Prof. Dr. Gregor Schöner December 7, 2017

**Exercise 11 (live exercise), Dec 7, 2017**

The simulator for this exercise is provided in the file `launcherRobotSimulator`. Running this file will open a main graphical user interface (GUI) window showing a dynamic neural field with a coupled attractor dynamics and control elements, and an additional window showing a top-down view of a simulated robot in a small arena.

The robot is depicted as a gray circle with an arrow indicating its heading direction (think of a simple differential-drive robot like the Khepera here). It has nine directional sensors (such as light sensors) placed equidistant along its front half. The noisy outputs of these sensors are shown in the bar plot in the top part of the window. The sensors respond to targets in the arena, with intensity depending on distance to the target. These targets are shown as smaller red circles in the arena plot.

You can add or remove targets at any time by clicking on the cor- responding button at the bottom of the window and then clicking on a location in the arena.

In the main GUI window, the top plot shows the activation of a one-dimensional field receiving inputs from the robots sensors. The field is defined over the space of robot orientations in an allocentric reference frame (fixed in the world, not rotating with the robot). Note that the x-axis is flipped to allow a more intuitive mapping to the sensor geometry on the robot while retaining the mathematical conven- tions for specifying orientations. The field provides input to the attractor dynamics shown in the bottom plot, in the form introduced in this chapter. The red plot gives the turning rate (rate of change in head- ing direction) for every possible heading direction; the red circle on this plot indicates the actual head- ing direction and instantaneous turning rate of the simulated robot. You can control field parameters and strength of coupling between field and attrac- tor dynamics via the sliders at the bottom (hover over the slider to get a description of the controlled parameter). In addition, you can control the forward speed of the robot via the slider on the bottom right. Clicking the Reset button will reset the field activa- tion and also put the robot back in its initial position. The goal of this exercise is to explore the role of the detection and selection instabilities for the ori- entation behavior of the robot.

1. **Detection Instability**

   When you start up the interactive simulator the robot environment is created with a single target in the upper left. As it is quite far from the robot, the target affects the activation field only weakly. This is the perfect setup to study the detection instability. You can start with the preset parameter values. Use the slider $v_r$ to set the forward speed of the robot to positive values. The robot will drive until it has reached the target and then automatically sets its speed back to zero. You can re-place the robot at its initial position by clicking the reset robot button. At some point during the robots movement, a detection instability will occur. You can pause the simulator when this happens and also reproduce this event several times by using the reset button.

(a) How does the detection instability manifest itself? Input fluctuates due to noise modeled for the sensors. Does the peak fluctuate with input after the detection instability?

(b) What happens to the behavioral dynamics at the detection instability? How does this affect the motor behavior of the robot?

(c) After the detection instability, the peak tracks sensory input from the target. What does this do to the behavioral dynamics?

(d) If you run through this path at a higher speed, the robot turns later in the path. Why?

2. **Selection Instability**

   For selection instability, add a second target and place it to the upper right of the robot. Try to place it at a distance equal to that of the other target. Again, activate the forward velocity and let the robot run.

   (a) Observe the input profile (green curve) and watch how detection instability occurs. What happens to the alternate peak when one target is selected?

   (b) Reset the robot and repeat the trial. Can you observe different selection outcomes?

   (c) By removing and then again adding a target, you can vary its location. Can you manipulate the probability that the target will be selected over the default target? Set up a situation with symmetric targets in which either target can be selected. Turn off the neural interaction by setting all three parameters of the interaction kernel to zero. What happens now when the robot heads for the targets? Go back to the initial setting by quitting and restarting the simulator. Add several targets near the initial target. What does that do to the detection instability? What happens in this case as you approach the target? Do you see a transition from monomodal input to the field to multimodal input? What happens to the self-stabilized peak itself? You can play with the $h$-level to enable sustained peaks or not. Try removing a target right as the robot is heading toward a target, perhaps in the presence of another target. Can you see the effect of sustained activation?

3. **Avoidance**

   You can explore a simple form of avoidance by changing the sign of the coupling from the field into the attractor dynamics to negative (use the parameters button to get access to the parameter values for the attractor dynamics). Explore avoidance behavior as a form of obstacle avoidance.