CS 7001 Mini-Project: Braitenberg Vehicles

Huaman, Ana C.

November 10, 2010

1 Introduction

This report explains the work done during the development of ***BVSim***, a simulator for Braitenberg Vehicles.

In his book "Vehicles: Experiments in Synthetic Psychology"[1], Valentino Braitenberg explores -in the form of thought experiments- how simple machines (vehicles) can present apparently complex behaviors that seem the result of some "intelligent" process, but that are no more than the combination of sensorial input applied to the actuators. This gives as a result behaviors that seems to reflect patterns that we can describe as "Optimism", "Dislike", "Cowardice", "Love" and many others. These behaviors serve to illustrate the concept of *"law of uphill analyisis and downhill invention"*, which says that it is more difficult to understand a process by observing it than if you would have created it (mainly because we tend to attribute more complexity to an unknown internal structure, when in fact it is usually simpler).

The Braitenberg vehicles are simple machines that have two wheels, each one controlled with a motor (actuator). These motors are connected directly to sensorial input, such as light and temperature intensity. These inputs control the output of the actuators in a positive or negative way (that is, exciting or "relaxing" them). Also, there are many ways to connect these signals to the motors, so there are virtually many types of possible behaviors to be implemented.

This reports begins explaining briefly the kinematic model used for the Vehicle (to compute the movement parameters of the Vehicle) and the sensors used to interact with the world. In the following section, I explain with more detail the simulated entities, the world in which the Vehicle operates and its attributes. Next, I present the results for six (06) demos for different behaviors of the vehicles. In the final section, I present a brief summary of what was learned with this project and some future projects related to this mini-project.

2 The Vehicle: Modelling

For this mini-project, the Braitenberg Vehicle (from now on called "B-Vehicle") was modelled as a ***differential drive robot***. As can be seen in the following figure, this robot has three degrees of freedom (two for position in the plane and one for orientation). To specify the position of the robot in global coordinates, we choose a reference point in the robot chassis. We call this point P and locate it in the middle of the imaginary line joining both wheels.

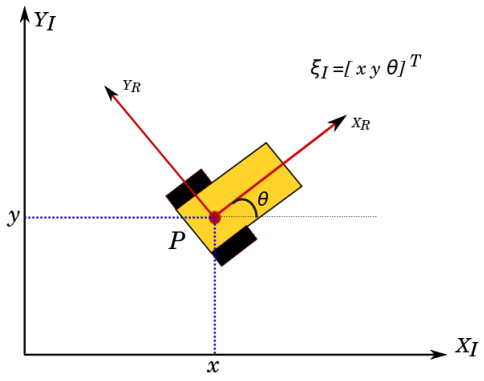


Figure 1: Our B-Vehicle initial model, the global reference frame and the robot's local reference frame

So, our robot is defined in the global reference frame with the column vector ξ_I:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

We express the position of point P in general coordinates with the following rotation matrix[4]:

$$R(\theta) = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Such as:

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I$$

2.1 Forward Kinematics Model

According to our robot representation:

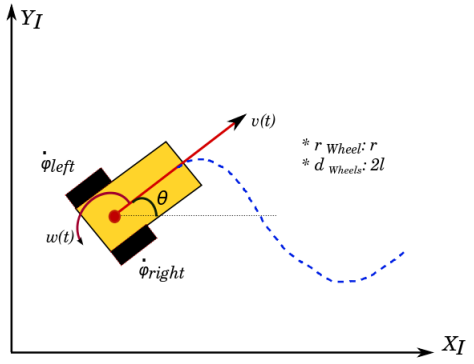


Figure 2: Our B-Vehicle model

Our kinematic model for the B-Vehicle differential-drive robot is the following:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_x(t) \\ v_y(t) \\ w(t) \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\varphi}_{right}}{2} + \frac{r\dot{\varphi}_{left}}{2} \\ 0 \\ \frac{r\dot{\varphi}_{right}}{2l} - \frac{r\dot{\varphi}_{left}}{2l} \end{bmatrix}$$

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} cos(\theta) * (\frac{r\dot{\varphi}_{right}}{2} + \frac{r\dot{\varphi}_{left}}{2}) \\ sin(\theta) * (\frac{r\dot{\varphi}_{right}}{2} + \frac{r\dot{\varphi}_{left}}{2}) \\ \frac{r\dot{\varphi}_{right}}{2l} - \frac{r\dot{\varphi}_{left}}{2l} \end{bmatrix}$$

Where each wheel has a radius r, the distance between the point P and each wheel is l and $\dot{\varphi}_{left}$ and $\dot{\varphi}_{right}$ are the velocity of the left and right wheel, respectively. As it is clear, $v_y(t)$ is zero in the robot's local reference system because the wheels don't allow a sideway motion.

2.2 The Vehicle: Sensors

Our B-Vehicle can potentially have ***03*** different kind of sensors:

1. Light sensors
2. Odor sensors *(theoretically)*
3. Temperature sensors

For simulation purposes, each B-Vehicle can have one couple of those sensors, each of them connected to one motor. How to connect these sensors and the type of stimulus they give to each motor is described in the following table:

Table 1: Sensor connections and resulting output

	Switch (T)		***Switch (NIL)***	
	Inhibitory	***Excitatory***	***Inhibitory***	***Excitatory***
Left Wheel	When *Right sensor* ↓, *Left speed* ↑	When *Right sensor* ↑, *Left speed* ↑	When *Left sensor* ↓, *Left speed* ↑	When *Left sensor* ↑, *Left speed* ↑
Right Wheel	When *Left sensor* ↓, *Right speed* ↑	When *Left sensor* ↑, *Right speed* ↑	When *Right sensor* ↓, *Right speed* ↑	When *Right sensor* ↑, *Right speed* ↑

The table mentioned represents the configurations shown in the following figure:

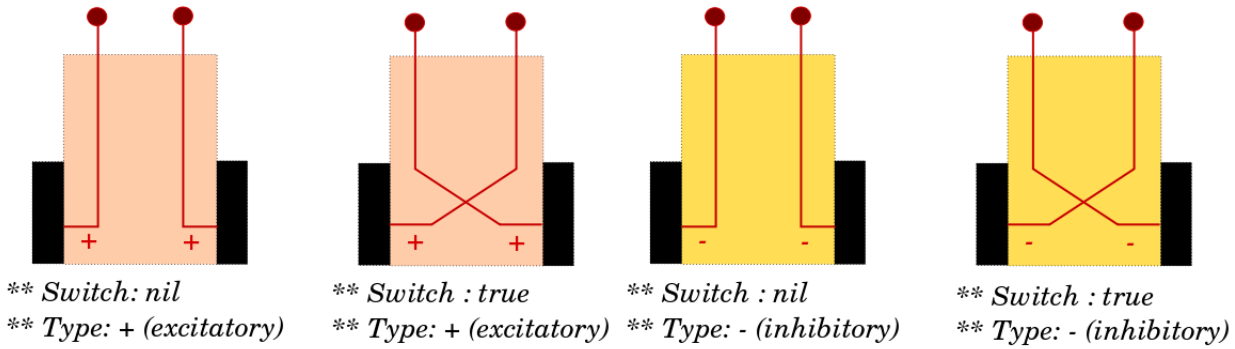


Figure 3: Connections and type of stimulus given by the sensors

Subsequently, there are ***03*** sources of sensorial data in our world, but we will describe that in the section describing the Braitenberg World.

3 Representation in Simulator

The *BVSim* simulator has the following characteristics:

- *BVSim* consists of a main GUI, as shown in Figure 4. It is entirely written in Lisp[3] with a binding to Tk (Ltk)[2] and have been tested in Ubuntu 9.10, using SBCL 1.0.29.11, LTK 0.91 and Slime
- As you may see, the GUI have two main components:
  1. *Canvas:* Where the simulation is going to be seen while running, it shows the vehicles as well as the sources simulated
  2. *User data and commands:* Here the user can enter the following information for the simulator:
     (a) *Load scene:* This button opens a dialog box that let you choose a .lisp file with a description of the world you want to load
     (b) *Start simulation:* Starts (or re-start, if stopped previously) the simulation
     (c) *Stop simulation:* Stops the simulation
     (d) *Select source:* This group of radio buttons let you add online whichever of the 03 types of sources. They appear with the text "default" on your screen. You add them by clicking in an empty space on the canvas surface (whether the simulation is running or not)
- The user can load an scene file (.lisp) specifying different models of B-Vehicles, their number, name, color, current sensors, among other useful characteristics. The scene file can also contain diverse sources of sensorial data (such as lights) which can be placed anywhere and in any number. (For a sample of a .scene file, please refer to the code appended at the end of this report.

In the following subsections, we will describe briefly the entities simulated:

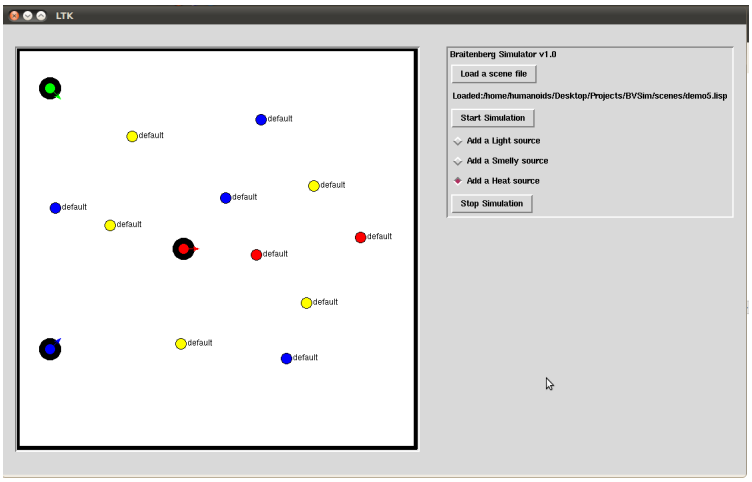


Figure 4: Main interface of *BVSim*

3.1 The B-Vehicle

The B-Vehicle in the simulator follows the same kinematic equations shown in the previous section. However, for simplicity in programming and GUI design, the actual representation of the B-Vehicle in the simulator *BVSim* is that of a circular black shape with its top of a color chosen by the user (not a rectangle with two wheels). In the same way, the wheels are not drawn in the simulation, but theoretically, they are located each on the extreme of the diameter that rests on local axis Y (see figure below)

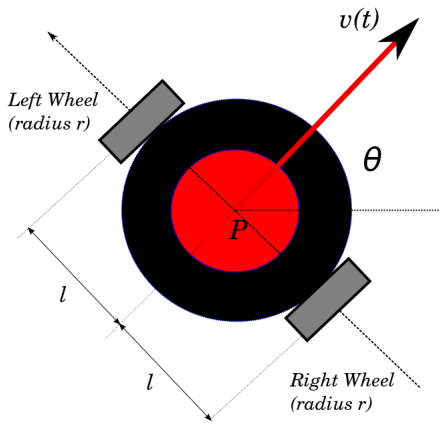


Figure 5: Real representation of B-Vehicle in (the wheels are not depicted in the Simulator, just the chassis) *BVSim*

The sensors, as described in the section before, should be located in the chassis, in front of the robot. However, due to our circular representation of the robot, we locate the sensors as if they were in the same position of the wheels,and they are not represented graphically.

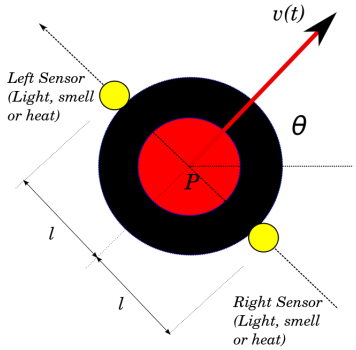


Figure 6: Location of sensors in robot chassis (they are not depicted in the simulator)

In conclusion, the B-Vehicle described is fully armored with its two wheels and its sensors, but in canvas, just the chassis and the linear velocity vector are shown.

3.2 The World or Scene

As it was mentioned, the world consists of 03 sources of sensorial data. How this sensorial input is perceived by the Vehicle will be explained in the following section (Behaviors).

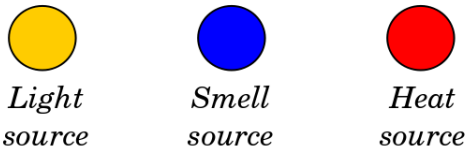


Figure 7: Sources of sensorial data in *BVSim* world

4 Behaviors

In this section, we will explain briefly the behaviors observed by combining different sensorial input and applying it to the actuators (motors) of the Vehicle. Moreover.

4.1 Demo 1: Cowardness

The first behavior to be simulated is obtained in the following way:

1. The vehicle consists of two motors and two sensors at each side of the chassis (left and right). For this demo, we use light sensors
2. The sensor output is proportional to the intensity of the signal (more intensity, greater sensor output)
3. The sensor output is inversely proportional to the distance of the sensor to the source
4. The output of each sensor is connected to its correspondent motor (left sensor with left motor and right sensor with right motor)
5. The equation for the sensor signal sent to the wheel (in form of a velocity command) is:

$$\delta_w = K_v * \frac{D_{ref}}{distance_{sensor-source}} * w_{nominal}$$

As can be deduced, if we are near the source of signal, the intensity is going to grow and the motor that is nearest to the source is going to go faster. Consequently, the B-vehicle is going to turn in the opposite way, avoiding the source (as it goes away, its velocity diminishes). This going-away of the Vehicle suggests the name of "Coward" for this behavior. As can be seen in the figure, it seems that the vehicle *does not like* the source very much, given that it keeps running away from it:

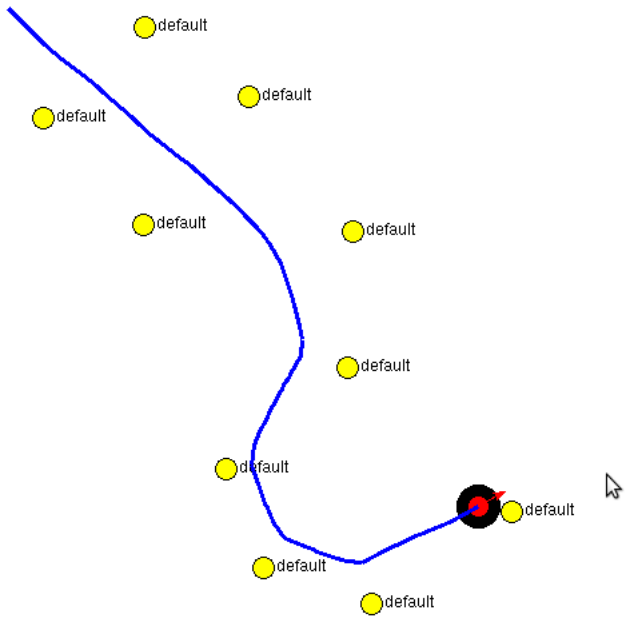


Figure 8: Path followed by our Coward Vehicle *BVSim* world

4.2 Demo 2: Aggressiveness

This behavior is simulated with the following characteristics:

1. The sensor output connected to the actuators acts in an "excitatory" mode. That is, the greater the sensor output, the greater the speed of the motor

2. The sensor output is connected switching sides: the right sensor is connected to the left motor and the left sensor is connected to the right motor

3. The equation for the sensor output is similar to the one shown for Coward behavior.
   Imagine that as the vehicle is moving, it detects one source of signal sideways. The nearest sensor is going to detect a bigger input, so the motor that is controlled by it (the one in the opposite side) will run faster. That would mean that the robot is going to turn *towards* the source and actually it is going to hit it, if there is not any other source that changes its path. Given this tendency of turning towards the source and speeding up along the way, we can call this behavior *Aggressive*

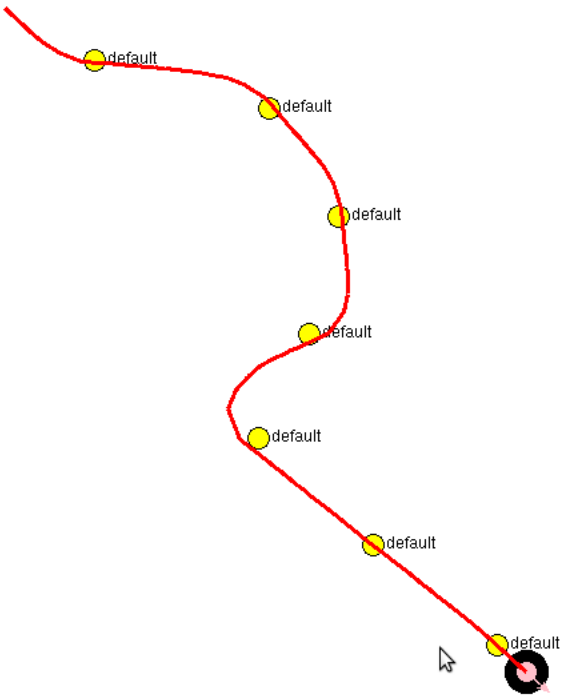


Figure 9: Path followed by our Aggressive Vehicle *BVSim* world

4.3 Demo 3: Love

This behavior is simulated with the following characteristics:

1. The sensor output connected to the actuators acts in an "inhibitory" mode. That is, the greater the sensor output, the smaller the speed of the motor

2. The sensor output is connected directly: the right sensor is connected to the right motor and the left sensor is connected to the left motor

3. The equation for the sensor output is similar to the one shown for Coward and Aggresive behavior, but now its effect is negative (to do so, we use a referential or nominal w and substract, that is:

$$\delta_w = w_{nominal} - K_v * \frac{D_{ref}}{distance_{sensor-source}} * w_{nominal}$$

In this behavior we can appreciate that when the vehicle is near the source, it slows down its movement. Additionally, as the motor nearer to the source is the slower of both actuators, the robot virates towards the source(sort of "looking" at the source). The robot seems to "stare" at the source indifinitely, which is why this behavior is denominated *"Love"*

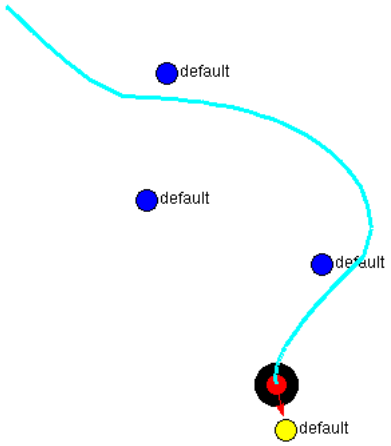
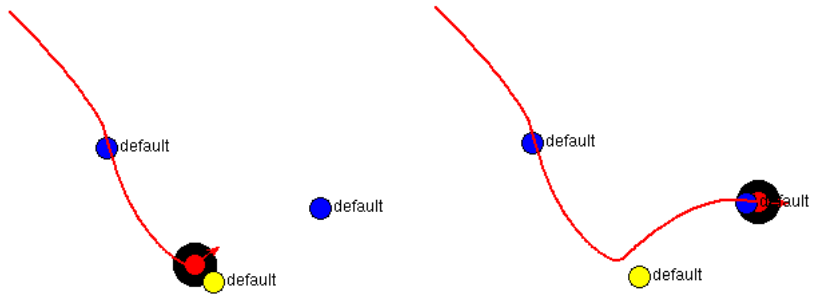


Figure 10: Path followed by our Loving Vehicle *BVSim* world

4.4 Demo 4: Explorer

In a similar fashion, this robot shares the same characteristics than the "Love" vehicle, with the difference that the sensor outputs are connected switched to the motors (that is, left sensor to right motor and right motor to left sensor). This gives as a result that the Vehicle slows down when near a source, but -contrarily to the last behavior, it does not keep "staring" at the source, but it turns towards the exterior (given that the nearest sensor will slow down the opposite motor more strongly). From looking the figure below, we can appreciate that the robot seems to like the source, but is ready to leave as soon as it perceives a new stimulus. That is the reason why this type of vehicle is christened as "Explorer". In the figure below we can see how initially it stays near the source, but as soon as it perceives a new source (Smell), it runs towards it.



(a) Explorer's path. First it stays near the source (light)

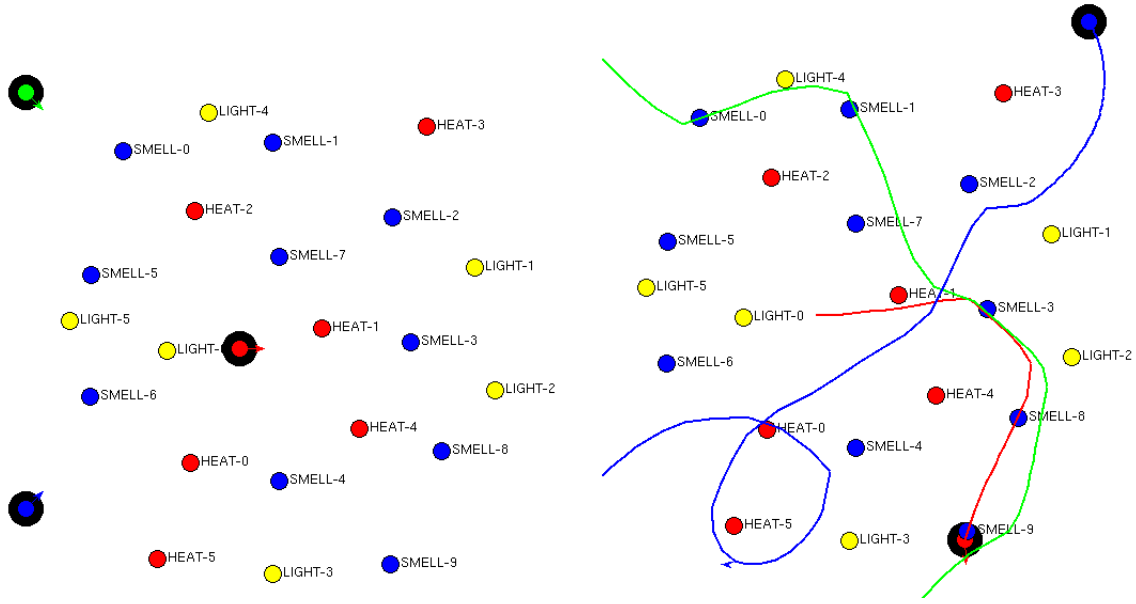
(b) At the first chance, it will run away towards another source (smell)

Figure 11: Explorer's path in *BVSim*

4.5 Demo 5: A nice mix

After experimenting with just one kind of sensor for each of the four behaviors described above, we can try a more complex behavior for more Vehicles. In this demo, we simulate three cars (A, B and C), which have the following characteristics: (The .lisp scene file for this demo is included in the appendix of this report)

1. *Car A (Green):* Coward to Heat, aggressive to Smell, Explorer towards Light
2. *Car B (Red):* Love Heat, aggressive to Smell and fears Light
3. *Car C (Blue):* Deadly aggresive to Heat, fears Smell, Explorer towards Light



(a) Initial scene

(b) Final scene

Figure 12: Sequence of Movements of vehicles with different sensor's connections

As we can see in Figure 12, the Vehicles exhibit-somehow- particular behaviors that may seem complex, but in fact are quite understandable. Based on the description given above, we can explain the paths followed by the Vehicles with this reasoning:

1. *Car A (Green):* We see in figure (b) that it first runs towards *SMELL-0*, which is natural since it is aggresive towards it. After that, we see that it spends some time near *LIGHT-4* (Explorer), but then, detecting *SMELL-1* it runs to attack it. Afterwards, it does the same going towards *SMELL-3, SMELL-8 & SMELL-9*. In its way, it finds a Heat source, which obligues it to run away from it (Coward), deviating it from its path and finding in its way to *SMELL-0* Coward to Heat, aggressive to Smell, Explorer towards Light. We can add that it also senses *LIGHT-2* and seems to try to go near it, but with the combined effects of different sources, it follows the path shown.
2. *Car B (Red):* Its path is shorter. First it aims towards *HEAT-1*, given that it seems to love it. However, the attractive effect of *SMELL-3* combined with its fear to *LIGHT-2* changes its path and make it go for *SMELL-8*
3. *Car C (Blue):* It goes right away to attack *HEAT-0* and while doing that, it also notices another heat source (*HEAT-5*), so it makes an arabesque and tries to hit it. On its way back, it notices other heat sources, however, the presence of *SMELL-3* makes it run away. In its way it finds to *SMELL-2* -which he fears of- and *LIGHT-1* - which it sort of likes but not permanently- so, it speed off searching for another sources.

As we can see, we can explain this -apparently- complex behaviors just by addressing the connections between the sensors and actuators and the kind of signal received (excitatory or inhibitory).

4.6 Demo 6: Instinctive

For our last demo, we make a slight change to the way our sensors process information received from the world. The sensor input is going to be processed according to the following graph:

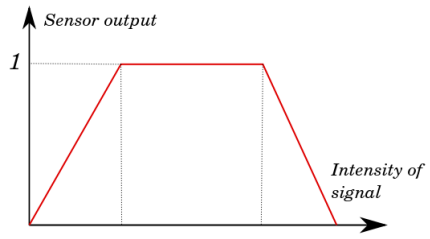
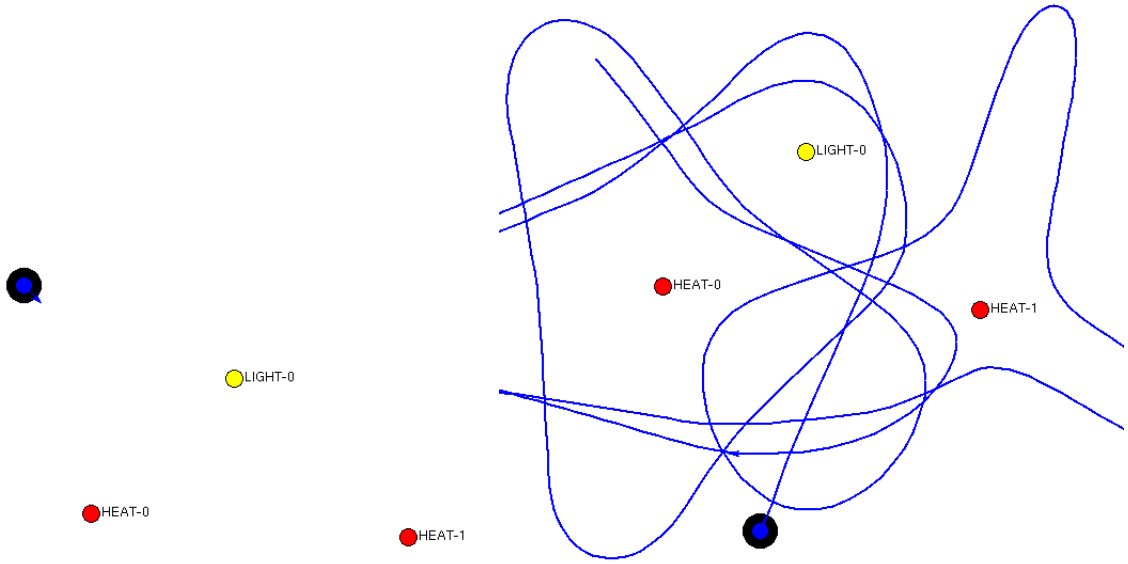


Figure 13: Sensor output for Instinct Vehicle in *BVSim* world

As we see, the sensor data is going to be processed in a special pattern. The output of the sensor is going to behave differently according to the distance of the sensor to the source. In the first period, the sensor's output is going to be greater if the intensity of the signal is greater. Then, it is going to be constant during another intervarl of distance. Finally, when the signal is reaching its maximum intensity, the output of the sensor is going to decrease proportionally to the growth of the signal.

So, how is that going to be reflected in the path followed by a B-Vehicle? Let's see:



(a) Initial scene (b) Final scene

Figure 14: Our instinctive behavior, represented by a B-Vehicle in *BVSim*

For this demo we configurate the Vehicle to have a "Coward" behavior towards Ligth and Heat (being the sensorial input from Light stronger than Heat). As we can see in the figure above, first the Vehicle moves in the middle of *LIGHT-0* and *HEAT-0* (slightly nearer *HEAT-0*). After that, it begins to describe a movement that keep it orbitating around these three sources. The path in blue shows that the Vehicle gets near the source (Heat) but never gets too close (Coward). However, when it is farther, its velocity seems to increase and then it comes back, repeating -more and less- the loop.

We could say - from observing this figure- that the vehicle has a *instinct* to remain close to the heating source, but not too close as to burn itself. However, from what we have just seen, this behavior is originated by changing the function that relates the sensorial data with the output of the sensors.

Here comes the question: What is intelligence? Are we able to say that a Vehicle (or a robot, for that matter) is intelligent just by observing its behavior?

5 Learning- Conclusion

After finishing this mini-project, I think that I have learned the following:

- *Development of a more critical mind and to ask questions to myself* This mini-project started as a short task required for a course. However, while developing it I found out some few situations that posed me questions as a Researcher. For example, the question about *What is intelligence?*. After doing a few dozen tests of *BVSim* (to get good screeshots for this report), I realized that the criteria to qualify a behavior of "intelligent" or "autonomous" must be based on real knowledge of the system, not just in the observation of the reactions, as that can be misleading. i.e. the behaviors just presented.
- Doing some research about mobile robots for this project, I had to use Kinematics equations to give a theoretical basis to my simulator (about how the car moves). As trivial as it was in this case (differential motor), I understood that for working with mobile platforms, there are many things to consider besides an algorithm to navigate between obstacles. It also gave me the chance to read more about Mobile Robots (the book by Siegwart cited in my bibliography)
- Also, about doing research. I had to look for functions that represented the real output of a light sensor, temperature sensor and the like (I don't include smell sensor because that is a theoretic concept). To know how the sensors, or in fact, how "real" stuff works is really important, specially if our software program is going to be implemented. All in all, they (the sensors) are the main communication bridge between the environment and our robot, so a good knowledge of it is essential. Equally important is to know the physical laws that govern mobile robots, given that they act in real vehicles (in the simulator, no physical laws were implemented, but it is a good point to think about).
- As a technical plus, I learned to use LISP and to design simple GUIs with this language. I do hope to use it in future AI applications or projects.
- As a personal benefit, I also learned to work under a time constraint and to divide my time properly between Lab's projects and personal projects.

References

[1] Valentino Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* 1986.

[2] Peter Herth. *LTK - A Lisp binding to the Tk toolkit.* 2006.

[3] Guy L. Steele Jr. *Common Lisp, The Language.* 1984.

[4] Roland Siegwart and Illa R. Nourbakhsh. *Introduction to Autonomous Mobile Robots.* 2004.