

RÉSOLUTION NUMÉRIQUE D'UN PROBLÈME DE TRANSFERT INTERPLANÉTAIRE : MÉTHODE DE TIR INDIRECTE ET MÉTHODE DE COLLOCATION DIRECTE

I Maximisation du rayon final d'un transfert interplanétaire en temps fixé : partie théorique

Le problème à résoudre est un problème de transfert interplanétaire dans le système solaire, sous hypothèse de poussée faible, d'un véhicule spatial dont la modélisation reprend celle du transfert orbital vue en cours. Le véhicule considéré de masse $m(t)$ est mû par une poussée de norme constante T dont l'orientation $\phi(t)$ peut varier. Il doit atteindre en un temps fixé (t_f) l'orbite circulaire de rayon maximal ($r(t_f)$) à partir d'une orbite circulaire donnée $r(0)$ correspondant à l'orbite de la terre et avec une vitesse initiale donnée v_0 (cf. Figure 1).

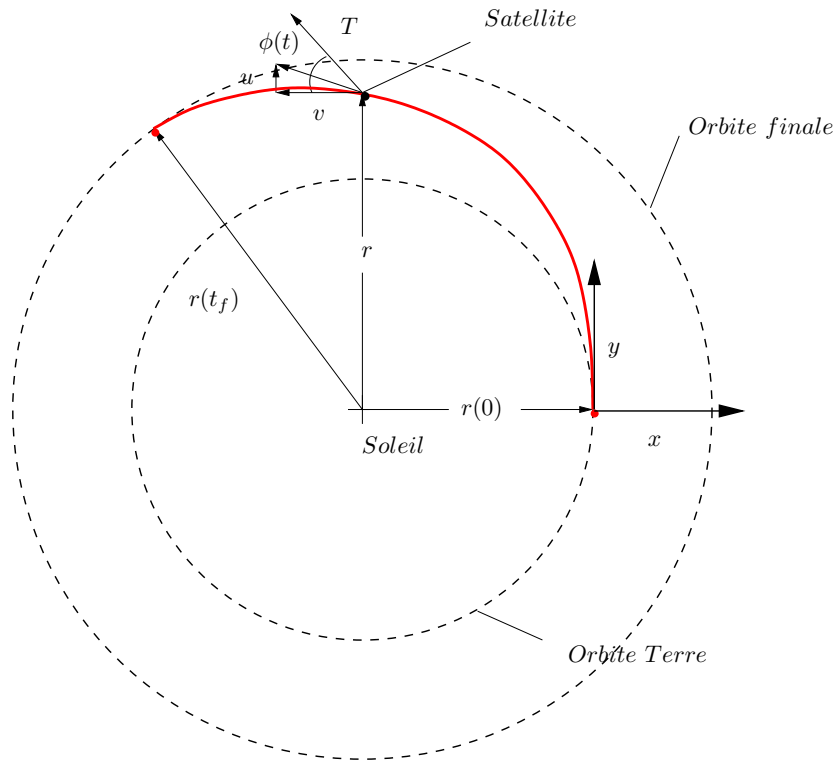


FIGURE 1: Transfert Interplanétaire

Le modèle de la dynamique du véhicule est donné par les équations suivantes :

$$\begin{aligned}
 \dot{r}(t) &= u(t) \\
 \dot{u}(t) &= \frac{v^2(t)}{r} - \frac{\mu}{r^2} + \frac{T \sin \phi}{m_0 - |\dot{m}(t)|t} \\
 \dot{v}(t) &= -\frac{uv}{r} + \frac{T \cos \phi}{m_0 - |\dot{m}(t)|t}
 \end{aligned} \tag{1}$$

où :

- r est l'amplitude du rayon vecteur du centre du soleil au véhicule ;
- u et v sont respectivement les composantes radiale et tangentielle de la vitesse du véhicule exprimées dans le repère local (O, x, y) avec le point O attaché au véhicule ;
- $m_0 = 4535.92$ kg. est la masse initiale du véhicule ;
- $|\dot{m}| = 5.8419$ kg./jour est la consommation de carburant ;
- ϕ est l'angle de la poussée ;
- $T = 3.7792$ N. est l'amplitude la poussée ;
- $\mu = 1.32712440018(9) \times 10^{20} \text{ m}^3 \cdot \text{s}^{-2}$ est le paramètre gravitationnel standard du soleil,

Les conditions aux limites sont données par :

$$r(0) = r_0 = 149597870.691 \times 10^3 \text{ m}, \quad u(0) = u_0 = 0 \text{ m/s}, \quad v(0) = v_0 = \sqrt{\frac{\mu}{r_0}} \text{ m/s},$$

$$u(t_f) = u_f = 0 \text{ m/s}, \quad v(t_f) = v_f = \sqrt{\frac{\mu}{r(t_f)}} \text{ m/s}, \quad r(t_f) = r_f \text{ m}, \quad t_f = 193 \text{ jours.}$$

I.1 Modélisation du problème de commande optimale

1. Ecrire le problème de commande optimale ainsi posé.

Solution:

$$\begin{array}{ll} \min_{\phi(t)} & -r(t_f) \\ \text{sous} & \dot{r}(t) = u(t) \end{array}$$

$$\dot{u}(t) = \frac{v^2(t)}{r(t)} - \frac{\mu}{r^2(t)} + \frac{T \sin \phi(t)}{m_0 - |\dot{m}|t}$$

$$\dot{v}(t) = -\frac{u(t)v(t)}{r(t)} + \frac{T \cos \phi(t)}{m_0 - |\dot{m}|t} \quad (2)$$

$$r(0) = r_0, \quad u(0) = u_0 = 0, \quad v(0) = v_0 = \sqrt{\frac{\mu}{r_0}},$$

$$\psi_1(t_f) = u(t_f) = 0, \quad \psi_2(t_f) = v(t_f) - \sqrt{\frac{\mu}{r(t_f)}} = 0, \quad t_f = \text{fixé}$$

Afin de normaliser le problème précédent et éviter des problèmes numériques lors de la résolution par une méthode de tir indirecte, les changements de variables suivants sont effectués :

$$\begin{array}{lll} \bar{m} = \frac{m}{m_0}, & \bar{r} = \frac{r}{r_0}, & \tau = \frac{v_0}{r_0} t \\ \bar{u} = \frac{u}{v_0}, & \bar{v} = \frac{v}{v_0}, & \bar{T} = \frac{\mu m_0}{r_0^2} T. \end{array} \quad (3)$$

2. Réécrire le problème de commande optimale précédent et les conditions aux limites avec les nouvelles variables et où le temps t a été remplacé par la nouvelle variable indépendante τ .

Solution: On calcule :

$$dt = \frac{r_0}{v_0} d\tau, \quad \frac{du}{dt} = \frac{\mu}{r_0^2} \frac{d\bar{u}}{d\tau}, \quad \frac{dv}{dt} = \frac{\mu}{r_0^2} \frac{d\bar{v}}{d\tau}, \quad \frac{dm}{dt} = m_0 \sqrt{\frac{\mu}{r_0^3}} \frac{d\bar{m}}{d\tau}$$

afin d'obtenir :

$$\min_{\phi(\tau)} -\bar{r}(\tau_f)$$

$$\text{sous } \dot{\bar{r}}(\tau) = \bar{u}(\tau)$$

$$\dot{\bar{u}}(\tau) = \frac{\bar{v}^2(\tau)}{\bar{r}(\tau)} - \frac{1}{\bar{r}^2(\tau)} + \frac{\bar{T} \sin \phi(\tau)}{1 - |\dot{\bar{m}}|\tau} \quad (4)$$

$$\dot{\bar{v}}(\tau) = -\frac{\bar{u}(\tau)\bar{v}(\tau)}{\bar{r}(\tau)} + \frac{\bar{T} \cos \phi(\tau)}{1 - |\dot{\bar{m}}|\tau}$$

$$\bar{r}(0) = 1, \bar{u}(0) = \bar{u}_0 = 0, \bar{v}(0) = \bar{v}_0 = 1,$$

$$\bar{\psi}_1(\tau_f) = \bar{u}(\tau_f) = 0, \bar{\psi}_2(\tau_f) = \bar{v}(\tau_f) - \sqrt{\frac{1}{\bar{r}(\tau_f)}} = 0, \tau_f = 3.32$$

où $\bar{T} = 0.1405$ et $|\dot{\bar{m}}| = 0.07487$.

I.2 Obtention du problème aux deux bouts

1. Ecrire le Hamiltonien du problème de commande optimale transformé.

Solution: Le Hamiltonien est donné par :

$$H(\bar{r}, \bar{u}, \bar{v}, \phi, \lambda) = \lambda_1(\tau)\bar{u} + \lambda_2(\tau) \left(\frac{\bar{v}^2(\tau)}{\bar{r}(\tau)} - \frac{1}{\bar{r}^2(\tau)} + \frac{\bar{T} \sin \phi(\tau)}{1 - |\dot{\bar{m}}|\tau} \right) + \lambda_3(\tau) \left(-\frac{\bar{u}(\tau)\bar{v}(\tau)}{\bar{r}(\tau)} + \frac{\bar{T} \cos \phi(\tau)}{1 - |\dot{\bar{m}}|\tau} \right) \quad (5)$$

où $\lambda^T = [\lambda_1 \quad \lambda_2 \quad \lambda_3]^T$ est le vecteur d'état adjoint.

2. Ecrire les conditions nécessaires d'optimalité au premier ordre.

Solution: Les équations nécessaires d'optimalité sont données par le système d'équations canoniques de Hamilton et par la condition d'optimalité.

$$\dot{\bar{r}}(\tau) = \bar{u}(\tau)$$

$$\dot{\bar{u}}(\tau) = \frac{\bar{v}^2(\tau)}{\bar{r}(\tau)} - \frac{1}{\bar{r}^2(\tau)} + \frac{\bar{T} \sin \phi(\tau)}{1 - |\dot{\bar{m}}|\tau}$$

$$\dot{\bar{v}}(\tau) = -\frac{\bar{u}(\tau)\bar{v}(\tau)}{\bar{r}(\tau)} + \frac{\bar{T} \cos \phi(\tau)}{1 - |\dot{\bar{m}}|\tau}$$

$$\dot{\lambda}_1 = -H_{\bar{r}}(\bar{r}, \bar{u}, \bar{v}, \phi, \lambda) = -\lambda_2 \left(-\frac{\bar{v}^2}{\bar{r}^2} + \frac{2}{\bar{r}^3} \right) - \lambda_3 \frac{\bar{u}\bar{v}}{\bar{r}^2} \quad (6)$$

$$\dot{\lambda}_2 = -H_{\bar{u}}(\bar{r}, \bar{u}, \bar{v}, \phi, \lambda) = -\lambda_1 + \lambda_3 \frac{\bar{v}}{\bar{r}}$$

$$\dot{\lambda}_3 = -H_{\bar{v}}(\bar{r}, \bar{u}, \bar{v}, \phi, \lambda) = -2\lambda_2 \frac{\bar{v}}{\bar{r}} + \lambda_3 \frac{\bar{u}}{\bar{r}}$$

$$\frac{\partial H}{\partial \phi}(\bar{r}, \bar{u}, \bar{v}, \phi, \lambda) = \frac{\bar{T}}{1 - |\dot{\bar{m}}|\tau} (\lambda_2 \cos \phi - \lambda_3 \sin \phi) = 0.$$

Cette dernière relation signifie que les vecteurs $\begin{bmatrix} \lambda_2 \\ -\lambda_3 \end{bmatrix}$ et $\begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix}$ sont orthogonaux donc :

$$\cos \phi = \frac{\pm \lambda_3}{\sqrt{\lambda_2^2 + \lambda_3^2}}, \quad \sin \phi = \frac{\pm \lambda_2}{\sqrt{\lambda_2^2 + \lambda_3^2}}.$$

Il reste une indétermination sur le calcul de la commande optimale.

3. Appliquer la condition nécessaire d'optimalité au second ordre de Legendre-Clebsch et en déduire la commande optimale comme une fonction des états adjoints.

Solution: La condition nécessaire d'optimalité au second ordre de Legendre-Clebsch est donnée par $\frac{\partial^2 H}{\partial u^2} \geq 0$. On obtient ainsi :

$$\frac{\partial^2 H}{\partial u^2} = \frac{\bar{T}}{1 - |\bar{m}| \tau} (-\lambda_2 \sin \phi - \lambda_3 \cos \phi).$$

Cela signifie que nécessairement la commande optimale doit vérifier :

$$\cos \phi = \frac{-\lambda_3}{\sqrt{\lambda_2^2 + \lambda_3^2}}, \quad \sin \phi = \frac{-\lambda_2}{\sqrt{\lambda_2^2 + \lambda_3^2}}.$$

afin de vérifier cette condition d'optimalité.

4. Ecrire les conditions de transversalité.

Solution: On définit la fonction de contrainte sur les conditions finales comme :

$$\tilde{\psi}(\tau_f) = -\bar{r}(\tau_f) + \nu_1 \bar{u}(\tau_f) + \nu_2 \left[\bar{v}(\tau_f) - \sqrt{\frac{1}{\bar{r}(\tau_f)}} \right].$$

On obtient ainsi les conditions de transversalité :

$$\begin{aligned} \lambda_1(\tau_f) &= -1 + \frac{\nu_2}{2\bar{r}^{3/2}(\tau_f)} \\ \lambda_2(\tau_f) &= \nu_1 \\ \lambda_3(\tau_f) &= \nu_2 \end{aligned} \tag{7}$$

5. Rappeler les conditions aux limites.

Solution:

$$\begin{aligned} \bar{r}(0) &= 1 \\ \bar{u}(0) &= 0 \\ \bar{v}(0) &= 1 \\ \bar{u}(\tau_f) &= 0 \\ \bar{v}(\tau_f) - \sqrt{\frac{1}{\bar{r}(\tau_f)}} &= 0 \end{aligned} \tag{8}$$

6. En déduire le problème aux deux bouts (TPBVP).

Solution: Le problème aux deux bouts s'écrit donc comme :

$$\begin{aligned}
 \dot{\bar{r}}(\tau) &= \bar{u}(\tau), \quad \bar{r}(0) = 1, \quad \bar{r}(\tau_f) \\
 \dot{\bar{u}}(\tau) &= \frac{\bar{v}^2(\tau)}{\bar{r}(\tau)} - \frac{1}{\bar{r}^2(\tau)} - \frac{\bar{T}}{1 - |\dot{m}| \tau} \frac{\lambda_2}{\sqrt{\lambda_2^2 + \lambda_3^2}}, \quad \bar{u}(0) = 0, \quad \bar{u}(\tau_f) = 0 \\
 \dot{\bar{v}}(\tau) &= -\frac{\bar{u}(\tau)\bar{v}(\tau)}{\bar{r}(\tau)} - \frac{\bar{T}}{1 - |\dot{m}| \tau} \frac{\lambda_3}{\sqrt{\lambda_2^2 + \lambda_3^2}}, \quad \bar{v}(0) = 1, \quad \bar{v}(\tau_f) = \sqrt{\frac{1}{\bar{r}(\tau_f)}} \\
 \dot{\lambda}_1 - \lambda_2 \left(-\frac{\bar{v}^2}{\bar{r}^2} + \frac{2}{\bar{r}^3} \right) - \lambda_3 \frac{\bar{u}\bar{v}}{\bar{r}^2}, \quad \lambda_1(\tau_f) &= -1 + \frac{\lambda_3(\tau_f)}{2\bar{r}^{3/2}(\tau_f)} \\
 \dot{\lambda}_2 &= -\lambda_1 + \lambda_3 \frac{\bar{v}}{\bar{r}}, \quad \lambda_2(\tau_f) \\
 \dot{\lambda}_3 &= -2\lambda_2 \frac{\bar{v}}{\bar{r}} + \lambda_3 \frac{\bar{u}}{\bar{r}}, \quad \lambda_3(\tau_f)
 \end{aligned} \tag{9}$$

II Maximisation du rayon final d'un transfert interplanétaire en temps fixé : résolution par une méthode de tir indirecte

Afin de résoudre le problème numériquement par une méthode de tir indirecte, on va utiliser la fonction `bvp4c` de MATLAB®. La syntaxe de celle-ci est donnée comme :

```
>> sol = bvp4c(odefun,bcfun,solinit,options)
```

où

- `odefun` est un pointeur (*function handle*) vers la fonction décrivant le système d'équations différentielles à intégrer ;
- `bcfun` est un pointeur (*function handle*) vers la fonction décrivant le système d'équations aux limites du problème aux deux bouts ;
- `solinit` est une structure contenant l'initialisation de la solution ;
- `options` est une structure contenant les options du solveur et définie par la fonction `bvpset`.

On se reportera à la page d'aide en ligne de MATLAB® afin d'avoir plus de précisions sur cette fonction.

1. Ecrire la fonction décrivant les équations différentielles du TPBVP.

Solution:

```
function [dx] = Eq_Orbite(t,x)
global mu m0 mdot T

%Etat
r = x(1);u = x(2);v = x(3);
lamr = x(4);lamu = x(5);lamv = x(6);

%Substitution de la commande optimale
sinphi = -lamu./sqrt(lamu.^2+lamv.^2);
cosphi = -lamv./sqrt(lamu.^2+lamv.^2);

%Equations Dynamiques
dr = u;
```

```

du = v^2/r -mu/r^2 + T*sinphi/(m0 + mdot*t);
dv = -u*v/r + T*cosphi/(m0 + mdot*t);

dlamr = -lamu*(-v^2/r^2 + 2*mu/r^3) -lamv*(u*v/r^2);
dlamu = -lamr + lamv*v/r;
dlamv = -lamu*2*v/r + lamv*u/r;

dx = [dr; du; dv; dlamr; dlamu; dlamv];
end
    
```

2. Ecrire la fonction décrivant le système d'équations aux limites du problème aux deux bouts.

```

Solution:
function [res] = Conditions_Limites_Orbites(x,xf)

global mu m0 mdot T

%Etat Initial
r = x(1);u = x(2);v = x(3);
lamr1 = x(4);lamu1 = x(5);lamv1 = x(6);

%Etat Final
rf = xf(1);uf = xf(2);vf = xf(3);
lamrf = xf(4);lamuf = xf(5);lamvf = xf(6);

% Contraintes aux limites
b1 = r -1;
b2 = u;
b3 = v -sqrt(mu/r);
b4 = uf;
b5 = vf -sqrt(mu/rf);
b6 = lamrf + 1 -lamvf*sqrt(mu)/2/rf^(3/2);

%Residu
res = [b1;b2;b3;b4;b5;b6];
end
    
```

3. Ecrire le script MATLAB® permettant la résolution du problème aux deux bouts. Pour l'initialisation, on choisira une grille de 100 points, une tolérance relative et absolue de 10^{-10} et un nombre maximum de points pour la fonction bvp4c de 2000 points.

```

Solution:
%Temps final fixé
Tf = 3.32;
%Constantes
global mu m0 mdot T
mu=1; m0=1; mdot=-0.07485; T= 0.1405;
%Initialisation
n=100;
y = [ones(1,n); %r
zeros(1,n); %u
ones(1,n); %v
-ones(1,n); %lambda_1
    
```

```

-ones(1,n); %lambda_2
-ones(1,n)]; %lambda_3
x = linspace(0,Tf,n); %vecteur temps
solinit.x = x;solinit.y = y;
%Options de l'optimiseur de bvp4c
tol = 1E-10;
options = bvpset('RelTol',tol,'AbsTol',[tol tol tol tol tol tol],'Nmax', 2000);
%Résolution
%if four
sol = bvp4c(@Eq_Orbite,@Conditions_Limites_Orbites,solinit,options)
    
```

4. A l'aide la solution obtenue à la question précédente, donner la valeur du coût optimal en km.

Solution: La valeur du rayon à l'instant final est récupérée de la structure des solutions `sol`. Afin de calculer sa valeur en km, il faut revenir à la variable $r(t_f) = r_0 * \bar{r}_f = 1.5252 * 149597870.691 = 228170462.4485$ km.

```

>> rfbar=sol.y(1,end)

ans =

    1.5252

>> 149597870.691*ans

ans =

 2.281704624485204e+08
    
```

5. A l'aide la solution obtenue, tracer la commande optimale en degrés en fonction du temps mesuré en jours.

Solution:

```

r0_t=149597870.691e3;
v0_t=sqrt(1.32712441933e20/r0_t);
ang2=atan2(sol.y([5],:),sol.y([6],:))+pi;
plot(sol.x*(r0_t/v0_t)/(24*3600),180/pi*ang2,'LineWidth',2)
grid on;
axis([0 200 0 360])
    
```

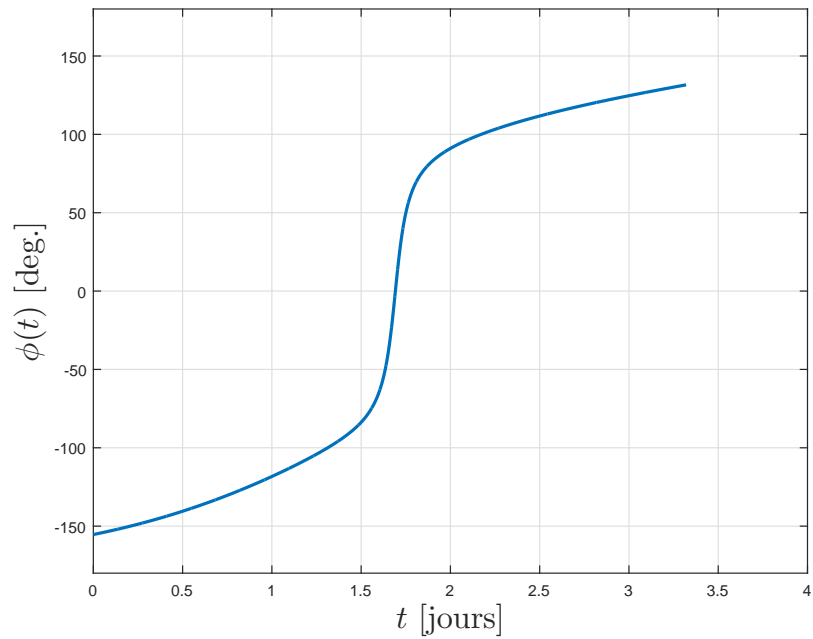


FIGURE 2: Commande optimale

6. A l'aide la solution obtenue, tracer les trajectoires d'état après mise à l'échelle (variables surlignées) sur un même graphique en fonction du temps exprimé en jours.

Solution:

```
plot(sol.x*(r0_t/v0_t)/(24*3600),sol.y(1:3,:), 'LineWidth', 2)
legend({'$\overline{r}(t)$', '$\overline{u}(t)$', '$\overline{v}(t)$'}, ...
'Location', 'NorthWest', 'Interpreter', 'latex')
grid on;
axis([0 200 0 2])
```

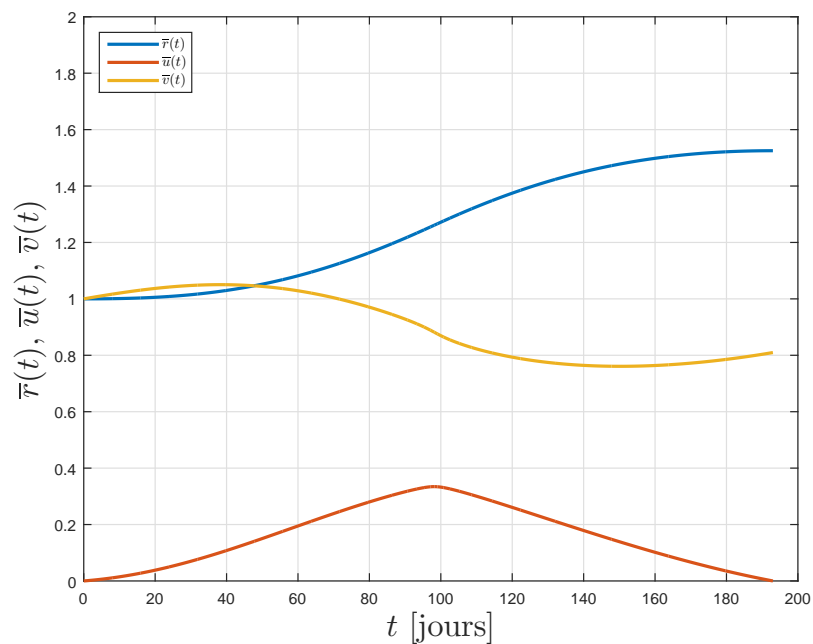


FIGURE 3: Commande optimale

7. A l'aide la solution obtenue, tracer les trajectoires des états adjoints sur un même graphique en fonction du temps exprimé en jours.

Solution:

```
plot(sol.x*(r0_t/v0_t)/(24*3600),sol.y(4:6,:), 'LineWidth',2)
legend({'$\lambda_1(t)$','$\lambda_2(t)$','$\lambda_3(t)$'},...
'Location','NorthWest','Interpreter','latex')
grid on;
axis([0 200 -3 2])
```

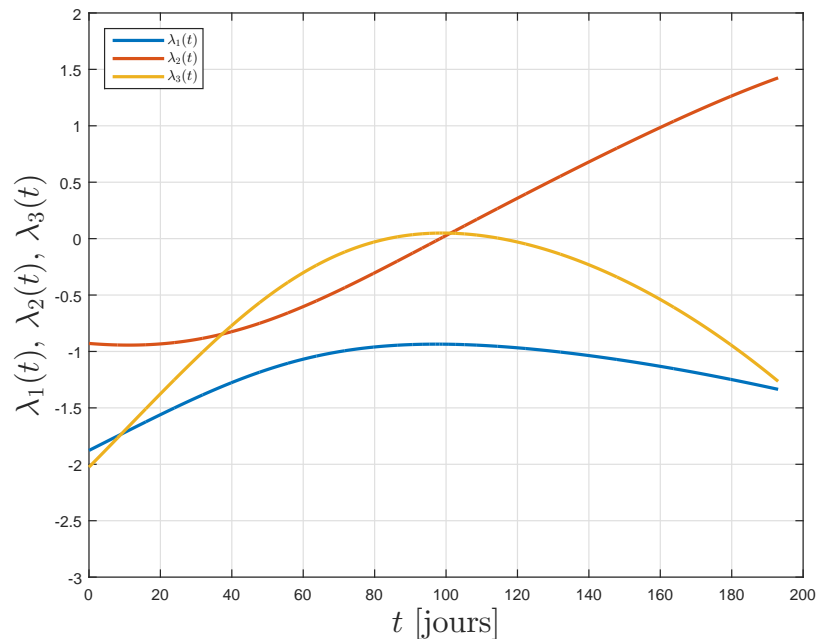


FIGURE 4: Commande optimale

8. A l'aide la solution obtenue, tracer la trajectoire du véhicule spatial dans le plan cartésien centré sur le soleil ainsi que les orbites de départ et d'arrivée.

Solution:

```
dt=diff(sol.x);
dth=(sol.y(3,1:end-1)./sol.y(1,1:end-1)).*dt; % \dot{\theta} = v_t/r
th=0+cumsum(dth');
traj=[sol.y(1,1:end-1)'.*cos(th) sol.y(1,1:end-1)'.*sin(th)];
plot(traj(:,1),traj(:,2),'k-', 'LineWidth',2)
hold on
zz=exp(sqrt(-1)*[0:.01:2*pi]');
r0=sol.y(1,1);rf=sol.y(1,end);
plot(r0*real(zz),r0*imag(zz),'r--', 'LineWidth',2)
plot(rf*real(zz),rf*imag(zz),'b--', 'LineWidth',2)
plot(r0,0,'ro', 'MarkerFace','r')
plot(rf*cos(th(end)),rf*sin(th(end)),'bo', 'MarkerFace','b')
```

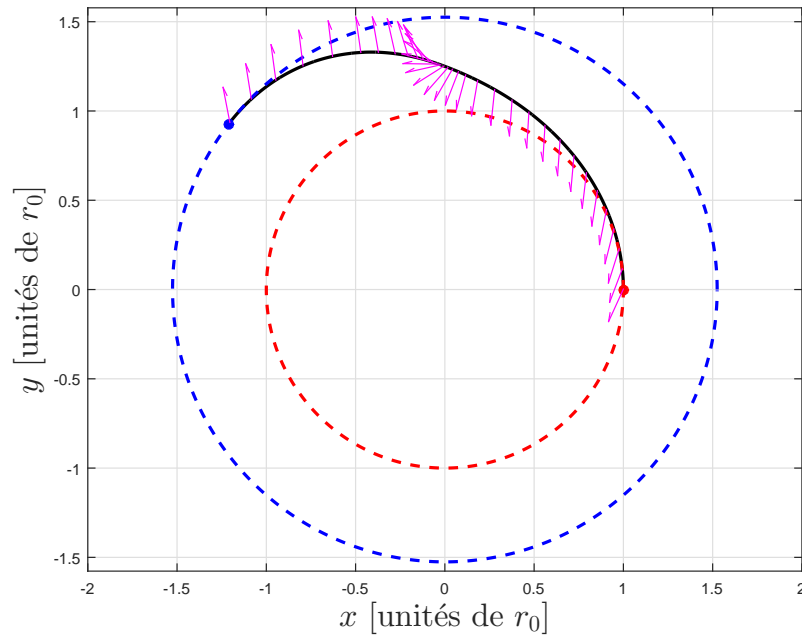


FIGURE 5: Commande optimale

III Maximisation du rayon final d'un transfert interplanétaire en temps fixé : résolution par une méthode de collocation directe

Afin de résoudre le problème numériquement par une méthode de collocation directe, la fonction `fmincon` de MATLAB© va être utilisée. La syntaxe de celle-ci est donnée comme :

```
>> sol = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

où

- `fun` est un pointeur (*function handle*) vers la fonction décrivant le critère à optimiser ;
- `x0` est le vecteur des variables d'initialisation des variables de décision pour le problème de PNL ;
- `A` est une matrice contenant les coefficients des contraintes inégalité linéaires ;
- `b` est un vecteur contenant les constantes des contraintes inégalité linéaires ;
- `Aeq` est une matrice contenant les coefficients des contraintes égalité linéaires ;
- `beq` est un vecteur contenant les constantes des contraintes égalité linéaires ;
- `lb` est un vecteur contenant les coefficients des bornes inférieures sur les variables de décision ;
- `ub` est un vecteur contenant les coefficients des bornes supérieures sur les variables de décision ;
- `nonlcon` est un pointeur (*function handle*) vers la fonction décrivant les contraintes non linéaires inégalité et égalité ;
- `options` est une structure contenant les options du solveur et définie par la fonction `bvpset`.

On se reportera à la page d'aide en ligne de MATLAB© afin d'avoir plus de précisions sur cette fonction. On choisit une grille uniforme sur l'horizon de temps $[0, \tau_f]$ avec $N = 70$ points.

1. Ecrire la fonction décrivant le critère du problème discrétisé.

Solution:

```
function [val] = critere_interplanetaire(x)
%
N=length(x)/5;
val=-x(N);
end
```

2. Ecrire la fonction décrivant les contraintes du problème discrétisé avec un schéma de collocation de type Euler.

Solution:

```
function [c,ceq] = cons_interplanetaire(x)
%
mu=1; m0=1; mdot=-0.07485; T= 0.1405;
N=length(x)/5;
c=0;
tf=3.32;xf=0;yf=0;h=tf/N;ceq=[];
t=linspace(0,tf,N);
ceq=[ceq;x(1)-1;x(N+1);x(2*N+1)-1];
for i=1:N-1
    ceq=[ceq;x(i+1)-x(i)-h*x(N+i)];
    ceq=[ceq;x(N+i+1)-x(N+i)-h*((x(2*N+i)^2/x(i))-...
(mu/x(i)^2)+((T*x(3*N+i))/(m0+mdot*t(i))))];
    ceq=[ceq;x(2*N+i+1)-x(2*N+i)-h*(-(x(N+i)*x(2*N+i)/x(i))-...
((T*x(4*N+i))/(m0+mdot*t(i))))];
    ceq=[ceq;x(3*N+i)^2+x(4*N+i)^2-1];
end
ceq=[ceq;x(2*N);x(3*N)-sqrt(1/x(N))];
end
```

3. Ecrire le script MATLAB® permettant la résolution du problème par la méthode de collocation. Pour l'initialisation, on choisira des variables aléatoires normalisées à l'aide de la fonction rand.

Solution:

```
clear all; clc;%close all; clc;
N=70;
tt=linspace(0,3.32,N);
tbar=sqrt(149597870691^3/1.32712441933e20)*tt;
t_jours=(1/(24*3600))*tbar;
rinit=0.5*rand(N,1)+1;
uinit=rand(N,1);
sin_init=rand(N,1);
cos_init=rand(N,1);
vinit=1.1*rand(N,1)+0.7;
phiinit=pi*rand(N,1);
yinit=[rinit;uinit;vinit;cos_init;sin_init];
options = optimoptions('fmincon','Algorithm','sqp','MaxFunEvals',100000,...
'MaxIter',10000);%,'tolfun',1e-11);
lb=[0.1*ones(N,1);-10*ones(N,1);-10*ones(N,1);-5*ones(N,1);-5*ones(N,1)];
ub=[10*ones(N,1);10*ones(N,1);10*ones(N,1);5*ones(N,1);5*ones(N,1)];
[rep,Fval,exitflag]=fmincon(@critere_interplanetaire,...
yinit,[],[],[],[],lb,ub,@cons_interplanetaire,options);
exitflag
```

4. A l'aide la solution obtenue à la question précédente, donner la valeur du coût optimal en km.

Solution: La valeur du rayon à l'instant final est récupérée comme la variable à la position N dans rep. Afin de calculer sa valeur en km, il faut revenir à la variable $r(t_f) = r_0 * \bar{r}_f = 1.5252 * 149597870.691 = 228170462.4485$ km.

```
>> rf=rep(N)

ans =

    1.5101

>> 149597870.691*ans

ans =

    2.281704624485204e+08
```

5. A l'aide la solution obtenue, tracer la commande optimale en degrés en fonction du temps mesuré en jours.

Solution:

```
for i=1:N
    u1(i)=rep(3*N+i);
    u2(i)=rep(4*N+i);
end
ang2=atan2(u1,u2);
tau=linspace(0,3.32,N);
tbar=sqrt(149597870691^3/1.32712441933e20)*tau;
t_jours=(1/(24*3600))*tbar;
plot(t_jours,180*ang2/pi);
axis square;
```

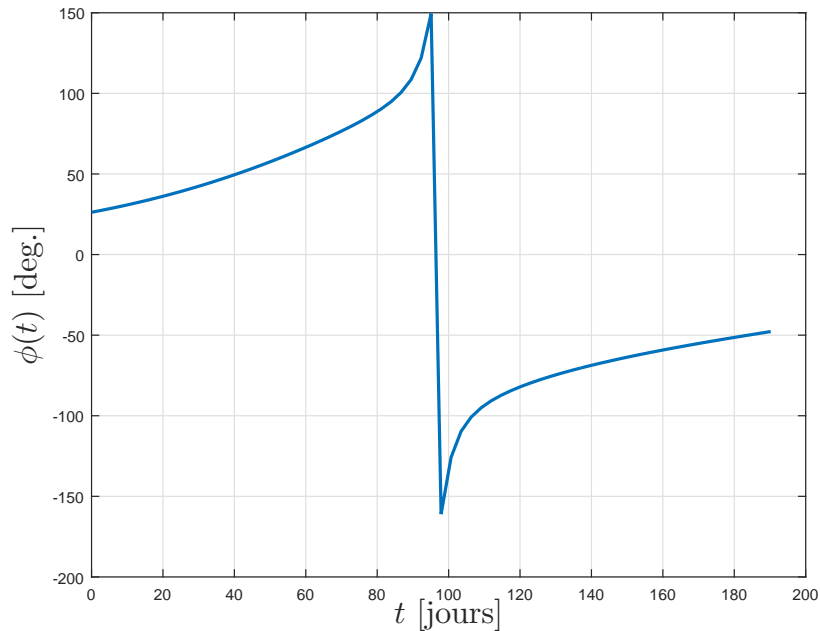


FIGURE 6: Commande optimale

6. A l'aide la solution obtenue, tracer les trajectoires d'état après mise à l'échelle (variables surlignées) sur un même graphique en fonction du temps exprimé en jours.

Solution:

```

for i=1:N
    x1(i)=rep(i);
    x2(i)=rep(N+i);
    x3(i)=rep(2*N+i);
end
figure(1);plot(t_jours,x1);
hold on
plot(t_jours,x2);
plot(t_jours,x3);axis square;

```

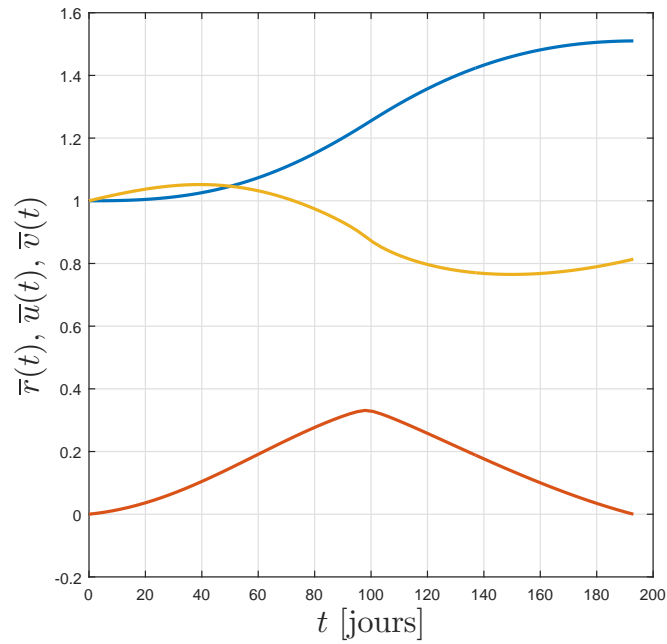


FIGURE 7: Commande optimale