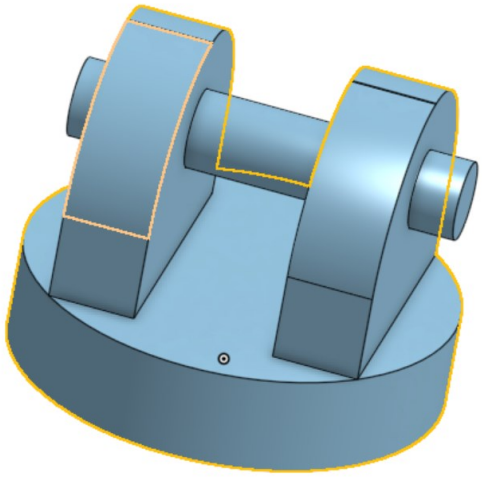
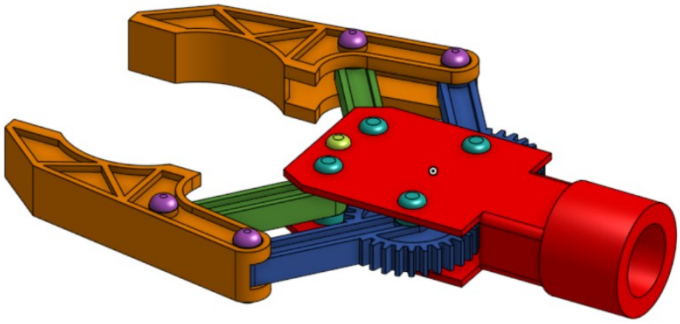
Rapport de séance 13/12/21

Partie modélisation :

Pour la modélisation du bras, nous utilisons le logiciel OnShape sur lequel nous avons commencé à reflechir sur les liaisons possible. La base va être motorisée par un servo principal, elle reposera sur une butée à bille. Un axe viendra se placer entre la base et le bras, qui seront reliés par un goupille. L'axe sera motorisé par un servo. Meme principe pour les autres articulations. La pince sera controlé par un servo placé sur le dessous d'un engrenage qui entrainera toute la pince



Base et axe

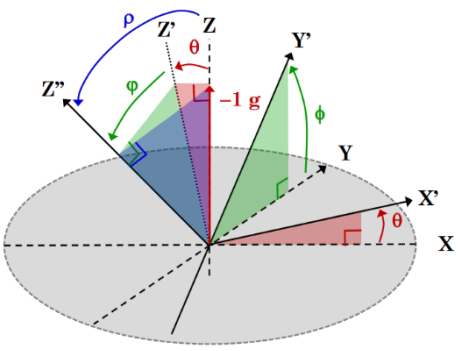


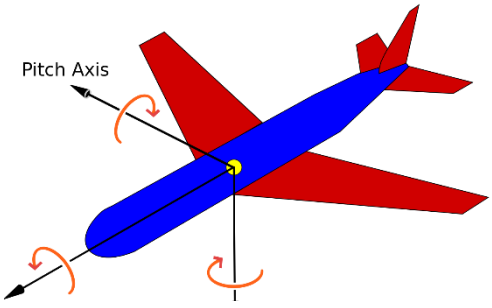
Pince

Partie Arduino :

Côté Mathis : Il a programmé tout le circuit concernant la résistance flexible. Cette resistance servira à diriger la pince à l'aide du gant. Suivant les differentes valeurs de cette dernière, le servo moteur controlant la pince va plus ou moins « serrer » l'objet. Des problèmes ont été rencontrés notamment pas de trop faibles valeurs de la résistance vers le servo. Pour cela il a été judicieux, qu'il change le map pour passer de 0 à 1023 vers 600 à 900. Ainsi on augmente la sensibilité. c.f. rapport Mathis.

De mon côté je me suis penché vers la compréhension de l'accéleromètre à l'aide du cours de M.Masson. J'ai donc compris comment marche un accélèromètre sur 3 axe et comment communiquer avec.





1.2 Tangage Roulis Lacet Schéma

Ainsi j'ai programmé l'accélèromètre pour qu'il donne une valeur sur les 3 axes. Avec l'aide de Mathis, nous avons relié un servomoteur à l'accelerometre pour qu'il tourne suivant les valeurs sur l'axe z : mvt haut bas de la main.

accelerometre

```
#include<Wire.h>
#include<Servo.h>

Servo servo1 ;
const int MPU=0x68;
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

void setup(){
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
  servo1.attach(9);
  servo1.write(0);
}
void loop(){
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,12,true);
  AcX=Wire.read()<<8|Wire.read();
  AcY=Wire.read()<<8|Wire.read();
  AcZ=Wire.read()<<8|Wire.read();

  Serial.print("Accelerometer: ");
  Serial.print("X = "); Serial.print(AcX);
  Serial.print(" | Y = "); Serial.print(AcY);
  Serial.print(" | Z = "); Serial.println(AcZ);

  servo1.write(map(AcZ,-10000,15000,0,180));
  delay(333);

  /* PARTIE GYROSCOPE
  GyX=Wire.read()<<8|Wire.read();
  GyY=Wire.read()<<8|Wire.read();
  GyZ=Wire.read()<<8|Wire.read();
  Serial.print("Gyroscope: ");
  Serial.print("X = "); Serial.print(GyX);
  Serial.print(" | Y = "); Serial.print(GyY);
  Serial.print(" | Z = "); Serial.println(GyZ);
  Serial.println(" ");
  */

}
```

On inclut les librairies Wire (accélèromètre) et cellle du servo Servo.
On définit l'adresse I_2C du MPU-6050 qui est 0x68 pour communiquer avec la composante.
On indique alors que l'on va communiquer avec l'accéléromètre. Par défaut, le MPU-6050 est en veille et on le réveille en écrivant 0 dans le registre 0x6B. Finalement on arrête la communication I_2C pour libérer le BUS.

On indique alors que l'on va communiquer avec l'accéléromètre. Par défaut, le MPU-6050 est en veille et on le réveille en écrivant 0 dans le registre 0x6B.
Finalement on arrête la communication I_2C pour libérer le BUS.
On demande l'accès au premier registre de la liste. On indique alors à l'accéléromètre que l'on va lire 12 octets (requestFrom)

Les problèmes rencontrés lors de cette séance ont surtout été de comprendre le fonctionnement de l'accélèromètre. Il faudra aussi programmé l'axe y qui permettra une rotation sur elle-même de la main. Peut être rajouter un condensateur qui permettrait d'éviter les parasites.

Prochaine séance :

Avoir les plans et la modélisation terminée pour commencer à s'occuper de la partie construction grâce au FABLAB.