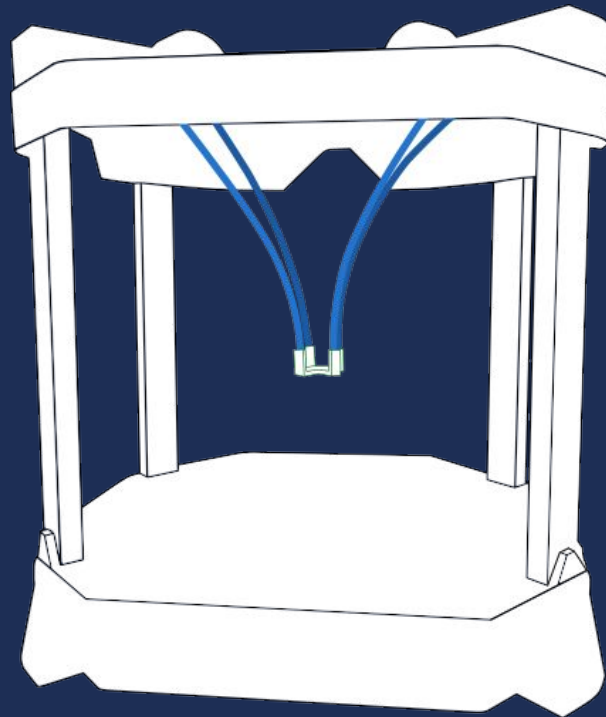


Emio

Apprendre la robotique déformable

Hanaë Rateau



Une plateforme pédagogique



Un robot

+

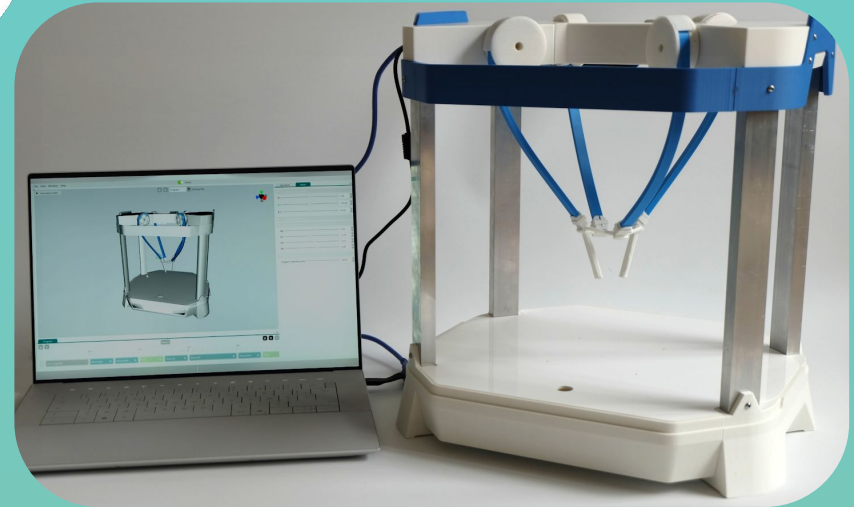


Un didacticiel

+

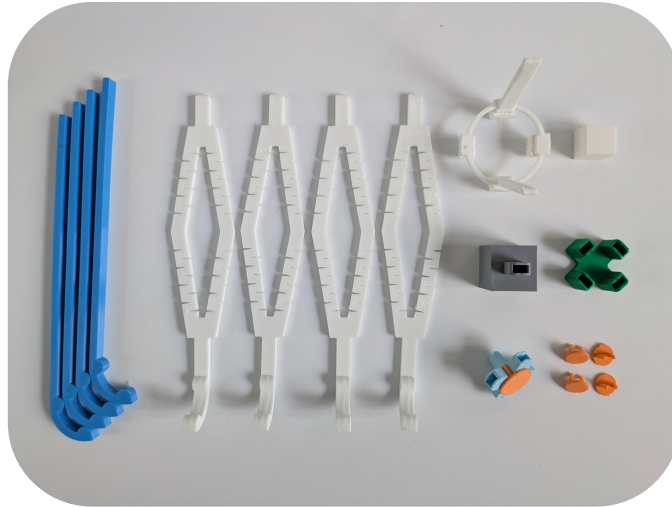
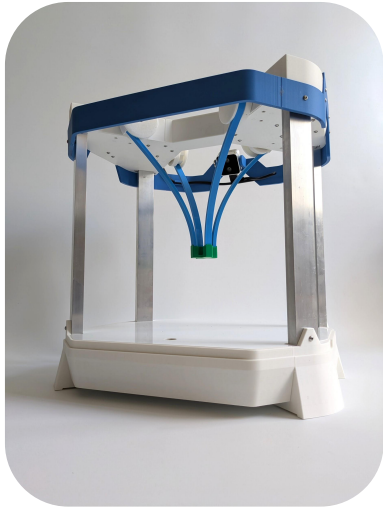


Un simulateur



Un robot - Emio

Reconfigurable et parallèle



Un didacticiel - Emio Labs

- Gratuit avec les contenus pédagogiques.
- Contenus modifiables et création facile via Markdown
- **Quatre TP et deux projets:**
 - Lab 1- Modelling
 - Lab 2- Inverse Kinematics
 - Project 1- Pick & Place
 - Lab 3- Conception
 - Lab 4- Closed Loop
 - Project 2- Inspection

emio-labs v25.x

View Window Help

Previous

Next



Labs Contents

[Introduction](#): discover Emio and its software GUI

[Lab 1. Models](#): discover and analyse different models

[Lab 2. Inverse Kinematics](#): implement your own quadratic program

[Project Pick & Place](#): solve a pick and place task

[Lab 3. Design](#): design your own legs and gripper for Emio

[Lab 4. Closed Loop](#): discover and implement a closed loop controller

[Sandbox](#): set up Emio exactly how you want

Introduction

Overview

Before we begin the lab sessions, let's take some time to explore Emio and its software GUI (Graphical User Interface).

Emio

Software GUI

Let's Try Emio

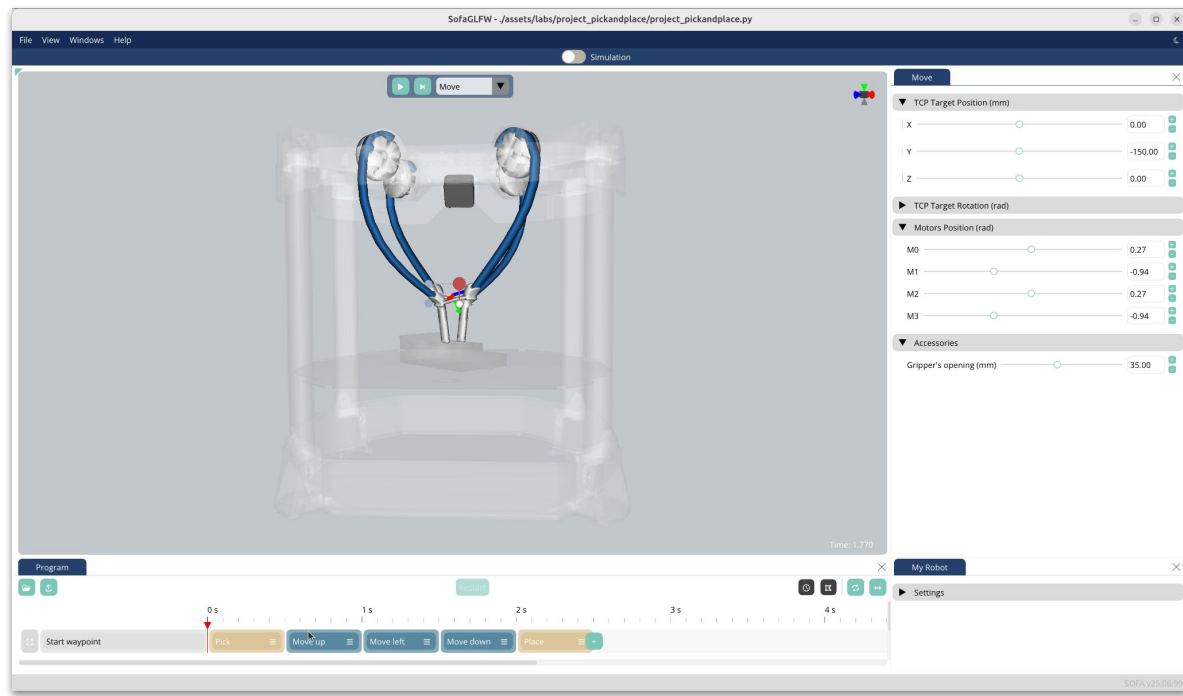
Previous

Next

4

Un simulateur - Sofa Robotics

- Gratuit et libre
- Sofa pour la robotique
- Inclus dans Emio Labs
- Connexion robot facilitée





A venir...



Prochaines versions:



Sofa Robotics

- Disponible sur MacOS
- Compatible avec ROS2



Emio Labs

- Disponible sur MacOS



ROS

API Emio en Python

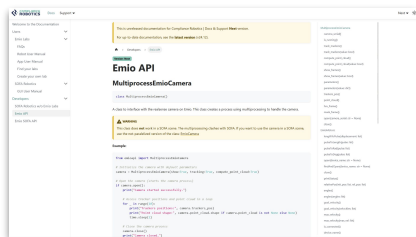
Disponible sur PyPI et sous license GPL3

- `pip install emioapi`

Pas de Sofa

- Contrôle des moteurs
- Et de la caméra

[Documentation](#) et [repo Github](#) disponibles:



```
14 logging.basicConfig(format=FORMAT, level=logging.DEBUG)
15 logger = logging.getLogger(__name__)
16
17
18 """
19 A simple example using both the camera and the motors.
20 Here new commands are sent every seconds to the motors.
21 Because we use a 1s sleep in the while loop, the camera won't be fluid because by default, it is 1
22 """
23
24 def main():
25     emio = EmioAPI()
26
27     # Camera
28     emio.camera.show_frames = True # Show camera frames. Default is False.
29     emio.camera.track_markers = True # Track objects. Default is True.
30     emio.camera.compute_point_cloud = True # Compute point cloud. Default is False.
31
32     # Camera tracking parameters. If None, a config file is read. If there is no config file, defa
33     emio.camera.parameters = None
34
35     try:
36         logger.info("List of all Emio devices: "+str(EmioAPI.listEmioDevices()))
37         if emio.connectToEmioDevice(): # Open the camera with the current parameters. This will s
38             logger.info("Successfully connected to Emio.")
39
40             logger.info("List of used Emio devices: "+str(EmioAPI.listUsedEmioDevices())) # list a
41             logger.info("List of unused Emio devices: "+str(EmioAPI.listUnusedEmioDevices())) # li
42
43             # Motors
44             initial_pos_pulse = [0] * 4
45             logger.info(f"Initial position in rad: {initial_pos_pulse}")
46             emio.motors.max_velocity = [1000] * 4
47             emio.motors.angles = initial_pos_pulse
48             time.sleep(1)
49             emio.printStatus()
50
51             i = 0
52             while i<10:
53                 time.sleep(1)
54                 print("--" * 20)
55
56                 # camera
57                 emio.camera.update()
58                 logger.info(f"Camera parameters: {emio.camera.parameters}")
59                 logger.info(f"Camera show: {emio.camera.show_frames}")
60                 logger.info(f"Camera tracking: {emio.camera.track_markers}")
61                 logger.info(f"Camera compute point cloud: {emio.camera.compute_point_cloud}")
62                 logger.info(f"Camera is running: {emio.camera.is_running}")
63                 logger.info(f"Count tracker positions: {len(emio.camera.trackers_pos)}")
64                 logger.info(f"Point cloud shape: {emio.camera.point_cloud.shape}")
65                 logger.info(f"HSV Frame shape: {emio.camera.hsv_frame.shape}")
66                 logger.info(f"Mask Frame shape: {emio.camera.mask_frame.shape}")
67
68                 print("")
69
70             # Motors
71             new_pos = [((2*3.14)*((i+1)%8)/8)] * 4
```


Nouveaux contenus pédagogiques

- **Tic Tac Toe:** TP IA & Vision par Ordinateur
- **ROS2:** TP introduction à ROS2
- **ROSA:** TP LLM avec ROSA
- **SOFA:** TP introduction à SOFA

Emio et la XR

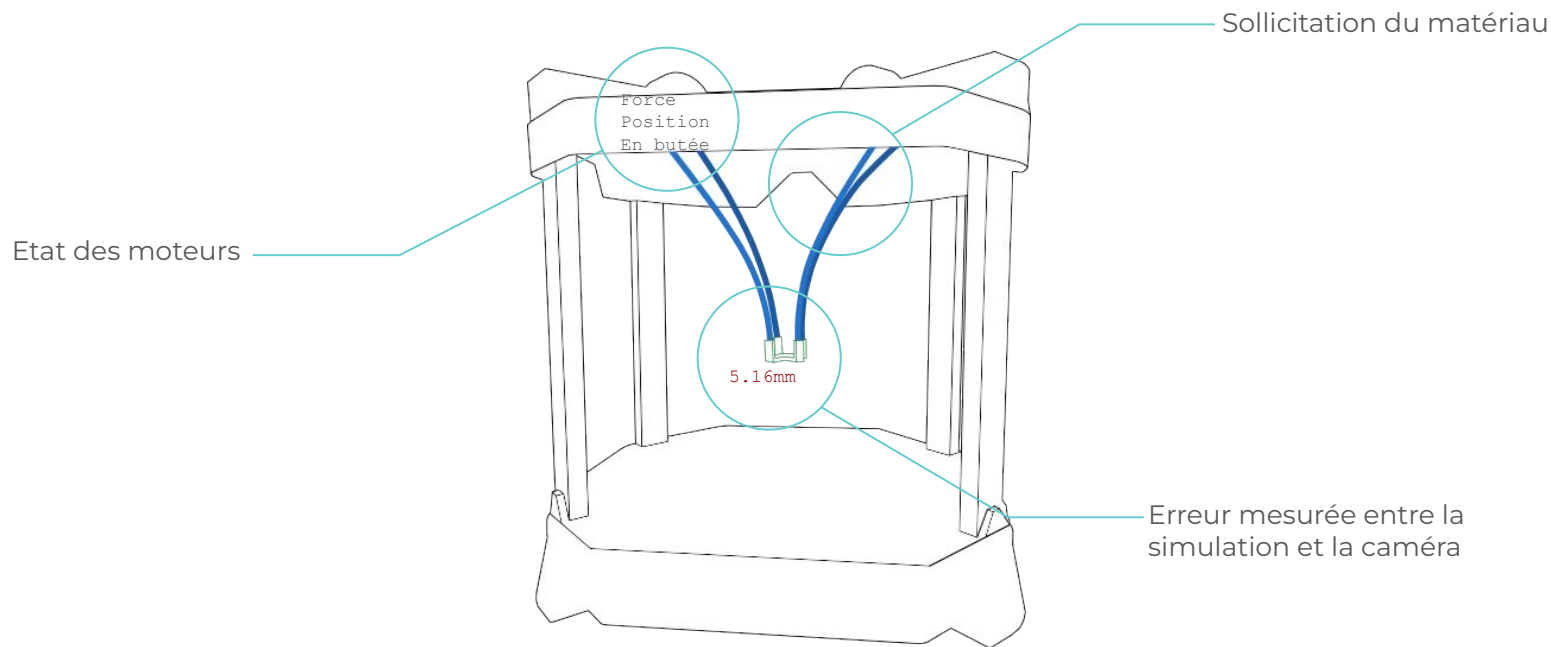


Scénarios XR envisagés

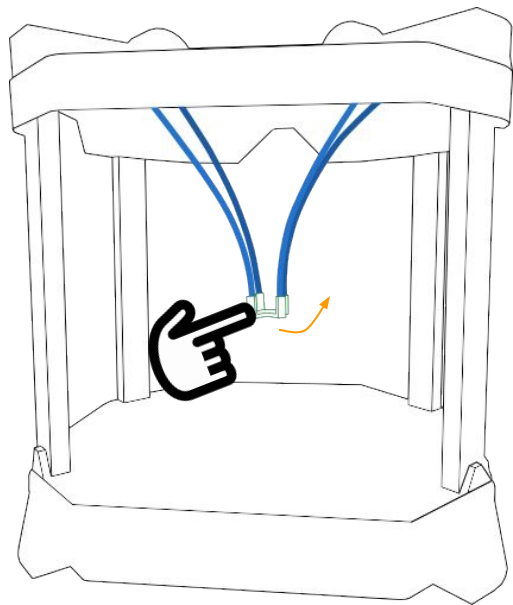
- Visualisation d'état
- Téléopération
- Boucle haptique pour la téléprésence

Visualisation d'état

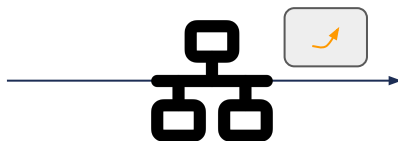
Réalité augmentée portative  et réalité augmentée immersive 



Téléopération - Virtuel → Réel

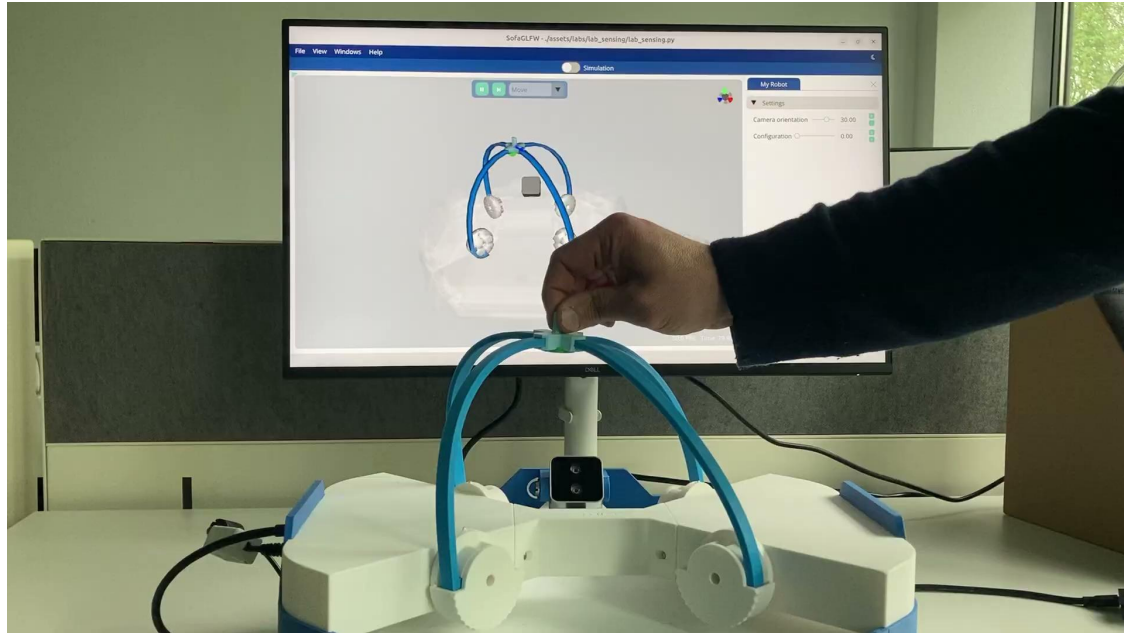


Simulation



Réel

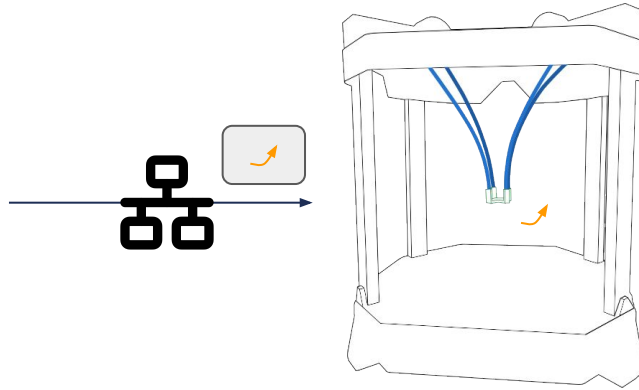
Téléopération - Réel → Virtuel → Réel



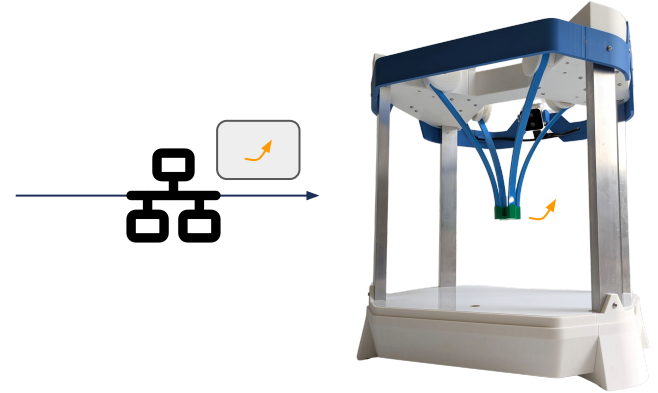
Téléopération - Réel → Virtuel → Réel



Réel

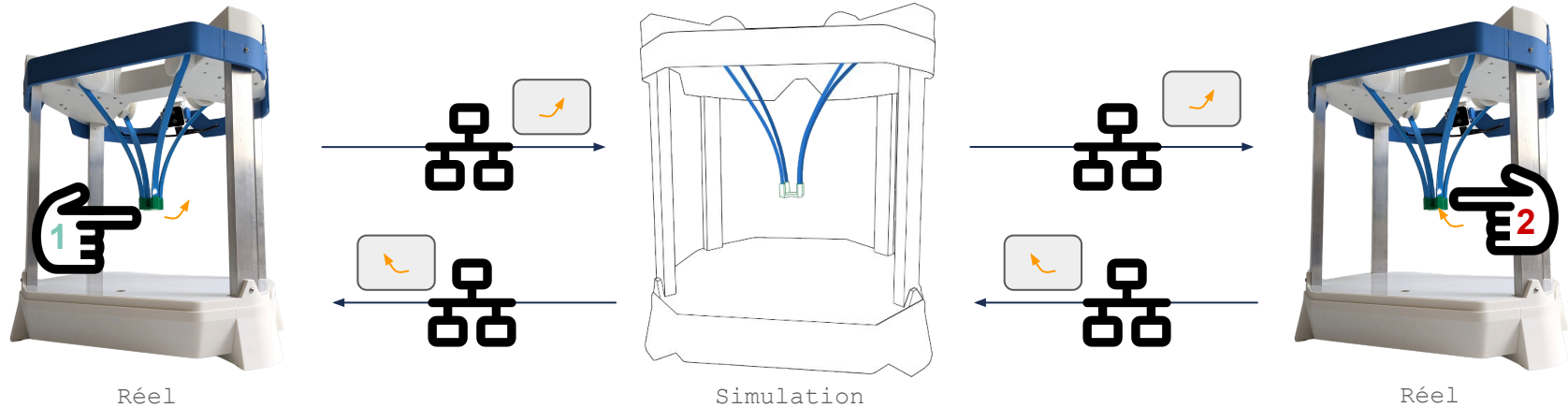


Simulation

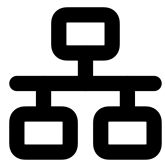


Réel

Boucle haptique pour la télé-opération/-présence



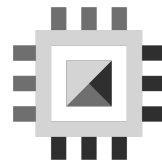
Challenges



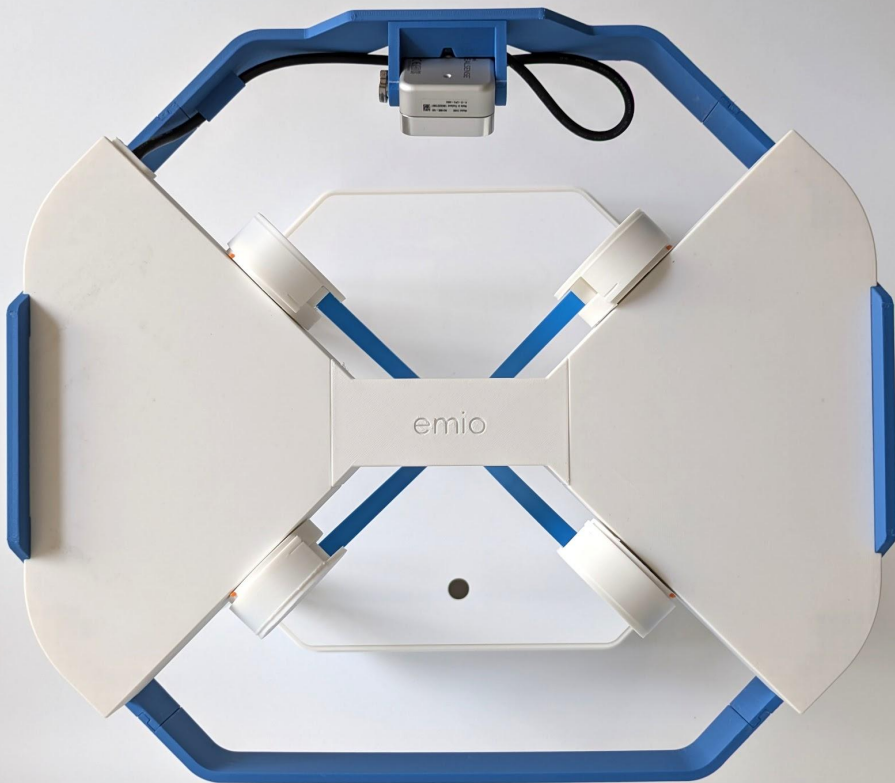
Communication entre
la simulation et la
scène virtuelle



Insérer le jumeau
numérique de la
simulation dans la
scène virtuelle



Vitesse de la
simulation et de
communication pour
le retour haptique



Page web Emio Labs

MERCI

RDV pendant les TechDays TIRREX

