MCS 320 Project Two : problems of pursuit

The goal of this project is to use Maple to study problems of pursuit via solving ordinary differential equations. This document was generated on the basis of a Maple worksheet, available for download from the course web site. A good start on the project is to run that worksheet.

1. the path of a trailer

Imagine a child pulling the rigid bar of cart. Given the path of the child, what is the path of the cart? In general this is known as the tractrix problem. We denote time by the variable `t`. The path of the tractor in the plane is given by coordinate function `(x[1](t),x[2](t))`. The tractor is connected to the trailer by a rigid bar of of length `L`. Given `(x[1](t),x[2](t))` and `L`, the problem is to compute coordinate functions `(y[1](t),y[2](t))` of the path of the trailer. In the situation of the child pulling the cart, the child is the tractor, the cart is the trailer. If the tractor moves on a circle of radius one, expressions for the coordinates of the tractor are in the list `x`:

```
[> x := [cos(t),sin(t)];
```

Taking derivatives of the coordinate functions of the tractor, the velocity vector of the tractor have coordinate functions:

```
[> vx := map(e -> diff(e,t),x);
```

The trailer has unknown coordinate functions in `y` and the distance between tractor and trailer equals `L`. We start by computing the difference between the coordinates of the tractor and the coordinates of the trailer.

```
[> u := Vector([y[1](t) - x[1],y[2](t) - x[2]]);
```

To express that the length of the rigid bar between tractor and trailer equals `L`, we divide the direction vector by its length, and then multiply by `L`.

```
[> u := L*u/LinearAlgebra[Norm](u,2);
```

Via a symbolic simplification, we verify that the distance between trailer and tractor equals `L`:

```
[> simplify(LinearAlgebra[Norm](u,2),symbolic);
```

The trailer moves in the direction of the rigid bar. Equivalently, the velocity vector of the trailer is parallel to the direction of the bar, to the vector u. The velocity vector of the tractor is known, given in vx. We can compute the projection of the vector `vx` onto the velocity vector `u` of the trailer:

```
[> v := Vector(vx);
```

The formula `v*u` with `*` as the dot product is evaluated without the complex conjugation:

```
[> p := LinearAlgebra[DotProduct](v,u,conjugate=false);
```

Multiplying p with u gives the righthandside vector of the system of differential equations:

```
[> r := p*u;
```

The symbolic form of the tractrix problem is stating that the velocity vector of the trailer equals the vector `r`. The list of ordinary differential equations is in ode:

```
[> ode := seq(diff(y[k](t),t) = r[k],k=1..2);
```

We solve for the position of the trailer, in the list `var`:

```
[> var := [seq(y[k](t),k=1..2)];
```

As we now have the symbolic setup for the tractrix problem, a closed form symbolic solution (even for `L` = 1) does not exist. To solve the problem numerically, we setup an initial value problem. We take `L` = 1, and for `t` = 0, the position of the tractor at (1,0). The trailer at distance `L` from the initial position is then at (2,0).

```
[> ode1 := subs(L=1,[ode]);
[> ini := y[1](0) = 2, y[2](0) = 0;
```

The initial value problems is then defined as a list of ordinary differential equations, initial equations, and variables:

```
[> ivp := [op(ode1),ini]; var;
```

We call desolve with the numeric option:

```
[> s := dsolve(ivp,numeric,var);
```

On return is a function, for example we evaluate at t = 0 and t = 1:

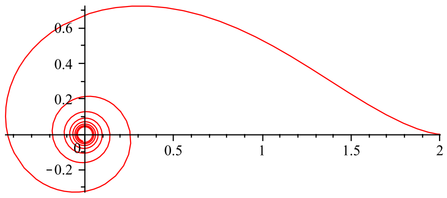
```
[> s(0); s(1);
```

To plot the trajectory of the trailer, we create separate coordinate functions:

```
[> fy1 := t -> rhs(s(t)[2]); fy2 := t -> rhs(s(t)[3]);
[> fy1(0); fy1(1);
```

We see that the path of the trailer defines a spiral:

```
[> plot([fy1,fy2,0..50],scaling=constrained);
```



Assignment One. Augment the picture of the trailer (the spiral above) with the path of the tractor drawn in green. On the same picture, draw the rigid bar in blue for regular time intervals, e.g. at t = 0, 5, 10, 15, etc. Make an animation of the rigid bar where the end points of the bar move on the paths of tractor and trailer.

Assignment Two. Extend the model to three dimensions. Imagine an airplane pulling a glider plane. To generalize the path of the tractor, use the helix: `(cos(t),sin(t),t)`. To visualize space curves, the `tubeplot` command is appropriate.

In addition to the extended initial value problem, make a plot of the trajectory of the trailer and an animation of the rigid bar between tractor and trailer.

2. modeling predator-prey pursuits

Instead of a tractor and a trailer remaining at a fixed distance from each other, we now consider the tractor as a prey fleeing from a predator. The path of the prey in the plane is given via two coordinate functions `(x[1](t),x[2](t))` and `(y[1](t),y[2](t))` are the coordinates of the predator. The predator moves in the direction of the prey. As in the first part of the project we compute the projection of the velocity vector onto the direction of pursuit.

```
[> x := 'x':
```

The direction of the predator towards the prey is compute as follows:

```
[> w := Vector([x[1](t) - y[1](t),x[2](t) - y[2](t)]);
[> w := w/LinearAlgebra[Norm](w,2);
```

If the speed of the predator is defined by the expression `s(t)`, then the righthandside vector of the system of ordinary differential equations is `r = s(t)*w`:

```
[> r := s(t)*w;
[> ode := seq(diff(y[k](t),t) = r[k],k=1..2);
```

Assignment Three. Consider a prey moving at constant speed as $(8t, 0)$. At $t = 0$, the predator starts at (20,20), and with constant speed $s(t) = 10$. Compute the trajectory of the predator. Find the right time interval till the predator meets the prey. Plot the path of the prey in red and that of the predator in green.

Make an animation of the predator chasing the prey, representing the prey as a red dot and predator as a green dot.

Assignment Four. The tuple $(10 + 20\cos(t), 20 + 15\sin(t))$ defines an elliptic path for the prey. For a predator as in the previous assignment, starting at (20,20) and with constant speed $s(t) = 10$, will the predator every catch the prey? Plot the trajectories and answer the question.

Assignment Five. For the same path of the prey as in the previous assignment and same initial position for the predator, approximate the value of the constant speed of the predator so the prey gets caught. Illustrate your answer with appropriate plots of the trajectories.

3. the deadline is Wednesday 6 April 2011, at 11AM

Bring to class the printout of a Maple worksheet that contains the setup of the initial value problems and the plots. To illustrate the animations, include a couple appropriate snapshots.

Structure your document along the answers to the assignments. Write appropriate comments.

This project must be solved individually, collaborations are not allowed.

Your worksheet should run like a program, via the menu `Edit` → `Execute` → `Worksheet`.

In addition to the printout, email your Maple worksheet (one single worksheet with all output removed) to `jan@math.uic.edu` so I can verify your calculations.

If you have questions, concerns, or technical difficulties, feel free to come to my office for help.