Feeling the Swarm: Enhancing Drone Control with Haptics

Darius Dayu GIANNOLI
Contact: darius.giannoli@epfl.ch
Supervisors: Yang Chen, Benjamin Jarvis

2025-11-07

Abstract

To preserve the integrity of upcoming publication, I present an overview of the technologies and methods used rather than specific research outcomes, alongside descriptions of complementary work undertaken during the internship.

EPFL
Laboratory of Intelligent Systems
Switzerland

1 Acknowledgments

I would like to thank the Laidlaw Foundation for giving me the opportunity to carry out this internship, which has taught me so much. A big thank you to Yang and Ben, as well as to BingJian and Pablo,for their warm welcome, valuable advice, and trust. Thank you for integrating me and allowing me to actively participate in all your projects. I would also like to thank all the members of the LIS laboratory for their welcome throughout the summer. Thank you to my friend Gabriel Taieb for helping me through the internship, event though he was sometimes at the other end of the world.

Figure 1: The LIS team going for a summer hike

Table of Contents

1 Acknowledgments

2 Introduction

2.1 Drone Swarms: Collective Power in Motion

Drone swarm systems, composed of multiple Unmanned Aerial Vehicles (UAVs) operating in a coordinated manner, represent a paradigm shift in autonomous robotics. Rather than relying on a single, complex agent, a swarm leverages distributed intelligence to accomplish collective objectives. This collaborative approach provides several intrinsic advantages:

- **Scalability:** The swarm's size can be easily adjusted by adding or removing individual units to fit the mission's scope.
- **Robustness:** The system remains functional even if one or more agents fail, ensuring high fault tolerance.
- **Efficiency:** Tasks can be executed in parallel, enabling rapid area coverage and efficient mission completion.

These capabilities have led to increasing deployment in domains such as search and rescue, precision agriculture, logistics, and defense surveillance.

Figure 2: Illustration of a coordinated drone swarm performing a collective mission.

2.2 Haptics: Touching the Future

Haptics is a field of technology that enriches human-machine interaction by interfacing with the user's sense of touch. By simulating physical sensations, such as targeted vibrations, forces, or pressures haptic systems allow an operator to "feel" a digital or remote environment, not just observe it visually. This tactile feedback channel is critical in complex control scenarios. When an operator is faced with a high volume of visual information (a state known as high cognitive load), haptics can deliver intuitive, low-latency alerts and signals. This ability to transform complex data into a seamless, sensory-driven experience can significantly enhance an operator's precision and reaction time, making it a technology of high interest for telerobotics and vehicle-swarm management.

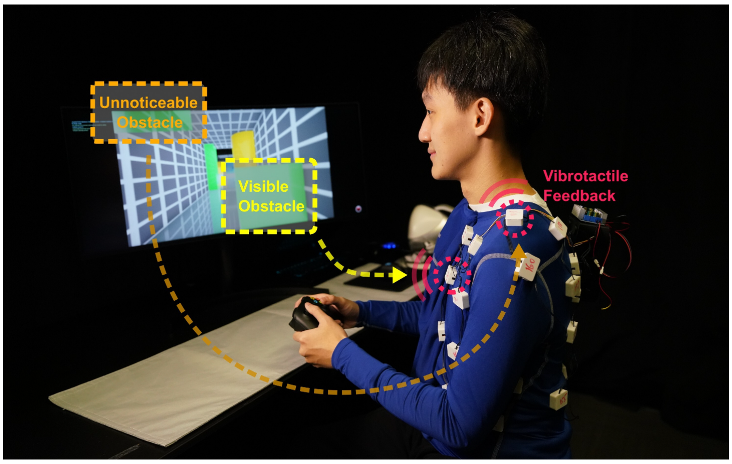


Figure 3: A man wearing haptic devices for vibrotactile feedback to feel obstacles in a simulation

2.3 Haptics Meets Swarms: A Game-Changer

While drone swarms offer significant advantages (as discussed in 1.1), their management presents a profound human-computer interaction challenge. A single human operator is quickly overwhelmed by the task of monitoring the status, trajectory, and sensor data of every agent simultaneously. This results in a high cognitive load and a "bottleneck" in situational awareness.

This is precisely the gap where haptic feedback (Section 1.2) offers a promising solution. By mapping critical swarm-state information such as proximity alerts, collision warnings, or mission waypoints to the operator's sense of touch, we can offload data from the overloaded visual channel.

This "Feeling the Swarm" approach transforms swarm control. Instead of just visually tracking data, the operator can intuitively "feel" the swarm's collective state. This report investigates this synergy, hypothesizing that a haptic-enabled interface can significantly enhance operator precision, reduce reaction times, and improve overall mission success.

Figure 4: A conceptual illustration of an operator using an immersive interface (VR headset and haptic controls) to manage a multi-drone swarm.

3 Devices

The Laboratory of Intelligent Systems (LIS) employs multiple modalities to generate haptic feedback. In this project we combine high–bandwidth *vibrotactile* cues using the VibraForge toolkit with *pneumatic* compression using McKibben artificial muscles.

3.1 VibraForge

Link to paper: arXiv:2409.17420

3.1.1 Intro

To create vibrations, we used the VibraForge haptic toolkit developed by BingJian Huang (University of Toronto).

VibraForge is a scalable prototyping toolkit for spatialized vibrotactile feedback. It comprises *control units* and *vibration units*. Control units receive high-level commands over Bluetooth Low Energy (BLE) and relay low-level control frames to daisy-chained vibration units. Each vibration unit is a self-contained module with a microcontroller driving a miniature vibrotactile actuator. The units are connected with a chain-connection method: a single GPIO/UART channel on the control unit can address up to sixteen vibration units via a lightweight custom protocol.

3.1.2 Control Unit

The control unit decodes BLE commands, translates them to UART frames, and forwards them to the appropriate chain(s). Our design integrates:

- An Adafruit QT Py ESP32-S3 MCU handling BLE and the UART bus.
- Two 3.7 V, 500 mAh LiPo cells providing portable power.
- Four independent chain connectors, each supporting a daisy chain of vibration units (up to 16 per chain in our configuration).

In addition to routing, the firmware supports per-unit addressing, broadcast messages, update rate throttling, and a watchdog to stop all output on link loss or low battery.

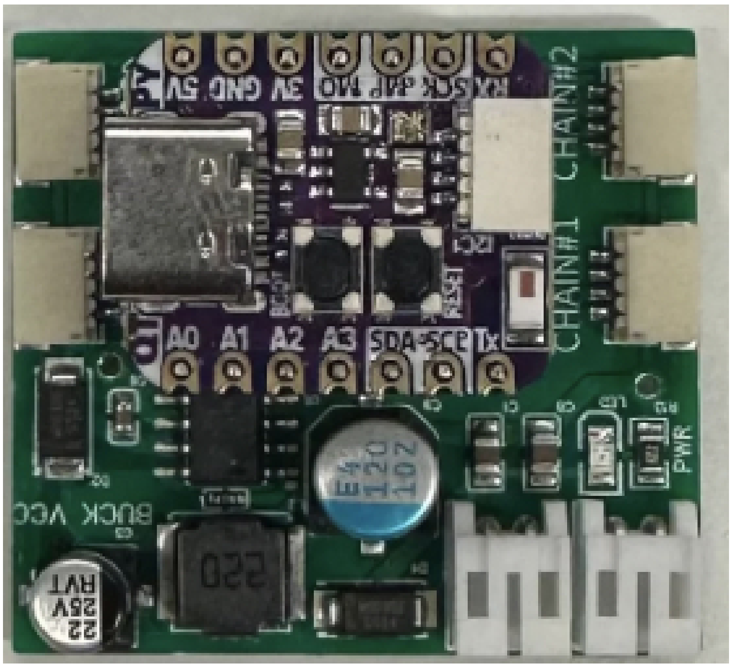


Figure 5: The control unit.

3.1.3 Vibration Unit

We use a voice coil actuator (VCA) as the core haptic rendering element in each vibration unit. The VCA converts electrical signals into precise mechanical vibrations through the interaction of a current-carrying coil within a permanent magnetic field. The unit supports independent modulation of amplitude, carrier frequency, burst duration, and envelope shaping (attack/decay profiles), enabling rich haptic texture generation and precise force control. It also reproduces rapid spatiotemporal patterns such as directional waves, ripples, and frequency sweeps with high fidelity.

Each VCA is individually addressable through the daisy-chain communication protocol, allowing precise control over distributed haptic displays while maintaining a simple wiring topology.

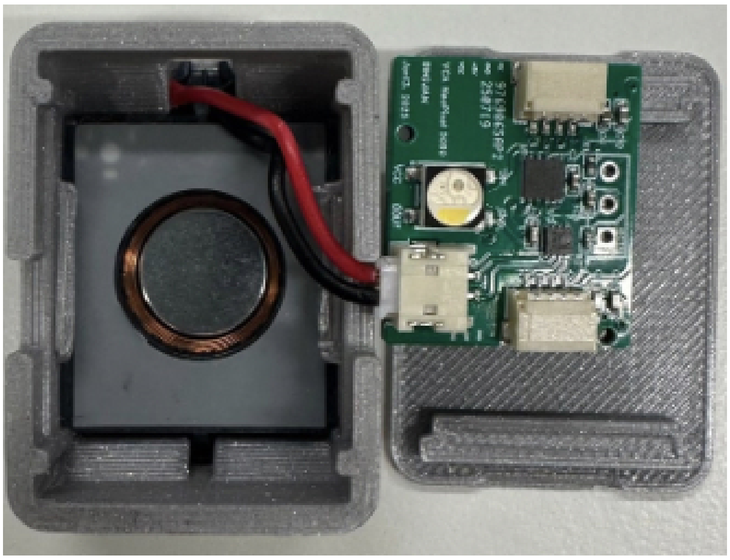
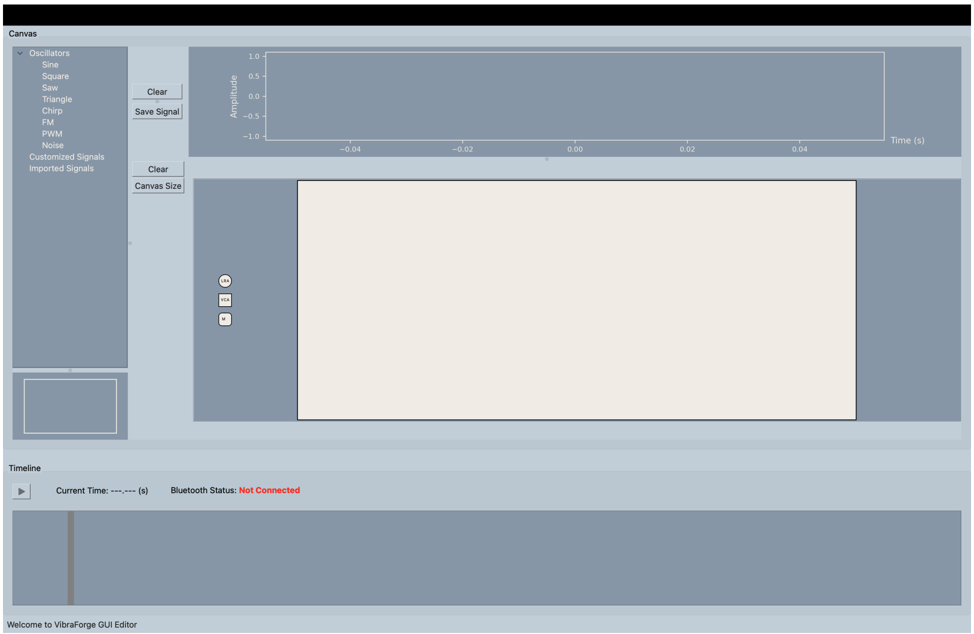


Figure 6: A voice coil actuator (VCA) unit.

3.1.4 Original VibraForge GUI

The original VibraForge editor supports rapid authoring of vibrotactile patterns:

- **Signal design:** generate waveforms from built-in oscillators (sine, square, triangle, noise) or import custom signals.
- **Spatial layout:** place actuators on a canvas to match their physical arrangement.
- **Timeline authoring:** schedule start/stop events to compose complex sequences.
- **Export/stream:** stream patterns over BLE or export to files for later playback.



Current Time: ---.--- (s)

Bluetooth Status: Not Connected

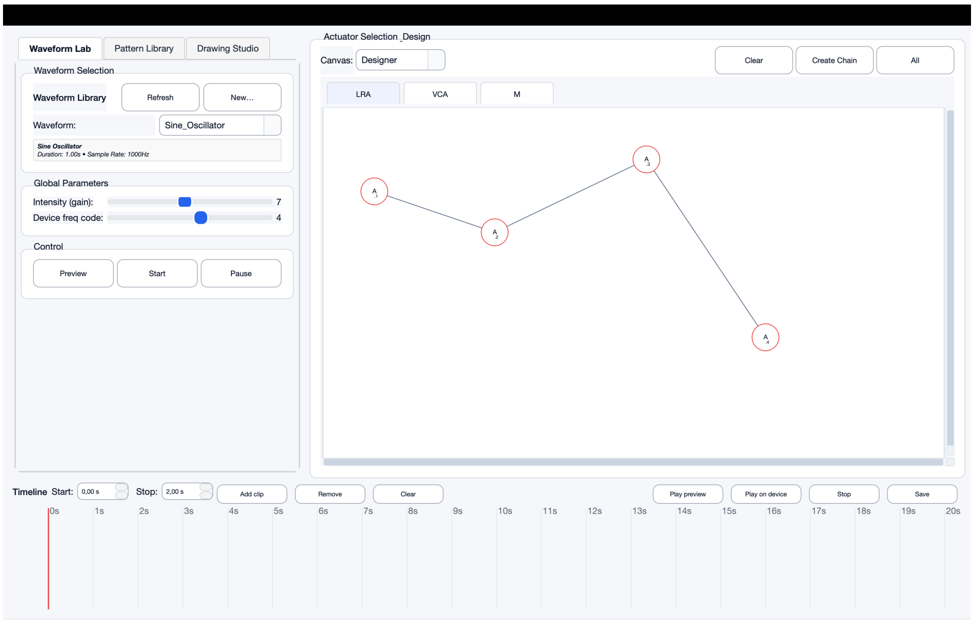
Welcome to VibraForge GUI Editor

Figure 7: The original VibraForge GUI.

3.1.5 Our GUI Editor

We redesigned the interface and extended the feature set while remaining compatible with VibraForge's playback model. Key features include:

- **Waveform Designer:** create custom signals using oscillators, mathematical generators, or import from Meta Haptics Studio (.haptic files). Includes ADSR envelope controls and amplitude modifiers.
- **Canvas layouts:** pre-built templates (3×3 grid, back device 2-4-4-2 layout) with drag-and-drop actuator positioning for rapid prototyping.
- **Drawing Library & phantom rendering:** freehand drawing tool with automatic phantom generation (Park et al. 2016) for spatial sweeps. Supports live playback mode and high-density trajectory creation with adjustable sampling rates.
- **Timeline sequencer:** precise clip-based pattern authoring with individual actuator control, timing parameters, and multi-layer composition.
- **Pattern Library:** save, version, and replay both pre-made patterns (Trio Burst, Sweep, Ring, Pulse Train) and custom sequences for reproducible experiments.
- **Live preview & testing:** per-actuator testing, real-time canvas preview, and device playback with adjustable global intensity and frequency parameters.



Waveform: Sine_Oscillator

Sine Oscillator
Duration: 1.00s • Sample Rate: 1000Hz

Intensity (gain): 7
Device freq code: 4
Canvas: Designer
Start: 0,00 s
Stop: 2,00 s

Figure 8: Our VibraForge GUI.

3.2 McKibben Muscle

A McKibben muscle (pneumatic artificial muscle) is a compliant actuator consisting of an inner elastic bladder encased in a braided mesh sleeve. When the bladder is pressurized, the braid's geometry forces the actuator to *contract* along its length while expanding radially, producing tensile force along the muscle's axis.

Operating principle. Pressure increases generate axial contraction; the achievable stroke is typically on the order of 20–30% of the resting length, with force approximately proportional to internal pressure and dependent on the braid angle. McKibben muscles are lightweight, backdrivable, and inherently safe for contact properties that make them well-suited for wearable haptics.

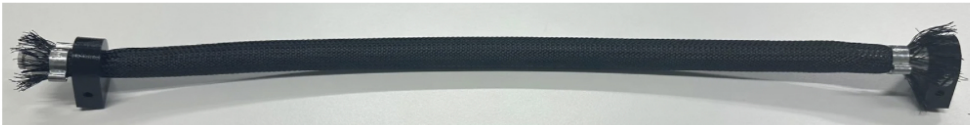


Figure 9: A McKibben muscle.

4 Setup

To suit our needs we developped our own attachment systems and drone swarm simulation.

4.1 Back and Arm Device

To integrate vibrotactile feedback into our haptic control system, we developed a custom wearable devices that ensures reliable actuator-to-skin contact. The back device consists of a flexible substrate fabricated from thermoplastic polyurethane (TPU) via 3D printing. The substrate accommodates 16 vibration units arranged in a 4×4 grid configuration, with each unit secured using velcro attachments for easy repositioning and maintenance. The complete assembly is mounted to the operator's back using adjustable velcro straps, providing secure fixation while maintaining comfort during extended use. We also developed an arm sleeve as shown below :

(a) Back Device. **(b)** Arm Sleeve

Figure 10: The home - made back device and arm sleeve for haptic feedback

4.2 The Unity Simulation Environment

Unity is a cross-platform game engine widely used for its capabilities in 3D/2D visualization and real-time physics simulation. While primarily known for video game development, its robust physics engine and powerful rendering tools make it an ideal platform for creating high-fidelity robotics simulators.

Drone swarms are inherently complex systems. While the LIS DroneDome provides a controlled environment for physical experiments, safely operating a large-scale swarm (e.g., 20 or more agents) simultaneously presents significant logistical and safety challenges.

To mitigate these risks and allow for rapid, iterative development, a dedicated swarm simulator was developed by Pablo Palle using the Unity environment. This simulator allows us to conduct reproducible tests and user studies efficiently before deploying formations and commands on physical hardware.

Figure 11: Compact Swarm.

Figure 12: Spread Swarm.

5 Vibration Modalities

To convey information to the operator, we designed and implemented three distinct vibrotactile modalities. Each modality creates a unique perceptual sensation and is intended for a different typeof communication, moving from simple alerts to complex directional cues.

5.1 Buzz

The Buzz modality is implemented as a high-frequency, continuous vibrotactile signal. This creates a steady, non-changing sensation of vibration for the user.

Its primary function is to convey persistent, binary (on/off) state information. For example, it can be used to confirm that a specific swarm formation is active or to provide a constant, low-priority warning (e.g., "low battery state"). Its continuous nature makes it unambiguous but also highly suitable for background information that the user should be aware of but not necessarily act on immediately.

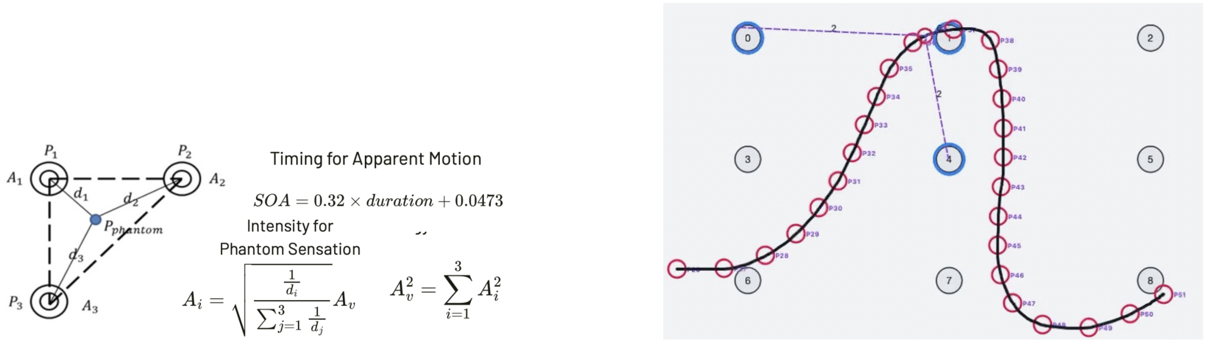
5.2 Pulse

The Pulse modality consists of an intermittent, rhythmic vibration. It is characterized by a repeating pattern of a short vibration burst followed by a silent "off" period.

This modality is designed to be highly salient and "attention-grabbing." By modulating the frequency (how many pulses per second), we can encode dynamic or quantitative information. For example, a slow pulse might indicate a new waypoint, while a rapidly accelerating pulse could signal increasing proximity to an obstacle, thus conveying a sense of urgency.

5.3 Motion

The Motion modality is the most complex and is designed to create a dynamic, directional sensation. It is not produced by a single actuator but by the sequential activation of multiple vibration motors in a precisely timed pattern. By triggering a series of actuators in a line, we create a haptic illusion of "apparent tactile motion", a feeling that feels like a "brush" or "swipe" across the skin.



(a) Motion equations **(b)** Phantoms generator

Figure 13: VibraForge GUI and phantom generation.

6 Study

We aim to publish our research, so a lot of details of our study are omitted on purpose. Only a general overview of our idea and methodology is shown here. Only the necessary details are given, without justifications or explanations. Especially, all of our results are omitted.

6.1 Overview

A significant problem in haptics, especially with vibration, is perceptual confusion. When three or more points vibrate simultaneously, users tend to get confused and cannot reliably recognize the displayed patterns. This is a critical limiting factor that requires further study. Therefore, we are interested in solving this problem through a detailed analysis of the three vibration modalities presented previously.

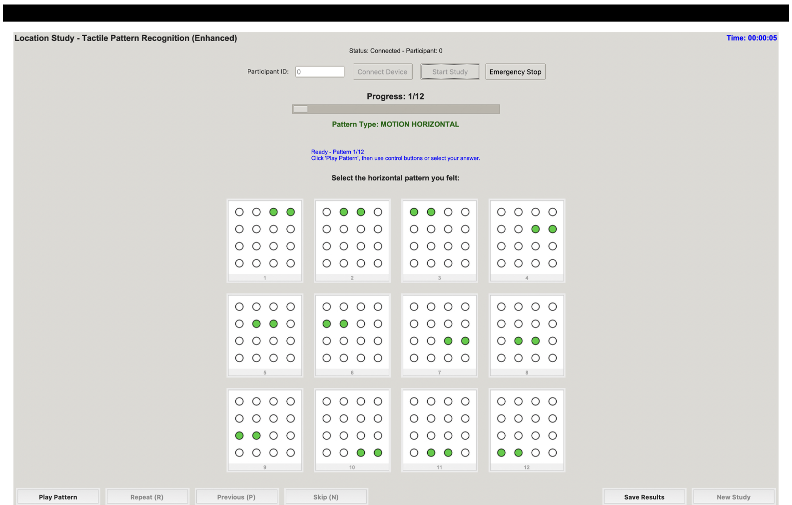
Our goal is to maximize the amount of information we can display simultaneously. To do this, we designed three studies that, when combined, will allow us to create an autonomous algorithm capable of generating optimal pattern combinations.

Below is a succinct explanation of the three phases of the proposed study.

6.2 Graphical Interface

For each of the phases, a distinct Graphical Interface has been developed to allow the user select his answers and gather user data. It was built using **Tkinter**, Python’s standard GUI package, along with its themed widget toolkit (ttk). This includes four separate interfaces tailored to each category:

- **Location:** A 3x4 grid displaying 12 clickable images representing the possible pattern locations for selection.
- **Shape:** A 2x3 grid showing 6 clickable images, one for each of the tested shapes (e.g., cross, circle, square).
- **Size:** A dynamic interface that presents only the available sizes (e.g., 'big', 'medium', 'small') for the specific shape being tested.
- **Direction:** A 3x3 compass-like layout presenting 8 clickable directional arrows (e.g., 'north', 'northeast').



Location Study - Tactile Pattern Recognition (Enhanced)

Time: 00:00:05

Status: Connected - Participant: 0

Participant ID: 0

Progress: 1/12

Pattern Type: MOTION HORIZONTAL

Ready - Pattern 1/12
Click 'Play Pattern', then use control buttons or select your answer.

Select the horizontal pattern you felt:

Figure 14: Our location study interface

In all study interfaces, the standard workflow requires the researcher to enter a participant ID and connect to the serial device. During the study, the user plays a pattern, clicks an image button to record their answer, and is then prompted to provide a confidence rating. The GUI is responsible for the complete randomization of the study sequence and automatically logs the participant's accuracy (correct vs. selected answer), reaction time in milliseconds, and confidence level for each trial. Upon completion, all results are saved to both JSON and CSV files for analysis. A separate diagnostic interface was also developed to manually test and calibrate individual actuators before a study begins.

6.3 Phase A: Single Pattern Comparison

The objective of the first phase is to compare 4 categories: location, shape, size and direction in order to determine which modality between buzz, pulse, and vibration is best to represent each of these categories. Many data is gathered (accuracy, reaction time, Information-Transmission, ...) through rigorous testing, with each time only one different parameter: the vibration type.

We are going to run through every category, and compare the same patterns for each of the vibration modality at random.

The outcomes of phase A are the following: Best Pattern type for each event, Information Transmission rate measurements for each pattern type and better grasp of what can and cannot be represented on the dorsal area throughout vibrations

6.3.1 Location

For the location category, we are going to vibrate a combination of single point and dual points throughout the whole array of 16 actuators at random and the user must select where he felt the vibration by selecting the correct area on the Graphical interface.

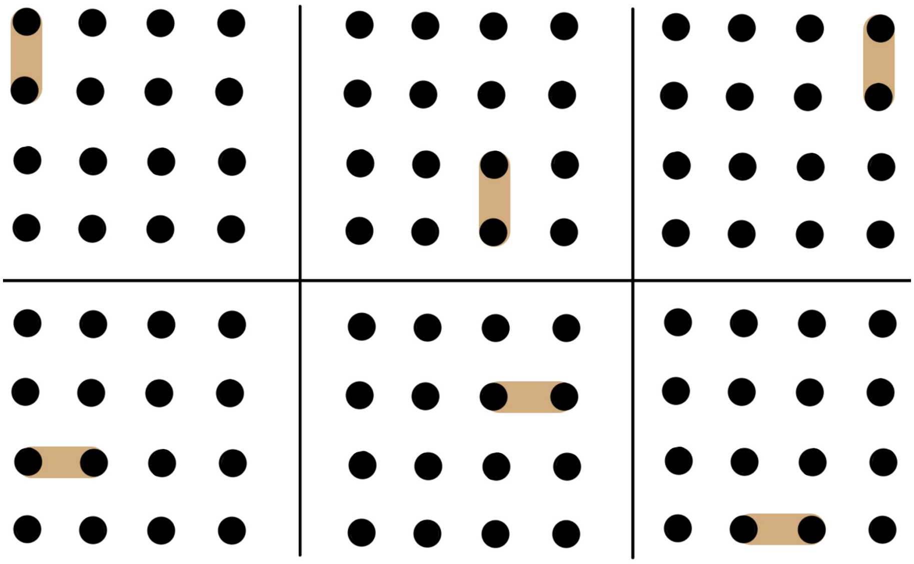


Figure 15: Location Experiment

6.3.2 Shape

For the shape category, 6 distinct shapes will be represented on the user's back and he must select the correct shape on the Graphical Interface.

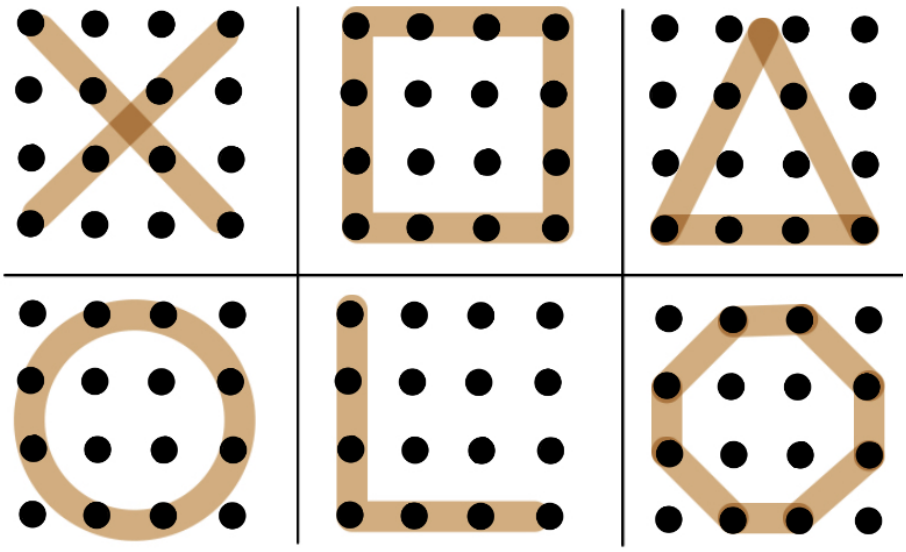


Figure 16: Shape Experiment

6.3.3 Size

For the size category, 3 different vertical and horizontal line sizes will be represented on the user's back and he must select the correct pattern on the Graphical Interface.

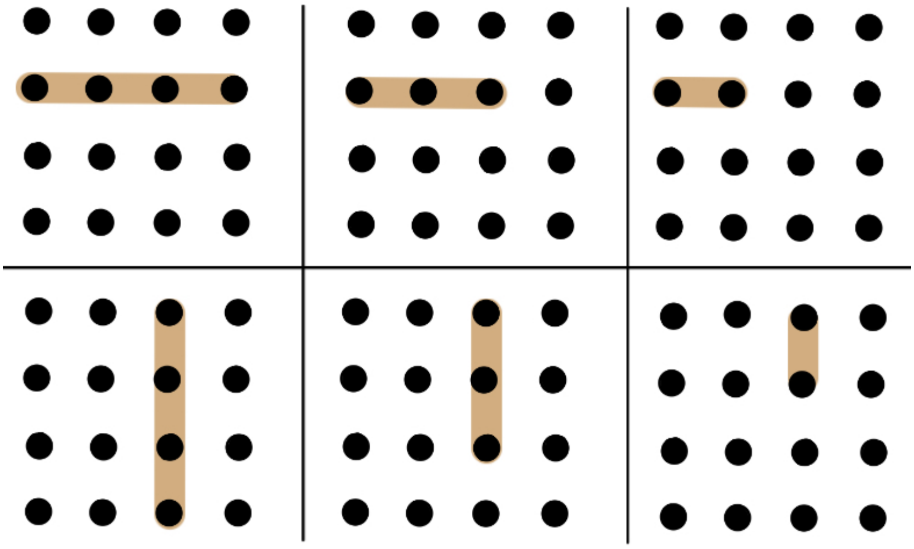


Figure 17: Size Experiment

6.3.4 Direction

For the direction category, 2 cardinal and 4 inter-cardinal directions will be represented on the user's back and he must select the correct pattern on the Graphical Interface.

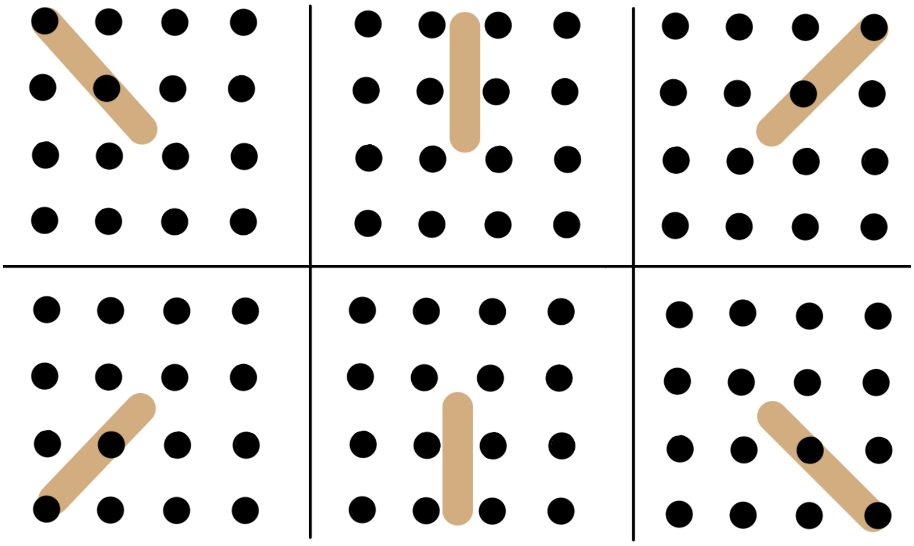


Figure 18: Direction Experiment

6.4 Phase B: Multi Pattern combination

The idea of phase B is to run multiple patterns simultaneously, including conflict zones, to study their interactions. We want to know if some patterns nullify each other, or on the contrary if they complement each other. What are the minimum distances, delays, contact zones, ... All of these questions are answered through this part of the study.

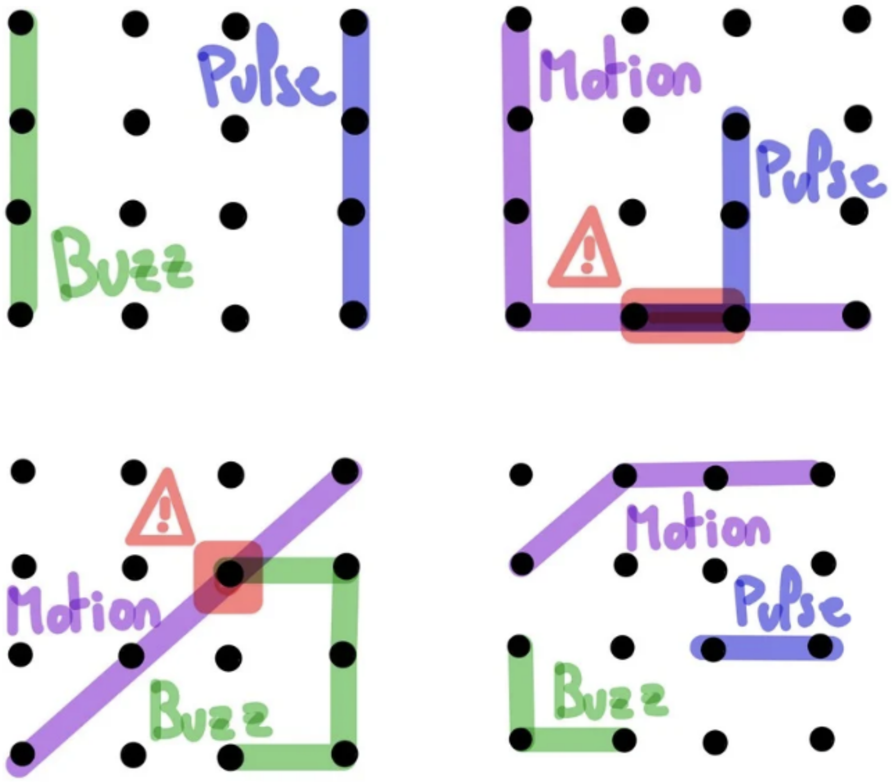


Figure 19: Phase B Experiment

6.5 Phase C: Adaptive Haptic Orchestrator

The objective of phase C is to create an algorithm that uses the results of phase A and phase B to create optimal vibration designs. The algorithm is the core part of what we propose to allow multiple simultaneous and overlapping vibration patterns on a sparse array. For that reason, only an overview of the algorithm tree can be shown.

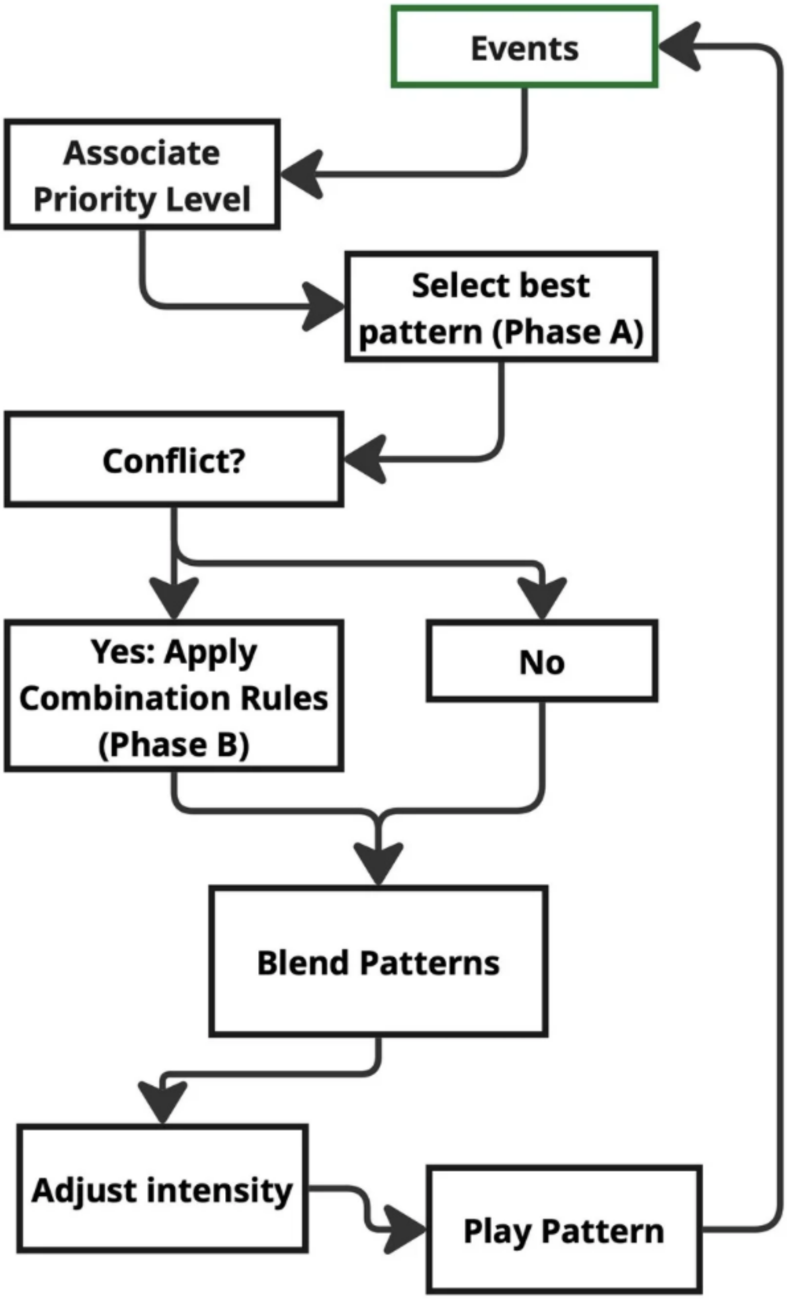


Figure 20: Phase C Experiment