Courbes de poursuite

C(t=0)

L(t=0)

L(t)

I

C(t)

r

r

L : Lièvre. Trajectoire en

C : Chien. Trajectoire en

r : Ligne de vue : segment [CL].

I : Point de collision.



..

Il s’agit d’étudier plusieurs stratégies de poursuites mettant en lice le poursuivi ( ici le lièvre) et le poursuivant ( ici le chien ).

Conditions d'étude :

- Les mouvements sont de classe .Les mobiles évoluent avec vitesses de normes et constantes. On pose: la vitesse du Chien est supérieure à celle du Lièvre .

-Étude dans le plan ou dans un espace de dimension 3.

I. Poursuite dite de l'interception et poursuite pure

a) L'interception

-Le Chien connaît la trajectoire *L* du lièvre dans sa globalité, et donc à chaque instant t la position .

-On cherche le point I= de *L* tel que :

ce qui correspond à une interception en ligne droite

-On montre l’existence de I, mais il peut exister plusieurs positions possibles pour ce point, on choisira alors celui de paramètre tI minimal.

L(tI)

I

L(0))

C(0)

-A chaque instant t on connaîtet(α et β réels) or

tI: Plus petite solution réelle, positive de (E).

**droite := proc (x, y, xp, yp, b)
local A;
solve(((x0(t)-xp)^2+(y0(t)-yp)^2)^(1/2) = 2\*t, t); A := %; print(T = A);**

**plot({[[x(A), y(A)], [xp, yp]], [x, y, 0 .. A]}) end ;**



tI=0.5

tI=1.54

Sur ces deux exemples le fuyard en vert a une trajectoire rectiligne, puis circulaire. Nous avons indiqué le temps d’interception pour chaque cas.

Il s’agit bien entendu de la poursuite la plus efficace donnant le temps d'impact le plus court.Mais cette solution est la moins réaliste car elle nécessite des informations inconnues dans la réalité.

b) Poursuite pure : Définition

-Poursuite usuelle: à chaque instant t le poursuivant vise le poursuivi : est colinéaire de même sens à .

I

L=Lièvre.

C=Chien.

r=Ligne de vue: segment [CL]

I=Point de collision.

r

L(t)

C(t)

C(t=0)



r

L(t=0)



c) Poursuite pure : Programmation

On a

On projette :

On résout à l’aide de Maple.

En entrée :

**>courbe:=proc(x0,y0,cix,ciy,i,j)**

**local eq1,eq2,L, g,h ;**

**eq1:= diff(x(t),t) = 2\*( diff(x0(t),t)^2 + diff(y0(t),t)^2 )^(1/2)\*(x0(t)-x(t))/((x0(t)-x(t))^2+(y0(t)-y(t))^2)^(1/2);**

**eq2:=diff(y(t),t) = 2\*( diff(x0(t),t)^2 + diff(y0(t),t)^2 )^(1/2)\*(y0(t)-y(t))/((x0(t)-x(t))^2+(y0(t)-y(t))^2)^(1/2);**

**L:=dsolve({eq1,eq2,cix,ciy},numeric) ;**

**g:=t->op(2,op(2,L(t))); h:=t->op(2,op(3,L(t)));**

**plot([[x0(t),y0(t),t=i..j],[g,h,i..j]],axes=FRAME);**

**end;**

En sortie : représentation des trajectoires de C et L.

En entrée :

**tempspoursuite:=proc(x0,y0,cix,ciy,i,j)**

**local eq1,eq2,L,g;**

**eq1:= diff(x(t),t) = 2\*( diff(x0(t),t)^2 + diff(y0(t),t)^2 )^(1/2)\*(x0(t)-x(t))/((x0(t)-x(t))^2+(y0(t)-y(t))^2)^(1/2);**

**eq2:=diff(y(t),t) = 2\*( diff(x0(t),t)^2 + diff(y0(t),t)^2 )^(1/2)\*(y0(t)-y(t))/((x0(t)-**

**x(t))^2+(y0(t)-y(t))^2)^(1/2);**

**L:=dsolve({eq1,eq2,cix,ciy},numeric;**

**g:=t->op(2,op(2,L(t)));**

**return(myfsolve(x0-g,i,j));**

**end;**

En sortie : le temps d’interception tI.

**>myfsolve:=proc(f,a,b)**

**x:=a; y:=b;**

**while abs(y-x)> 10^(-10)**

**do ifevalf(f((x+y)/2)\*f(x))>0**

**then x:=(x+y)/2**

**else y:=(x+y)/2**

**fi;**

**od;**

**return(evalf((x+y)/2));**

**end;**



tI= 0.61

tI= 1.33

-C’est la poursuite la plus simple : calcul peu complexe pour diriger la poursuite. Cependant le calcul du temps d’impact nécessite l’intégration d’une équation différentielle pouvant présenter des difficultés ( Maple intègre par dsolve/numeric)

-Cette poursuite est la moins efficace en terme de temps d'impact.

- pour la mise en œuvre le poursuivant n’a besoin que de la position L(t) à l’instant t pour orienter sa vitesse.

II. Poursuite parallèle

1. Définition

Etudions l’exemple d’une poursuite d’interception lorsque le lièvre suit une trajectoire rectiligne

I

C(0)

L(0)

L(t)

On a

D’où

Donc pour chaque instant t :

 parallèle à C(0)L(0)

C(t)

C’est la définition que nous donnerons pour une poursuite parallèle : la variation de direction dela ligne de vue , c’est à dire de la direction de la droite C(t)L(t),est nulleau cours du temps. En notant le vecteur , on aura en dimension 2 l’angle λ =qui est constant ce qui sera la cas étudié, en dimension 3 , il suffira d’écrire que , soit

Ce calcul est un peu délicat à mener et nous avons préféré traiter la dimension 2

Shéma en dimension 2

λ

C(t1)

C(t2)

L(t1)

L(t2)

r

r

λ

 : Axe de référence.

r : ligne de vue.

λ : Orientation de r.

Cette définition parait être meilleure que la poursuite pure puisqu’elle revient, pour le poursuivant, à viser à chaque instant non pas le fuyard, mais la position qu’il occuperait au moment de la collision en supposant qu’il suive une trajectoire rectiligne.Elle est simple à mettre en œuvre, mais nécessite à priori plus d’informations que la précédente pour sa mise en œuvre.

1. Equations du mouvementProgrammation

On se place en dimension 2, les calculs sont plus simples car on peut utiliser des arguments de trigonomètrie .

-A l’instant t on connait

On cherche C(xc(t), yc(t)) avec la condition .

α

λ

B

C

D

θ

A

On a et

Cherchons α

α

θ

A

B

D

Zoom:

C

Dans le triangle :

-ABD: (1)

-BCD: (2)

D’où avec (1) et (2) :

De plus

Donc

D’où

On résout à l’aide de Maple.
En entrée :

**>poursuiteparallele:=proc(x,y,k)**

**local x1,y1,Vc,Vm,alpha,lambda,X1,Y1,eq1,eq2,L,X,Y,i,j,g,h,s;**

**x1:=D(x); y1:=D(y);**

**Vc:=t->(x1(t)^2+y1(t)^2)^(1/2); Vm:=t->2\*Vc(t);**

**alpha:=t->arcsin(1/k\*sin(arctan(y1(t)/x1(t))-lambda));**

**lambda:=arctan((y(0))/(x(0)));**

**X1:=t->Vm(t)\*cos(alpha(t)+lambda);**

**Y1:=t->Vm(t)\*sin(alpha(t)+lambda);**

**eq1:=Diff(X(t),t)=X1(t);**

**eq2:=Diff(Y(t),t)=abs(Y1(t));**

**L:=dsolve({eq1,eq2,X(0)=0,Y(0)=0},numeric);**

**g:=t->op(2,op(2,L(t)));**

**h:=t->op(2,op(3,L(t)));**

**s:=myfsolve(x-g,0,100);**

**plot({[g,h,0..s],[x,y,0..s]});**

**end;**

En sortie : les trajectoires de C et L



tI=0.5

tI=1.15

Conclusion

|  |  |  |
| --- | --- | --- |
| Trajectoire | Rectiligne :  | Circulaire. |
| Poursuite | Courbe du chien. |  |
| Interception | 1.15 | 0.5 |
| Pure | 1.33 | 0.61 |
| Parallèle | 1.15 | 0.56 |