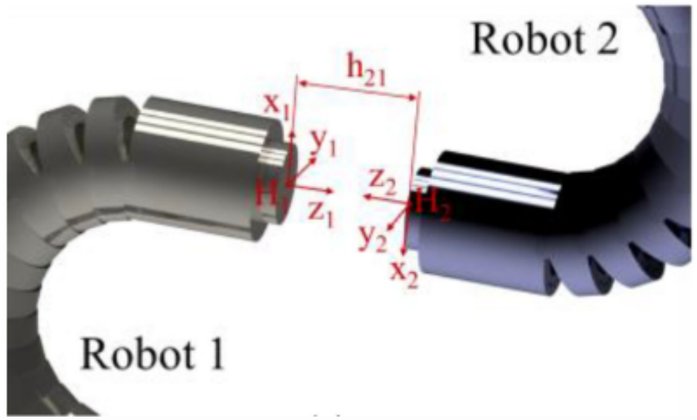
PROJET 5: ROBOT CONTINU BI-BRAS



SOFA

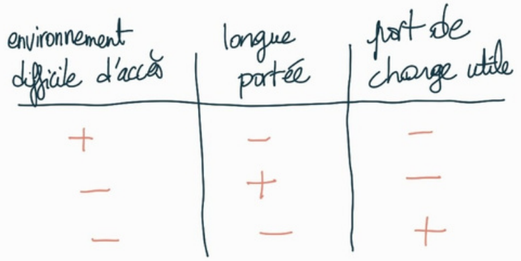
PRÉSENTÉ PAR:

-ALEXIS BOUYER
-GUILLAUME FLAMA
-ANAÏS TAÏATI
-EMNA ADHAR

RÉSUMÉ DE L'ARTICLE

1. État de l'art:

a. Limites et avantages des robots continus individuels



environnement difficile d'accès	longue portée	fort de charge utile
+	−	−
−	+	−
−	−	+

RÉSUMÉ DE L'ARTICLE

1. État de l'art:

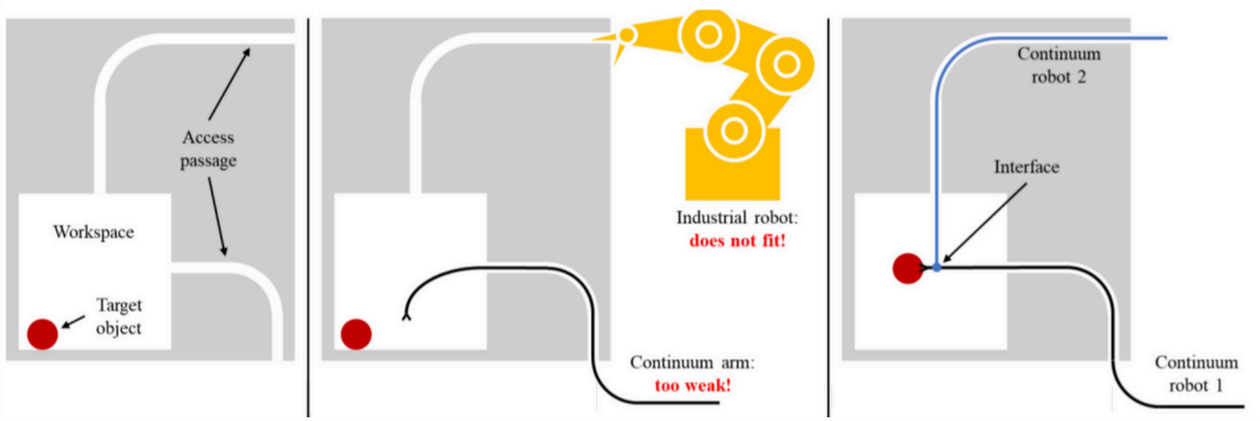
a. Limites et avantages des robots continus individuels

environnement difficile d'accès	longue portée	fort de charge utile
+	−	−
−	+	−
−	−	+

b. Limites et avantages des robots continus coopératifs, jusque là étudiés

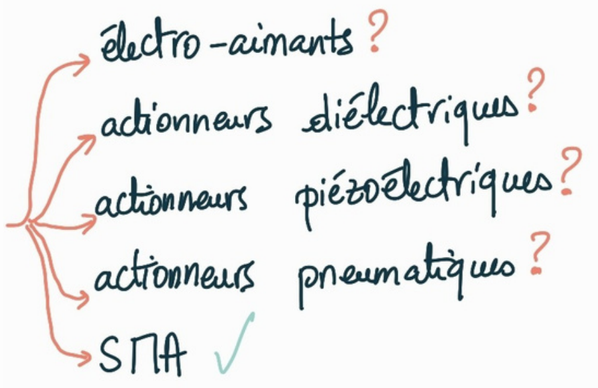
RÉSUMÉ DE L'ARTICLE

2. Solution proposée:



a. Contraintes à respecter

b. solution mécanique choisie: mécanisme de libération basé sur ...



RÉSUMÉ DE L'ARTICLE

3. Tests expérimentaux et évaluations

a. rigidité

TABLE I
DEFLECTION TEST PARAMETERS AND RESULTS

Parameter	RAIN-Snake	FLARE	Cooperating System
Length	1015 mm	715 mm	-
Outer diameter	20 mm	12 mm	-
Inner diameter	8 mm	6 mm	-
Maximum hysteresis	25 mm 2.5%	21 mm 2.9%	6 mm 0.7%
Maximum deflection	139 mm 13.7%	132 mm 18.5%	38 mm 4.5%
Displacement on cycle 1	25 mm 2.5%	67 mm 9.4%	13 mm 1.5%
Max. displ. on cycles 2+	2 mm 0.2%	8 mm 1.1%	2 mm 0.2%
Stiffness coeff.	0.0147 N/mm	0.0112 N/mm	0.0437 N/mm

b. levée de charge utile

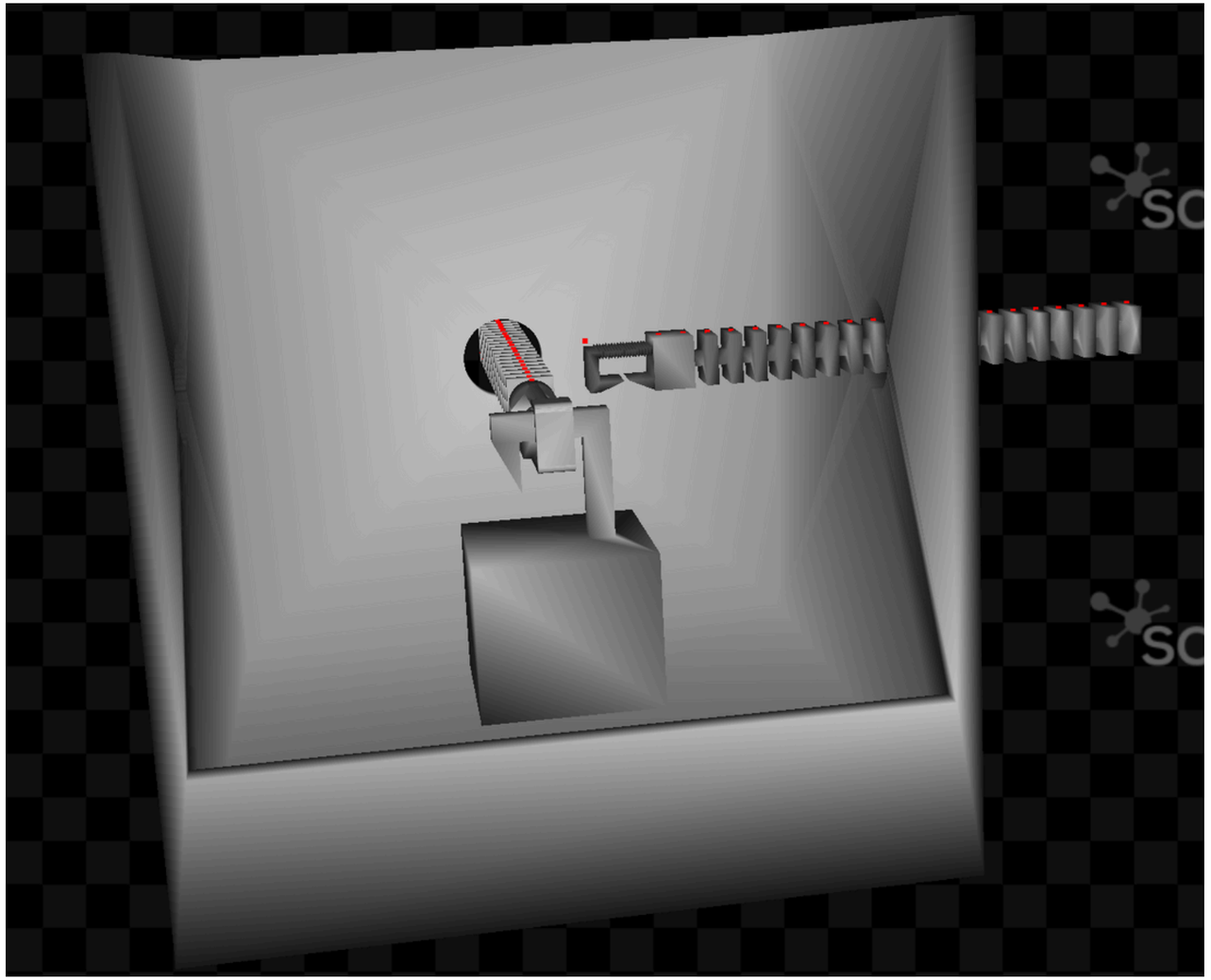
MODÉLISATION DANS SOFA

Dimensions:

Boite: 30cm de côté, D_trous: 4cm

Objet: poids 2kg, E= 69GPa (Rigide en Al)

Bras: poids: 100g, E=0.5GPa



STRUCTURE DU CODE

main.py

```
def createScene(rootNode):···

def createRobot(modeling, pos, rot, num = 1

def createBox(modeling, pos=[0,0,0], rot=[0

def createPoids(modeling, pos=[0,0,0], rot=
```

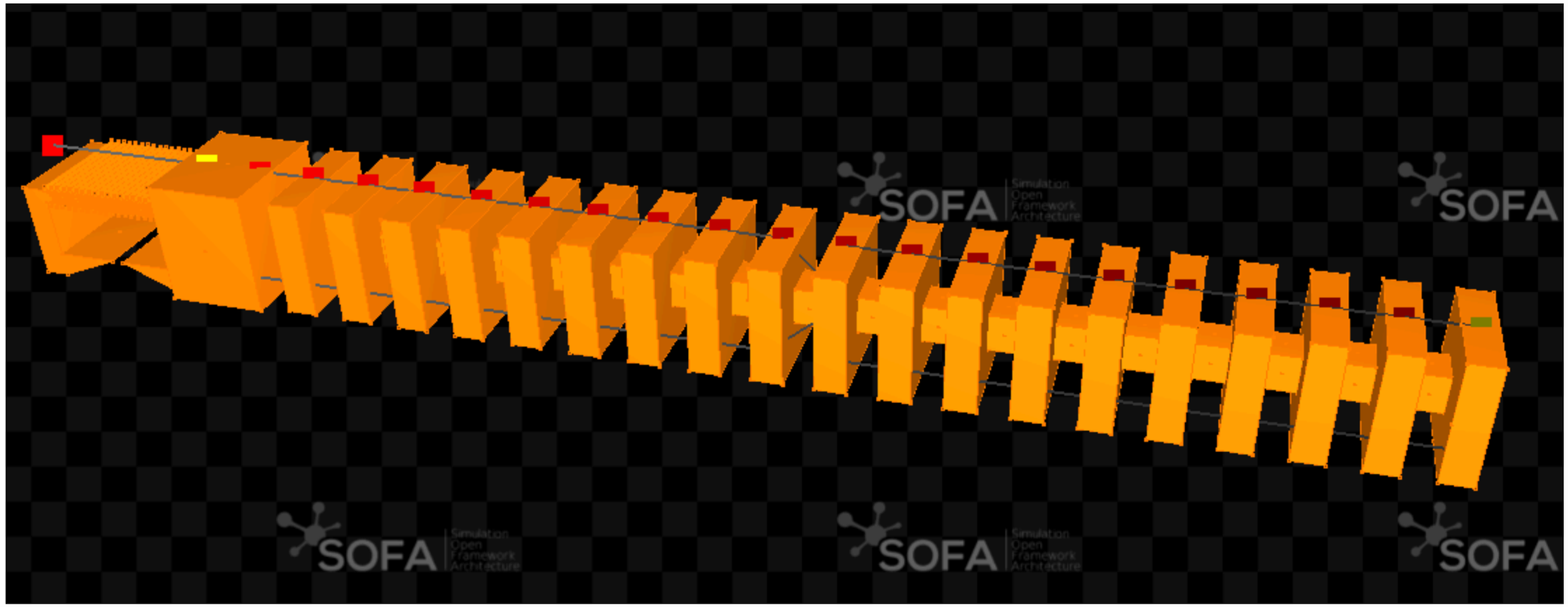
controler.py

```
class ControllerDirect(Sofa.Core.Controller):
    def __init__(self, *args, **kwargs):···

    def onAnimateBeginEvent(self, event):···

    def onKeypressedEvent(self, event):···
```

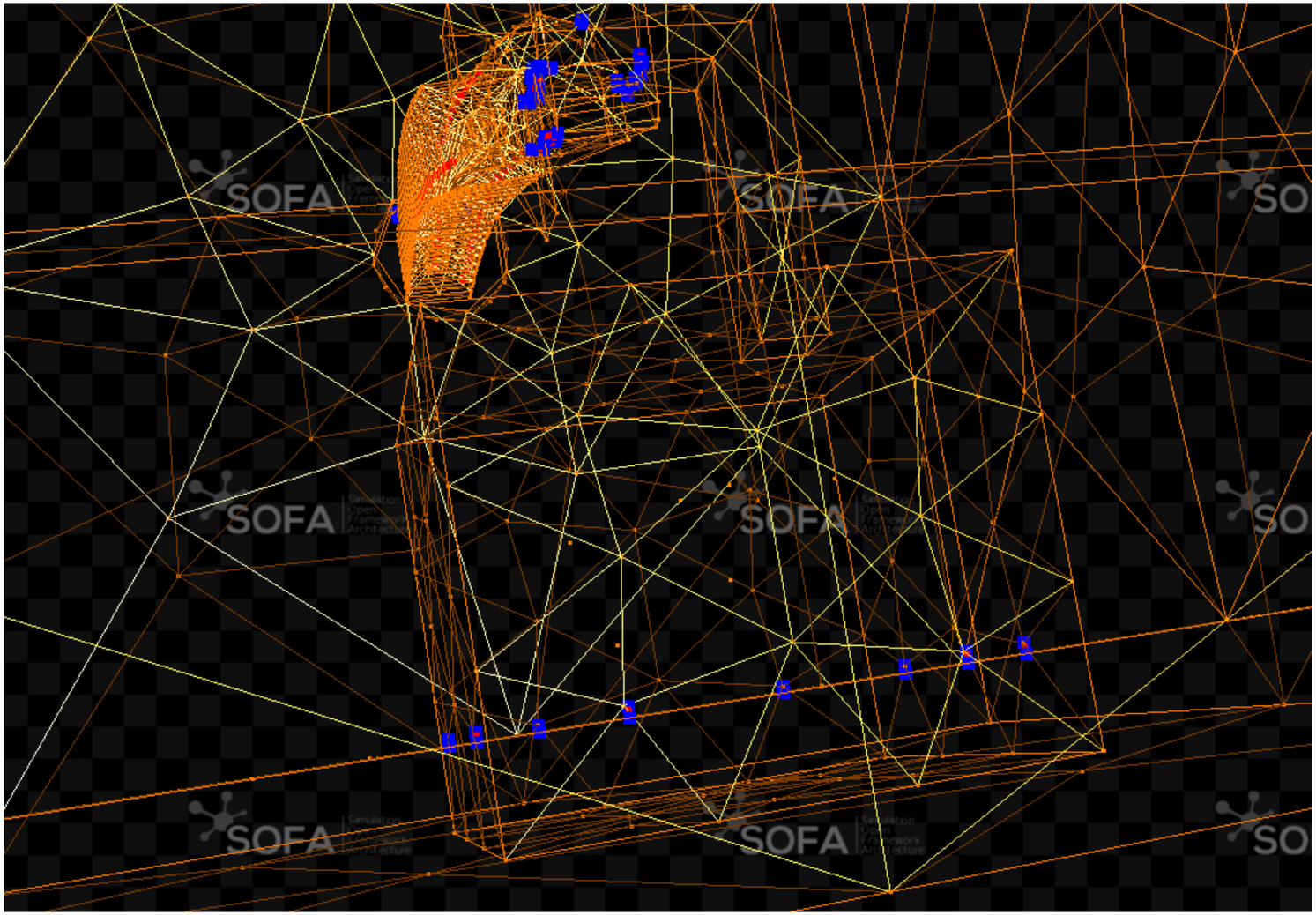
fonctions.py

```
def createObject(robot, filename, pos=[0,0,0], rot=[0,0,0]):

def createVisual(robot, filename, pos=[0,0,0], rot=[0,0,0],

def createCable(robot, i, r, angle, h, pos, rot):···

def createCollision(robot, loader, pos=[0,0,0], rot=[0,0,0],

def setupCollisionPipeline(node):···

def import_plugins(root):···
```

DÉTAILS DE LA SIMULATION

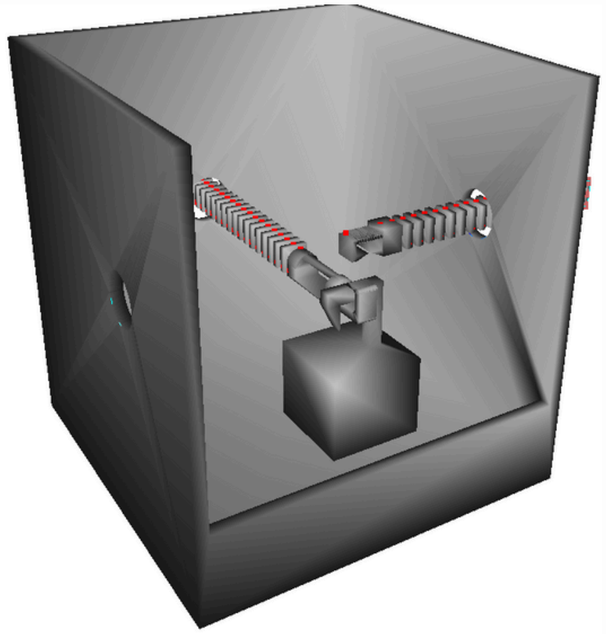


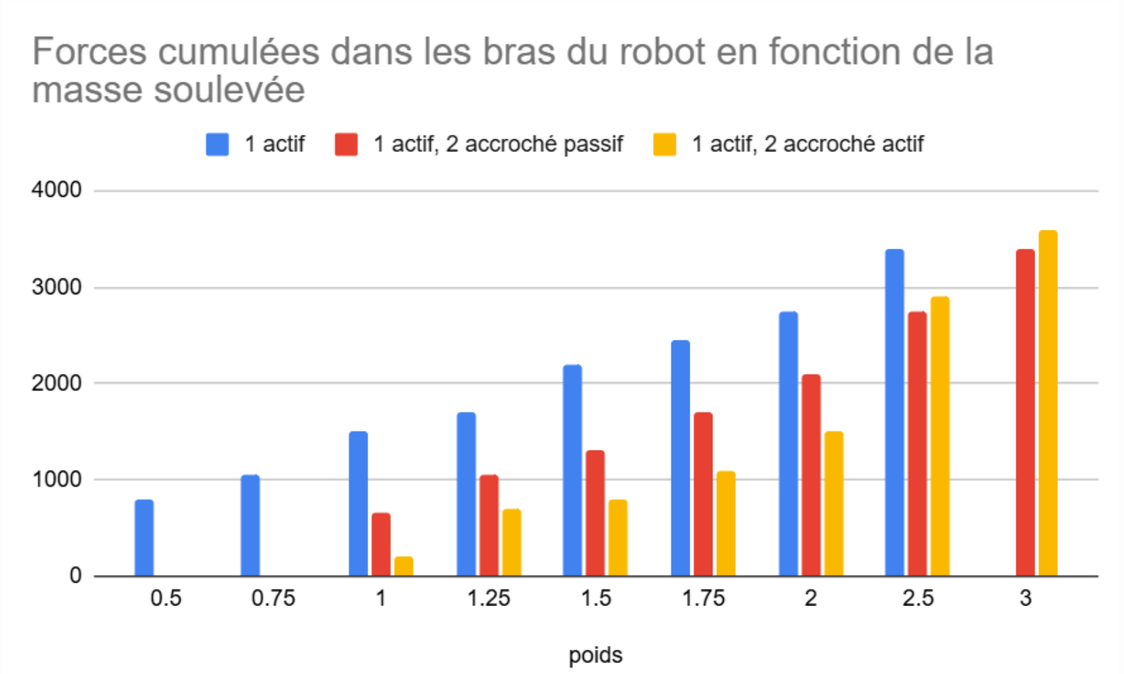
BUGS RENCONTRÉS



RÉSULTATS

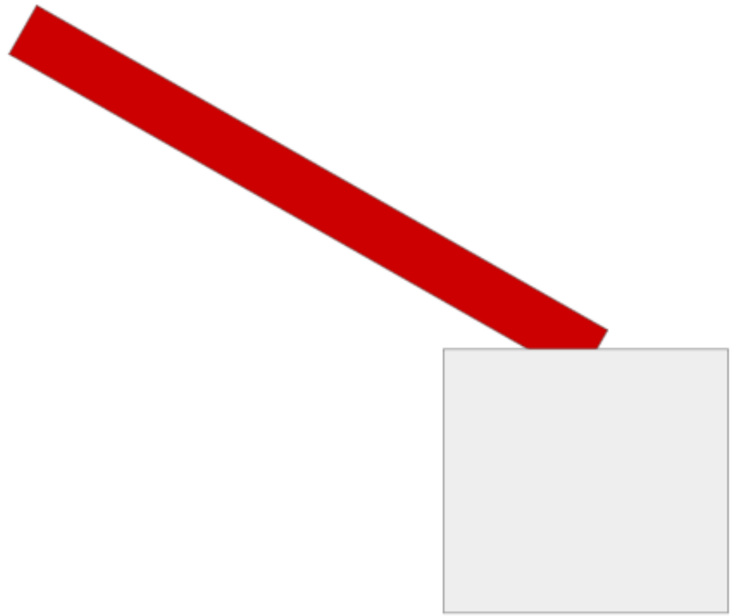
- Force nécessaire pour soulever une mass
- 3 type de levage



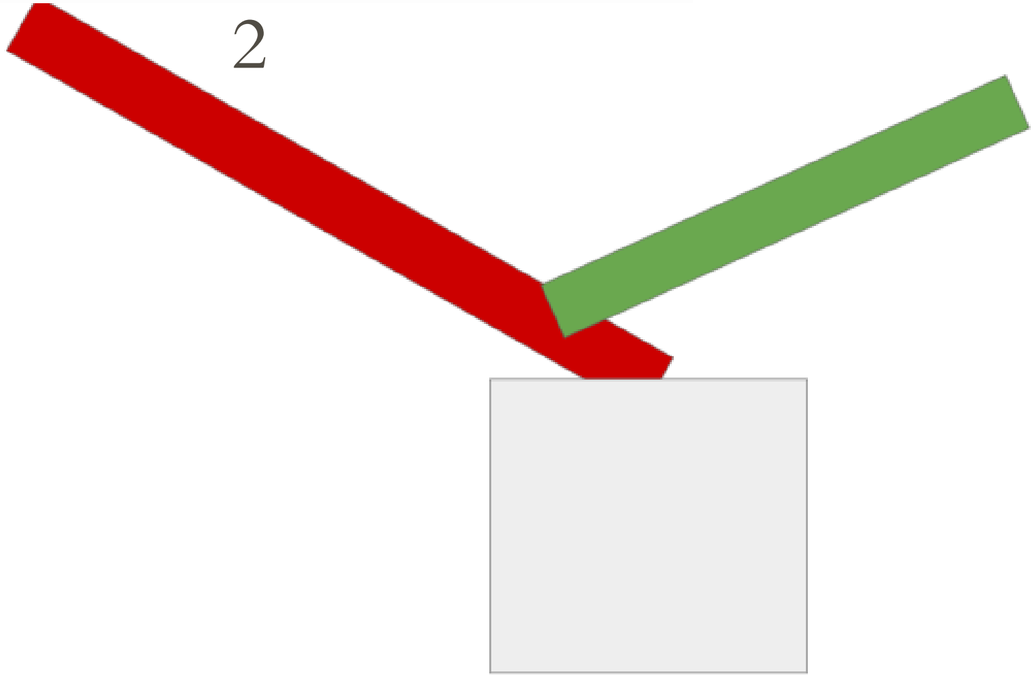


EXPÉRIENCES

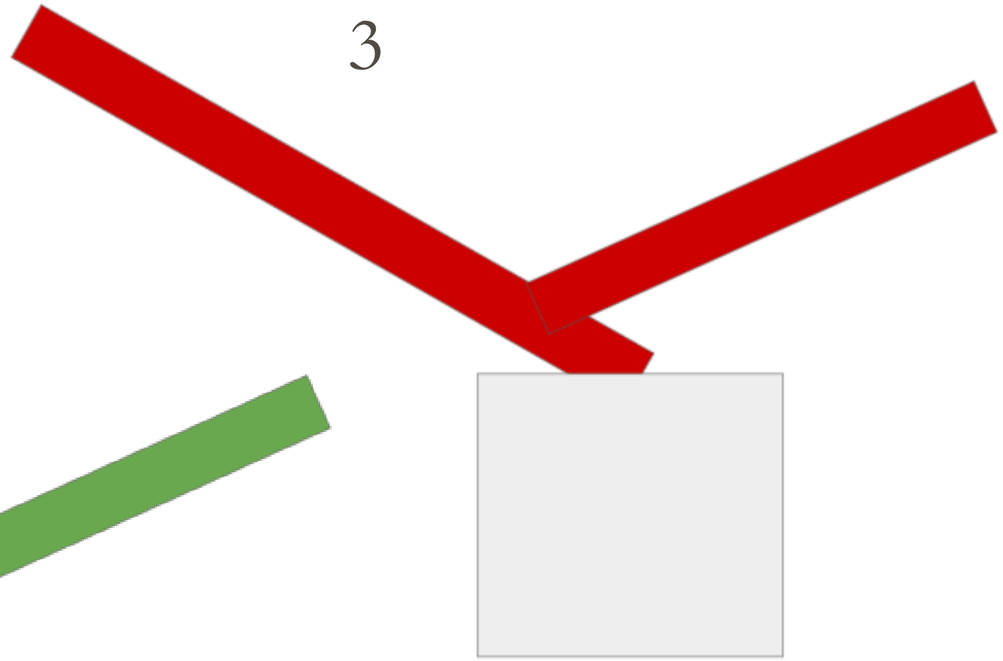
1



2

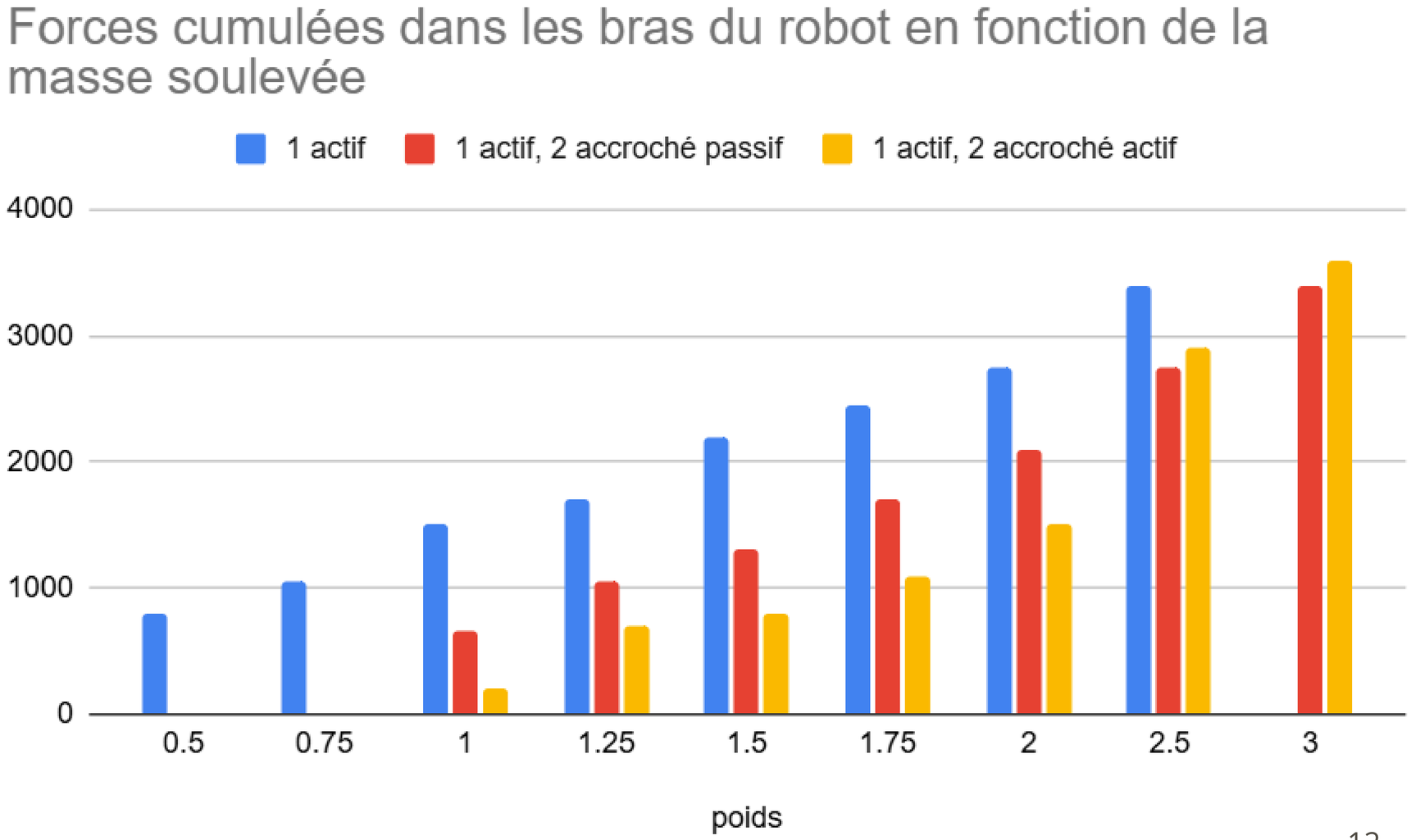


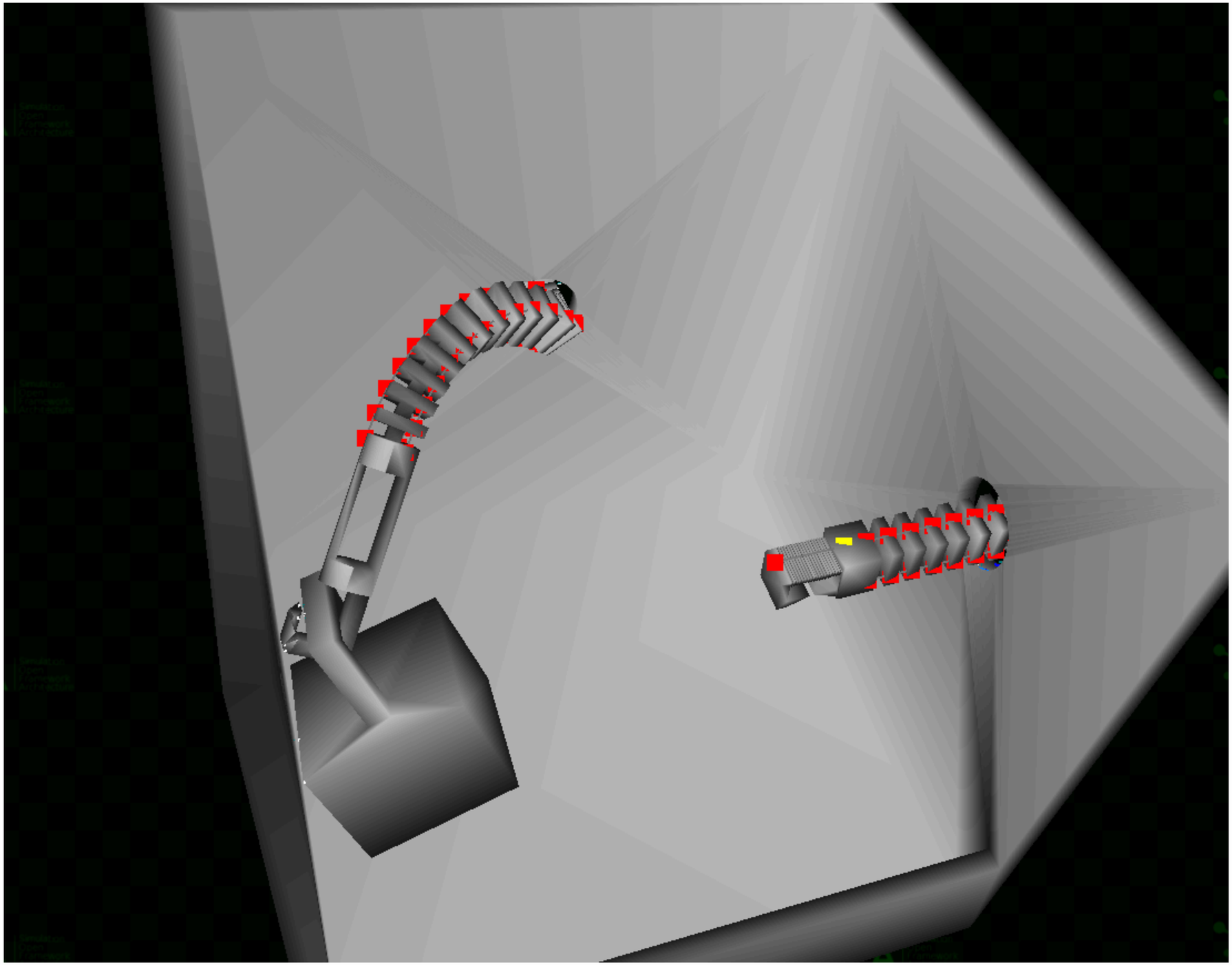
3



ANALYSE

- Rigidité
- Stabilité





MERCI POUR VOTRE ATTENTION