

# Study and development of an ergonomic haptic interface using flexible coils

**Candidate:** Morgan CASALE

**Supervisors:**

Alessandro RIZZO (PoliTO)

Domenico PRATTICIZZO (UniSi)

**Co-supervisors:**

Tommaso LISINI BALDI (UniSi)

Leonardo FRANCO (UniSi)

**Abstract**—This thesis aims to explore the use of flexible PCB coils in haptic applications to create realistic touch sensations. Current piezo actuators, commonly used for generating vibratory cues, have limitations such as limited low-frequency responses and lack of flexibility, which are crucial for realistic and wearable haptic interfaces. Flexible PCB coils, capable of producing low-frequency vibrations and withstanding bending stresses, offer a potential solution. The thesis will first create a mathematical model of the force transmission chain of a voice actuator-style haptic device created with flexible coils, considering also how the human finger reacts to its stimuli. It will then present the necessary electronics for driving the device and conclude with the design and testing of prototypes to evaluate the technology's potential and limitations.

## I. PHYSIC'S MODEL STUDY

The haptic device we tried to develop for this thesis is at its heart a multi-physical system, where electrical, magnetic, and mechanical components interact with each other. Voice coil actuators are characterized by three main components: a coil, a magnet, and a membrane. The coil generates a magnetic field when a current flows through it, which interacts with the magnetic field generated by the magnet, creating a force that moves the membrane. The membrane is then used to transmit this force to the user's finger, which can feel the force as a vibration. In this section, we will try to create a complete model to explain its behavior.

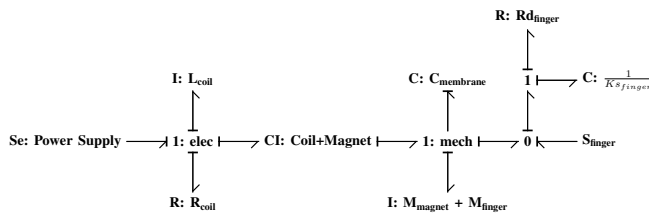


Fig. 1: Bond graph of the coil-magnet-membrane system.

The best way to model this system is by using a bond graph.

### A. Electrical and Power Aspects of Coils

The electrical part consists of the voltage power supply and the coil which is modeled as a resistor and an inductor in series. The coils we're using are planar concentric coils built using flexible PCB technology, this allows us freedom of design in

creating a flexible actuator. Due to their nature, these coils are very thin, which results in high resistance and high heat production. The power flowing through the coil must be kept low to avoid damaging it. This in turn means that the magnetic field generated by the coil will be weak, which can be a problem when trying to create strong haptic feedback.

### B. Electro-mechanical Transducer

The electrical energy is converted into mechanical motion through the magnetic repulsion force between coil and magnet, so the system acts as a transducer.

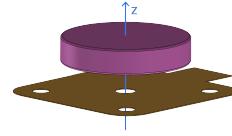


Fig. 2: Coil and magnet position in space.

The force is calculated through the magnetic levitation formula between the magnetic moment of the magnet and the magnetic field generated by planar windings:

$$F = \nabla(\vec{m}_M \cdot \vec{B}_C) = \frac{d}{dz} \left( \frac{B_M(z)}{\mu} \cdot \frac{\mu N I R_C^2}{2(R_C^2 + z^2)^{\frac{3}{2}}} \right)$$

### C. Mechanical Membrane

To limit the motion of the magnet only to the z-axis the magnet is inserted in a flexible membrane, this membrane is also used to transmit the force generated by the magnet to the user's finger. For the last prototype, we designed a celtic-cross-shaped silicone membrane where the magnet is inserted in the center of the cross.

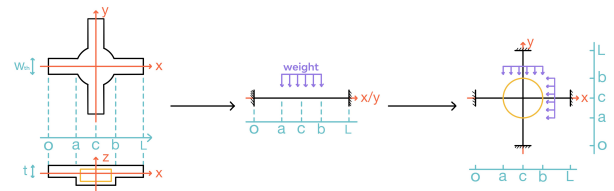


Fig. 3: Membrane structure and mechanical two fixed beams model of the last prototype.

The membrane can be modeled as a spring-damper system, where the spring represents the stiffness of the membrane, and the damping is negligible. Finally, the finger grasping the membrane can also be modeled as a **mass-spring-damper** system where the inertia component is represented by the finger's mass, while the spring and damper represent respectively the stiffness and damping effect of the finger's skin.

## II. HAPTICS OVERVIEW

Computer Haptic Feedback is the research field that deals with the need to be able to digitalize the human sense of touch and reproduce it. The main reason for this is the complexity of the human sense of touch which we still don't understand fully. The human tactile sensing system can **measure specific properties of materials**, such as temperature, texture, force, friction, hardness and viscoelasticity, **through physical contact** between the human skin and the object. Even the **changing state of the interaction**, such as gravitational and inertial effects, can be perceived through the sense of touch. The sense of touch is based on the somatosensory system, which is a complex system of nerve endings and touch receptors in the skin. The somatosensory system is composed of four main types of receptors, each one is specialized in detecting a specific type of stimulus: **mechanoreceptors, thermoreceptors, nociceptors and proprioceptors**. Mechanoreceptors are the most important receptors for haptic feedback, as they are responsible for detecting vibration and pressure. Their sensitivity is limited in both the frequency and amplitude of vibrations they can detect. The force amplitude limit also depends on the constant pressure applied to the skin.

## III. CONTROL UNIT

Using coils to produce magnetic fields of usable intensity requires running them at pretty high currents. This is because the magnetic field strength is proportional to the current flowing through the coil. The goal of our device is to generate vibrations at different frequencies so we need to develop an AC-capable power supply. Standard AC circuits are usually built to generate only low-current signals (in the order of tens of  $mA$ ) but we need to generate signals in the order of  $1A$ . For this reason, the control unit we are going to use has in its signal path a power amplifier based on power op-amps. Power op-amps have an integrated power stage that allows them to deliver high currents with a wide frequency bandwidth. The signal can be generated through a microcontroller with a built-in DAC, in our case we tested the system with an ESP32 board.

## IV. IMPLEMENTATION AND PROTOTYPES

This research was originally born from the willingness to explore the possibility of exploiting the technology of flexible coil inductors by using one produced by the research group **Helmholtz-Zentrum Dresden-Rossendorf** (HZDR) in Dresden, Germany [?]. After testing this coil, we realized its **limitations** and decided to explore other alternatives, such as flexible PCB coils. Flexible PCB coils are much sturdier than

the HZDR coil, easily designable and manufacturable, and have a much higher power rating. The low power rating meant that the HZDR coil was not able to generate a magnetic field strong enough for our application.

### A. Rigid Prototype

The first prototype was designed to test the capabilities of the Dresden coils. In the previous research done by the HZDR team, they tested the coil using a simple piece of **flexible magnetic tape as a membrane**. This membrane is shaped like a "fish" so the tail can be fixed on a plane and the head can be **free to bend up and down**.

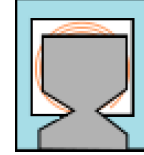
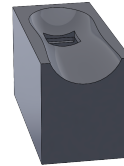


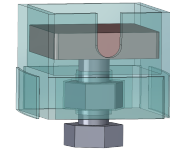
Fig. 4: Dresden coil HZDR test setup

The pulp needed to be suspended at a certain distance to **avoid pressing** on the membrane, this would have caused the membrane to **stop vibrating**.

To solve this problem we developed a structure able to keep the coil and membrane at a certain distance from the pulp.



(a) Finger resting surface



(b) Finger platform

Fig. 5: Components of the first prototype

The coil and the membrane are placed on top of the height-adjustable platform, then the finger would lay on the hole of the resting surface.

### B. Wearable Rigid Prototype

The following prototype was based on the flexible PCB coil. A structure was designed to keep the coil in place and allow it to be worn on the finger through the use of silicon sleeves. This sleeve was designed to be adaptable to different finger sizes and to integrate a cylindrical magnet on the pulp.

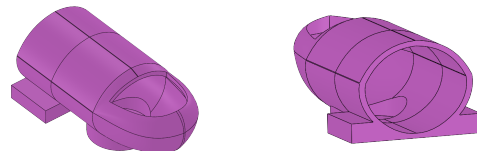


Fig. 6: Finger silicon sleeve front and back view

The silicon sleeve could be then connected to the assembly where the coil and heatsink were placed through its lateral mounting wings.

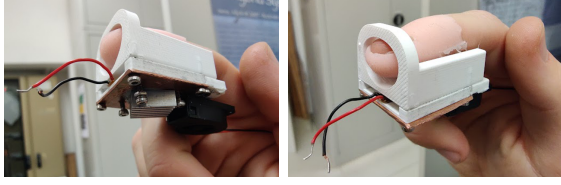


Fig. 7: Bottom and top view of the real prototype

This prototype is much better than the previous one, the magnet was kept at the right distance from the coil and the vibrations were much more noticeable, also being wearable made it much easier to use. Its biggest problem was the silicon sleeve, as the silicon tends to absorb some of the vibrations and the softness of the material made the mounting mechanism a bit finicky. We also had to add a small blowing fan to the heatsink to keep the coil cool as it would heat a lot after a few minutes of use.

### C. Flexible Mat Prototypes

The final prototype was designed to be a flexible silicon mat where all components were integrated, including the membrane.

## V. INTRODUCTION

pizza Satellite missions and on-orbit services are crucial for various applications, and the integration of AI algorithms has significantly advanced technological capabilities. The rise in in-orbit satellites has increased the complexity of navigation and missions, making AI-driven pose estimation invaluable. The EU-funded EROSS project aims to enhance orbital services, particularly in satellite servicing operations, using robotic space technologies for greater autonomy and safety. AI algorithms, with machine learning capabilities, enable precise processing of onboard sensor data, enhancing accuracy in determining a satellite's position and trajectory. Monocular camera-based pose estimation, a notable AI-driven innovation, reduces hardware complexity and empowers satellites to autonomously process visual data for informed decision-making. This increased autonomy minimizes human intervention, reducing operational costs and enhancing mission efficiency. The thesis focuses on implementing AI algorithms for the rendezvous of a non-collaborative satellite, emphasizing mono camera-based visual pose estimation within the 200-20cm distance range. The project delves into critical aspects of pose estimation throughout the entire rendezvous trajectory, showcasing the transformative potential of AI in space exploration and satellite services.

## VI. ALGORITHMS AND METHODS

The methodology involves a pipeline that consists in a *Landmark Regression* module for landmarks' location estimation

in the image, *Landmark Mapping* module for landmark 3D position prediction and *CPD* module for pose refinement. Nine landmarks are manually selected on the satellite CAD model, refined for better recognition.

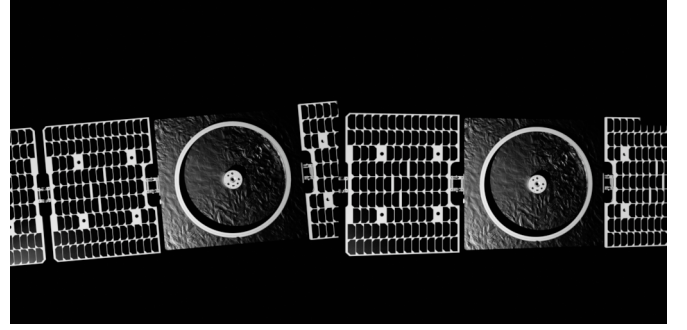


Fig. 8: Sample images from the *TASI EROSS IOD Simulated Dataset N°2*.

### A. Landmark Regression

The training images are pre-processed and coupled with ground truth 2D landmarks and visibility coefficients.

$$v_i = \begin{cases} 1, & \text{if } \mathbf{z}_i \text{ is inside image frame.} \\ 0, & \text{otherwise.} \end{cases}$$

The output of the model is a tensor of 9 heatmaps, one for each landmark  $h(\mathbf{z}_i^p)$ . The ground truth heatmaps  $h(\mathbf{z}_i)$  are generated as 2D normal distributions with mean equal to the ground truth location of each landmark, and standard deviation of 1-pixel. The model is trained from scratch by minimizing the following customized loss:

$$l = \frac{1}{N} \sum_{i=1}^N v_i (h(\mathbf{z}_i^p) - h(\mathbf{z}_i))^2$$

The loss function  $l$  is defined on a single image and in a mini batch  $l$  is simply averaged. The model is trained in 100 epochs with the *Adam optimizer*. Landmark positions are selected from heatmaps based on thresholds and variance.

### B. Landmark Mapping

The Landmark Mapping is a neural network designed to estimate the 3D positions of landmarks using their 2D positions, mapping the 2D-3D relation.

The network has two hidden layers: the first consists in 128 units and processes the input data learning complex patterns and relationships between the 2D and 3D coordinates. The second hidden layer has 64 units and further refines the features learned in the previous layer. The output layer is responsible for regressing the 3D positions of the landmarks. Each landmark is represented by a 3D coordinate  $(x, y, z)$ . The output layer produces these 3D coordinates for all the landmarks.

The model is trained from scratch by minimizing the following loss:

$$l = \frac{1}{N} \sum_{i=1}^N v_i (\mathbf{x}_i^p - \mathbf{x}_i)^2$$

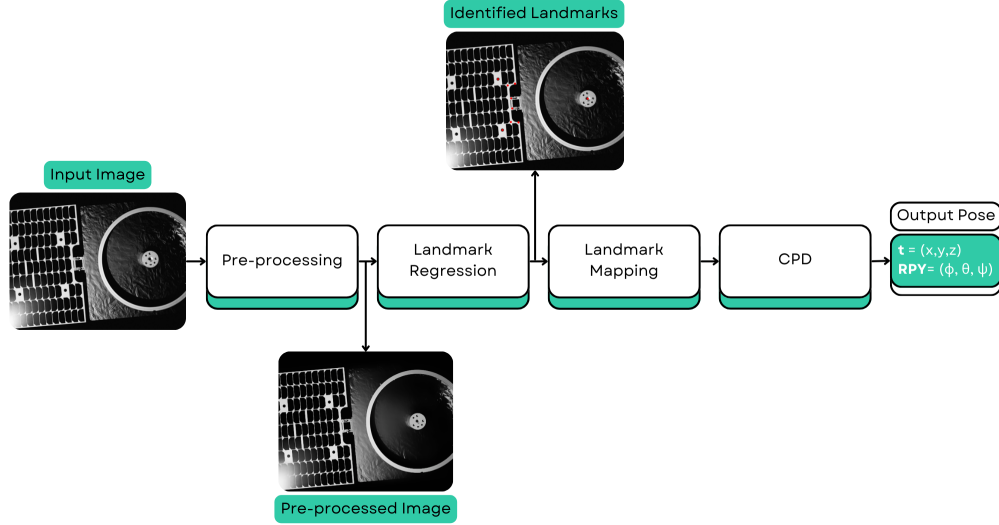


Fig. 9: Online pipeline of the satellite pose estimator

where  $\mathbf{x}_i^p$  represents the 3D position of the  $i$ -th landmark predicted by the model, while  $\mathbf{x}_i$  is its ground truth position. As in the previous model, the loss function  $l$  is defined on a single group of landmarks and in a mini batch it is simply averaged. The model is trained in a maximum of 150 batches with a *Adam optimizer*.

In order to improve the accuracy of the algorithm, three models are created, each of them specialized in a specific range of distance from the target. The training dataset is so split in three subdatasets depending on the distance on the  $z$  axis from the satellite with a 100 images superposition per model from each trajectory. The table below shows the specifications of each model.

Moreover, in order to prevent overfitting, an Early Stopping algorithm is employed. Overfitting is where a neural network becomes excessively specialized in training data, hindering generalization, is mitigated by Early Stopping. The algorithm periodically halts the training process to evaluate the network's performance on a validation set, preventing it from becoming too specialized. If consecutive evaluations indicate declining performance (based on a "patience" parameter), it signals potential overfitting. Training ceases at this point, considering the neural network optimized for generalization to new, unseen data, with the parameters finalized for the model.

### C. Coherent Point Drift (CPD)

The *Landmark Mapping* module, predicting each landmark's position independently, introduces misalignment in the 3D point cloud compared to the rigid CAD model reference. To address this and estimate the satellite's 6DOF pose, the Coherent Point Drift technique is employed. Two sets of 3D points are utilized: the "target" represents landmark positions assuming no translation or rotation, and the "source" represents predicted landmark positions. The accuracy of pose estimation is influenced by the number of visible landmarks,

reducing as some landmarks are cut from the image frame when the camera approaches the target.

## VII. EVALUATION AND RESULTS

### A. Metrics

The final pose estimation scores are derived based on the errors in rotation and translation. For rotation error ( $E_R$ ), the angular distance between predicted ( $q^*$ ) and ground truth ( $q$ ) unit quaternions is calculated using the Hamilton product:

$$E_R = 2 \cos^{-1}(|z|)$$

where,  $z$  is the real part of the Hamilton product between  $q^*$  and the conjugate of  $q$ . Translation error ( $E_T$ ) is the 2-norm of the difference between ground truth ( $\mathbf{t}^*$ ) and estimated ( $\mathbf{t}$ ) position vectors:

$$E_T = \|\mathbf{t}^* - \mathbf{t}\|_2$$

The rotation score ( $S_R$ ) equals  $E_R$ , but in radians, and the translation score ( $S_T$ ) is  $E_T$  normalized by ground truth translation:

$$S_T = \frac{\|\mathbf{t}^* - \mathbf{t}\|_2}{\|\mathbf{t}^*\|_2}$$

The final score ( $S$ ) is the sum of both scores:  $S = S_R + S_T$ . Iterative assessments after each module execution allow dissecting their impact, providing insights into system dynamics and optimization potential.

For the evaluation of the *Landmark Regression* module, the mean error among identified landmarks is computed:

$$E_{2D} = \frac{\sum_{i=1}^N (\mathbf{z}_i^* - \mathbf{z}_i)}{N}$$

where  $\mathbf{z}_i^*$  and  $\mathbf{z}_i$  are the ground truth and predicted pixel coordinates, respectively, for landmarks with  $v = 1$  visibility. Overall module error is the L2 norm of the mean error:  $E_{CNN} = \|E_{2D}\|_2$ .

Similar to the previous, the evaluation of the *Landmark*

Trajectory	$E_{CNN}$ (pxls)	$E_{NN}$ (cm)	$E_T$ (cm)	$E_R$ (°)	$S_T$	$S_R$	$S$
multi-model	$1.38 \pm 0.06$	$1.12 \pm 0.48$	$2.18 \pm 2.09$	0.046	0.0200	0.0008	<b>0.0208</b>
single-model	$1.38 \pm 0.06$	$0.85 \pm 0.35$	$1.85 \pm 1.19$	0.058	0.0168	0.0010	<b>0.0178</b>

TABLE I: Results on train set with multi and single model configurations.

Trajectory	$E_{CNN}$ (pxls)	$E_{NN}$ (cm)	$E_T$ (cm)	$E_R$ (°)	$S_T$	$S_R$	$S$
TRAY_A	$1.34 \pm 0.05$	$1.08 \pm 0.29$	2.96	0.601	0.0238	0.0105	0.0343
TRAY_B	$1.37 \pm 0.05$	$1.89 \pm 1.60$	3.62	0.281	0.0287	0.0049	0.0336
Less_D_Tray	$0.00 \pm 0.00$	$1.27 \pm 0.15$	3.32	0.057	0.0337	0.0010	0.0347
Difficult_Tray	$0.00 \pm 0.00$	$5.58 \pm 17.96$	7.73	0.375	0.0697	0.0066	0.0763
Overall	$1.35 \pm 0.04$	$2.46 \pm 8.38$	4.41	0.328	0.0390	0.0057	<b>0.0447</b>

TABLE II: Results on test set with multi-model configuration.

Mapping module is performed using the L2 norm of mean 3D position error for landmarks with  $v = 1$  is computed:

$$E_{3D} = \frac{\sum_{i=1}^N (\mathbf{x}_i^* - \mathbf{x}_i)}{N}$$

where  $\mathbf{x}_i^*$  and  $\mathbf{x}_i$  are the ground truth and predicted 3D coordinates, respectively. The overall module error is the L2 norm of the mean error:  $E_{NN} = \|E_{3D}\|_2$ .

### B. Experiments

Two comprehensive experiments assess the system’s performance. The initial experiment, conducted on the training dataset, refines parameters through fine-tuning. The *Landmark Regression* module achieves a commendable mean 2D error of  $1.38 \pm 0.06$  pixels, showcasing its accuracy. The Landmark Mapping module undergoes testing in both single-model and multi-model configurations to assess its performance across different proximity bands. The multi-model configuration exhibits jumps during transitions between models, impacting the accuracy of 3D position predictions. The single-model configuration, due to more consistency, surpasses the multi-model configuration accuracy in 3D prediction on the training set.

The *CPD* module introduces additional error during point set alignment, in particular due to its sensitivity to the number of predicted landmarks. Overall, the system proves its robustness during testing, as 3D errors consistently fall within the stringent EROSS project guidelines of less than 2-5 centimeters. Notably, the multi-model setup shows promise but reveals challenges in the abrupt transitions between models, suggesting the need for a more refined transition logic.

In the second experiment, the system is tested on a distinct test dataset. Trajectories *A* and *B* mirror training images light conditions, while *Less\_Difficult\_Trajectory* and *Difficult\_Trajectory* differ significantly, challenging the system’s landmark prediction. Given the Landmark Mapping

module’s sensitivity, evaluation on the latter dataset focuses on Landmark Mapping and CPD modules only, assuming precise landmark predictions (*2D Error* = 0). A nuanced observation reveals an overfitting tendency in the single-model configuration during testing. On the contrary, the multi-model setup displays promising generalization, maintaining proximity in performance between training and testing phases. These results underscore the system’s potential for reliable and accurate pose estimation in spaceborne applications, with areas for refinement identified for future optimization. Tables **I** and **II** show the results on training and test datasets respectively.

## VIII. CONCLUSIONS

This project introduces a specialized monocular pose estimation framework tailored for spaceborne objects, with a focus on satellite rendezvous maneuvers. Utilizing deep neural networks, it seamlessly integrates feature learning and establishes robust 2D-3D correspondence mapping. The inclusion of HRNet, renowned for high-resolution image representation, enhances pose prediction precision. Additionally, the framework employs geometric optimization techniques, ensuring accurate point set alignment and overall system robustness.