A slow afternoon chez PARKAS and a very fast fly (a fun talk)

Timothy Bourke[1,2] Marc Pouzet[2,1]

1. INRIA Paris-Rocquencourt

2. École normale supérieure (DI)

Synchron Workshop, Dagstuhl, 19 November 2013

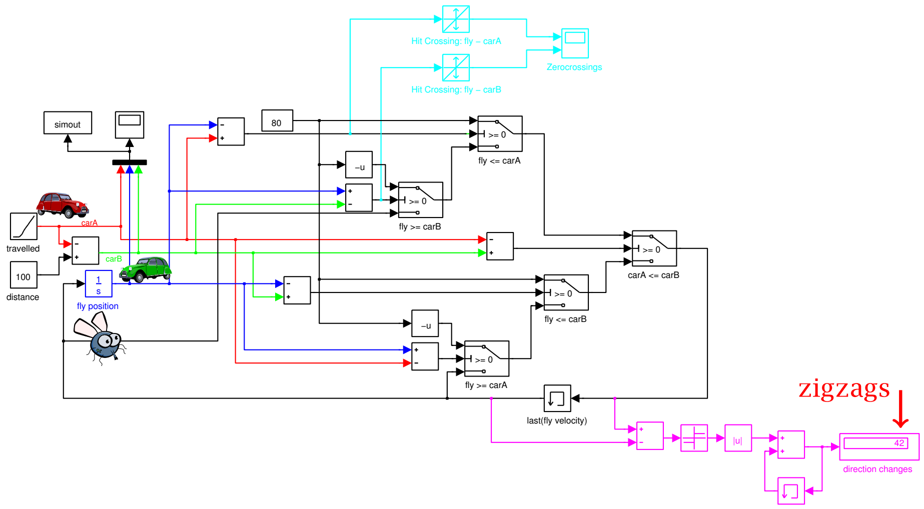
We briefly present a problem posed to use by Rafel Cases and Jordi Cortadella during a lunch organised by Gérard Berry. We propose solutions in the Simulink tool[1] and our language Zélus[2].

Imagine two cars. One starts at Barcelona and travels at 50 km h^{-1} toward Girona—a distance of 100 km. The other starts at Girona and travels at 50 km h^{-1} toward Barcelona. Between the two is a fly travelling at 80 km h^{-1}, initially from Barcelona toward Girona, and changing direction instantaneously whenever it meets either car. There are two questions.

1. How many zig-zags does the fly do during the two hours of travel?

2. Where will the fly be when the two cars reach their destinations?

We first modelled this problem in Simulink.



The number of zig-zags, to our great surprise and pleasure, was **42**! [1] (Using R2012a with the ODE45 solver and a relative tolerance of 1×10^{-3}.)

[1] `http://www.mathworks.com/products/simulink/`

[2] `http://zelus.di.ens.fr`

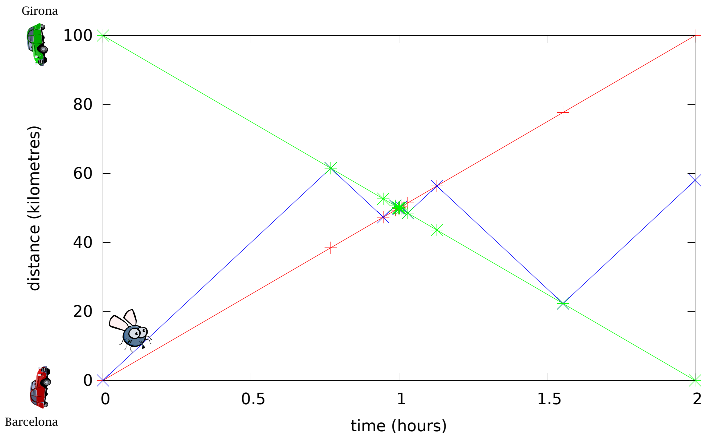
We then modelled the problem in Zélus.

```
let barcelona = 0.0
let girona = 100.0

let fly_velocity = 80.0
let car_velocity = 50.0

let hybrid model () =  (car1, car2, fly, zigzag, zeros) where
  rec der car1 = car_velocity      init barcelona
  and der car2 = −. car_velocity init girona
  and der fly = dir *. fly_velocity init barcelona
  and automaton
        | Above →
             do car_above = car2
             and car_below = car1
             until up(car1  −. car2) then Below
        | Below →
             do car_above = car1
             and car_below = car2
             done
        end
  and present
         up (car_below −. fly) | up(fly −. car_above) →
           do
              dir = −. (last dir)
              and zeros = last zeros + 1
              and emit zigzag = ()
           done
  and init dir = 1.0
  and init zeros = 0
```

This gave an answer of **48** and the graph shown below. (Using the Sundials CVODE solver and a custom implementation of the Illinois algorithm.)



Obviously neither answer is correct since the system is not well defined at the instant the cars pass each other. The important questions are whether we should, or even can, statically detect and reject such cases? or stop with an error at runtime?

References

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy.* Pan Books, 1979.