Commande optimale des systèmes dynamiques
Méthodes numériques de résolution

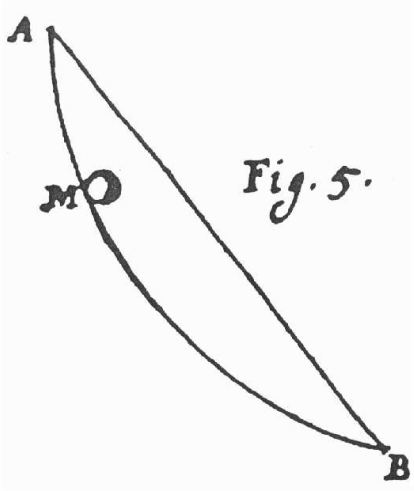


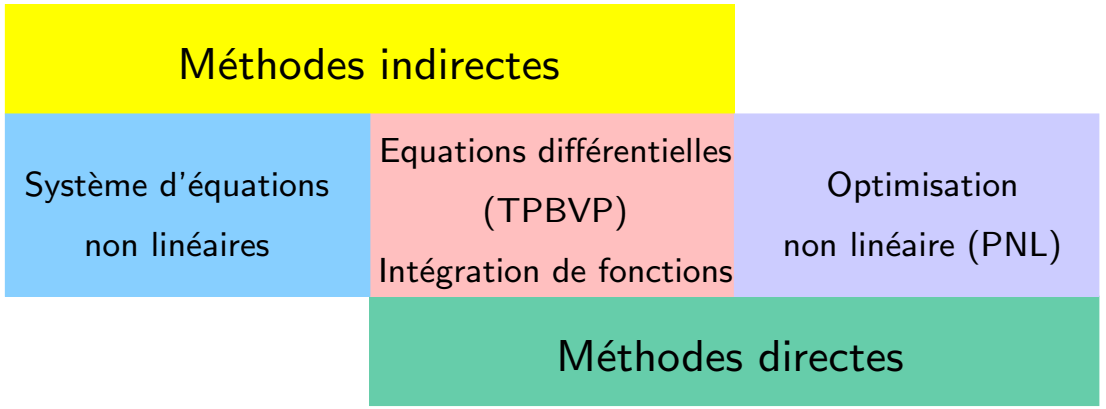
Fig. 5.

Problème de Commande Optimale (PCO) :

$$\begin{aligned} \min_{u\in\mathcal{U}} \quad & J(x,u) = \int_{t_0}^{t_f} L(t,x(t),u(t))dt + \psi_0(t_f,x(t_f)) \\ \text{sous} \quad & \dot{x}(t) = f(t,x(t),u(t)) \\ & \psi(t_0,x(t_0),t_f,x(t_f)) = 0 \quad t_0,\ t_f \text{ libres ou fixés} \\ & \underline{h} \leq h(x,u,t) \leq \overline{h} \end{aligned}$$

✔ Méthodes Indirectes (MI) : CV + PMP : $H \geq H^* \rightarrow$ CNO : $H_\lambda = \dot{x},\ H_x = -\dot{\lambda}^*(t) \rightarrow$ TPBVP : $\dot{z} = F(t,z),\ F_0(z(0),z(t_f)) = 0 \rightarrow$ extrêmales : (x^*,u^*)

✔ Méthodes Directes (MD) : PCO $\rightarrow$ Etat et/ou commande discrétisés : $(x(t_k),u(t_k)) \rightarrow$ PNL : $\min_y\ J(y)$ sous $y \in Y \rightarrow$ solutions sous-optimales : $(\hat{x},\hat{u})$



Les méthodes indirectes (I)

Problème de Commande Optimale (PCO) :

$$\begin{aligned} \min_{u\in\mathcal{U}} \quad & J(x,u) = \int_{t_0}^{t_f} L(t,x(t),u(t))dt + \psi_0(t_f,x(t_f)) \\ \text{sous} \quad & \dot{x}(t) = f(t,x(t),u(t)),\ x(t_0) = x_0 \\ & \psi(t_f,x(t_f)) = 0,\ t_f \text{ libre ou fixé} \end{aligned}$$

CN d'optimalité au premier ordre $H(t,x,\lambda,u) = L(t,x,u) + \lambda^T f(t,x,u)$

❶ Equations canoniques de Hamilton + Equation de commande

$$\dot{x}^*(t) = H_\lambda(t,x^*,\lambda^*,u^*) \quad \dot{\lambda}^*(t) = -H_x(t,x^*,\lambda^*,u^*) \quad u^*(t) = \min_{u\in\mathcal{U}} H(t,x,\lambda,u)$$

❷ Conditions de transversalité

$$\begin{aligned} & x(t_0) - x_0 = 0,\ \psi(t_f^*,x^*(t_f)) = 0 \\ & \left[\nabla_{x_f}\psi_0^* + \psi_{x_f}^{*T}\nu - \lambda^*\right]_{|t_f} = 0 \\ & \left[H^* + \nu^T\psi_{t_f}^* + \psi_{0t_f}^*\right]_{|t_f} = 0 \text{ si } t_f \text{ libre} \end{aligned}$$

Les méthodes indirectes (II)

Principe général :

1. Exprimer $u^*(t) = u(x^*, \lambda^*)$
2. Déduire le TPBVP $\dot{z} = F(t,z),\ z(t_0) = z_0$ avec

$$z^T = \begin{bmatrix} x^T & \lambda^T \end{bmatrix}^T \quad F(t,z)^T = \begin{bmatrix} H_\lambda^T & -H_x^T \end{bmatrix}^T \quad z_0^T = \begin{bmatrix} x_0^T & \lambda(t_0)^T \end{bmatrix}^T$$

3. Initialiser $\lambda(t_0) = \lambda_0,\ t_f,\ \nu$
4. Trouver itérativement les racines de $F_0(z_0, z(t_f), \nu) = 0$ sous les contraintes $\dot{z} = F(t,z),\ z(t_0) = z_0$

$$\text{Résoudre} \quad F_0(\lambda_0, x(t_f), \lambda(t_f), \nu) = \begin{bmatrix} \psi(t_f^*, x^*(t_f)) \\ \left[\nabla_{x_f}\psi_0^* + \psi_{x_f}^{*T}\nu - \lambda^*\right]_{|t_f} \\ \left[H^* + \nu^T\psi_{t_f}^* + \psi_{0t_f}^*\right]_{|t_f} \end{bmatrix} = 0$$

$$\text{sous} \quad \dot{x} = H_\lambda(t,x,\lambda),\ x(t_0) = x_0$$
$$\dot{\lambda} = -H_x(t,x,\lambda),\ \lambda(t_0) = \lambda_0$$

Nota : 2 ensembles de CN - les contraintes satisfaites à chaque étape et les contraintes satisfaites à la fin du processus itératif

Algorithme 1 *(Méthode de Newton)*

❶ $k \leftarrow 0$; **Initialiser** λ_0^k, t_f^k, ν^k, $\epsilon > 0$;

❷ **Tant que** $\|F_0(\lambda_0^k, x(t_f)^k, \lambda(t_f)^k, \nu^k)\| > \epsilon$ **alors**

4.1 **Calculer** $x(t_f)^k$, $\lambda(t_f)^k$;

4.2 **Calculer** $F_0(\lambda_0^k, x(t_f)^k, \lambda(t_f)^k, \nu^k)$;

4.3 **Calculer** *la matrice gradient* $\nabla F_0^k = \left[\begin{array}{ccc} \nabla_{\lambda_0} F_0^k & \nabla_\nu F_0^k & \nabla_{t_f} F_0^k \end{array} \right]$;

4.4 **Calculer** *la direction de descente* $d_k^T = \left[\begin{array}{ccc} d_\lambda^{kT} & d_\nu^{kT} & d_{t_f}^k \end{array} \right]^T$ *solution de*

$$\left[\begin{array}{c} \nabla_\lambda F_0(\lambda_0^k, x(t_f)^k, \lambda(t_f)^k, \nu^k)^T \\ \nabla_\nu F_0(\lambda_0^k, x(t_f)^k, \lambda(t_f)^k, \nu^k)^T \\ \nabla_{t_f} F_0(\lambda_0^k, x(t_f)^k, \lambda(t_f)^k, \nu^k)^T \end{array} \right]^T \left[\begin{array}{c} d_\lambda^k \\ d_\nu^k \\ d_{t_f}^k \end{array} \right] = -F_0(\lambda_0^k, x(t_f)^k, \lambda(t_f)^k, \nu^k)$$

4.5 **Actualiser** $\lambda_0^{k+1} \leftarrow \lambda_0^k + d_\lambda^k$, $\nu^{k+1} \leftarrow \nu^k + d_\nu^k$, $t_f^{k+1} \leftarrow t_f^k + d_{t_f}^k$, $k+1 \leftarrow k$;

❸ **Fin Tant que**

Nota : L'étape 4.1 consiste à appliquer un schéma numérique d'intégration du TPBVP

Méthode de tir simple indirecte : exemple (I)

Exemple 1 *[Trélat 2008] Le problème de commande optimale étudié est défini par :*

$$\begin{array}{ll} \min\limits_{|u(t)|\leq 1} & J = t_f \\ \text{sous} & \dot{x}_1(t) = x_2(t),\ x_1(0) = 0,\ x_2(0) = 0 \\ & \dot{x}_2(t) = u(t),\ x_1(t_f) = x_{1f},\ x_2(t_f) = x_{2f} \end{array}$$

- *Hamiltonien :* $H(x, u, \lambda) = \lambda_1(t)x_2(t) + \lambda_2(t)u(t)$
- *Equations canoniques de Hamilton :*

$$\begin{aligned} \dot{x}_1^*(t) &= x_2^*(t),\ x_1^*(0) = 0,\ x_2^*(0) = 0 \\ \dot{x}_2^*(t) &= u^*(t),\ x_1^*(t_f) = x_{1f},\ x_2^*(t_f) = x_{2f} \\ \dot{\lambda}_1^*(t) &= 0, \\ \dot{\lambda}_2^*(t) &= -\lambda_1^*(t) \end{aligned}$$

- *Principe de Pontryagin :*

$$u^*(t) = -\text{sign}(\lambda_2^*(t)) = \begin{cases} -1 & \text{si} \quad \lambda_2^*(t) > 0 \\ 1 & \text{si} \quad \lambda_2^*(t) < 0 \\ \text{singulière} & \text{si} \quad \lambda_2^*(t) = 0 \end{cases}$$

Méthode de tir simple indirecte : exemple (II)

- Analyse des arcs singuliers : $\text{rang}\left(\left[\begin{array}{cc} B & AB \end{array}\right]\right) = 2$, pas d'arc singulier (Th. 5, C. 2)
- Existence et unicité de la commande optimale : (Th. 6 et Th. 7, C. 2)
- Structure de la commande optimale : au + une commutation en τ (Th. 6, C. 2)
- Les séquences optimales : $u^*(t) = \{1\}, \{-1\}, \{+1,-1\}, \{-1,+1\}$

- $u^*(t) = \{+1,-1\}$, $\lambda_1^*(t) = \dfrac{-2}{\sqrt{2x_{2f}^2+4x_{1f}}}$, $\lambda_2^*(t) = -\lambda_1^* t - 1$,

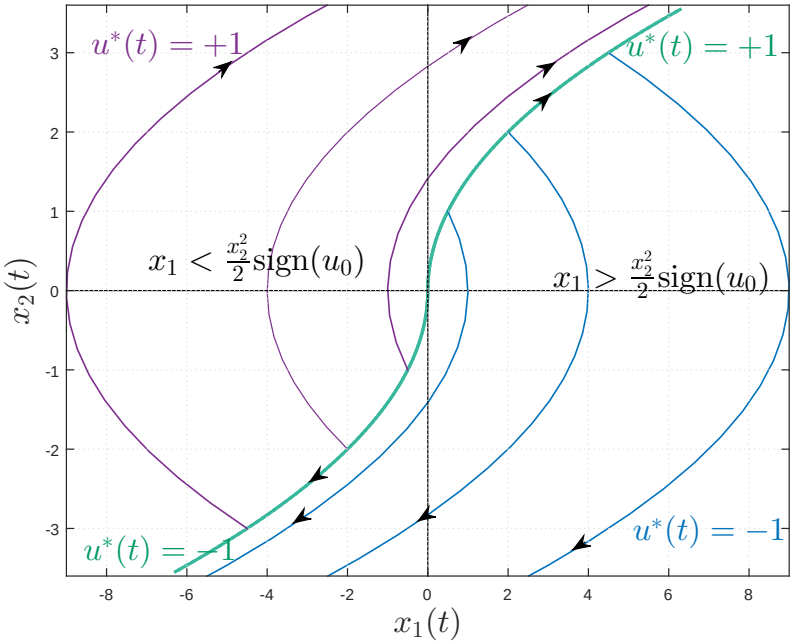
$$t_f^* = -x_{2f} + \sqrt{2x_{2f}^2 + 4x_{1f}}, \tau = \frac{\sqrt{2x_{2f}^2+4x_{1f}}}{2}$$

$$\begin{array}{ll} 0 \le t \le \tau & x_1^*(t) = \frac{t^2}{2} \\ & x_2^*(t) = t \\ \tau \le t \le t_f & x_1^*(t) = -\frac{t^2}{2} + 2\tau t - \tau^2 \\ & x_2^*(t) = -t + 2\tau \end{array}$$

- $u^*(t) = \{-1,+1\}$, $\lambda_1^*(t) = \dfrac{2}{\sqrt{2x_{2f}^2-4x_{1f}}}$, $\lambda_2^*(t) = -\lambda_1^* t + 1$,

$$t_f^* = x_{2f} + \sqrt{2x_{2f}^2 - 4x_{1f}}, \tau = \frac{\sqrt{2x_{2f}^2-4x_{1f}}}{2}$$

$$\begin{array}{ll} 0 \le t \le \tau & x_1^*(t) = -\frac{t^2}{2} \\ & x_2^*(t) = -t \\ 0 \le t \le \tau & x_1^*(t) = \frac{t^2}{2} - 2\tau t + \tau^2 \\ & x_2^*(t) = t - 2\tau \end{array}$$



Méthode de tir simple indirecte : exemple (III)

❶ TPBVP : $\dot{z} = F(t,z),\ z(t_0) = z_0$ avec $z^T = \left[\begin{array}{cc} x^T & \lambda^T \end{array}\right]^T$, $z_0^T = \left[\begin{array}{cc} x_0^T & \lambda(t_0)^T \end{array}\right]^T$

et $F(t,z)^T = \left[\begin{array}{cccc} x_2 & -\text{sign}(\lambda_2) & 0 & -\lambda_1 \end{array}\right]^T$

```
function [zdot] = double_int(t,z)
% Système double intégrateur
zdot=[z(2);-sign(z(4));0;-z(3)];
end
```

❷ Fonction de tir

$F_0(z_0, z(t_f)) = \left[\begin{array}{c|c|c} x_1(t_f) - x_{1f} & x_2(t_f) - x_{2f} & \lambda_1(t_f)x_2(t_f) - |\lambda_2(t_f)| + 1 \end{array}\right]^T = 0$

```
function [Zzero] = F_0(Z)
% Fonction F0(lambda,tf)
x0=[0;0];
zf=[1;-2];
options = odeset('RelTol',1e-12,'AbsTol',1e-12);
[t,z]=ode113(@double_int,[0;Z(3)],[x0;Z(1);Z(2)],options);
Zzero=[z(end,1)-zf(1);z(end,2)-zf(2);z(end,3)*z(end,2)-abs(z(end,4))+1];
end
```

❸ Calcul des zéros de la fonction de tir

```
function [lambda0_tf,fval,flag] = tir_simple_double_integrator(x0,xf)
% Fonction de tir simple
clear all
lambda0=[-2;-2];tf=10;
x0=[0;0];
options=optimoptions('fsolve','display','iter');
[lambda0_tf,fval,flag]=fsolve(@F_0,[lambda0;tf]',options);
options_ode=odeset('abstol',1e-9,'reltol',1e-9);
[t,z]=ode45(@double_int,...
[0;lambda0_tf(3)],[x0;lambda0_tf(1);lambda0_tf(2)],options_ode);
plot(z(:,1),z(:,2))
end
```

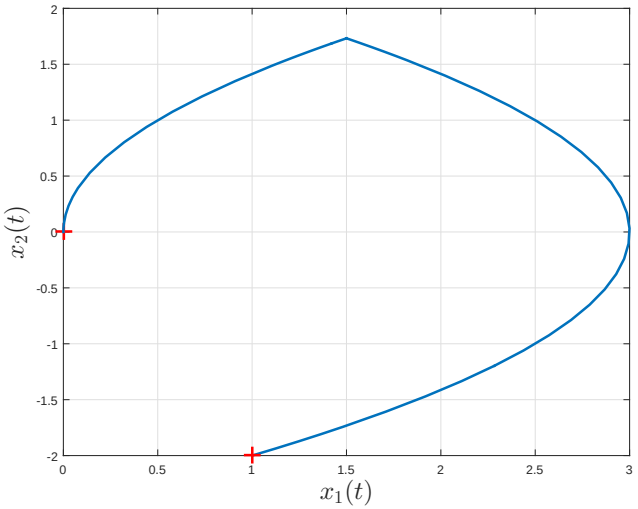
```
>> [lambda0_tf,fval,flag] = tir_simple_double_integrator([0;0],[0;-1])

lambda0_tf =

   -0.5774   -1.0000    5.4641
```

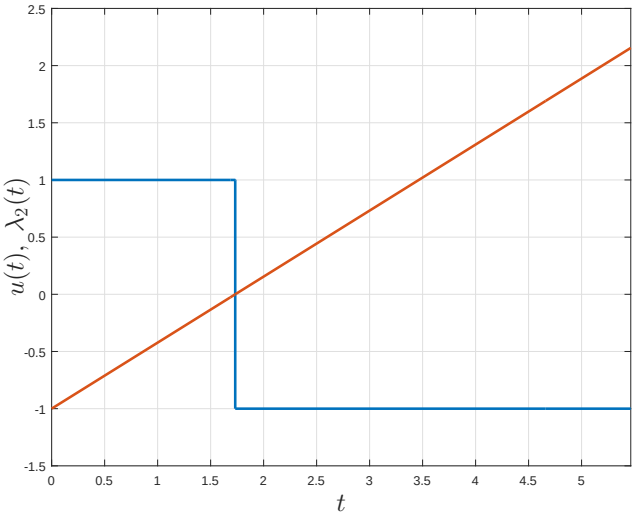
Méthode de tir simple indirecte : exemple (V)

$$u^*(t) = \{+1, -1\}\,,\ \tau = \frac{\sqrt{2x_{2f}^2+4x_{1f}}}{2} = \sqrt{3}$$
$$t_f = -x_{2f} + \sqrt{2x_{2f}^2 + 4x_{1f}} = 2 + 2\sqrt{3}\ \text{s},$$

$$0 \le t \le \sqrt{3} \qquad x_1^*(t) = \frac{t^2}{2}$$
$$x_2^*(t) = t$$

$$\sqrt{3} \le t \le t_f \qquad x_1^*(t) = -\frac{t^2}{2} + 2\sqrt{3}t - 3$$
$$x_2^*(t) = -t + 2\sqrt{3}$$



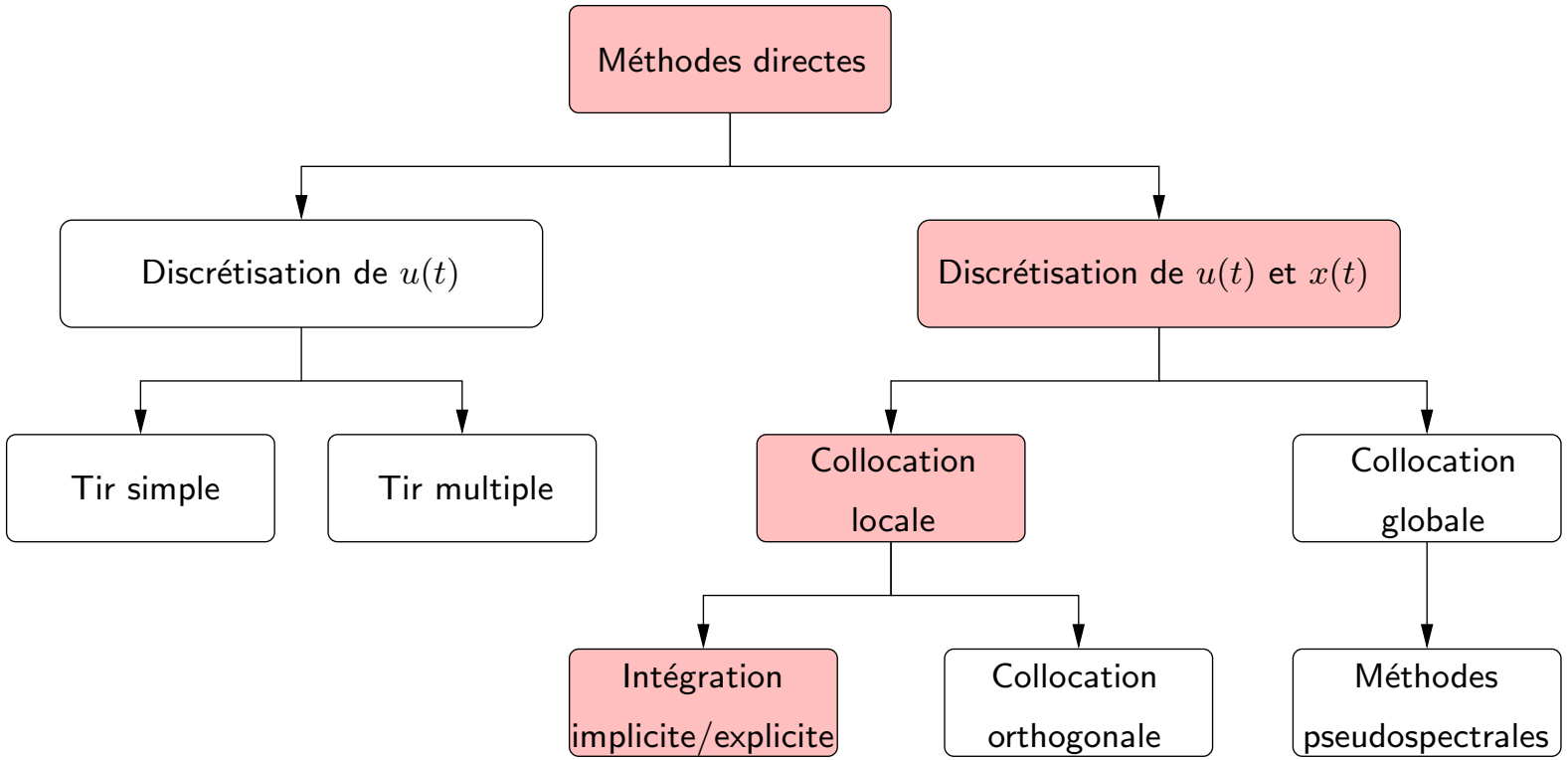


$$u^*(t) = \{+1, -1\}\,,\ \tau = \frac{\sqrt{2x_{2f}^2+4x_{1f}}}{2} = \sqrt{3}$$
$$t_f = -x_{2f} + \sqrt{2x_{2f}^2 + 4x_{1f}} = 2 + 2\sqrt{3},$$

$$\lambda_1^*(t) = -\frac{2}{\sqrt{2x_{2f}^2+4x_{1f}}} = -\frac{1}{\sqrt{3}}$$
$$\lambda_2^*(t) = -\lambda_1^* t - 1 = \frac{1}{\sqrt{3}} - 1$$

Problème de Commande Optimale (PCO) :

$$\begin{aligned}
\min_{u\in\mathcal{U}} \quad & J(x,u) = \int_{t_0}^{t_f} L(t,x(t),u(t))dt + \psi_0(t_f,x(t_f)) \\
\text{sous} \quad & \dot{x}(t) = f(t,x(t),u(t)),\ x(t_0) = x_0 \\
& \psi(t_f,x(t_f)) = 0 \quad t_0,\ t_f \text{ libres ou fixés} \\
& g(x,u,t) \leq 0
\end{aligned}$$

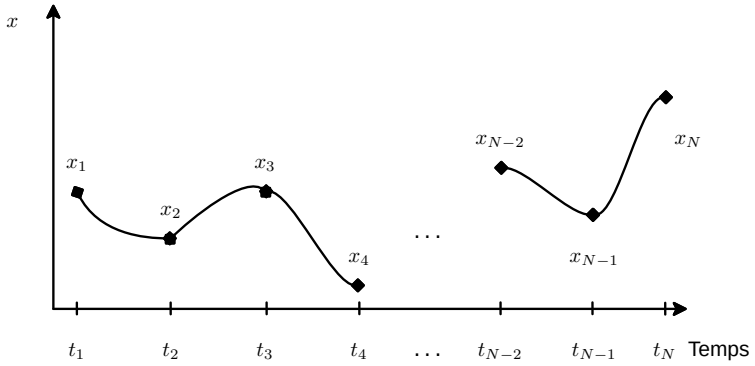


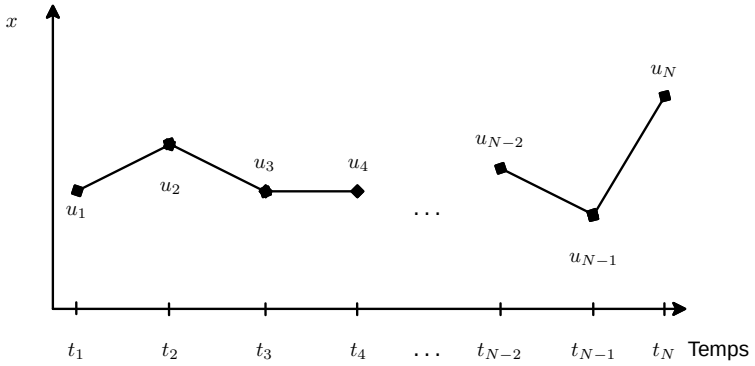
Méthode directe par collocation locale

1. Discrétisation d'un horizon fini : $t_0 = t_1 \leq t_2 \leq \cdots \leq t_{N-1} \leq t_N = t_f$
2. Les variables d'optimisation sont les états et les commandes aux instants $\{t_i\}$

$$Y = \left[\begin{array}{ccccccccccc} x_1^T & x_2^T & \cdots & x_{N-1}^T & x_N^T & u_1^T & u_2^T & \cdots & u_{N-1}^T & u_N^T & t_N \end{array} \right]^T \in \mathbb{R}^{N(n+m)+1}$$

3. $x(t)$ et $u(t)$, $t_i < t < t_{i+1}$ sont interpolés par des fonctions polynomiales (état) et linéaires (commande) $\hat{u}(t) = u_i + \dfrac{t - t_i}{t_{i+1} - t_i}(u_{i+1} - u_i)$





4. Résolution du problème de PNL

$$\min_{Y} F(Y)$$

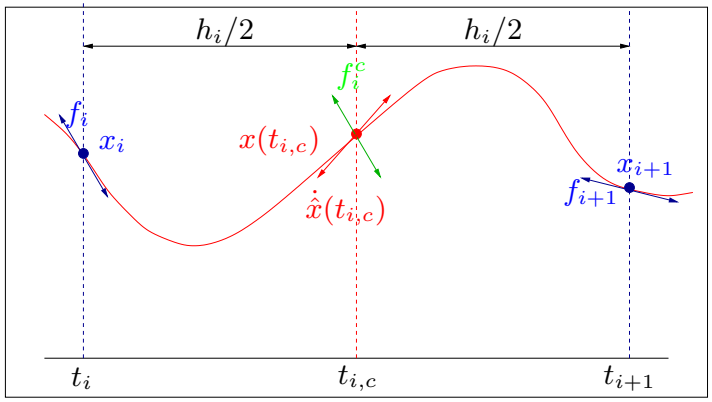
$$\text{sous} \begin{cases} R(Y) = 0 & \text{Résidus sur la dynamique} \\ g_d(Y) \leq 0 & \text{Contraintes discrétisées sur les trajectoires} \end{cases}$$

✔ Interpolation polynomiale cubique d'Hermite-Simpson

$$\hat{x}(t) = \sum_{k=0}^{3} c_k^i \frac{(t-t_i)^k}{h_i}, \ t_i \le t \le t_{i+1}, \ i = 1, \cdots, N-1, \ h_i = t_{i+1} - t_i$$

En notant $f_i = f(x_i, u_i), \ t_{c,i} = (t_i + t_{i+1})/2$

$$c_0^i = x_i, \quad (\hat{x}(t_i) = x_i)$$
$$c_1^i = h_i f_i, \quad (\dot{\hat{x}}(t_i) = f_i)$$
$$c_2^i = -3x_i - 2h_i f_i + 3x_{i+1} - h_i f_{i+1}, \quad (\hat{x}(t_{i+1}) = x_{i+1})$$
$$c_3^i = 2x_i + h_i f_i - 2x_{i+1} + h_i f_{i+1}, \quad (\dot{\hat{x}}(t_{i+1}) = f_{i+1})$$



✔ Estimation de l'état au centre de l'intervalle

$$\hat{x}(t_{c,i}) = \frac{1}{2}(x_i + x_{i+1}) - \frac{h_i}{8}(f_{i+1} - f_i) \quad \text{et} \quad \dot{\hat{x}}(t_{c,i}) = \frac{-3}{2h_i}(x_i - x_{i+1}) - \frac{1}{4}(f_{i+1} + f_i)$$

✔ Contraintes de collocation

$$\hat{R}_i = \dot{\hat{x}}(t_{i,c}) - f(\hat{x}(t_{i,c}), \hat{u}(t_{i,c})) = \dot{\hat{x}}(t_{i,c}) - f_i^c = 0, \quad i = 1, \ldots, N-1$$
$$R_i = x_{i+1} - x_i - \frac{h_i}{6}(f_i + 4f_i^c + f_{i+1}) = 0, \quad i = 1, \ldots, N-1$$

Quadrature et POC discrétisé

✔ Quadrature cohérente avec la méthode de discrétisation de la dynamique (Simpson)

$$
\begin{aligned}
J(x,u) &= \int_{t_0}^{t_f} L(t,x,u)dt + \psi_0(t_f, x(t_f)) = \psi_0(t_N, x_N) + \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} L(t,x,u)\mathrm{d}t \\
&\simeq \psi_0(t_N, x_N) + \sum_{i=0}^{N-1} \frac{h_i}{6} \left[L(t_i, x_i, u_i) + 4L(t_{i,c}, x_{i,c}, u_{i,c}) + L(t_{i+1}, x_{i+1}, u_{i+1}) \right] \\
&\simeq F(Y)
\end{aligned}
$$

✔ POC discrétisé

$$
\begin{aligned}
\min_{Y} \quad & \psi_0(t_N, x_N) + \sum_{i=0}^{N-1} \frac{h_i}{6} \left[L(t_i, x_i, u_i) + 4L(t_{i,c}, x_{i,c}, u_{i,c}) + L(t_{i+1}, x_{i+1}, u_{i+1}) \right] \\
& R_i(x_i, x_{i+1}, f(x_i, u_i), f(x_{i,c}, u_{i,c}), f(x_{i+1}, u_{i+1})) = 0, \ i = 1, \cdots, N-1 \\
\text{sous} \quad & \psi(t_N, x_N) = 0 \\
& x_1 = x_0 \\
& g(x_i, u_i, t_i) \leq 0, \ i = 1, \cdots, N
\end{aligned}
$$

Quadrature et POC discrétisé

✔ Méthode d'Euler

- Variables $Y^T = \begin{bmatrix} x_1 & u_1 & \cdots & x_N & u_N \end{bmatrix}$
- Résidus $R_i = x_{i+1} - x_i - h_i f_i$

✔ Méthode Trapézoidale

- Variables $Y^T = \begin{bmatrix} x_1 & u_1 & \cdots & x_N & u_N \end{bmatrix}$
- Résidus $R_i = x_{i+1} - x_i - \frac{h_i}{2}(f_i + f_{i+1})$

✔ Méthode de Runge-Kutta classique

- Variables $Y^T = \begin{bmatrix} x_1 & u_1 & u_{1,c} & \cdots & u_{N-1,c} & x_N & u_N \end{bmatrix}$
- Résidus $R_i = x_{i+1} - x_i - \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ où

$$\begin{aligned} k_1 &= h_i f_i \\ k_2 &= h_i f(x_i + \tfrac{1}{2}k_1, u_{i+1,c}, t_i + \tfrac{h_i}{2}) \\ k_3 &= h_i f(x_i + \tfrac{1}{2}k_2, u_{i+1,c}, t_i + \tfrac{h_i}{2}) \\ k_4 &= h_i f(x_i + k_3, u_{i+1}, t_{i+1}) \end{aligned}$$

Méthode de collocation : exemple (I)

❶ Discrétisation d'un horizon fini : $t_0 = t_1 \leq t_2 \leq \cdots \leq t_{N-1} \leq t_N = t_f$, $h_i = t_{i+1} - t_i$

❷ Variables d'optimisation

$$Y = \left[\begin{array}{ccccccccc} x_1^1 & \cdots & x_1^N & x_2^1 & \cdots & x_2^N & u^1 & \cdots & u^N & t_N \end{array} \right]^T \in \mathbb{R}^{3N+1}$$

❸ Contraintes de résidus : R_i, $i = 1, \cdots, N-1$: (Hermite-Simpson)

$$\begin{aligned} x_1^{i+1} - x_1^i - \frac{h_i}{6}\left[3(x_2^i + x_2^{i+1}) - \frac{h_i}{2}(u^{i+1} - u^i)\right] &= 0 \\ x_2^{i+1} - x_2^i - \frac{h_i}{2}\left[u^{i+1} + u^i\right] &= 0 \end{aligned}$$

❹ Contraintes initiales et finales : $x_1^1 = 0$, $x_1^N = 1$, $x_2^1 = 0$, $x_2^N = -2$

```
function [c,ceq] = cons(x)
N=(length(x)-1)/3;
c=0;tf=x(end);xf=0;yf=0;h=tf/N;ceq=[];
for i=1:N-1
 ceq=[ceq;x(i+1)-x(i)-(h/6)*(3*(x(N+i)+x(N+i+1))-(h/2)*(x(2*N+i+1)-x(2*N+i)))];
 ceq=[ceq;x(N+i+1)-x(N+i)-(h/2)*(x(2*N+i)+x(2*N+i+1))];
end
ceq=[ceq;x(N)-1;x(2*N)+2;x(1);x(N+1)];
```

Méthode de collocation : exemple (II)

❺ Critère : t_N

```
function [val] = tempsfinal(x)
val=x(end);
end
```

❻ Problème d'optimisation :

```
function [Y,x1,x2,u,tf]=Collocation_double_integrator(N)
clear all; close all; clc;
N=50;yinit=2*rand(3*N,1);tfinit=1;Yinit=[yinit;tfinit];
options = optimoptions('fmincon','Algorithm','sqp',...
'MaxFunEvals',100000,'MaxIter',1000);
lb=[-5*ones(2*N,1);-ones(N+1,1)];lb(3*N+1)=0;
ub=[5*ones(2*N,1);ones(N+1,1)];ub(3*N+1)=10;
[Y,Fval,exitflag]=fmincon(@tempsfinal,Yinit,[],[],[],[],lb,ub,...
@cons,options);
exitflag;tf=Y(end);
for i=1:N
    x1(i)=Y(i);x2(i)=Y(N+i);
    u(i)=Y(2*N+i);
end
```

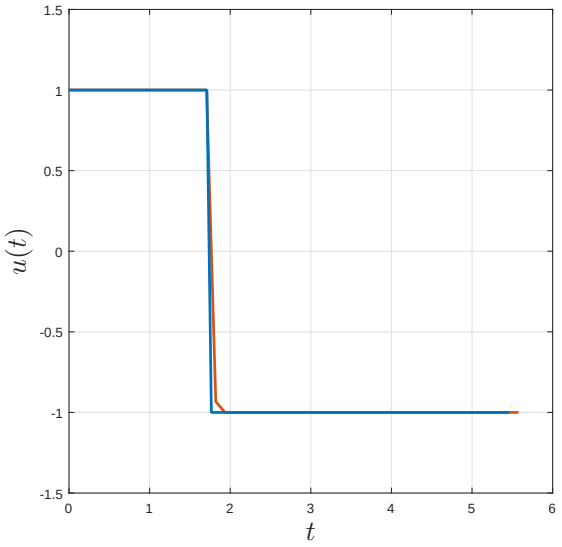
Méthode de tir simple indirecte : exemple (III)

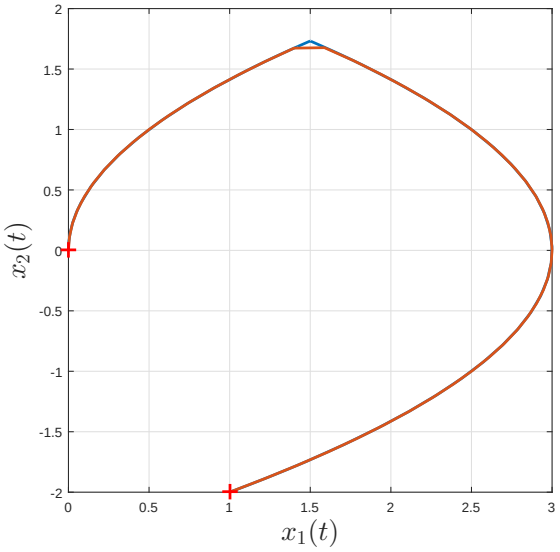
$$u^*(t) = \{+1, -1\}\,,\ \tau = \frac{\sqrt{2x_{2f}^2 + 4x_{1f}}}{2} = \sqrt{3}$$

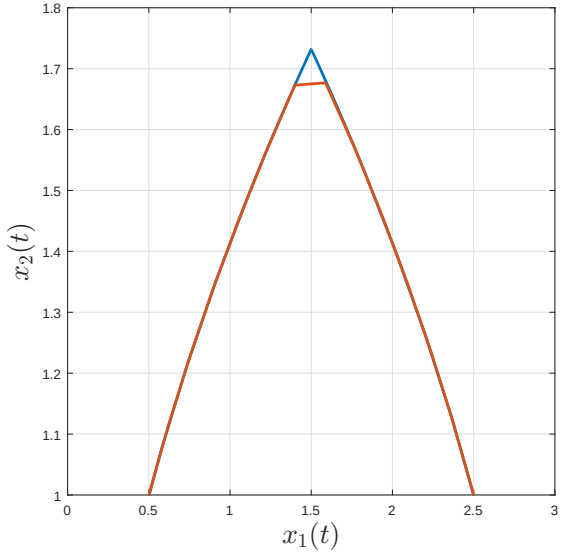
$$t_f = -x_{2f} + \sqrt{2x_{2f}^2 + 4x_{1f}} = 2 + 2\sqrt{3}\ \text{s},$$

$$0 \le t \le \sqrt{3} \qquad x_1^*(t) = \frac{t^2}{2}, \quad x_2^*(t) = t$$

$$\sqrt{3} \le t \le t_f \qquad x_1^*(t) = -\frac{t^2}{2} + 2\sqrt{3}t - 3, \quad x_2^*(t) = -t + 2\sqrt{3}$$







Résultats collocation :

$N = 50,\ t_f = 5.5765$ s

Quelle méthode choisir ?

La méthode à choisir dépend du problème posé !

- ✔ Les méthodes indirectes :
  - ♡ Bonne précision numérique
  - ♠ CN d'optimalité
  - ♠ Rigidité de la méthode (structure des commuations)
  - ♠ Difficulté des contraintes sur l'état
  - ♠ Initialisation difficile
- ✔ Les méthodes directes :
  - ♡ Mise en œuvre simple
  - ♡ Bonne robustesse
  - ♡ Contraintes sur l'état
  - ♠ Peu précises
  - ♠ Pas de garantie entre les points de discrétisation
  - ♠ Possibilité de minima locaux pour le PNL
  - ♠ Taille du problème de PNL