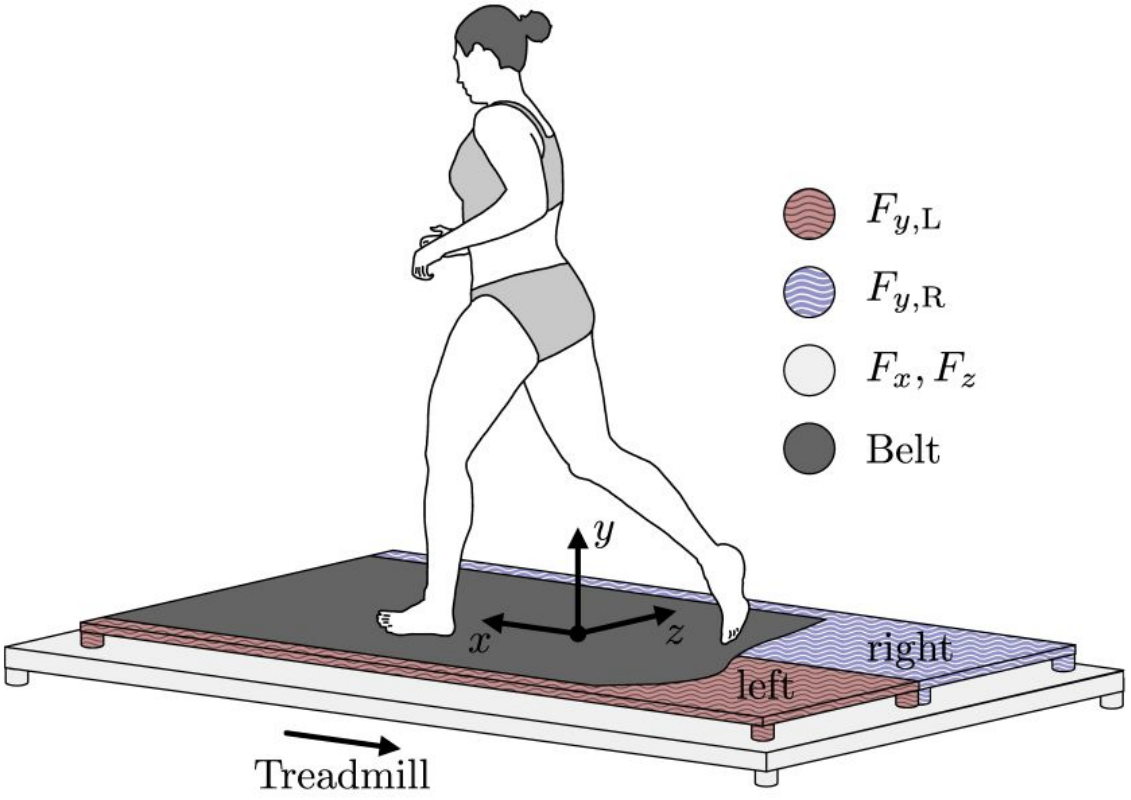
Rapport interne - Simulateur ROBIBIO

Ce document accompagne le simulateur 2D pour le calcul du positionnement des moteurs dans le cadre du projet ROBIBIO.

1. Hypothèses de travail

1.1. Repère et définition des axes

Le repère général considéré est celui utilisé dans les données HuMoD. Il présente l'avantage et rester direct et intuitif, même après projection en 2D :



Définition générale du repère 3D

1.2. Paramètres du modèle

Le modèle HuMod est une femme dont les paramètres (longueur des segments) ont été estimés à partir des données capturées. Cette approximation produit un jeu de paramètres qui n'est pas symétrique entre les parties gauche et droite du corps. Dans le cadre de la transposition à un système robotisé, il semble préférable de corriger ce jeu de données afin de le rendre symétrique. Ce post-traitement a été réalisé en deux étapes :

- calcul de la moyenne des segments droit et gauche pour chaque paramètre
- calcul de l'arrondi à 0.5mm.

Les paramètres du modèle considérés pour la simulation 2D sont les suivants :

	X [mm]	Y [mm]
Cuisse	0	-380
Mollet	0	-358,5
Pied	121	-54

Ces paramètres sont reportés dans chacun des matrices du dossier `/dataset/robibio/`.

2. Structure de l'archive

- **`dataset`** : dossier contenant les jeux de données relatifs aux mouvements
    - **`humod-data`** : dossier contenant les données HuMoD
        - **`1.1.mat`** : marche à 1m/s
        - **`1.2.mat`** : marche à 1,5m/s
        - **`1.3.mat`** : marche à 2m/s
        - **`2.1.mat`** : course à 2m/s
        - **`2.2.mat`** : course à 3m/s
        - **`2.3.mat`** : course à 4m/s
        - **`3.mat`** : pas de côté à 3m/s
        - **`4.mat`** : accélération de 0 à 4m/s (course)
        - **`5.1.mat`** : marche avec évitement d'un obstacle long
        - **`5.2.mat`** : marche avec évitement d'un obstacle large
        - **`6.mat`** : squats
        - **`7.mat`** : tirs dans un ballon

        - **8.mat** : sauts sur place

    - **newton-euler-torques** : dossier contenant les couples calculés par Nicolas Testard avec la méthode de Newton-Euler

        - **1.1_Torques.mat** : couples articulaire pour la marche à 1m/s
        - **1.2_Torques.mat** : couples articulaire pour la marche à 1,5m/s
        - **1.3_Torques.mat** : couples articulaire pour la marche à 2m/s
        - **2.1_Torques.mat** : couples articulaire pour la course à 2m/s
        - **2.2_Torques.mat** : couples articulaire pour la course à 3m/s
        - **2.3_Torques.mat** : couples articulaire pour la course à 4m/s
        - **3_Torques.mat** : couples articulaire pour les pas de côté à 3m/s
        - **4_Torques.mat** : couples articulaire pour l'accélération de 0 à 4m/s (course)
        - **5.1_Torques.mat** : couples articulaire pour la marche avec évitement d'un obstacle long
        - **5.2_Torques.mat** : couples articulaire pour la marche avec évitement d'un obstacle large
        - **6_Torques.mat** : couples articulaire pour les squats
        - **7_Torques.mat** : couples articulaire pour les tirs dans un ballon
        - **8_Torques.mat** : couples articulaire pour les sauts sur place

    - **robibio** : dossier contenant les données 2D préparées pour l'optimisation (couples, positions articulaires, positions angulaires et paramètre du sujet).

        - **1.1.mat** : données 2D préparées pour la marche à 1m/s
        - **1.2.mat** : données 2D préparées pour la marche à 1,5m/s
        - **1.3.mat** : données 2D préparées pour la marche à 2m/s
        - **2.1.mat** : données 2D préparées pour la course à 2m/s
        - **2.2.mat** : données 2D préparées pour la course à 3m/s
        - **2.3.mat** : données 2D préparées pour la course à 4m/s
        - **3.mat** : données 2D préparées pour les pas de côté à 3m/s
        - **4.mat** : données 2D préparées pour l'accélération de 0 à 4m/s (course)
        - **5.1.mat** : données 2D préparées pour la marche avec évitement d'un obstacle long
        - **5.2.mat** : données 2D préparées pour la marche avec évitement d'un obstacle large
        - **6.mat** : données 2D préparées pour les squats
        - **7.mat** : données 2D préparées pour les tirs dans un ballon
        - **8.mat** : données 2D préparées pour les sauts sur place

    - **simscape-torques :** dossier contenant les couples calculés par Franck Mercier avec Simscape.

        - **1.1_S_Torques.mat** : couples articulaire pour la marche à 1m/s

        - **`1.2_S_Torques.mat`** : couples articulaire pour la marche à 1,5m/s
        - **`1.3_S_Torques.mat`** : couples articulaire pour la marche à 2m/s
        - **`2.1_S_Torques.mat`** : couples articulaire pour la course à 2m/s
        - **`2.2_S_Torques.mat`** : couples articulaire pour la course à 3m/s
        - **`2.3_S_Torques.mat`** : couples articulaire pour la course à 4m/s
        - **`4_S_Torques.mat`** : couples articulaire pour l'accélération de 0 à 4m/s (course)
        - **`5.1_S_Torques.mat`** : couples articulaire pour la marche avec évitement d'un obstacle long
        - **`5.2_S_Torques.mat`** : couples articulaire pour la marche avec évitement d'un obstacle large

- **`functions`** : dossier contenant les fonctions primaires utilisées dans les scripts Matlab.

    - **`core.m` :** fonction principale du simulateur (simule le robot).
    - **`init_figure_robot.m` :** fonction qui initialise la figure du robot et retourne une structure contenant les objets graphiques.
    - **`motor_P01_48x240_30x240.m` :** fonction qui calcule la force maximal d'un moteur P01 48x240 30x40 en fonction de sa longueur et de son offset.
    - **`motor_P01_48x360F_60x210.m` :** fonction qui calcule la force maximal d'un moteur P01 48x360F 60x120 en fonction de sa longueur et de son offset.
    - **`plot_motor_xxxx.m` :** fonctions qui affichent les données calculées (couples, forces, état des moteurs …) à l'issue d'une simulation.
    - **`rot_x.m` :** fonction qui retourne la matrice de rotation autour de l'axe X pour un angle donné.
    - **`rot_y.m` :** fonction qui retourne la matrice de rotation autour de l'axe Y pour un angle donné.
    - **`rot_z.m` :** fonction qui retourne la matrice de rotation autour de l'axe Z pour un angle donné.
    - **`tra_x.m` :** fonction qui retourne la matrice de translation le long de l'axe X pour une longueur donnée.
    - **`tra_y.m` :** fonction qui retourne la matrice de translation le long de l'axe X pour une longueur donnée.
    - **`tra_z.m` :** fonction qui retourne la matrice de translation le long de l'axe X pour une longueur donnée.
    - **`update_figure_robot.m` :** fonction qui actualise la figure du robot créer avec `init_figure_robot.`

- **`prepare_2d_dataset`** : dossier contenant le code pour convertir les données HuMoD en 2D et agréger les données supplémentaires.

    - **`prepare_2d_dataset.m`** : script de prépartion des données 2D. Ce script écrit les données préparées dans le dossier `/dataset/robibio/`.
    - **`prepare_2d_dataset.m`** : script de prépartion des données 2D. Ce script écrit les données préparées dans le dossier `/dataset/robibio/`.

- **`unit_test`** : dossier contenant les codes de tests unitaires.

3. Jeux de données

Les jeux de données dans le dossier `/dataset/robibio/` sont des fichiers MATLAB structurés de la façon suivante :

- **`trajectories`** : contient les trajectoires du mouvement
- **`torques`** : contient les couples articulaires
- **`power`** : contient les puissances articulaires et leur somme.
- **`transformation_matrices`** : contient les matrices de transformation de la hanche vers une articulation donnée
- **`timestamp`** : libellé temporel des échantillons dans les données HuMoD.
- **`frame_rate`** : nombre d'échantillons par seconde
- **`frames`** : nombre d'échantillons total
- **`parameters`** : contient les paramètres du sujet (longueur des segments)
- **`labels`** : contient l'ordre des articulation du jeu de données
- **`title`** : nom du mouvement
- **`id`** : numéro du mouvement

3.1 Trajectoires

La structure **`trajectories`** contient les trajectoires du mouvement.

- **`x`** : position cartésienne en x de chaque articulation en millimètres
- **`x`** : position cartésienne en y de chaque articulation en millimètres
- **`q`** : position angulaire de chaque articulation en radians
- **`dqdt`** : vitesse angulaire de chaque articulation en radians/seconde
- **`ddqddt`** : accélération angulaire de chaque articulation en radians/seconde².

L'ordre des articulations est donné dans la structure **`labels`**.

3.2 Couples

La structure **`torques`** contient les couples articulaires :

- **`q`** : couple de chaque articulation en Newtons.mètres

L'ordre des articulations est donné dans la structure **`labels.`**

3.3 Puissances

La structure **`power`** contient les puissances articulaires et la puissance instantanée globale :

- **`q`** : puissance développée pour chaque articulation en Watts.
- **`total`** : puissance globale instantanée, somme des puissance articulaires.

L'ordre des articulations est donné dans la structure **`labels.`**

3.4 Matrices de transformation

La structure **`transformation_matrices`** contient les matrices de transformation du repère origine (la hanche) vers chacune des articulations :

- **`hip_to_knee_left`** : matrice de transformation de la hanche vers le genou gauche
- **`hip_to_knee_right`** : matrice de transformation de la hanche vers le genou droit
- **`hip_to_ankle_left`** : matrice de transformation de la hanche vers la cheville gauche
- **`hip_to_ankle_right`** : matrice de transformation de la hanche vers la cheville droite
- **`hip_to_foot_left`** : matrice de transformation de la hanche vers le pied gauche
- **`hip_to_foot_right`** : matrice de transformation de la hanche vers le pied droit

Pour être stockées, les matrices ont été arrangées en vecteurs colonne. La commande suivante permet de les reconvertir en matrices carrées :

```
reshape (dataset.transformation_matrices.hip_to_knee_left(:,1), [4,4])
```

3.5 Timestamp

Libellé temporel des échantillons dans l'espace HuMoD.

3.6 Frame rate

Fréquence d'échantillonnage des données. Il s'agit d'un paramètre commun aux données HuMoD et ROBIBIO.

3.7 Frames

Nombre d'échantillon dans le jeu de données. Ce paramètre est propre aux données ROBIBIO.

3.8 Paramètres

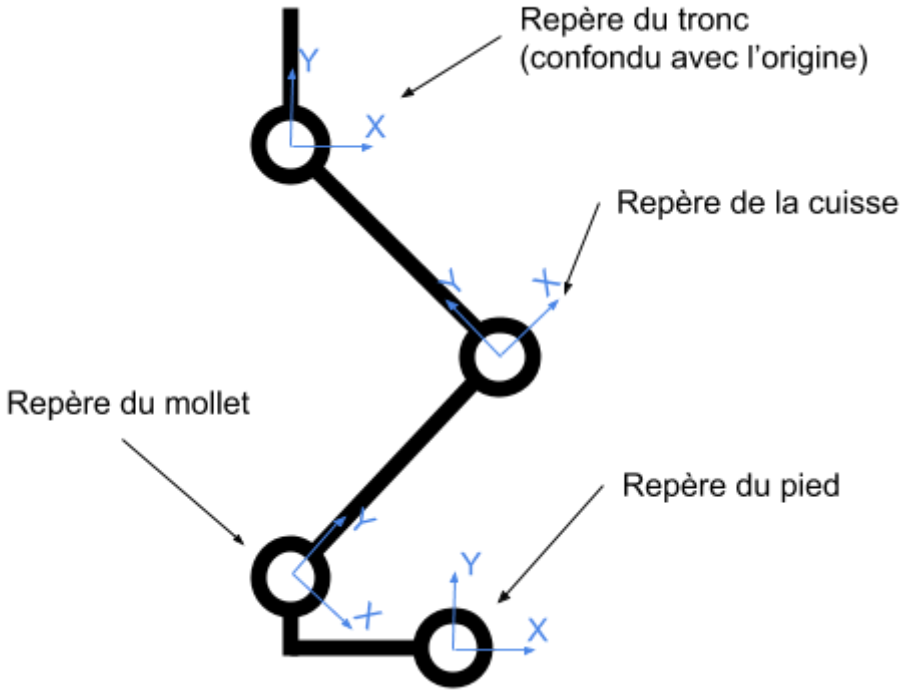
La structure **`parameters`** contient les longueurs des segments du sujet.

- **`thigh`** : paramètres de la cuisse en millimètres [x, y]
- **`shang`** : paramètres du tibia en millimètres [x, y]
- **`foot`** : paramètres du pied en millimètres [x, y]

4. Moteurs

4.1 Repères des segments

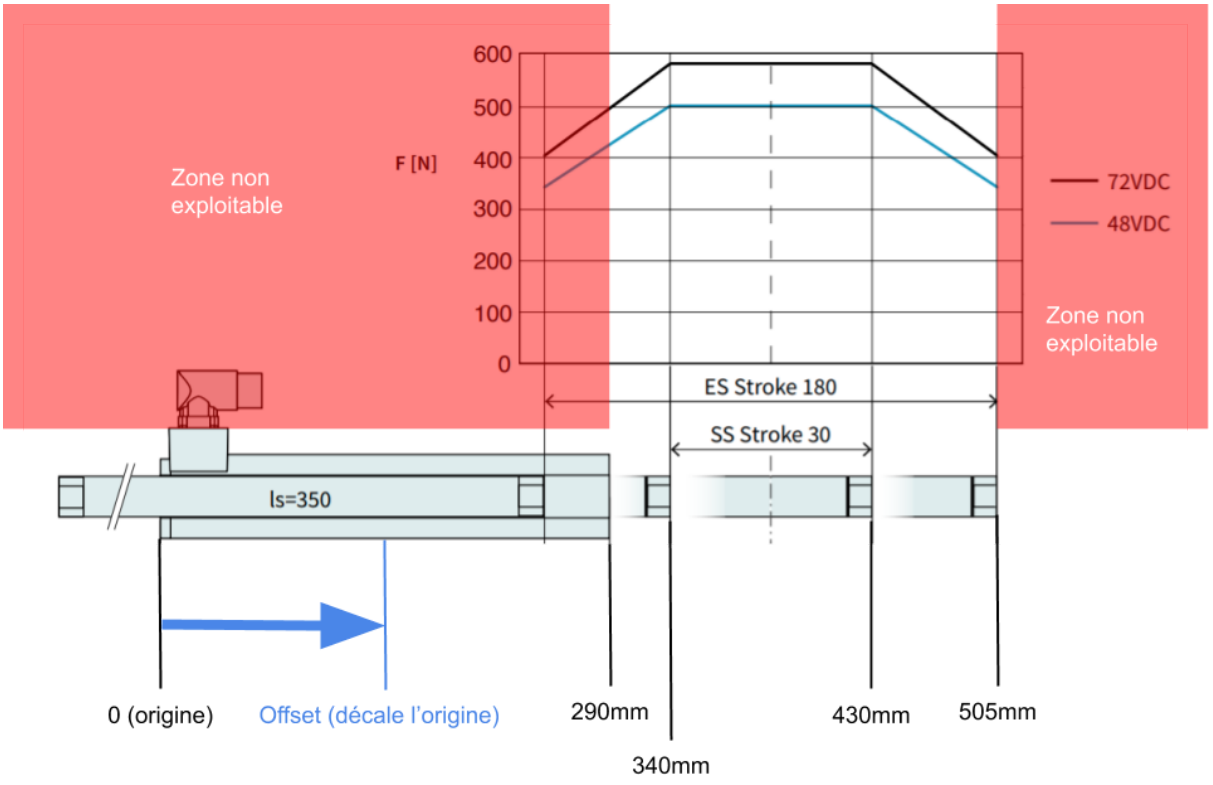
Les coordonnées des points d'accroche des moteurs sont exprimées dans les repères des différents segments définis ci-dessous :



Identification des repères de chaque segment du corps.

4.2 Moteurs mono-articulaires

Les moteurs mono-articulaires sont référencés P01-48x240/30x180. La documentation est disponible sur le site du fabriquant LinMot (page 4). Le référentiel du moteur est donné par l'illustration suivante :



La fonction MATLAB `motor_P01_48x240_30x240` calcule la force maximale que peut produire le moteur pour une longueur et un offset donné :

```
function [Force] = motor_P01_48x240_30x240(Length, Offset)
```

Par exemple :

```
>> motor_P01_48x240_30x240(340,0)

ans =

   585

>> motor_P01_48x240_30x240(340,-50)

ans =

  464.1000
```

4.3 Moteurs bi-articulaires

Les moteurs bi-articulaires sont référencés P01-48x360F/60x210. La documentation est disponible sur le site du fabriquant LinMot (page 3). Le référentiel du moteur est similaire aux moteur mono-articulaires.

La fonction MATLAB `motor_P01_48x360F_60x210` calcule la force maximale que peut produire le moteur pour une longueur et un offset donné :

```
function [Force] = motor_P01_48x360F_60x210(Length, Offset)
```

Par exemple :

```
>> motor_P01_48x360F_60x210(460,0)

ans =

      1020

>> motor_P01_48x360F_60x210(460,-50)

ans =

  877.2000
```