Axe x = avant-arrière

Axe y = droite-gauche

Axe z = vertical

# Multipatte araignée

## Contexte

Servo H orienté sur l’axe z

Servo M orienté sur l’axe x

Servo L orienté sur l’axe x

* lH = Longueur jambe H
* lM = Longueur jambe M
* lL = Longueur jambe L
* d = Distance horizontale entre axe bras H et position du pied
* h = Hauteur entre axe bras H et touché
* Dmax la distance max du pied par rapport à l’axe

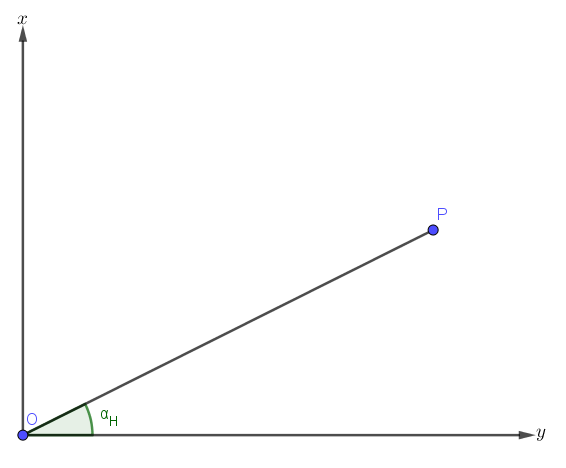
## Commandes de position

### Problématique

On se place dans le référentiel dont le centre est le servo H.

Etant donnée une position (x,y,z) du pied, trouver les angles des servos H, M et L, notés , et .

### Angle du servo H



Dans le plan horizontal, la position du pied en coordonnées polaires est donnée par

On a donc

### Angles des servos M et L

On se place dans le plan de la jambe. On note :

* a la distance entre l’axe du servo M et l’aplomb du pied,
* h la hauteur,
* c la distance entre le servo M et le pied

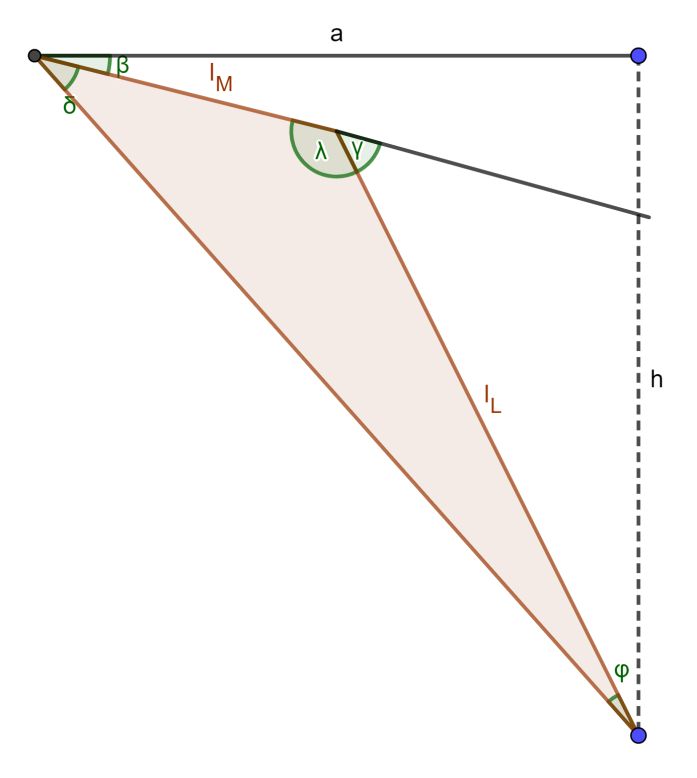


Figure  : parties M et L de la jambe.

La distance c entre le servo M et le pied est donnée par :

et l’angle entre l’axe horizontal et la droite menant jusqu’au pied est :

D’après la formule de Pythagore généralisée, on a

Donc :

Finalement,

De même, on a :

Donc

Finalement,

### Conditions

Le calcul de impose :

Le pied doit pouvoir toujours toucher le sol, donc

On a et . Pour une hauteur h, la plus grande valeur de est :

La plus petite valeur de est obtenue quand la 3e partie de la jambe est verticale. On a alors

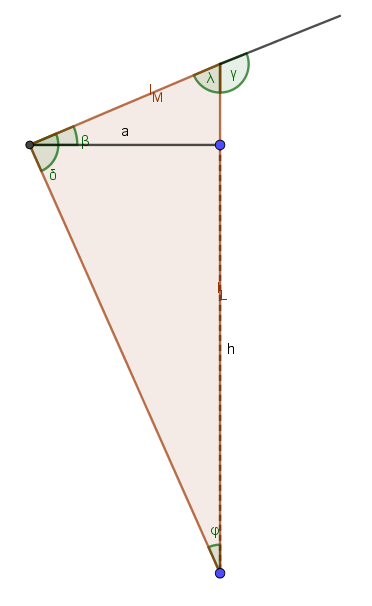
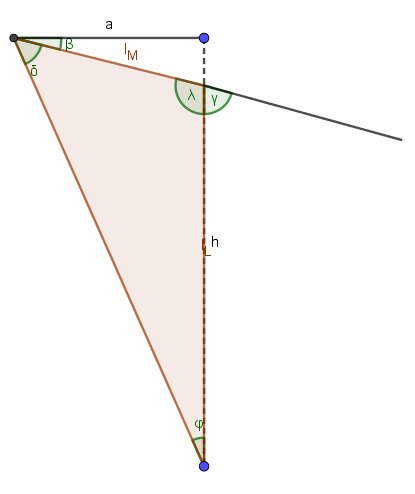


Figure  : Composante L verticale, pour et

En conclusion,

Pour des valeurs , et , on a

Si , on a

### Position de départ

Soit l’angle de départ. , la position en fonction de la hauteur est donnée par :

## Avance en ligne droite

### Micro-déplacements

Pour avancer le robot d’une distance , les pattes doivent reculer de cette même distance.

Soient le centre du robot, le servo et le pied, les coordonnées du servo dans le repère du robot, et les coordonnées du pied dans le repère du servo.

Donc pour une avance , le servo avance de par rapport au sol et le pied recule de dans le repère du servo. Donc

### Séquencement à vitesse constante

Soient

* la distance maximale qu’une patte peut parcourir pendant une séquence
* la fréquence d’échantillonnage du mouvement (50 Hz typiquement)

Pour avancer à une vitesse , le robot doit avancer de à chaque période.

La patte ne doit pas dépasser la distance D. Le nombre d’étapes avant d’arriver en butée est donc l’entier .

Pour un robot à pattes, chaque patte doit parcourir à chaque séquence. Le nombre d’étapes de chaque séquence est donc .

On suppose que la patte 0 est à la position 0, la patte 1 à la distance , la patte 2 à la distance et la patte 3 à la distance . La patte 3 sera en retour pendant les prochaines périodes et les autres pattes vont avancer de la distance . Pendant chaque période, les pattes 0, 1 et 2 vont avancer de et la patte 3 va reculer de .

**Algorithme** :

Initialisation :

* Nombre de pattes
* Vitesse
* Patte positionnée à la distance
* Micro-déplacement
* nombre d’étapes pour une séquence

Boucle pour une séquence :

* Les pattes 0 à avancent de
* La patte recule de

En fin de séquence, la patte en retour est revenue à la position 0, et la patte devient la nouvelle patte en retour.

### Séquencement à vitesse variable

Lorsque la vitesse est variable, il faut recalculer à chaque étape le nombre d’étapes restant avant la fin de la séquence. Si la vitesse varie de à , la vitesse de la patte en retour varie de si le nombre d’étapes ne varie pas. Si ne nombre d’étapes diminue, la vitesse de la patte en retour peut augmenter de plus de .

Si est la distance maximale que peut parcourir la patte avant de passer en retour, on note la vitesse maximale. Si , il y a diminution du nombre d’étapes avant passage en retour. De même, si , il y a augmentation du nombre d’étapes avant passage en retour.

Le cas le plus critique apparait quand et que . Dans ce cas, et la vitesse de la patte en retour doit doubler puisqu’elle aura la même distance à parcourir en 2 fois moins de temps.

Pour une fréquence de 50 Hz et une distance cm

## Avance en courbe

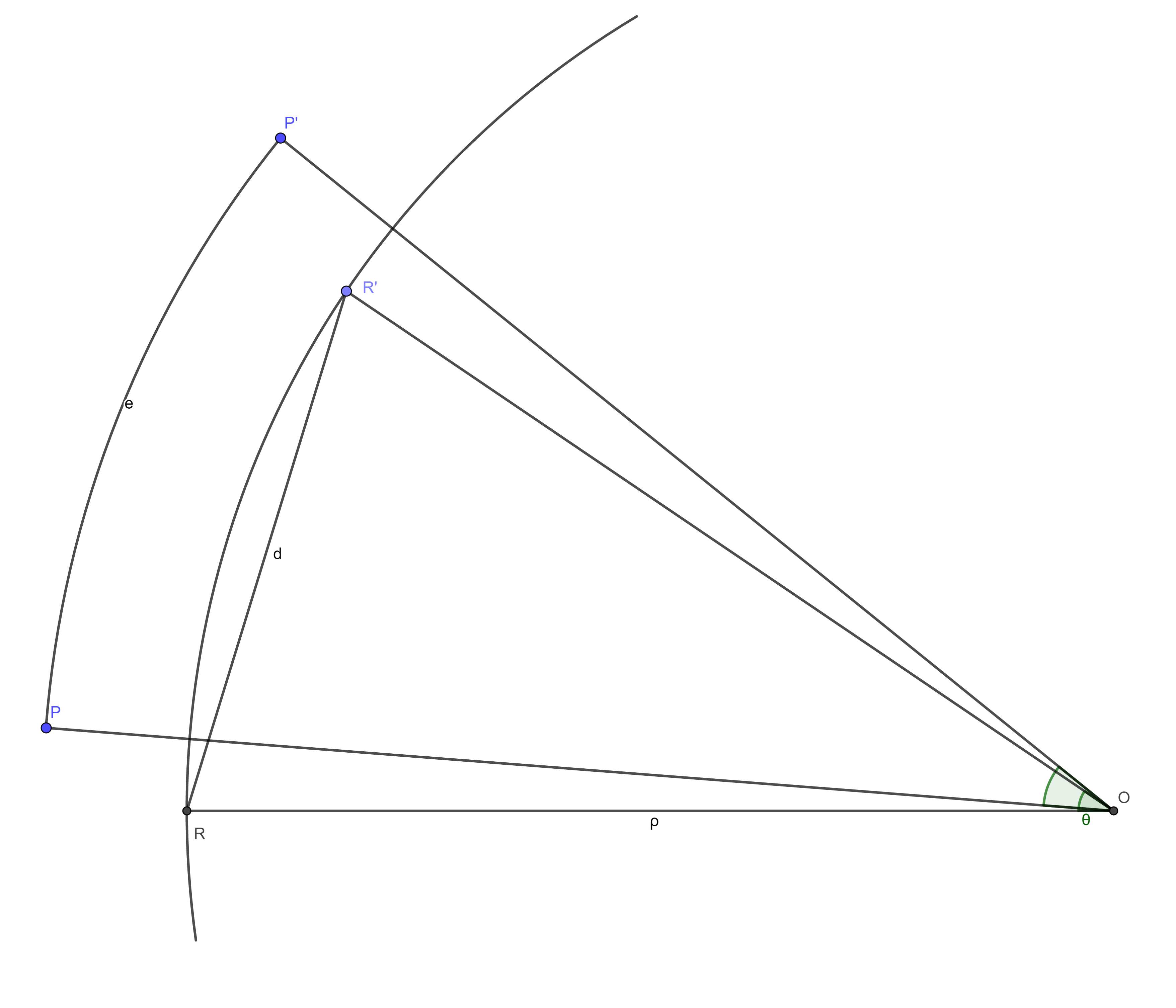
### Déplacement d’un point

Soient

* c la courbure de la trajectoire,
* d la distance d’avance du robot sur une étape
* p=(x,y).

On veut déterminer la position p’ de p après un déplacement de d le long du cercle de courbure c.

Le rayon de courbure est .



Dans le triangle 0RR’, on a

Donc

Soit R la matrice de rotation

La nouvelle position d’un point est donc :

Pour , on a et :

La position du robot est obtenue par une translation de vecteur D et d’une rotation d’angle par rapport au centre du robot, où le vecteur D est l’image du vecteur (0,d) par la rotation.

Donc l’avance du point dans le référentiel du centre du robot est donnée par :



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dans le référentiel du centre du cercle de courbure, le centre du RM a pour coordonnées (. Les coordonnées du pied après déplacement sont donc :

Finalement :

Si la courbure est nulle, on a simplement

### Avance

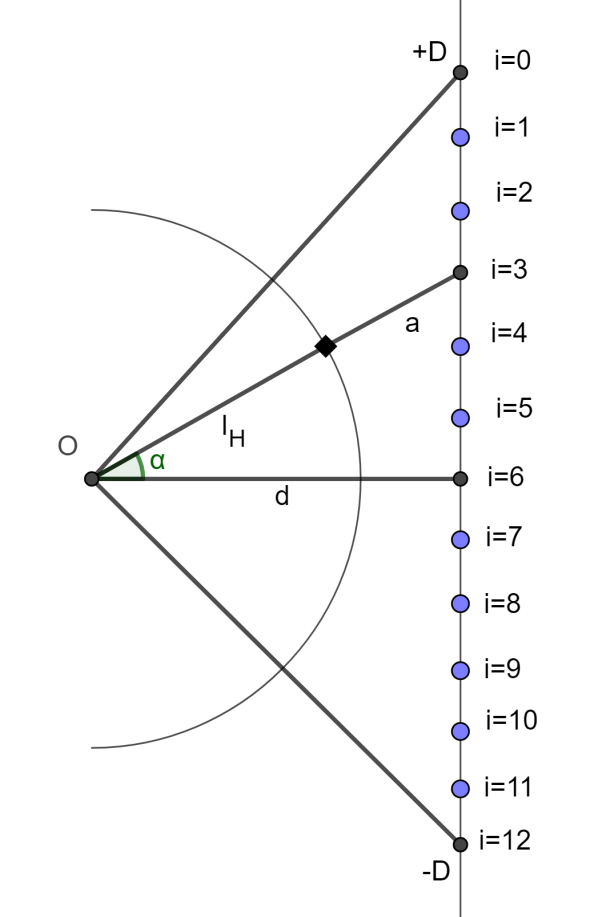
Pour une longueur d de bras et une distance D par rapport à l’axe, l’angle du bras est

L’angle est donné par

A l’inverse, la distance parcourue maximale est . Pour un angle de 45°, la distance parcourue est

Le pied doit se déplacer parallèlement à l’axe du robot.

La vitesse du pied doit être constante 🡺 la vitesse angulaire varie car la distance axe – pied varie. Pour une vitesse v, la vitesse angulaire dépend de sin a.



Si la distance parcourue est décomposée en N étapes (i=0=>=, i=N=>=-), alors à l’étape i, la distance est

et l’angle est

### Retour du bras

Le robot ayant 4 bras, le mouvement d’avance est décomposé en 3 phases d’avance + une phase de retour. On doit donc avoir N divisible par 3.

## Calcul des bornes

Pour une position de départ à , on cherche la valeur et l’angle associé tels que, quels que soient la vitesse et la courbure, le mouvement reste dans le domaine de définition.

## Bras H

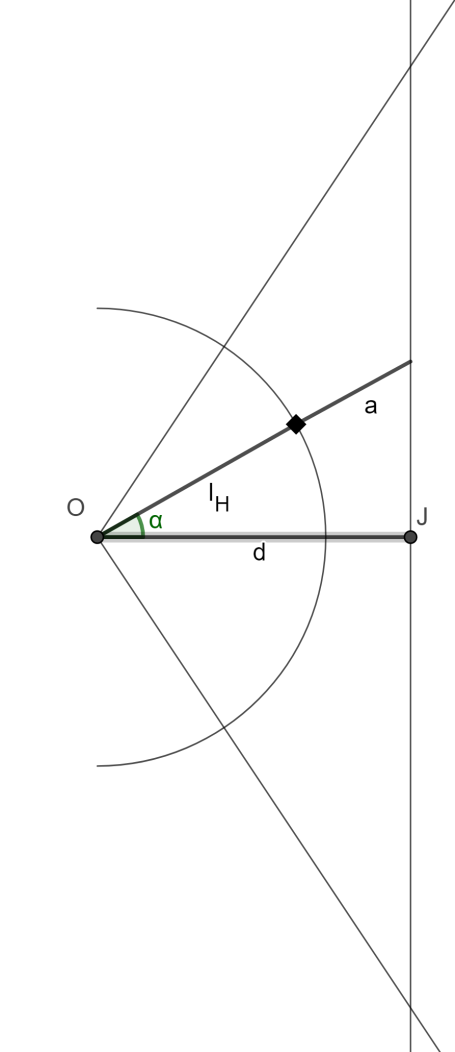


Figure  : Position du bras H

Pour un angle α du bras H, la distance entre le centre O et l’aplomb du pied est de d/cos(α). L’origine du bras M est donc à une distance a=d/cos(α)-lH de l’aplomb du pied.

## Bras M + bras L

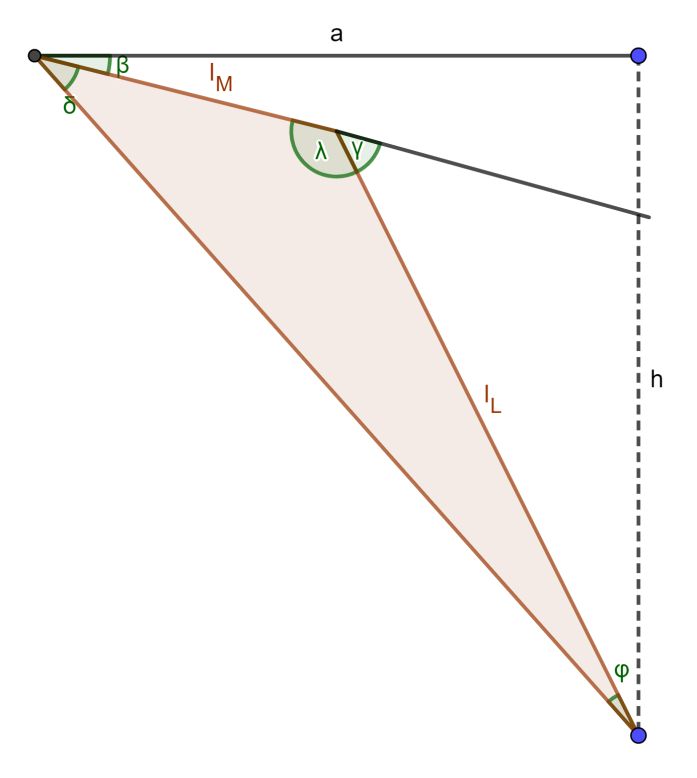


Figure  : position des bras M et L

Soit a la distance entre l’origine du bras M et l’aplomb du pied.

La distance entre l’origine du bras et le pied est et l’angle .

D’après la formule de Pythagore généralisée, on a

Donc :

Finalement,

De même, on a :

Donc

Finalement,

## Domaine de définition

donc

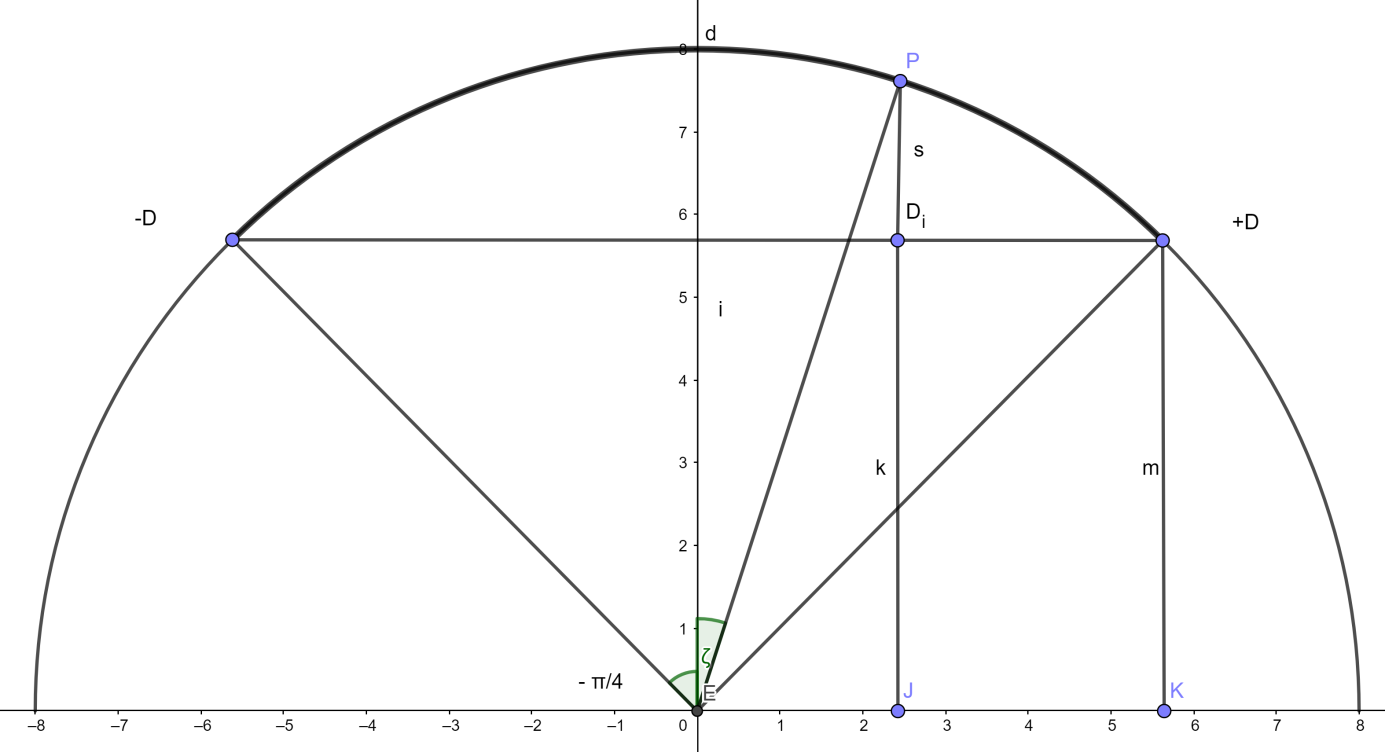
Donc

Donc

Il faut que le pied touche le sol lorsque le pied est à la distance Dmax. L’hypoténuse du triangle de côtés a et h doit être . A la distance maximale, l’angle alpha vaut , donc on doit avoir .

## Retour du bras

Pour le retour, l’angle doit être modifié pour que le pied ne touche pas le sol. Le pied doit parcourir un quart de cercle comme indiqué sur la figure suivante.



On ne considère que le mouvement du servo M, le servo L pouvant rester fixe pendant le retour.

A la distance D, le pied touche le sol, sa hauteur est donc 0. On note s la hauteur du pied à la distance . Le mouvement du pied pendant le retour est paramétré par l’angle variant de à .

Le rayon du cercle est égal à . La hauteur m du point P si

Le segment OD étant une diagonale du carré, pour , la hauteur du point P sur le cercle est égale à m=D. Pour quelconque, la hauteur du point P sur le cercle est :

Lorsque le pied est à la distance , l’angle est défini par :

Soit l’angle à soustraire à pour que le pied soit à une hauteur s et non au sol. L’angle vérifie , donc

# Miltipatte chien

## Contexte

Servo H orienté sur l’axe x

Servo M orienté sur l’axe y

Servo L orienté sur l’axe y

* lH = Longueur jambe H
* lM = Longueur jambe M
* lL = Longueur jambe L
* d = Distance horizontale entre axe bras H et position du pied
* h = Hauteur entre axe bras H et touché
* Dmax la distance max du pied par rapport à l’axe



Figure  : Pattes du robot-chien dans le plan (y,z) à gauche, et dans le plan de la patte à droite.

## Orientation du plan de la patte



Soit l’angle du plan de la patte dans le plan vertical . Alors .

## Genou et talon



Dans le plan de la patte, la hauteur de la patte vaut . On définit .

La distance est le troisième côté du triangle formé par les pattes M et L, mais aussi d’un triangle rectangle de côtés et . Donc .

D’après la *loi des cosinus*, donc



Dans le triangle, la même loi permet de trouver l’angle d :



L’angle entre u et la verticale est . L’angle est égal à l’angle du servo M.

# Avance du robot

## Etat initial

Soit n le nombre de pattes, n pair. Les pattes sont indexées de la façon suivante :

* i pair => patte gauche, i impair => patte droite
* i=0 => patte avant gauche

5

4

3

2

1

0

Les pattes sont positionnées pour que la répartition sur le domaine de définition soit uniforme : la patte 0 à ymin, la patte n-1 à ymax et les autres pattes à ymin+i/(n-1)(ymax-ymin)

Une patte est en mode retour. A l’initialisation, c’est celle qui est à y=ymax, c’est-à-dire la patte n-1.

## Changement de patte retour

L’avance des pattes fait que la prochaine patte à sortir de son domaine de définition est la patte suivant la patte en mode retour.

On note l’indice de la patte en mode retour et iN=. Dès que la patte iN sort de son domaine de définition, on fait iR=iR+1, iN=iN+1.

On note, lors du changement de patte en retour

* (xN0, yN0) la position de la prochaine patte
* uN= yMax-yN, vecteur restant à parcourir pour la prochaine patte, sur l’axe y
* dN=|yMax-yN|, représentant la distance à parcourir pour la prochaine patte
* uR=(xR-xMin, yR-yMin), vecteur restant à parcourir pour la patte en retour
* dR=sqr((xR)2+ (yMax-yR)2), distance restant à parcourir

## Avance d’une patte en retour

La patte en retour doit se poser en même temps que la prochaine patte sort de son domaine de définition. On calcule la distance d’avance de la patte en retour en fonction de la distance restant à parcourir pour la prochaine patte.

La hauteur de la patte en retour doit faire une portion de cercle, et toucher le sol au début et à la fin du retour. La hauteur est donc paramétrée par la distance à parcourir.

A un instant t, on note :

* xR, yR la position de la patte en retour
* xN, yN la position de la prochaine patte

La patte en retour doit parcourir une proportion du vecteur uR équivalente à la proportion du vecteur uN parcourue par la prochaine patte. Donc la patte en retour doit être à la position (xR0, yR0)+uR\*(yN-yN0)/dN