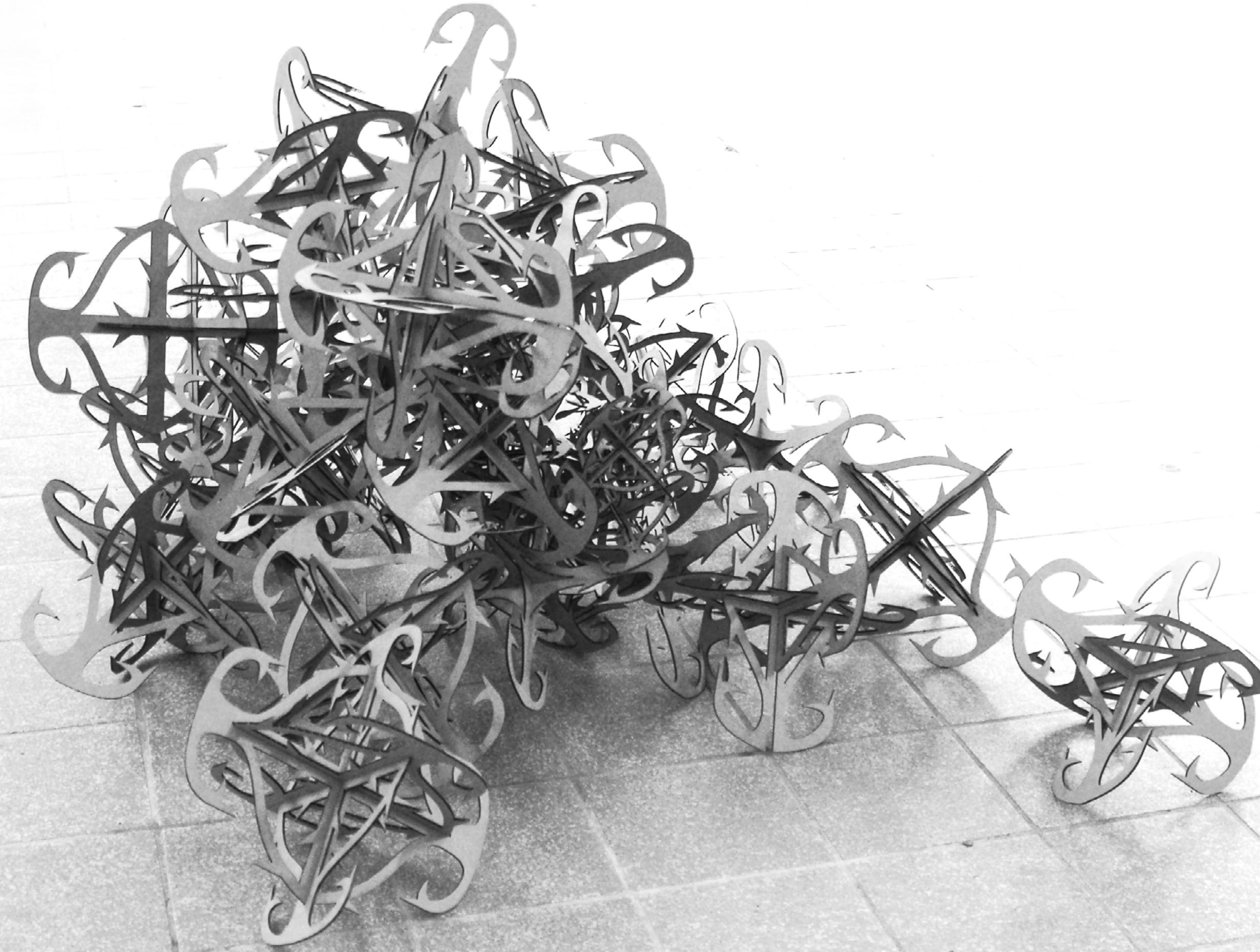


# OFF THE HOOK

Charles Bouyssou

Oswald Pfeiffer

Mathieu Venot



Philippe Morel // Jean-Aimé Shu

Malaquais - Digital Knowledge 2014-2015

# INTRODUCTION

## Positionnement théorique

L'objectif est de concevoir un système de construction autonome et robotisé exploitant les capacités d'auto-organisation des structures en agrégats.

L'utilisation de capteurs (vision machine) permet au système d'avoir un feedback de son propre environnement, et donc de corriger en temps réel les erreurs induites par un matériau au comportement imprévisible.

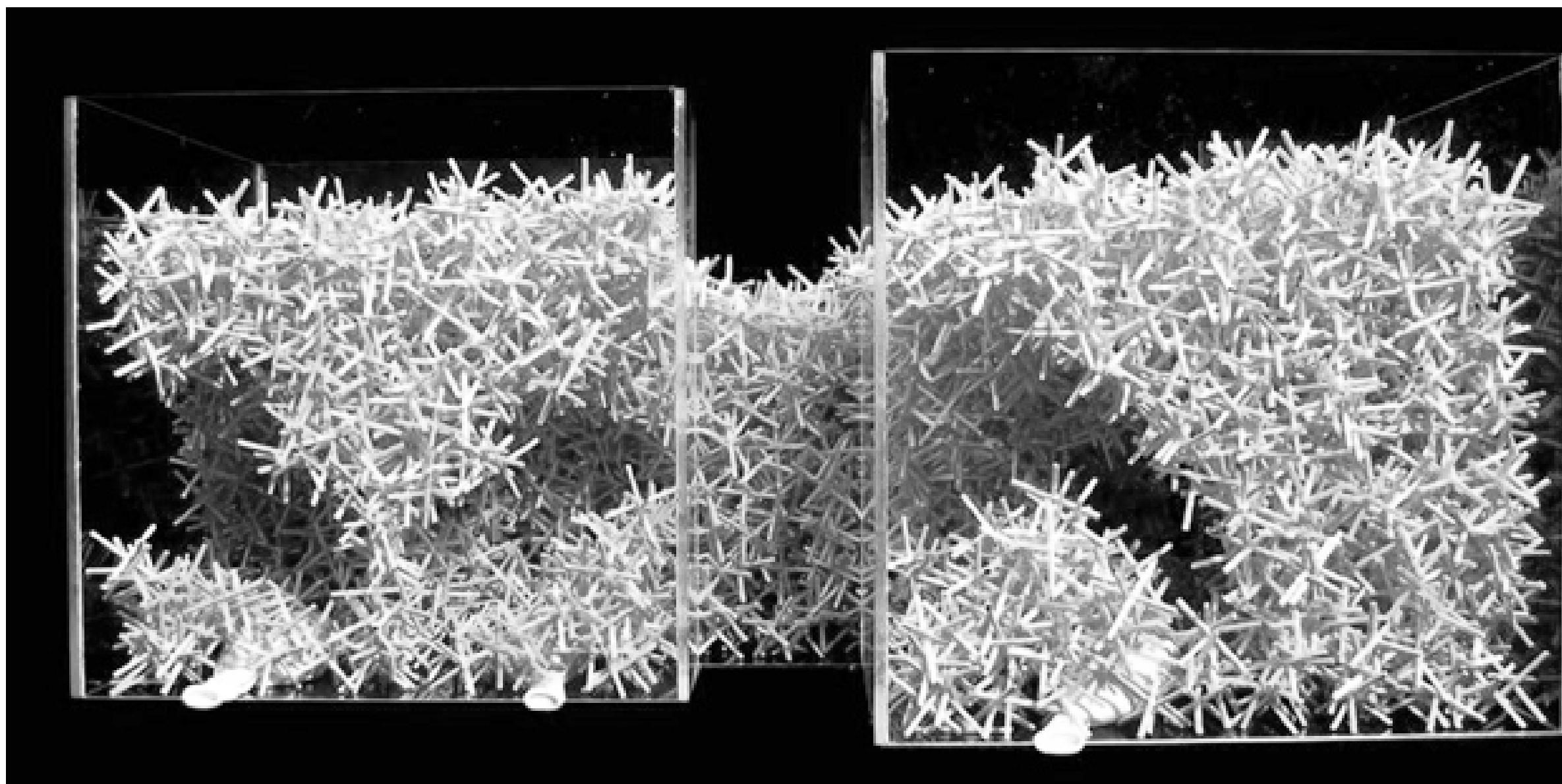
Le système n'a pas pour but de reproduire une forme à l'identique. L'utilisateur doit pouvoir donner une forme générale à construire mais ce sera l'intelligence artificielle du système qui définira la forme finale s'en approchant plus ou moins.



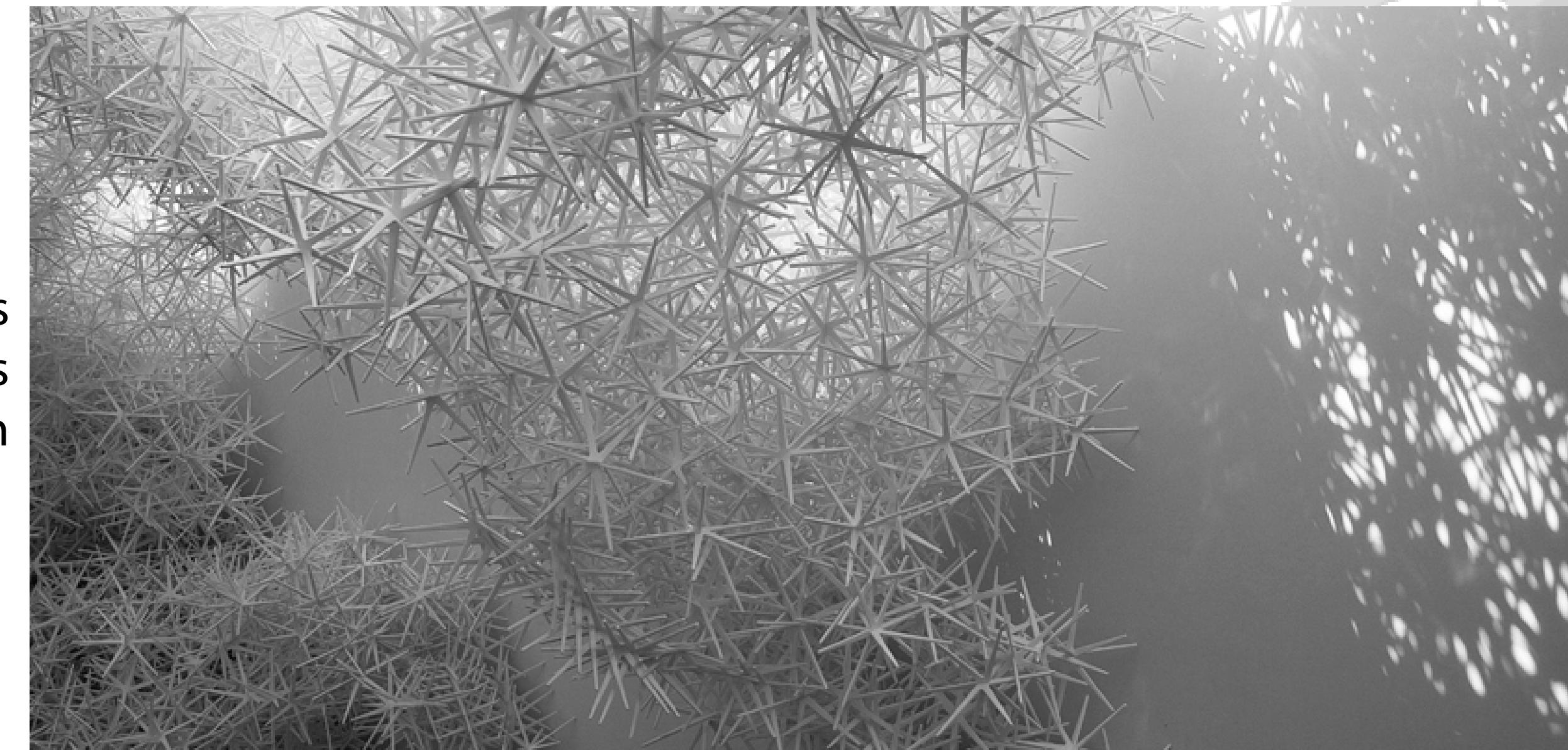
# INTRODUCTION

## Références

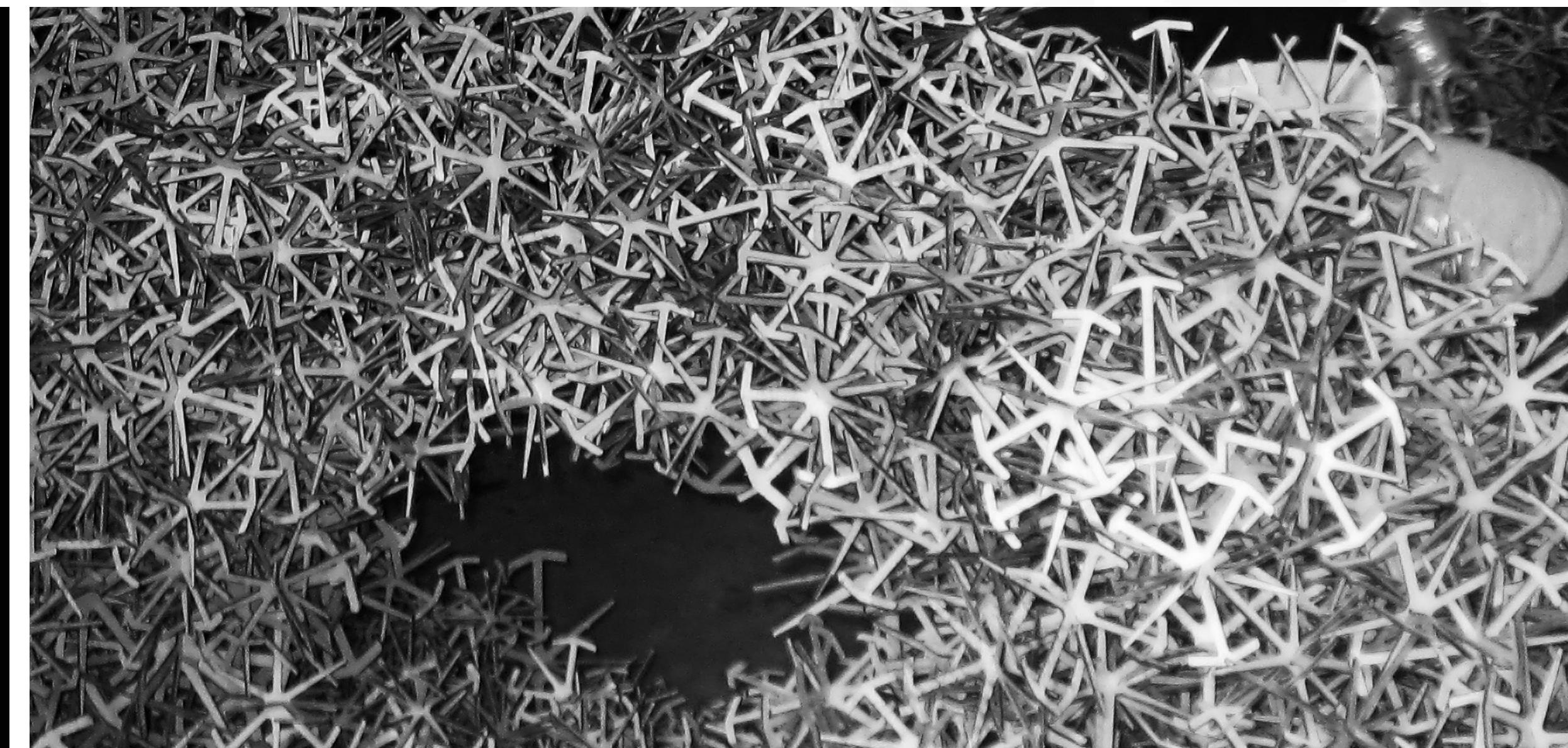
Plusieurs architectes ont travaillé sur l'agrégation et les formes émergentes à travers l'étude statistique et la simulation. Cependant ces réflexions n'ont jamais été couplées à une recherche sur l'automatisation de la mise en oeuvre.



MATSUDA Eichi, Designed Particles Aggregations 01, AA London (Menges and Hensel), 2004



DIERICH K., Aggregate architectures, Prof. A. Menges, 2010

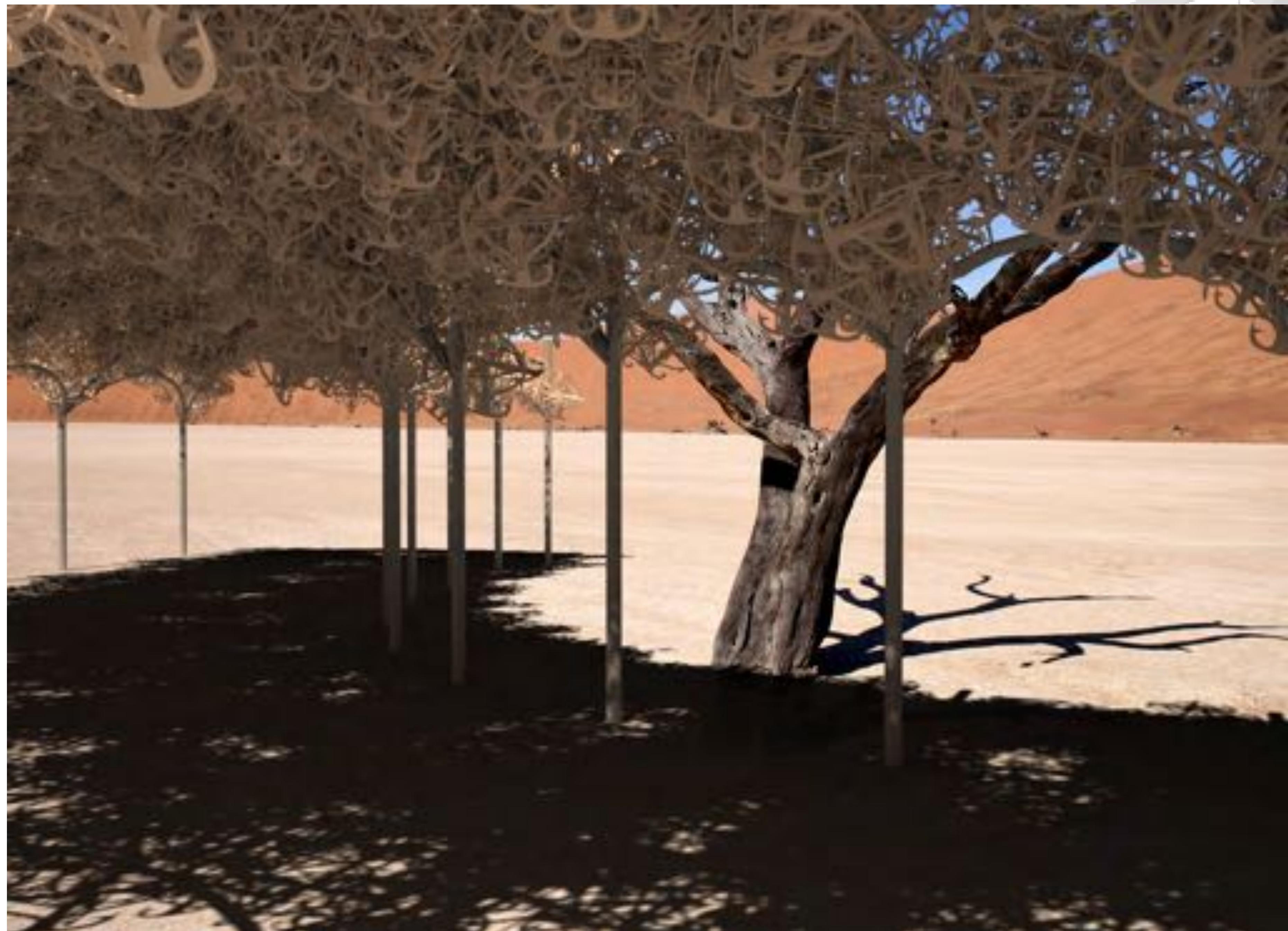


SPYROPOULOS Theodore, Imogen, 2014

# INTRODUCTION

## Prospective

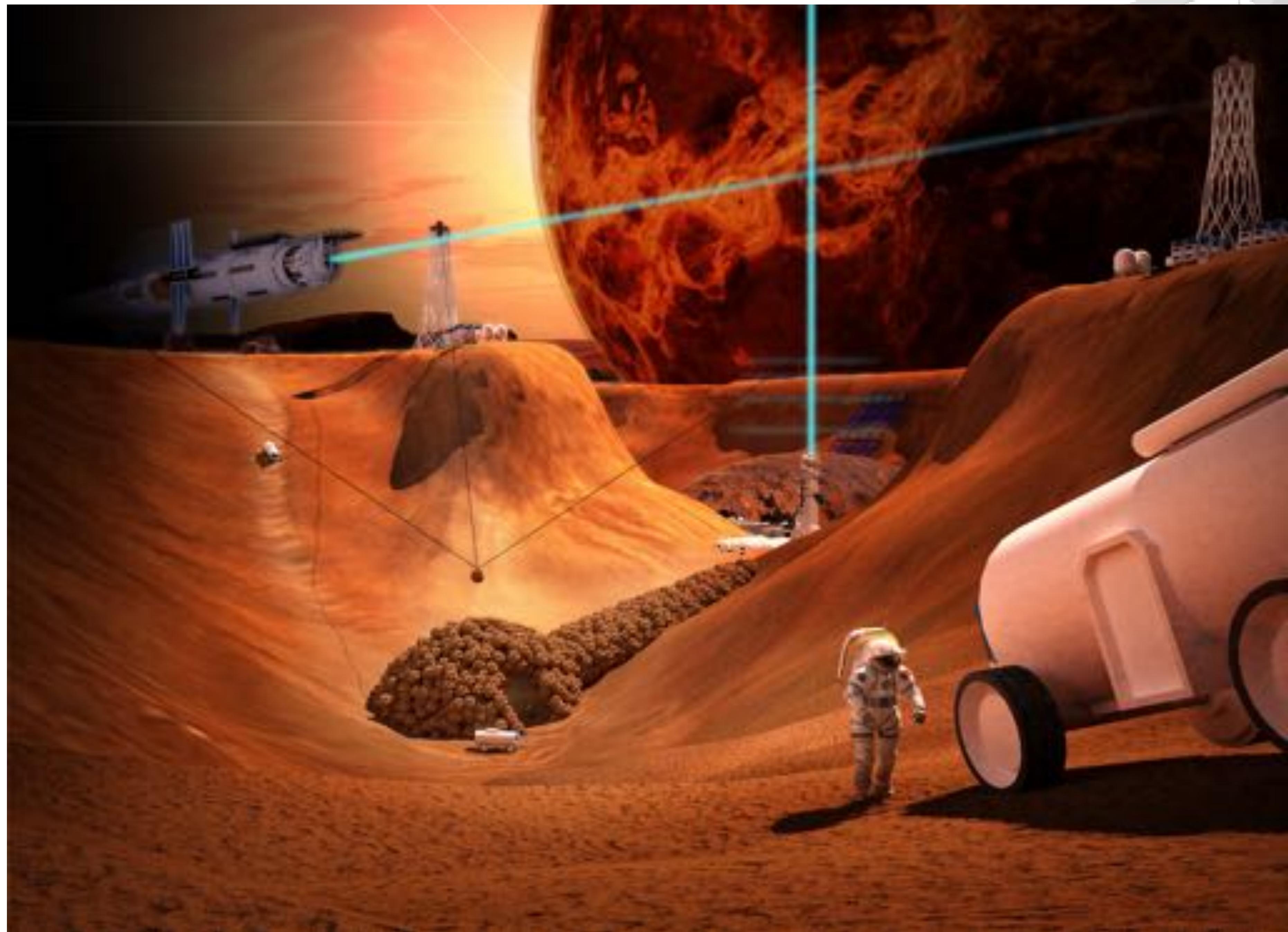
Une esthétique et un langage architectural.



# INTRODUCTION

## Prospective

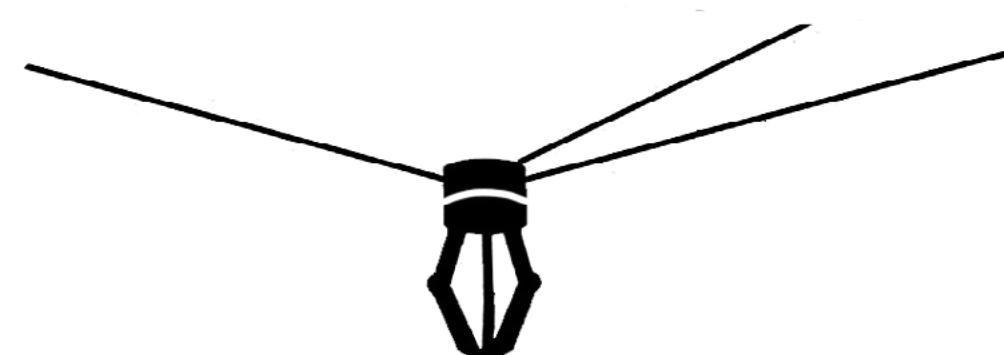
Un moyen de bâtir en conditions extrêmes.



# SYNERGIE

## 3 axes de recherche

Le projet de système de construction autonome peut être décomposé en 3 axes de recherche principaux :



### Agrégation

Définition d'un module constructif capable de former des structures par agrégation.



### Robotique

Conception d'un robot à câble permettant l'assemblage automatisé des modules sur de grandes portées.

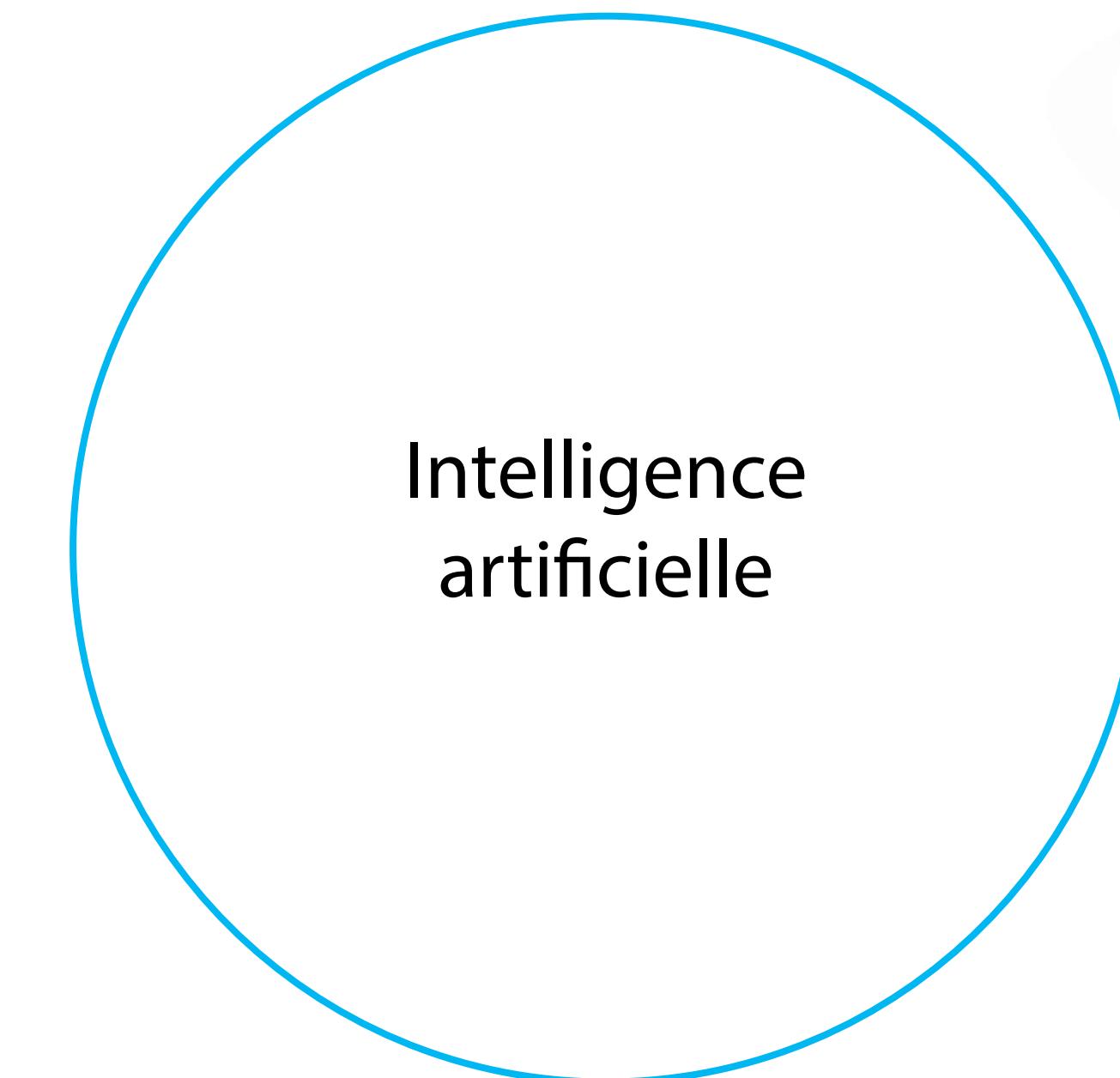
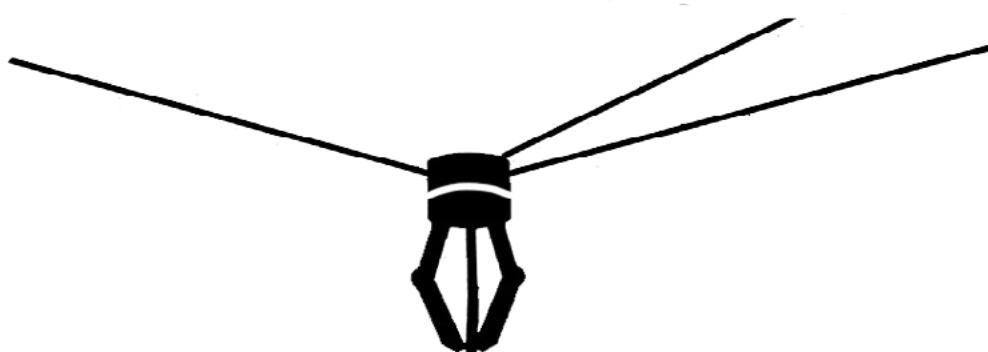
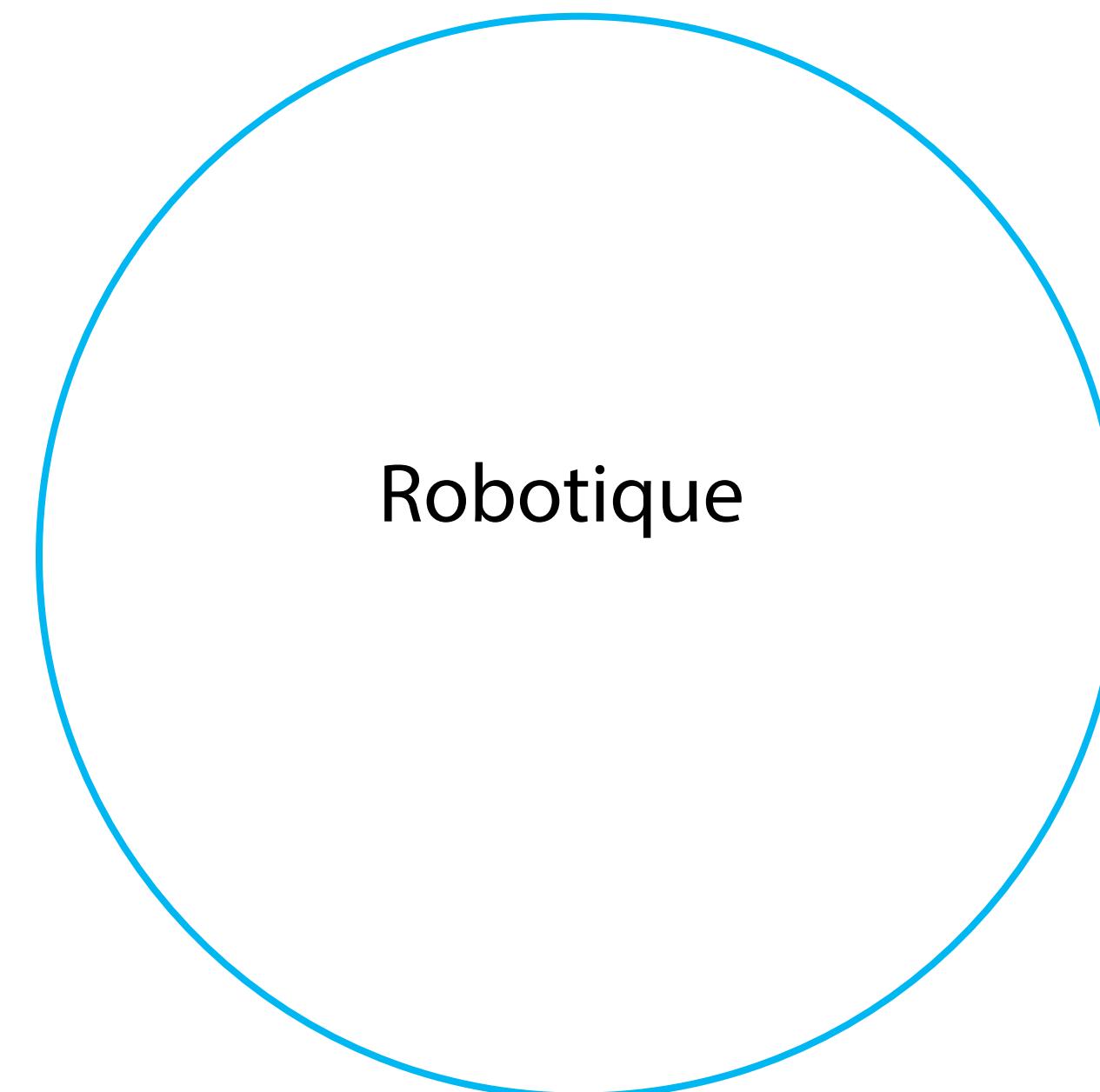
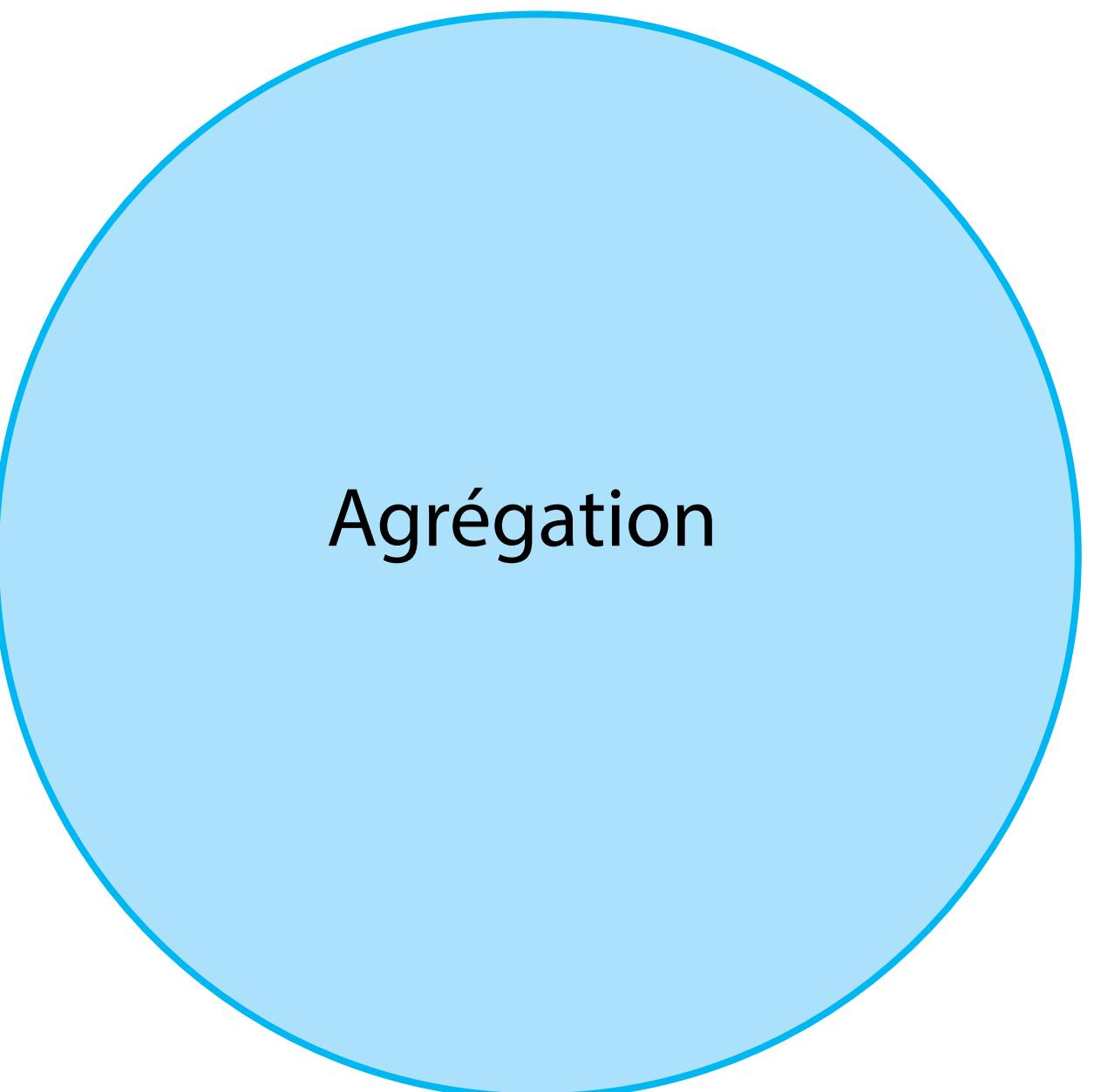


### Intelligence artificielle

Correction en temps réel des erreurs de construction par le biais de la vision machine et du principe d'asservissement.

# SYNERGIE

3 axes de recherche

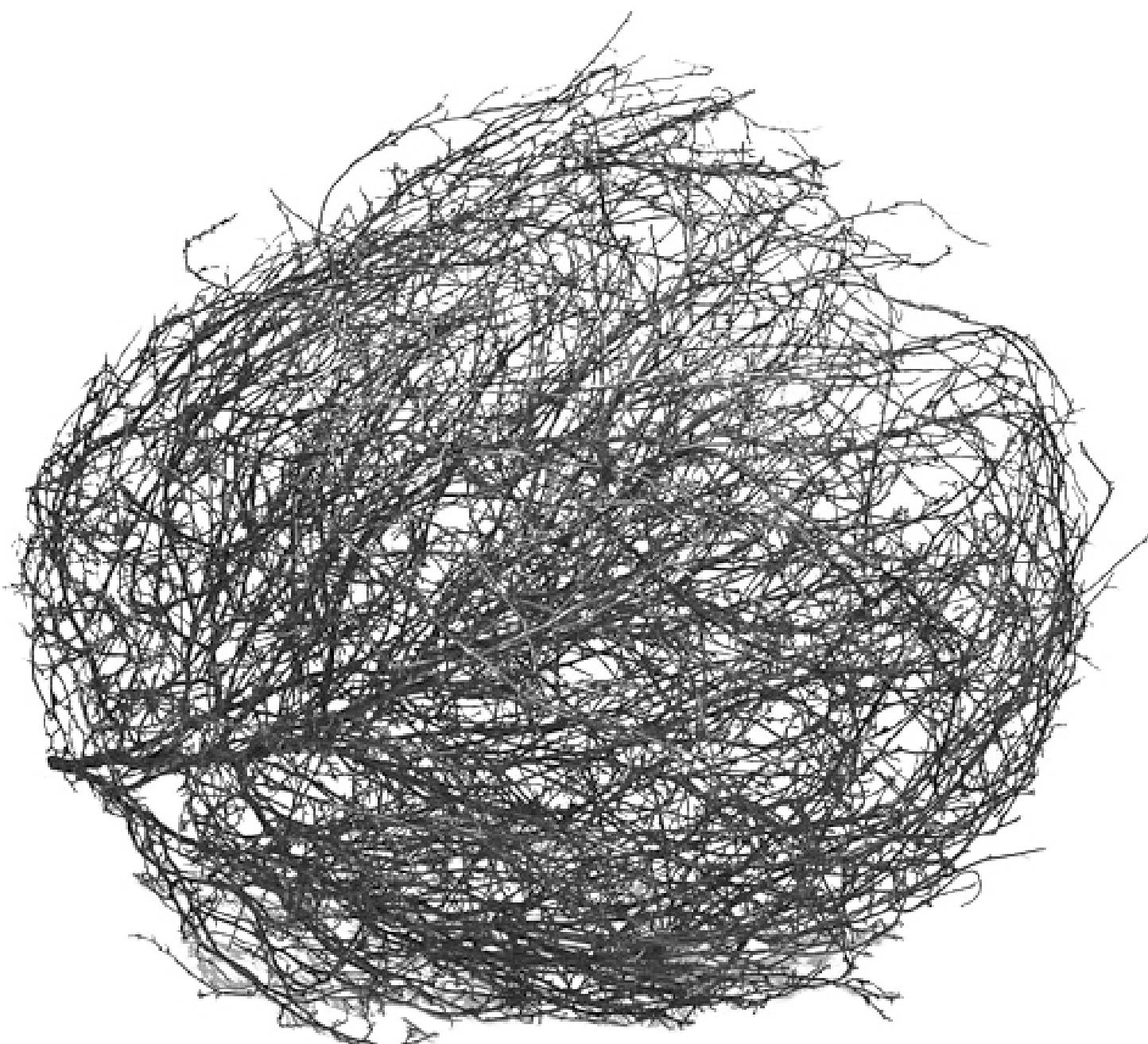


# AGGREGATION

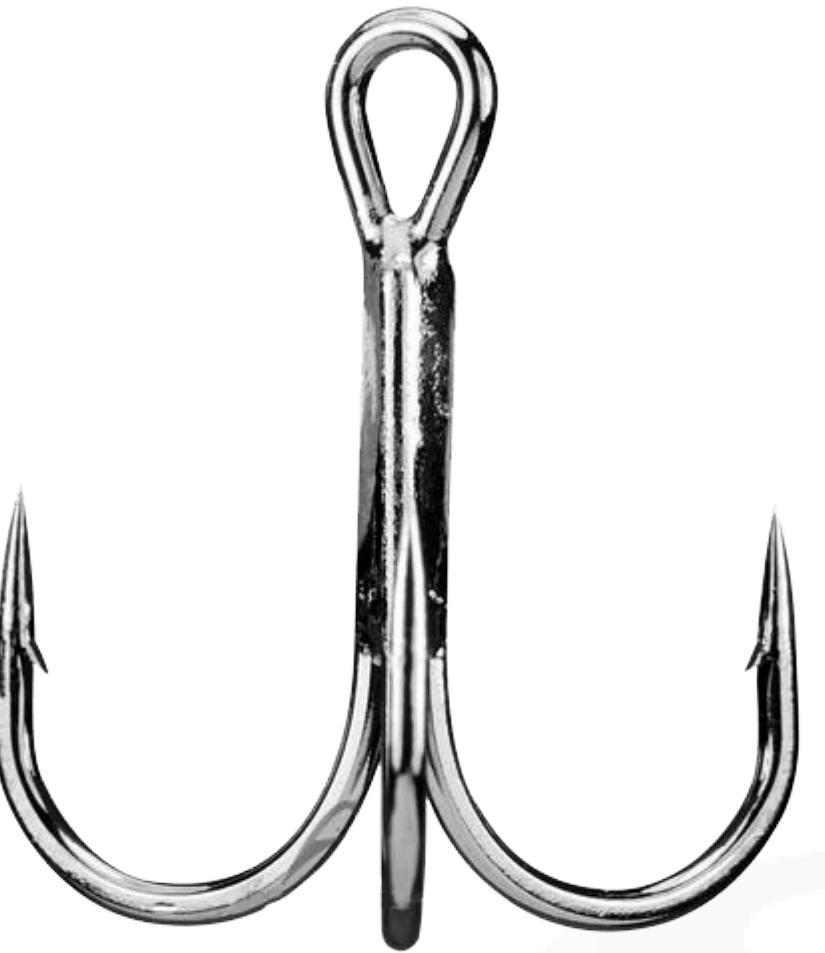
## Références

En observant à la fois des formes naturelles et des systèmes d'accroche, deux paramètres dominent :

- La récursivité de la structure
- La présence d'éléments de retenue de type crochets



Tumbleweed



Hameçon de pêche

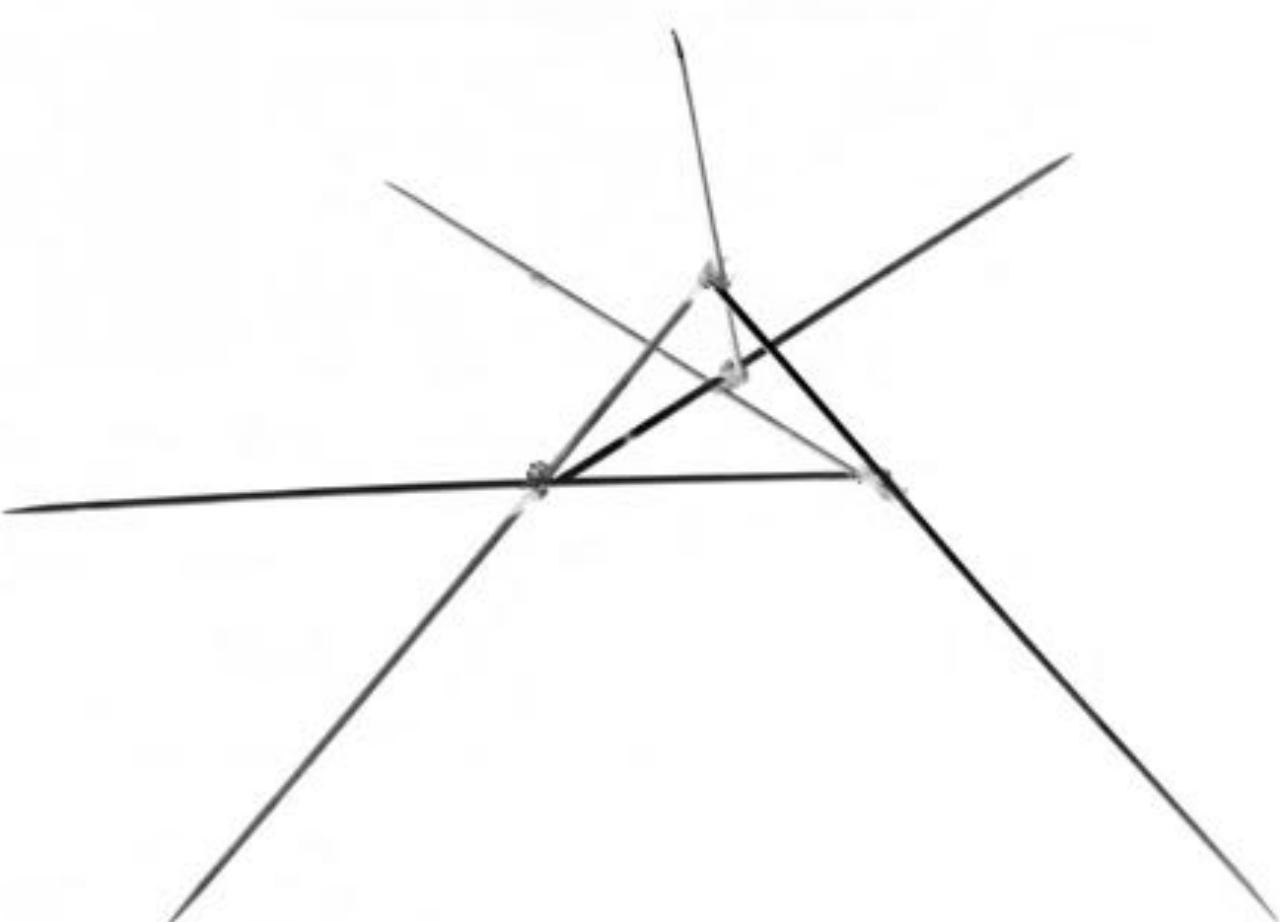
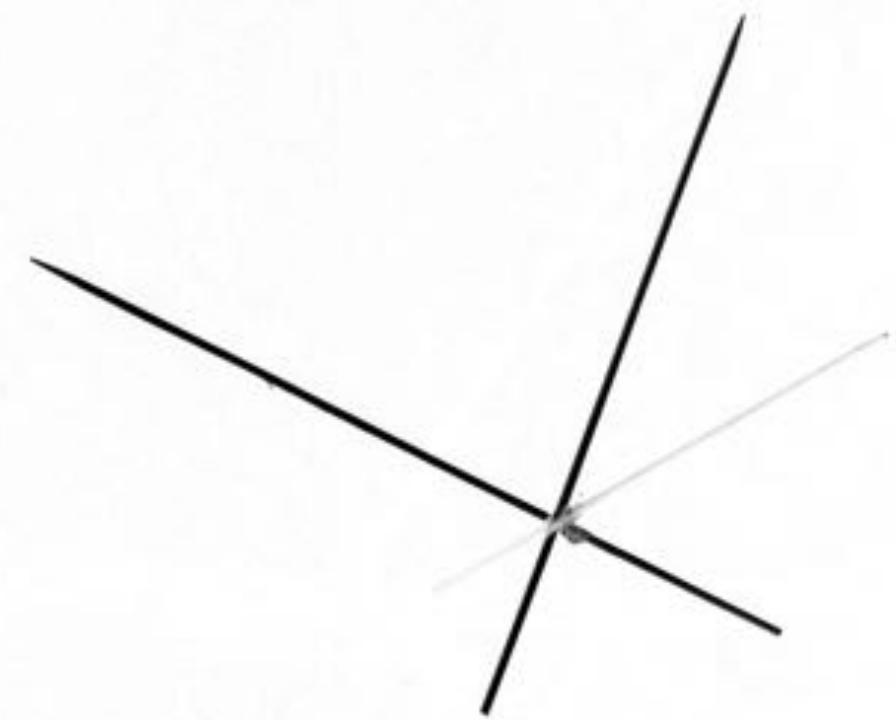


Vue agrandie du Velcro

# AGGREGATION

## Expérimentation

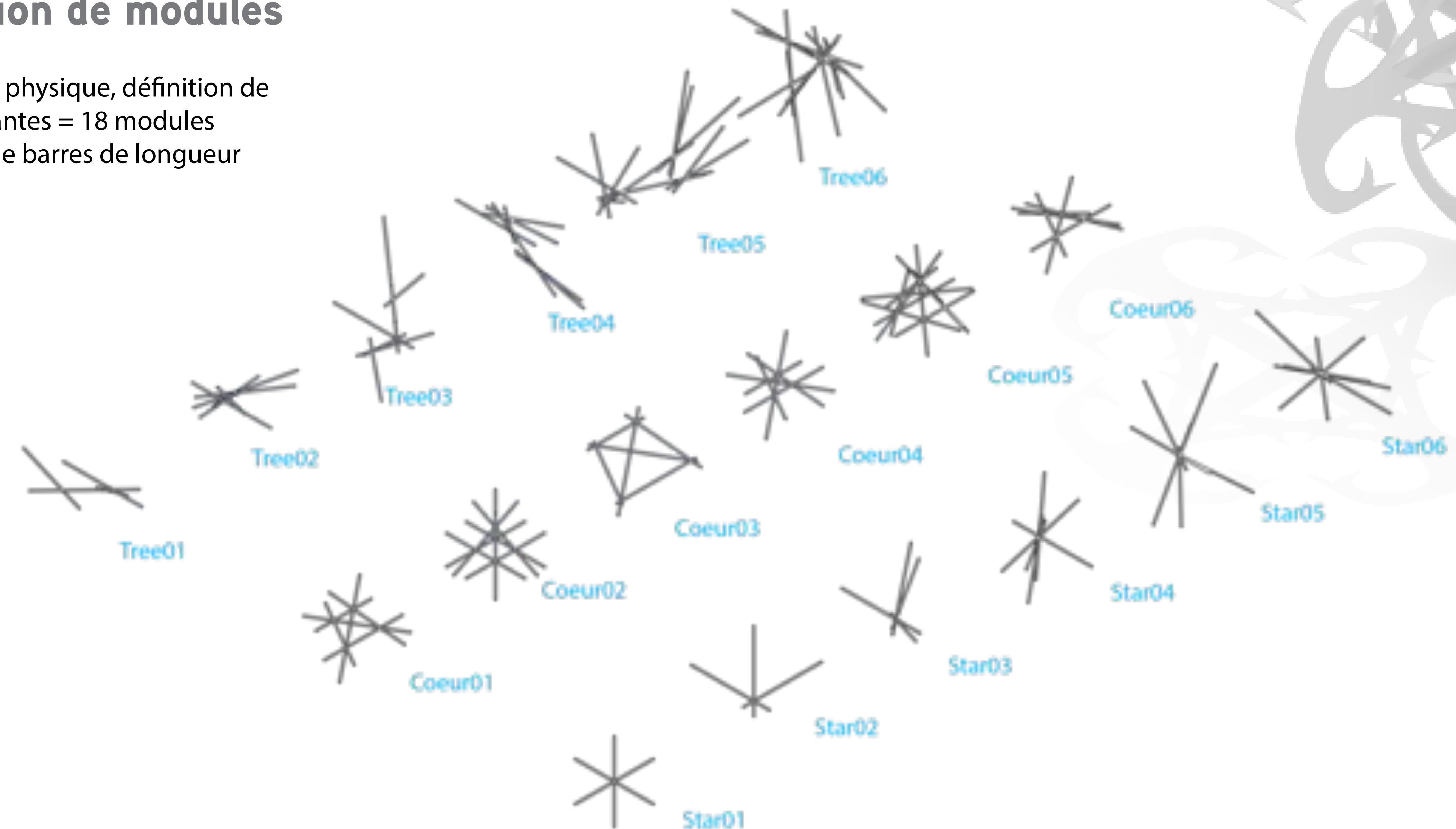
Recherche de géométries pour les modules à l'aide de barres de longueurs identiques.



# AGGREGATION

## 1ère génération de modules

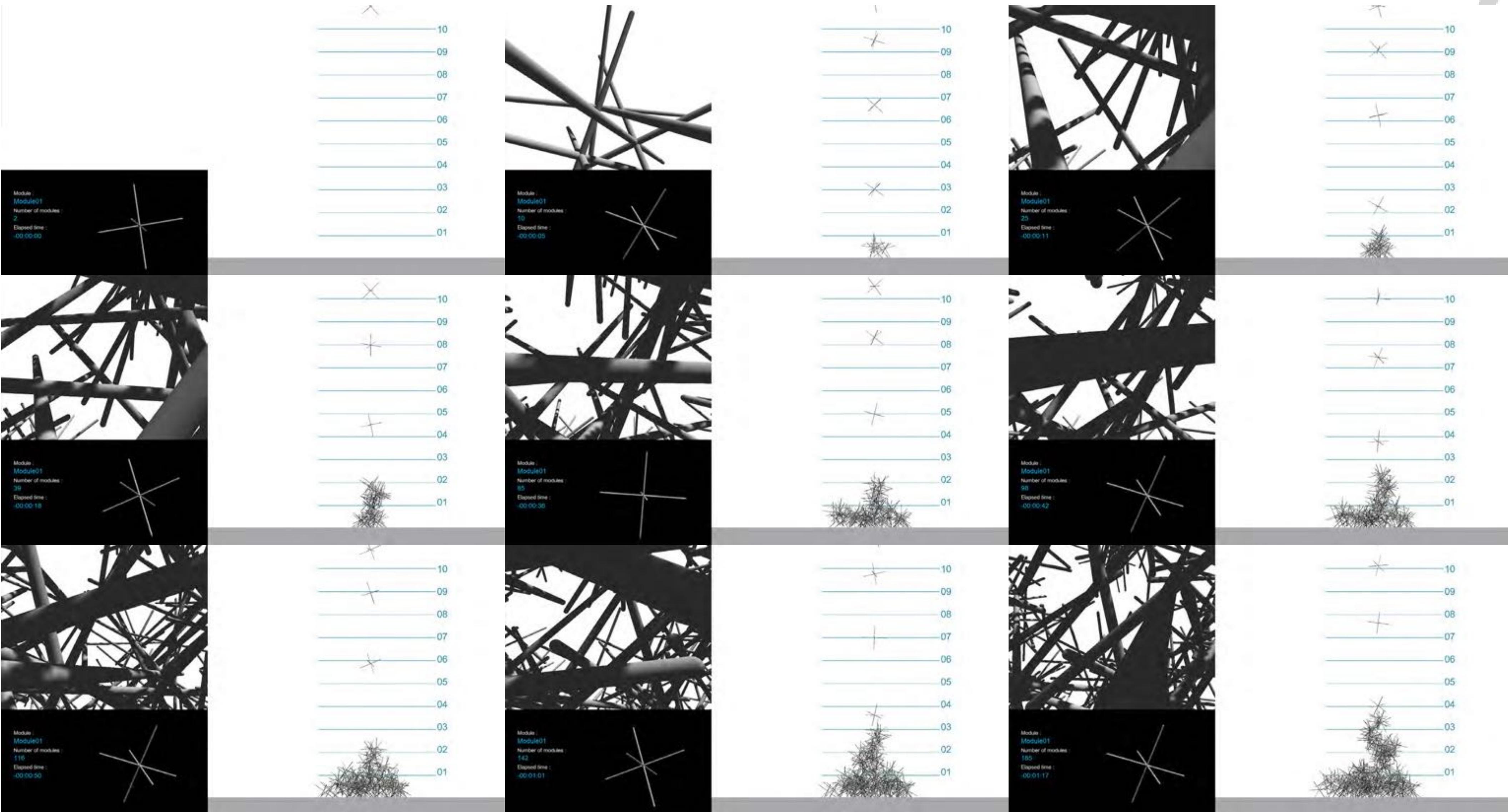
A partir des essais physique, définition de 3 typologies x 6 variantes = 18 modules toujours composés de barres de longueur identique.



# AGGREGATION

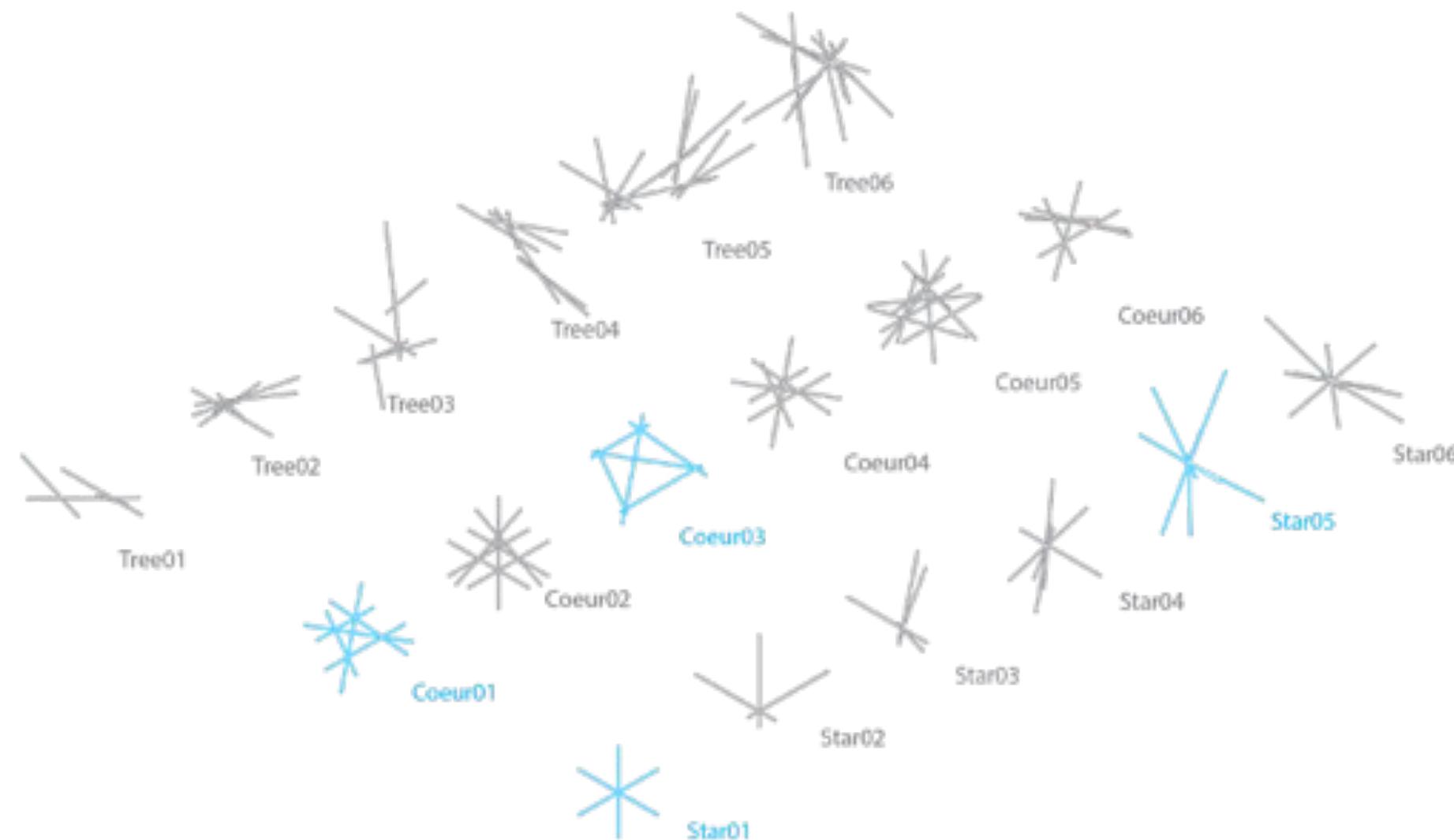
## Protocole d'expérimentation

- \* Simulation d'agrégation par chute sous Unity3D (10 tests/module)
- \* Export de données sous forme de fichier texte.
- \* Traitement statistique de celles-ci dans un tableur.



# AGREGATION

## 1ère série de résultats



Essai/Module	Star01	Star02	Star03	Star04	Star05	Star06	Coeur01	Coeur02	Coeur03	Coeur04	Coeur05	Coeur06	Tree01	Tree02	Tree03	Tree04	Tree05	Tree06
1	1.95	1.76	1.86	2.86	3.12	2.77	2.31	3.84	3.49	2.55	2.66	2.54	1.47	2.41	3.96	2.46	3.5	3
2	1.89	1.39	2	2.05	3.33	3.21	3.22	3.04	3.86	3.01	2.54	2.43	1.15	1.41	3.69	2.01	2.9	2.58
3	2.06	1.42	1.69	1.98	3.21	2.76	3.06	4.73	4.16	2.76	2.19	2.9	1.93	1.76	4.2	1.9	3.21	3.12
4	1.86	1.6	2.13	1.76	3.09	3.16	3.06	3.64	3.4	2.85	3.31	3.17	1.43	2.2	4.6	1.63	2.84	3.65
5	1.59	1.52	2.28	2.25	3.55	3.84	3.24	4.2	3.24	3.2	2.19	2.38	1.71	2.42	3.95	2.2	4.02	2.33
6	1.91	1.36	1.99	2.19	3.58	2.8	2.62	3.65	3.66	3.3	2.57	2.8	1.6	1.58	3.74	1.48	2.89	2.54
7	1.97	1.66	1.84	1.98	3.47	2.8	3.83	3.86	2.98	2.5	2.87	2.84	1.48	2.03	3.79	2.12	3.54	3.36
8	2.14	1.62	2.24	2.67	3.37	3.25	3.64	3.58	4.4	2.61	2.52	2.6	1.98	2.18	3.86	1.61	3.28	3.54
9	2.12	1.66	2.26	1.95	3.48	3.01	2.89	3.58	3.99	3.1	3.61	2.59	1.82	2.43	3.8	1.5	2.99	2.65
10	1.63	1.74	1.79	2.06	3.44	3.48	3.08	3.61	3.6	2.58	2.95	3.15	1.56	2.44	3.2	2.12	3	3.02
Moyenne	1.912	1.573	2.008	2.175	3.364	3.108	3.095	3.773	3.678	2.846	2.741	2.74	1.613	2.086	3.879	1.903	3.217	2.979
Valeur min	1.59	1.36	1.69	1.76	3.09	2.76	2.31	3.04	2.98	2.5	2.19	2.38	1.15	1.41	3.2	1.48	2.84	2.33
Valeur max	2.14	1.76	2.28	2.86	3.58	3.84	3.83	4.73	4.4	3.3	3.61	3.17	1.98	2.44	4.6	2.46	4.02	3.65
Écart	0.55	0.4	0.59	1.1	0.49	1.08	1.52	1.69	1.42	0.8	1.42	0.79	0.83	1.03	1.4	0.98	1.18	1.32
Balances	3	3	4	5	6	6	8	6	6	11	6	4	7	7	7	13	13	
Moy/balancé	0.64	0.52	0.50	0.44	0.56	0.52	0.52	0.47	0.61	0.47	0.25	0.46	0.40	0.30	0.55	0.27	0.25	0.23
Moy Type	0.528956						0.46344						0.333303					

# AGGREGATION

## 2ème série de résultats

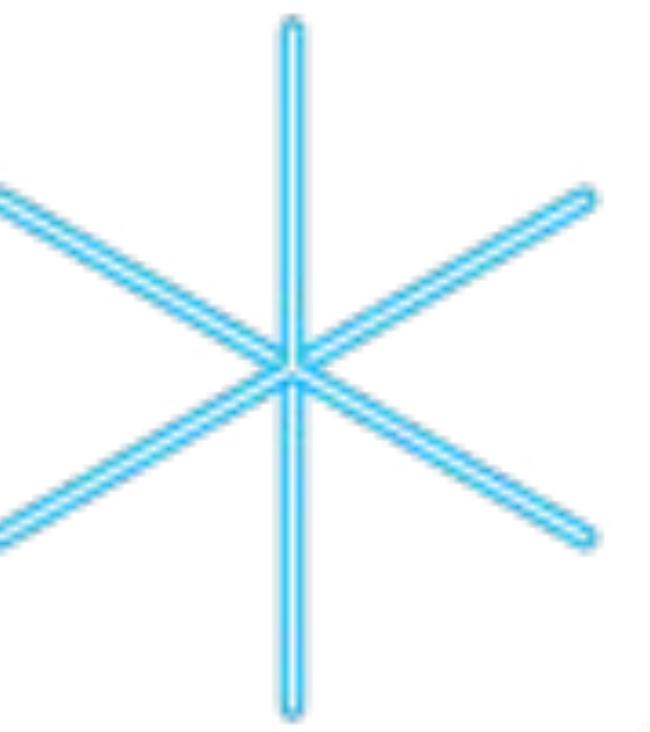
Même protocole d'expérimentation répété 100 fois pour chacun des modules.

TechModule	Star01	Star05	Coeur01	Coeur03
1	1.00	2.00	2.00	2.79
2	2.20	3.48	2.90	3.87
3	2.25	3.50	2.96	3.84
4	1.87	2.44	2.51	4.23
5	2.00	2.86	2.73	3.64
6	2.04	3.37	3.81	3.00
7	1.88	2.88	2.89	3.88
8	1.89	3.24	3.26	3.88
9	2.27	2.87	2.86	3.52
10	1.90	3.45	3.60	4.85
11	1.72	2.95	3.01	3.64
12	1.68	3.21	2.78	4.30
13	1.80	3.49	2.99	3.98
14	2.31	3.48	3.38	4.18
15	1.85	3.40	3.32	3.83
16	2.25	3.59	3.90	3.01
17	2.23	3.39	3.63	3.62
18	2.00	2.88	2.66	3.89
19	2.07	3.63	3.24	3.88
20	1.84	2.88	3.27	3.53
21	2.06	2.79	3.26	3.38
22	2.42	3.29	2.76	4.17
23	1.97	3.25	3.14	3.05
24	2.18	3.37	4.41	3.13
25	1.85	3.34	2.79	3.85
26	1.84	3.39	3.07	3.64
27	2.17	3.38	3.11	4.06
28	2.14	3.26	2.62	3.28
29	2.22	3.18	3.21	3.72
30	1.95	3.14	2.70	3.50
31	2.08	2.89	3.12	4.05
32	2.19	3.09	3.24	3.50
33	1.99	3.15	3.06	3.53
34	2.07	3.60	2.91	3.62
35	2.04	3.27	3.78	3.89
36	2.28	3.10	2.86	3.68
37	1.92	3.63	2.86	3.45
38	2.04	3.29	3.06	4.00
39	2.02	3.02	3.01	3.69
40	2.39	3.62	3.91	3.66
41	2.00	3.40	2.72	3.72
42	2.15	3.70	3.46	3.69
43	1.94	3.41	3.72	2.85
44	2.20	3.08	3.18	3.72
45	2.49	3.23	3.19	3.79
46	2.25	2.88	2.53	3.50
47	1.94	2.89	2.58	3.79
48	2.00	3.09	3.37	3.75
49	2.04	2.89	2.46	4.20
50	1.84	2.44	2.89	3.42
51	2.18	2.96	3.41	3.21
52	2.33	3.01	3.49	4.06
53	2.22	3.03	3.08	3.78
54	2.00	3.03	3.66	3.68
55	2.40	2.89	2.89	3.89
56	2.39	2.89	2.95	3.64
57	2.49	3.43	3.11	3.61
58	2.55	2.51	2.66	3.40
59	1.80	3.08	3.28	3.08
60	1.93	3.48	3.05	3.29
61	1.97	2.70	2.26	4.25
62	2.04	3.36	2.98	3.17
63	2.48	3.49	3.49	3.69
64	2.54	3.15	3.30	3.34
65	2.25	2.85	2.64	3.69
66	2.36	2.80	3.26	3.88
67	1.94	2.88	2.94	3.08
68	1.95	3.32	3.18	3.45
69	2.00	2.89	2.89	3.10
70	1.87	3.15	3.06	3.00
71	2.06	3.05	3.41	3.80
72	2.49	3.50	3.56	3.80
73	2.18	2.82	2.42	4.28
74	2.11	3.29	3.12	2.79
75	2.27	2.97	2.86	3.17
76	2.06	3.43	2.96	3.81
77	1.96	2.95	3.36	3.49
78	2.38	3.42	3.37	3.47
79	1.92	2.12	2.14	2.91
80	2.29	3.08	2.63	3.96
81	1.85	2.98	3.09	3.30
82	2.47	2.49	2.80	3.80
83	2.00	3.23	3.11	3.95
84	2.18	3.25	2.64	3.81
85	2.35	3.63	2.79	3.04
86	2.7	3.12	3.64	3.61
87	2.00	3.44	3.46	3.27
88	2.18	3.44	3.34	3.27
89	2.17	3.81	3.00	3.71
90	2.17	3.18	4.06	3.27
91	1.75	2.85	2.56	3.93
92	2.00	2.74	3.25	3.40
93	2.39	3.23	2.89	4.38
94	2.39	3.80	3.39	3.51
95	2.28	3.80	3.39	3.94
96	2.15	3.89	2.51	3.29
97	2.02	2.41	3.25	4.28
98	2.45	3.83	2.36	3.95
99	1.85	3.84	3.04	3.09
100	2.00	3.85	3.56	3.62

Min  
Max  
Ecart  
Moyenne  
Barres  
Moy/barre

	Star01	Star05	Coeur01	Coeur03
Min	1.65	2.41	2.02	2.34
Max	2.55	4.20	4.41	4.65
Ecart	0.90	1.79	2.40	2.31
Moyenne	2.10	3.20	3.03	3.45
Barres	3.00	6	6	6
Moy/barre	0.70	0.53	0.50	0.58

	Star01	Star05	Coeur01	Coeur03
Min	2.42	2.01	2.38	2.38
Max	4.20	4.41	4.65	4.65
Ecart	2.79	2.40	2.33	2.33
Moyenne	3.28	3.63	3.65	3.65
Barres	6	6	6	6
Moy/barre	0.83	0.60	0.58	0.58



Star01  
Module le plus efficace



Coeur03  
2ème module le plus efficace

# AGGREGATION

## Essais de fabrication



Star01



Coeur03

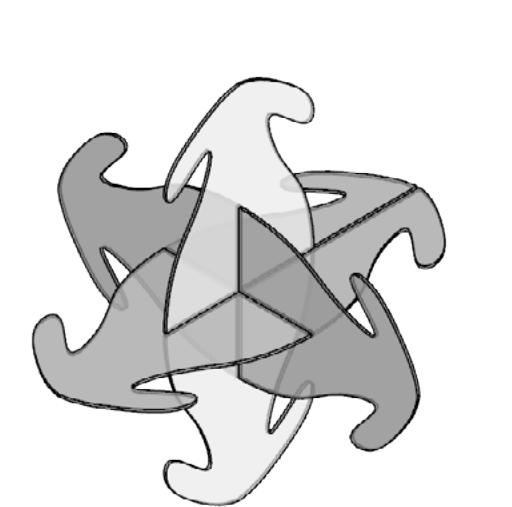
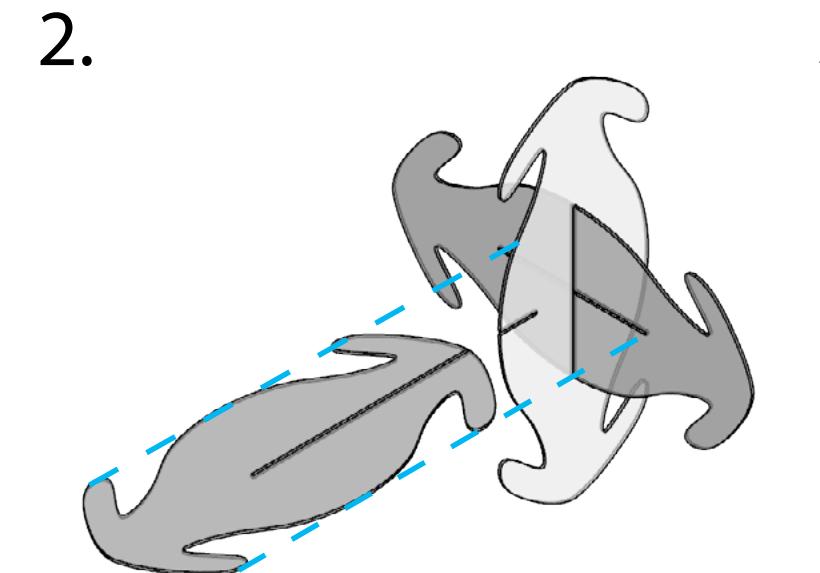
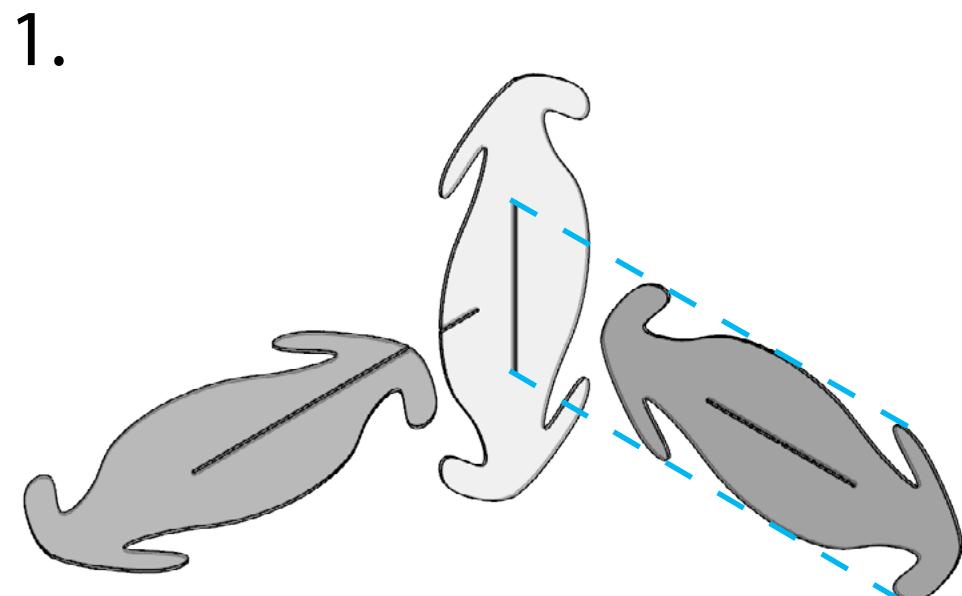
- \* Fabrication longue et fastidieuse
- \* Pouvoir de cohésion faible (pas de réel accrochage)
- \* Géométrie rapidement limitée en compléxité

# AGGREGATION

## 2ème génération de modules

Choix d'un mode de production plus adapté : découpe laser et assemblage par encoches. Le mode de découpe en plan nous permet une plus grande liberté dans la géométrie des modules.

3 densités de modules suivant la forme des 2 meilleurs modules de la génération une.



Assemblage des modules découpés à la laser

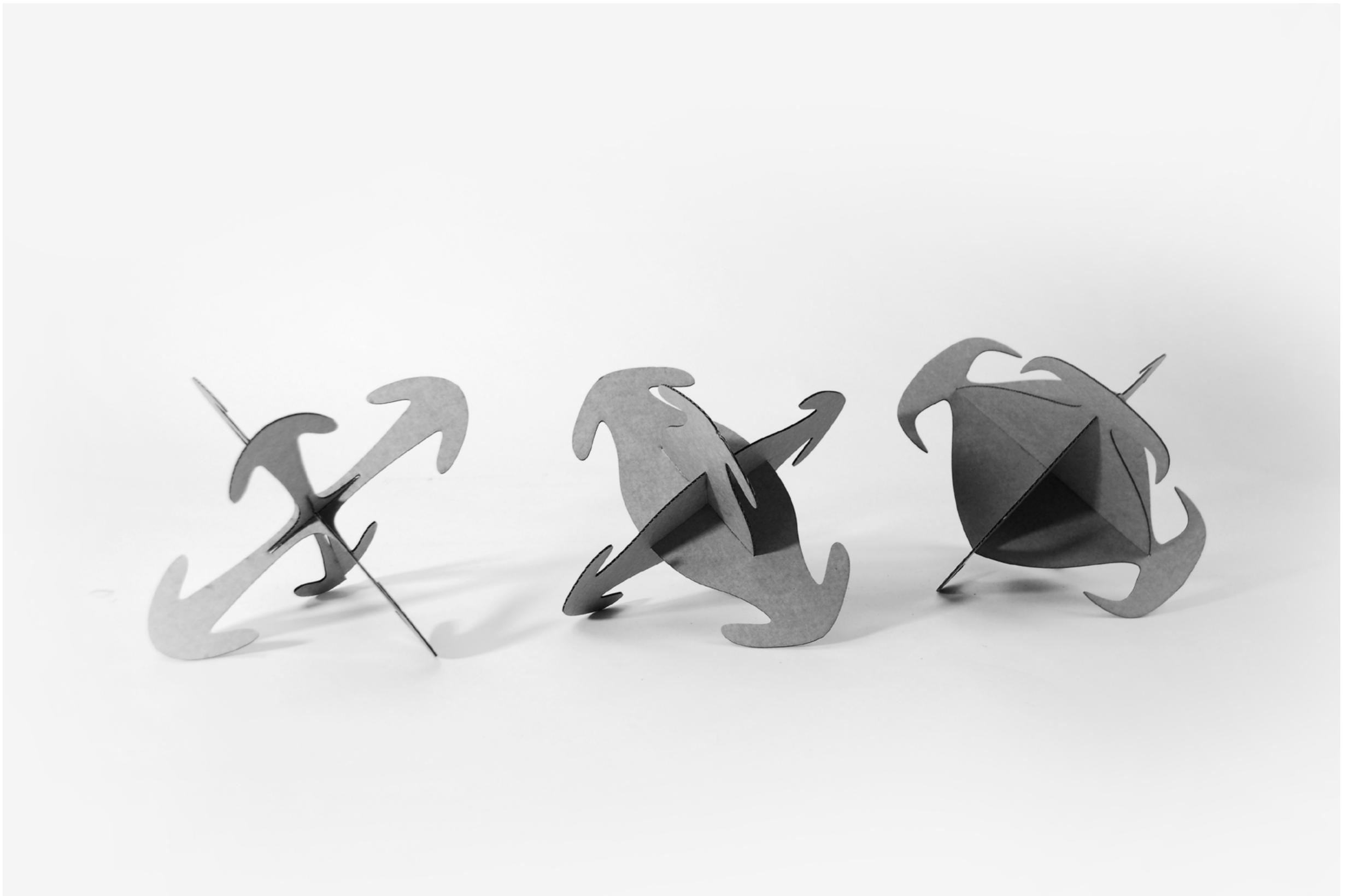


Géométrie héritant des tests précédents



# AGGREGATION

## Essais de fabrication



- \* Assemblage ais  
- \* D  coupe rapide
- \* Variations possibles dans la densit   des structures
- \* Agr  gation efficace mais qui peut  tre am  lior  e

# AGGREGATION

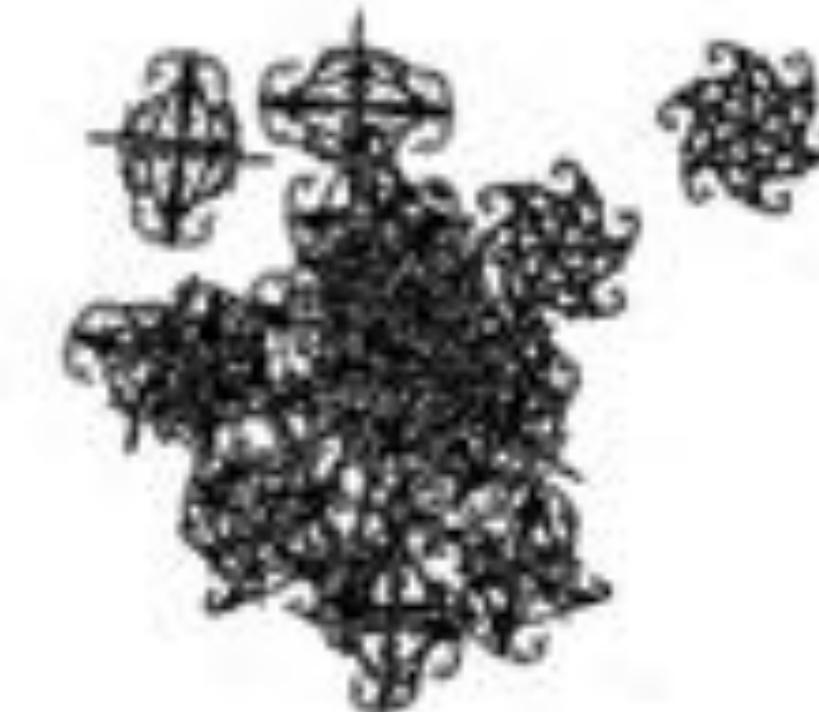
## 3ème génération de modules

20 Modules aggregation

Type



Top view



Front view



Ajout de crochets et compléxification de la géométrie pour optimiser la cohésion entre les modules.

# AGREGATION

## Variations de densité

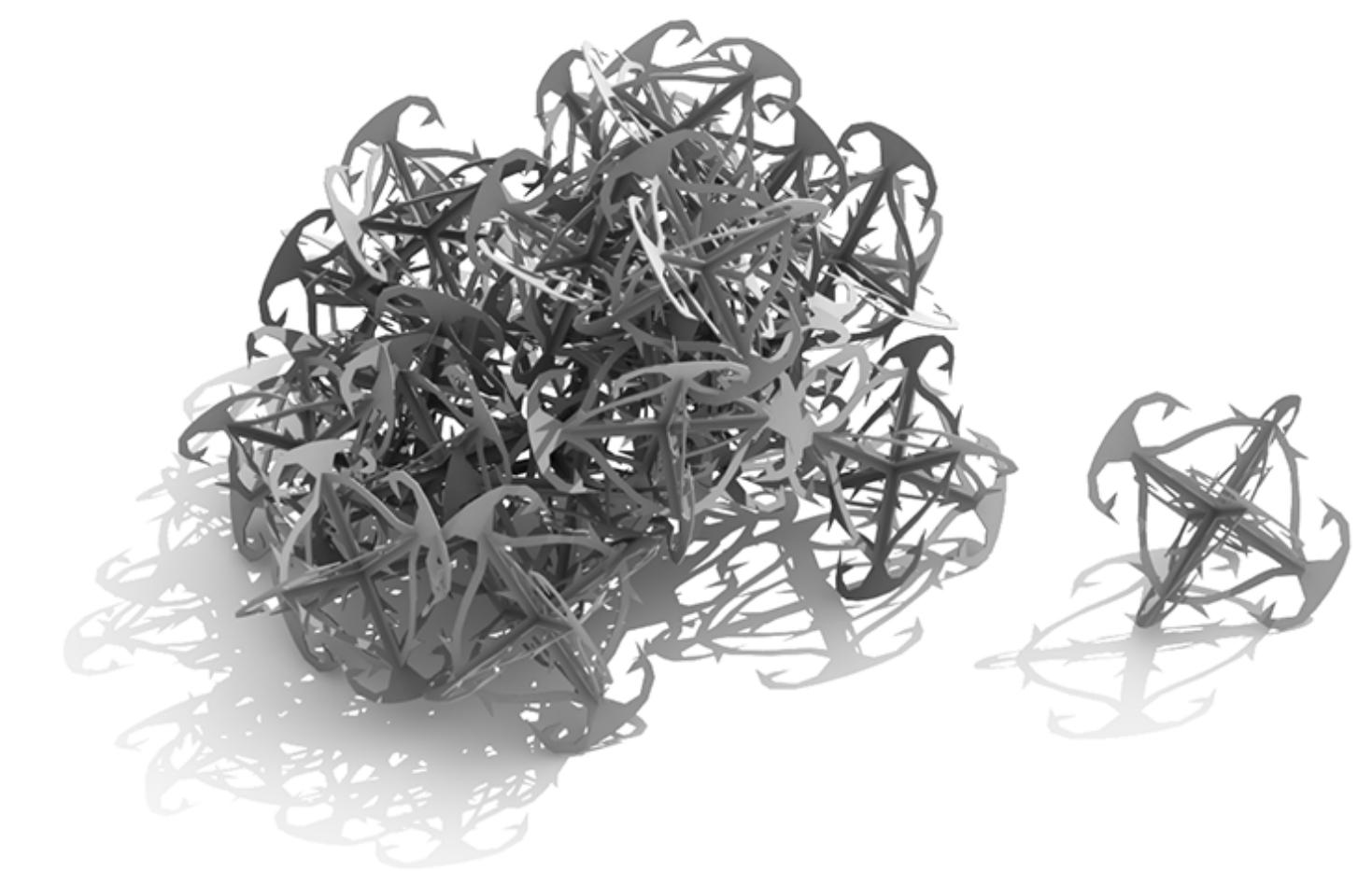
Module01Hooked



Module02Hooked



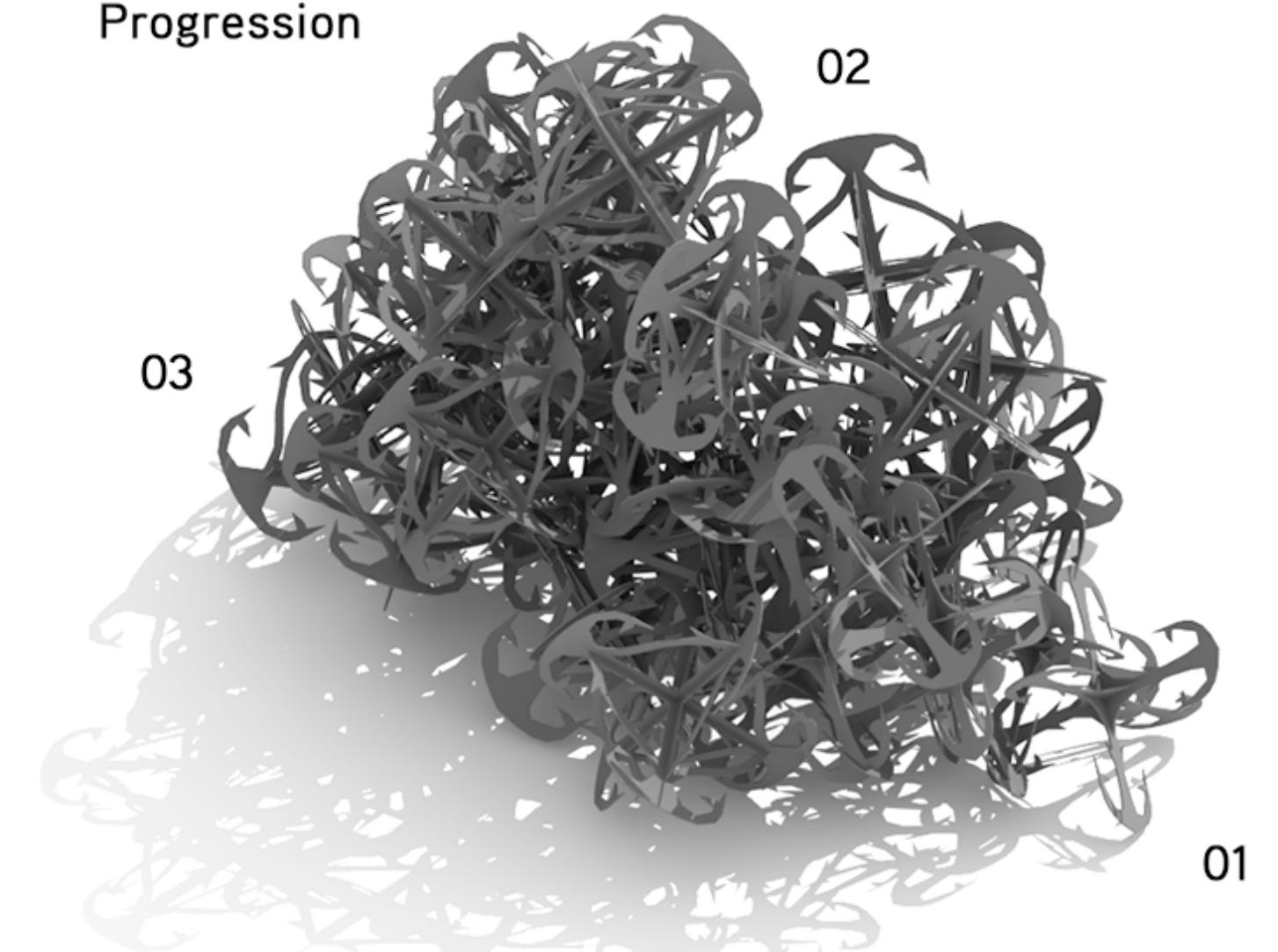
Module03Hooked



ModuleMix

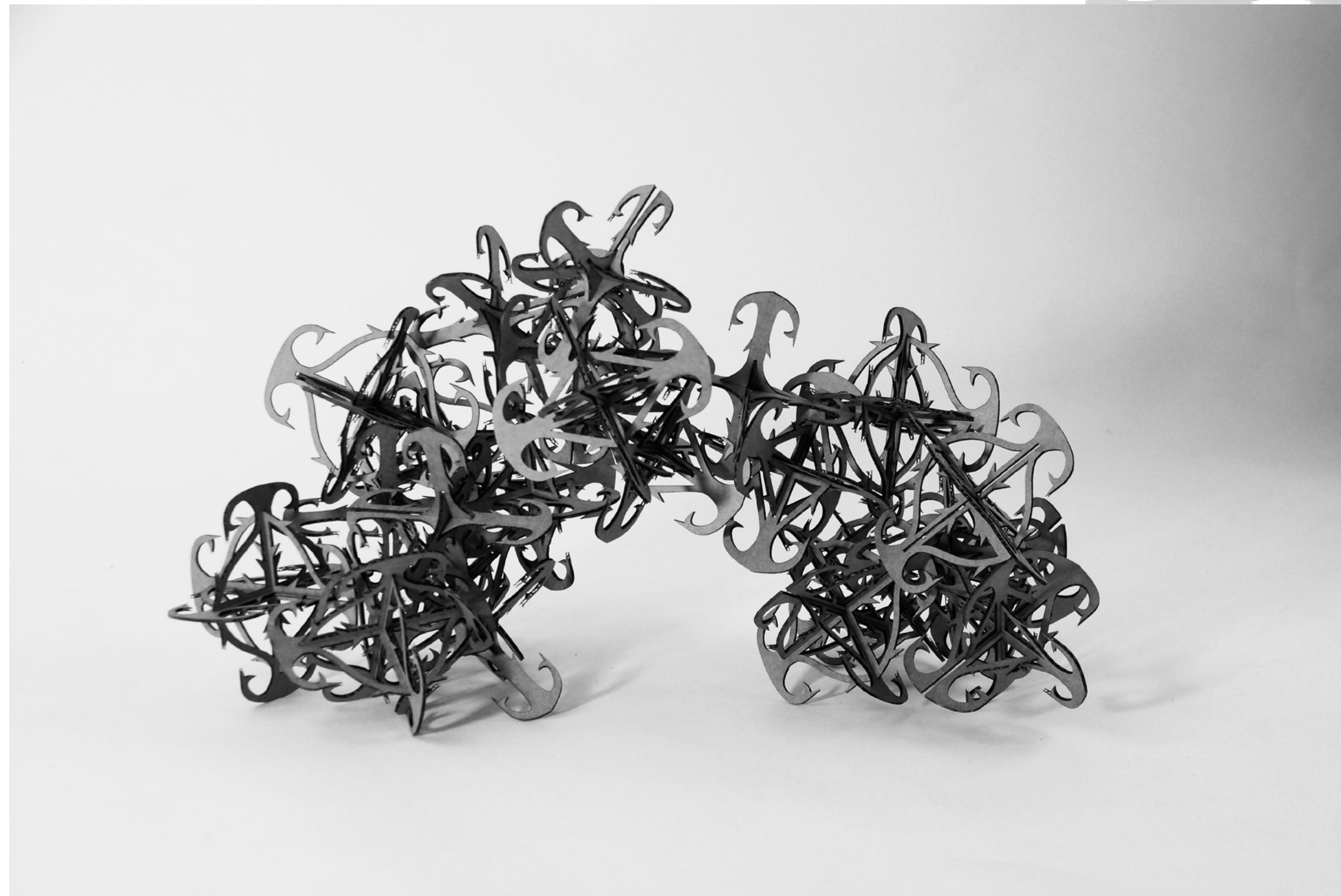
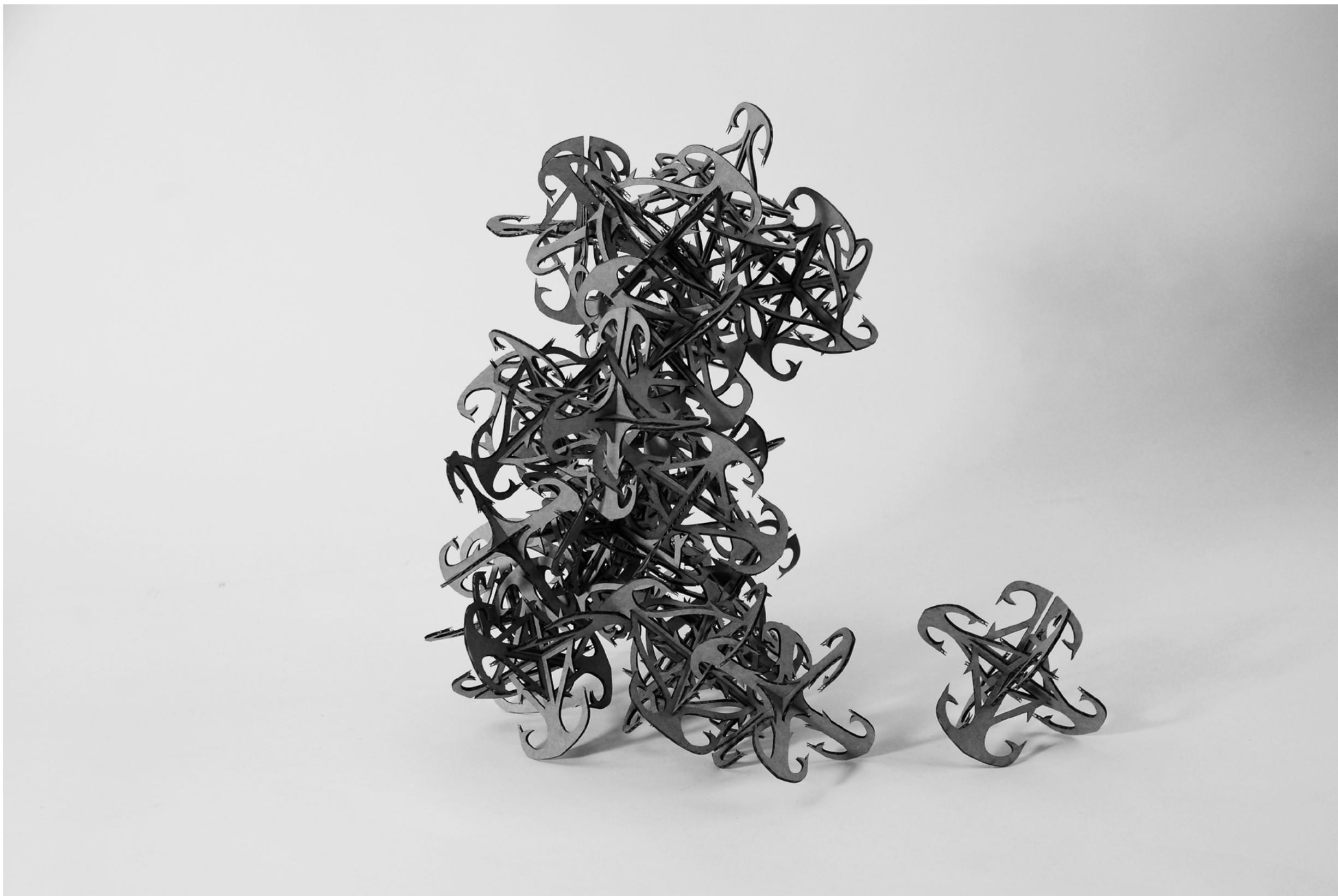


Progression



# AGREGATION

## Essais de fabrication



\* Conserve les mêmes qualités que la génération précédente

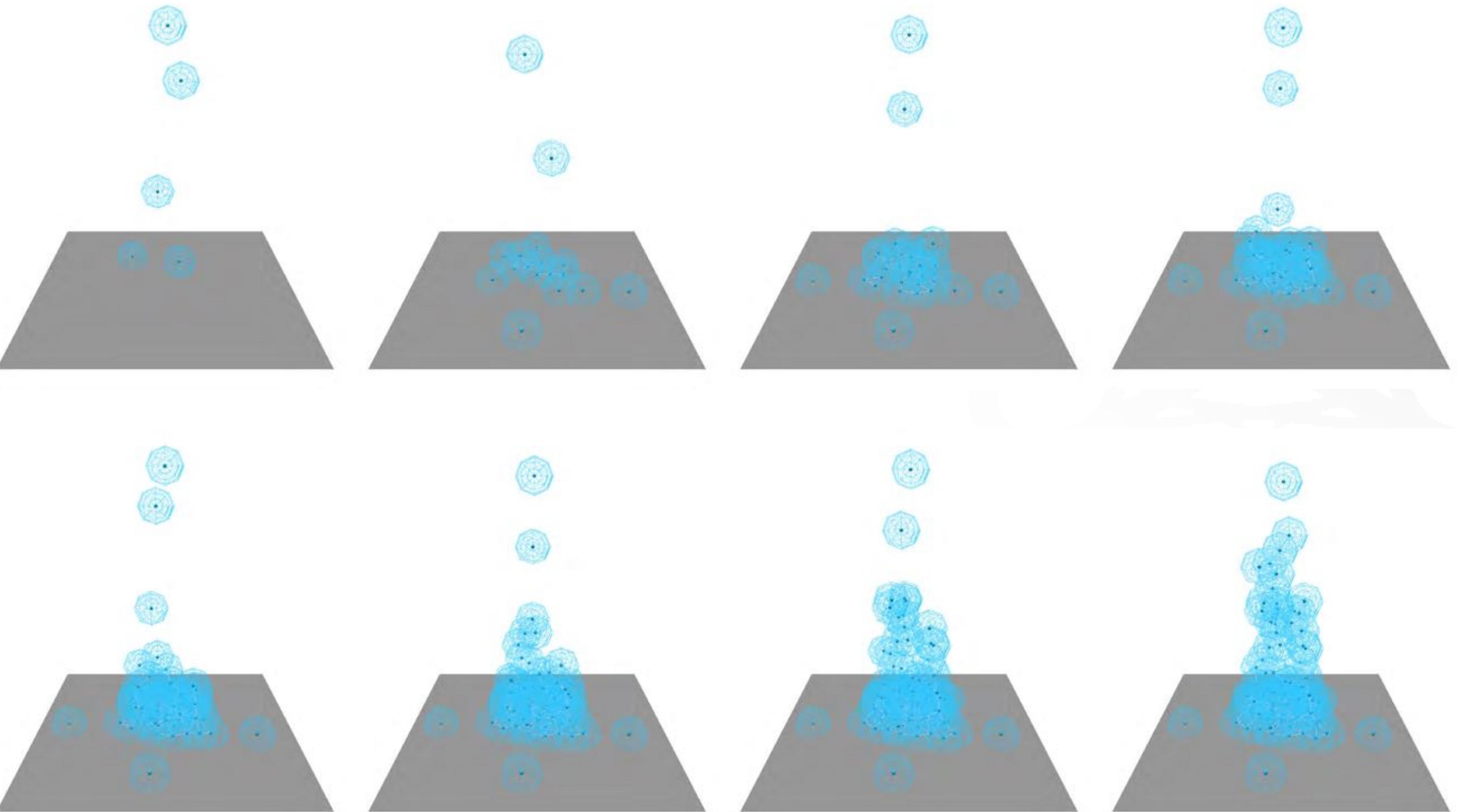
\* Agrégation largement améliorée

# AGGREGATION

## Simulation à plus grande échelle

La physique de Unity3D demandant beaucoup de puissance de calcul (collisions de mesh), les simulations ne peuvent aller au delà d'une cinquantaine de modules.

Une nouvelle simulation sous Processing utilisant un SMA (système multi-agents) dont le comportement de ces derniers simule au mieux la physique de cohésion des module nous permet d'approximer la construction de structures composées d'un nombre important de modules.



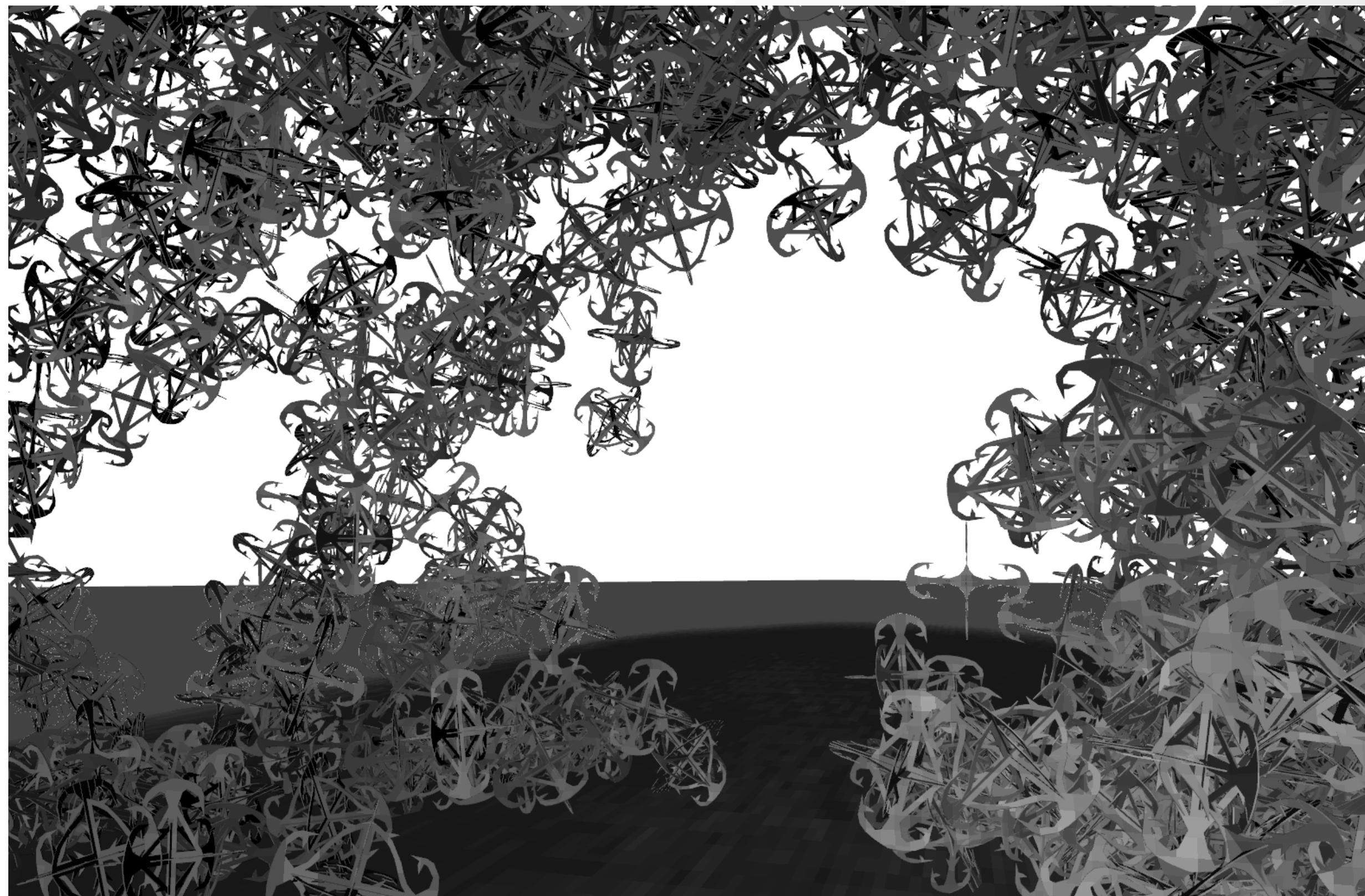
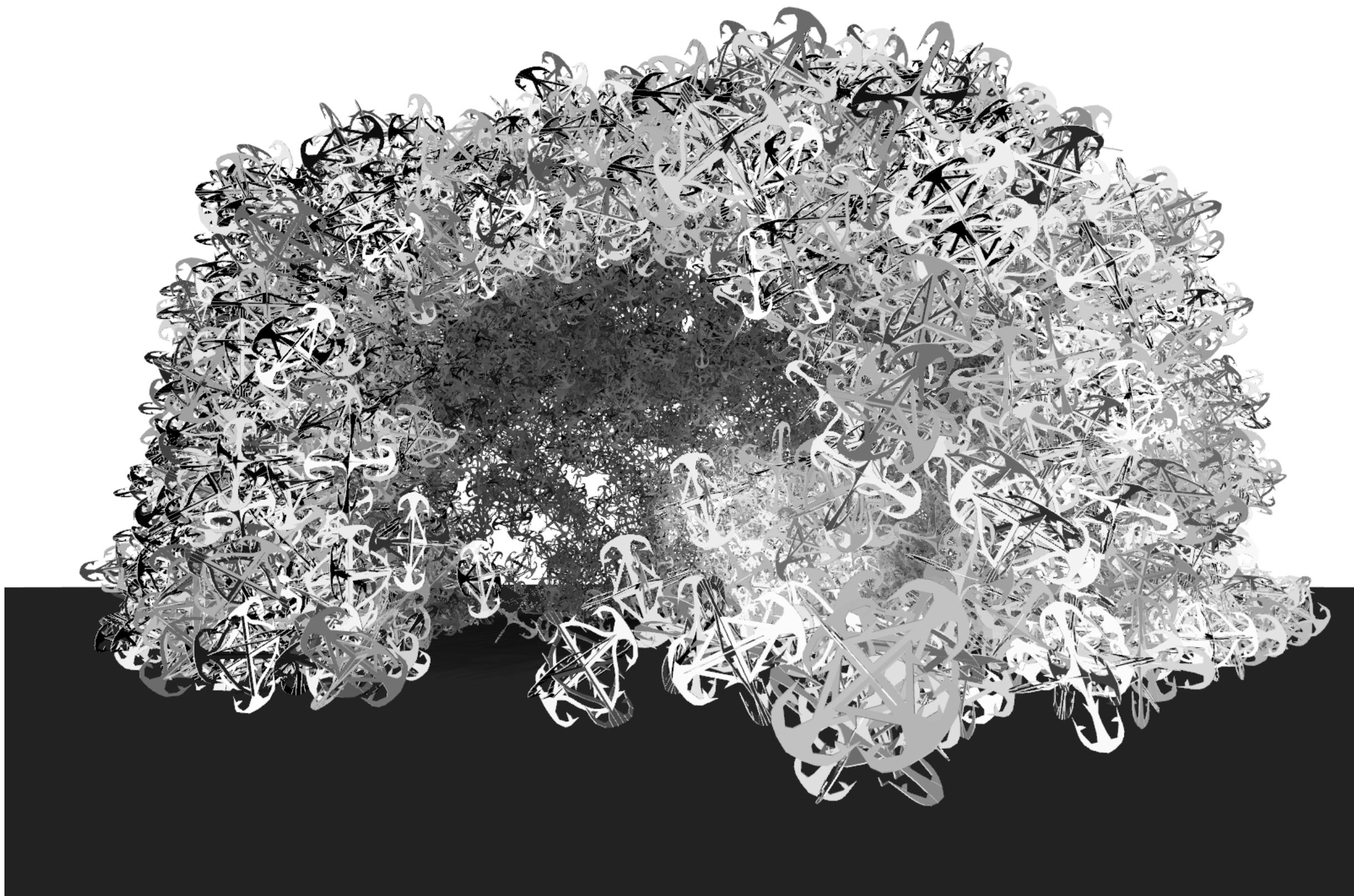
Simulation totalement fluide comprenant environ 120 particules

# AGREGATION

## Simulation de la formation d'un dome

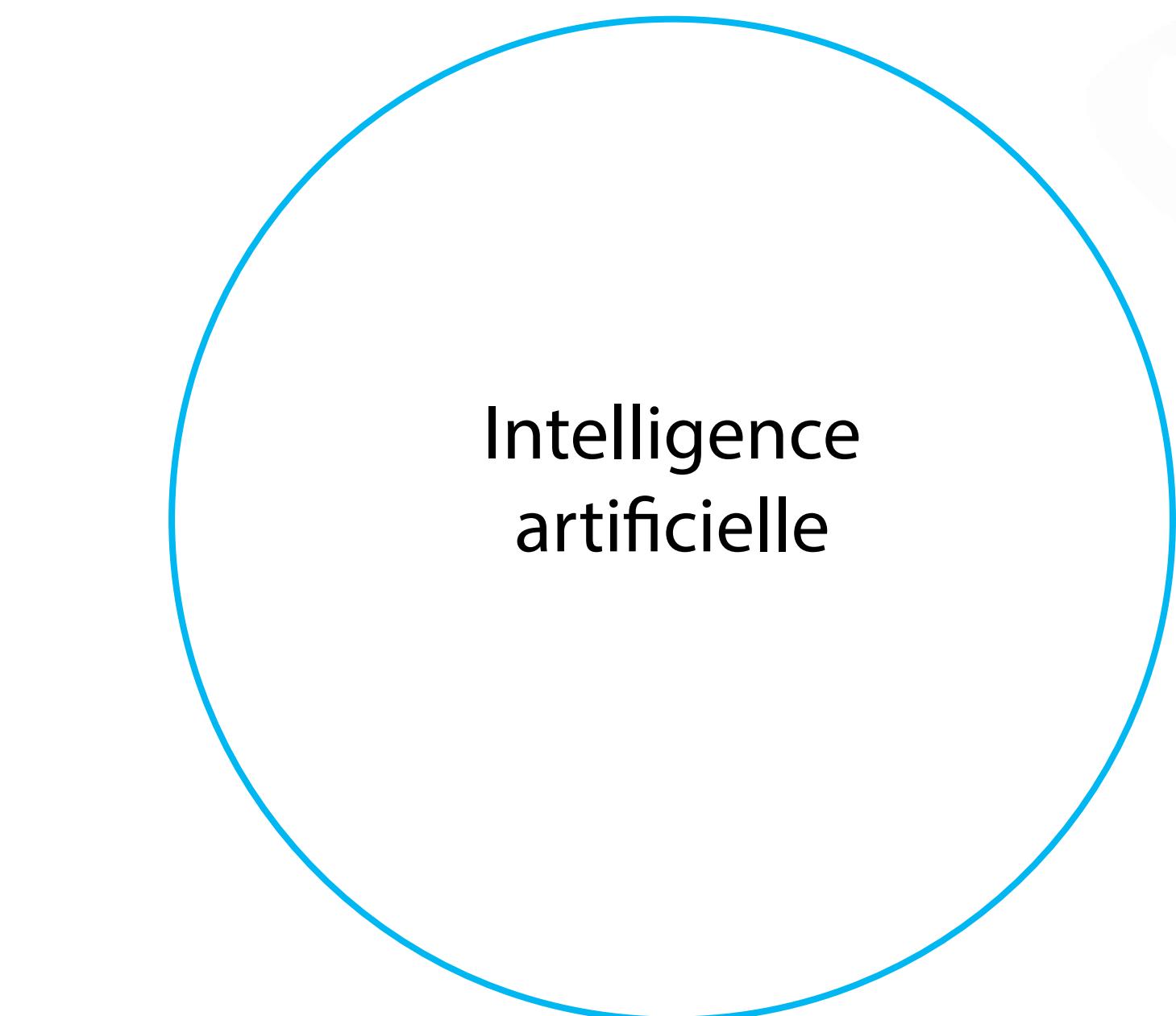
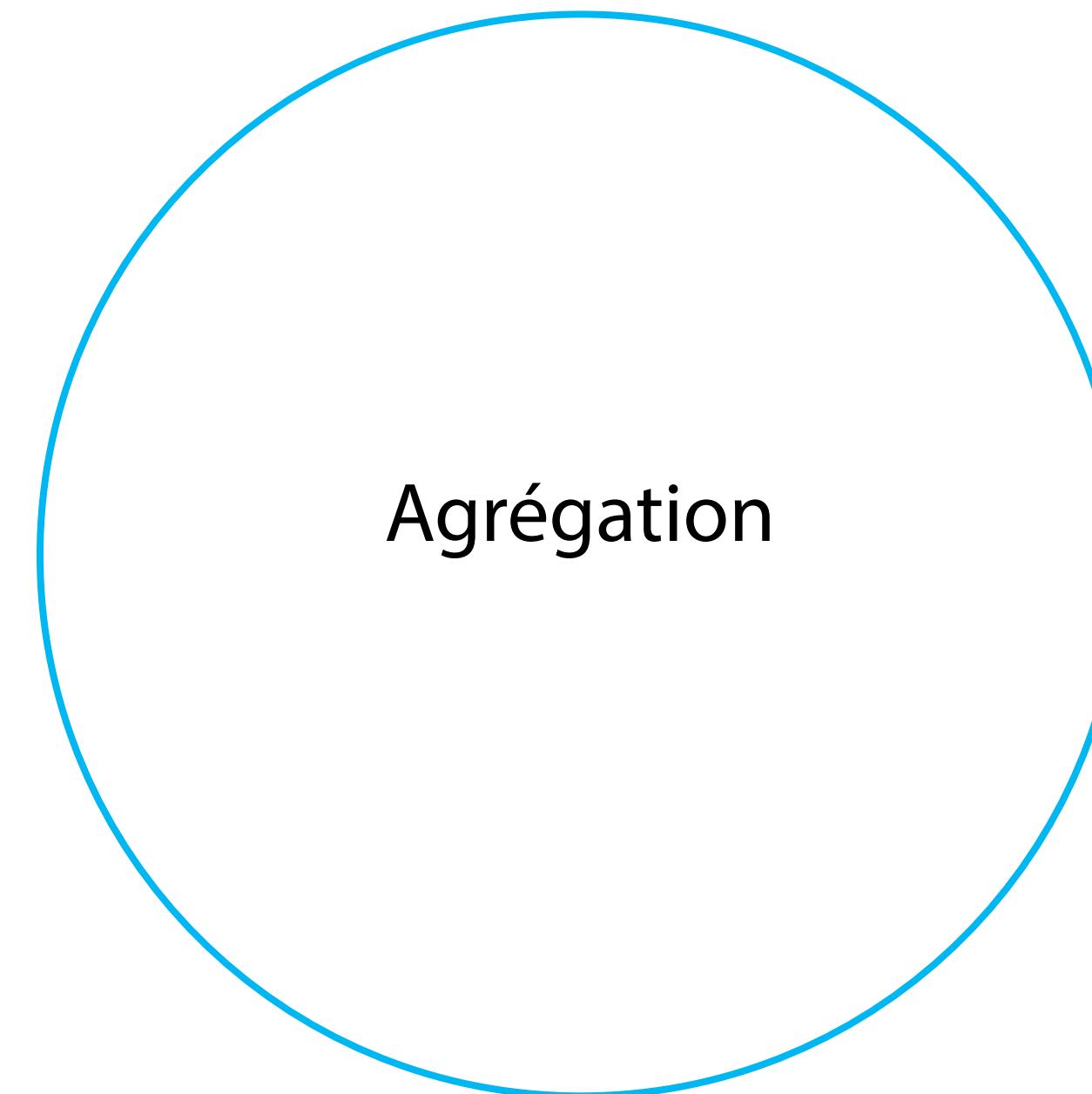
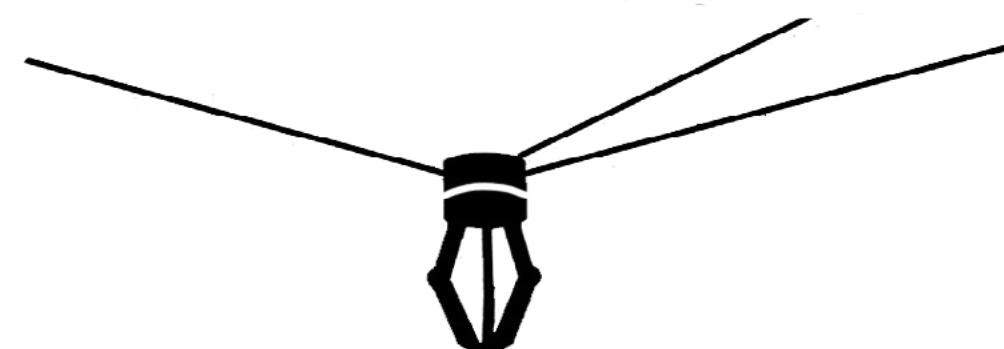
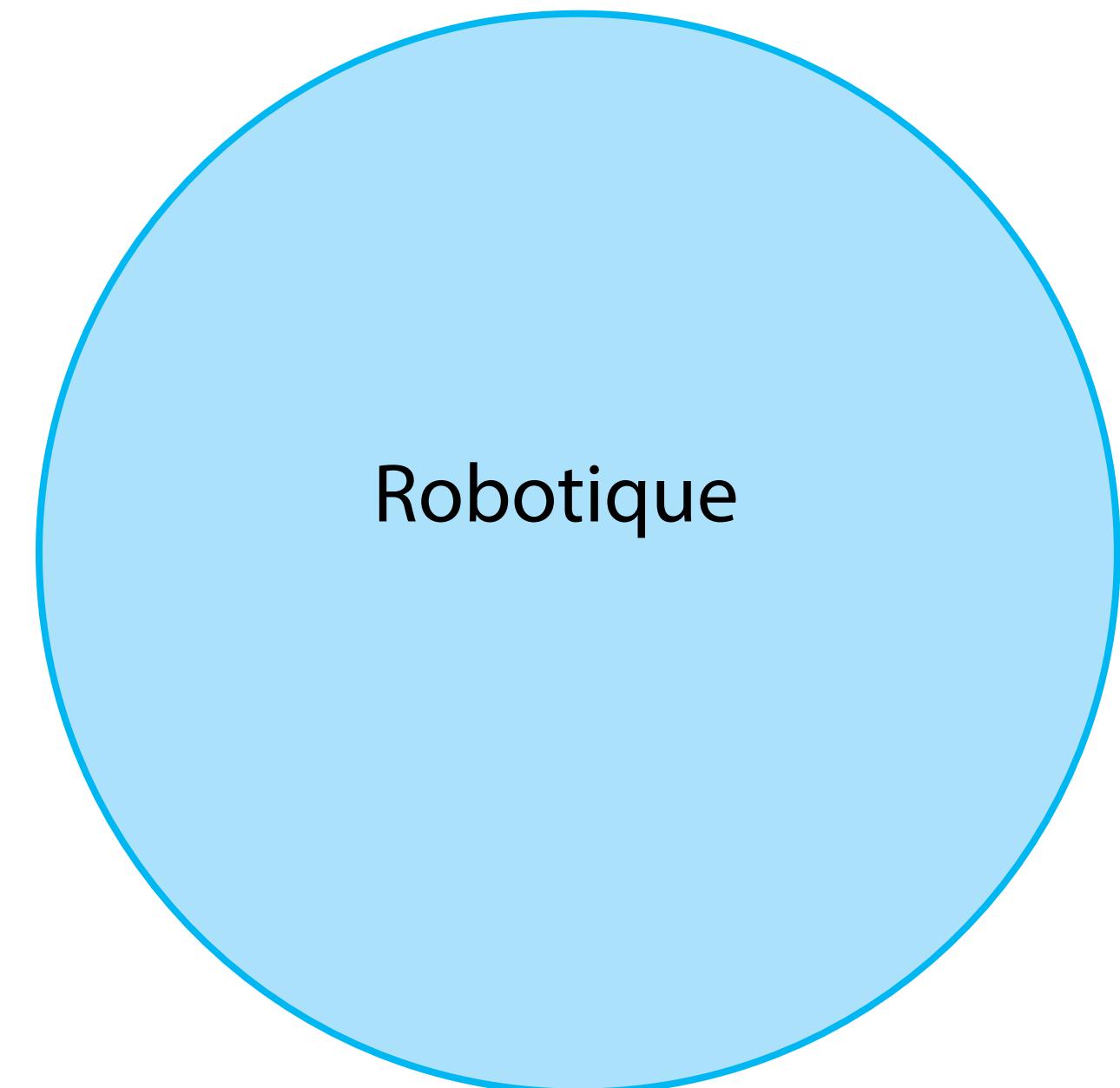
Simulation réalisée sur Processing (3600 particules)

Post-traitement sur Rhino et Grasshopper.



# SYNERGIE

3 axes de recherche

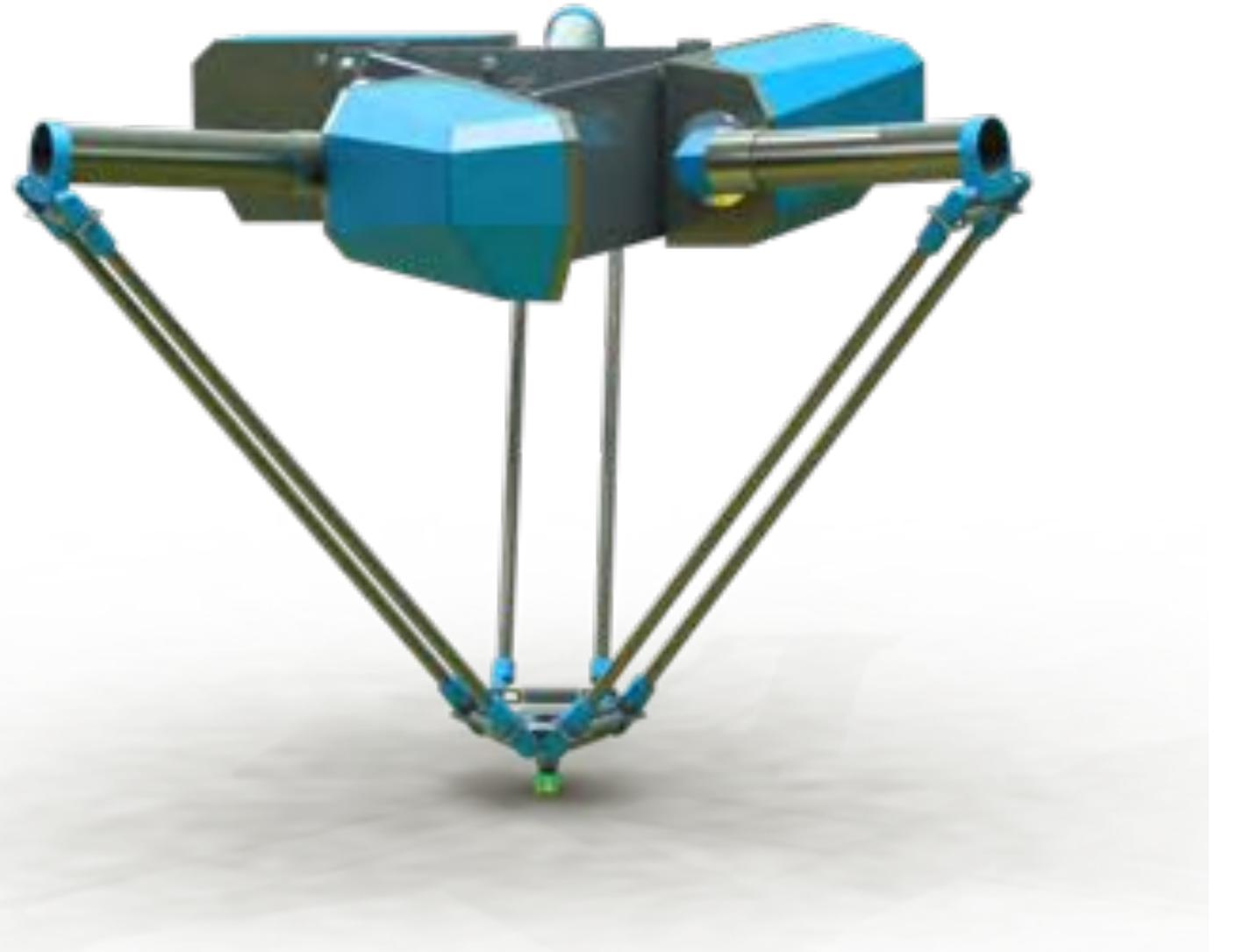


# ROBOTIQUE

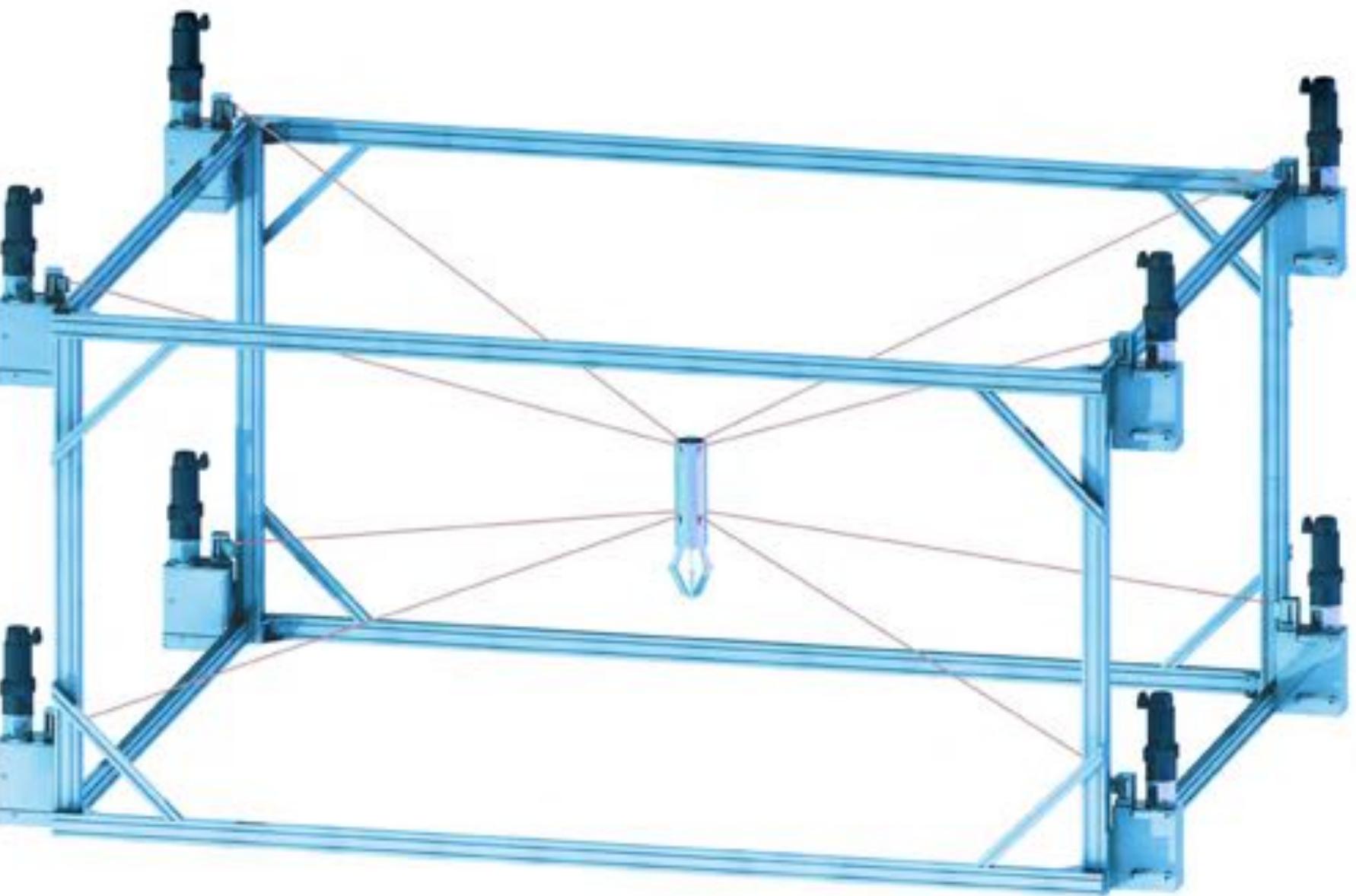
## Références

Après avoir effectué des recherches sur les systèmes de fabrication automatisés, plusieurs objectifs ont été isolés :

- Grande portée
- Vitesse adaptable
- Précision adaptable
- Polyvalence



ABIflexx Delta Robot



IPAnema Cable Robot



SkyCam

# ROBOTIQUE

## Systèmes envisagés

Trois systèmes de fabrication automatisée ont été étudiés :

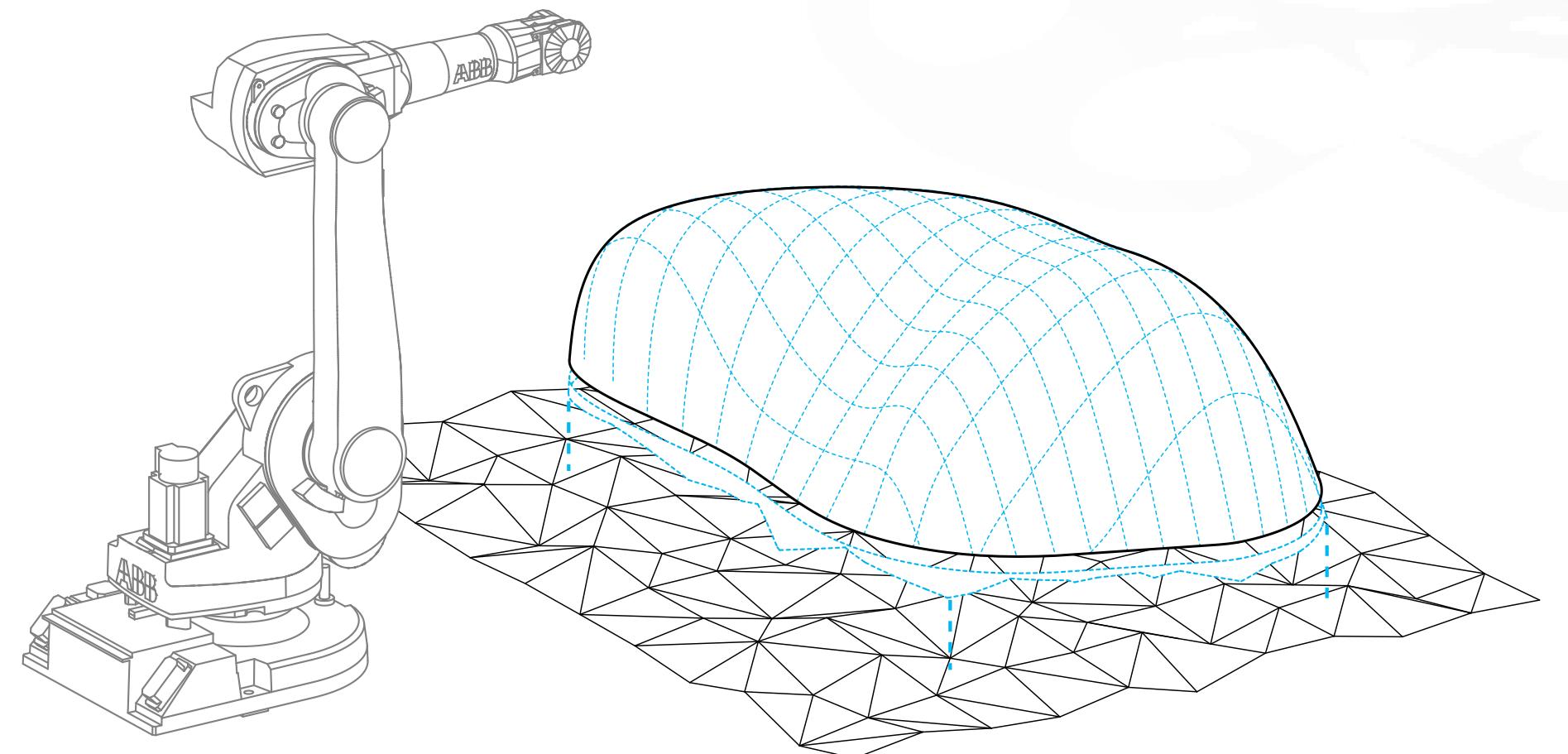
- Robot à câbles
- Balistique d'un projectile
- Robot 6 axes



Balistique (Gramazio Kohler - 2014)



Cables



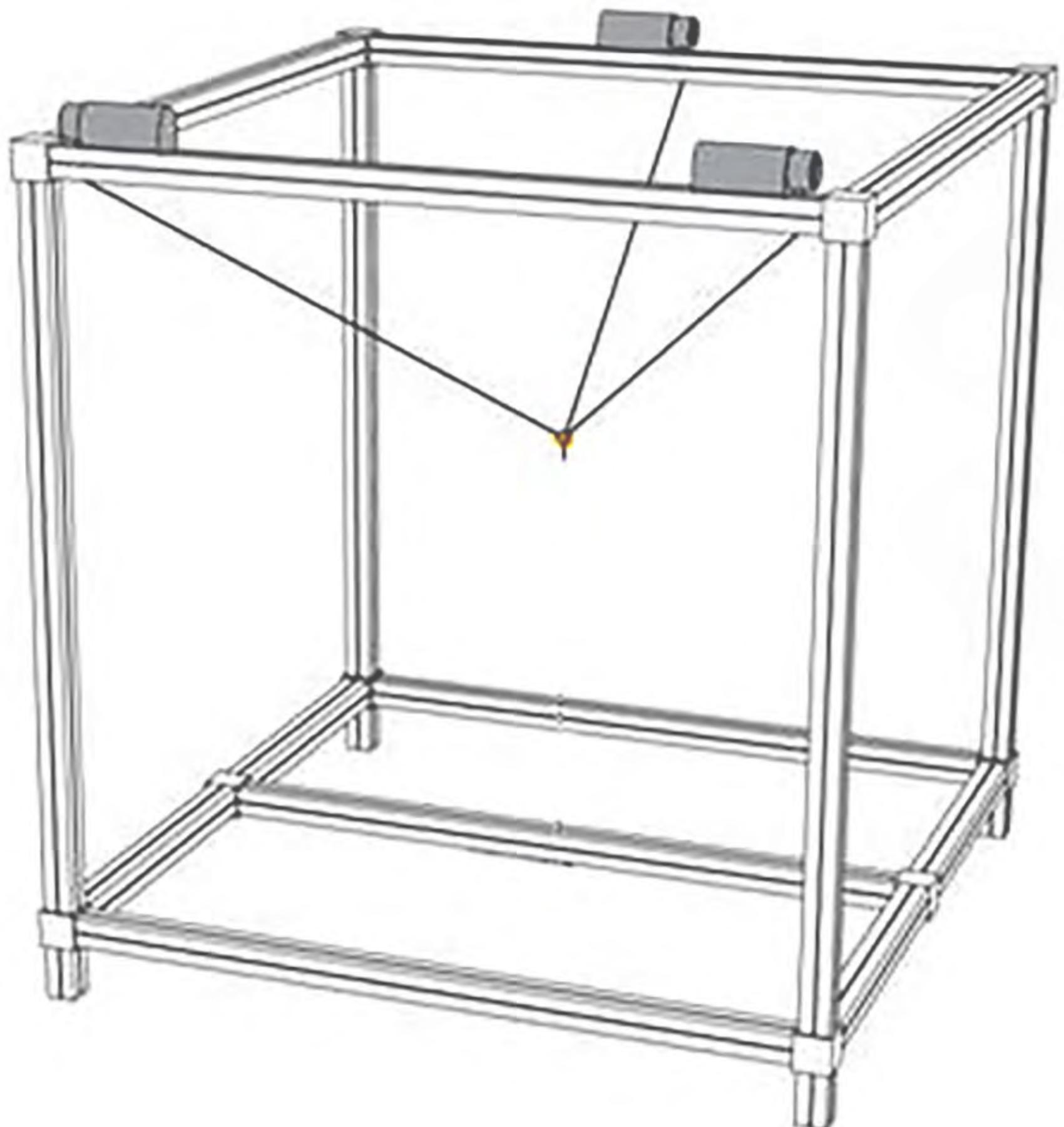
6 axes (P7/P8 2013-2014)

# ROBOTIQUE

## Système retenu

Robot à câbles :

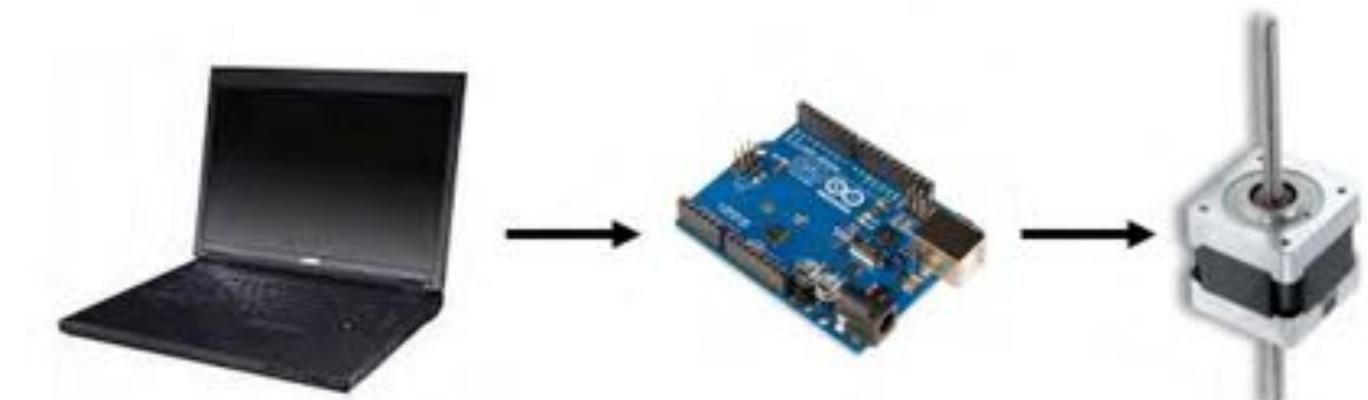
- Grande portée
- Précision
- Portatif
- Manipulable (base Arduino)



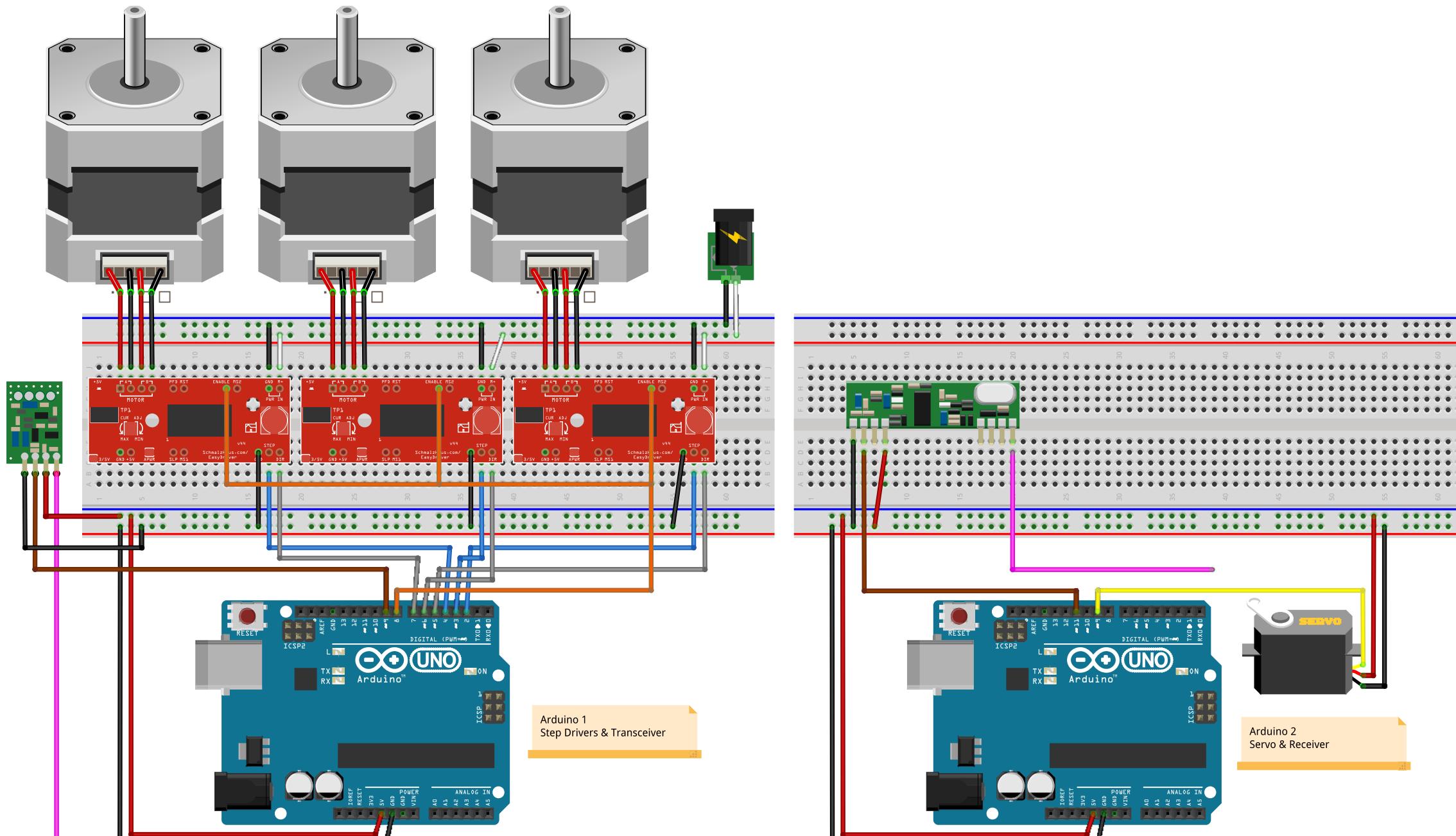
# ROBOTIQUE

## Matériel

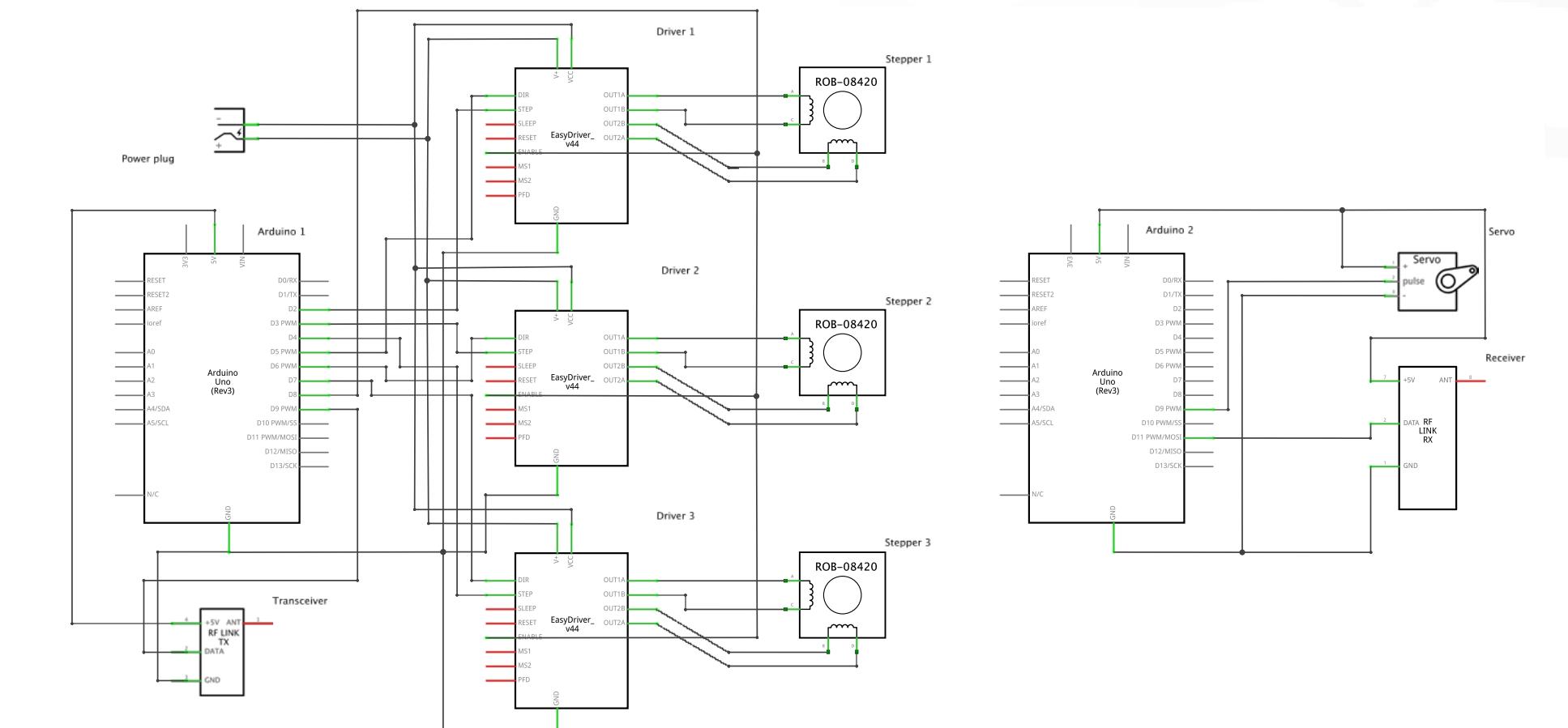
- Moteurs pas à pas (Nema 17)
- Cartes Arduino (UNO)
- Step Drivers (Polulu)
- Cartes de transmission RF (433Mhz)
- Servomoteurs



Principe de contrôle



Arduino Wiring



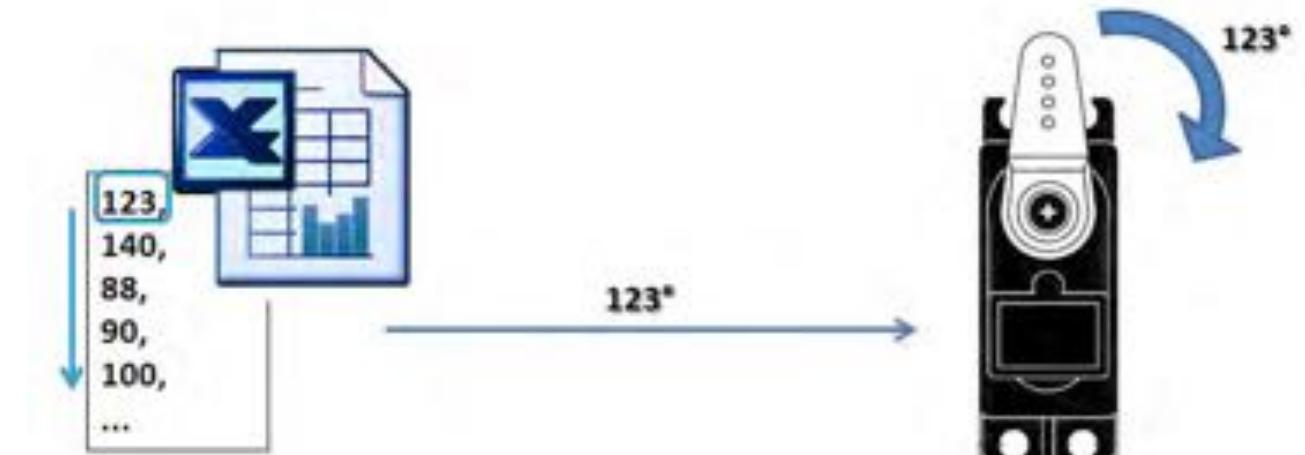
fritzing

Electrical Wiring

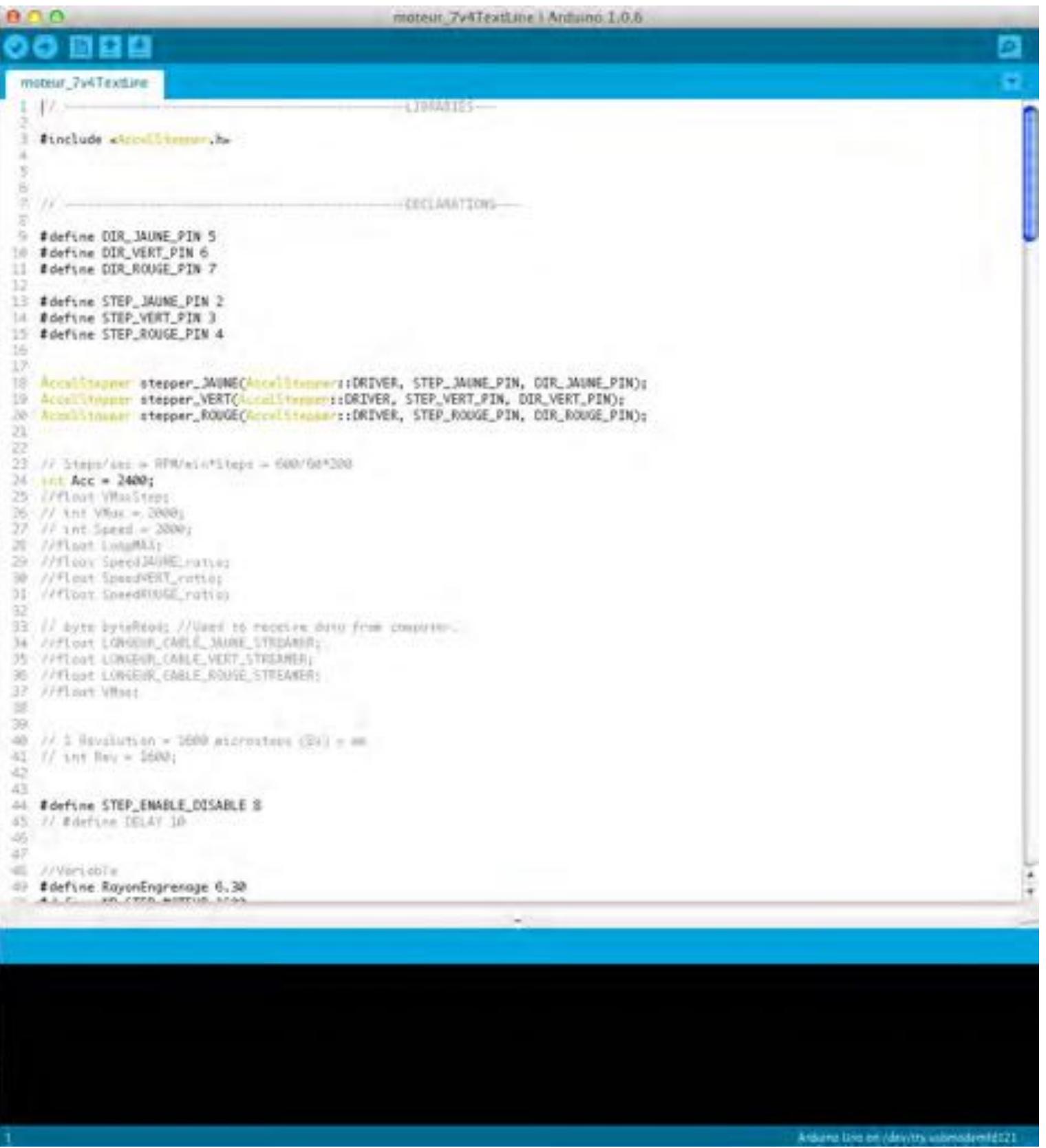
# ROBOTIQUE

## Logiciel

- Arduino IDE
- Processing IDE
- Grasshopper / Rhinoceros



Principe de contrôle



```
moteur_3v4TextLine | Arduino 1.0.6

LIBRARIES—
DECLARATIONS—
DEFINITIONS—
#include <AccelStepper.h>
#defines DIR_JAUNE_PIN 5
#defines DIR_VERT_PIN 6
#defines DIR_ROUGE_PIN 7
#defines STEP_JAUNE_PIN 2
#defines STEP_VERT_PIN 3
#defines STEP_ROUGE_PIN 4
AccelStepper stepper_JAUNE(AccelStepper::DRIVER, STEP_JAUNE_PIN, DIR_JAUNE_PIN);
AccelStepper stepper_VERT(AccelStepper::DRIVER, STEP_VERT_PIN, DIR_VERT_PIN);
AccelStepper stepper_ROUGE(AccelStepper::DRIVER, STEP_ROUGE_PIN, DIR_ROUGE_PIN);

void setup() {
  // Steps/deg = RPM/min*Stepd = 600/60*200
  int Acc = 2400;
  float VMaxstep;
  // int VMax = 2000;
  // int Speed = 2000;
  //float LmaxMAX;
  //float SpeedJAUNE_ratio;
  //float SpeedVERT_ratio;
  //float SpeedROUGE_ratio;
  //int byteReads; //Used to receive data from computer
  float LONGEUR_CABLE_JAUNE_STREAKER;
  float LONGEUR_CABLE_VERT_STREAKER;
  float LONGEUR_CABLE_ROUGE_STREAKER;
  float VMax;
  // 1 Revolution = 2600 microsteps (29) * 80
  int Rev = 1600;
  #define STEP_ENABLE_DISABLE 8
  // #define DELAY 10
  //Variables
  #define RayonEngrenage 6.38
  #define INCREMENT 0.001
}

void loop() {
  if (Serial.available() > 0) {
    String data = Serial.readStringUntil('\n');
    if (data != "") {
      String tokens[5];
      tokens[0] = data.substring(0, data.indexOf(' '));
      tokens[1] = data.substring(data.indexOf(' ') + 1, data.indexOf(' '));
      tokens[2] = data.substring(data.indexOf(' ') + 1, data.indexOf(' '));
      tokens[3] = data.substring(data.indexOf(' ') + 1, data.indexOf(' '));
      tokens[4] = data.substring(data.indexOf(' ') + 1, data.indexOf(' '));
      int jaune = tokens[0].toInt();
      int vert = tokens[1].toInt();
      int rouge = tokens[2].toInt();
      int jauneDir = tokens[3].toInt();
      int vertDir = tokens[4].toInt();
      stepper_JAUNE.setSpeed(jaune);
      stepper_VERT.setSpeed(vert);
      stepper_ROUGE.setSpeed(rouge);
      stepper_JAUNE.setDirection(jauneDir);
      stepper_VERT.setDirection(vertDir);
      stepper_ROUGE.setDirection(rouge);
    }
  }
}
```

Firmware Arduino



```
TextLine_v2 | Processing 3.0.5

TextFile Data.sendel((stage 1))
Written by ScottC on 24th April 2013
using Processing Version 2.0b8

The full tutorial can be found here:
http://arduinobasics.blogspot.com/2013/04/serial-communication-tutorial-part-3.html
import processing.serial.*;

Serial comPort; //The com port used between the computer and Arduino
int counter=0; //Helps to keep track of values sent.
int numItems=0; //Keep track of the number of values in text file
String comPortString; //String received From Arduino
String textFileLines[]; //Array of text file lines
String lineItems[]; //Array of line items

void setup() {
  println(serial.list());
  println();
  comPort = new Serial(this, serial.list()[5], 9600); //Setup the COM port.
  comPort.bufferUntil('\n'); //Generate a SerialEvent when a newline is received
  background(255, 0, 0); //Start with a Red background
}

/* Draw method is not used in this sketch */
void draw() {
}

//When the mouse is pressed, write an "E" to COM port.
//The Arduino should send back an "A" in return. This will
//generate a serialEvent - see below.
void mousePressed() {
  comPort.write("m");
}

void serialEvent(Serial cPort) {
  comPortString = cPort.readStringUntil('\n');
  if (comPortString != null) {
    comPortString.trim(comPortString);
    println(comPortString);
  }
}
```

Streaming Processing

# ROBOTIQUE

## Historique

Solution 1 :

- Emmagasineur

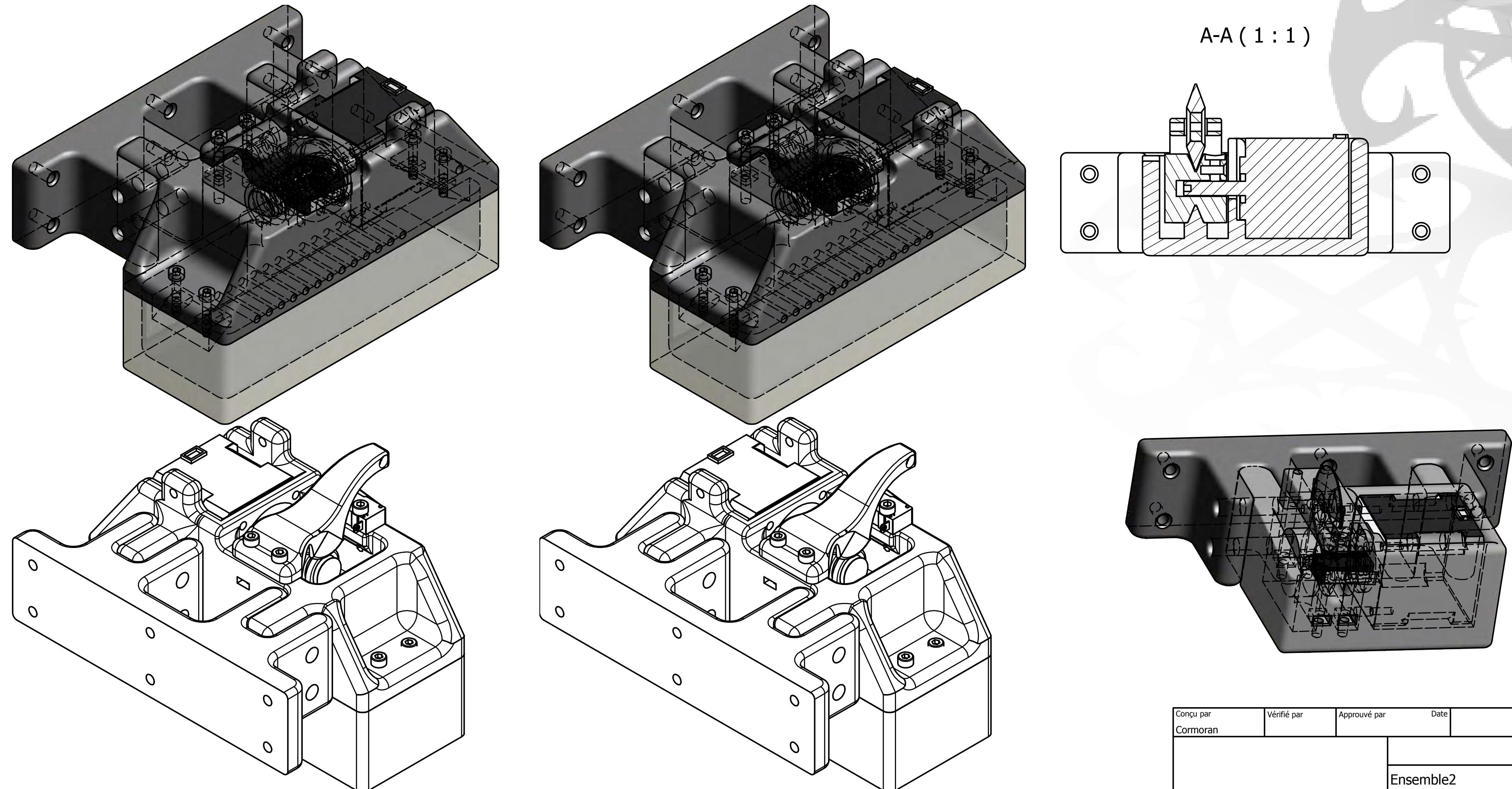
Pas de stockage direct du câble, enroulement et déroulement par taquet coincer

Avantages :

- Pas de gestion du câble
- Faible encombrement
- Pièces en impression 3D
- Roulements à billes

Problèmes :

- Mauvais entraînement



# ROBOTIQUE

## Historique

Solution 2 :

- Enrouleur

Système utilisé pour validation des tests de trajectoires, codes, programmation...

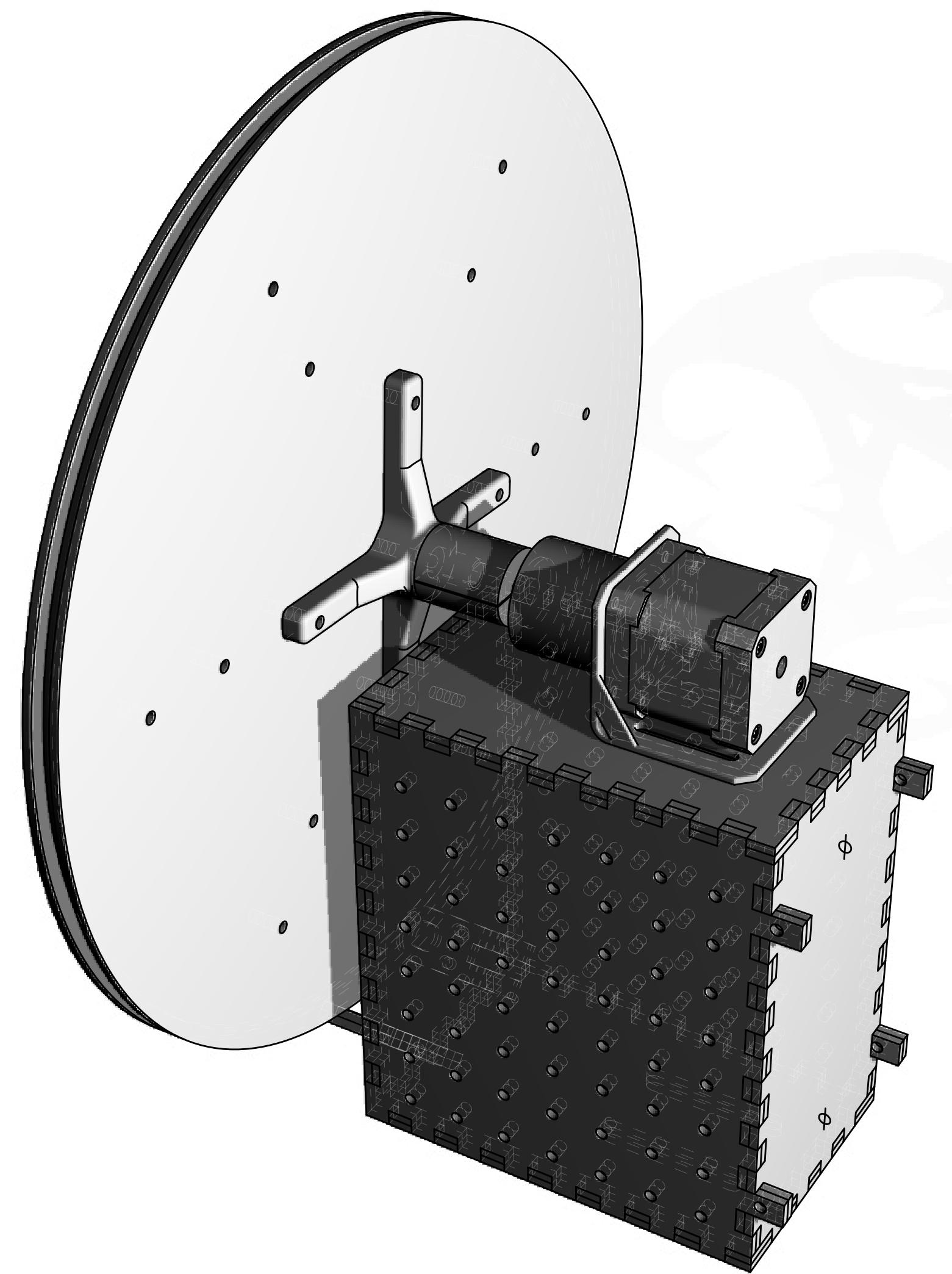
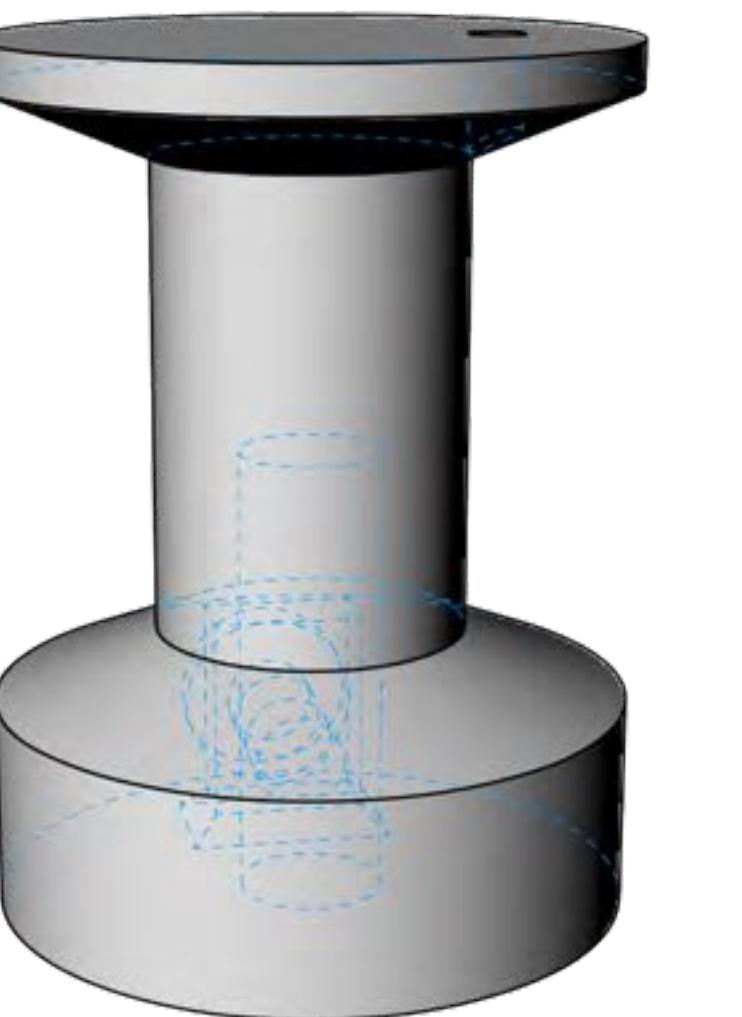
Eviter le glissement du câble

Avantages :

- Pas de glissement du câble
- Démultiplication avec réducteur

Problèmes :

- Erreurs de longueurs dues au sur-enroulement du câble
- Longueur limitée à la taille de poulie
- Couple variable (réducteur requis)
- Encombrement variable

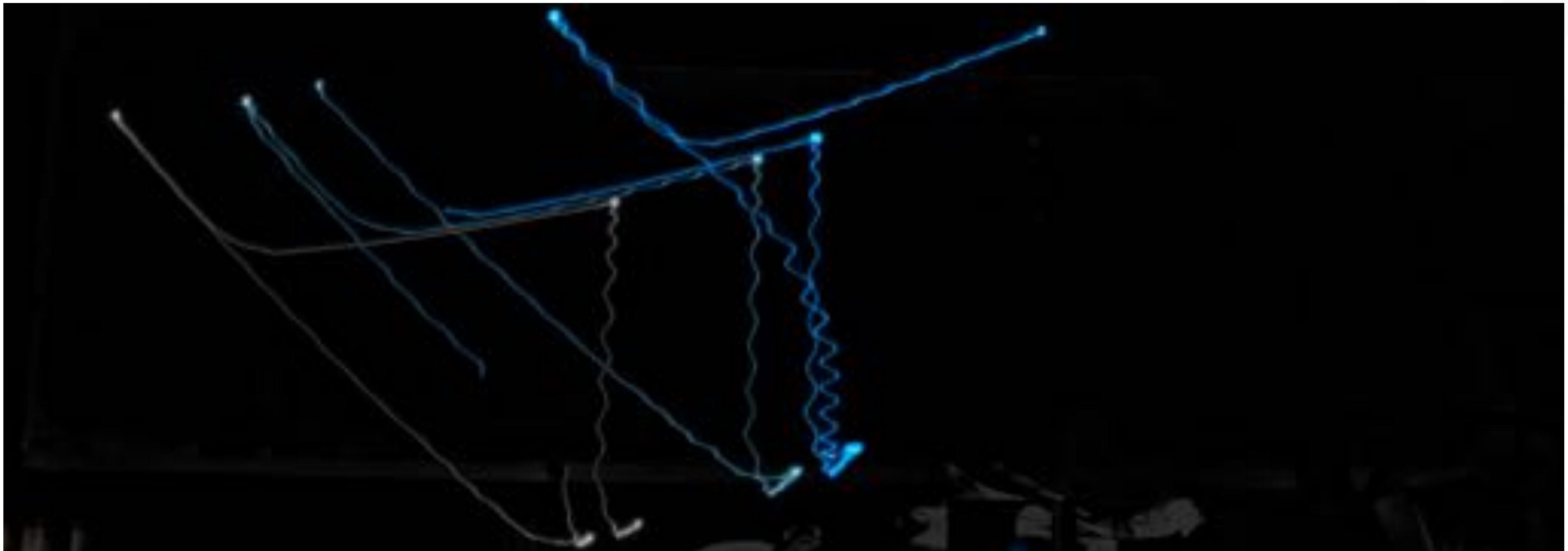


# ROBOTIQUE

## Dimensionnement des moteurs

Nom	ref	PrixUnité	Prix pondéré	No Step	PrecisionDirecte	Couple	ForceDirecte	Reduction	Rendement	PrecisionReduceur	couple sortie	ForceDemultiplié		
					mm	Nm	N	Kg	x/1	%	mm	Nm	N	Kg
17R.51/1	17HS19-16845-PG51	28,09	104,36	200	0,942	4	133,333	13,5916	50,9	73	0,019	148,628	4854,266667	505,022
17R.27/1	17HS19-16845-PG27	34,21	126,84	200	0,942	3	100	10,1937	26,85	81	0,035	65,2455	2174,85	221,697
17R.14/1	17HS19-16845-PG14	23,24	88,96	200	0,942	3	100	10,1937	13,73	81	0,069	33,3639	1112,13	111,061
17R.5/1	17HS19-16845-PG51	22,71	85,84	200	0,942	2	66,6667	6,79579	5,18	90	0,182	9,324	310,8	31,682
24	24HS14-35040	18,91	70,64	200	0,942	3,1	100	30,5335						
34	34HS59-50045	48,28	185,96	200	0,942	13	433,333	44,1726						
Nom	ref		Vmax											
			RPM	RPS										
17R.51/1	17HS19-16845-PG51													
17R.27/1	17HS19-16845-PG27													
17R.14/1	17HS19-16845-PG14													
17R.5/1	17HS19-16845-PG51													

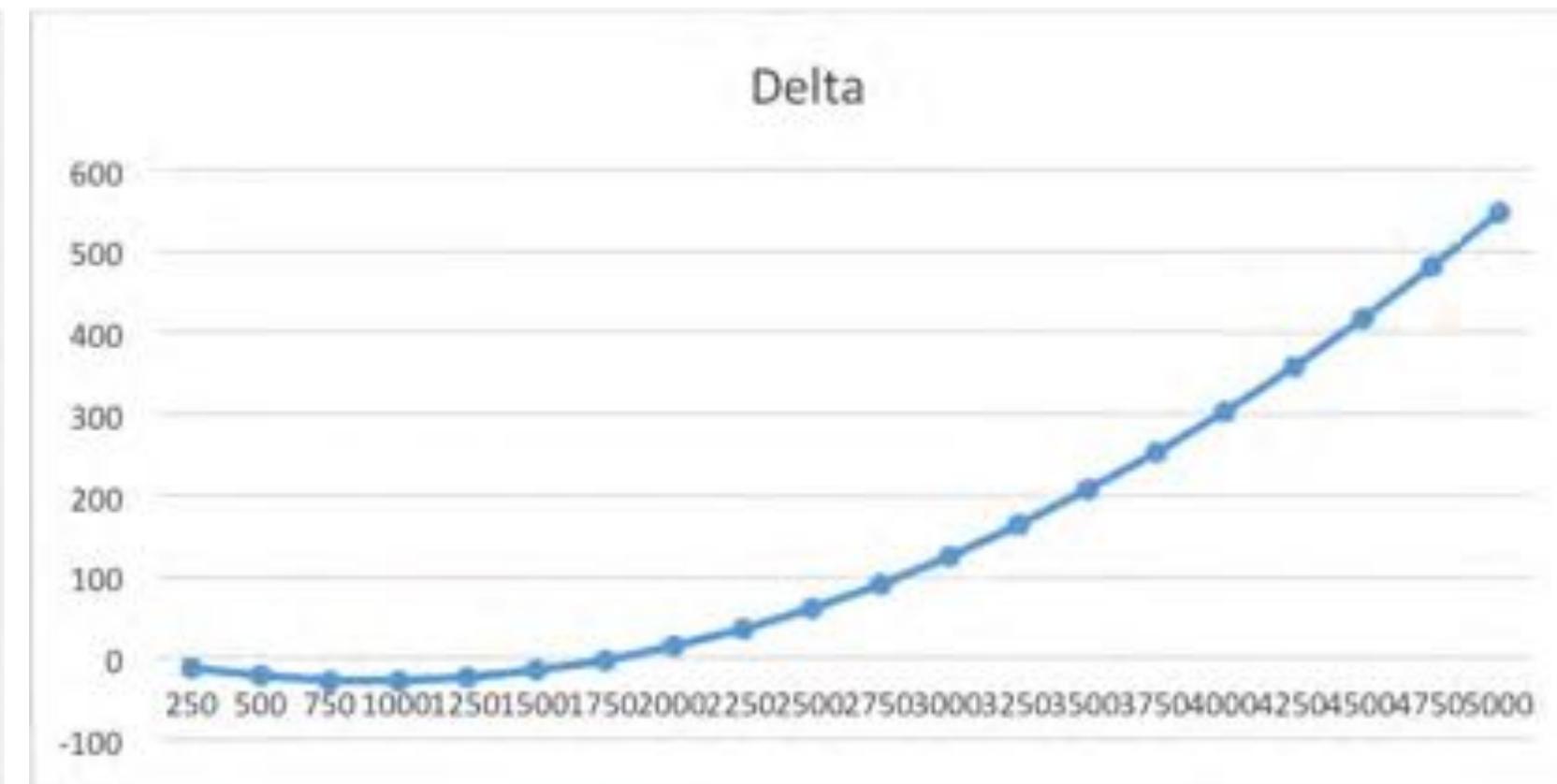
Charges des différents moteurs



Test de trajectoire

# ROBOTIQUE

## Prototype

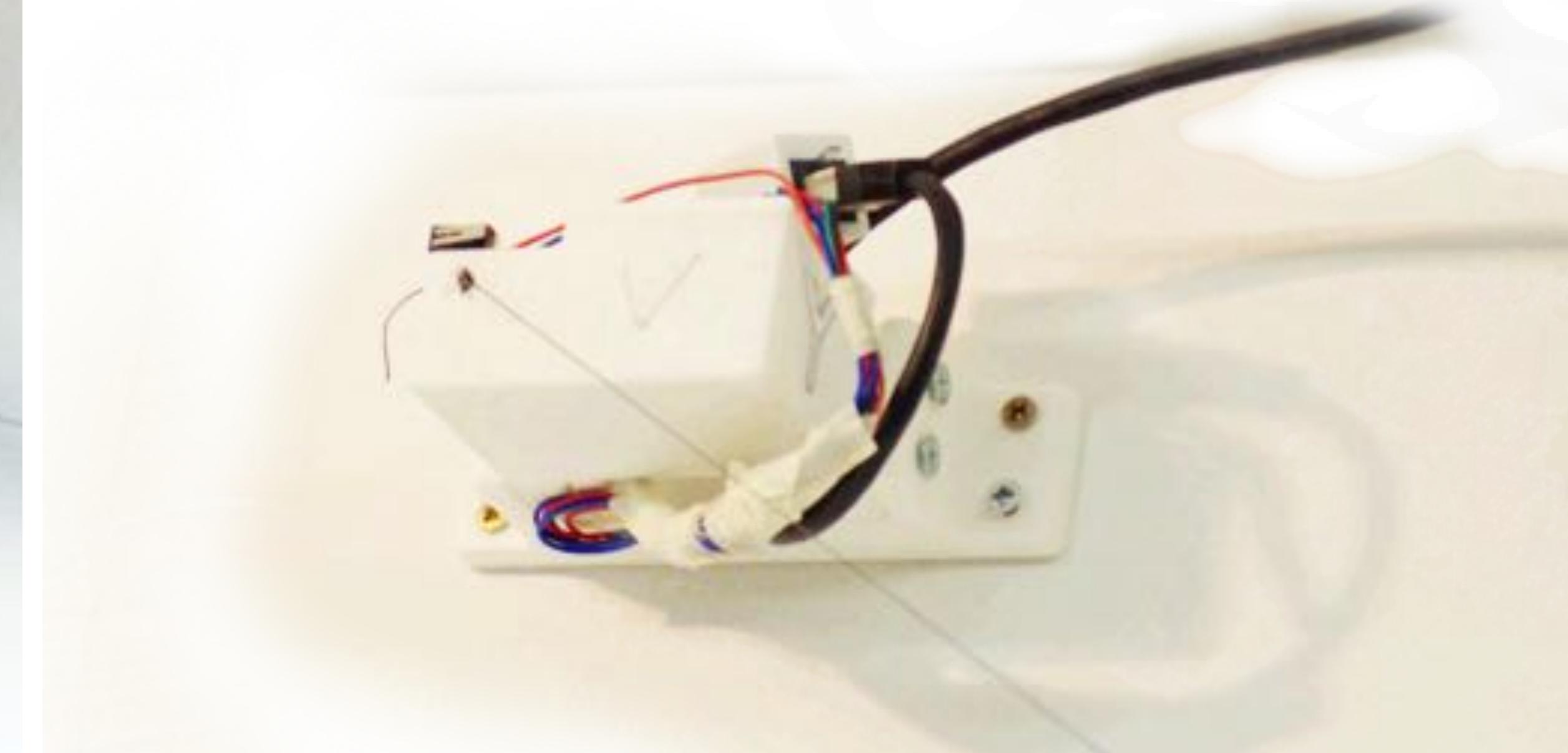


Distance mm	mm/Pas	Nb Pas	Nb Pas T	coeff f	Delta
0	0,1832	0	0	0	0
250	0,1835	1362,39782	1348,770461	-13,62735932	
500	0,1838	2720,3482	2697,540922	-22,80728287	
750	0,1841	4073,8729	4046,311383	-27,56151262	
1000	0,1844	5422,99349	5395,081843	-27,91164902	
1250	0,1847	6767,73146	6743,852304	-23,87915218	
1500	0,185	8108,10811	8092,622765	-15,48534302	
1750	0,1853	9444,14463	9441,393226	-2,751404394	
2000	0,1856	10775,8621	10790,16369	14,30161782	
2250	0,1859	12103,2813	12138,93415	35,65281358	
2500	0,1862	13426,4232	13487,70461	61,28140762	
2750	0,1865	14745,3083	14836,47507	91,16675833	
3000	0,1868	16059,9572	16185,24553	125,2883567	
3250	0,1871	17370,3902	17534,01599	163,6258253	
3500	0,1874	18676,6275	18882,78645	206,1589172	
3750	0,1877	19978,6894	20231,55691	252,8675147	
4000	0,188	21276,5957	21580,32737	303,7316289	
4250	0,1883	22570,3664	22929,09783	358,7313979	
4500	0,1886	23860,0212	24277,8683	417,8470864	
4750	0,1889	25145,5797	25626,63876	481,0590843	
5000	0,1892	26427,0613	26975,40922	548,3479062	

Calculs des erreurs d'enroulements

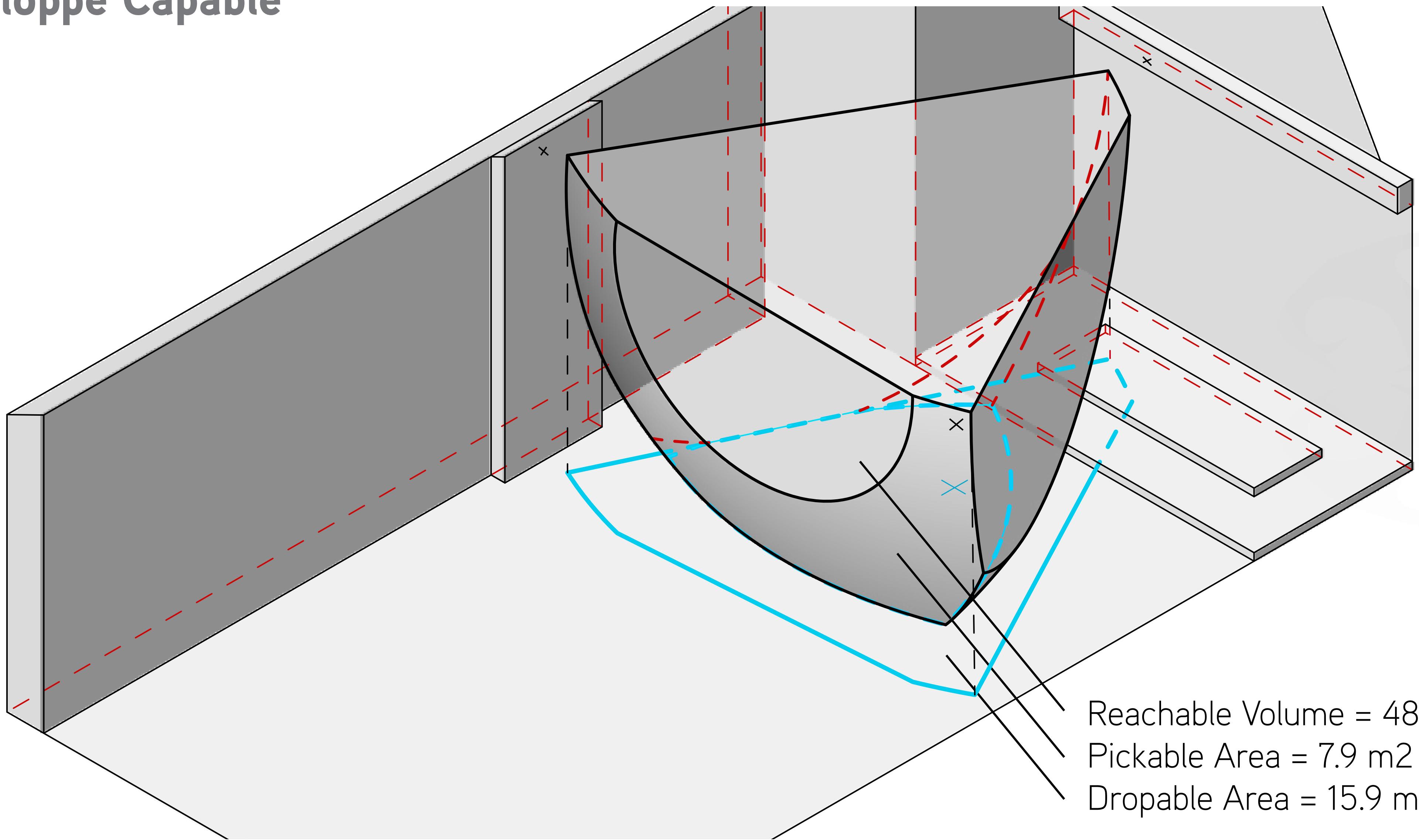


WormRope v4



# ROBOTIQUE

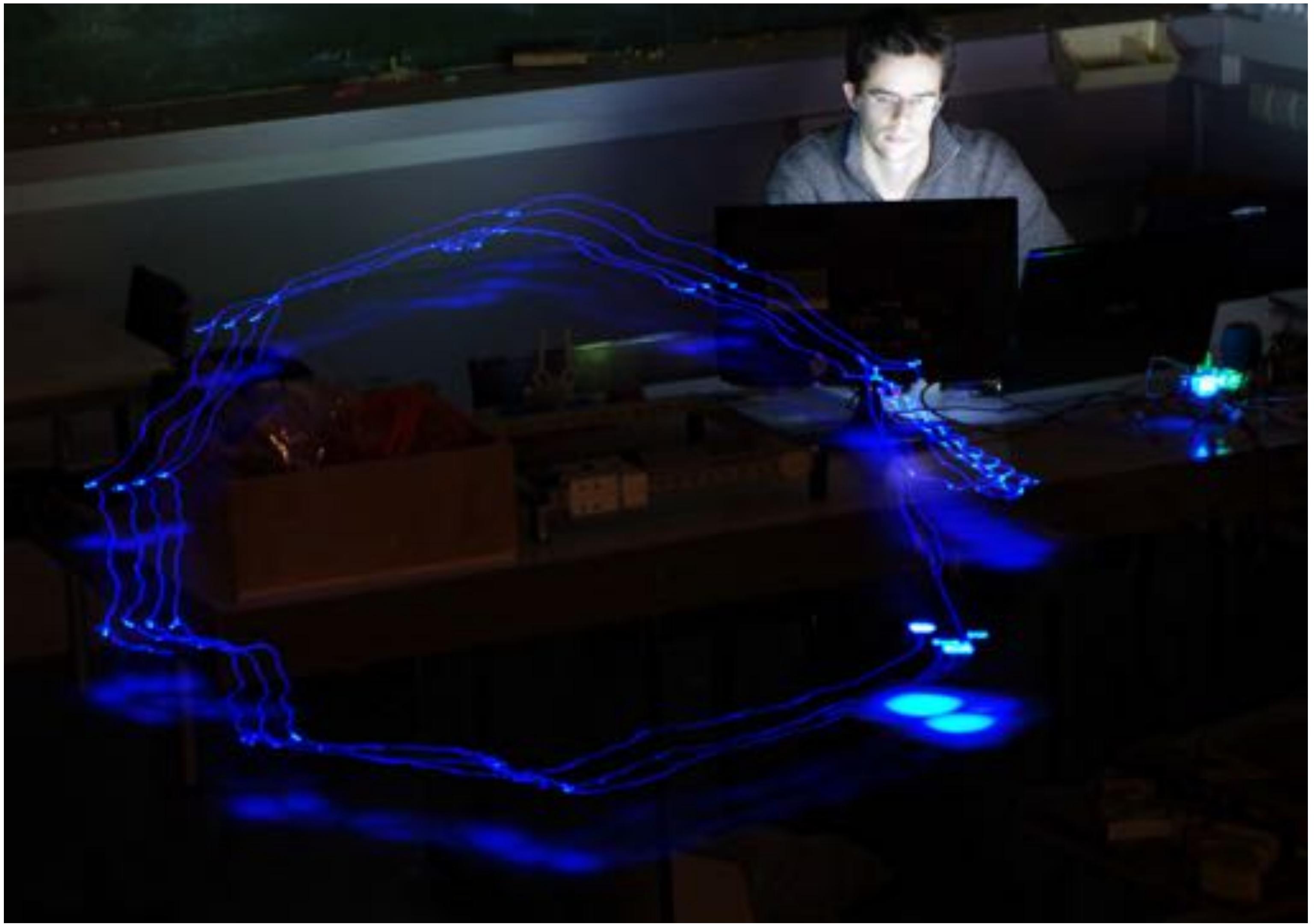
## Enveloppe Capable



Reachable Volume = 48.6 m<sup>3</sup>  
Pickable Area = 7.9 m<sup>2</sup>  
Dropable Area = 15.9 m<sup>2</sup>

# ROBOTIQUE

## Prototype



WormRope v4 // Test des trajectoires

# ROBOTIQUE

## Système à terme

Solution 3 :

- Treuil

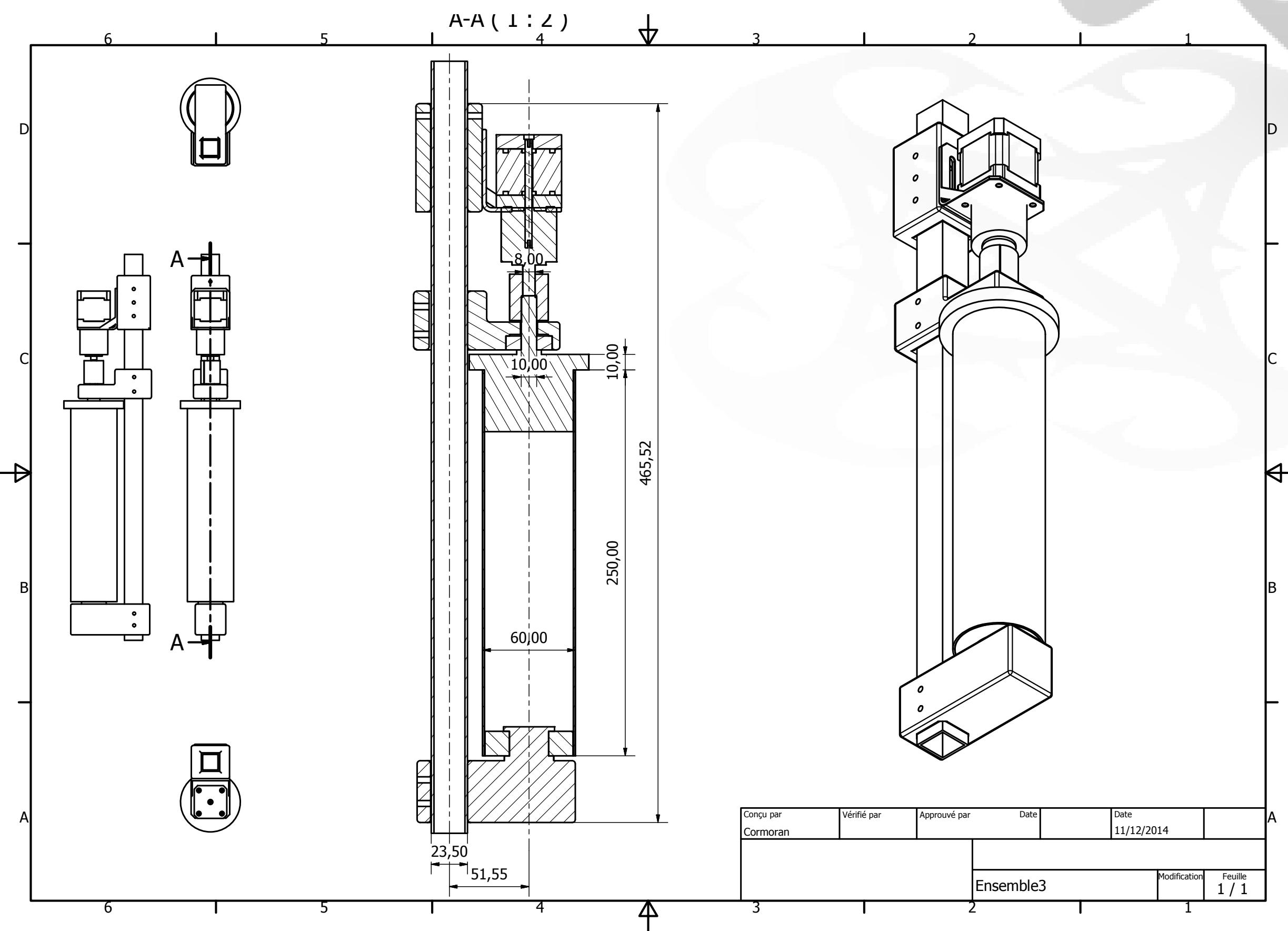
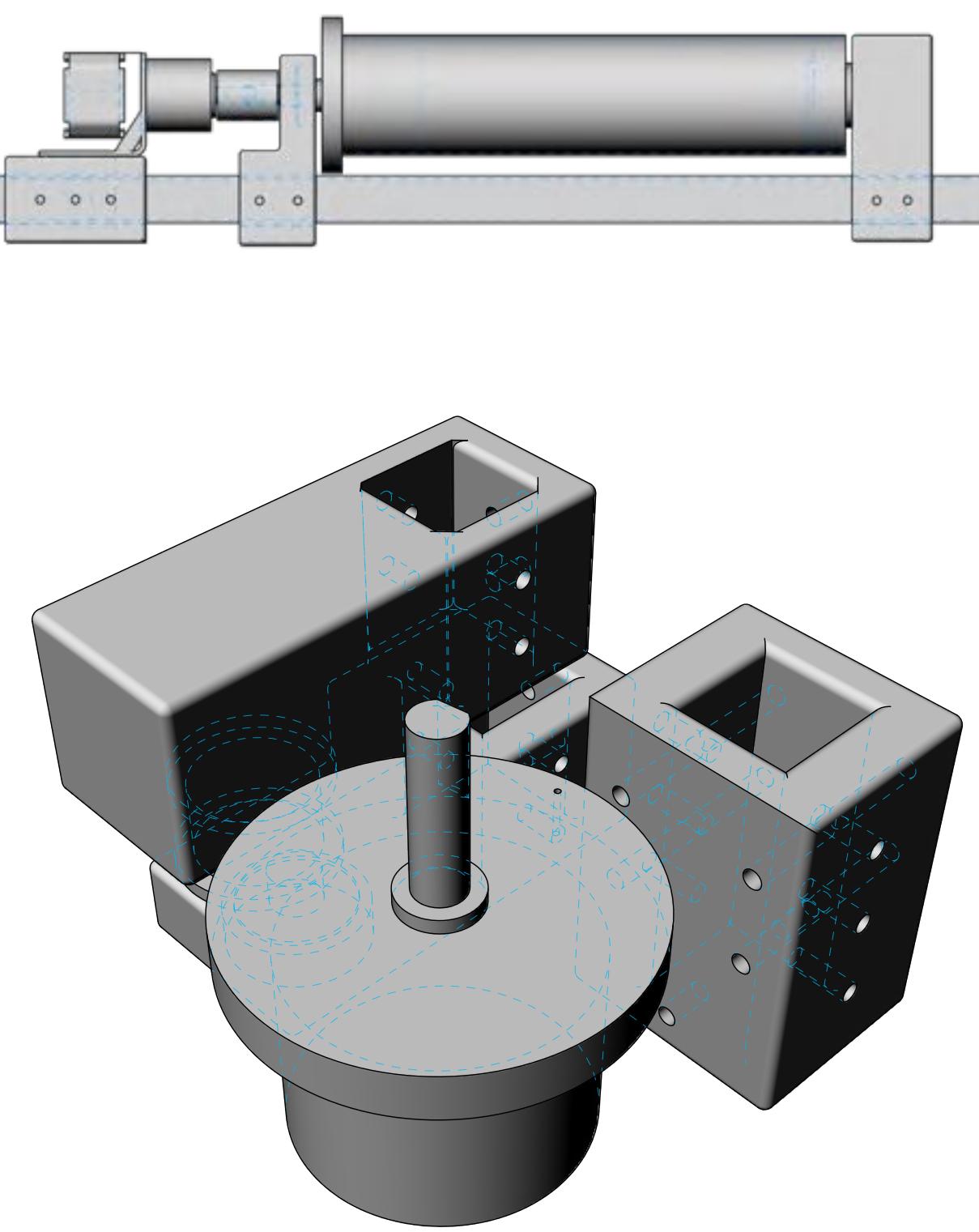
Guidage du câble et minimiser les sur-  
épaisseurs

Avantages :

- Robuste
- Grande longueur
- Aucun effort d'axe moteur
- Roulements à billes

Problèmes :

- Encombrement



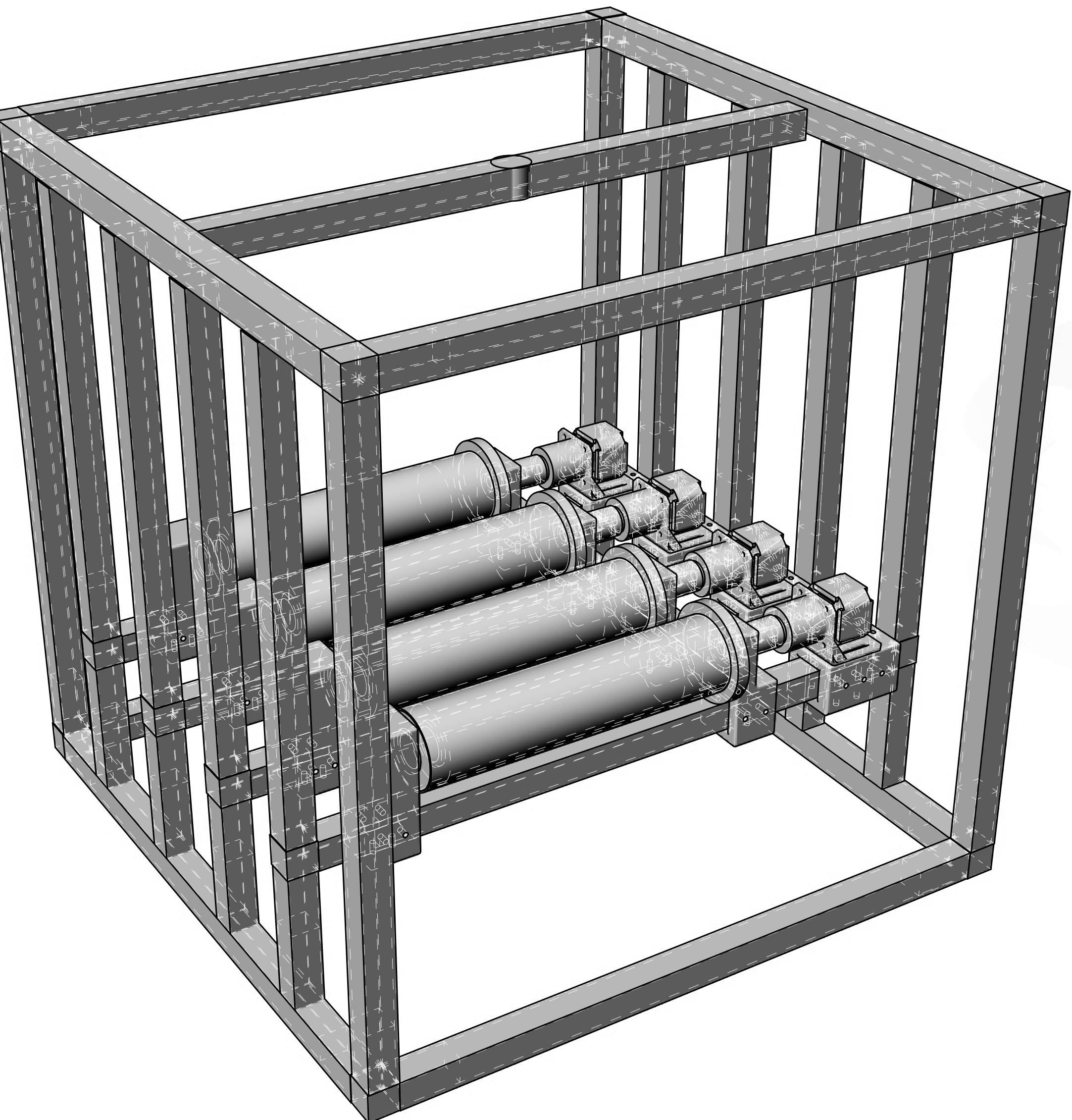
# ROBOTIQUE

## Système à terme

Solution 3 :

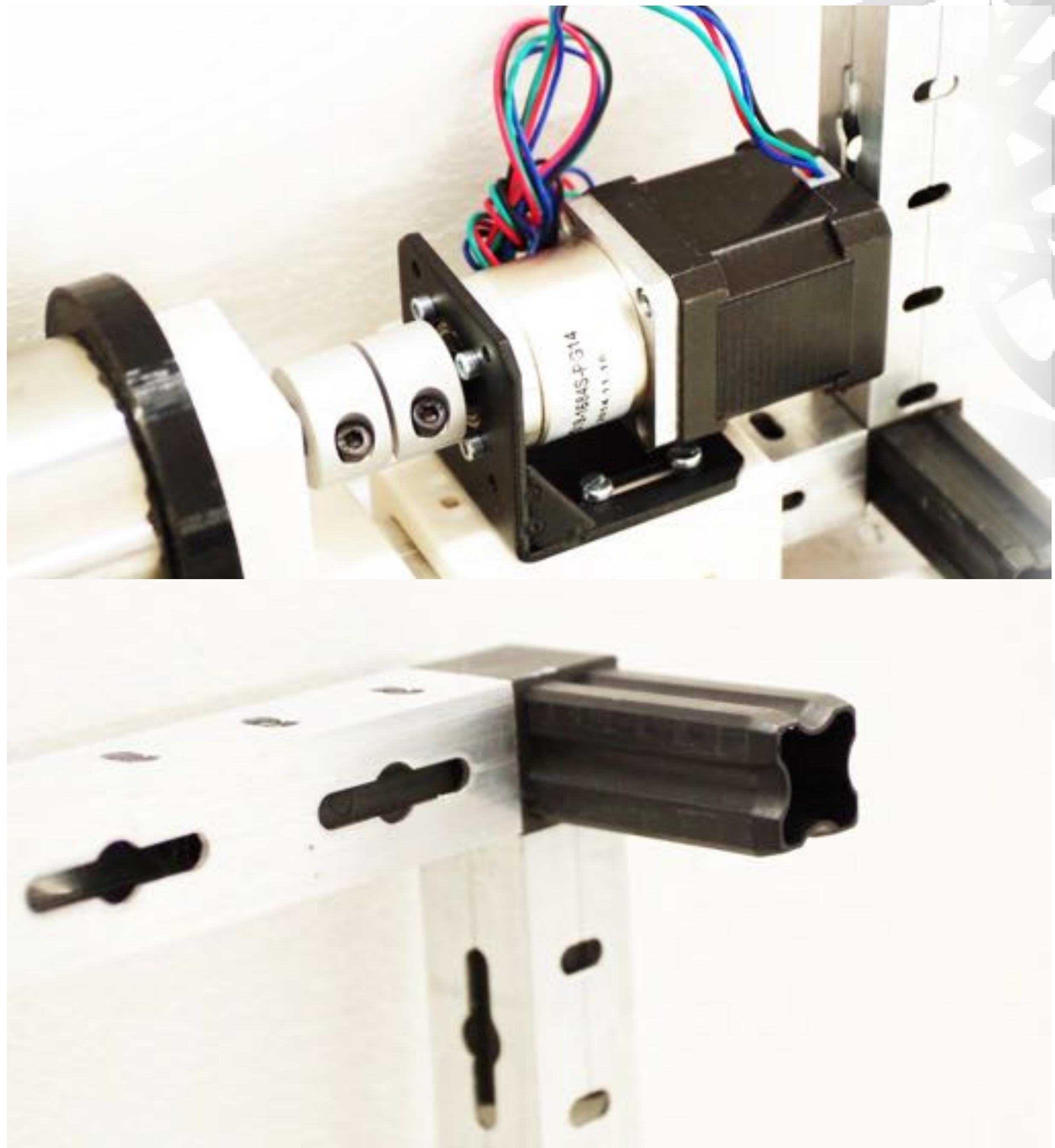
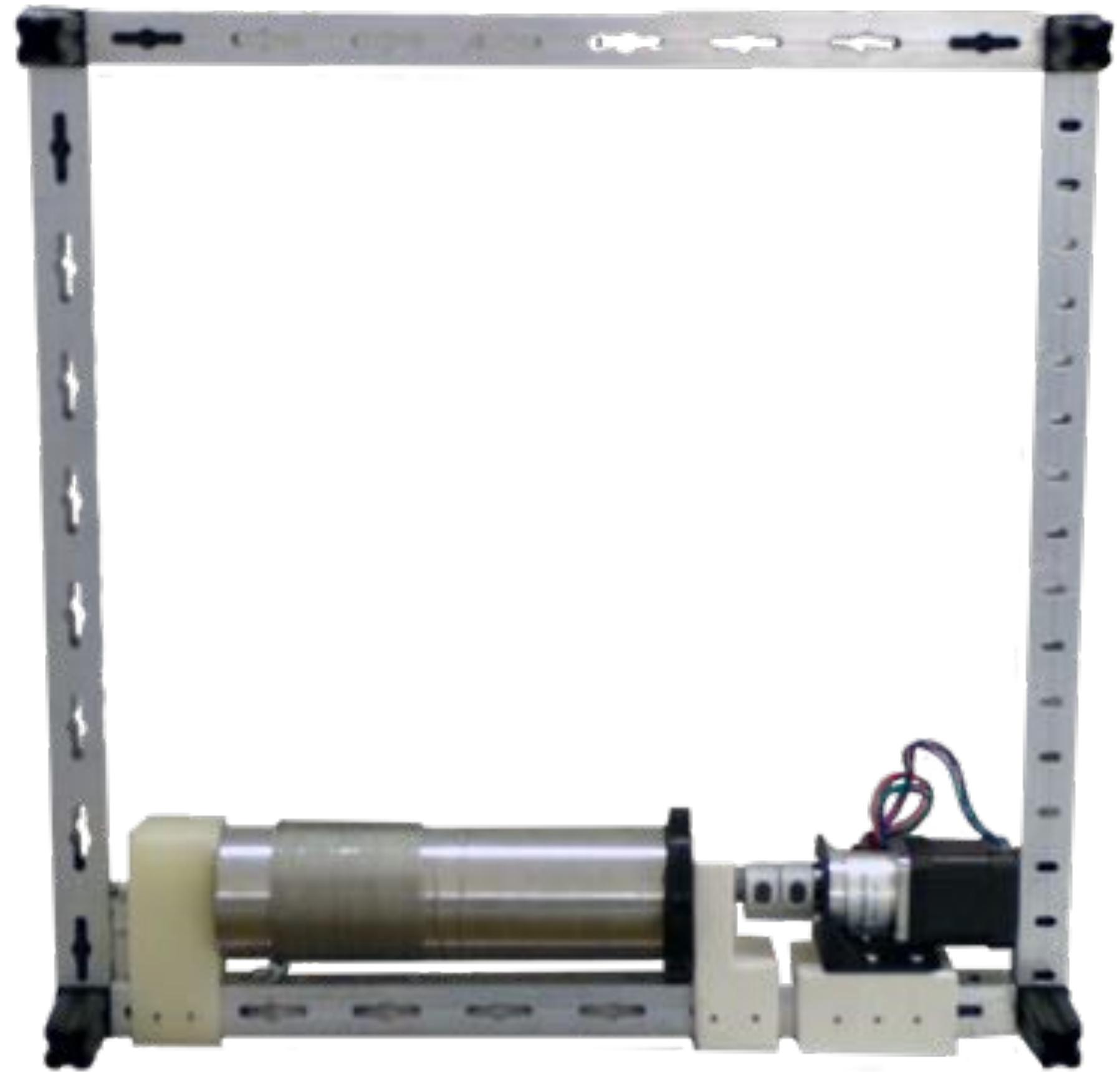
- Treuil

Bloc central unique, regroupant tout le système avec renvois



# ROBOTIQUE

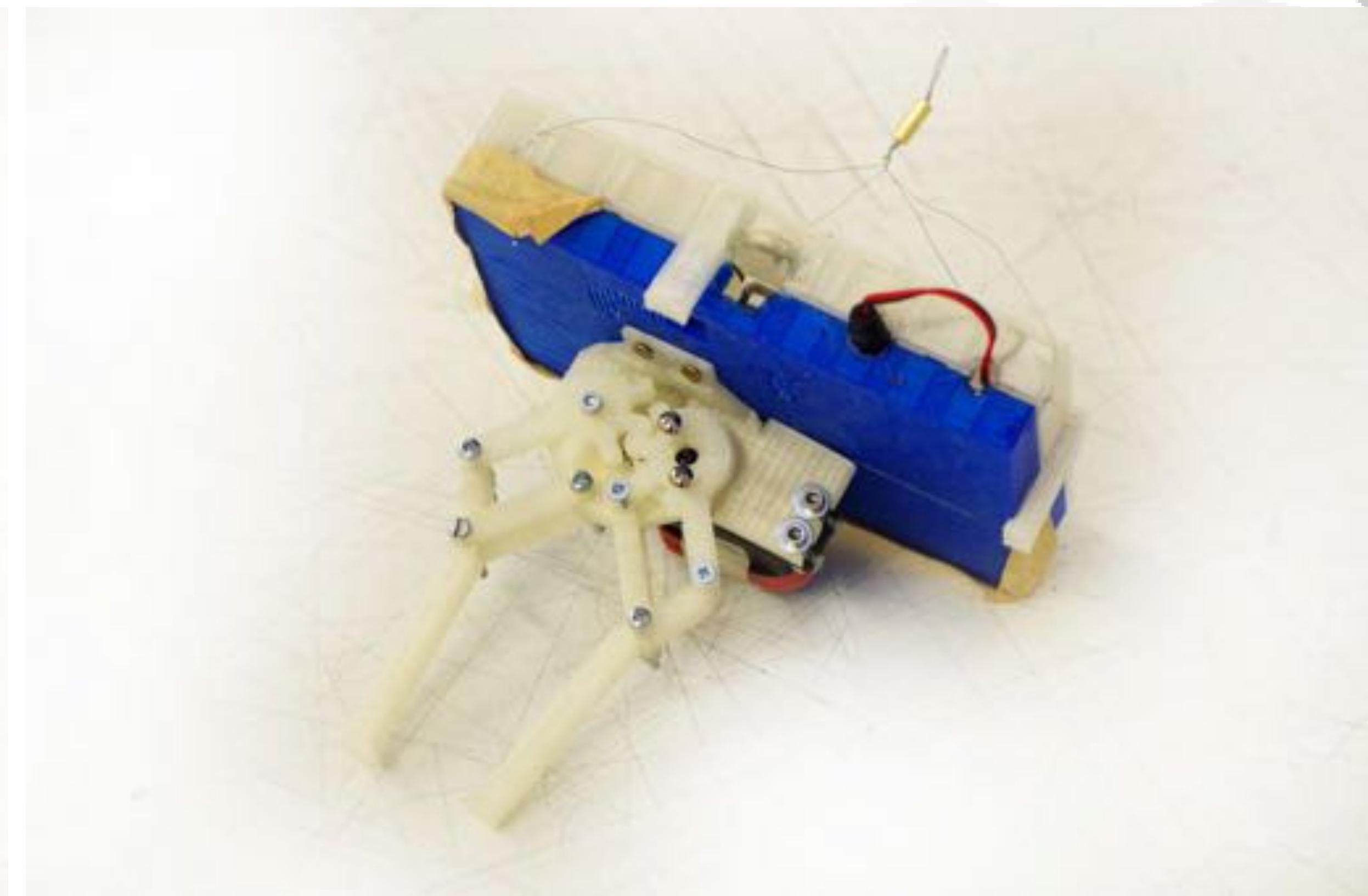
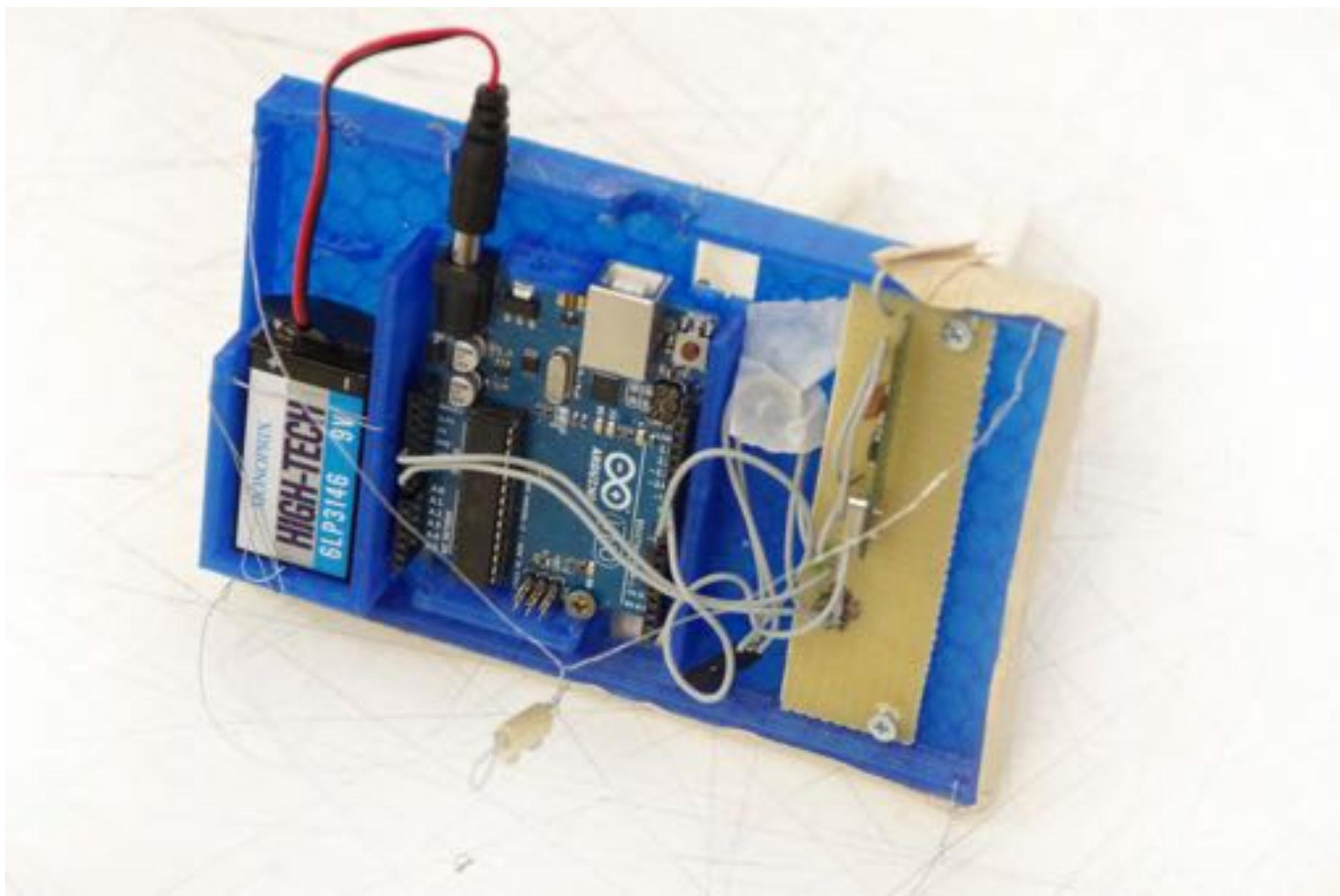
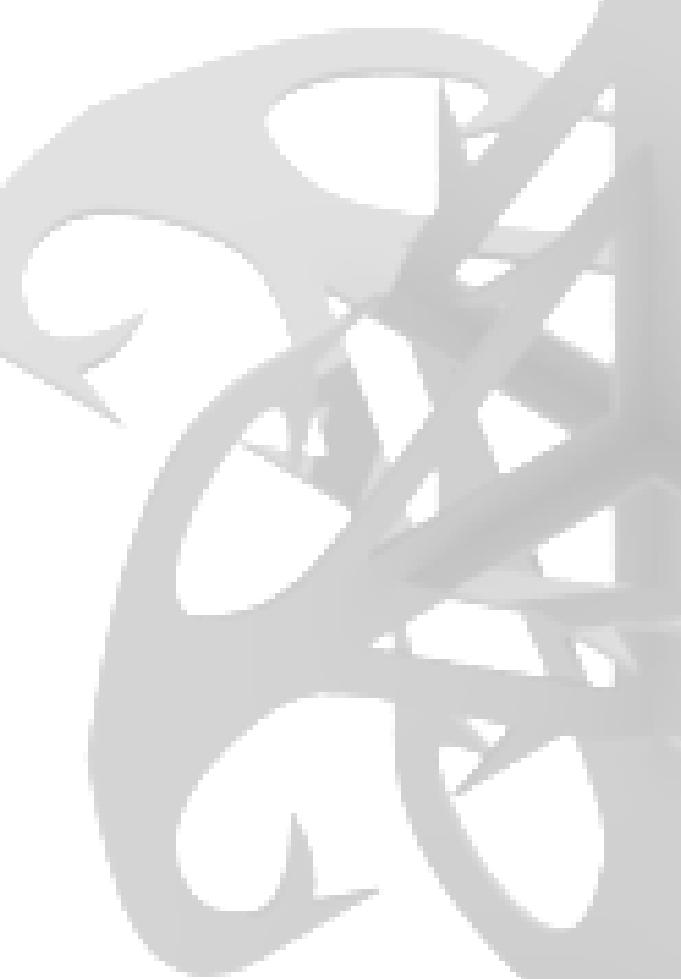
## Moteur et enrouleur



WormRope v5

# ROBOTIQUE

## Tête (Gripper v1)

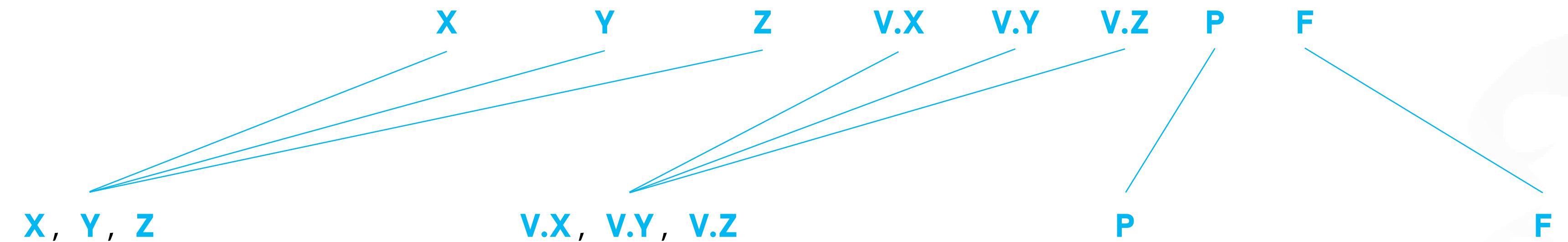


Gripper v1

# ROBOTIQUE

## Instruction type

23156, 29456, 12987, 768, 465, 943, 1, 200



Nombres de pas  
de chaque moteur  
(step)

Vitesses de  
chaque moteur  
(step/s)

Pince  
0 : Ouverture  
1 : Fermeture  
(Boolean)

Fonction  
-2 : Arrêt Urgence  
-1 : Reset Position  
0 - 200 : Pince Délai  
(Int)

# ROBOTIQUE

## Avancement

### Travail validé :

#### // Matériel

- Prototype v4 (Poulies)



#### // Contrôle moteur

- Mouvements
- Accélérations
- Synchronisation
- Arrêt urgence
- Calibration
- Communication Serial (PC)



#### // Contrôle tête

- Radiofréquence
- Servo (Pince)



#### // Interface Graphique (Processing)

- Envoi des instructions depuis fichier texte
- Contrôles basiques (reset, redo, stop, line input)



#### // Grasshopper

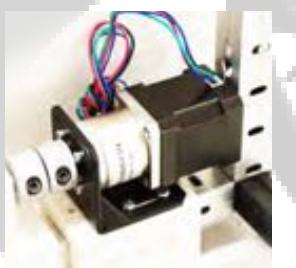
- Export automatique des positions



### Travail en cours :

#### // Matériel

- Test Prototype v5 (Treuil)
- Optimisation du design



#### // Contrôle moteur

- Interrupteurs fin de course
- Compteur cable



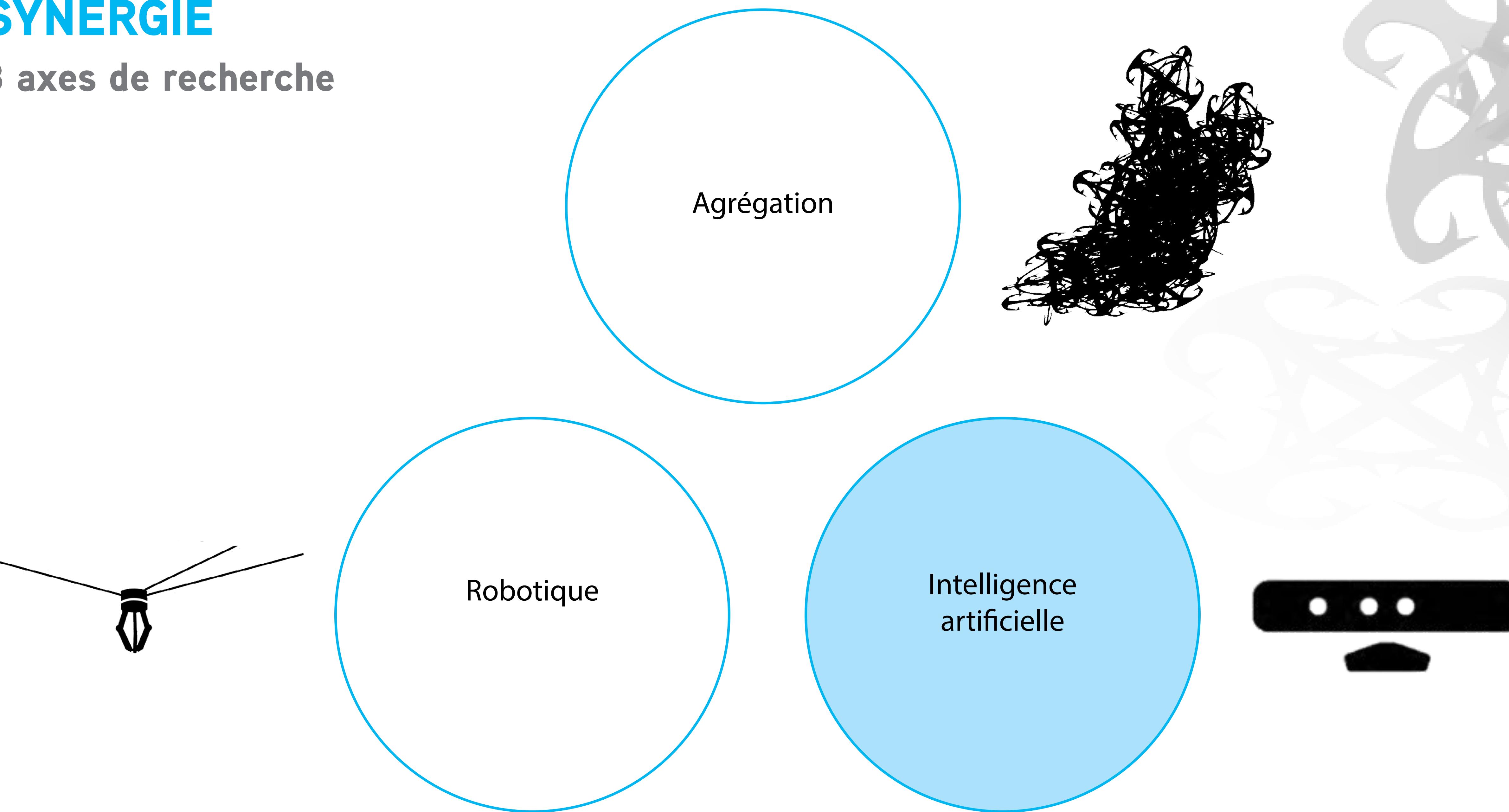
#### // Interface Graphique (Processing)

- Optimisation de l'interface et des contrôles



# SYNERGIE

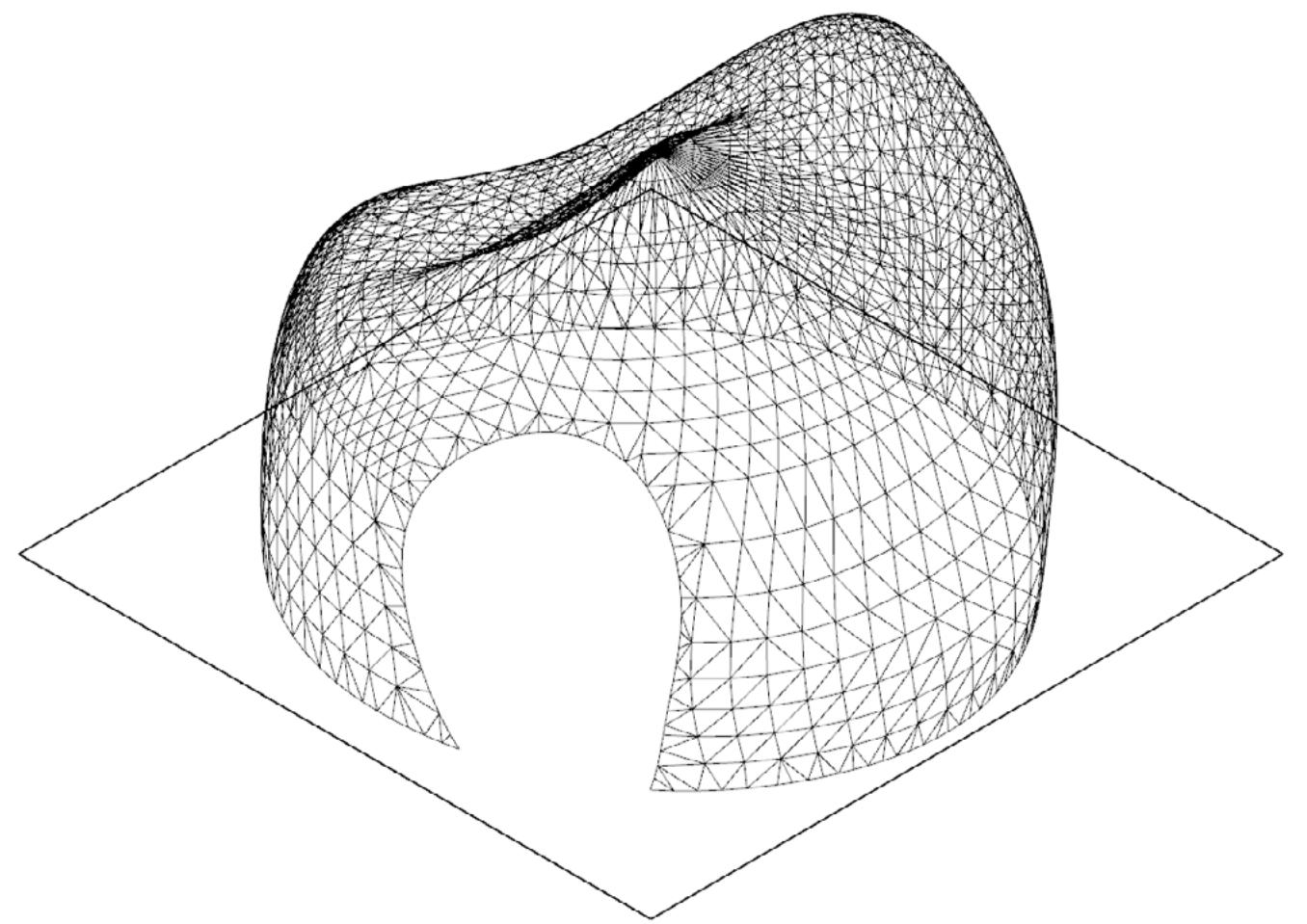
3 axes de recherche



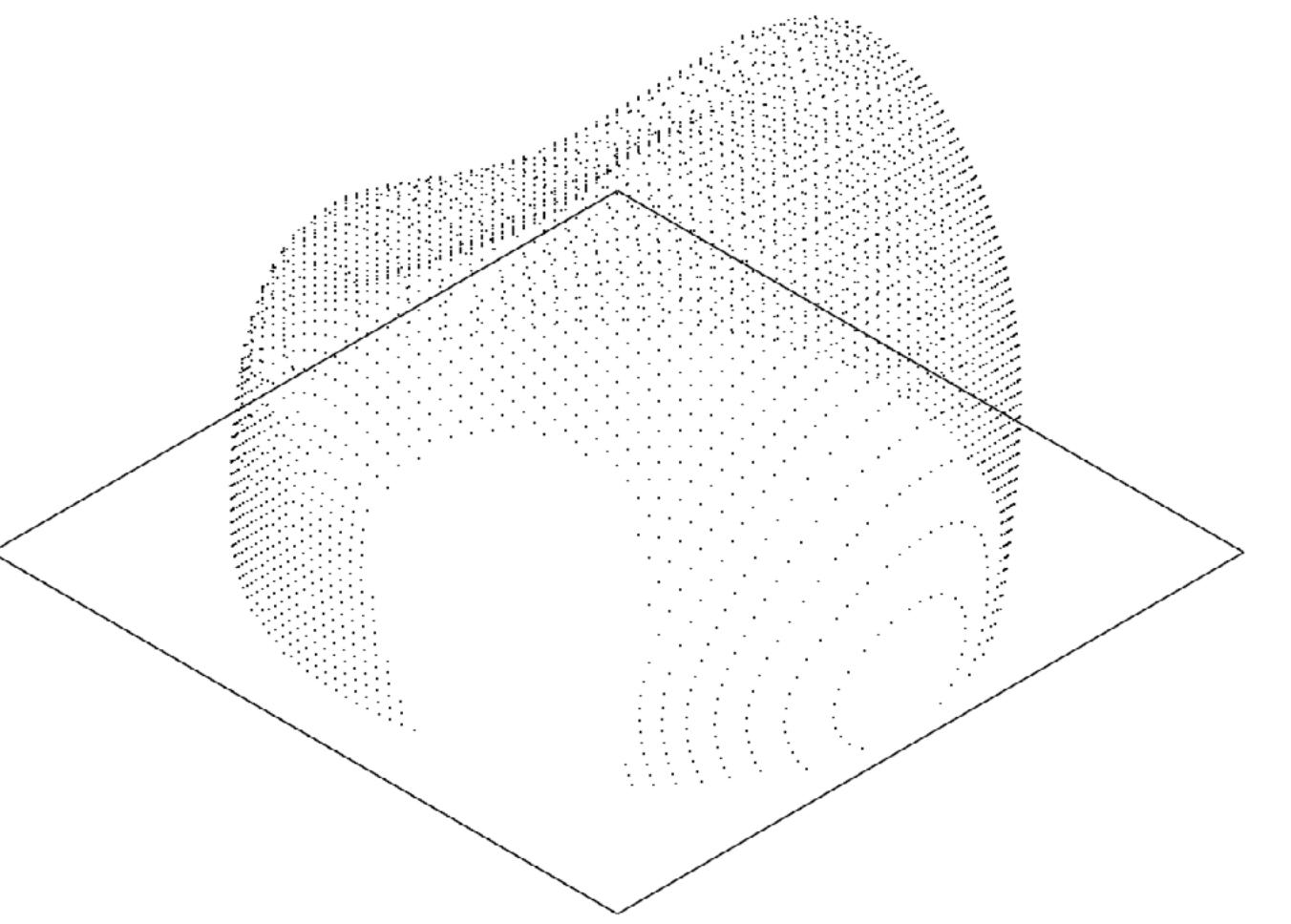
# INTELLIGENCE ARTIFICIELLE

## Scénario simplifié

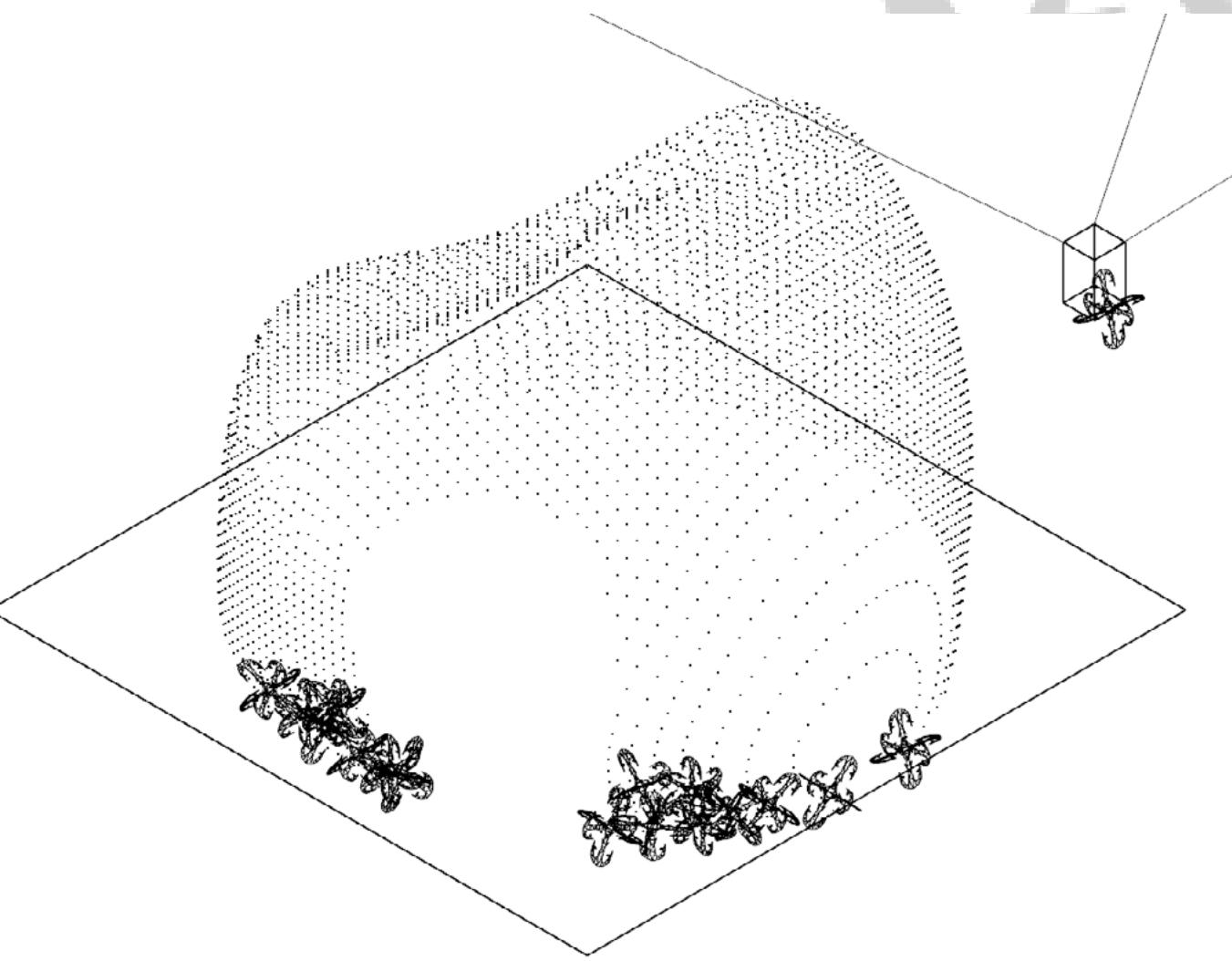
Définition d'un scenario de fonctionnement du robot sans intelligence artificielle.



1. Définition d'une géométrie



2. Discretisation de la géométrie en points



3. Dépose des modules (un par point)

Problèmes soulevés :

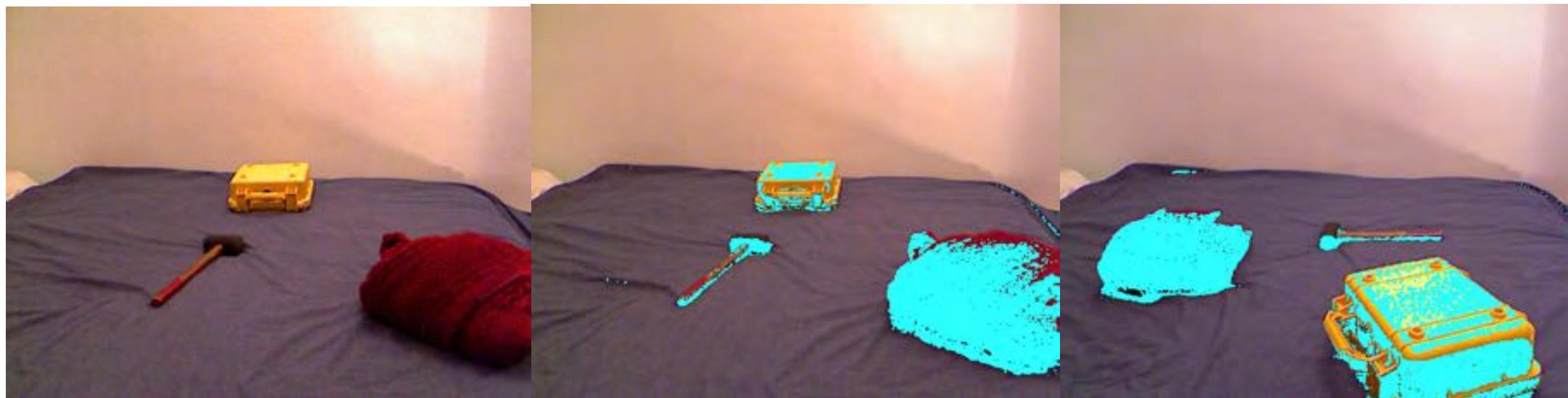
- Rien ne garantit qu'un module tombera à l'emplacement définit (rebonds) ni qu'il y restera (chute, basculement).
- Difficile de prévoir à l'avance la densité du nuage de points qu'il faudra exporter.  
(trop dense = surabondance de modules ; trop peu dense = la structure formée peu devenir incohérente)

# INTELLIGENCE ARTIFICIELLE

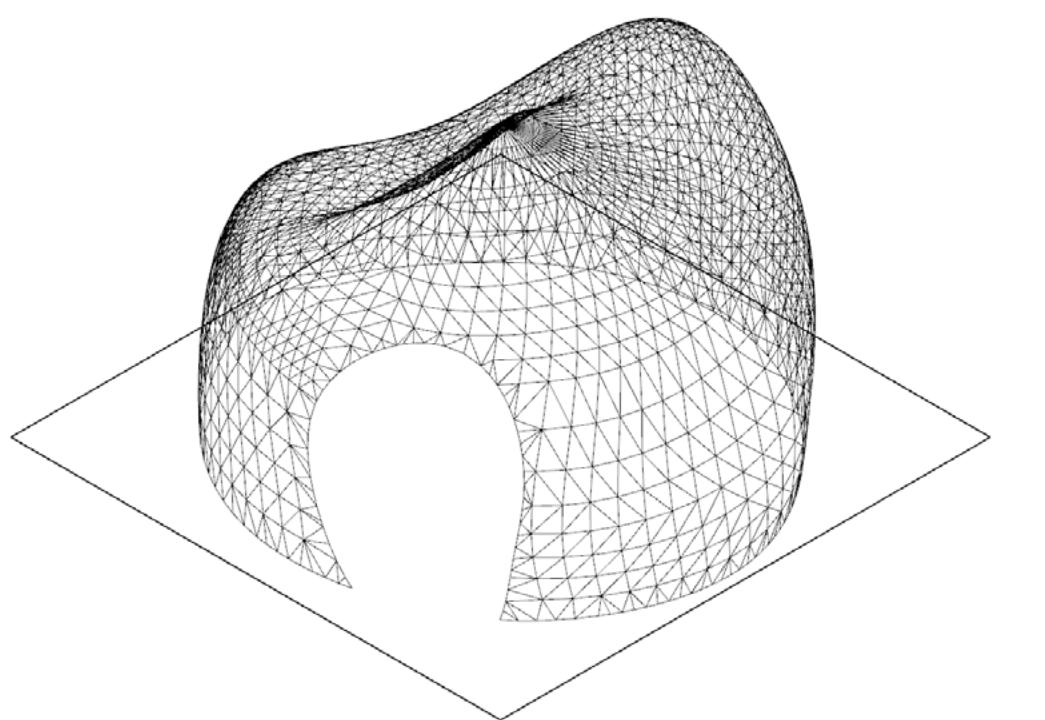
## Vision machine : nuage de point

A l'aide d'une ou de plusieurs Kinects, il est possible de capturer une image "améliorée" des modules déposés (couleurs + distances), puis d'isoler certains pixels en fonction de leur teinte. Le nuage de points obtenu reprend la forme globale de l'agrégation (il est impossible d'obtenir un point par module dans ce cas).

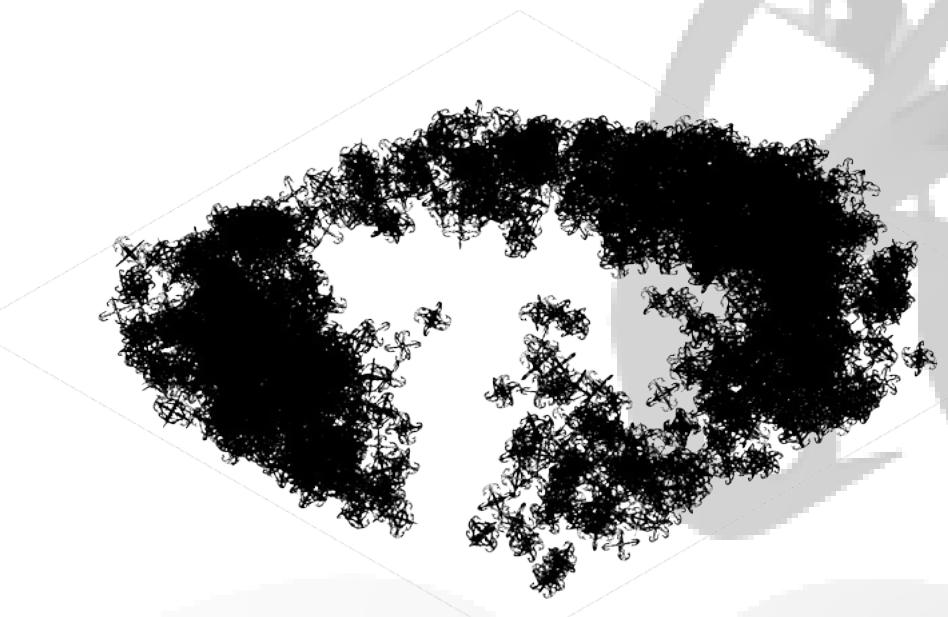
Comment exploiter ce nouveau nuage de points de taille indéfinie pour corriger les erreurs rapportées à notre premier nuage de points ?



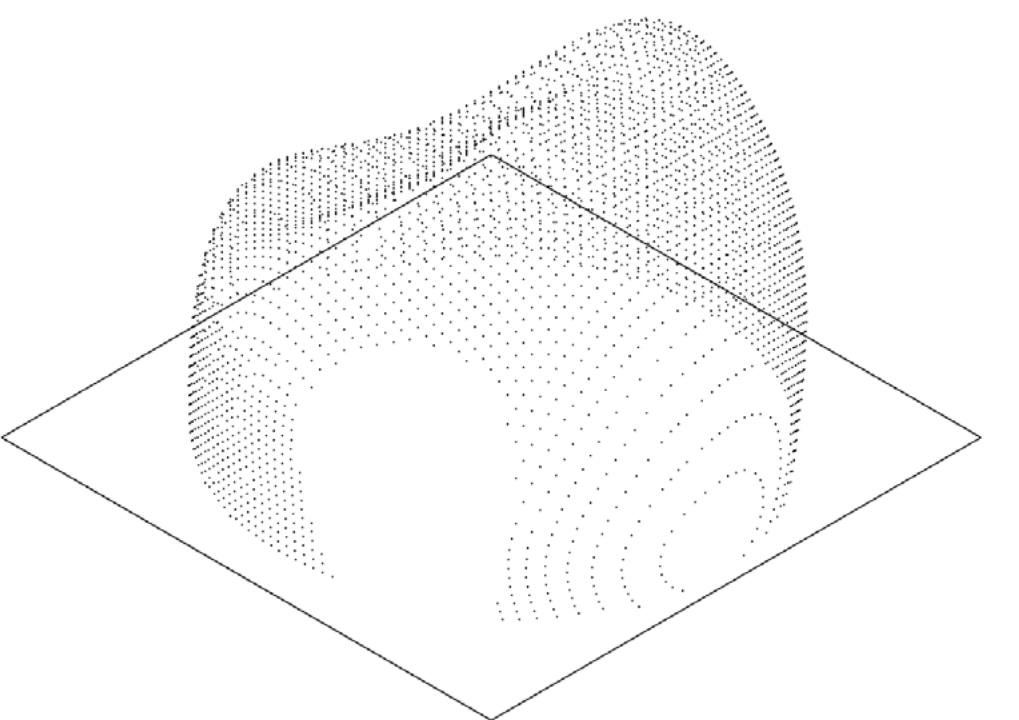
Géométrie en input



Dépose réelle des modules



Discretisation



Vision machine



Nuage de points initial

Nuage de points scanné

# INTELLIGENCE ARTIFICIELLE

## Vision machine : détection de blob

A l'inverse, cet autre type d'analyse d'image nous permet de détecter un ensemble ou un groupe cohérent, un module isolé pourra ainsi être reconnu puis saisi par le robot.

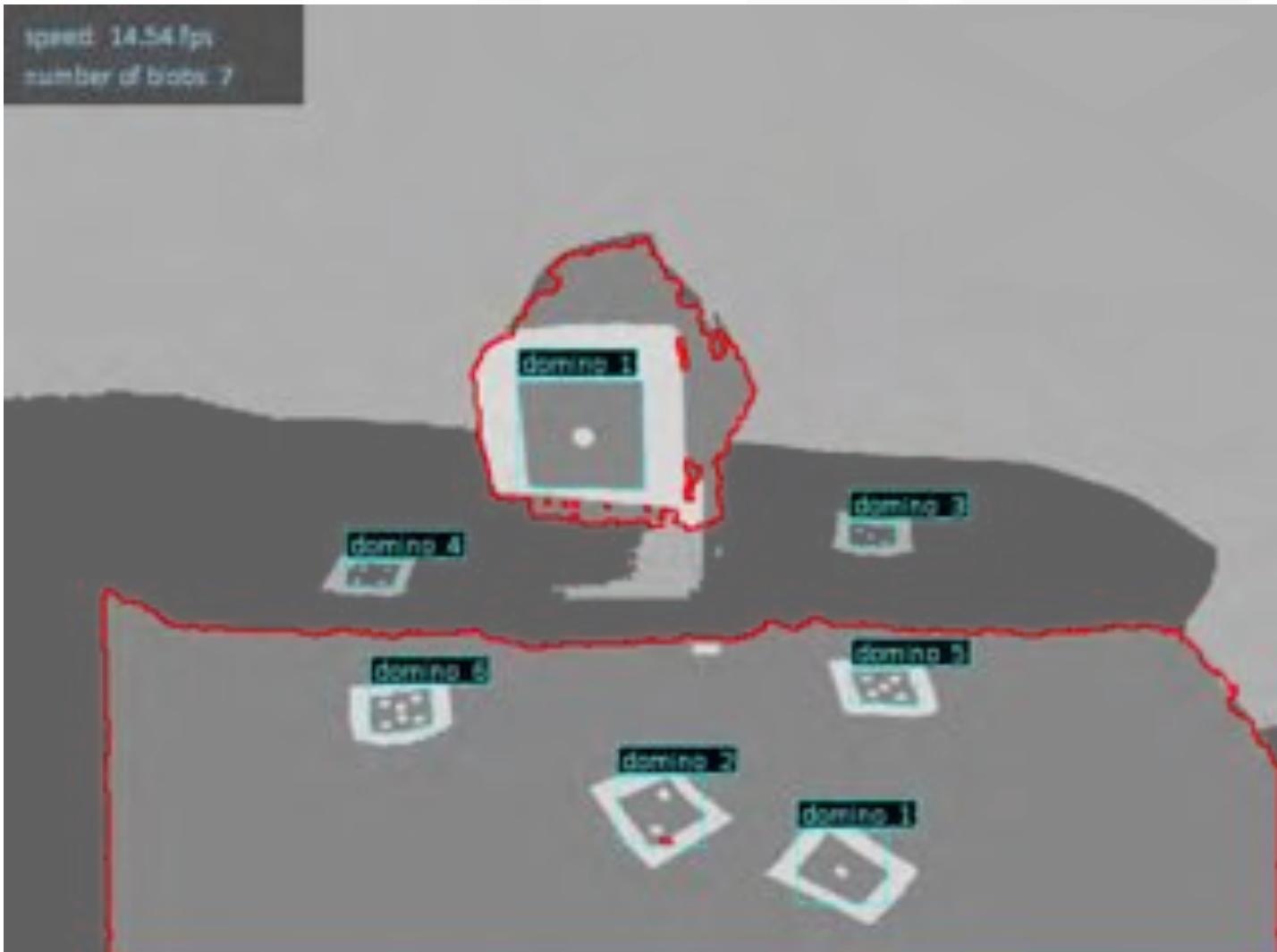
Ce traitement est parfois complété par l'ajout de marqueurs sur les différents éléments.



Multiple Markers tracking



Multiple Blobs Distinction



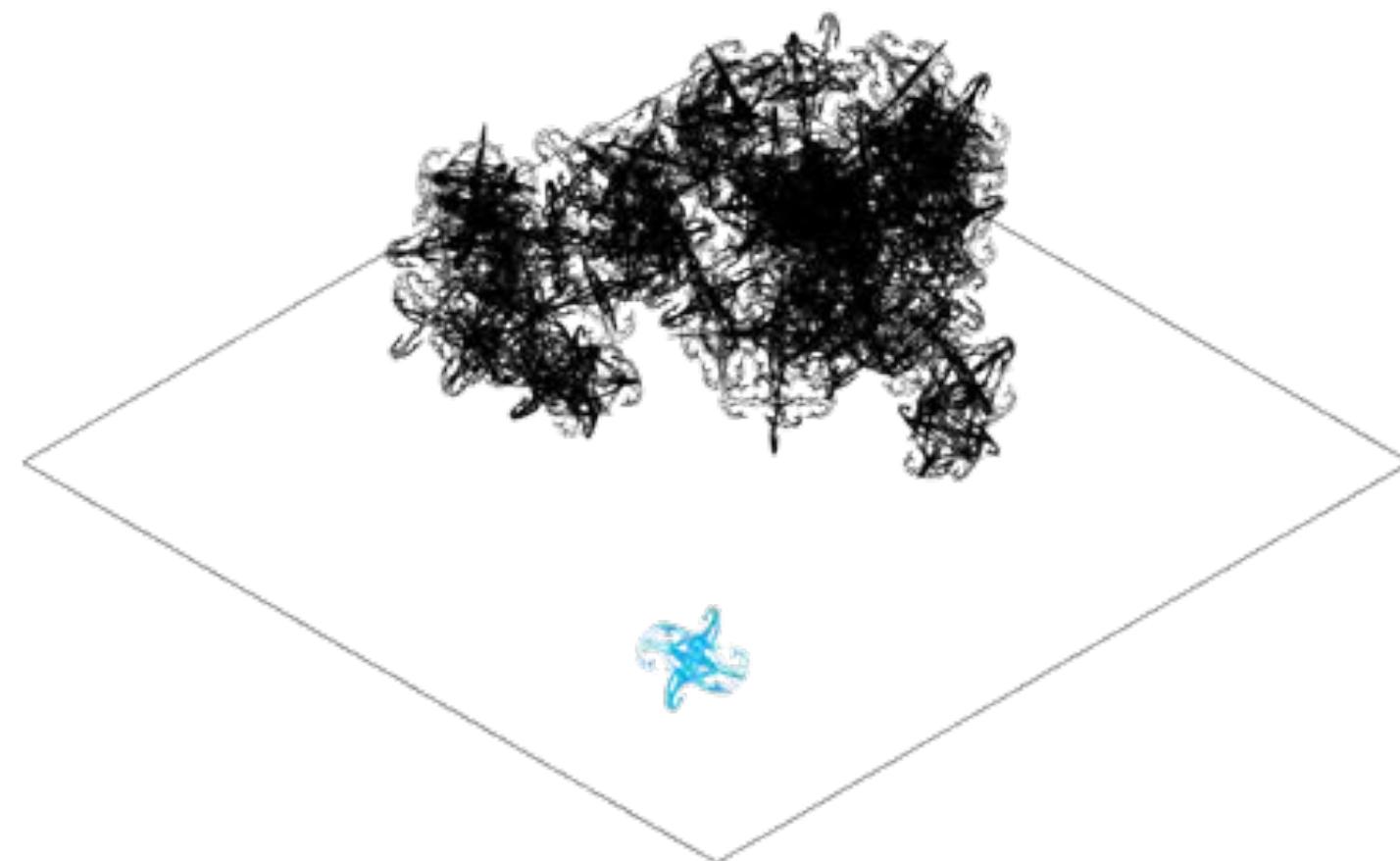
Multiple Markers tracking  
and Blob Distinction

# INTELLIGENCE ARTIFICIELLE

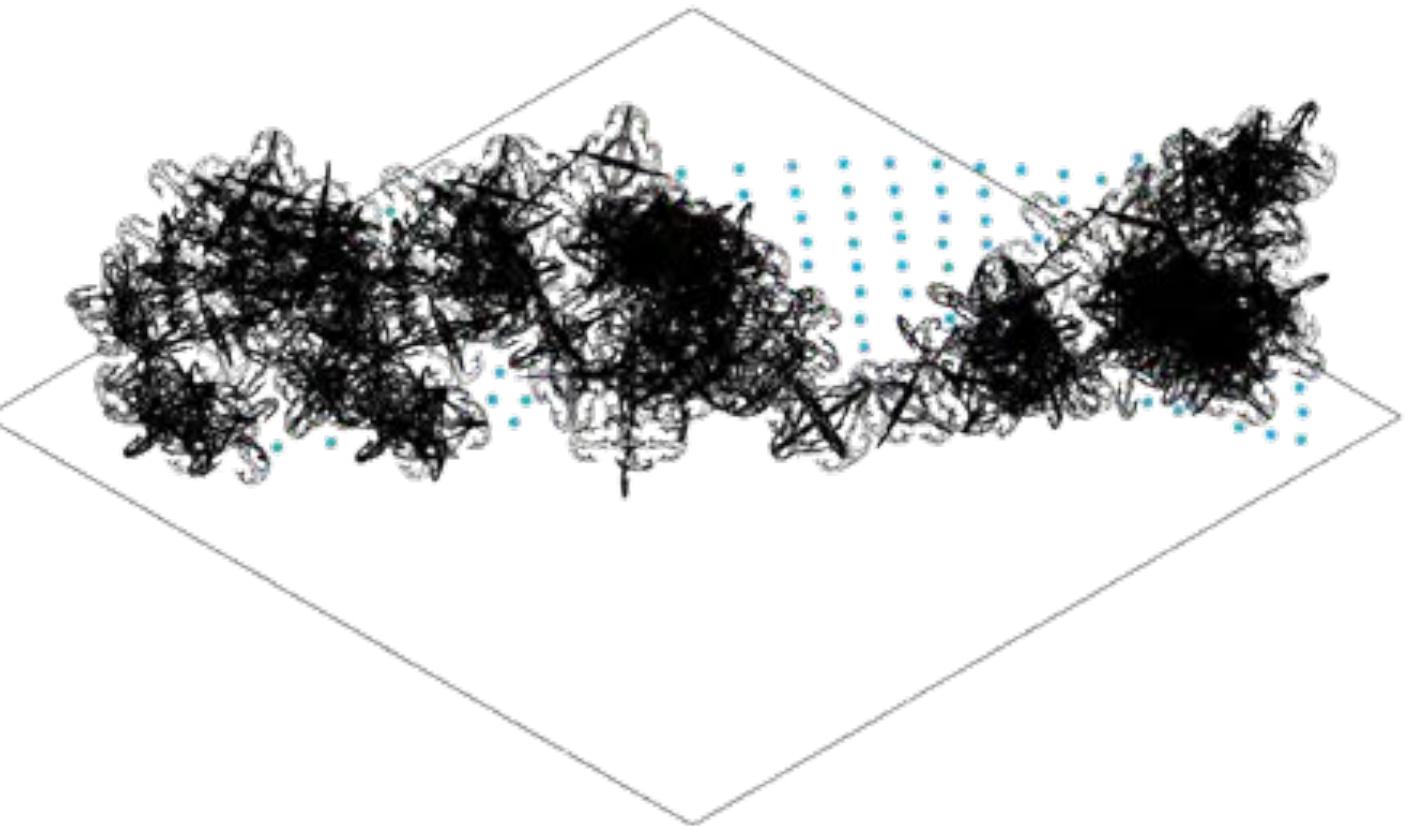
## Identification des différents cas de figure

La simulation nous a permis d'identifier trois types d'erreurs :

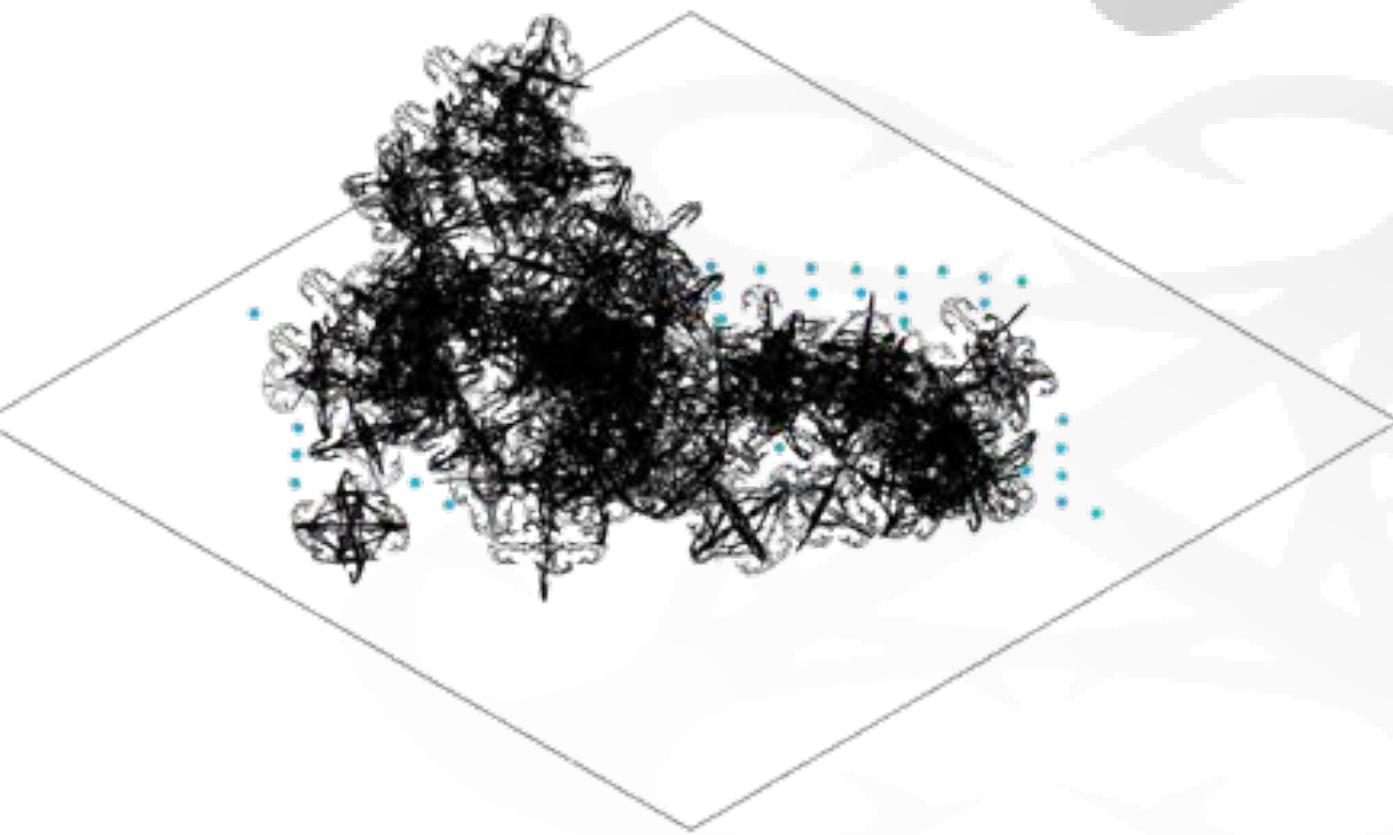
Module tombé trop loin de l'agrégat



Manque de densité à un endroit de l'agrégat



Trop de densité en un point de l'agrégat



Cause :

Comportement physique des modules

Chute de modules ou densité trop faible du nuage initial

Densité trop importante du nuage initial

Mode de détection :

Détection de blob (permet de détecter un élément isolé)

Comparaison des densités des deux nuages de points

Comparaison des densités des deux nuages de points

Réaction :

Si la position des câbles du robot le permettent, saisit  
du module pour réutilisation

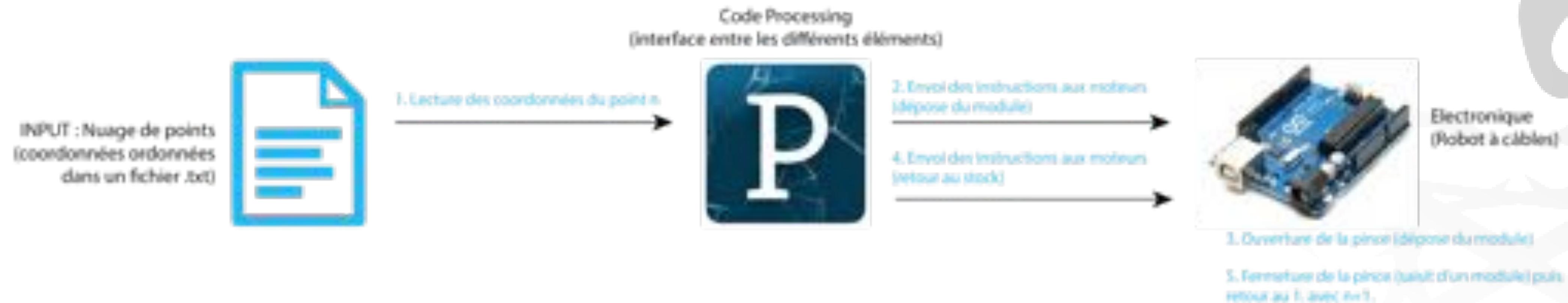
Ajout de modules à cet endroit pour rééquilibrer la  
distribution de l'agrégat

Déplacement ou basculement de la pile de modules  
surélevés

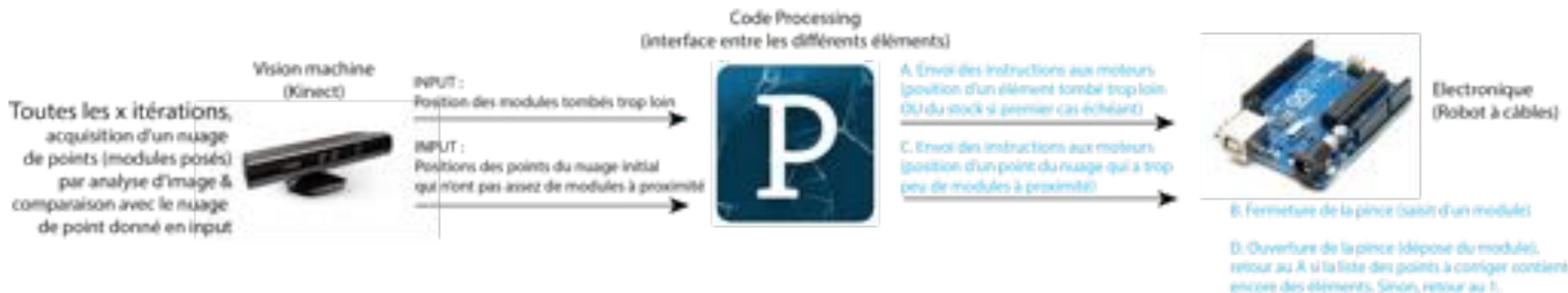
# INTELLIGENCE ARTIFICIELLE

## Scénario complet

### Phase de dépose



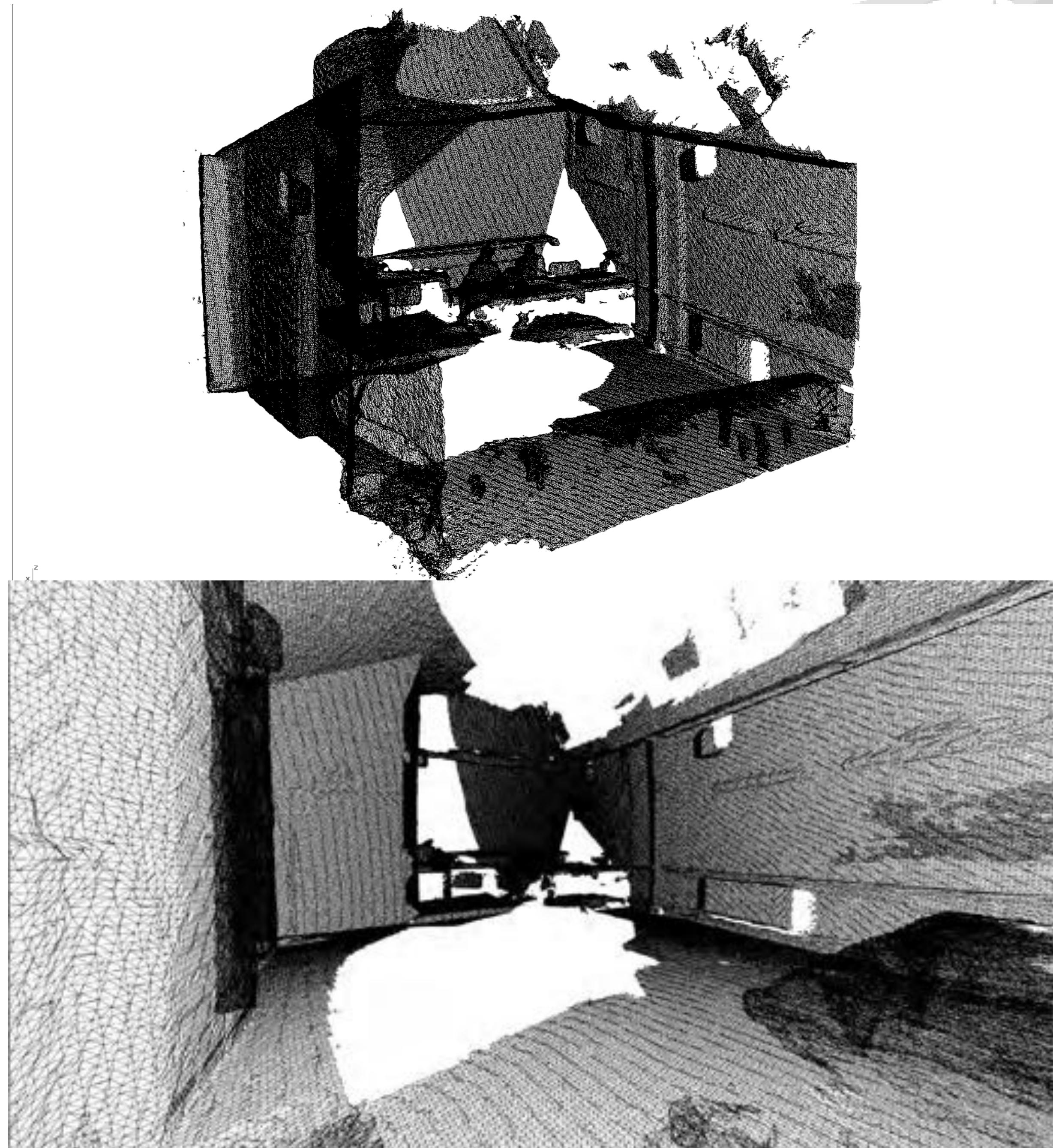
### Phase de correction



# INTELLIGENCE ARTIFICIELLE

## Scan 3D de la salle

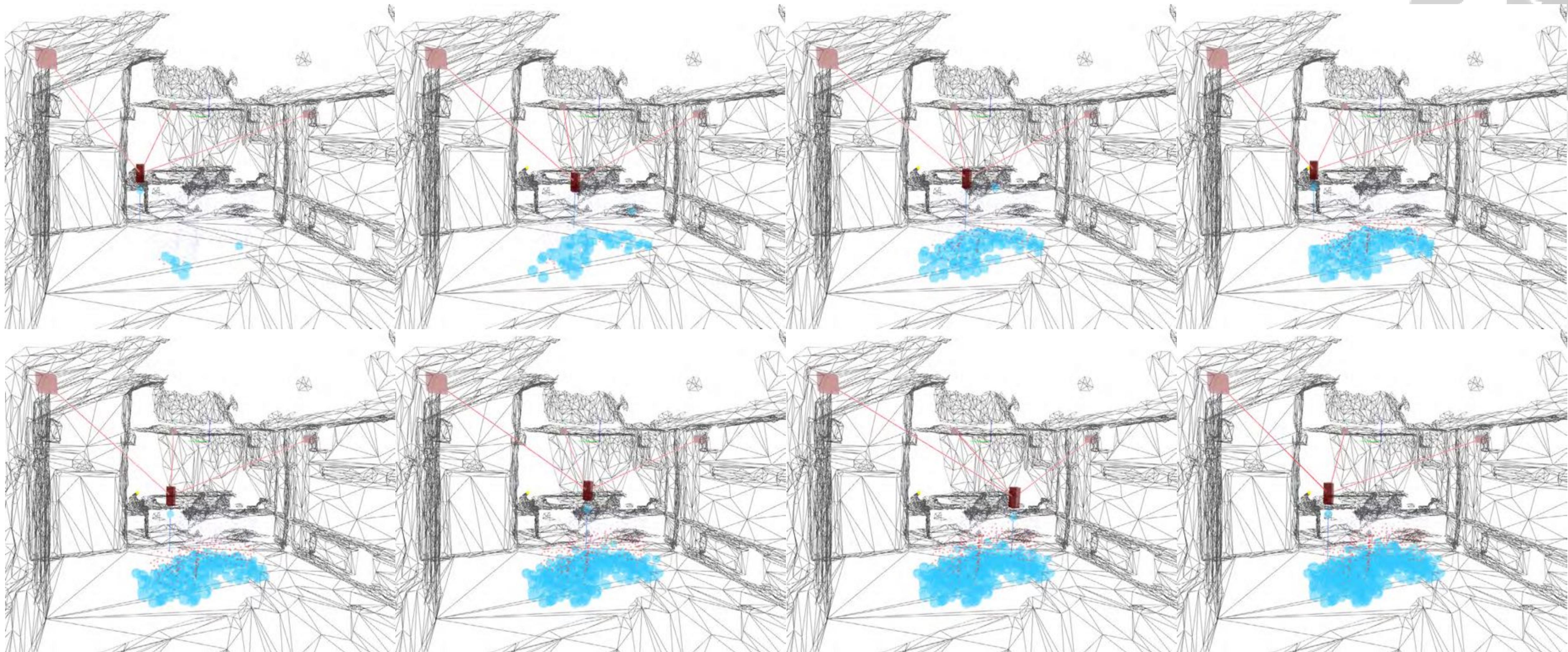
Prend en compte l'environnement, permet de positionner la géométrie précisément dans l'espace.



# INTELLIGENCE ARTIFICIELLE

## Simulation du scénario complet

Comportement du robot réaliste et intégration de l'environnement.



# **Test Final**

**Realisation du scénario complet**



# TO BE CONTINUED

.....

