Commande optimale

1. Introduction (fin)

ω w

$\vec{V} = \cancel{V}\,1\,(\cos\theta, \sin\theta)$ "cap" du bateau

vitesse dans le repère eau (= sur l'eau)

i.e. : $\begin{bmatrix}\dot x(t)\\ \dot y(t)\end{bmatrix} - \begin{bmatrix} w\\ 0\end{bmatrix} = \begin{bmatrix}\cos\theta(t)\\ \sin\theta(t)\end{bmatrix} \simeq e^{i\theta(t)}$

$\left(\theta(t) = \arg\left[(\dot x(t)-w) + i\dot y(t)\right]\right)$

De plus :

$\dot\theta(t) = u(t)$, $|u(t)| \leq 1$ virages à courbure limitée

$x(0)=y(0)=0$, $\theta(0)=\theta_0$; $x(t_f)=x_f$, $y(t_f)=y_f$, $\theta(t_f)=\theta_f$

Coût :

$t_f \to \min$

Il s'agit d'un problème de commande optimale de type Lagrange (ou Mayer) : posons $X := (x, y, \theta) \in \mathbb{R}^3$ $(n=3)$, $U := [-1, 1]$; alors,

$$\dot{X}(t) = f(X(t), u(t)), \quad t \in [0, t_f] \ (\text{p.p.})$$
$$X(0) = X_0, \quad X(t_f) = X_f$$
$$u(t) \in U$$

avec $f : \mathbb{R}^3 \times \mathbb{R} \longrightarrow \mathbb{R}^3$

$$(X, u) = (\underbrace{x, y, \theta}_{X}, u) \longmapsto \begin{bmatrix} w + \cos\theta \\ \sin\theta \\ u \end{bmatrix}, \quad X_0 := (x_0, y_0, \theta_0), \quad X_f := (x_f, y_f, \theta_f)$$

Et, $f^0(X, u) := 1$ (cf. temps min). t_f libre

$$\left(\int_0^{t_f} \overbrace{f^0(X(t), u(t))}^{1}\, dt = t_f \ !\right)$$

On admet :

- la commandabilité, sous l'hypothèse $0 \leq w < 1$ (pb dit de "Zermelo" à courant "faible", ie que le contrôle peut compenser ; a contrario, si $w = 1$, $\dot{x}(t) = 1 + \cos\theta(t) \geq 0$: on ne peut atteindre $x_f < x_0$...)
- l'existence de solution (conséquence des propriétés de compacité et de convexité du problème : Th. Filippov, Th. d'Ascoli...)

On verra au Ch. 3 - PMP comment préciser mathématiquement l'allure des solutions (x, u, t_f), et au Ch. 2 comment les calculer numériquement.

2. Méthodes directes

Soit à résoudre le problème de Lagrange suivant :

$$\left\{\begin{array}{l} \displaystyle\int_0^{tf} f^0(x(t),u(t))\,dt \longrightarrow \min \\ \dot{x}(t) = f(x(t),u(t)),\ t\in[0,tf]\ (p.p.) \\ x(0)=x_0,\ x(tf)=x_f \\ u(t)\in U \end{array}\right.$$

$$\boxed{\begin{array}{l} x(t)\in\mathbb{R}^n \\ u(t)\in\mathbb{R}^m \end{array}}$$

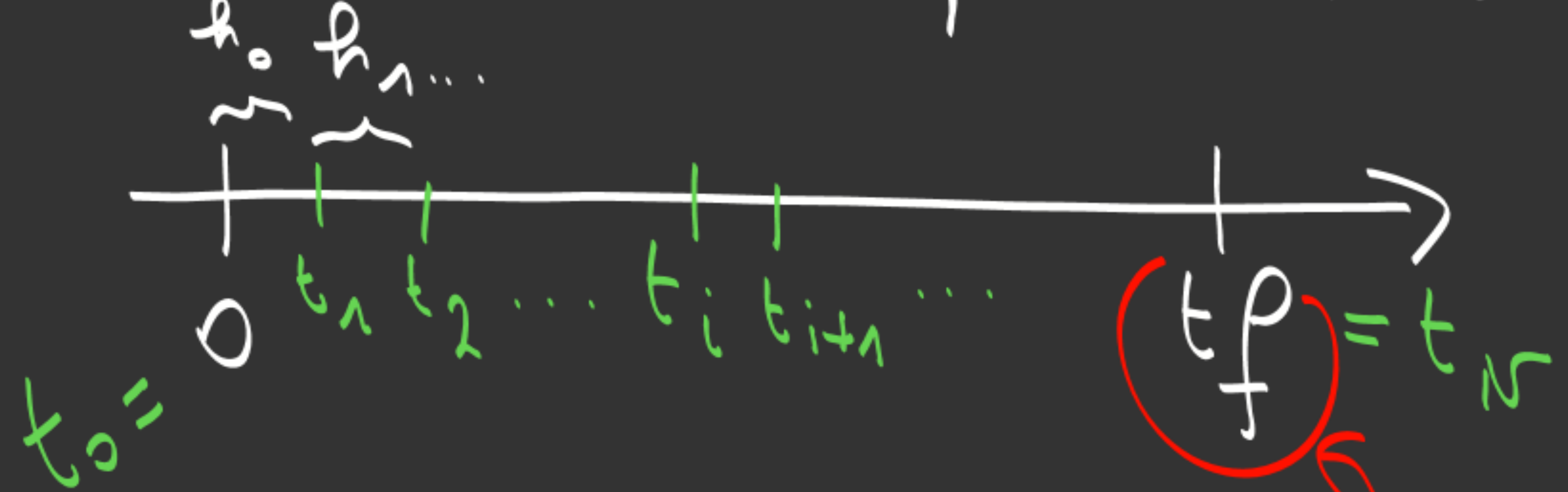
$\rightarrow$ ou, plus généralement $h(x(t_0),x(tf))=0$

$h : \mathbb{R}^n\times\mathbb{R}^n \longrightarrow \mathbb{R}^q$

$\rightarrow \Leftrightarrow k(u(t))\leq 0$

(ie $U=\{u\in\mathbb{R}^m \mid k(u)\leq 0\}$, $k:\mathbb{R}^m\longrightarrow\mathbb{R}^r$)

On discrétise "brutalement" ce problème (de dimension infinie) en un problème d'optimisation en <u>dim finie</u> (on dit encore "math programming" — NLP = NonLinear Program / LP = Linear Program ; "Program" = pb d'optimisation) en utilisant une grille de temps discrets :



éventuellement inconnu

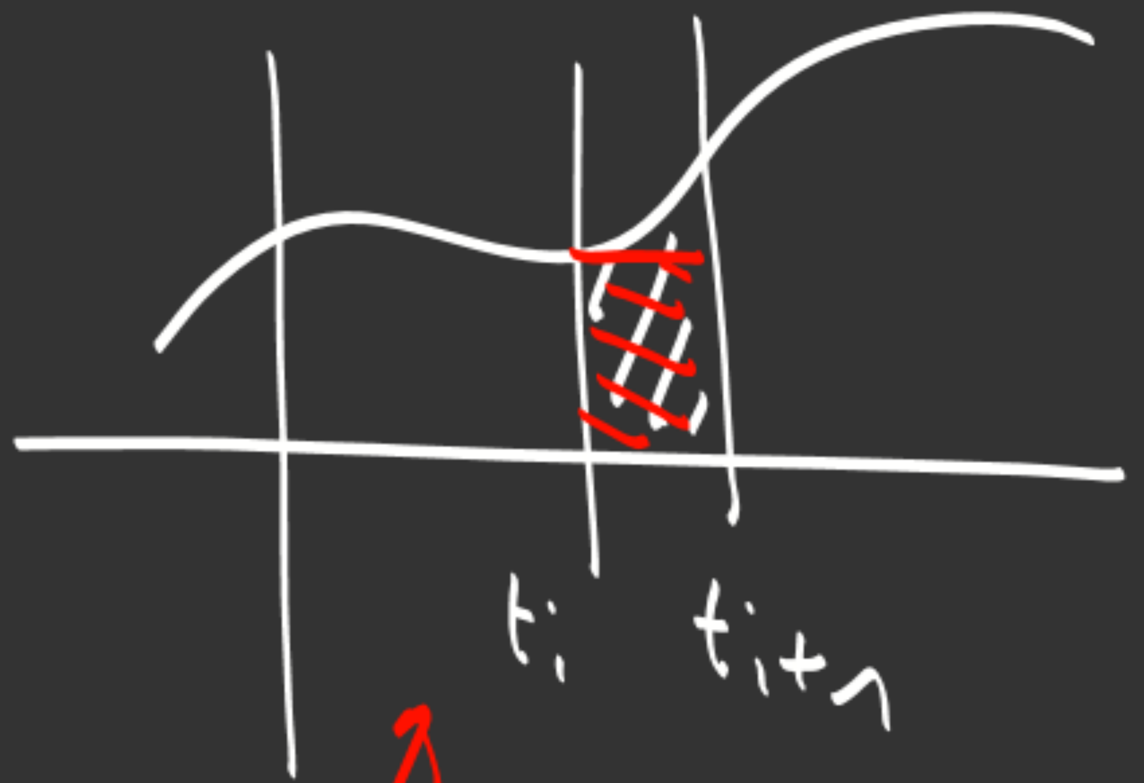
$$(t_0, t_1, \dots, t_N) \in \mathbb{R}^{N+1}$$

$$h_i := t_{i+1} - t_i, \quad i = 0, \dots, N-1$$

ième pas de temps

<u>Rq</u> : par exemple grille uniforme :
$$h_i = h = \frac{t_f - t_0}{N}, \quad i = 0, \dots, N.$$

Etant donnée cette grille de temps, on dispose de tous les schémas usuels (à un pas – Runge-Kutta –, multipas...), par exemple le schéma Euler (explicite/progressif) : on approche la solution de l'EDO (paramétrée par le contrôle $u : [0, t_f] \longrightarrow \mathbb{R}^m$) selon



$$
\begin{aligned}
x(t_{i+1}) &= x(t_i) + \int_{t_i}^{t_{i+1}} \dot{x}(t)\,dt \\
&= x(t_i) + \int_{t_i}^{t_{i+1}} f(x(t), u(t))\,dt \\
&\simeq x(t_i) + h_i \cdot f(x(t_i), u(t_i))
\end{aligned}
$$

consistance

Euler explicite

$\Longrightarrow$ schéma : $x_{i+1} = x_i + h_i \cdot f(x_i, u_i)$ avec $x_i \simeq x(t_i)$, $u_i \simeq u(t_i)$.
$i = 0, \dots, N-1$

On utilise (par souci de cohérence numérique — on pourrait augmenter la dynamique pour y adjoindre le terme de Lagrange) la même quadrature pour approcher le coût intégral :

$$\int_0^{tf} f^0(x(t),u(t))\,dt \simeq \sum_{i=0}^{N-1} h_i \cdot f^0(x_i,u_i) \longrightarrow \min$$

Enfin : $h(x(0),x(tf))=0 \longrightarrow h(x_0,x_N)=0$

$u(t)\in U,\ t\in[0,tf] \longrightarrow g_k(u_i)\leq 0,\ i=0,\dots,N$

ie $g_k(u(t))\leq 0$

On se ramène ainsi à résoudre un problème (approximation du précédent — convergence à étudier !) d'optimisation en dim finie :

$$\begin{cases} F(Z) \longrightarrow \min \\ H(Z) = 0 \\ G(Z) \leq 0 \end{cases}$$

si tf libre

avec $Z := (x_0, \dots, x_N, u_0, \dots, u_N, \underset{\uparrow}{tf}) \in \mathbb{R}^{(N+1)(n+m)+1}$

$$F(Z) := \sum_{i=0}^{N-1} h_i \, f^0(\underbrace{x_i}_{Z_i}, \overbrace{u_i}^{Z_{i+N+1}})$$

$$H(Z) := \begin{bmatrix} x_1 - x_0 - h_0 \, f(x_0, u_0) \\ \vdots \\ x_N - x_{N-1} - h_{N-1} f(x_{N-1}, u_{N-1}) \\ h(x_0, x_N) \end{bmatrix} \in \mathbb{R}^{N \cdot n + q}, \quad G(Z) := \begin{bmatrix} k(u_0) \\ \vdots \\ k(u_N) \end{bmatrix} \in \mathbb{R}^{(N+1)n}$$

Au final, on a donc un pb de dimension (inconnues + contraintes égalités & inégalités)

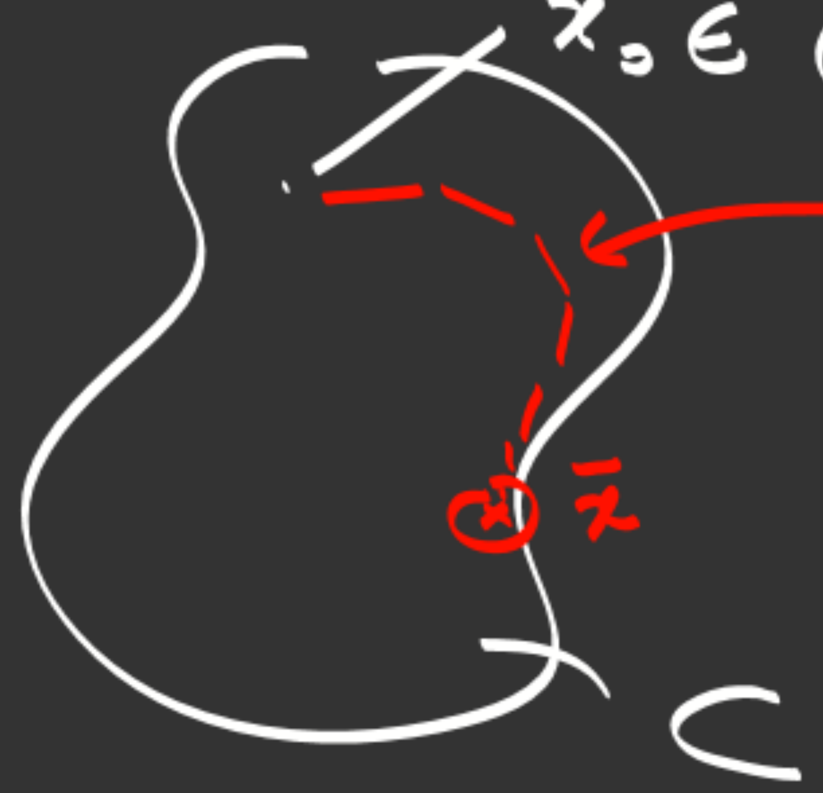
$$(N+1)(n+m) + 1 + N.n + q + (N+1)r$$
$$= O(\underline{\underline{N}}(2n+m+r))$$

Rq : le pb d'opti a une structure creuse (block-sparse) (= beaucoup de zéros dans les dérivées) ; cf. :

$$H'(z) = \begin{bmatrix} \overset{n\times n}{\boxed{\frac{\partial H_0}{\partial x_0}}} & \frac{\partial H_0}{\partial x_1} & 0 \cdots 0 & \frac{\partial H_0}{\partial u_0} & 0 \cdots \\ & \ddots & \ddots & & \ddots \\ 0 \cdots 0 & \frac{\partial H_{N-1}}{\partial x_{N-1}} & \frac{\partial H_{N-1}}{\partial x_N} & 0 & \frac{\partial H_{N-1}}{\partial u_{N-1}} \\ \frac{\partial h}{\partial x_0} & 0 \cdots & \frac{\partial h}{\partial x_N} & 0 \cdots & 0 \end{bmatrix}, \quad G'(z) = \begin{bmatrix} \bigcirc & \begin{matrix} \overset{r\times m}{\boxed{k'(u_0)}} & & 0 \\ & \ddots & \\ 0 & & k'(u_N) \end{matrix} \end{bmatrix}$$

Rq. : Ipopt (= Interior Point Solver)

$$\begin{cases} f(x) \longrightarrow \min \\ x \in C \subset \mathbb{R}^n \end{cases}$$



$$\hookrightarrow \quad f(x) + \varepsilon \overbrace{b_C(x)}^{\xrightarrow{x \to \partial C} \infty} \longrightarrow \min$$

+ itération sur ε ($\longrightarrow 0+$)

fonction barrière

• Autre approche : KNITRO ($\to$ SQP = Sequential Quadratic Prog.)