**Designing a Smart Sensor Cushion**

This exercise aims to teach you how to specify spatial and temporal resolution, as well as spatial and temporal scales, in the context of designing a smart sensor application. Today, you will be conceptualizing a smart sensor cushion that monitors sitting posture.

**Key Concepts**

* **Spatial Resolution:** The smallest physical area distinguishable by the sensor. For our cushion, this refers to the size of individual pressure sensors.
* **Temporal Resolution:** The minimum time interval between data recordings. This determines how often our cushion captures sitting posture data.
* **Spatial Scale:** The total area covered by the sensor system. In our case, the entire surface area of the cushion equipped with sensors.
* **Temporal Scale:** The overall time period for data collection and analysis, which could range from moments to months.

**Your Task:** Design a smart sensor cushion that can:

* Detect occupancy (someone sitting).
* Identify and alert about poor sitting posture.
* Track sitting habits over time for health insights.

**Spatial:** Decide on the sensor size and their distribution to accurately monitor posture. (Hint: consider different resolution & scale based on system dynamics, think about the example of selecting the placement of sensors for wind monitoring that we discussed in Lecture 2)

**Temporal:** Determine how frequently data should be recorded and the duration of monitoring to balance detail with resource management.

**Questions**

* What size should each pressure sensor be?
* How closely should they be placed to accurately detect sitting posture?
* How often should the cushion record data to effectively monitor sitting posture without unnecessary data overload?
* What is the total area of the cushion that needs to be covered by sensors to monitor sitting posture effectively?
* Over what period should the cushion monitor sitting habits?

Sketch your cushion design, highlighting sensor placement and specify your chosen spatial and temporal resolutions and scales, justifying your choices based on the application's needs.