(La)Place Cells for Robot Navigation

Howard Goldowsky, Tufts University, Medford, MA, United States (howard.goldowsky@tufts.edu)

We model the entorhinal cortex as a Laplace Transform of sensory input signals [2]. The spatial information stored within the transform builds a computational space that affords vector arithmetic, while the transform's inverse builds a geometric space that resembles Border and Place cells [1]. The model navigates a virtual robot within a 2x2 meter room (Figure 1). For training, the robot is guided to a landmark with known coordinates and then led to one of the room's walls. During training the robot builds a "map" of the room, using the Laplace domain as its computational medium. The robot then uses this "map" to navigate to a random previously unvisited location within the room.

Equation 1 describes neuronal dynamics within the entorhinal cortex (EC) [2]. $\alpha(t)$ equals the agent's instantaneous component velocity, $\frac{dx}{dt}$ or $\frac{dy}{dt}$.

$$\frac{dF(s,t)}{dt} = \alpha(t)[-sF(s,t) + f(t)] \tag{1}$$

After substituting the x-component velocity into Equation 1 and applying the chain rule, we get Equation 2. Same can be done for the y-component.

$$\frac{dF(s,x)}{dx} = -sF(s,x) + f(x) \tag{2}$$

Solving for $F(s,x)$ obtains the firing rates of the agent's EC neurons as a function of neuron number and position along the x-direction. This is the Laplace Transform, shown in Equation 3.

$$F(s,x) = \int\limits_0^\infty f(x)e^{-sx}dx \tag{3}$$

Equation 4 shows that the inverse transform can be estimated by a parameterized computation [3].

$$\tilde{f}(x) = \mathscr{L}_k^{-1}F(s) \equiv C_k s^{k+1}\frac{d^k}{ds^k}F(s) \tag{4}$$

The larger k, the more accurate the estimate. C_k is a constant that depends on k. Figure 3 shows $\tilde{f}(x)$, an estimate of the recovered memory of the landmark's position using $k = 24$, which is a memory of the driving function, $f(x)$.

The navigation procedure. Please see Figure 1.

Step 1: The agent is led from the origin to the edge of the room, via the landmark. At the landmark, the robot is notified by a sensory impulse, $f(x)$, modeled as a Kronecker delta (which in real life would be a tap on the shoulder or a hand clap, etc.).[1] Each cell's firing rate then immediately begins to decay by its own $\frac{1}{s}$. Figure 2 shows the firing rate, the Laplace representation, as a function of position.

Step 2: The agent is delivered back to the origin.

Step 3: Navigate back to the landmark.

Step 4: Navigate to a new random via-point. This requires the computation of a new velocity vector $(u,v) - (x,y)$ using the implementation of a subtraction operator within the Laplace domain.

For now, let's limit the math to one dimension. Training has provided $F(s,x)$ and $\tilde{f}(x) = \mathscr{L}_k^{-1}F(s,x)$. The agent wants to return $\tilde{x} - \tilde{u}$, where

[1] In the EC, we use the Real-valued Laplace Transform with $s = \sigma + i\omega$ and $\omega = 0$.