

## Documentation Code version longueurs de bras

- **Main :**

Ce fichier est la main et lance la suite de fonction pour calculer les différents éléments présents sur une série d'image enregistrée et faire les mouvements

- **Prog\_cinematic\_movement\_max4 :**

`gama` = rotation de la main sur l'axe x en degré  
`beta` = rotation de la main sur l'axe y en degré  
`alpha` = rotation de la main sur l'axe z en degré  
`xyz`, position de la main dans l'espace avec comme référence le centre du robot

`Full_matrice_Rota(gama,beta,alpha,x,y,z):`

Donne la matrice 4x4 nécessaire pour utiliser la cinématique inverse.

(voir : [Arms kinematics - Reachy 2023 \(pollen-robotics.com\)](https://pollen-robotics.com))

- **Inverse\_kinematic\_max4 :**

`gama` = rotation de la main sur l'axe x en degré  
`beta` = rotation de la main sur l'axe y en degré  
`alpha` = rotation de la main sur l'axe z en degré  
`right_wrist_coords` = position du poignet droit sur l'image enregistrée (position centrée par rapport au repère du robot et non de l'image)  
`left_wrist_coords` = position du poignet gauche sur l'image enregistrée (position centrée par rapport au repère du robot et non de l'image)  
`compteur_image` = nombre d'image de la séquence vidéo

`inverse_kinematic(gamma, beta, alpha, right_wrist_coords, left_wrist_coords, compteur_image) :`

Détermine la cinématique inverse pour le bras droit puis le bras gauche pour chaque image. On y trouve également des mesures de sécurité premièrement au niveau des positions des mains puis, au niveau de la position finale qui est vérifiée avec la cinématique avant. De plus, une méthode d'approximation du coude est aussi présente sous la forme d'une boucle while utilisant de l'aléatoire. Cette fonction enregistre la liste des 7 angles dont les bras ont besoin pour se déplacer pour chaque image dans un fichier.

`compteur_image` = nombre d'image de la séquence vidéo

`Mouvement_reachy(compteur_image)` :

Réalise les mouvements déterminé avec  
`inverse_kinematic(gamma, beta, alpha, right_wrist_coords,`  
`left_wrist_coords, compteur_image)`.

- **Caméra\_control\_max4**

`compteur_image` = nombre d'image de la séquence vidéo

`camera_control(compteur_image):`

Filme la séquence vidéo depuis le robot Reachy.

- **Camera\_control\_webcam\_max4**

Ce fichier filme une séquence vidéo depuis votre webcam

- **Conversion\_px\_cm\_max4**

`coeff_px_to_cm():`

Récupère la longueur du bras droit lors de la première image et en déduit le coef de conversion entre la taille du bras sur la vidéo et celle du vrai robot. Retourne le coefficient.

- **Coordinate\_transposition\_max4**

`member` = coordonnée d'un membre (ex : wrist , shoulder , elbow) dans la base de l'image afin de le placer dans la base du Reachy

`name` = nom du membre pour enregistrer les valeurs dans un dossier avec le bon nom

`compteur_image` = nombre d'image de la séquence vidéo

`coordinate_transposition_right_xy(member, name, compteur_image):`

Reposition les coordonnées des mains (ici main droite)  
depuis la base de l'image vers la base du Reachy.

`coordinate_transposition_left_xy(member, name, compteur_image):`

Reposition les coordonnées des mains (ici main gauche)  
depuis la base de l'image vers la base du Reachy.

- **Arm\_recognition\_max4**

`Coords` = coordonné y (hauteur) d'un membre

`invert_y_coords(coords):`

Inverse l'axe vertical d'une coordonnée pour avoir des axes similaires à ceux du robot Reachy.

`compteur_image` = nombre d'image de la séquence vidéo

`arm_recognition(compteur_image):`

Utilise mediapipe pour poser des « Landmark » sur chacune des images de la vidéo.

- **Posi\_elbow\_max4**

`U = i` pour la boucle actuelle

`posi_elbow_r(u):`

Détermine avec un calcul de trigo l'angle du `shoulder_roll` pour le bras droit. Cet angle ne prend pas en compte le `shoulder_pitch` et est donc faussé. Cet angle sert de référence pour l'aléatoire de la cinématique inverse dans `inverse_kinematic(gamma, beta, alpha, right_wrist_coords, left_wrist_coords, compteur_image)`.

`posi_elbow_l(u):`

Détermine avec un calcul de trigo l'angle du `shoulder_roll` pour le bras gauche. Cet angle ne prend pas en compte le `shoulder_pitch` et est donc faussé. Cet angle sert de référence pour l'aléatoire de la cinématique inverse dans `inverse_kinematic(gamma, beta, alpha, right_wrist_coords, left_wrist_coords, compteur_image)`.

- **Z\_coordinate\_max4**

`i = i` pour la boucle actuelle

`get_depth_r(i):`

Détermine la profondeur du bras droit à partir de la longueurs des bras.

`get_depth_l(i):`

Détermine la profondeur du bras gauche à partir de la longueurs des bras.

