AKKOUCHE – BENEFICE – BREFORT – CARBONERA – DESROCHERS – ISSAUTIER – LARANJEIRA MOREIRA – LE DOZE – MONNET – OUBELHAJ

25/02/2014

Safe nonlinear control with IBEX

Goals

- Studying autonomous systems' safety with an interval method
- Software developed in C++, with the IBEX 2.0 library
- Five main tasks distributed between ten students

DISTRIBUTION

1. Scenarios definition
2. Simulation
3. Safety determination
4. Draw paving and vectors’ field
5. GUI designing and integration

1. Scenarios and function initialization

- Scenarios :
  - Simple :

$$\mathrm{f}\,(\mathrm{x},t) = \begin{pmatrix} -x_1 + t \\ -x_2 \\ -x_3 \end{pmatrix}$$

$$\mathrm{g}\,(\mathrm{x},t) = \begin{pmatrix} ((x_1 - t)^2 + (x_2)^2 - 1) \\ ((cos(x_3) - 1)^2 + (sin(x_3))^2 - 0.2) \end{pmatrix}$$

1. Scenarios and function initialization

- Non holonome with a linear state equation target :

$$\mathrm{f}(\mathrm{x},t) = \begin{pmatrix} \sqrt{(7t-x_1+7)^2+x_2^2}cos(x_3) \\ \sqrt{(7t-x_1+7)^2+x_2^2}sin(x_3) \\ 10cos(x_3)(\frac{(-x_2)}{\sqrt{((7t-x_1+7)^2+(-x_2)^2)}}) - sin(x_3)(\frac{(7t-x_1+7)}{\sqrt{((7t-x_1+7)^2+(-x_2)^2)}}) \end{pmatrix}$$

$$\mathrm{g}(\mathrm{x},t) = \begin{pmatrix} ((x_1-t)^2+(x_2)^2-1) \\ (cos(x_3)-(\frac{(7t-x_1+7)}{\sqrt{((7t-x_1+7)^2+(-x_2)^2)}}))^2+(sin(x_3)-(\frac{(-x_2)}{\sqrt{((7t-x_1+7)^2+(-x_2)^2)}}))^2-0.01 \end{pmatrix}$$

1. Scenarios and function initialization

- Non holonome :

$$f(x,t) = \begin{pmatrix} \sqrt{(7t - x_1 + 7)^2 + \left(\sin\frac{t}{10} - x_2 + \frac{1}{10}\cos\frac{t}{10}\right)^2} \cdot \cos x_3 \\ \sqrt{(7t - x_1 + 7)^2 + \left(\sin\frac{t}{10} - x_2 + \frac{1}{10}\cos\frac{t}{10}\right)^2} \cdot \sin x_3 \\ \frac{(\cos x_3)\left(\sin\frac{t}{10} - x_2 + \frac{1}{10}\cos\frac{t}{10}\right) - (\sin x_3)(7t - x_1 + 7)}{\sqrt{(7t-x_1+7)^2 + \left(\sin\frac{t}{10} - x_2 + \frac{1}{10}\cos\frac{t}{10}\right)^2}} \end{pmatrix}$$

$$g(x,t) = \begin{pmatrix} (x_1 - 7t)^2 + (x_2 - \sin\frac{t}{10})^2 - 1 \\ \left(\cos x_3 - \frac{7t - x_1 + 7}{\sqrt{(7t-x_1+7)^2 + \left(\sin\frac{t}{10} - x_2 + \frac{1}{10}\cos\frac{t}{10}\right)^2}}\right)^2 \cdots \\ \cdots + \left(\sin x_3 - \frac{\sin\frac{t}{10} - x_2 + \frac{1}{10}\cos\frac{t}{10}}{\sqrt{(7t-x_1+7)^2 + \left(\sin\frac{t}{10} - x_2 + \frac{1}{10}\cos\frac{t}{10}\right)^2}}\right)^2 - 0.2 \end{pmatrix}$$

1. Scenarios and function initialization

- Function initialization
  Save : create the .txt files from the text editing boxes
  Load : fill the text editing box if there already are a f.txt and g.txt files in the folder.
  Init : store the default function and return a Qstring that contains them in a bibex format.

2. Simulation

- Draw the path and the position of a Robot through time
- To be Safe, the robot must be inconsistent to the following equation:

$$\begin{cases} \text{(i)} & \frac{\partial g_i}{\partial \mathbf{x}}(\mathbf{x}, t)\,.\mathbf{f}(\mathbf{x}, t) + \frac{\partial g_i}{\partial t}(\mathbf{x}, t) \geq 0 \\ \text{(ii)} & g_i(\mathbf{x}, t) = 0 \\ \text{(iii)} & \mathbf{g}(\mathbf{x}, t) \leq 0 \end{cases}$$

- A random initial position which satisfy the previous condition (`t=0`)

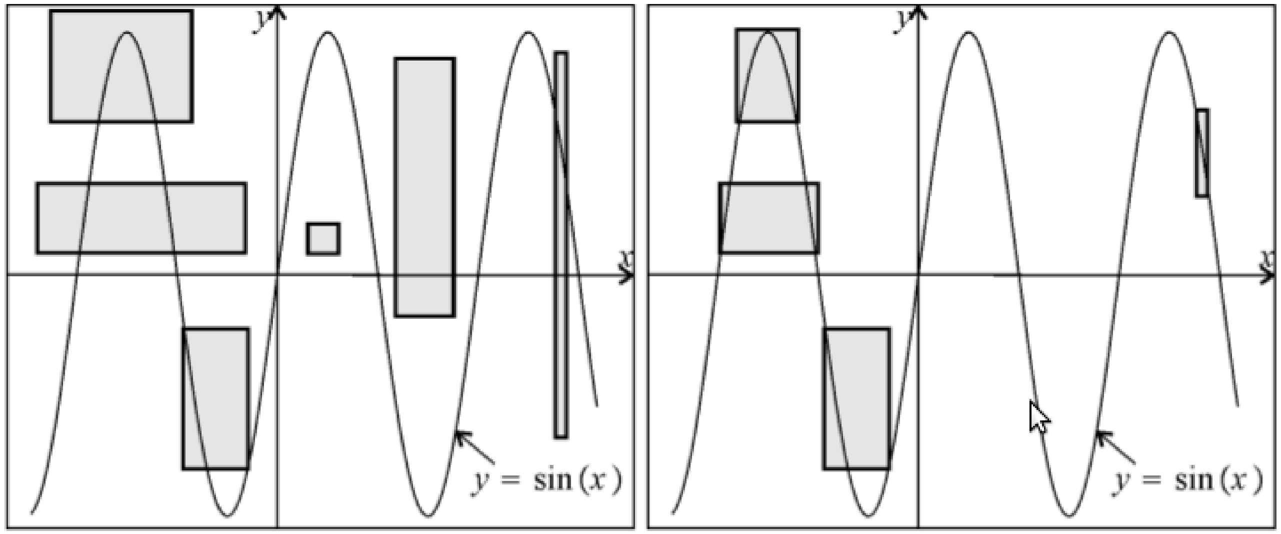
2. Implementation in BUBBIBEX

- A Monte-Carlo method computes a set of random initial states which satisfy the equation G
- From this initial state, BUBBIBEX computes the path of the robot from its regulated state equation
- Draw the robot state at a time retrieved from a Track Bar
- Debug :
  - In the bubble "perhaps" in which the robot is not insured of its safe position, addition of 10 robots to see if they converge/diverge from solution.

3. CheckCapture

- \- 4-Dimensions box : x, y, θ, t (time).
- \- Is the system safe ?
- \- System of constraints:

$$\begin{cases} \text{(i)} & \frac{\partial g_i}{\partial \mathbf{x}}(\mathbf{x},t)\,.\mathbf{f}(\mathbf{x},t)+\frac{\partial g_i}{\partial t}(\mathbf{x},t)\geq 0 \\ \text{(ii)} & g_i(\mathbf{x},t)=0 \\ \text{(iii)} & \mathbf{g}(\mathbf{x},t)\leq 0 \end{cases}$$



(Left) before contraction; (Right) after contraction

3. CheckCapture

- With 5 equations and 3 unknown variables, we made 2 intermediate contractors and a global one which is the two intermediate ones union

```
NumConstraint ci0(x1,x2,x3,t,fconst(x1,x2,x2,t)[0] >=0);  // Equation (i) of theorem pour g1
NumConstraint ci1(x1,x2,x3,t,fconst(x1,x2,x2,t)[1] >=0);  // Equation (i) of theorem pour g2
NumConstraint cii0(x1,x2,x3,t,g(x1,x2,x2,t)[0] =0);       // Equation (ii) of theorem pour g1
NumConstraint cii1(x1,x2,x3,t,g(x1,x2,x2,t)[1] =0);       // Equation (ii) of theorem pour g2
NumConstraint ciii(g, LEQ);                               // Equation (iii) of theorem

Array<NumConstraint> c0,c1;
c0.add(ci0);
c0.add(cii0);
c0.add(ciii);

c1.add(ci1);
c1.add(cii1);
c1.add(ciii);
CtcHC4 ctc0(c0);
CtcHC4 ctc1(c1);
CtcUnion un(ctc0,ctc1);
```

$$\begin{cases} \text{(i)} & \frac{\partial g_i}{\partial \mathbf{x}}(\mathbf{x},t)\,.\mathbf{f}(\mathbf{x},t) + \frac{\partial g_i}{\partial t}(\mathbf{x},t) \geq 0 \\ \text{(ii)} & g_i(\mathbf{x},t) = 0 \\ \text{(iii)} & \mathbf{g}(\mathbf{x},t) \leq 0 \end{cases}$$

4. Paving

- With a list of the « out » and « perhaps » boxes and the time set with trackbar we can start drawing theses boxes using color codes

- If the time selected by the trackbar is included in the time interval of the box (third interval of the intervalVector) we represent the box. If it is not we skip it.

4. Function field

- Using the list of « perhaps » boxes, we evaluate the function in the middle of each box that can be represented in the time selected by the trackbar.

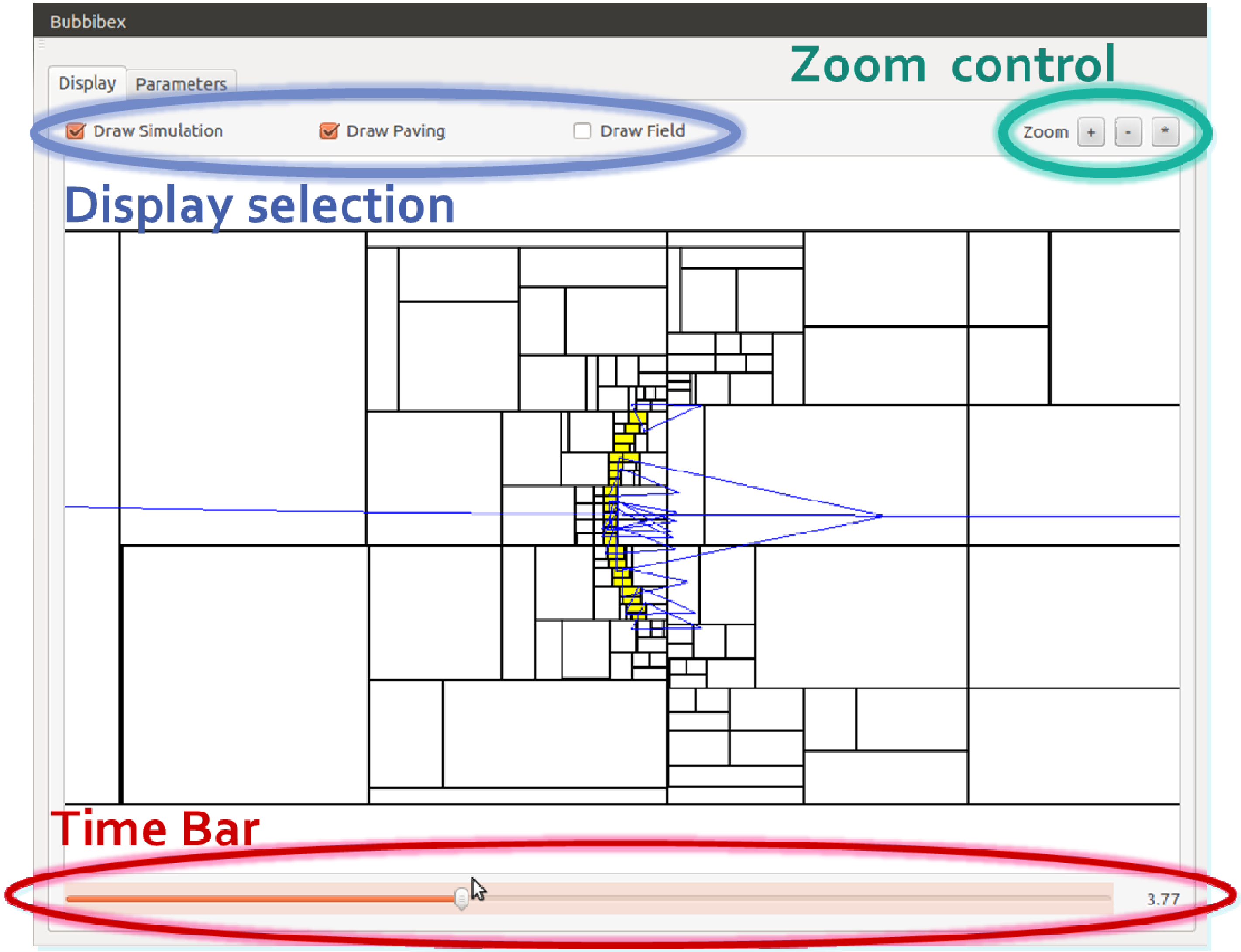
- We then represent this evaluation in the form of an arrow.

5. Integration

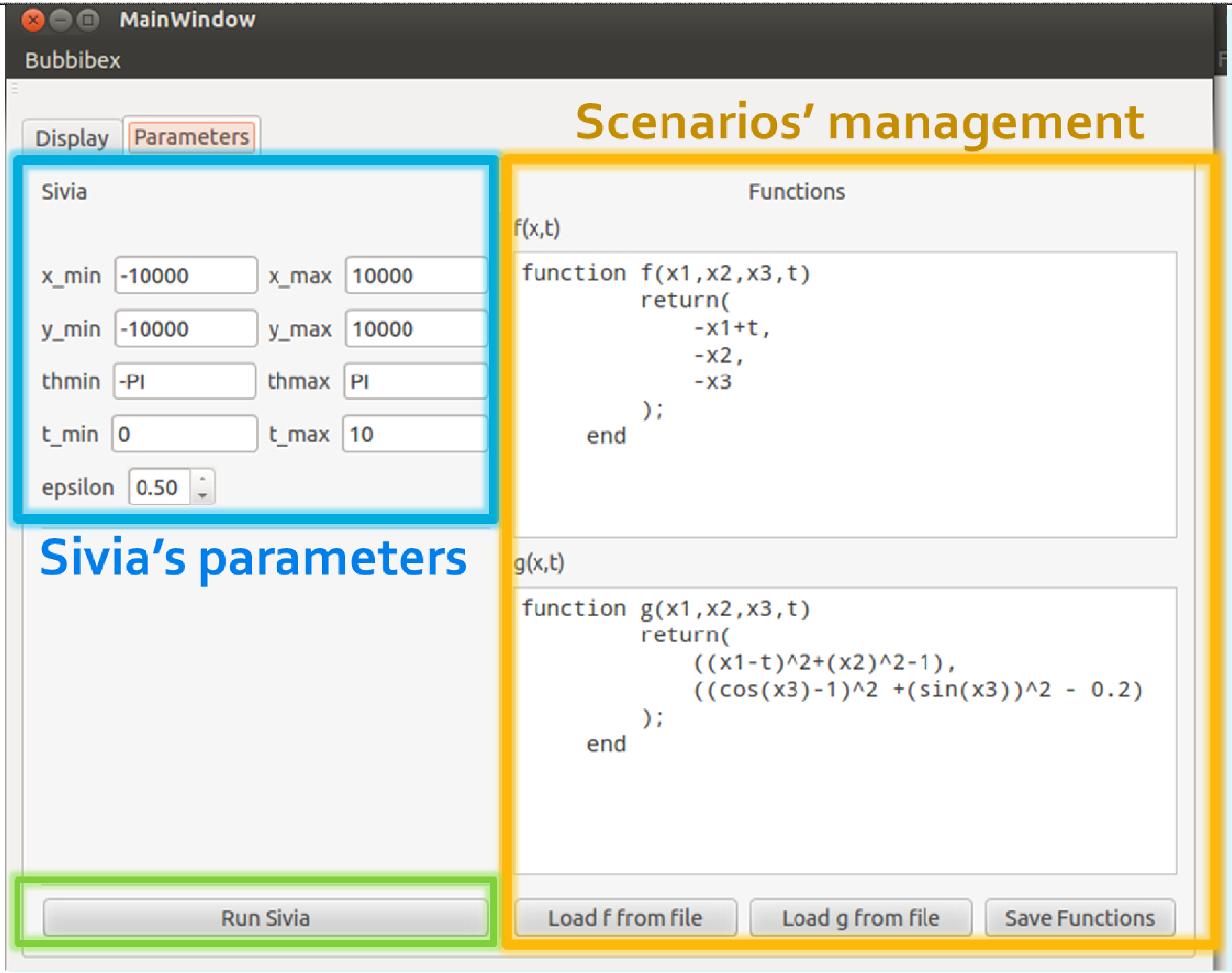
- Use of a git repository to manage the different tasks

- Coordination regarding inputs and outputs of functions

5. Interface - display



5. Interface - parameters



Results & conclusions

- See Demo