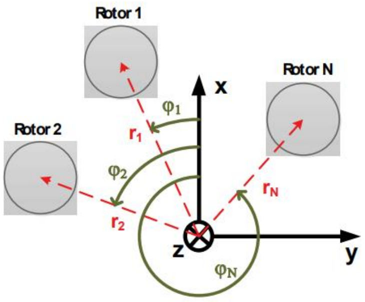
# Numerical Optimization in Robotics
# Homework_3 Instruction

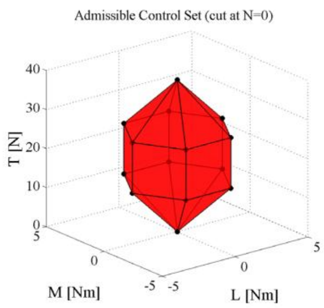
## 1. Problem description:

### ● KKT condition

You have to prove the question of the course in your homework report.

### ● Control Allocation Problem




Control allocation as constrained optimization

$$\min_{\nu_1,\dots,\nu_N} \left\| \begin{bmatrix} \bar{f}_{\text{total}} \\ \bar{\tau}_{\text{roll}} \\ \bar{\tau}_{\text{pitch}} \\ \bar{\tau}_{\text{yaw}} \end{bmatrix} - \begin{bmatrix} t_1 & \cdots & t_N \\ l_1 & \cdots & l_N \\ m_1 & \cdots & m_N \\ n_1 & \cdots & n_N \end{bmatrix} \begin{bmatrix} \bar{\omega}_1^2 + \nu_1 \\ \vdots \\ \bar{\omega}_N^2 + \nu_N \end{bmatrix} \right\|_W^2 + \rho \left\| \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_N \end{bmatrix} \right\|^2$$

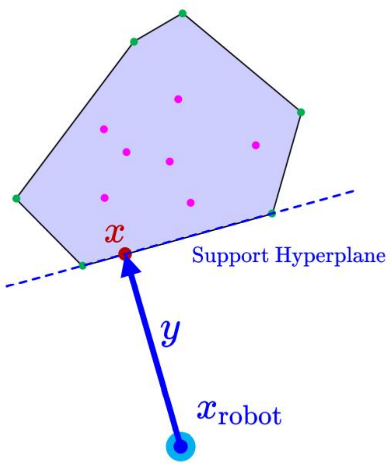
$$\text{s.t.} \quad \begin{bmatrix} \omega_{\min}^2 - \bar{\omega}_1^2 \\ \vdots \\ \omega_{\min}^2 - \bar{\omega}_N^2 \end{bmatrix} \le \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_N \end{bmatrix} \le \begin{bmatrix} \omega_{\max}^2 - \bar{\omega}_1^2 \\ \vdots \\ \omega_{\max}^2 - \bar{\omega}_N^2 \end{bmatrix}$$

$$A \begin{bmatrix} t_1 & \cdots & t_N \\ l_1 & \cdots & l_N \\ m_1 & \cdots & m_N \\ n_1 & \cdots & n_N \end{bmatrix} \begin{bmatrix} \bar{\omega}_1^2 + \nu_1 \\ \vdots \\ \bar{\omega}_N^2 + \nu_N \end{bmatrix} \le b$$

This is a strictly convex low-dim QP as long as N<10, whose exact solution can be obtain in linear time.

### ● Collision Distance Computation

Collision vector from a robot to a polytope obstacle: ***V-rep*** cases




Any collision vector is the normal of its support hyperplane.

Due to the convexity, the following two problems are equivalent

Minimize length of the V-polytope's collision vector

Maximize length of the hyperplane's normal vector

Obviously, the separating halfspace is $\{x \in \mathbb{R}^d \mid y^{\mathrm{T}}(x - x_{\text{robot}}) \le y^{\mathrm{T}} y\}$

$$\max_{y \in \mathbb{R}^d} y^{\mathrm{T}} y, \quad \text{s.t.} \ (v_i - x_{\text{robot}})^{\mathrm{T}} y \ge y^{\mathrm{T}} y, \ \forall i \in \{1, \dots, m\}$$

Notice that if we use $z = y/(y^{\mathrm{T}} y)$ or equivalently $y = z/(z^{\mathrm{T}} z)$

$$\min_{z \in \mathbb{R}^d} z^{\mathrm{T}} z, \quad \text{s.t.} \ (v_i - x_{\text{robot}})^{\mathrm{T}} z \ge 1, \ \forall i \in \{1, \dots, m\}$$

Infeasibility implies collision occurs, else $x = y + x_{\text{robot}} = z/(z^{\mathrm{T}} z) + x_{\text{robot}}$

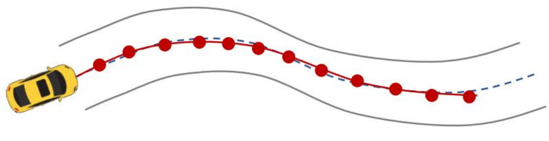
This is a low-dim QP again !

Complete the code for Lec_3_homework_part_2, make sure it could work.

**Optional: you can implement the low-dim QP to solve the CAP or CDC problem.**

- ## Nonlinear Model Predict Control

Example: Control longitudinal acceleration and steering angle of the vehicle simultaneously for autonomous driving of tracking a reference trajectory



Time-discretized state and control input:

$$s_{k+1} = F(s_k, u_k) := f(s_k, u_k)\tau + s_k$$

where

$$s_k = \begin{bmatrix} x_k \\ y_k \\ \phi_k \\ v_k \end{bmatrix},\ u_k = \begin{bmatrix} a_k \\ \delta_k \end{bmatrix},\ \tau \text{ is a constant duration}$$

Our objective function is to minimize tracking error and energy:

$$J(s_1, \ldots, s_N, u_0, \ldots, u_N) := \sum_{k=1}^{N} \left[ (x_k - x_k^{\text{ref}})^2 + (y_k - y_k^{\text{ref}})^2 + w_v(a_k - a_{k-1})^2 + w_\delta(\delta_k - \delta_{k-1})^2 \right]$$

Physical limits requires

$$\begin{aligned} &\forall\, k \in \{0, \ldots, N\} \\ &a_{\min} \le a_k \le a_{\max} \\ &\delta_{\min} \le \delta_k \le \delta_{\max} \\ &v_{\min} \le v_k \le v_{\max} \end{aligned} \qquad G(s_k, u_k) \le 0$$

You have to implement PHR-ALM to solve the MPC problem given above.

## 2. Problem analysis

***Model Predict Control:*** MPC is an iterative process of optimizing the predictions of robot states in the future limited horizon while manipulating inputs for a given horizon. The forecasting is achieved using the process model. Thus, a dynamic model is essential while implementing MPC. These process models are generally nonlinear, but for short periods of time, there are methods such as tailor expansion to linearize these models.

***For more:***

*https://www.bilibili.com/video/BV1cL411n7KV?spm_id_from=333.999.0.0*
*https://www.bilibili.com/video/BV16U4y1c7EG?spm_id_from=333.999.0.0&vd_source=a1c2e0e10d18e90004496c6ea8093833*

## 3. Assignment requirements

- ✓ Your homework ***should*** be a ***zip*** including your code, an documentation and an instruction.
- ✓ You can complete this chapter assignment in different programming languages, but I suggest you apply ***MATLAB*** or ***C++*** for this homework.
- ✓ You ***must*** given an instruction named ***'readme'*** to tell the reader how to run your code and check your answer.
- ✓ You ***have to*** give an report for this assignment which includes (1) the ***workflow and result*** of your homework; (2) your ***analysis*** of the homework; (3) any ***question or suggestion*** of the course and the homework.
- ✓ You can add your notes of this course to your homework.

## 4. Scoring Criteria

***Unqualified:*** The results are incorrect, or the assignment is not written in the required format.

***Qualified:*** Finish one of the first or second task completely.

***Good:*** Finish both the first and the second tasks completely or complete the MPC project correctly.

***OutStanding:*** Finish all the task completely!