The constraints on tangential velocity

$$-\sqrt{\frac{a^c_{max}|v^r_k|}{|\omega^r_k|}} \leq v_k \leq \sqrt{\frac{a^c_{max}|v^r_k|}{|\omega^r_k|}}, \quad \text{where} \quad \omega^r_k = \frac{\theta^r_{k+1} - \theta^r_k}{\Delta T}, \quad (25)$$

- The constraints on tangential velocity

$$-\sqrt{\frac{a_{max}^{c}|v_k^r|}{|\omega_k^r|}} \leq v_k \leq \sqrt{\frac{a_{max}^{c}|v_k^r|}{|\omega_k^r|}}, \quad \text{where} \quad \omega_k^r = \frac{\theta_{k+1}^r - \theta_k^r}{\Delta T}, \quad (25)$$

- Considering the linearized system,

$$-\sqrt{\frac{a_{max}^{c}|v_k^r|}{|\omega_k^r|}} - v_k^r \leq v_k - v_k^r \leq \sqrt{\frac{a_{max}^{c}|v_k^r|}{|\omega_k^r|}} - v_k^r, \quad \omega_k^r \neq 0 \qquad (26)$$

ω^r is zero when **a car follows a straight line** and **hence does not have centripetal acceleration**.

Discrete-time nonlinear system linearization

Consider discrete-time nonlinear system

$$\mathbf{x}(k+1) = \boldsymbol{f}_c(\mathbf{x}(k), \mathbf{u}(k)),$$

where $\boldsymbol{f}_c : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, and reference trajectory

$$\mathbf{x}^r(k+1) = \boldsymbol{f}_c(\mathbf{x}^r(k), \mathbf{u}^r(k)).$$

If $\mathbf{x}^r(k)$ is "close to" $\mathbf{x}(k)$,

$$\boldsymbol{f}(\mathbf{x},\mathbf{u}) \approx \boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r) + \frac{\partial}{\partial \mathbf{x}}\boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r)(\mathbf{x}-\mathbf{x}^r) + \frac{\partial}{\partial \mathbf{u}}\boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r)(\mathbf{u}-\mathbf{u}^r),$$

Hence,

$$\begin{aligned}\mathbf{x}(k+1) - \mathbf{x}^r(k+1) &= \boldsymbol{f}(\mathbf{x},\mathbf{u}) - \boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r) \\ &\approx \frac{\partial}{\partial \mathbf{x}}\boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r)(\mathbf{x}-\mathbf{x}^r) + \frac{\partial}{\partial \mathbf{u}}\boldsymbol{f}(\mathbf{x}^r,\mathbf{u}^r)(\mathbf{u}-\mathbf{u}^r).\end{aligned} \tag{27}$$

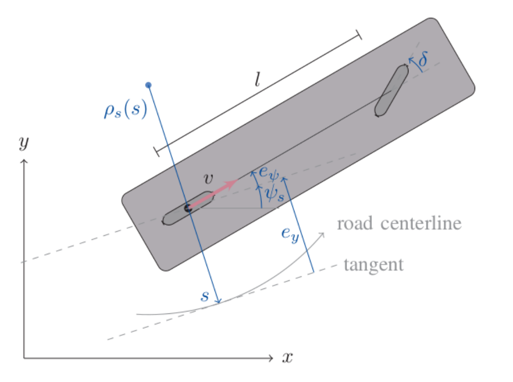
$$\underbrace{\mathbf{x}_{k+1}-\mathbf{x}^{r}_{k+1}}_{\boldsymbol{x}_{k+1}} \approx \underbrace{\left(\frac{\partial}{\partial \mathbf{x}}\boldsymbol{f}(\mathbf{x}^{r}_{k},\mathbf{u}^{r}_{k})\right)}_{\boldsymbol{A}_k}\underbrace{(\mathbf{x}_{k}-\mathbf{x}^{r}_{k})}_{\boldsymbol{x}_k}+\underbrace{\frac{\partial}{\partial \mathbf{u}}\boldsymbol{f}(\mathbf{x}^{r}_{k},\mathbf{u}^{r}_{k})}_{\boldsymbol{B}_k}\underbrace{(\mathbf{u}_{k}-\mathbf{u}^{r}_{k})}_{\boldsymbol{u}_k}.$$

This produces **a first order approximation** of the considered **nonlinear system** as a **linear time-varying** dynamical system.

$$\boldsymbol{x}_{k+1}=\boldsymbol{A}_k\boldsymbol{x}_k+\boldsymbol{B}_k\boldsymbol{u}_k, \quad k=0,\dots,N_e-1.$$

When system is subject to **feedback control law** $\mathbf{u}=K(k)\mathbf{x}(k)$, where $K(k)$ is the **time varying feedback control gain**. Therefor, **discrete-time autonomous system** can be desgined

$$\mathbf{x}(k+1)=f(\mathbf{x}(k),K(k))$$



$$\dot{x} = \frac{dx}{dt} = v_x cos(\psi), \quad \dot{y} = \frac{dy}{dt} = v_x sin(\psi), \quad \dot{\psi} = \frac{d\psi}{dt} = \frac{v_x}{l} tan(\delta) \quad (28)$$

Lima, P. F., Mårtensson, J., Wahlberg, B. (2017, December). Stability conditions for linear time-varying model predictive control in autonomous driving. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC) (pp. 2775-2782). IEEE.

By considering the Linear Time-Varying Model Predictive Control, the following set of expressions can be derived.

$$\dot{e}_y = v_x sin(e_\psi),$$
$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_s,$$
$$\dot{s} = \omega \cdot \frac{1}{k_r} = \frac{v_x cos(e_\psi)}{k_r(1/k_r - d(t))} = \frac{v_x cos(e_\psi)}{1 - k_r d(t)} = \frac{\rho_s v cos(e_\psi)}{\rho_s - e_y}, \quad \omega = \frac{v_x cos(e\psi)}{(\frac{1}{k_r} - d(t))} \tag{29}$$

where ρ_s is the radius of curvature of the road ψ_s is the road heading angle.

The next step is to derive with respect to s, i.e., $\frac{d(\cdot)}{ds} = \frac{d(\cdot)}{dt}\frac{dt}{ds} = \frac{d(\cdot)}{dt}\frac{1}{\dot{s}}$.

$$e'_y = \frac{\dot{e}_y}{\dot{s}} = \frac{v_x sin(e_\psi)}{v_x cos(e_\psi)}(1-d(t)k_r) = \Big(1-d(t)k_r\Big)tan(e_\psi)$$

$$e'_\psi = \frac{\frac{v_x}{k} - \frac{v_x cos(e_\psi)}{1/k_r - d(t)}}{\frac{v_x cos(e_\psi)}{1-k_r d(t)}} = \frac{k(\rho_s - d(t))}{\rho_s cos(e_\psi)} - \frac{1}{\rho_s} = \frac{k(\rho_s - d(t))}{\rho_s cos(e_\psi)} - \psi'_s, \tag{30}$$

$$\psi'_s = \frac{\dot{\psi}_s}{\dot{s}} = \frac{\frac{v_x cos(e_\psi)k_r}{1-d(t)k_r}}{\frac{v_x cos(e_\psi)k_r}{1-d(t)}} = \frac{1}{\rho_s}$$

After linearizing the nonlinear model (eq.30), around the reference trajectory $[e_{y,r}(k)\, e_{\psi,r}(k)] = [0\, 0]$ for all $k \geq 0$, the reason for having zeros is that those were already considered when defining the state sequence $[e_y(k)\, e_\psi(k)]$

$$\begin{bmatrix} e_y(k+1) \\ e_\psi(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta_s \\ -k_r^2(k)\Delta_s & 1 \end{bmatrix} \begin{bmatrix} e_y(k) \\ e_\psi(k) \end{bmatrix} + \begin{bmatrix} 0) \\ \Delta_s \end{bmatrix} (k(k) - k_r(k)) \tag{31}$$

where $\Delta_s = vT_s$, T_s is the sampling time and v is a constant, i.e., vehicle traversal at constant speed, road curvature, denoted $k_r = \frac{1}{\rho_s}$. The state vector of the derived system $\bar{z}(k) = [e_y(k)\, e_\psi(k)]$ and the control inputs the system is given by $\bar{u}(k) = \bar{k} = k(k) - k_r(k)$

Lima, P. F., Mårtensson, J., Wahlberg, B. (2017, December). Stability conditions for linear time-varying model predictive control in autonomous driving. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC) (pp. 2775-2782). IEEE.

$$\min_{\bar{U}_t} \quad \sum_{k=t}^{t+N_e-1} \bar{z}_{k|t}^\top \mathbf{Q_k} \bar{z}_{k|t} + \bar{u}_{k|t}^\top \mathbf{R_k} \bar{z}_{k|t} + \bar{z}_{t+N_e|t}^\top \mathbf{Q_{N_e}} \bar{z}_{t+N_e|t}, \tag{32}$$

$$\begin{aligned} \text{s.t.} \quad & \bar{z}_{k+1|t} = \boldsymbol{A}_{k|t} \bar{z}_{k|t} + \boldsymbol{B}_k \bar{u}_{k|t}, \quad k = t, \ldots, t+N_e-1 \\ & \bar{z}_{t|t} \text{ is given,} \\ & \bar{z}_{k|t} \in \bar{Z}, \quad k = t+1, \ldots, t+N_e \\ & ,\bar{u}_{k|t} \in \bar{U} \quad k = t, \ldots, t+N_e-1 \end{aligned}$$

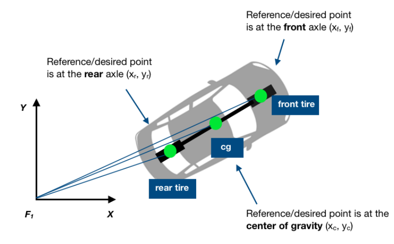
where $\bar{U}_t = \{\bar{u}(t|t), \ldots, \bar{u}(t+N_e-1|t)\}$, boundary constraints for states and controls, denoted $\bar{Z}$, $\bar{U}$

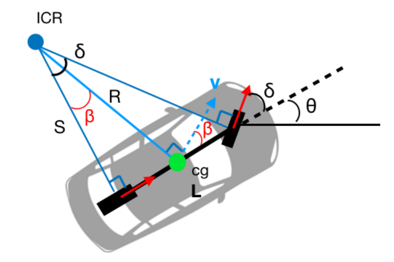
Path tracking control

- Model-free control: PID feedback controller, feedback and feedforward controller
- Model-based control: motion model (kinematics model)
- Differential flatness
- Model-predictive control: MPC, MPCC
- Linear quadratic regulator (LQR): Lateral control: controlling lateral error, rate of change of lateral error, heading error, and rate of change of heading with the control input: steering, acceleration and braking

PATH TRACKING CONTROL WITH MPC: KINEMATIC MODEL

The desired point is at the cg (center of gravity)





Assume that there is no side slippage, i.e., $\beta = 0$, therefore,

$$\dot{x} = vcos(\theta), \quad \dot{y} = vsin(\theta), \quad \dot{\theta} = \frac{v}{R} = \frac{vtan(\delta)}{L}$$

https://www.shuffleai.blog/blog/Simple_Understanding_of_Kinematic_Bicycle_Model.html

Path tracking control with MPC: trajectory generation

Obtain the next 6 waypoints of the reference path from the current pose of the vehicle, and use those 6 waypoints to fit a 3-rd order polynomial.

$$y = f(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$tan(\theta)_d = \frac{dy}{dx} = 3a_3x^2 + 2a_2x + a_1$$

Path tracking control with MPC: dynamic model

- $$x_{t+1} = x_t + v_t cos(\theta) dt$$
  $$y_{t+1} = y_t + v_t sin(\theta) dt$$
  $$\theta_{t+1} = \theta_t + \frac{v_t}{L} \delta_t dt$$
  $$v_{t+1} = v_t + a_t dt$$

- $$x_{t+1} = x_t + v_t cos(\theta) dt$$
  $$y_{t+1} = y_t + v_t sin(\theta) dt$$
  $$\theta_{t+1} = \theta_t + \frac{v_t}{L} \delta_t dt$$
  $$v_{t+1} = v_t + a_t dt$$
- Also, **the residual between the current pose and desired pose** is added to the model as part of the **states**:

$$e_{y,t+1} = f(x_t) - y_t + v_t sin(e_{\theta,t}) dt$$
$$e_{\theta,t+1} = \theta_t - \theta_{d,t} + \frac{v_t}{L} \delta_t dt$$

Path tracking control with MPC: cost

$$J = \Sigma_{t=1}^{N} \lambda_{res} \left\| e_{y,t} \right\|^2 + \lambda_{e_\theta} \left\| e_{\theta,t} \right\|^2 + \lambda_v \left\| v_t - v_{d,t} \right\|^2 + \lambda_\delta \left\| \delta_t \right\|^2 + \lambda_a \left\| a_t \right\|^2 + \lambda_{rate_a} \left\| a_t - a_{t-1} \right\|^2 + \lambda_{rate_\delta} \left\| \delta_t - \delta_{t-1} \right\|^2$$

- $\lambda_{res} \left\| e_{y,t} \right\|^2$ residual error
- $\lambda_{e_\theta} \left\| e_{\theta,t} \right\|^2$ heading error
- $\lambda_v \left\| v_t - v_{d,t} \right\|^2$ speed cost
- $\lambda_\delta \left\| \delta_t \right\|^2$ steering cost
- $\lambda_a \left\| a_t \right\|^2$ acceleration cost
- $\lambda_{rate_a} \left\| a_t - a_{t-1} \right\|^2$ acceleration rate cost
- $\lambda_{rate_\delta} \left\| \delta_t - \delta_{t-1} \right\|^2$ steering rate cost

where λ_* are weighting parameters

Path tracking control with MPC: MPC formulation

$$
\begin{aligned}
\min_{\mathbf{u}=a,\delta} \quad & J, \\
\text{s.t.} \quad & \boldsymbol{x}_{t+1} = f(x_t, y_t, \theta_t, v_t, e_{\theta,t}, \theta_{d,t}, e_{y,t}, a_t, dt), \quad t = 0, \ldots, N_e \\
& \boldsymbol{x}_0 = (x_0, y_0, \theta_0, v_0, e_{\theta,0}, e_{y,0}) \text{ is given}, \\
& -\pi/6 \le \delta \le \pi/6 \\
& a_{min} \le a \le a_{max}
\end{aligned} \tag{33}
$$