

# Table des matières

## I Interface Homme-Machine

### **1 Architecture générale**

### **2 Mise en œuvre**

### **3 Tableaux de bord**

- 3.1 Maintenance
- 3.2 Calibration
- 3.3 Prise en main - Pilotage par joystick
- 3.4 Asservissements
  - 3.4.1 Réglage du correcteur
  - 3.4.2 Saturation en vitesse
  - 3.4.3 Choix de l'axe piloté
  - 3.4.4 Consignes
- 3.5 Dynamique
- 3.6 Statique
- 3.7 Géométrie
- 3.8 Cinématique
- 3.9 Énergétique Puissance
- 3.10 API Python
- 3.11 IA Apprentissage par renforcement

### **4 Fonctions Python**

### **5 Mise à jour du logiciel MyViz**

### **6 Mise à jour du firmware du bras de pelleuse**

## II Modèles

### **7 Modélisation structurelle**

- 7.1 Diagramme de contexte
- 7.2 Diagramme de cas d'utilisation
- 7.3 Diagramme de définition des blocs
  - 7.3.1 Bras de la pelleuse
  - 7.3.2 Vérin électrique
- 7.4 Diagramme de blocs internes
  - 7.4.1 Bras de la pelleuse
  - 7.4.2 Vérin électrique

## II.I Modélisation géométrique

### 8 Modélisation plane

- 8.1 Schéma cinématique
- 8.2 Paramétrage global
  - 8.2.1 Paramétrage pièces
  - 8.2.2 Paramétrage liaisons
- 8.3 Hypothèse générales

### 9 Modèle géométrique direct

### 10 Modèle géométrique inverse

### 11 Cas du porteur seul

- 11.1 MGD
- 11.2 MGI
- 11.3 Deux solutions

### 12 Génération de mouvement point à point avec une trajectoire imposée

- 12.1 Équation d'une trajectoire du point
- 12.2 Application numérique

#### II.I.I Loi E/S

### 13 Loi E/S Flèche

- 13.1 Schéma cinématique
- 13.2 Paramétrage
- 13.3 Fermeture géométrique
- 13.4 Modèle géométrique
- 13.5 Résultats de simulation

### 14 Loi E/S balancier

- 14.1 Schéma cinématique
- 14.2 Paramétrage
- 14.3 Fermeture géométrique
- 14.4 Modèle géométrique
- 14.5 Résultats de simulation

### 15 Loi E/S triangle

- 15.1 Schéma cinématique
- 15.2 Paramétrage
- 15.3 Fermeture géométrique
- 15.4 Modèle géométrique
- 15.5 Résultats de simulation

### 16 Loi E/S godet

- 16.1 Schéma cinématique

- 16.2 Premier sous-système
  - 16.2.1 Paramétrage
  - 16.2.2 Fermeture géométrique
  - 16.2.3 Modèle géométrique
- 16.3 Deuxième sous-système
  - 16.3.1 Paramétrage
  - 16.3.2 Fermeture géométrique
  - 16.3.3 Modèle géométrique
- 16.4 Résultats de simulation

## II.II Modélisation des actions mécaniques

### 17 Vérin de flèche

- 17.1 Paramétrage
  - 17.1.1 Schéma cinématique
  - 17.1.2 Graphe de structure
  - 17.1.3 Hypothèses
- 17.2 Détermination de l'effort dans la tige du vérin flèche
  - 17.2.1 Inventaire des actions mécaniques
  - 17.2.2 Isolement de l'ensemble vérin flèche
  - 17.2.3 Isolement de la flèche
  - 17.2.4 Isolement de la tige du vérin

## II.III Modélisation volumique

### 18 Modélisation volumique du vérin électrique

- 18.1 Présentation
- 18.2 Arbre de construction dans SolidWorks
- 18.3 Détails des sous-ensembles

### 19 Modélisation spatiale minimale

- 19.1 Présentation
- 19.2 Analyse mécanique

## III Intelligence Artificielle

### 20 Apprentissage par renforcement

- 20.1 Qlearning
  - 20.1.1 Environnement
  - 20.1.2 Actions
  - 20.1.3 Récompenses

- 20.2 Epsilon-greedy

## IV Documentation technique

### IV.I Caractéristiques des pièces

#### **21 Caractéristiques géométriques**

- 21.1 Bâti
- 21.2 Flèche
- 21.3 Balancier
- 21.4 Triangle de renvoi
- 21.5 Bielle
- 21.6 Godet

#### **22 Centre de masse**

- 22.1 Flèche
- 22.2 Balancier
- 22.3 Triangle
- 22.4 Bielle
- 22.5 Godet
- 22.6 Vérin
  - 22.6.1 Corps
  - 22.6.2 Tige

### IV.II Chaîne d'information

#### **23 Mesures**

- 23.1 Codeurs d'axes
  - 23.1.1 Présentation
  - 23.1.2 Position angulaire des axes
  - 23.1.3 Vitesse angulaire des axes
- 23.2 Codeurs de vérins
  - 23.2.1 Présentation
  - 23.2.2 Position vérin
  - 23.2.3 Vitesse vérin
- 23.3 Capteur d'effort

#### **24 Commande des vérins**

- 24.1 Asservissement en vitesse
- 24.2 Asservissement en position

### IV.III Chaîne d'énergie

## **25 Verin électrique**

- 25.1 Référence
- 25.2 Constitution du vérin électrique
  - 25.2.1 Vue d'ensemble
  - 25.2.2 Transmission du mouvement arbre/vis de manœuvre
  - 25.2.3 Système vis/écrou
- 25.3 Schéma cinématique
- 25.4 Maquette numérique
- 25.5 Rapport de transmission
- 25.6 Masses des pièces du vérin électrique

## **26 Structure articulée**



Le **Bras de pelleteuse électrique autonome** est un produit inspiré des nouvelles générations de pelleteuses compactes **électriques** que les constructeurs commencent à développer afin de diminuer leur impact environnemental.

New Volvo Electric Com...



Certains constructeurs travaillent également à rendre leurs machines **autonomes**.

Robots That Build the W...



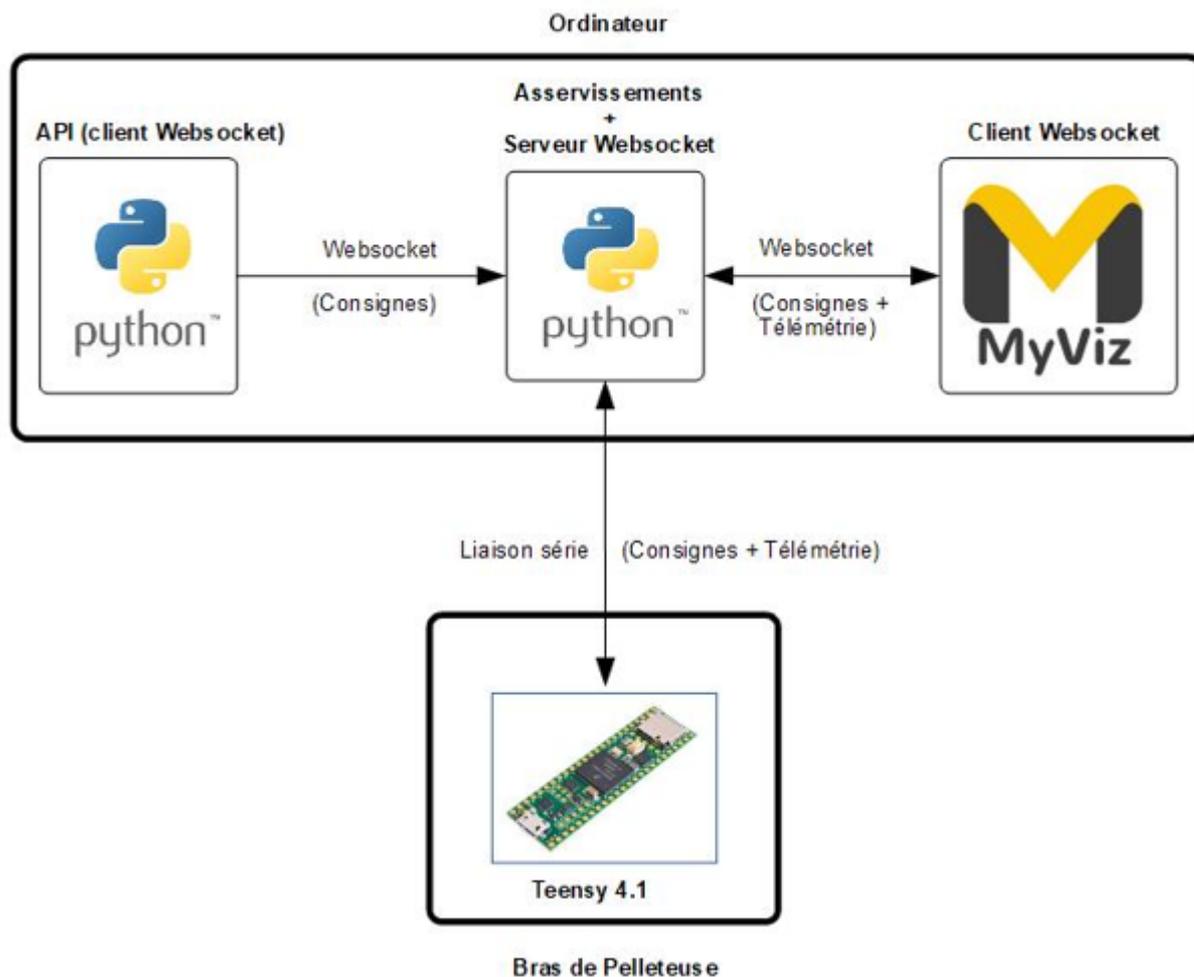
Le support de TP **Bras de pelleteuse électrique autonome** regroupe ces deux *technologies* et permet de réaliser des activités de travaux pratiques sur la quasi totalité du nouveau programme CPGE mis en place à la rentrée 2021.

Cette documentation est mise à jour dès qu'une erreur est rapportée. Pensez à vérifier que vous disposez de la dernière mise à jour.

# I. Interface Homme-Machine

# 1 Architecture générale

Le diagramme des échanges entre le bras de pelleuse et l'ordinateur est représenté ci-dessous :



Les **tableaux de bord** livrés avec le système permettent non seulement de piloter le bras de pelleuse réel, mais également un jumeau numérique de ce dernier.

Ce jumeau numérique n'est pas un modèle schéma-bloc simplifié mais un **simulateur très complet** réalisé avec MapleSim, logiciel de modélisation physique multi-domaine. Il intègre les sous-ensembles suivants :

- **partie mécanique** (importée à partir du modèle volumique SolidWorks) incluant le calcul des forces sur chaque vérin;
- **vérins électriques**;
- **boucles d'asservissement** en vitesse et position;
- **logiques de commutation** entre les différentes phases (initialisation / fonctionnement normal) et les différents types de pilotage (vitesses angulaires / vitesse des vérins / positions angulaires / longueurs des vérins);
- **cinématique inverse**;
- **calcul de trajectoire cartésienne**.

Ce modèle est mis en œuvre de la façon suivante :

- le modèle physique multi-domaine réalisé dans MapleSim est généré sous forme d'un composant de cosimulation de type FMU (code C compilé suivant le format standard « Functional Mockup Interface » en version 2.0);
- ce composant est exécuté en temps-réel par le programme Python de gestion du bras de pelleuse, grâce à la bibliothèque *FMPy*;
- les positions angulaires calculées par ce simulateur sont utilisées pour animer rendu 3D openGL du bras grâce à la bibliothèque *Pyrender*.

Enfin, les tableaux de bord de pilotage de ce jumeau numérique permettent de modifier les paramètres suivants :

- paramètres classiques des moteurs électriques des vérins (inertie, constante de couple, résistance, frottement visqueux et frottements secs)
- charge sur le godet : elle correspond à un glisseur sur le godet au point  $K$  de résultante égale à  $-m g \vec{y}$ .

Ceci illustre très bien l'intérêt d'un jumeau numérique, qui se situe entre le système réel et le schéma-bloc simplifié :

- il n'est pas possible de modifier les caractéristiques des moteurs sur le système réel. On peut bien sûr charger le godet sur ce dernier, mais le dépassement des capacités limites de la pelleuse pourrait détériorer celle-ci;
- on peut modifier les caractéristiques d'un moteur sur un schéma-bloc simplifié d'un seul vérin, mais on ne peut pas voir l'influence sur le système complet lorsqu'il est piloté en cartésien, par exemple;

- le jumeau numérique permet quant à lui :
  - d'avoir une très bonne idée de l'influence de certains paramètres sur le système complet;
  - de modifier des paramètres sans risque, ni pour le matériel, ni pour les opérateurs.

M Paramètres

## Moteurs vérins et charge godet

Réinitialiser aux valeurs par défaut

Inertie (entre 1e-6 et 9e-6 kg.m<sup>2</sup>):

Constante de couple (entre 0.01 et 0.09 N.m/A):

Résistance (entre 1 et 9 ohms):

Frottement visqueux (entre 0 et 0.01 N.m.s/rad):

Frottement sec (entre 0 et 5 N.m):

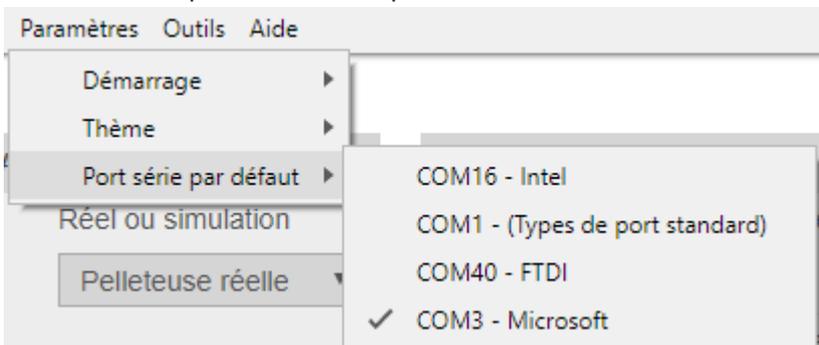
Charge godet (entre 0.01 et 200 kg):

Enregistrer

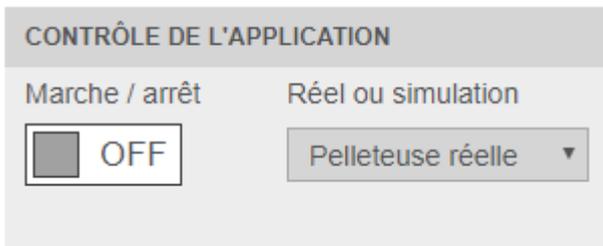
## 2 Mise en œuvre

La procédure à suivre est simple :

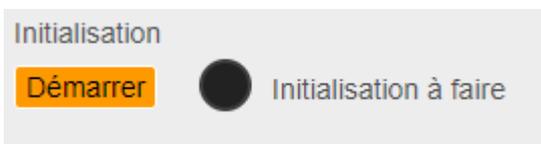
1. Brancher le câble d'alimentation du bras de pelleuseuse.
2. Connecter le câble USB entre l'ordinateur et le système.
3. Brancher le joystick sur l'ordinateur.
4. Basculer l'interrupteur d'alimentation du bras de pelleuseuse. Le voyant rouge de cet interrupteur s'allume alors.
5. Appuyer sur le bouton poussoir jaune. Deux LEDs rouge et verte s'allument presque instantanément, indiquant la présence du 24 V et du 5 V sur la carte de pilotage.
6. Lancer **MyViz** et ouvrir le tableau de bord **Pelleuseuse\_Maintenance.json**.
7. Sélectionner le port série utilisé par l'ordinateur via le menu Paramètres → Port série par défaut :



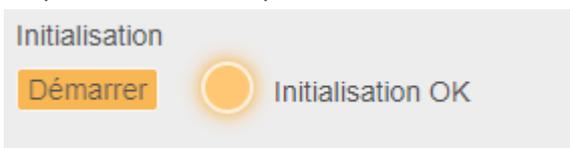
8. Cliquer sur le bouton Marche / arrêt en haut à gauche pour démarrer l'application



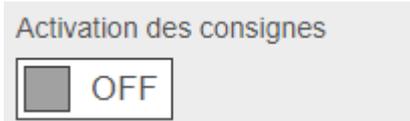
9. Le message **Attente de démarrage** s'affiche alors. Puis après quelques instants, le bouton d'initialisation devient actif :



10. Cliquer sur **Démarrer** pour lancer l'initialisation et attendre jusqu'à son terme :



11. L'interrupteur d'activation des consignes devient actif :



12. Cliquer dessus. Le bras de pelleteuse est maintenant pilotable avec le joystick.

**i Arrêt d'urgence**

En cas d'appui sur le bouton d'arrêt d'urgence puis de réarmement de ce dernier, il faut de nouveau appuyer sur le bouton poussoir jaune pour remettre sous tension la carte électronique.

**i Mise hors-tension**

Pour éteindre le système, il suffit de basculer sur 0 l'interrupteur d'alimentation général.

## 3 Tableaux de bord

L'application MyViz fonctionne en *tableaux de bord* associés à une activité pédagogique permettant ainsi de n'afficher que les informations utiles sans surcharger l'environnement.

### Note

Mêmes si les informations affichées sont limitées, toutes les grandeurs mesurables restent accessibles via le bouton "Tracer".

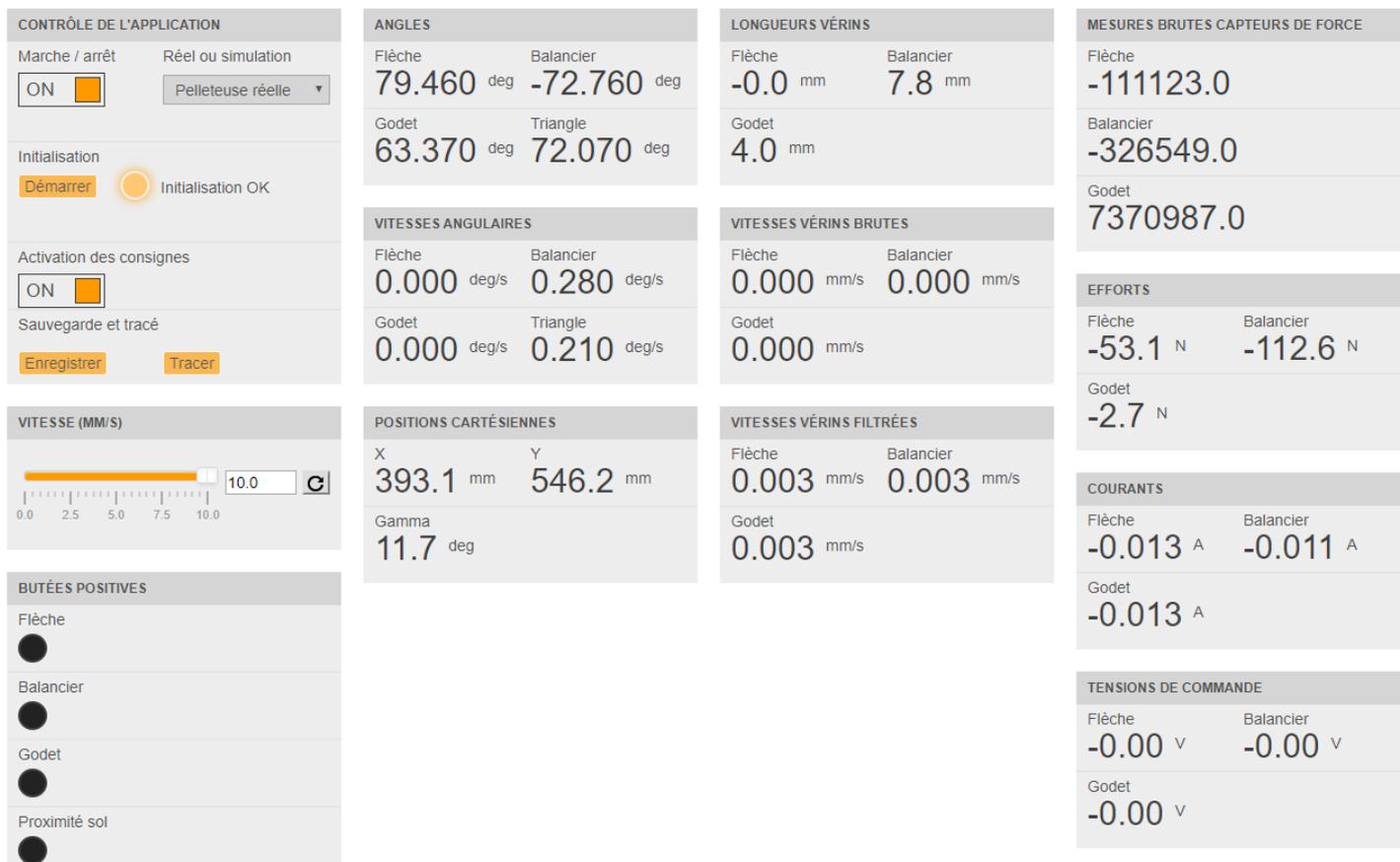
## 3.1 Maintenance

### Objectif

Ce tableau de bord permet de piloter le bras de pelleuse avec le joystick et de visualiser les valeurs des différents signaux mesurés par le système.

Ouvrir dans **MyViz** le fichier **Pelleteuse\_Maintenance.json**

Ci-dessous, une capture d'écran du tableau de bord MyViz.



### Warning

Ce tableau de bord ne permet pas de faire des expériences pédagogiques mais de vérifier que toutes les entrées / sorties sont correctement lues et pilotées.

## 3.2 Calibration

### Objectif

Ce tableau de bord permet de calibrer les capteurs d'effort situé sur chaque tige de vérin après un déplacement du banc de TP.

Ouvrir dans **MyViz** le fichier **Pelleteuse\_MaintenanceCalibration.json**

### Quand calibrer les capteurs ?

Il est conseillé d'effectuer la procédure de calibration dans les cas suivants

- après chaque déplacement du banc d'essai
- si les valeurs des capteurs d'effort semblent incohérentes

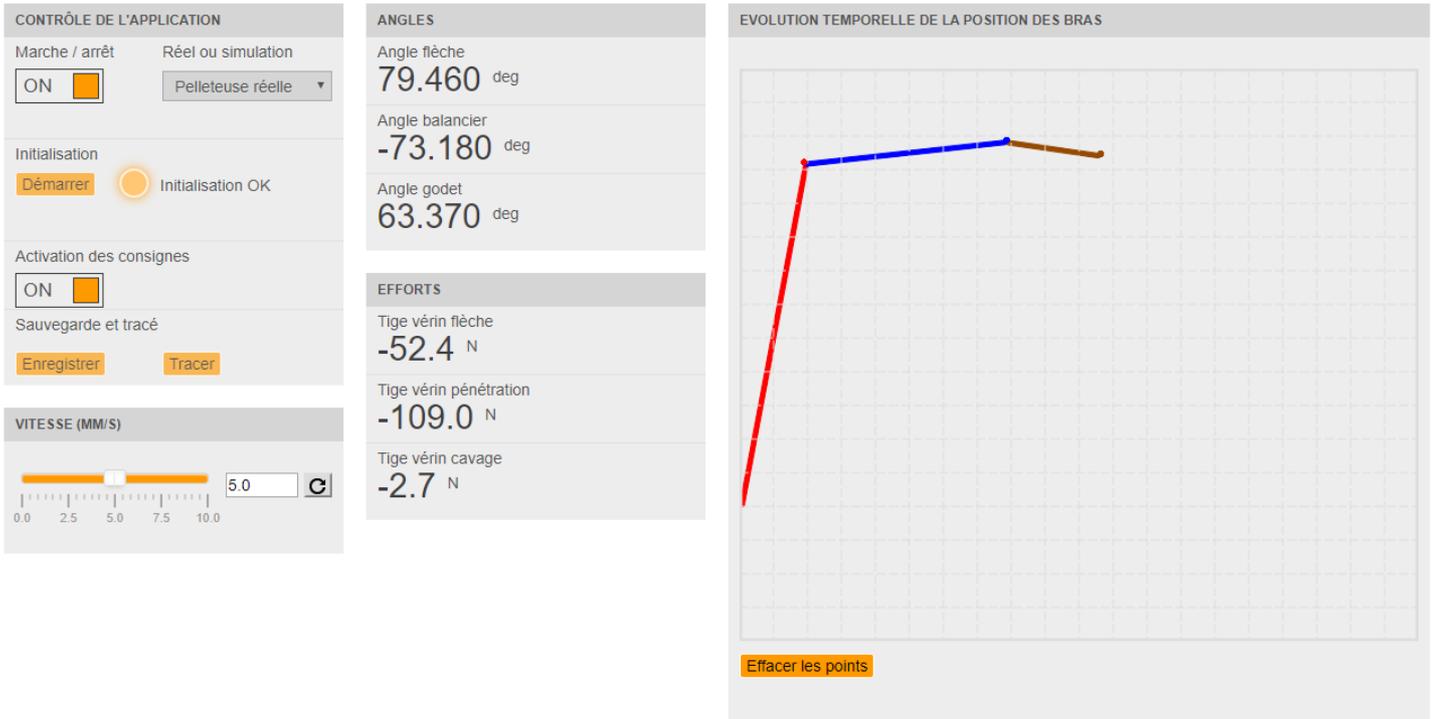
## 3.3 Prise en main - Pilotage par joystick

### Objectif

Ce tableau de bord permet de prendre en main le pilotage de la pelleteuse par joystick et de visualiser les angles et efforts associés.

Ouvrir dans **MyViz** le fichier **Pelleteuse\_PriseEnMain.json**

Ci-dessous, une capture d'écran du tableau de bord MyViz.



Il est possible de régler la vitesse des vérins entre 0 et 10 mm/s, la valeur par défaut étant de 5 mm/s.

La fenêtre de visualisation reprend l'historique des positions et orientations mesurées des extrémités du balancier (point  $O_2$ ), de la flèche (point  $O_3$ ) et du godet (point  $L$ ) sous forme de points. Il est possible d'effacer cet historique via le bouton **Effacer les points**.

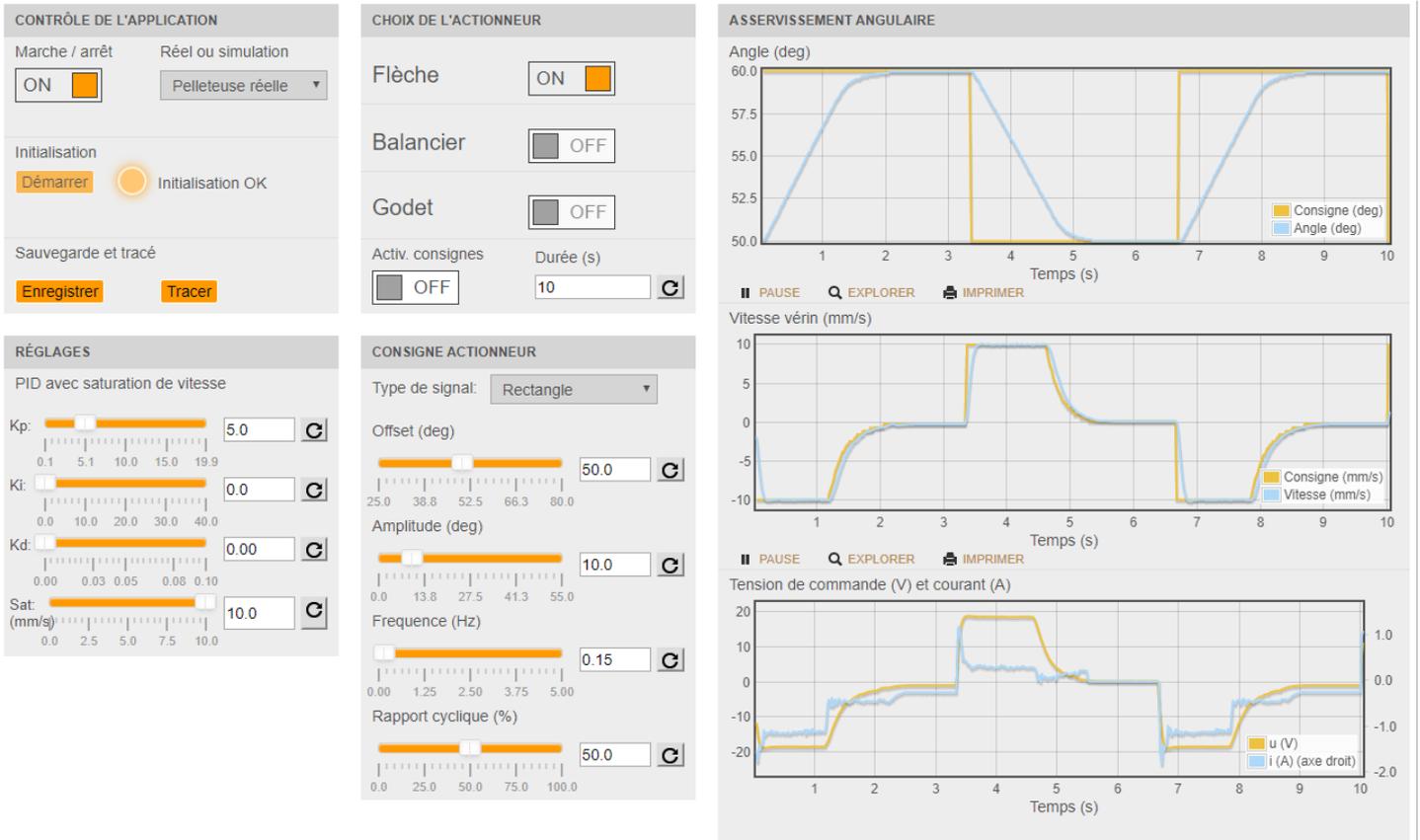
### 3.4 Asservissements

#### Objectif

Ce tableau de bord permet de spécifier différents types de consignes pour un sous-ensemble donné (flèche, balancier ou godet) et de voir sur les tracés graphiques l'influence de la modification des réglages des asservissements.

Ouvrir dans **MyViz** le fichier **Pelleteuse\_Asservissement.json**.

Voici une capture d'écran du tableau de bord MyViz :



### 3.4.1 Réglage du correcteur

Le choix du type de correcteur fait en imposant les gains  $K_p$ ,  $K_i$  et  $K_d$  correspond à la fonction de transfert du correcteur :  $C(p) = K_p \left( 1 + \frac{K_i}{p} + K_d p \right)$ . Ces réglages se font dans le cadre **Réglages**.

### 3.4.2 Saturation en vitesse

La boucle de vitesse présente une saturation dont la valeur peut être modifiée dans le cadre **Réglages**.

### 3.4.3 Choix de l'axe piloté

Dans ce tableau de bord, seul un seul axe peut-être piloté.

Le choix de l'axe se fait dans le cadre **Choix de l'actionneur**.

Le bouton **Activation des consignes** permet de lancer une mesure.

La durée de l'acquisition peut être modifiée (10 s par défaut). Il est nécessaire de désactiver les consignes pour pouvoir exploiter les résultats via les boutons **Enregistrer** et **Tracer** du cadre **Contrôle de l'application**.

### 3.4.4 Consignes

Plusieurs type de consignes sont proposées avec chacune leur paramètres de réglages

- rectangle : correspond à une consigne en échelon
- sinus

- trapèze
- triangle
- rampe

## 3.5 Dynamique

### Objectif

Ce tableau de bord permet de visualier les efforts dans les vérins au cours d'un mouvement imposé.

Ce tableau de bord est très similaire au précédent (asservissements), avec les différences suivantes :

- Les asservissements ne sont plus réglables
- Les tracés graphiques permettent de visualiser les forces sur chaque sous-ensemble

Ouvrir dans **MyViz** le fichier **Pelleteuse\_Dynamique.json**

Voici une capture d'écran du tableau de bord MyViz :



## 3.6 Statique

## Objectif

Ce tableau de bord permet de positionner le bras de pelleuse dans la posture angulaire spécifiée avec une vitesse de déplacement potentiellement très lente et de relever les forces correspondantes.

Ouvrir dans **MyViz** le fichier **Pelleteuse\_Statique.json**

Voici une capture d'écran du tableau de bord MyViz :



## 3.7 Géométrie

### Objectif

Ce tableau de bord permet de travailler sur la correspondance entre la longueur du vérin et la position angulaire d'un sous-ensemble donné (flèche, balancier ou godet).

Ouvrir dans **MyViz** le fichier **Pelleteuse\_Geometrique.json**.

Voici une capture d'écran du tableau de bord MyViz :

**CONTRÔLE DE L'APPLICATION**

Marche / arrêt  ON

Réel ou simulation  
Pelleteuse réelle

Initialisation  
Démarrer  Initialisation OK

Sauvegarde et tracé  
Enregistrer  Tracer

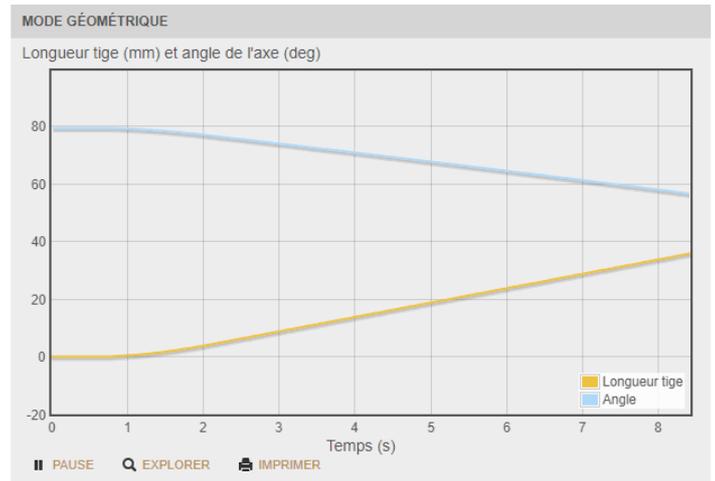
**PILOTAGE DE L'ACTIONNEUR**

Flèche  ON

Balancier  OFF

Godet  OFF

Excursion angulaire aller-retour  
 ON



Une fois l'axe choisi, l'activation de l'*excursion angulaire aller-retour* va déplacer l'axe sur toute la plage angulaire dans un sens puis dans l'autre.

## 3.8 Cinématique

### 🔥 Objectif

Ce tableau de bord permet de travailler sur la correspondance entre la vitesse de la tige du vérin et la vitesse angulaire d'un sous-ensemble donné (flèche, balancier ou godet).

Ouvrir dans **MyViz** le fichier **Pelleteuse\_Cinematique.json**.

Voici une capture d'écran du tableau de bord MyViz :

**CONTRÔLE DE L'APPLICATION**

Marche / arrêt  ON

Réel ou simulation  
Pelleteuse réelle

Initialisation  
Démarrer  Initialisation OK

Sauvegarde et tracé  
Enregistrer  Tracer

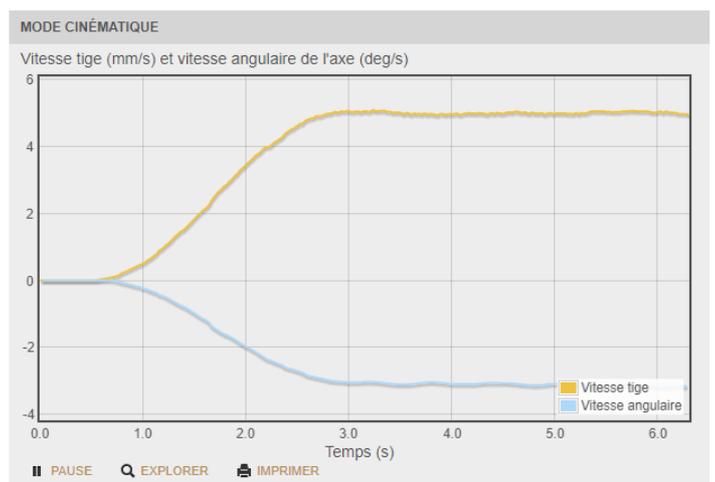
**PILOTAGE DE L'ACTIONNEUR**

Flèche  ON

Balancier  OFF

Godet  OFF

Excursion angulaire aller-retour  
 ON



## 3.9 Énergétique Puissance

## Objectif

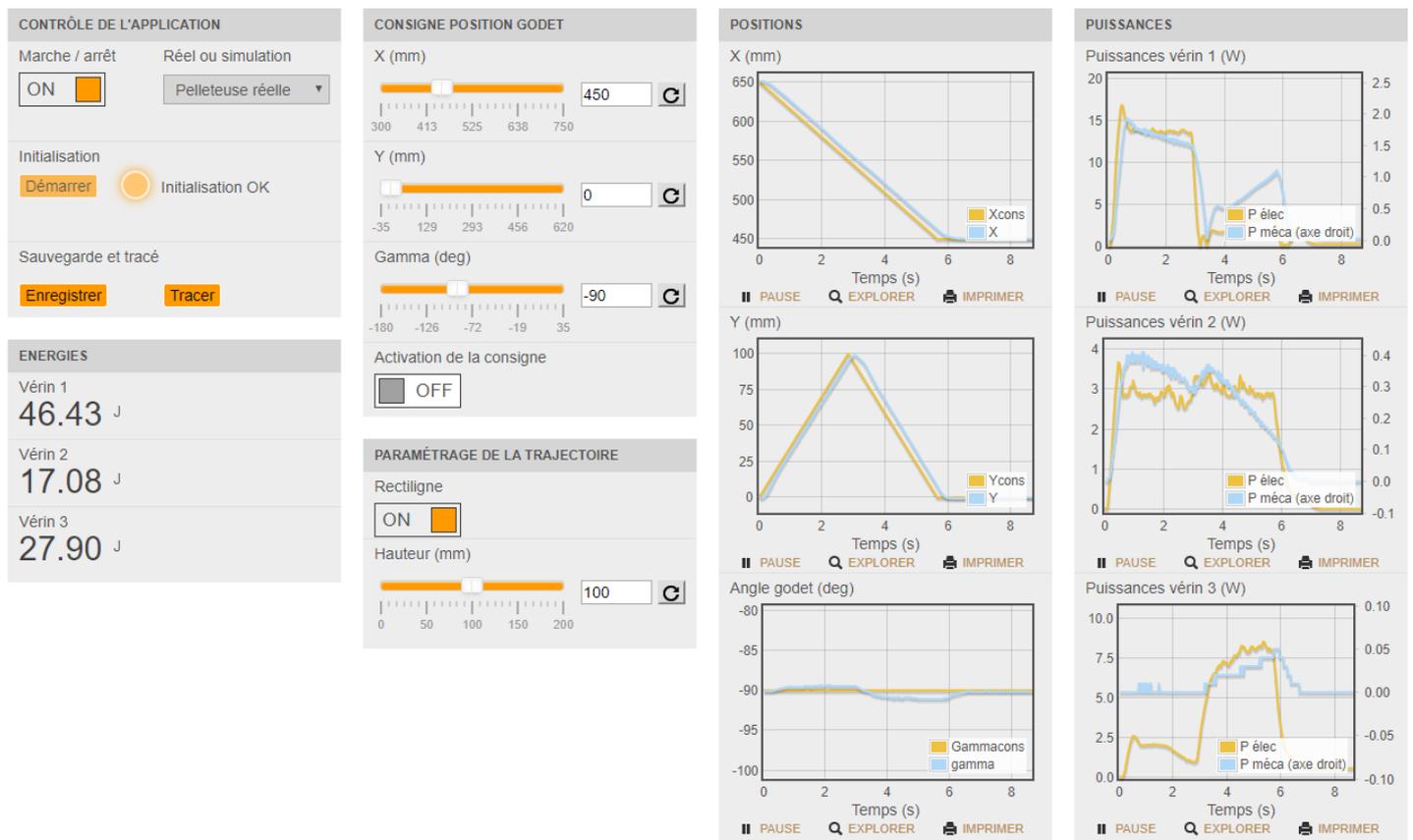
Ce tableau de bord permet de positionner le godet en coordonnées cartésiennes et de relever les différentes puissances et énergies mises en œuvre pour réaliser le mouvement.

Deux types de trajectoires sont possibles :

- Rectiligne : le godet va de sa position courante à sa position cible en suivant une trajectoire complètement rectiligne ou bien en suivant deux morceaux de trajectoires rectilignes passant par un sommet dont la hauteur est réglable par un curseur
- Non rectiligne : les axes ne sont plus synchronisés. Chaque vérin cherche à atteindre sa position cible le plus rapidement possible.

Ouvrir dans **MyViz** le fichier **Pelleteuse\_EnergetiquePuissance.json**.

Voici une capture d'écran du tableau de bord MyViz.



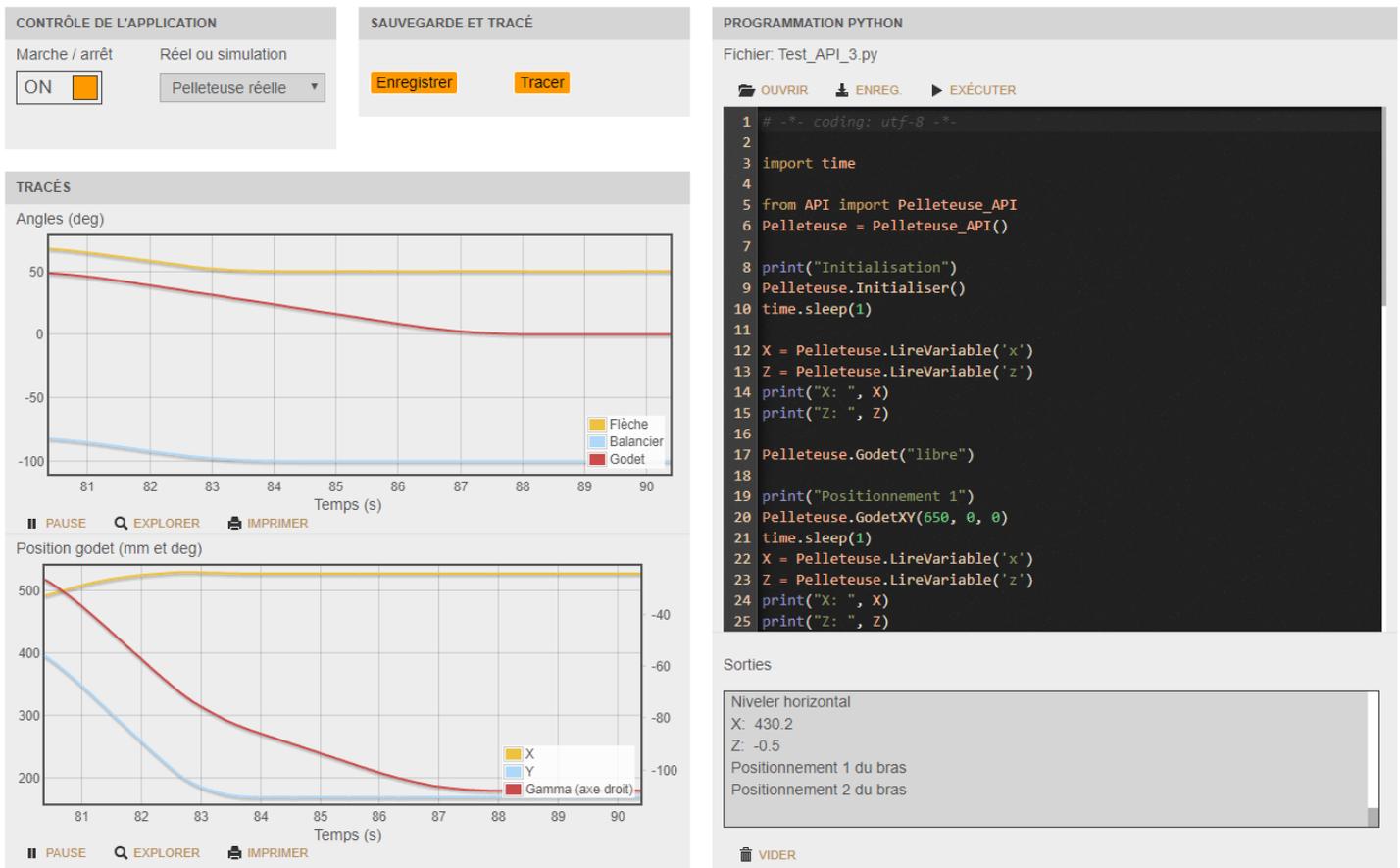
## 3.10 API Python

### Objectif

Piloter la pelleteuse (réelle ou virtuelle) à partir d'un programme python utilisant des fonctions de haut niveau.

Ouvrir dans **MyViz** le tableau de bord **Pelleteuse\_Python.json**

Ci-dessous une capture d'écran du tableau de bord



## 3.11 IA Apprentissage par renforcement

### Objectif

Réaliser un apprentissage par renforcement permettant de déterminer la meilleure trajectoire (d'un point de vue énergétique) pour aller d'un point de départ à un point d'arrivée.

Ouvrir dans **MyViz** le fichier **Pelleteuse\_ApprentissageRenforcement.json**.

Voici une capture d'écran du tableau de bord MyViz.

MyViz - C:\MyViz\Pelleteuse\_TableauxDeBord\_1.0\Pelleteuse\_ApprentissageRenforcement.json

Fichier Paramètres Outils Aide

**CONTRÔLE DE L'APPLICATION**

Marche / arrêt Pelleteuse simulée

OFF  Modifier le modèle

Mode

Apprendre

Sauvegarde et tracé

Enregistrer Tracer

**RÉGLAGES (MODE APPRENTISSAGE)**

Facteur d'apprentissage  $\alpha$

0.6 C

Facteur de dépréciation  $\gamma$

0.9 C

Bonus en cas de succès

10 C

Malus si obstacle ou hors domaine

-10 C

**FACTEUR D'ACCÉLÉRATION**

0: pause de 5 s ; 4: vitesse max

4 C

**EVOLUTION**

Nombre d'itérations

1000 C

Itération Facteur aléatoire

1000 0.01

Obstacle ou hors domaine

Récompense

0.06

Energie en cours Energie minimale

82.0 J 96.0 J

Nombre de succès

418

**ACTIONS POSSIBLES**

Haut

Gauche-Haut

Gauche

Gauche-bas

Bas

**EVOLUTION TEMPORELLE DE LA POSITION DU GODET**

Q des actions de l'état courant:

9.04 8.14

9.06 ← Q

-10.00 -10.00

Récompense: 0.06

**OBSTACLE**

Type d'obstacle

Rectangle

**RÉCOMPENSE**

Facteur multiplicatif

0.001 C

Comparaison distance

120 C

## Description de la méthode

Les listes des variables est décrite dans la partie [IA Apprentissage par renforcement](#).

Durant les calculs, le chemin en cours est représenté par des flèches oranges.

Les flèches vertes qui apparaissent dans les cases correspondent à la meilleure action à prendre lorsque l'agent se trouve dans cet état. Au cours des itérations, avec l'actualisation des valeurs de la matrice  $Q$ , la direction des flèches peut évoluer.

Les flèches violettes représentent la meilleure trajectoire (d'un point de vue énergétique) trouvée depuis le début du calcul. Suivant les paramètres définis et l'itération en cours, cette trajectoire n'est pas forcément la meilleure et peut donc encore évoluer à la suite du calcul.

# 4 Fonctions Python

## 4.1 Pelleteuse\_API()

### Description

Elle lance en particulier en tâche de fond le client Websocket destiné à envoyer les ordres de pilotage au système.

### Exemple

```
1 import time
2
3 from API import Pelleteuse_API
4 Pelleteuse = Pelleteuse_API()
```

 Cette fonction doit impérativement être exécutée au début de chaque programme.

## 4.2 Initialiser()

### Description

Cette fonction lance la prise d'origine du bras de pelleteuse.

### Exemple

```
1 import time
2
3 from API import Pelleteuse_API
4 Pelleteuse = Pelleteuse_API()
5
6 print("Initialisation")
7 Pelleteuse.Initialiser()
8 time.sleep(1)
9
10 Pelleteuse.Terminer()
```

 Cette fonction doit impérativement être exécutée au début de chaque programme.

## 4.3 Terminer()

### Description

Cette fonction permet l'arrêt du programme de manière *propre*.

### Exemple

```
1 import time
2 from API import Pelleteuse_API
3 Pelleteuse = Pelleteuse_API()
4
5 print("Initialisation")
6 Pelleteuse.Initialiser()
7 time.sleep(1)
8
9 Pelleteuse.Terminer()
```

 Cette fonction doit impérativement être exécutée à la fin de chaque programme.

## 4.4 PID\_position(Kp, Ki, Kd, vmax)

### Paramètres

- `Kp` : réel. Gain proportionnel du PID d'asservissement en position, saturé en interne entre 0,1 et 9,9.
- `Ki` : réel. Gain intégral du PID d'asservissement en position, saturé en interne entre 0 et 40.
- `Kd` : réel. Gain dérivé du PID d'asservissement en position, saturé en interne entre 0 et 0.1.
- `vmax` : réel. Vitesse maximale de consigne de la boucle de vitesse, saturée en interne entre 0 et 10 mm/s.

### Description

Cette fonction permet de changer les gains et la vitesse de saturation du PID d'asservissement en position.

## 4.5 Verin(sous-ensemble, mode, consigne, duree)

### Paramètres

- `sous-ensemble`
  - type : chaîne de caractères
  - valeurs : `fleche`, `balancier` ou `godet`
- `mode`
  - type : chaîne de caractères
  - valeurs : `position` ou `vitesse`
- `consigne` :
  - type : réel ou chaîne de caractères
  - valeurs : expression Python valide du type  $0.5 * \sin(t)$ ,  $t$  étant reconnu comme le temps courant, **uniquement en mode « vitesse »**. Consigne de position (en mm) ou de vitesse (en mm/s)
- `duree` : réel. Durée de la manœuvre en secondes.

### Description

Cette fonction donne une consigne de position ou de vitesse au vérin du sous-ensemble sélectionné pendant une certaine durée.

## 4.6 `Angle(sous-ensemble, consigne, duree)`

### Paramètres

- `sous-ensemble`
  - type : chaîne de caractères
  - valeurs : `fleche`, `balancier` ou `godet`
- `consigne` : réel. Consigne de position angulaire (en deg)
- `duree` : réel. Durée de la manœuvre en secondes.

### Description

Cette fonction donne une consigne de position angulaire au sous-ensemble sélectionné pendant une certaine durée.

## 4.7 `Bras(angle_fleche, angle_balancier, angle_godet)`

### Paramètres

- `angle_fleche` : réel. Consigne de position angulaire de la flèche (en deg)
- `angle_balancier` : réel. Consigne de position angulaire du balancier (en deg)
- `angle_godet` : réel. Consigne de position angulaire du godet (en deg)

### Description

Cette fonction donne une posture globale au bras de pelleuse. Le positionnement se fait d'une manière douce, sur une durée calculée automatiquement

## 4.8 `Godet(mode)`

### Paramètres

- `mode` : chaîne de caractères (`libre` ou `fixe`)

### Description

Cette fonction définit le mode de fonctionnement du godet lorsque ce dernier est piloté par la fonction `GodetXY` (voir plus loin) :

- `libre` : le godet conserve son orientation par rapport au bâti
- `fixe` : le godet conserve son orientation par rapport au balancier

## 4.9 `GodetXY(xcons, ycons, gammacons, rectiligne = 1, hauteur = 0)`

### Paramètres

- `xcons` : réel. Consigne de position cartésienne horizontale du pivot du godet (en mm)
- `ycons` : réel. Consigne de position cartésienne verticale du pivot du godet (en mm)
- `gammacons` : réel. Consigne de position angulaire du godet (en deg)
- `rectiligne` : entier (optionnel). 1 mode rectiligne (valeur par défaut), 0 : mode non synchronisé (voir tableau de bord Énergétique - Puissance).
- `hauteur` : entier (optionnel). Hauteur max de la trajectoire (en mode rectiligne uniquement, voir tableau de bord Énergétique - Puissance). Valeur par défaut : 0

### Description

Cette fonction positionne le godet en coordonnées cartésiennes, sur une durée calculée automatiquement.

## 4.10 Charger()

### Description

Positionne le godet à l'horizontale (dans la limite de ce qui est mécaniquement possible).

### Exemple

```
1 import time
2
3 from API import Pelleteuse_API
4 Pelleteuse = Pelleteuse_API()
5
6 print("Initialisation")
7 Pelleteuse.Initialiser()
8 time.sleep(1)
9
10 Pelleteuse.Bras(30,-80,50)
11
12 print("Debut chargement godet")
13 Pelleteuse.Charger()
14 print("Fin chargement godet")
15 time.sleep(1)
16
17 Pelleteuse.Terminer()
```

## 4.11 Decharger()

### Description

Rentre le vérin du godet au maximum.

## 4.12 Nivelier(mode, cons\_mini, cons\_maxi)

### Paramètres

- `mode` : chaîne de caractères (`horizontal` ou `vertical`)
- `cons_mini` : réel. Point de départ (en mm)
- `cons_maxi` : réel. Point d'arrivée (en mm)

### Description

Cette fonction permet de réaliser une opération de nivelage suivant le mode :

- `horizontal` : le godet se met alors en position verticale et le bras se déplace entre  $x_{min}=cons\_mini$  et  $x_{max}=cons\_maxi$  suivant un mouvement horizontal (c'est la partie « coupante » du godet qui réalise le nivelage, pour un fond de tranchée par exemple)
- `vertical` : le godet se met alors en position horizontale partie coupante vers la droite (vérin complètement rentré) et le bras se déplace entre  $y_{min}=cons\_mini$  et  $y_{max}=cons\_maxi$  suivant un mouvement vertical, pour une paroi de tranchée par exemple

## 4.13 LireVariable(variable)

### Paramètres

- `variable` : chaîne de caractères. Nom de la variable à lire

## Description

Cette fonction retourne la valeur de la variable spécifiée



## Liste des variables



- $\theta_{1cons}$  : consigne d'angle flèche (deg)
- $\theta_{2cons}$  : consigne d'angle balancier (deg)
- $\theta_{3cons}$  : consigne d'angle godet (deg)
- $\theta_1$  : angle flèche (deg)
- $\theta_2$  : angle balancier (deg)
- $\theta_3$  : angle godet (deg)
- $\theta_5$  : angle 4 barres (deg)
- $w_1$  : vitesse angulaire flèche (deg/s)
- $w_2$  : vitesse angulaire balancier (deg/s)
- $w_3$  : vitesse angulaire godet (deg/s)
- $w_5$  : vitesse angulaire 4 barres (deg/s)
- $x_c$  : consigne cartésienne en x (mm)
- $y_c$  : consigne cartésienne en y (mm)
- $\gamma$  : consigne cartésienne d'angle godet (deg)
- $x$  : position cartésienne calculée en x (mm)
- $y$  : position cartésienne calculée en y (mm)
- $\gamma$  : position cartésienne calculée d'angle godet (deg)
- $v_{1cons}$  : consigne de vitesse vérin de flèche (mm/s)
- $v_{2cons}$  : consigne de vitesse vérin de balancier (mm/s)
- $v_{3cons}$  : consigne de vitesse vérin de godet (mm/s)
- $v_1$  : vitesse vérin de flèche (mm/s)
- $v_2$  : vitesse vérin de balancier (mm/s)
- $v_3$  : vitesse vérin de godet (mm/s)
- $L_{1cons}$  : consigne de position vérin de flèche (mm)
- $L_{2cons}$  : consigne de position vérin de balancier (mm)
- $L_{3cons}$  : consigne de position vérin de godet (mm)
- $L_1$  : position vérin de flèche (mm)
- $L_2$  : position vérin de balancier (mm)
- $L_3$  : position vérin de godet (mm)
- $F_1$  : force vérin de flèche (N)
- $F_2$  : force vérin de balancier (N)
- $F_3$  : force vérin de godet (N)
- $i_1$  : courant vérin de flèche (A)
- $i_2$  : courant vérin de balancier (A)
- $i_3$  : courant vérin de godet (A)

- $u_1$  : tension de commande vérin de flèche (V)
- $u_2$  : tension de commande vérin de balancier (V)
- $u_3$  : tension de commande vérin de godet (V)
- $P_1$  : puissance électrique vérin de flèche (W)
- $P_2$  : puissance électrique vérin de balancier (W)
- $P_3$  : puissance électrique vérin de godet (W)
- $E_1$  : énergie vérin de flèche (J)
- $E_2$  : énergie vérin de balancier (J)
- $E_3$  : énergie vérin de godet (J)
- $PM_1$  : puissance mécanique vérin de flèche (W)
- $PM_2$  : puissance mécanique vérin de balancier (W)
- $PM_3$  : puissance mécanique vérin de godet (W)
- $\theta_{M1}$  : angle moteur vérin de flèche (rad)
- $\theta_{M2}$  : angle moteur vérin de balancier (rad)
- $\theta_{M3}$  : angle moteur vérin de godet (rad)
- $w_{M1}$  : vitesse moteur vérin de flèche (rad/s)
- $w_{M2}$  : vitesse moteur vérin de balancier (rad/s)
- $w_{M3}$  : vitesse moteur vérin de godet (rad/s)

## 5 Mise à jour du logiciel MyViz

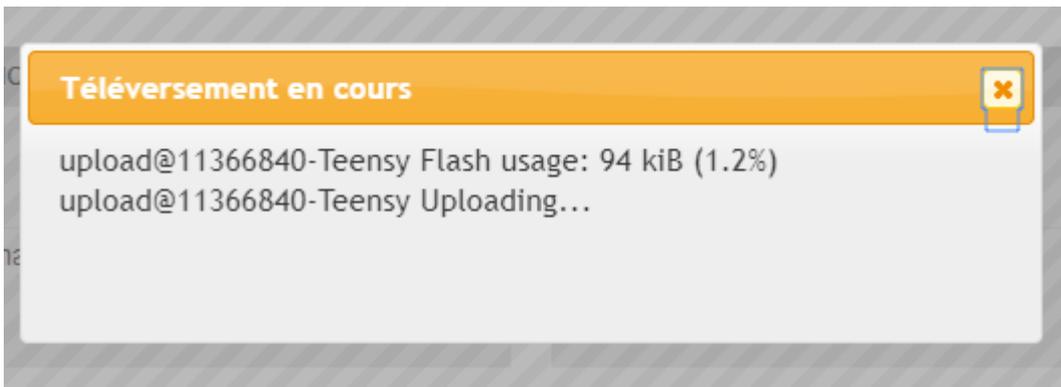
1. Télécharger la dernière version de MyViz en [cliquant ici](#).
2. Si une version antérieure de MyViz existe déjà sur votre ordinateur, ne pas la supprimer (pour l'instant).
3. Décompresser le .zip le plus haut possible dans l'arborescence de fichiers, idéalement à la racine (C:).
4. Copier les fichiers de licence (.lic) situés dans le répertoire principal de l'ancienne version pour les placer au même endroit (répertoire principal) de la nouvelle version.
5. L'ancienne version de MyViz peut alors être éventuellement supprimée (mais ce n'est pas indispensable de le faire).

## 6 Mise à jour du firmware du bras de pelleuseuse

### Version 0.12.9 minimum de MyViz

La procédure de mise à jour du firmware du bras de pelleuseuse est disponible à partir de la version 0.12.9 de MyViz. Pour mettre à jour MyViz, suivez la procédure de [mise à jour de MyViz](#).

1. Connecter l'ordinateur au bras de pelleuseuse et mettez le système sous tension.
2. Lancer la nouvelle version de MyViz en double-cliquant sur l'exécutable MyViz\_S2IDIDAC.
3. Sélectionner le port COM de l'ordinateur sur lequel est connecté le système dans le menu « Paramètres → Port série par défaut ».
4. Sélectionner le menu « Outils → Mises à jour de systèmes → Bras de pelleuseuse → Téléverser le firmware du bras de pelleuseuse ».
5. Un message indiquant l'évolution de la mise à jour apparaît assez furtivement (téléversement de la nouvelle version du firmware très rapide) :



6. Un nouveau message apparaît alors lorsque la mise à jour a réussi :



7. Si un message d'erreur tel que celui-ci apparaît, vérifier que le port COM sélectionné plus haut correspond bien à celui du bras de pelleuseuse :

### Téléversement échoué



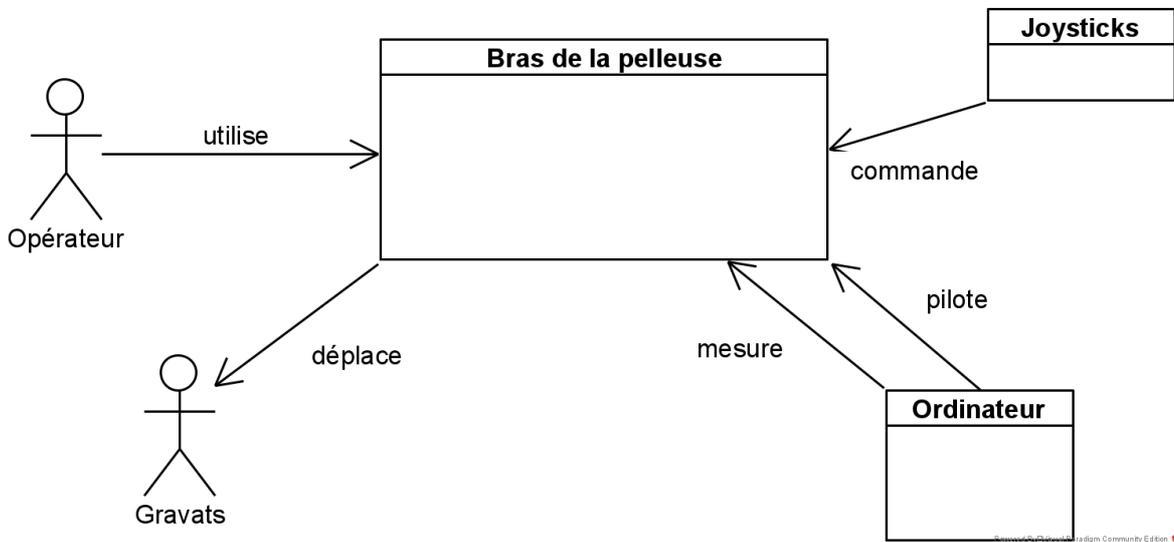
Veillez vérifier que la carte Teensy du banc est connectée, que l'alimentation est branchée et que vous avez sélectionné le bon port série dans le menu 'Paramètres --> Port série par défaut'

OK

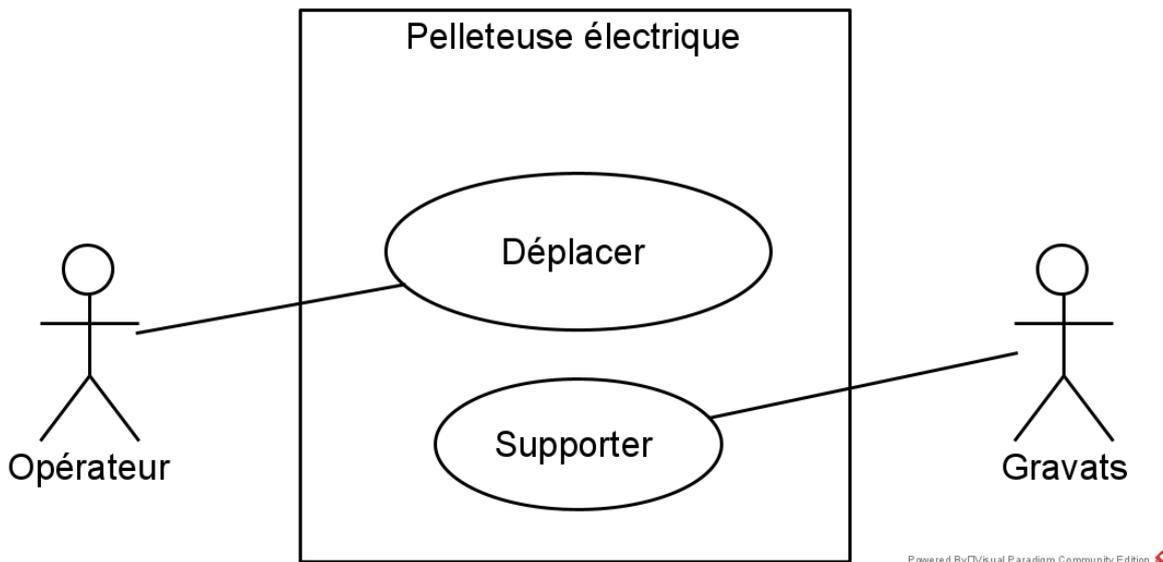
## II. Modèles

# 7 Modélisation structurelle

## 7.1 Diagramme de contexte

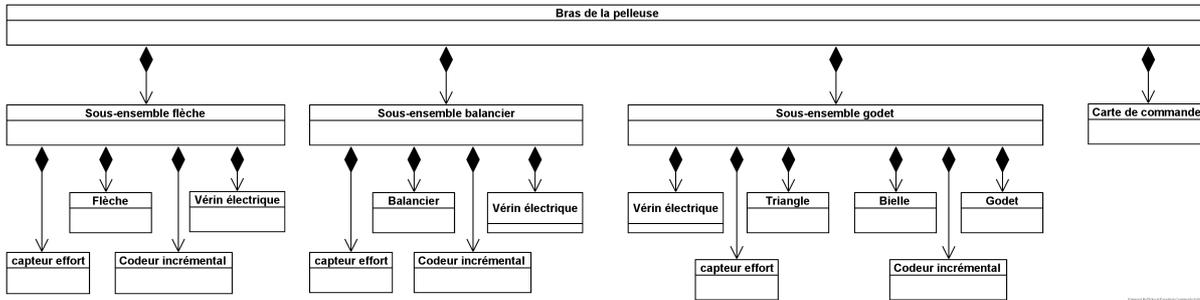


## 7.2 Diagramme de cas d'utilisation

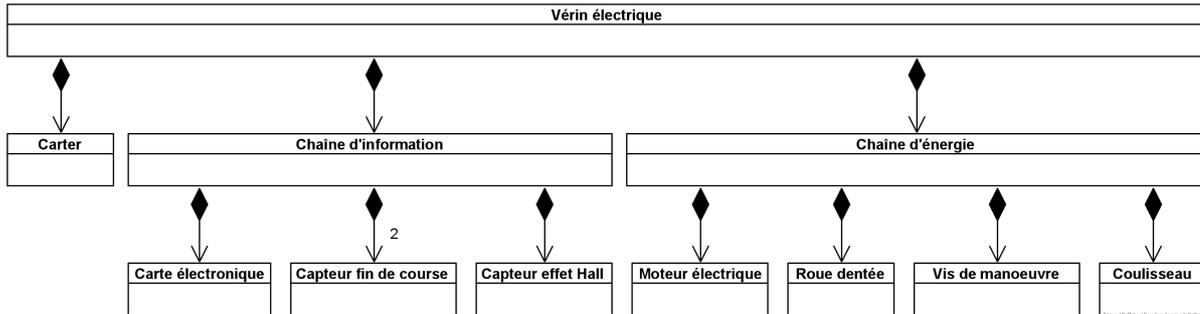


## 7.3 Diagramme de définition des blocs

### 7.3.1 Bras de la pelleuse

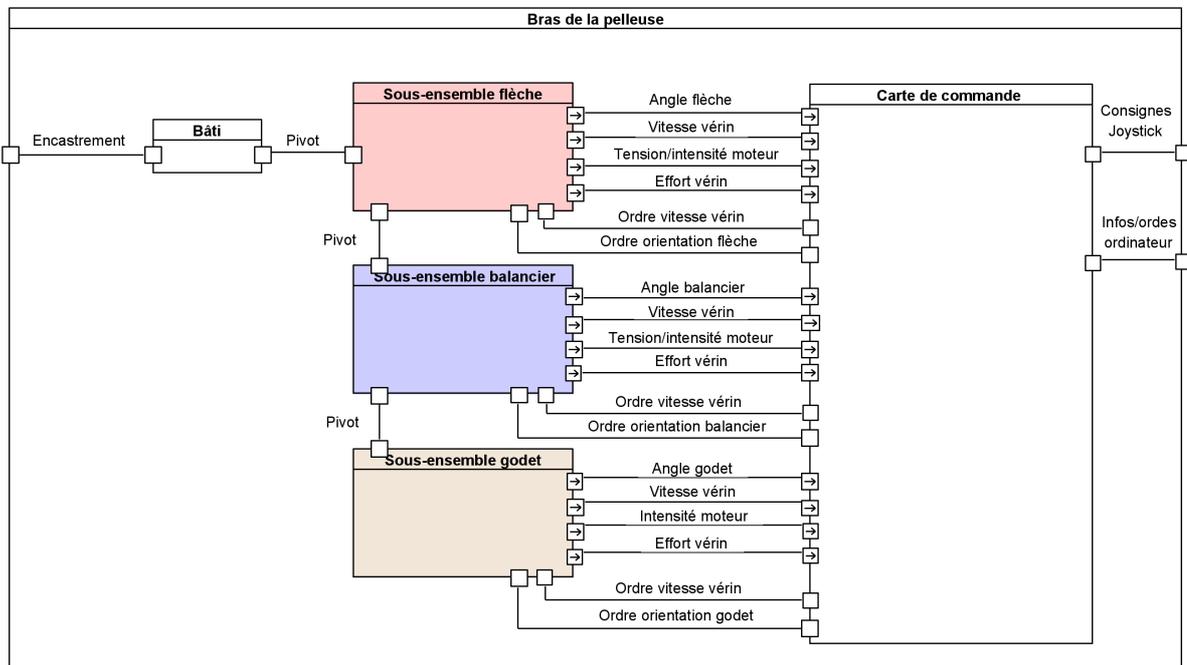


### 7.3.2 Vérin électrique

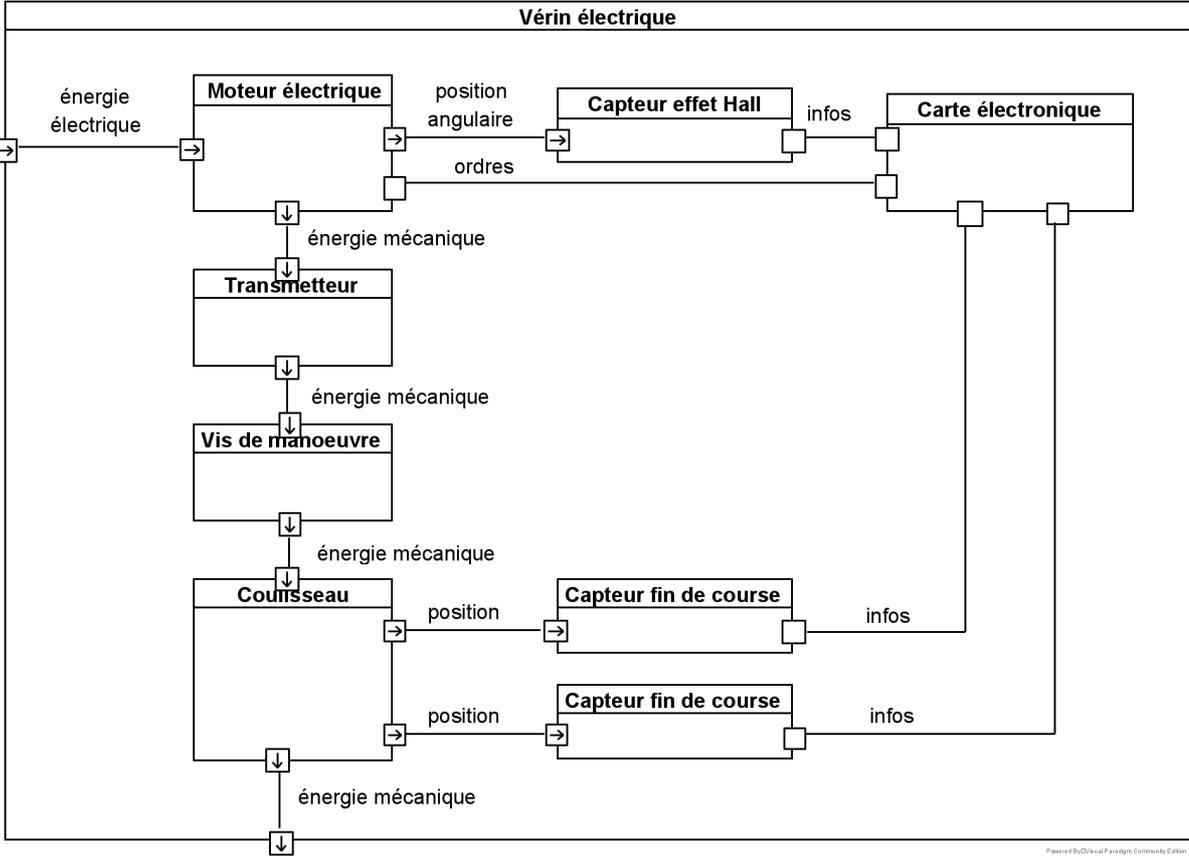


## 7.4 Diagramme de blocs internes

### 7.4.1 Bras de la pelleuse



### 7.4.2 Vérin électrique

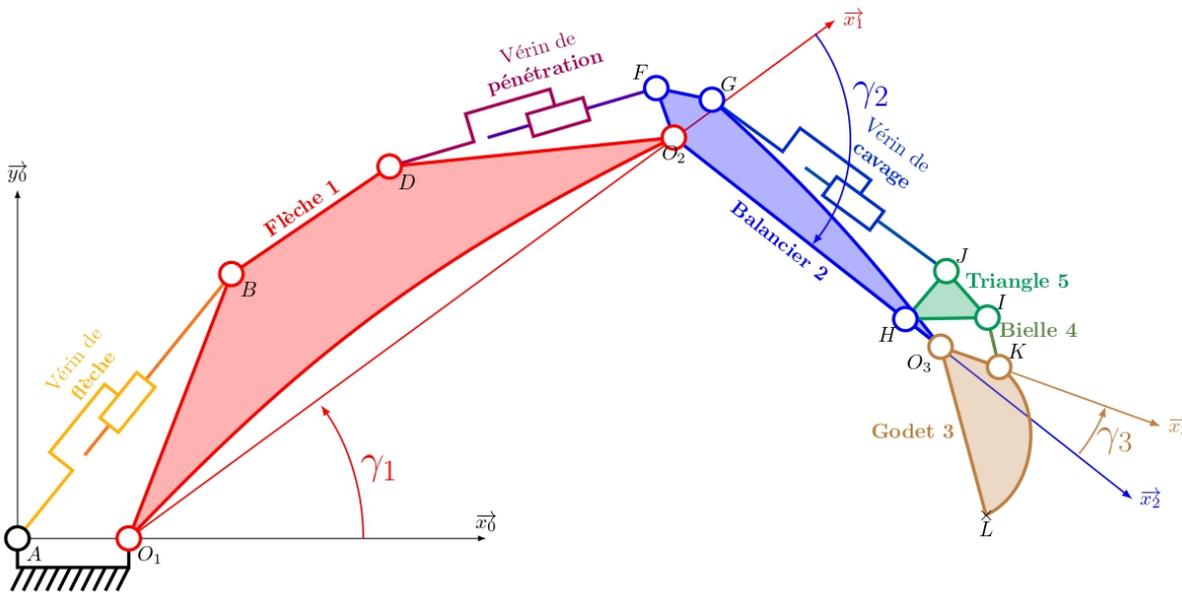


## II.I Modélisation géométrique

# 8 Modélisation plane

## 8.1 Schéma cinématique

La pelleuse, schématisée figure ci-dessous, est principalement constituée de six solides et de trois vérins qui actionnent respectivement la flèche 1, le balancier 2 et le godet 3 via le triangle 5 et la bielle 4.



Pour les activités de travaux pratique, il est souvent proposé une décomposition en sous-systèmes (SE) du bras :

- SE Bâti ;
- SE Flèche : vérin de flèche (corps et tige), flèche ;
- SE Balancier : vérin de pénétration (corps + tige), balancier ;
- SE Godet : vérin de cavage (corps + tige), triangle de renvoi, bielle, godet.

## 8.2 Paramétrage global

### 8.2.1 Paramétrage pièces

- Le repère  $(O_1, \vec{x}_0, \vec{y}_0, \vec{z}_0)$  est lié au socle 0.
- Le repère  $(O_1, \vec{x}_1, \vec{y}_1, \vec{z}_1)$  est lié à la flèche 1.
- Le repère  $(O_1, \vec{x}_2, \vec{y}_2, \vec{z}_2)$  est lié au balancier 2.
- Le repère  $(O_1, \vec{x}_3, \vec{y}_3, \vec{z}_3)$  est lié au godet 3.

La position des centres des liaisons dans les repères liés aux solides est définie par son vecteur position :

- $\vec{O_1O_2} = L_1 \vec{x}_1$
- $\vec{O_2O_3} = L_2 \vec{x}_2$

- $\overrightarrow{0_3K} = L_3 \vec{x}_3$

### 8.2.2 Paramétrage liaisons

Les trois liaisons pivot montées en série sont définies par les coordonnées dites articulaires de la pelleuse robot :

- $\gamma_1 = (\vec{x}_0, \vec{x}_1) = (\vec{y}_0, \vec{y}_1)$
- $\gamma_2 = (\vec{x}_1, \vec{x}_2) = (\vec{y}_1, \vec{y}_2)$
- $\gamma_3 = (\vec{x}_2, \vec{x}_3) = (\vec{y}_2, \vec{y}_3)$

### 8.3 Hypothèse générales

La pelleuse évolue dans le plan  $(O_0, \vec{x}_0, \vec{y}_0)$  et en conséquence tous les repères orthonormés directs utilisés sont tels que  $\vec{z}_0 = \vec{z}_1 = \vec{z}_2 = \vec{z}_3$ .

Toutes les liaisons sont géométriquement parfaites et sans frottement.

## 9 Modèle géométrique direct

### Info

Le modèle géométrique direct (MGD) décrit la position que prend le segment terminal de la structure (effecteur) lorsque la valeur des variables articulaires est connue, c'est-à-dire pour une configuration donnée de la structure.

Ce modèle est constitué de l'expression des coordonnées du repère lié au segment terminal dans le repère lié au solide de référence exprimé en fonction des coordonnées articulaires.

La situation de l'effecteur, ici le godet 3, est définie :

- par la position du point  $K$ ,  $\overrightarrow{O_1K} = x\vec{x}_0 + y\vec{y}_0$ ;
- et par l'orientation du godet 3 dans le repère lié au socle 0,  $\gamma = (\vec{x}_0, \vec{x}_3) = (\vec{y}_0, \vec{y}_3)$ .

### MGD

$$\begin{cases} x = L_1 \cos \gamma_1 + L_2 \cos (\gamma_1 + \gamma_2) + L_3 \cos (\gamma_1 + \gamma_2 + \gamma_3) \\ y = L_1 \sin \gamma_1 + L_2 \sin (\gamma_1 + \gamma_2) + L_3 \sin (\gamma_1 + \gamma_2 + \gamma_3) \\ \gamma = \gamma_1 + \gamma_2 + \gamma_3 \end{cases}$$

## 10 Modèle géométrique inverse

### Info

Si nous désirons piloter la pelleuse il faut définir le modèle géométrique inverse (MGI). On connaît la position et l'orientation que doit atteindre l'effecteur par rapport au solide de référence. Pour commander l'effecteur il faut connaître la valeur des coordonnées articulaires pour atteindre cette situation c'est-à-dire trouver l'expression des coordonnées articulaires en fonction de la position et de l'orientation de l'organe terminal.

Le MGI consiste à calculer les coordonnées articulaires,  $(\gamma_1, \gamma_2, \gamma_3)$ , correspondant à une situation donnée du godet 3,  $(x, y, \gamma)$ . Lorsqu'elle existe la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue le MGI.

D'après le MGD, posons  $A$  et  $B$  connus tels que :

$$\begin{cases} A = x - L_3 \cos \gamma = L_1 \cos \gamma_1 + L_2 \cos (\gamma_1 + \gamma_2) \\ B = y - L_3 \sin \gamma = L_1 \sin \gamma_1 + L_2 \sin (\gamma_1 + \gamma_2) \end{cases}$$

$$\begin{cases} \gamma_2 = \pm \arccos \left( \frac{A^2 + B^2 - (L_1^2 + L_2^2)}{2L_1L_2} \right) \\ \gamma_1 = \operatorname{atan2} \left( \frac{B(L_1 + L_2 \cos \gamma_2) - AL_2 \sin \gamma_2}{A(L_1 + L_2 \cos \gamma_2) + BL_2 \sin \gamma_2} \right) \\ \gamma_3 = \gamma - \gamma_1 - \gamma_2 \end{cases}$$

Tous les détails des calculs 

# 11 Cas du porteur seul

On considère le porteur seul constitué de la flèche 1 et du balancier 2 si le godet 3 est piloté indépendamment.

Soit la position du point  $O_3$  dans le repère lié au socle 0 :  $\overrightarrow{O_1 O_3} = x_3 \vec{x}_0 + y_3 \vec{y}_0$ .

## 11.1 MGD

Le MGD du bras complet se simplifie en :

$$\begin{cases} x_3 = L_1 \cos \gamma_1 + L_2 \cos (\gamma_1 + \gamma_2) \\ y_3 = L_1 \sin \gamma_1 + L_2 \sin (\gamma_1 + \gamma_2) \end{cases}$$

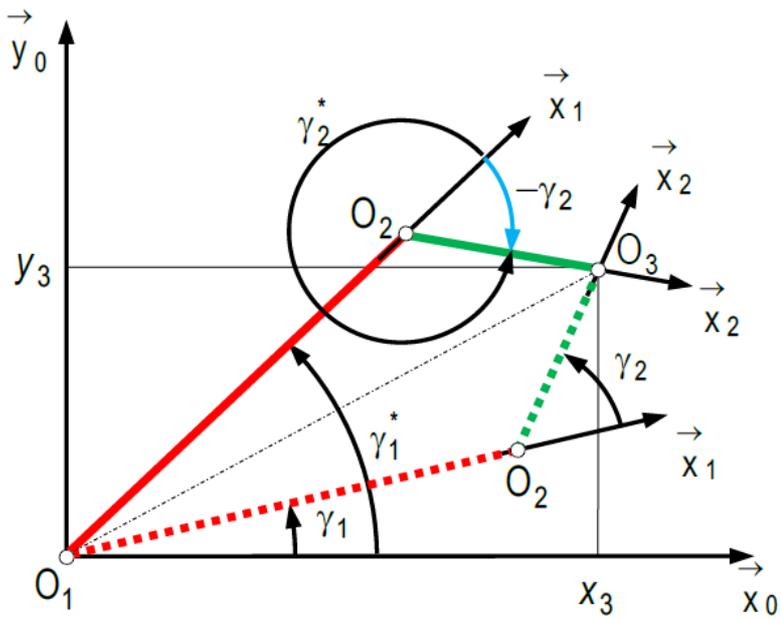
## 11.2 MGI

Le MGI du bras complet devient

$$\begin{cases} \gamma_2 = \pm \arccos \left( \frac{x_3^2 + y_3^2 - (L_1^2 + L_2^2)}{2L_1 L_2} \right) \\ \gamma_1 = \operatorname{atan2} \left( \frac{y_3 (L_1 + L_2 \cos \gamma_2) - x_3 L_2 \sin \gamma_2}{x_3 (L_1 + L_2 \cos \gamma_2) + y_3 L_2 \sin \gamma_2} \right) \end{cases}$$

## 11.3 Deux solutions

La figure ci-dessous montre que le problème possède deux solutions correspondant à deux postures différentes du bras 2 (sous réserve qu'il n'y ait pas de butées sur les articulations).



La position du point  $O_3$  dans le repère lié au socle 0 peut être atteinte avec le couple d'angles  $(\gamma_1, \gamma_2)$  ou le couple  $(\gamma_1^*, \gamma_2^* = -\gamma_2)$

#### Remarques

- L'angle  $\gamma_2$  peut aussi être trouvé en appliquant le théorème de Pythagore généralisé
- Connaissant  $\gamma_2$ , le système d'équations à deux inconnues  $\sin \gamma_2$  et  $\cos \gamma_1$  peut être résolu en utilisant la méthode de Cramer via le déterminant du système.

Tous les détails des calculs 

# 12 Génération de mouvement point à point avec une trajectoire imposée

## 12.1 Équation d'une trajectoire du point $O_3$

Considérons le cas *simple* d'un déplacement du point  $O_3$  suivant le segment  $[O_{3i}O_{3f}]$  illustré sur la figure ci-dessous.

L'équation de la trajectoire rectiligne  $y_3 = a x_3 + b$  est définie par

- la pente  $a$  de la droite :  $a = \frac{y_{3f} - y_{3i}}{x_{3f} - x_{3i}}$
- l'ordonnée à l'origine :  $b = \frac{x_{3f} y_{3i} - x_{3i} y_{3f}}{x_{3f} - x_{3i}}$

## 12.2 Application numérique

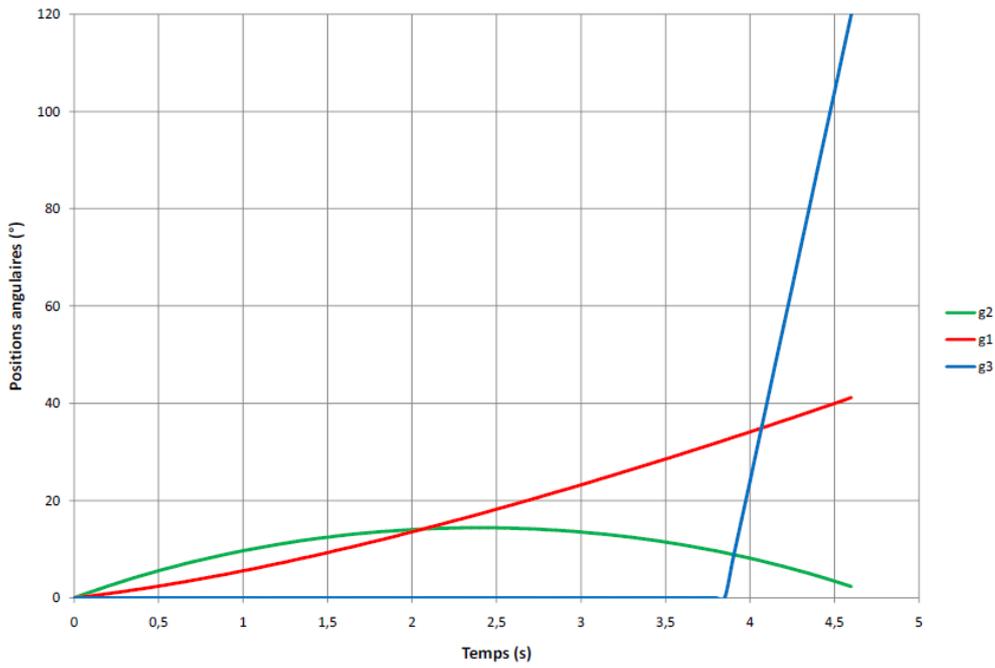
Applications numériques :

Paramètres	Valeurs (mm)
$L_1$	520
$L_2$	300
$x_{3i}$	530
$y_{3i}$	490
$x_{3f}$	715
$y_{3f}$	10

La durée de déplacement est arbitrairement de 4,6 s.

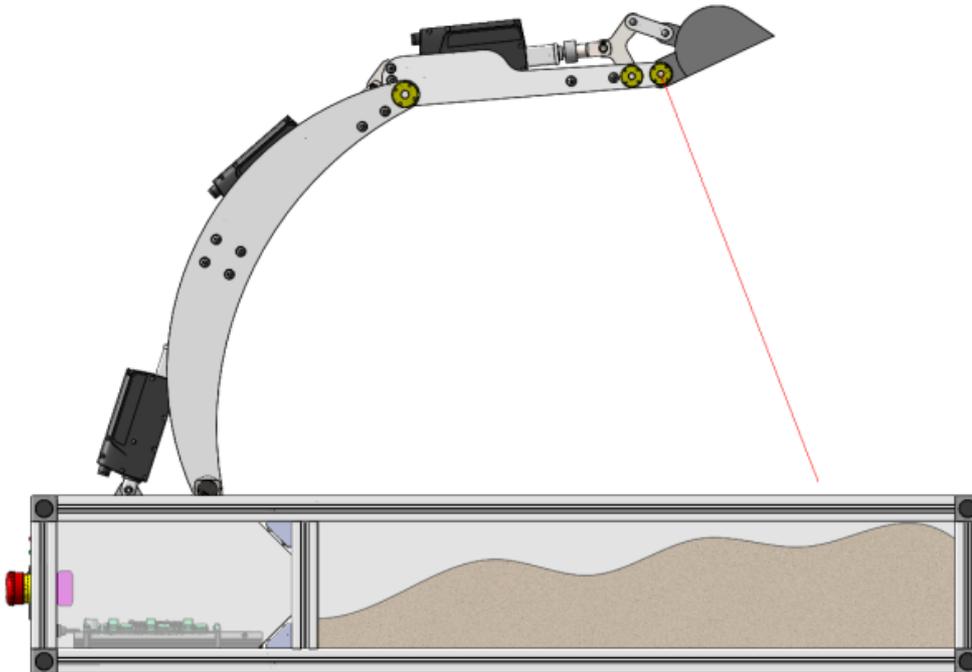
Les expressions de  $\gamma_1$  et  $\gamma_2$  du MGI permettent d'obtenir les courbes ci-dessous. L'angle  $\gamma_3$  est géré indépendamment des deux autres.

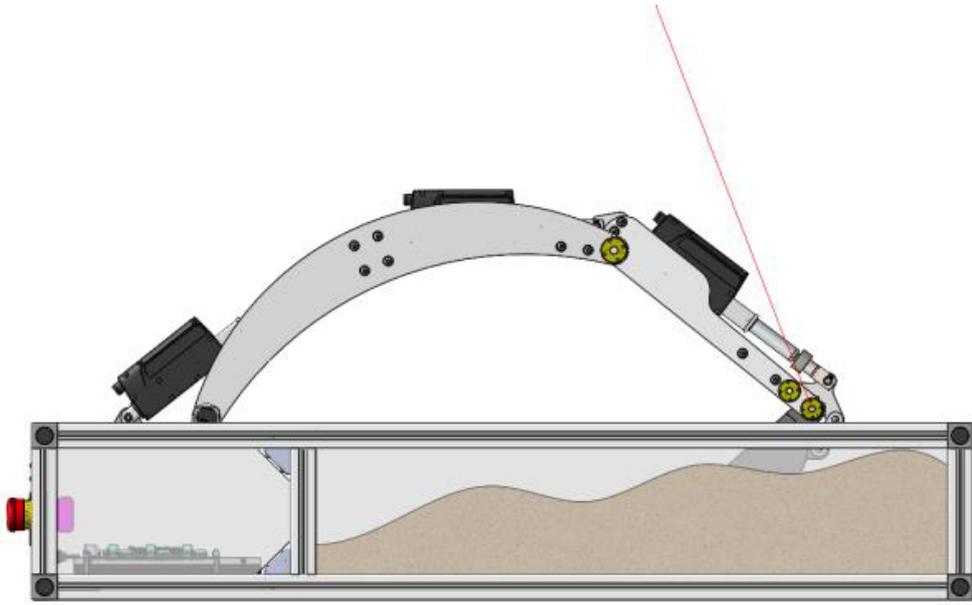
Evolution des angles  $\gamma_1$ ,  $\gamma_2$  et  $\gamma_3$  en fonction du temps (Méca3D)



Tous les calculs dans l'onglet "angles" 

Les figures suivantes donnent le résultat dans Solidworks.

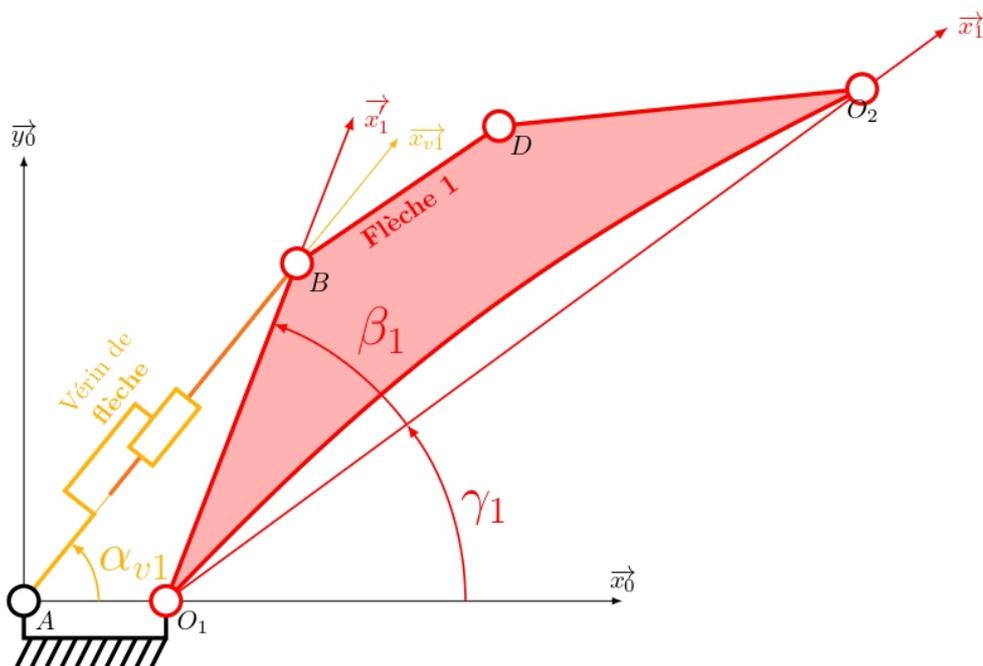




## II.I.I Loi E/S

# 13 Loi E/S Flèche

## 13.1 Schéma cinématique



## 13.2 Paramétrage

$$\overrightarrow{AO_1} = c \overrightarrow{x_0}, \quad \overrightarrow{AB} = l_1 \overrightarrow{x_{v1}}, \quad \overrightarrow{O_1B} = \lambda_1 \overrightarrow{x'_1}$$

## 13.3 Fermeture géométrique

$$\overrightarrow{AO_1} + \overrightarrow{O_1B} = \overrightarrow{AB} \Rightarrow \begin{cases} c + \lambda_1 \cos(\gamma_1 + \beta_1) = l_1 \cos \alpha_{v1} \\ 0 + \lambda_1 \sin(\gamma_1 + \beta_1) = l_1 \sin \alpha_{v1} \end{cases}$$

## 13.4 Modèle géométrique

direct

$$\gamma_1 = \arccos\left(\frac{l_1^2 - c^2 - \lambda_1^2}{2c\lambda_1}\right) - \beta_1$$

inverse

$$\lambda_1 = \sqrt{c^2 + l_1^2 + 2cl_1 \cos(\gamma_1 + \beta_1)}$$

### Info

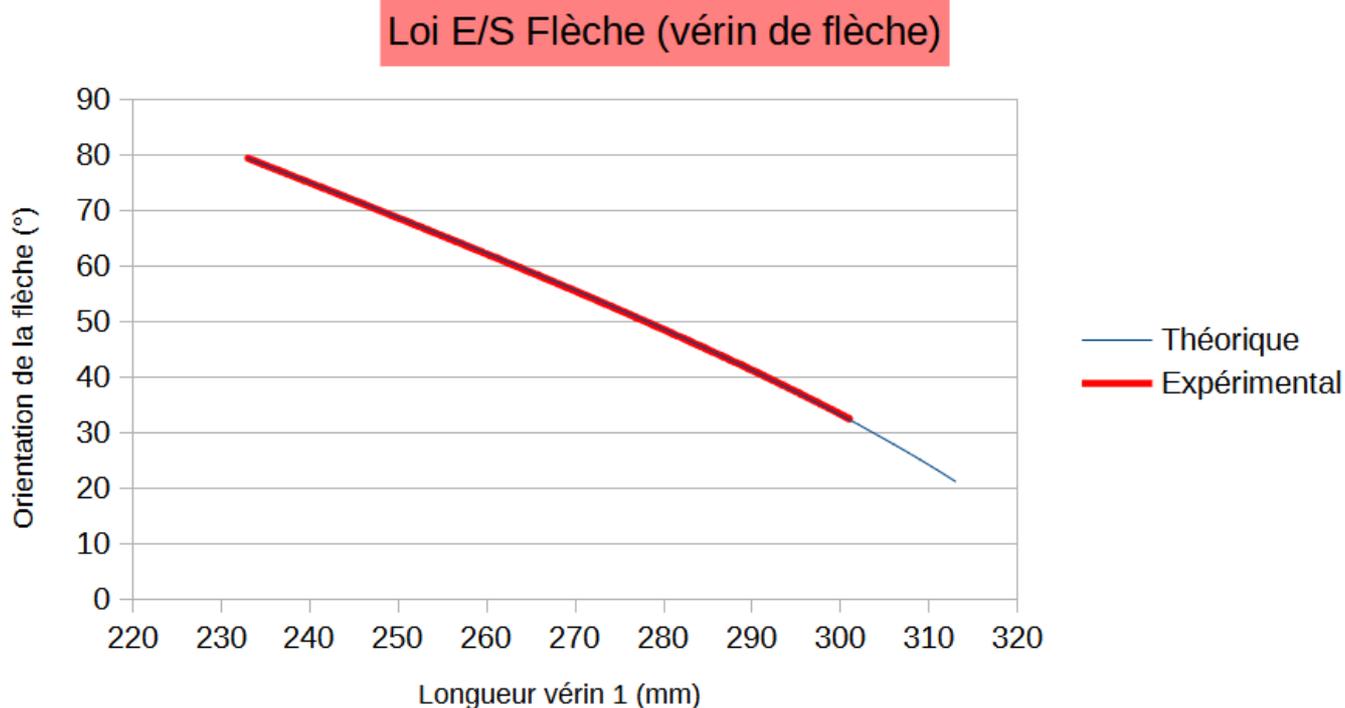
$\beta_1$  est un paramètre géométrique de la flèche  voir géométrie des pièces.

$$\beta_1 = \arctan\left(\frac{122}{214}\right) \approx 29,69^\circ$$

## 13.5 Résultats de simulation

Le vérin de flèche a une course de  $\Delta\lambda_1 = 80$  mm et  $233 \text{ mm} \leq l_1 \leq 313$  mm.

La figure ci-dessous présente les résultats de simulation du modèle géométrique et des mesures réalisées sur le bras de pelleuse.





direct

$$\gamma_2 = \arccos\left(\frac{l_2^2 - \lambda_1'^2 - \delta^2}{2\lambda_1' \delta}\right) - \beta_1' - \beta_2$$

inverse

$$l_2 = \sqrt{\lambda_1'^2 + \delta^2 + 2\lambda_1' \delta \cos(\beta_1' + \gamma_2 + \beta_2)}$$

#### Définition de $\beta_2$

Dans la paramétrage des pièces, l'angle  $\beta_2$  est défini par  $\beta_2 = (\vec{y}_2, \vec{x}'_2)$ .

Ici, dans le modèle proposé, on a pris  $\beta_2 = (\vec{x}_2, \vec{x}'_2)$ . Pour les calculs, il conviendra d'ajouter  $90^\circ$  à la valeur de  $\beta_2$  donnée.

#### Info

- $\beta_1'$  est un paramètre géométrique de la flèche voir [géométrie des pièces](#).

$$\beta_1 = \text{atan}\left(\frac{122}{226}\right) \approx 28,36^\circ$$

- $\beta_2$  est un paramètre géométrique du balancier voir [géométrie des pièces](#).

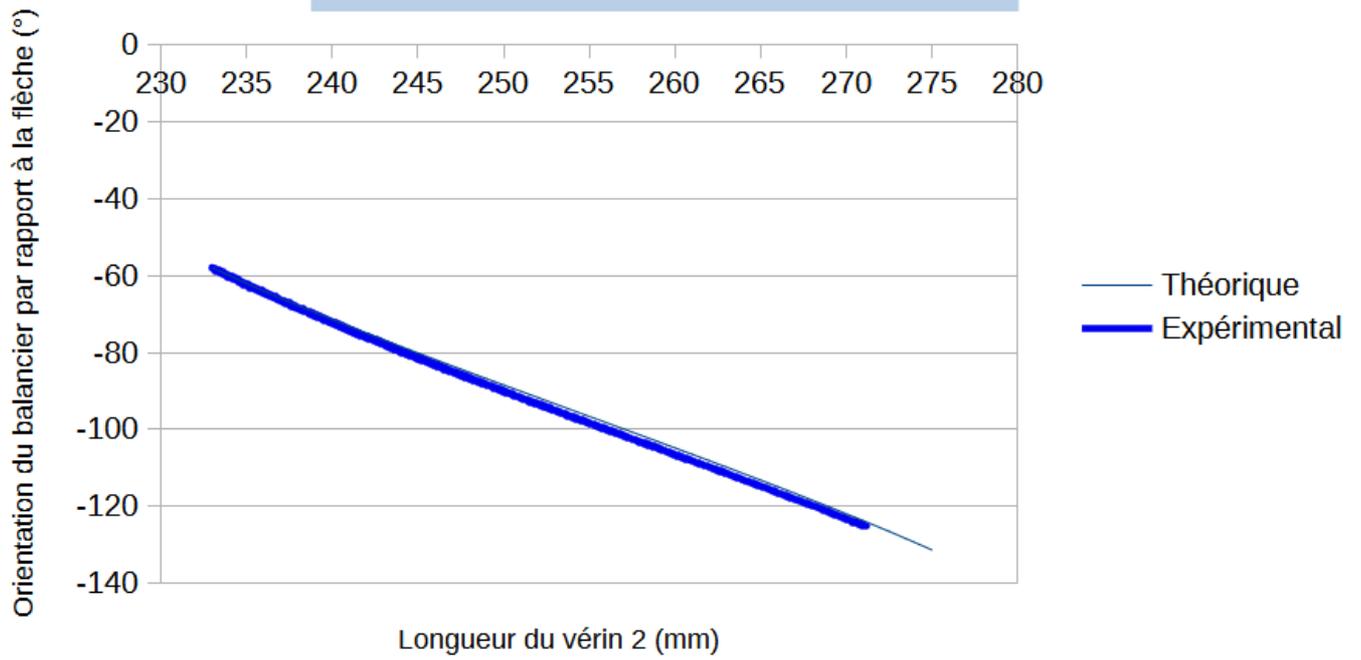
$$\beta_2 = 180 - \text{atan}\left(\frac{9}{34}\right) \approx 165,17^\circ$$

## 14.5 Résultats de simulation

Le vérin de pénétration a une course de  $\Delta\lambda_2 = 42 \text{ mm}$  et  $233 \text{ mm} \leq l_2 \leq 275 \text{ mm}$ .

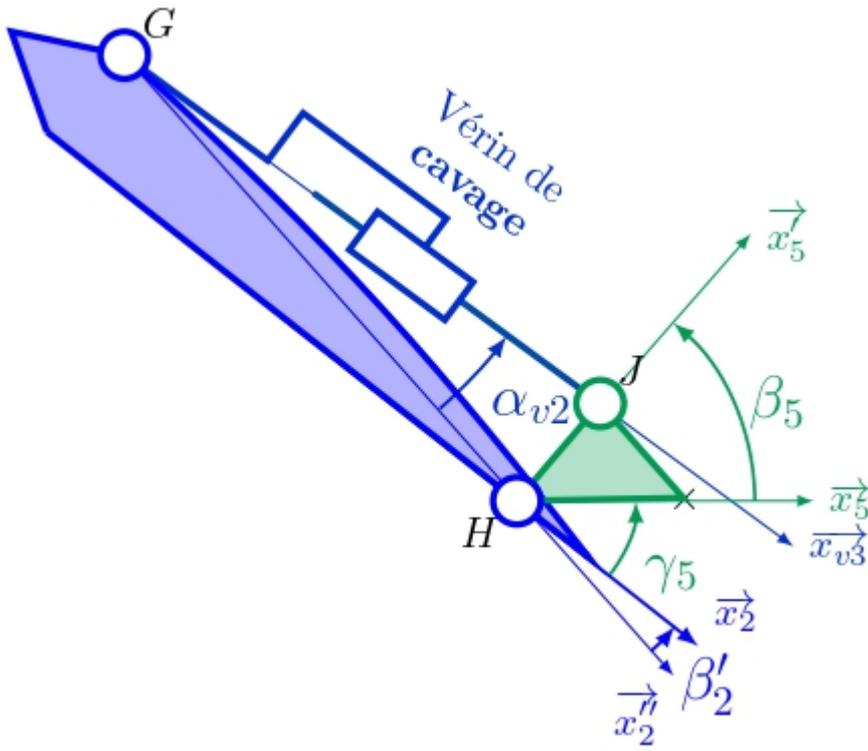
La figure ci-dessous présente les résultats de simulation du modèle géométrique et des mesures réalisées sur le bras de pelleuse.

### Loi E/S Balancier (vérin de pénétration)

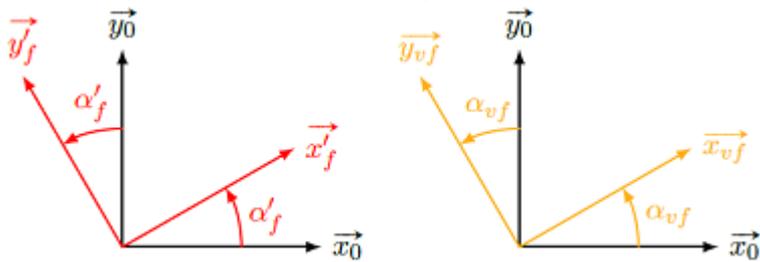


# 15 Loi E/S triangle

## 15.1 Schéma cinématique



## 15.2 Paramétrage



$$\vec{GH} = \lambda_2 \vec{x}_2'', \quad \vec{GJ} = l_3 \vec{x}_{v3}, \quad \vec{HJ} = l \vec{x}'_5$$

## 15.3 Fermeture géométrique

$$\vec{GH} + \vec{HJ} = \vec{GJ} \Rightarrow \begin{cases} \lambda_2 + l \cos(\beta'_2 + \gamma_5 + \beta_5) = l_3 \cos \alpha_{v3} \\ 0 + l \sin(\beta'_2 + \gamma_5 + \beta_5) = l_3 \sin \alpha_{v3} \end{cases}$$

## 15.4 Modèle géométrique

direct

$$\gamma_5 = \arccos\left(\frac{l_3^2 - \lambda_2^2 - l^2}{2\lambda_2 l}\right) - \beta'_2 - \beta_5$$

inverse

$$l_3 = \sqrt{\lambda_2^2 + l^2 + 2\lambda_2 l \cos(\beta'_2 + \gamma_5 + \beta_5)}$$

### **i** Info

- $\beta'_2$  est un paramètre géométrique du balancier voir [géométrie des pièces](#).

$$\beta'_2 = \operatorname{atan}\left(\frac{30}{266 - 6}\right) \approx 6,58^\circ$$

- $\beta_5$  est un paramètre géométrique du triangle voir [géométrie des pièces](#).

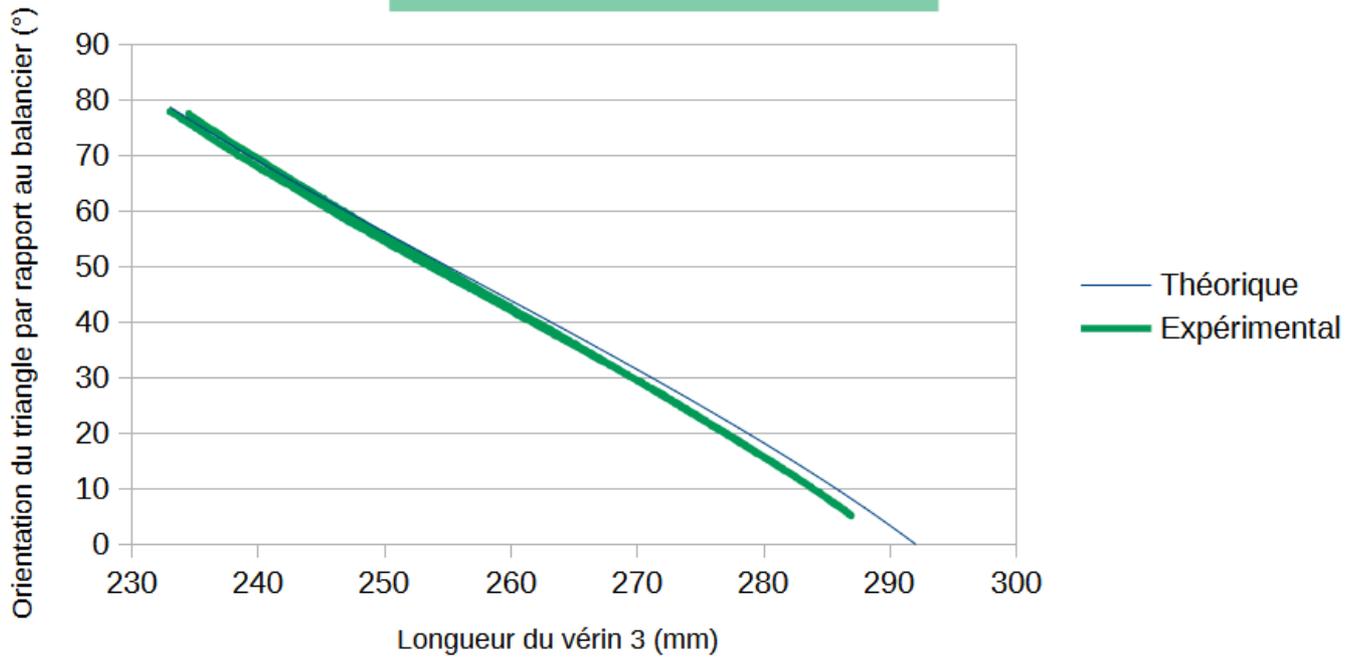
$$\beta_5 = 180 - \operatorname{asin}\left(\frac{34}{64}\right) \approx 32,09^\circ$$

## 15.5 Résultats de simulation

Le vérin de cavage a une course telle que  $233,55 \text{ mm} \leq l_3 \leq 292,03 \text{ mm}$ .

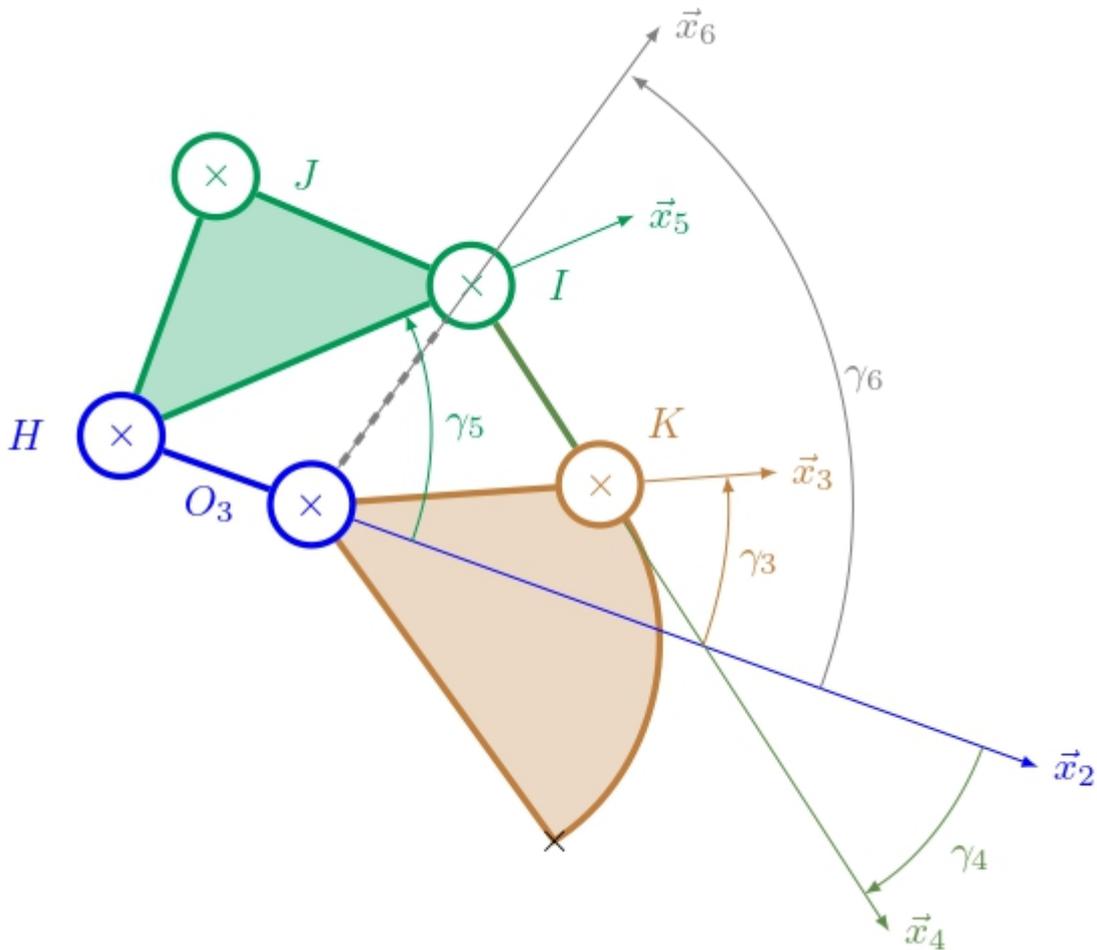
La figure ci-dessous présente les résultats de simulation du modèle géométrique et des mesures réalisées sur le bras de pelleuse.

### loi E/S Triangle (vérin cavage)



# 16 Loi E/S godet

## 16.1 Schéma cinématique



### 🔥 Système 4 barres

L'entraînement du godet par le triangle est réalisé par 4 barres montées en liaisons pivot : **système 4 barres**.

Pour établir la loi E/S de celui-ci, une méthode consiste à diviser ce système en deux sous-systèmes :

- le premier défini par les barres  $[HI]$ ,  $[HO_3]$  et une barre virtuelle  $[O_3I]$ . L'entrée sera l'orientation du triangle  $\gamma_5$  qui entraînera la modification de l'orientation  $\varphi$  et de longueur  $s$  de  $[O_3I]$ ;
- le deuxième défini par les barres  $[O_3K]$ ,  $[IK]$  et la barre  $[O_3I]$ . Cette dernière est l'entrée du sous-système est entraîné, entre autres, l'orientation du godet  $\gamma_3$ .

## 16.2 Premier sous-système

### 16.2.1 Paramétrage

$$\overrightarrow{HO_3} = (L_2 - k) \vec{x}_2, \quad \overrightarrow{O_3I} = s \vec{x}_6, \quad \overrightarrow{HI} = L_5 \vec{x}_5$$

### 16.2.2 Fermeture géométrique

$$\overrightarrow{HO_3} + \overrightarrow{O_3I} = \overrightarrow{HI} \Rightarrow \begin{cases} L_2 - k + s \cos \varphi = L_5 \cos \gamma_5 \\ 0 + s \sin \varphi = L_5 \sin \gamma_5 \end{cases}$$

### 16.2.3 Modèle géométrique

$$s = \sqrt{L_5^2 + (L_2 - l)^2 - 2L_5(L_2 - k) \cos \gamma_5}$$

et

$$\sin \varphi = \frac{L_5 \sin \gamma_5}{s}$$

## 16.3 Deuxième sous-système

### 16.3.1 Paramétrage

$$\overrightarrow{O_3I} = s \vec{x}_6, \quad \overrightarrow{IK} = L_4 \vec{x}_4, \quad \overrightarrow{O_3K} = L_3 \vec{x}_3$$

### 16.3.2 Fermeture géométrique

$$\overrightarrow{O_3I} + \overrightarrow{IK} = \overrightarrow{O_3K} \Rightarrow \begin{cases} s \cos \varphi + L_4 \cos \gamma_4 = L_3 \cos \gamma_3 \\ s \sin \varphi + L_4 \sin \gamma_4 = L_3 \sin \gamma_3 \end{cases}$$

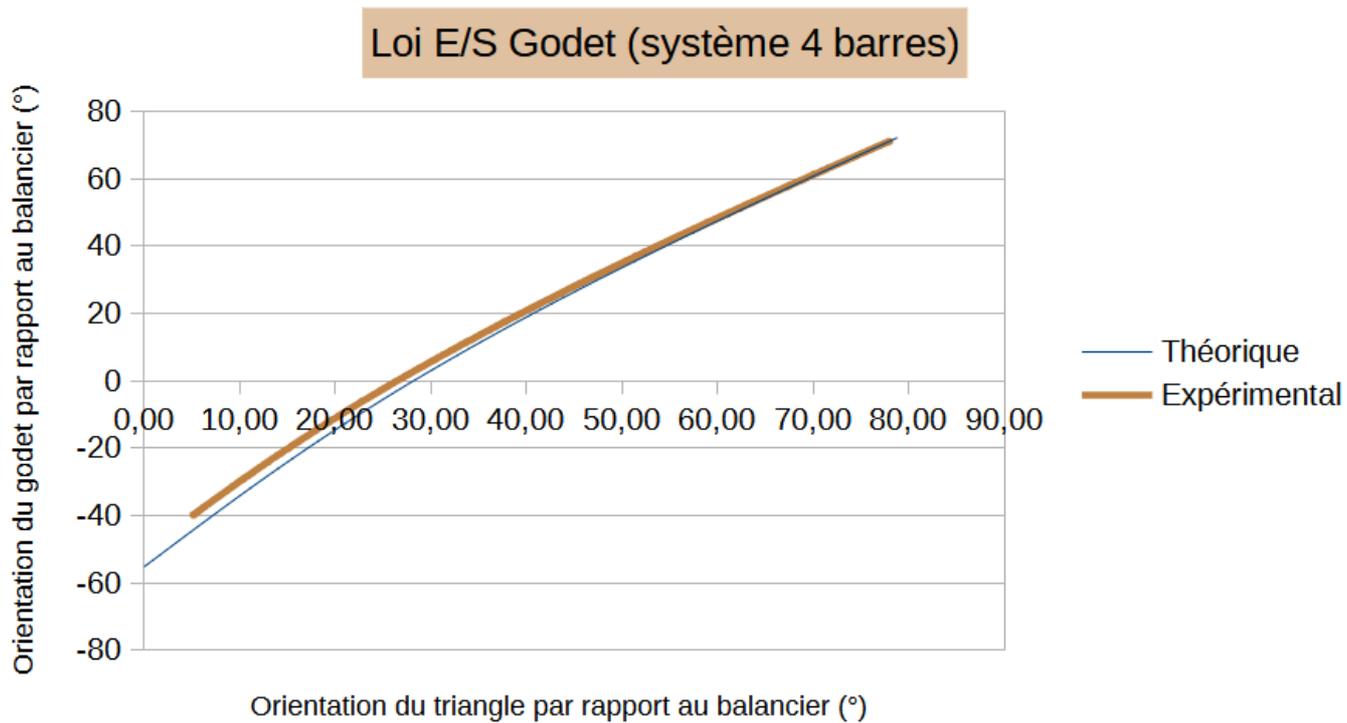
### 16.3.3 Modèle géométrique

$$\gamma_3 = \varphi - \arccos \left( \frac{s^2 + L_3^2 - L_4^2}{2 s L_3} \right)$$

## 16.4 Résultats de simulation

Le course du vérin de cavage implique un débattement angulaire du triangle tel que  $0^\circ \leq \gamma_5 \leq 78^\circ$ .

La figure ci-dessous présente les résultats de simulation du modèle géométrique et des mesures réalisées sur le bras de pelleuse.



## II.II Modélisation des actions mécaniques

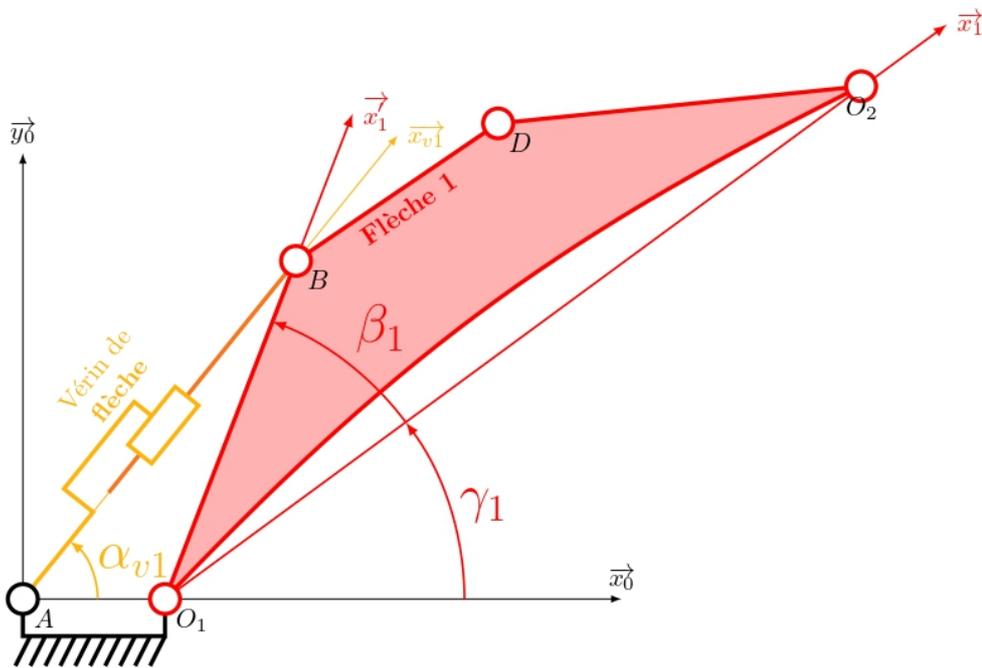
# 17 Vérin de flèche

## Objectif

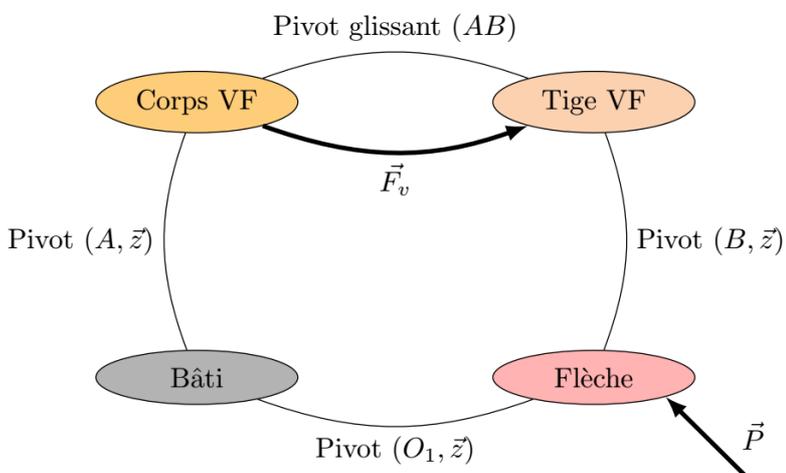
Déterminer l'effort dans le vérin de flèche

## 17.1 Paramétrage

### 17.1.1 Schéma cinématique



### 17.1.2 Graphe de structure



### 17.1.3 Hypothèses

Dans les calculs ci-dessous, la masse du vérin de flèche est négligée devant les autres masses mises en jeu.

$\vec{P}$  correspond au poids de l'ensemble du bras sans le vérin de flèche appliqué au centre de masse  $G_S$ . La position du centre de masse  $G_S$  dépend principalement de l'orientation du balancier par rapport à la flèche, du godet par rapport au balancier et de la masse embarquée dans le godet. Soit  $\overrightarrow{O_1G_S} = x_{G_S}\vec{x}_0 + y_{G_S}\vec{y}_0$ .

Les liaisons sont supposées parfaites.

## 17.2 Détermination de l'effort dans la tige du vérin flèche

### 17.2.1 Inventaire des actions mécaniques

Dans cette étude, on distingue :

- les actions mécaniques transmissibles dans les liaisons

$$\bullet \{ \mathcal{T}_{\text{bâti} \rightarrow \text{CVF}} \} = \begin{pmatrix} A_x & - \\ A_y & - \\ - & 0 \end{pmatrix}_{A, -, -, \vec{z}_0}$$

$$\bullet \{ \mathcal{T}_{\text{bâti} \rightarrow \text{F}} \} = \begin{pmatrix} O_{1x} & - \\ O_{1y} & - \\ - & 0 \end{pmatrix}_{0_1, -, -, \vec{z}_0}$$

$$\bullet \{ \mathcal{T}_{\text{CVF} \rightarrow \text{TVF}} \} = \begin{pmatrix} F_v & - \\ Y & - \\ - & N \end{pmatrix}_{A, \mathcal{B}_{v1}}$$

$$\bullet \{ \mathcal{T}_{\text{TVF} \rightarrow \text{F}} \} = \begin{pmatrix} B_x & - \\ B_y & - \\ - & 0 \end{pmatrix}_{B, -, -, \vec{z}_0}$$

- l'action de la pesanteur sur l'ensemble du bras sans le vérin de flèche

$$\bullet \{ \mathcal{T}_{\text{pes} \rightarrow \text{F}} \} = \begin{pmatrix} 0 & - \\ -P & - \\ - & 0 \end{pmatrix}_{G_S, \mathcal{B}_0}$$

### 17.2.2 Isolement de l'ensemble vérin flèche

L'ensemble vérin flèche est constitué du corps (CVF) et de la tige (TVF) du vérin.

Cet ensemble est en équilibre sous l'action de deux glisseurs aux points  $A$  et  $B$ .

D'après le principe fondamental de la statique, les résultantes de ces deux glisseurs sont portées par la direction  $\vec{x}_{v1}$ .

Ainsi :

- $\{\mathcal{T}_{\text{bâti} \rightarrow \text{CVF}}\} = \begin{pmatrix} \mathcal{R}_A & - \\ 0 & - \\ - & 0 \end{pmatrix}_{A, \mathcal{B}_{v1}}$
- $\{\mathcal{T}_{\text{TVF} \rightarrow \text{F}}\} = \begin{pmatrix} \mathcal{R}_B & - \\ 0 & - \\ - & 0 \end{pmatrix}_{B, \mathcal{B}_{v1}}$

### 17.2.3 Isolement de la flèche

La flèche est en équilibre sous l'action de 3 glisseurs. Pour déterminer l'effort dans la tige de vérin, il convient d'écrire le **théorème du moment statique au point  $O_1$  projeté sur la direction  $\vec{z}_0$**  :

$$\begin{aligned} \overrightarrow{\mathcal{M}}_{O_1, \text{bâti} \rightarrow \text{F}} \cdot \vec{z}_0 + \overrightarrow{\mathcal{M}}_{O_1, \text{pes} \rightarrow \text{F}} \cdot \vec{z}_0 + \overrightarrow{\mathcal{M}}_{O_1, \text{TVF} \rightarrow \text{F}} \cdot \vec{z}_0 &= 0 \\ 0 + \left( \overrightarrow{O_1 G_S} \wedge -P\vec{y}_0 \right) \cdot \vec{z}_0 + \left( \overrightarrow{O_1 B} \wedge \mathcal{R}_B \overrightarrow{x_{v1}} \right) \cdot \vec{z}_0 &= 0 \\ 0 + \left( (x_{G_S} \vec{x}_0 + y_{G_S} \vec{y}_0) \wedge -P\vec{y}_0 \right) \cdot \vec{z}_0 + \mathcal{R}_B L_1 \left( \vec{x}'_1 \wedge \vec{x}_{v1} \right) \cdot \vec{z}_0 &= 0 \\ -Px_{G_S} - \mathcal{R}_B L_1 \sin(\alpha_{v1} - \beta_1 - \gamma_1) &= 0 \end{aligned}$$

Soit donc

$$\mathcal{R}_B = -\frac{Px_{G_S}}{L_1 \sin(\alpha_{v1} - \beta_1 - \gamma_1)}$$

### 17.2.4 Isolement de la tige du vérin

L'application du **théorème de la résultante statique en projection sur  $\vec{x}_{v1}$**  donne :

$$F_v - \mathcal{R}_B = 0$$

soit donc

$$F_v = -\frac{Px_{G_S}}{L_1 \sin(\alpha_{v1} - \beta_1 - \gamma_1)}$$

## II.III Modélisation volumique

# 18 Modélisation volumique du vérin électrique

## 18.1 Présentation

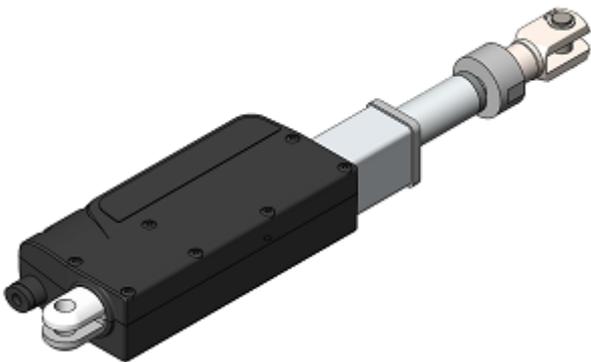
### Normes dimensionnelles

Le constructeur du vérin ne donne pas de renseignements sur les caractéristiques des roues dentées, de la vis sans fin et de la vis de manœuvre. Les normes utilisées pour ces éléments ne sont manifestement pas les normes européennes. De plus nous n'avons pas la désignation des matériaux utilisés. Les seules informations sont celles données sur le site du constructeur. En conséquence la maquette numérique réalisée, avec SolidWorks 2019-2020 (SW), est seulement proche du réel. Les dentures des roues 2, 3 et 4 ainsi que la vis sans fin 1 en représentation *réaliste* ne sont pas, géométriquement, correctes.

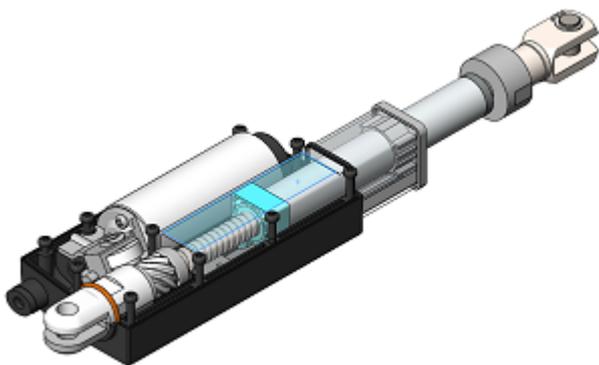
### Format neutre du SW

Afin d'éviter la complexité des arbres de construction la maquette a été convertie dans un format neutre.

La figure ci-dessous montre la maquette numérique du vérin complet.



L'intérieur du vérin est présenté sur la figure ci-dessous, 1/2 carter enlevé et tube de section rectangulaire transparent.



## **i** Info

Le vérin destiné à la pelleuse a été modifié de part la nécessité de placer à l'extrémité du coulisseau un capteur d'effort entre un raccord d'adaptation et une chape mobile.

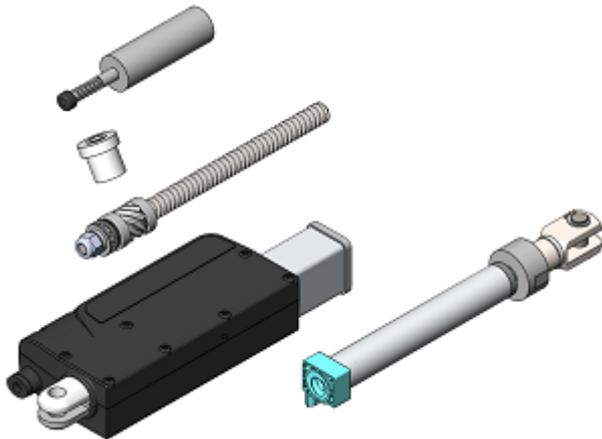
## 18.2 Arbre de construction dans SolidWorks

La représentation de l'ensemble des pièces du vérin est structurée en une association de classes d'équivalence (pièces liées complètement ou pièces cinématiquement équivalentes), notées par la suite SE.

Ainsi l'arbre de construction de premier niveau du vérin est constitué de cinq SE:

- SE Carter ;
- SE Coulisseau ;
- SE Vis de manoeuvre ;
- SE Arbre moteur ;
- SE Roues dentées.

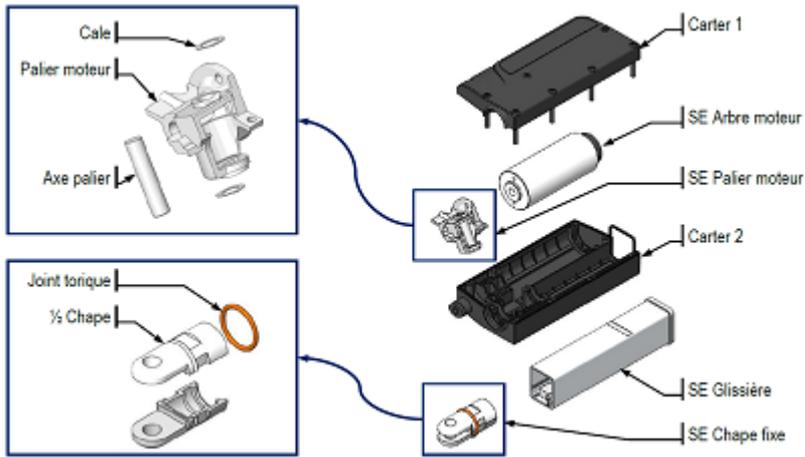
L'éclaté de la maquette numérique (figure ci-dessous) montre les cinq SE de premier niveau.



## 18.3 Détails des sous-ensembles

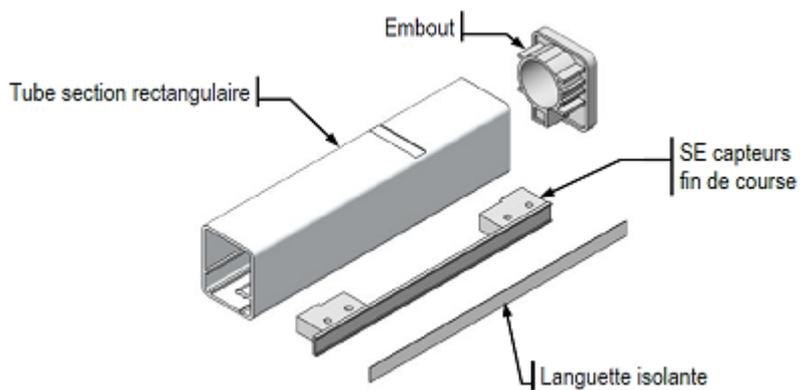
### 18.3.1 SE Carter

La figure ci-dessous présente les pièces constituant le SE Carter.



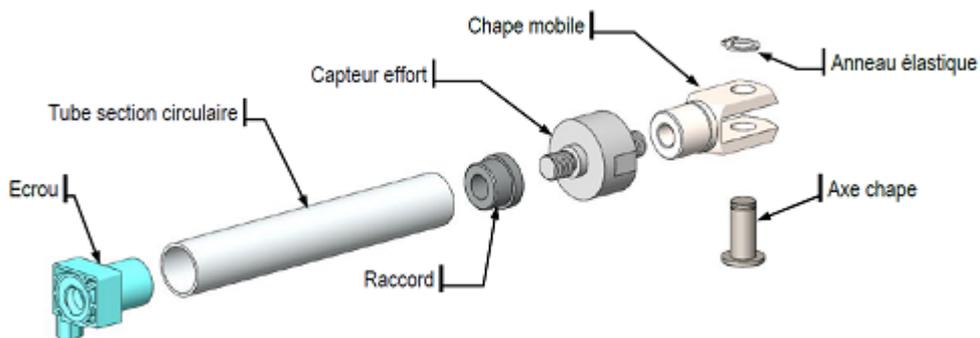
### 18.3.2 SE Glissière

La figure ci-dessous présente les pièces constituant le SE Glissière.



### 18.3.3 SE Coulisseau

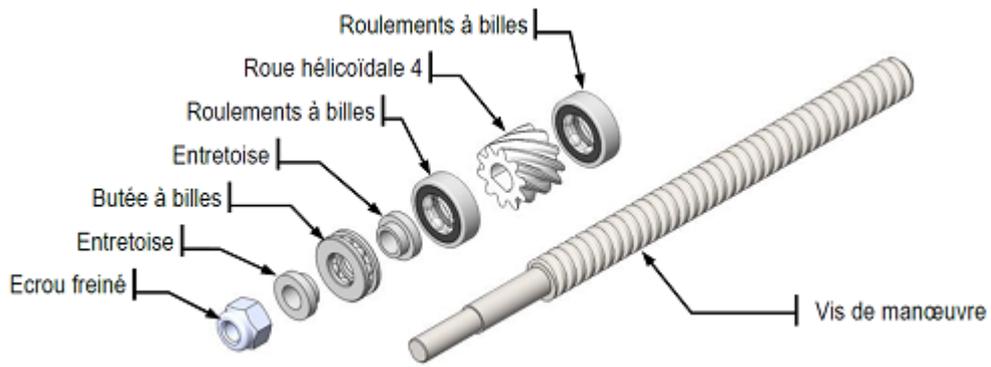
La figure ci-dessous présente les pièces constituant le SE Coulisseau.



L'écrou et le raccord sont vissés sur le tube de section circulaire et le capteur d'effort sur le raccord et la chape mobile.

### 18.3.4 SE Vis de manoeuvre

La figure ci-dessous présente les pièces constituant le SE Vis de manoeuvre.

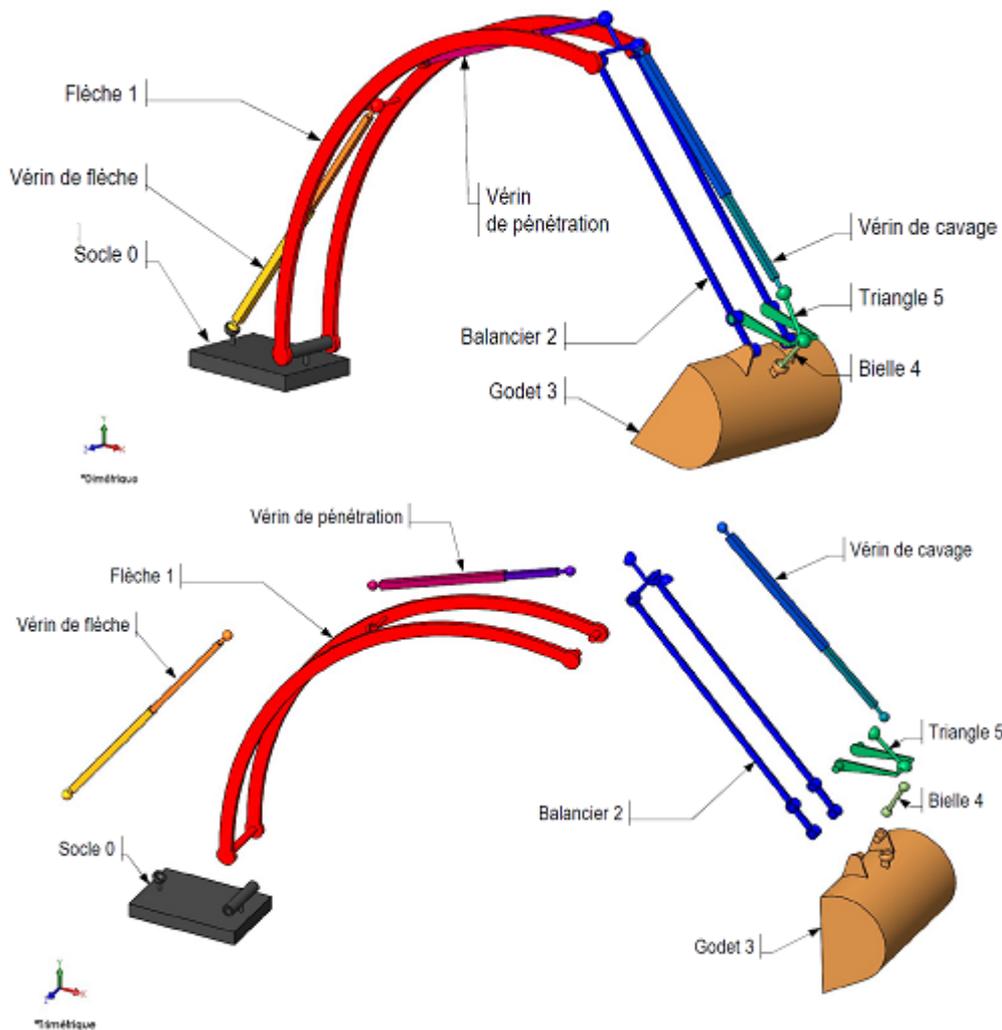


En toute rigueur, compte tenu du choix de la construction de la maquette par SE cinématiquement équivalents, les bagues extérieures des roulements et les rondelles de la butée à billes devraient être liées à la chape fixe liée au carter.

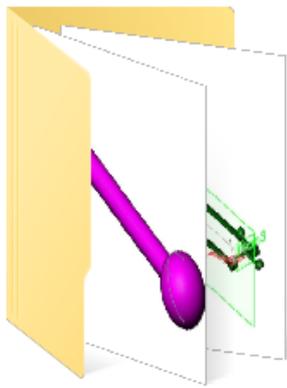
# 19 Modélisation spatiale minimale

## 19.1 Présentation

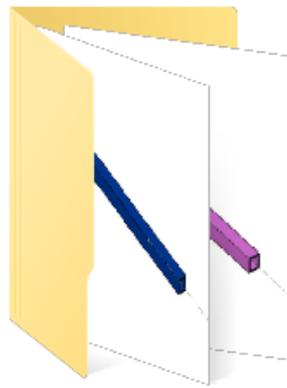
La pelleuse électrique schématisée, dont l'image de la maquette numérique est donnée ci-dessous (version assemblée et éclatée), est constituée essentiellement de **cinq pièces principales** et de **trois vérins**.



L'organisation des dossiers dans Windows est donnée ci-dessous.



Pièces

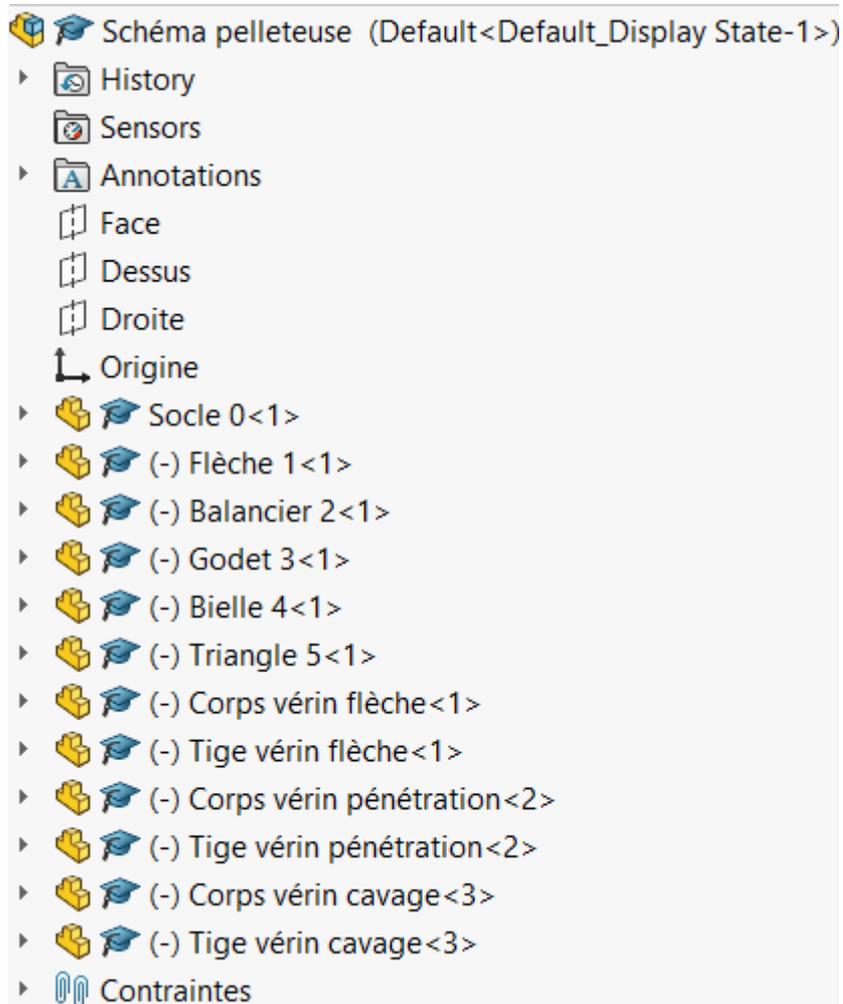


Vérins



Schéma pelleuse

Larbre de construction du schéma de la maquette dans SolidWorks (2019-2020) est donné ci-dessous.



## 19.2 Analyse mécanique

 **Note**

Dans cette représentation schématique, les vérins électriques sont représentés simplement par une tige (coulisseau) en liaison glissière avec le corps (glissière).

La transmission du mouvement du mouvement depuis le moteur électrique jusqu'au coulisseau n'est pas représentée.

D'après le schéma cinématique fig. 1 et le graphe de structure fig. 4, le **nombre de pièces** du bras de la pelleuse est  $N_P = 12$  et le **nombre de liaisons** est  $N_L = 15$  donc le **nombre cyclomatique**  $\nu$  définissant le nombre de cycles indépendants est

$$\nu = N_L - N_P + 1 = 15 - 12 + 4.$$

Il est donc possible d'écrire  $E_c = 6\nu = 6 \times 4 = 24$  **équations cinématiques de rang**  $r_c$  avec  $I_c = 31$  **inconnues cinématiques**.

Le nombre de degrés de **mobilités utiles**, correspond au mouvement de la flèche 1, du balancier 3 et du godet 3, est  $m_{cu} = 3$ .

Intuitivement on dénombre  $m_{ci} = 4$  **mobilités internes** (rotation des trois vérins et de la bielle 4 autour de leurs axes).

Donc au total le bras de la pelleuse possède  $m_c = m_{cu} + m_{ci} = 7$  **degrés de mobilité**.

Comme  $r_c = I_c - m_c = 31 - 7 = 24$  alors le **degré de surabondance** est

$$h = E_c - r_c = 24 - 24 = 0.$$

En conclusion le bras de la pelleuse comporte 3 degrés de mobilité utile et ainsi schématisé, n'est donc pas surabondante géométriquement (cinématiquement, ou isostatique).

En conséquence la maquette schématique de la pelleuse peut être utilisée directement dans Méca3D pour des études géométrique et/ou cinématique.

### III. Intelligence Artificielle

# 20 Apprentissage par renforcement

L'objectif de cet apprentissage est de permettre au bras de pelleteuse de déterminer de manière autonome la trajectoire à imposer au godet pour aller d'un point à un autre en évitant toute collision avec l'environnement.

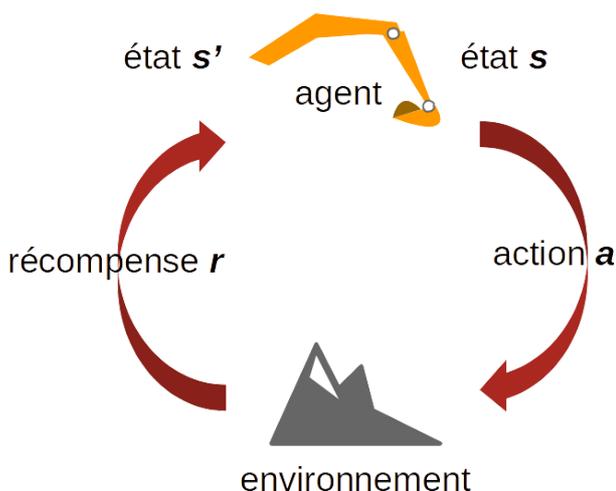
## 🔥 Tableau de bord associé

Consultez également le [tableau de bord associé](#).

## 20.1 Qlearning

L'apprentissage utilise la méthode Q-learning : une matrice  $Q[s, a]$  permet de déterminer l'action  $a$  à prendre par l'agent dans l'état  $s$  lui permettant d'obtenir la meilleure récompense  $r$  lorsqu'il sera dans l'état  $s'$  :

$$Q[s, a] := (1 - \alpha)Q[s, a] + \alpha \left( r + \max_{a'} Q[s', a'] \right).$$



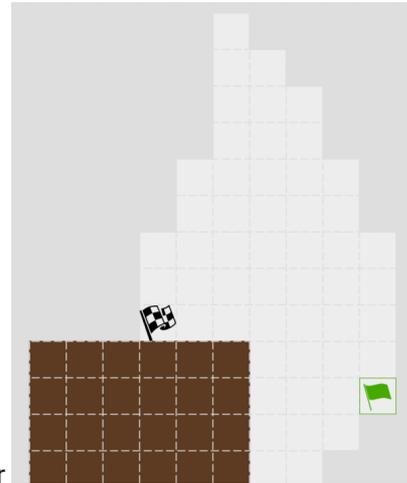
Les variables ajustables de ce modèle sont :

- le **facteur d'apprentissage**  $\alpha \in [0; 1]$  : permet de déterminer à quel point la nouvelle information calculée surpassera l'ancienne (valeur par défaut  $\alpha = 0,6$ )
  - $\alpha = 0$  : l'agent ignore le futur;
  - $\alpha = 1$  : l'agent ignore le passé.
- le **facteur de dépréciation**  $\gamma \in [0; 1]$  : détermine l'importance des actions futures  $a'$  (valeur par défaut  $\gamma = 0,9$ .)
  - $\gamma = 0$  : seule la récompense  $r$  est prise en compte;
  - $\gamma = 1$  : la récompense  $r$  est valorisée par les meilleures actions futures).

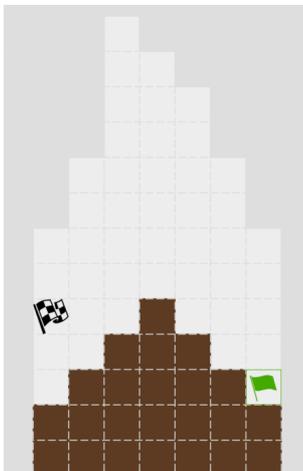
### 20.1.1 Environnement

L'environnement correspond à la zone de déplacement du godet tel que celui-ci puisse rester horizontal. Cette zone est discrétisée en carrés gris clairs de côté 60 mm.

Dans cet environnement sont disposés :



- un obstacle (rectangulaire ou triangulaire) que le godet devra éviter



- un point de départ  qu'il est possible de déplacer
- un point d'arrivée  qu'il est possible de déplacer mais sous la contrainte de toujours être à gauche du point de départ, ou au même niveau.

### 20.1.2 Actions

Les actions correspondent aux déplacements possibles du godet. On retient ici, sous la contrainte imposée du placement des points de départ et d'arrivée :

- Haut
- Gauche-Haut
- Gauche
- Gauche-Bas
- Bas

## 20.1.3 Récompenses

### 20.1.3.1 Au changement d'état

La récompense  $r$  qui intervient dans le calcul de la matrice  $Q$  prend en compte la distance qui sépare l'état  $s$  de l'agent au point d'arrivée :

$$r = [120 - d(s - s_f)]10^{-3}.$$

La distance 120 correspond à 3 cases de l'environnement, cette variable peut être ajustée dans la boîte **Récompenses** du **tableau de bord associé** dans MyViz. Il en va de même pour le facteur  $10^{-3}$ .

### 20.1.3.2 Au succès

Lorsque l'agent arrive au point d'arrivée, la récompense est fixée à  $r = 10$  (bonus). Cette valeur peut-être modifiée.

### 20.1.3.3 À l'échec

Lorsque l'agent arrive rencontre un obstacle ou sort du domaine, la récompense est fixée à  $r = -10$  (malus). Cette valeur peut-être modifiée.

## 20.2 Epsilon-greedy

Le choix de l'action à prendre, pour passer d'un état  $s$  à un état  $s'$ , est générée grâce à la méthode Epsilon-greedy. Un facteur aléatoire  $\epsilon$  est calculé en fonction du nombre d'itérations total de l'apprentissage  $N$  (par défaut  $N = 2000$ ) et par le numéro  $i$  de l'itération en cours :

$$\epsilon = 1 - i/N$$

Une variable aléatoire  $var$  entre 0 et 1 est calculée à pour chaque changement d'état :

- si  $var \leq \epsilon$  alors le choix de l'action  $a$  est aléatoire;
- si  $var > \epsilon$  alors le choix de l'action  $a$  est déterminé par  $\max_a Q[s, a]$ .

Dans MyViz, pour représenter le choix de l'action, un pictogramme apparaît



- dans le cas d'un choix aléatoire

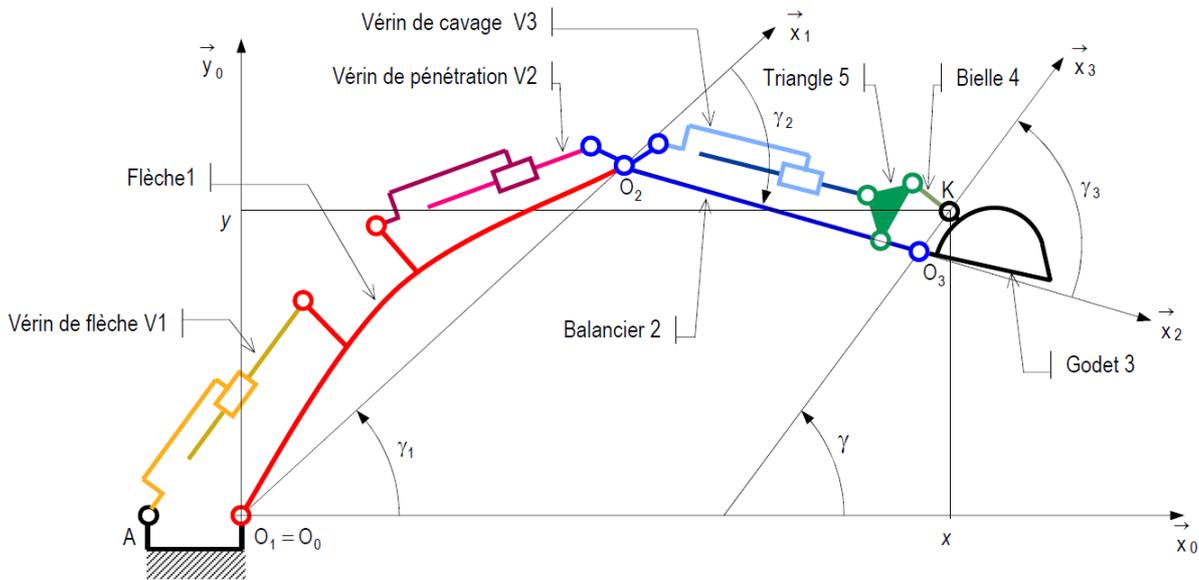


- dans le cas d'un choix par la matrice  $Q$

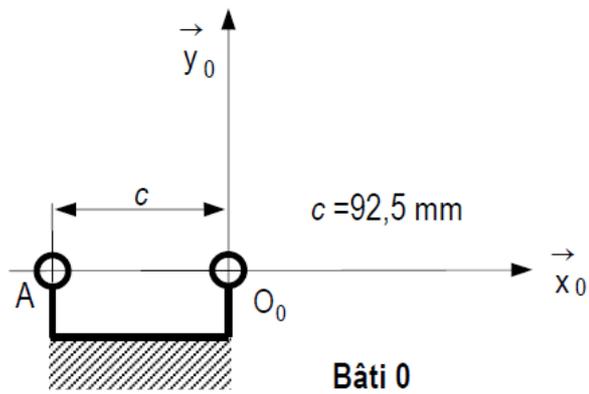
## IV. Documentation technique

## IV.1 Caractéristiques des pièces

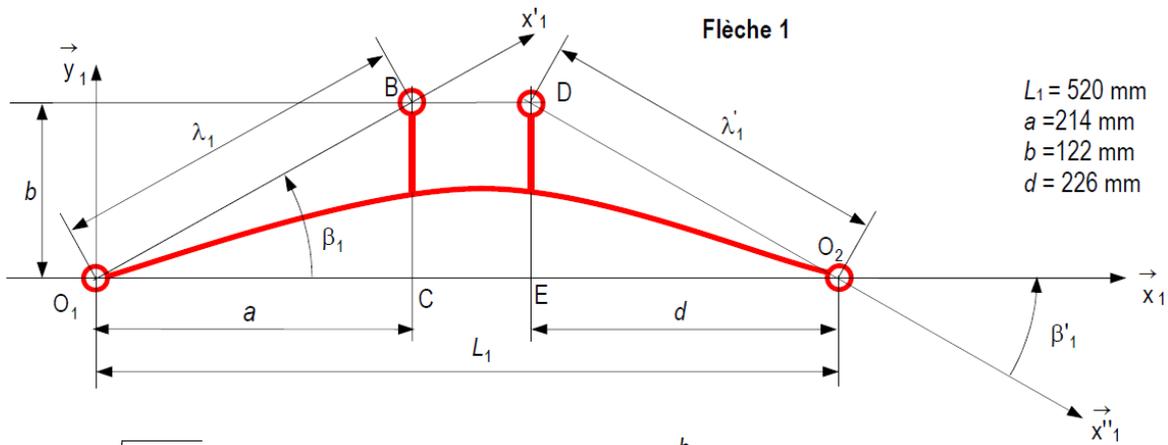
# 21 Caractéristiques géométriques



## 21.1 Bâti



## 21.2 Flèche

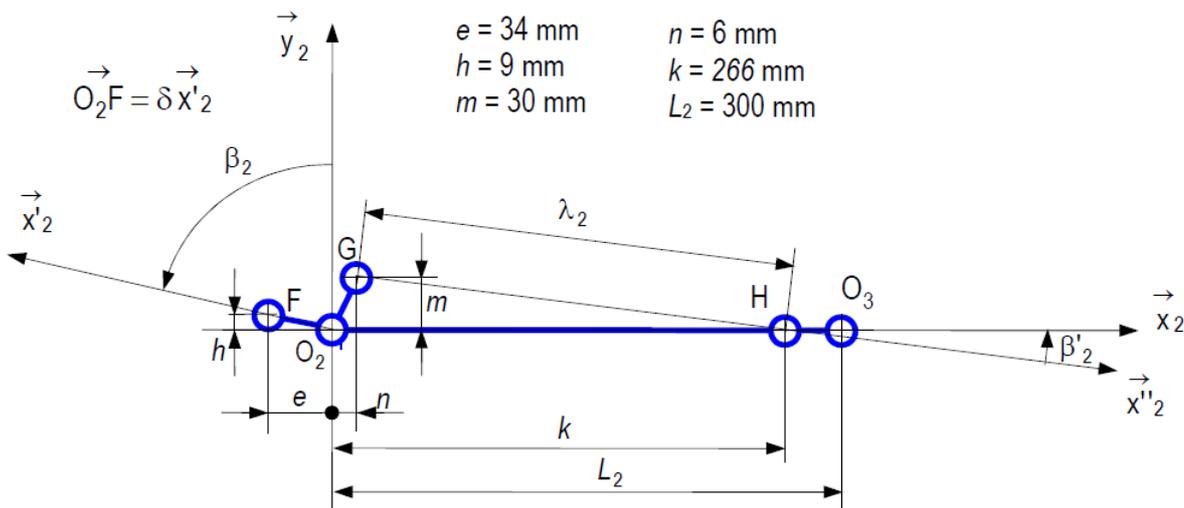


$L_1 = 520 \text{ mm}$   
 $a = 214 \text{ mm}$   
 $b = 122 \text{ mm}$   
 $d = 226 \text{ mm}$

$$\lambda_1 = \sqrt{a^2 + b^2} \quad \lambda_1 \cong 246,33 \text{ mm} \quad \beta_1 = \arctan \frac{b}{a} \quad \beta_1 \cong 29,69^\circ$$

$$\lambda'_1 = \sqrt{b^2 + d^2} \quad \lambda'_1 \cong 256,83 \text{ mm} \quad \beta'_1 = \arctan \frac{b}{d} \quad \beta'_1 \cong 28,36^\circ$$

### 21.3 Balancier

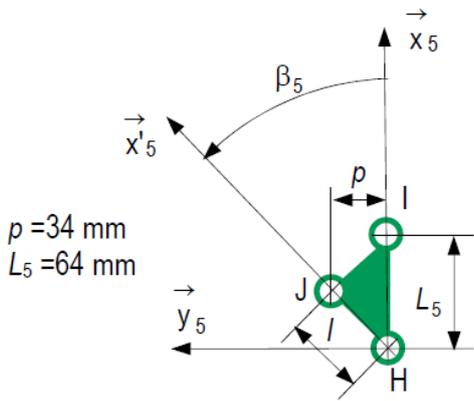


$e = 34 \text{ mm}$        $n = 6 \text{ mm}$   
 $h = 9 \text{ mm}$        $k = 266 \text{ mm}$   
 $m = 30 \text{ mm}$        $L_2 = 300 \text{ mm}$

$$\delta = \sqrt{e^2 + h^2} \quad \delta \cong 35,17 \text{ mm} \quad \beta_2 = \arctan \frac{e}{h} \quad \beta_2 \cong 75,17^\circ$$

$$\lambda_2 = \sqrt{m^2 + (k - n)^2} \quad \lambda_2 \cong 261,73 \text{ mm} \quad \beta'_2 = \arctan \frac{m}{k - n} \quad \beta'_2 \cong 6,58^\circ$$

### 21.4 Triangle de renvoi



$p = 34 \text{ mm}$   
 $L_5 = 64 \text{ mm}$

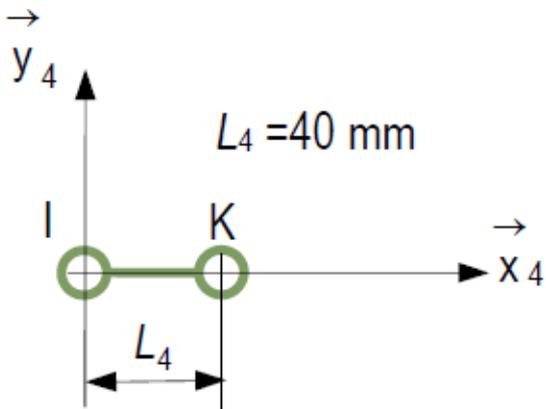
$$\beta_5 = \arctan \frac{2p}{L_5} \quad \beta_5 \cong 46,74^\circ$$

$$l = \sqrt{p^2 + \left(\frac{L_5}{2}\right)^2} \quad l \cong 46,69 \text{ mm}$$

Le triangle IJH est isocèle :  $IJ = JH$

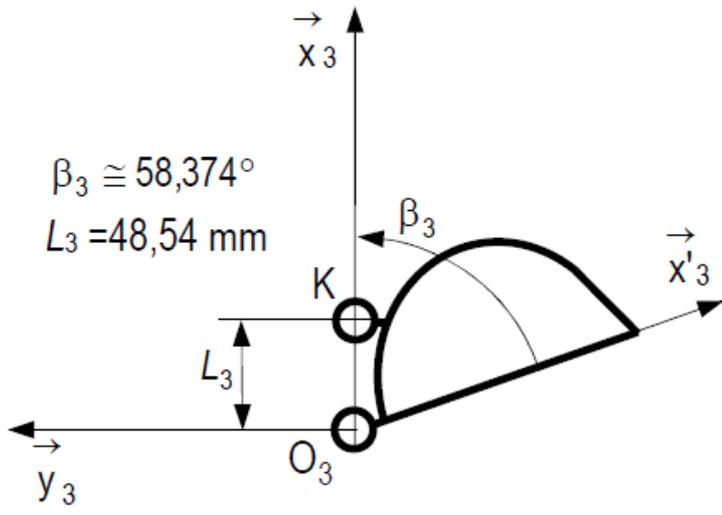
**Triangle 5**

### 21.5 Bielle



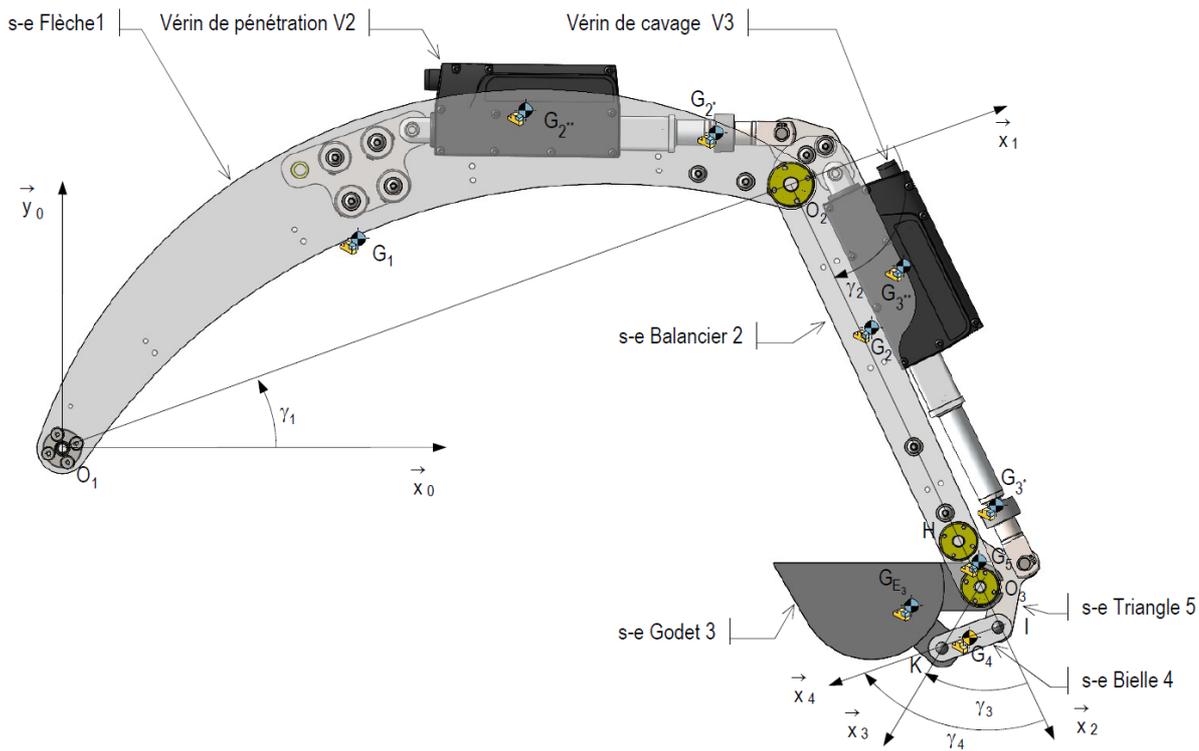
**Bielle 4**

### 21.6 Godet

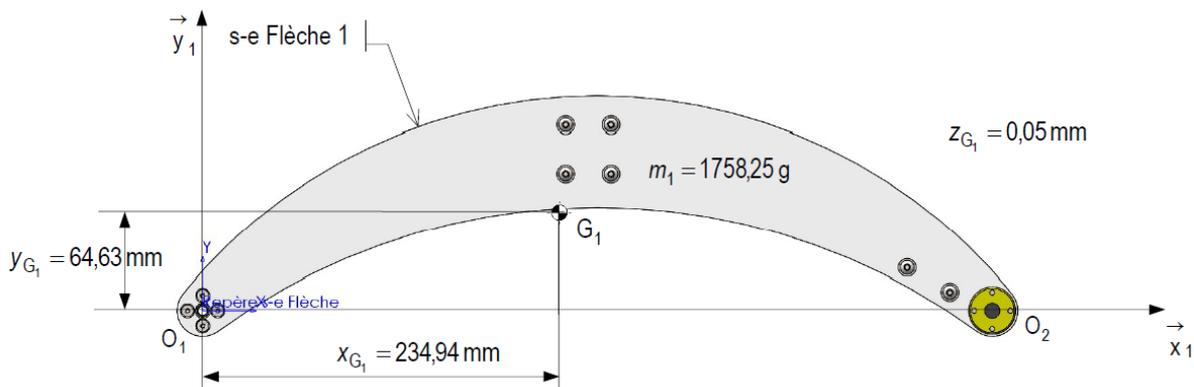


**Godet 3**

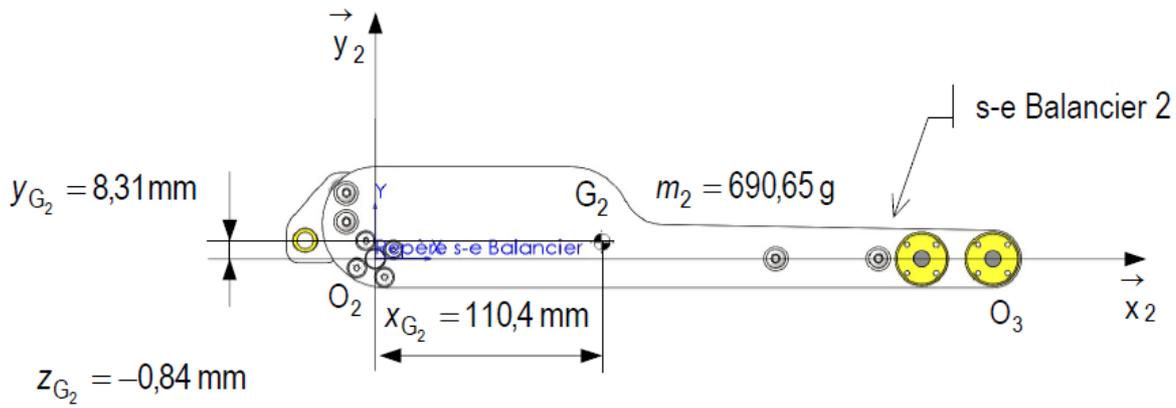
## 22 Centre de masse



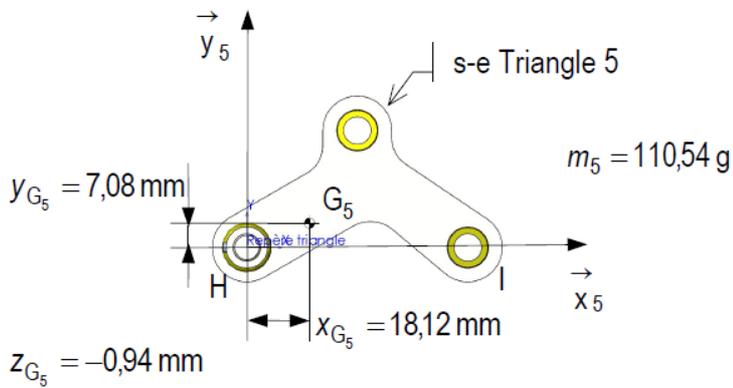
### 22.1 Flèche



### 22.2 Balancier



## 22.3 Triangle

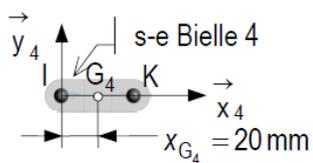


## 22.4 Bielle

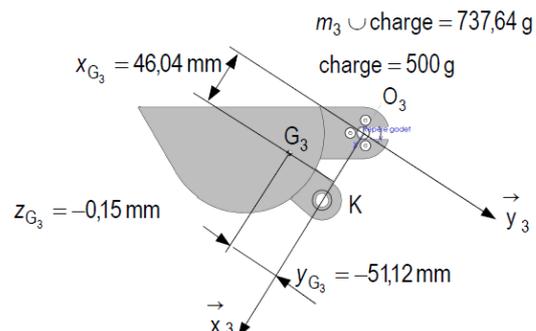
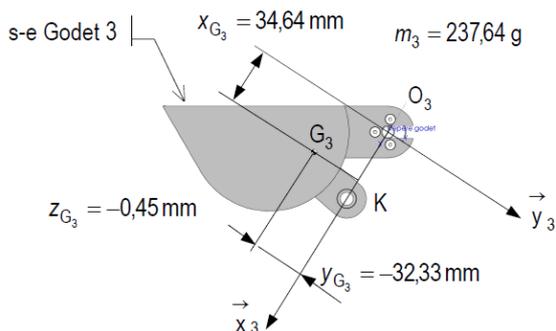
$m_4 = 44,22 \text{ g}$

$y_{G_4} = 0 \text{ mm}$

$z_{G_4} = -0,34 \text{ mm}$

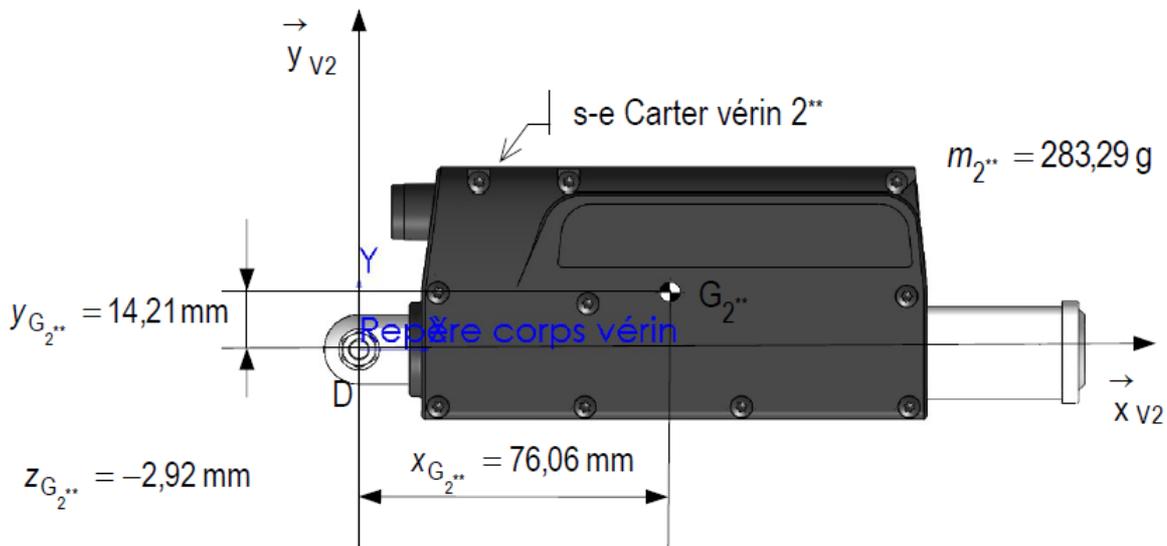


## 22.5 Godet

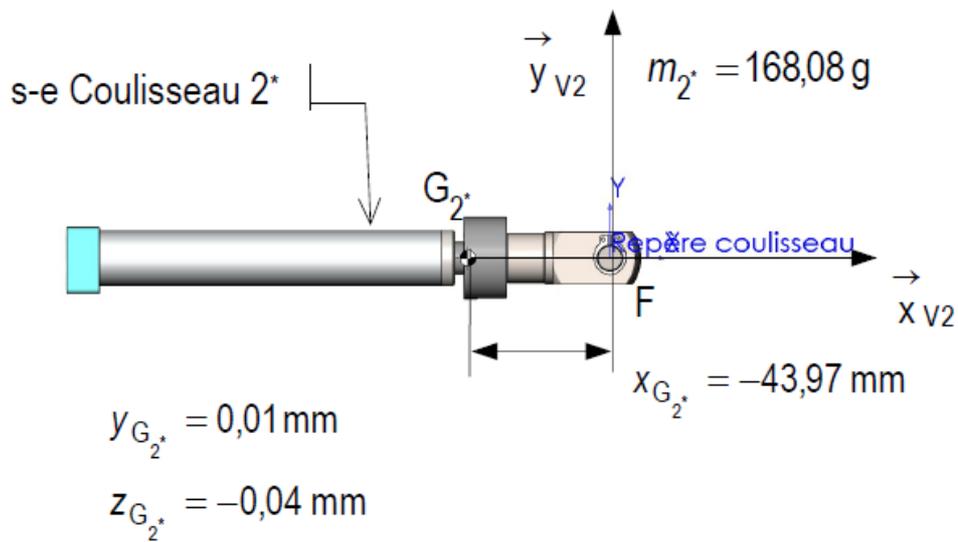


## 22.6 Vérin

### 22.6.1 Corps



### 22.6.2 Tige



## IV.II Chaîne d'information

# 23 Mesures

## 23.1 Codeurs d'axes

### 23.1.1 Présentation

Le codeur d'axe est un codeur absolu numérique Avago AEAT-6012-A06 avec interface SSI.



Le bras de pelleuse en compte 4, permettant la mesure des angles relatifs

- flèche \ bâti ( $\gamma_1$ );
- balancier \ flèche ( $\gamma_2$ );
- triangle \ balancier ( $\gamma_3$ );
- godet \ triangle ( $\gamma_4$ ).



[Document constructeur](#)

### 23.1.2 Position angulaire des axes

Sa mesure est codée sur 12 bits, ce qui signifie que la résolution angulaire brute est de 0,088 °, selon la formule

$$\frac{360}{2^{12}} = \frac{360}{4096} \approx 0,088 \text{ °}$$

Cependant, un algorithme particulier a été mis en place pour éliminer les mesures aberrantes observées dans certains cas de figure :

- une mesure est faite toutes les 2 ms;
- toutes les 10 ms, on fait la moyenne sur les 5 dernières mesures;
- les trois mesures les plus éloignées de la moyenne sont éliminées;
- la position angulaire fournie par le système est la moyenne des 2 mesures restantes. La résolution angulaire est alors de 0,044°;
- dans le cas particulier des modes « Statique » et « Energétique / Puissance », la position est en plus filtrée par un premier ordre de constante de temps 0,05 s.

### 23.1.3 Vitesse angulaire des axes

Ce capteur est également utilisé pour calculer, toutes les 10 ms, la vitesse angulaire des axes (en divisant par 0,01 l'écart entre la position courante et la position précédente) et en faisant une moyenne glissante sur 10 échantillons.

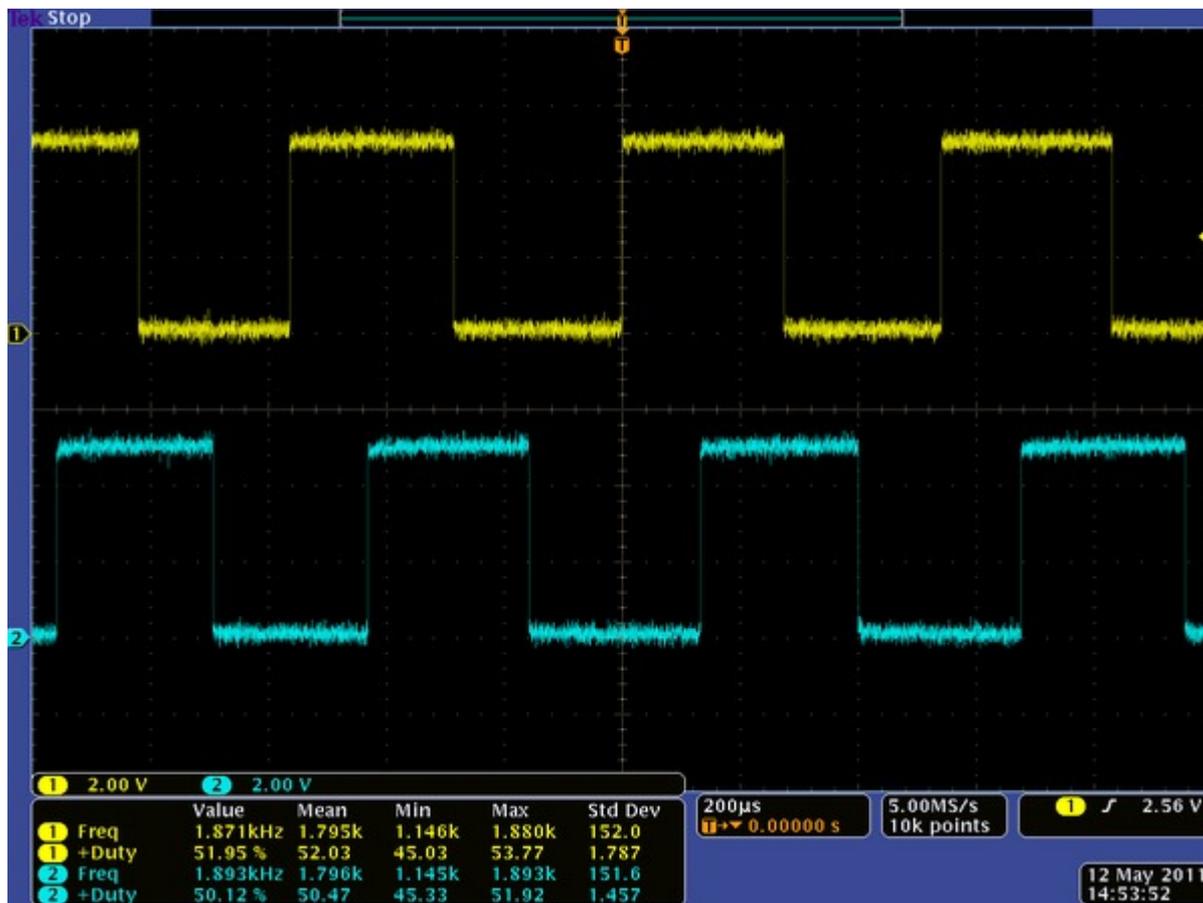
Avant la moyenne glissante, la résolution sur la valeur angulaire est égale à  $\frac{0,044}{0,01} = 4,4 \text{ °/s}$ .

La moyenne glissante permet d'obtenir une résolution 10 fois inférieure, de 0,44 deg/s.

## 23.2 Codeurs de vérins

### 23.2.1 Présentation

Le moteur du vérin est doté d'un codeur incrémental afin de mesurer sa position et de calculer sa vitesse de rotation. Ce codeur incrémental fournit deux signaux carrés en quadrature, comme sur la capture ci-dessous.



Deux signaux en quadrature sont indispensables pour connaître le sens de rotation. En effet, lorsqu'un front (montant ou descendant) est compté sur une voie, le niveau (haut ou bas) du signal sur l'autre voie permet de savoir si le moteur tourne dans le sens positif ou dans le sens négatif.

Le codeur du moteur du vérin intègre un seul aimant. Il génère donc 1 impulsion par tour (PPR) ce qui conduit à 4 fronts par tour (CPR) car on est capable de compter les fronts montants et descendants produits par cet aimant sur les deux capteurs à effet Hall décalés de 90°, attachés à la partie fixe du moteur.

### 23.2.2 Position vérin

Dans la mesure où la fonction essentielle du moteur est de déplacer la tige du vérin, nous allons nous intéresser ici à la résolution linéaire de la mesure de position et du calcul de vitesse.

La résolution sur la position se calcule sachant que le moteur fait 7,16 tours pour un déplacement de la tige de 1 mm.

Le CPR étant égal à 4, la résolution est donc égale à  $\frac{1}{4 \times 7,16} \approx 0,035$  mm.

### 23.2.3 Vitesse vérin

La vitesse est calculée toutes les 10 ms en divisant par 0,01 l'écart entre la position courante et la position précédente et en faisant une moyenne glissante sur 10 échantillons. Avant la moyenne glissante, la résolution est donc

$$\frac{0,035}{0,01} = 3,5 \text{ mm/s.}$$

La moyenne glissante permet d'obtenir une résolution 10 fois inférieure, de 0.35 mm/s.

## 23.3 Capteur d'effort

Les capteurs d'effort, monté entre les tiges de vérins et respectivement la flèche, le balancier et la bielle du godet, sont de type piezoélectrique.

### Etalonnage

À la première utilisation du banc de TP ou après un déplacement du banc de TP, il est conseillé de réaliser un étalonnage du capteur via le [Tableau de Bord MaintenanceCalibration](#)).

## 24 Commande des vérins

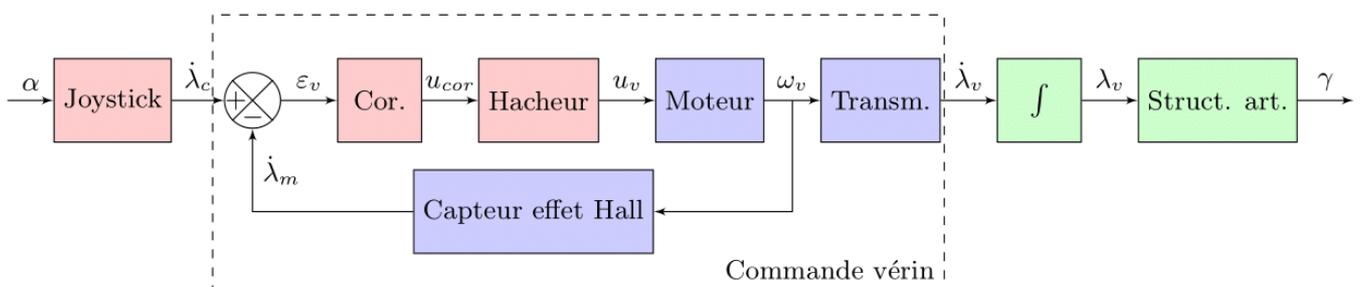
Chacun des 3 axes du bras de la pelleuse intègre deux boucles d'asservissement imbriquées :

- Asservissement de type PI sur la vitesse des vérins
- Asservissement de type PID sur la position angulaire des axes

### 24.1 Asservissement en vitesse

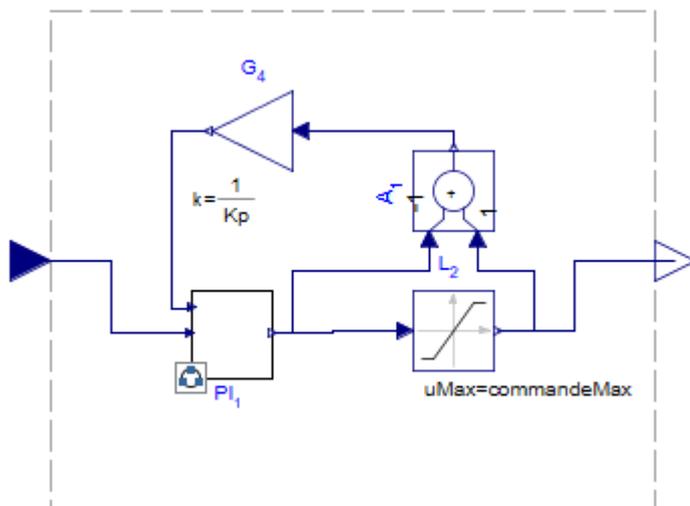
#### Commande en Vitesse

Lorsque la pelleuse est utilisée à partir du tableau de bord "Prise en main", les commandes sont passées via le joystick : l'inclinaison  $\alpha$  du joystick génère une commande proportionnelle à la vitesse désirée (l'inclinaison maximale correspond à la vitesse maximale de la tige)

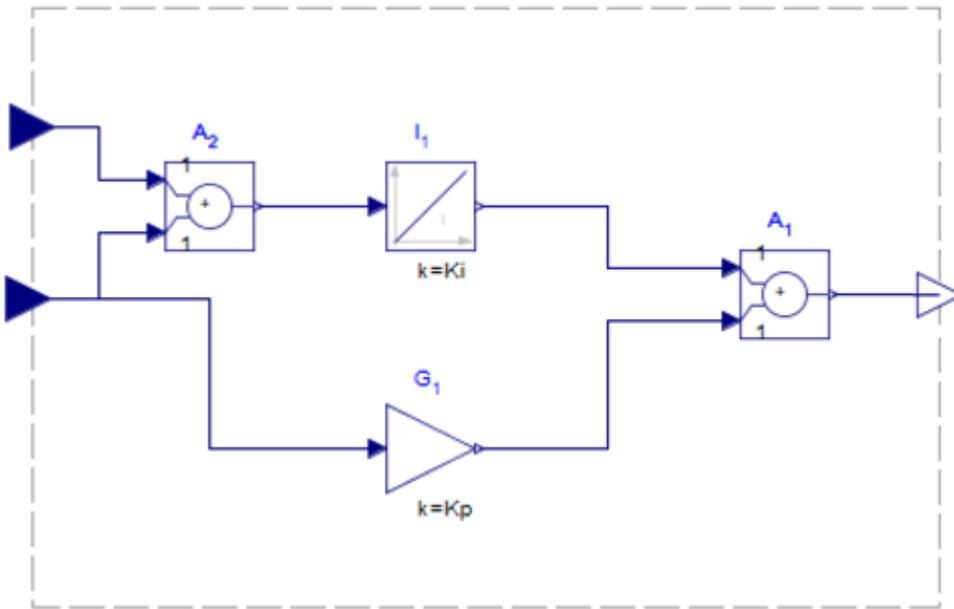


L'asservissement en vitesse des vérins est réalisé par un régulateur de type proportionnel-intégral avec saturation de tension de commande en sortie et dispositif d'anti-saturation de l'intégrateur (anti-emballement ou anti-windup).

Le régulateur a la forme générale ci-dessous.



Ci-dessous le dispositif d'anti-windup et le correcteur proportionnel-intégral dans le sous-système PI, implémenté.



La fonction de transfert du régulateur PI si la sortie n'est pas saturée est la suivante :

$$C(p) = K_p + \frac{K_i}{p}$$

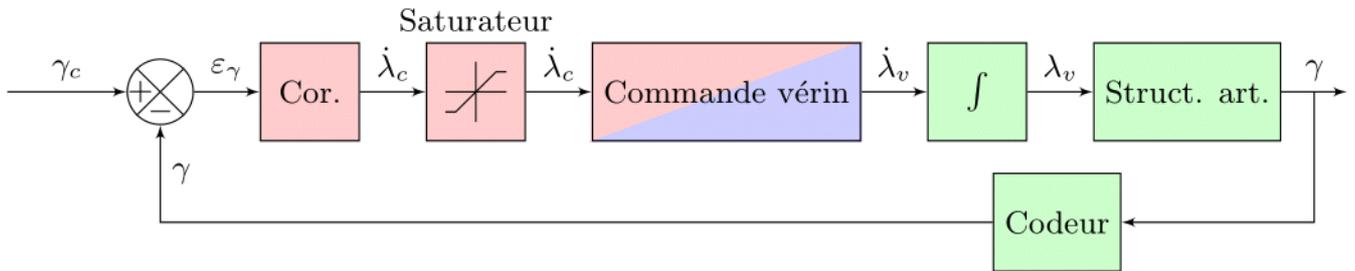
Si la sortie est saturée, la relation entre l'entrée ( $u$ ), la sortie non saturée ( $y$ ) et la sortie saturée ( $y_{sat}$ ) est la suivante :

$$y_{sat} = \left( K_p + \frac{K_i}{p} \right) u + \frac{K_i (y_{sat} - y)}{p K_p}$$

## 24.2 Asservissement en position

**i Info**

Sauf dans le cas d'une utilisation par joystick, c'est l'asservissement en position angulaire de l'axe qui est retenu, avec la boucle de vitesse imbriquée dans la commande du vérin (voir paragraphe précédent).

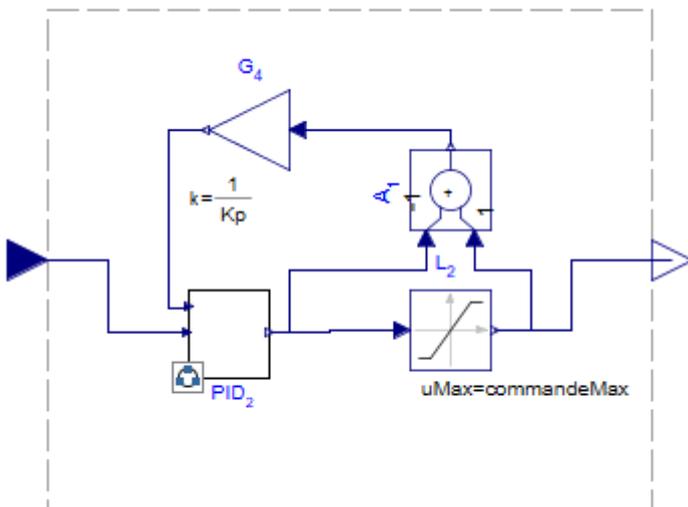


Le saturateur permet d'imposer une vitesse maximale de déplacement de la tige du vérin. Elle correspond à la valeur adaptée réglable dans MyViz.

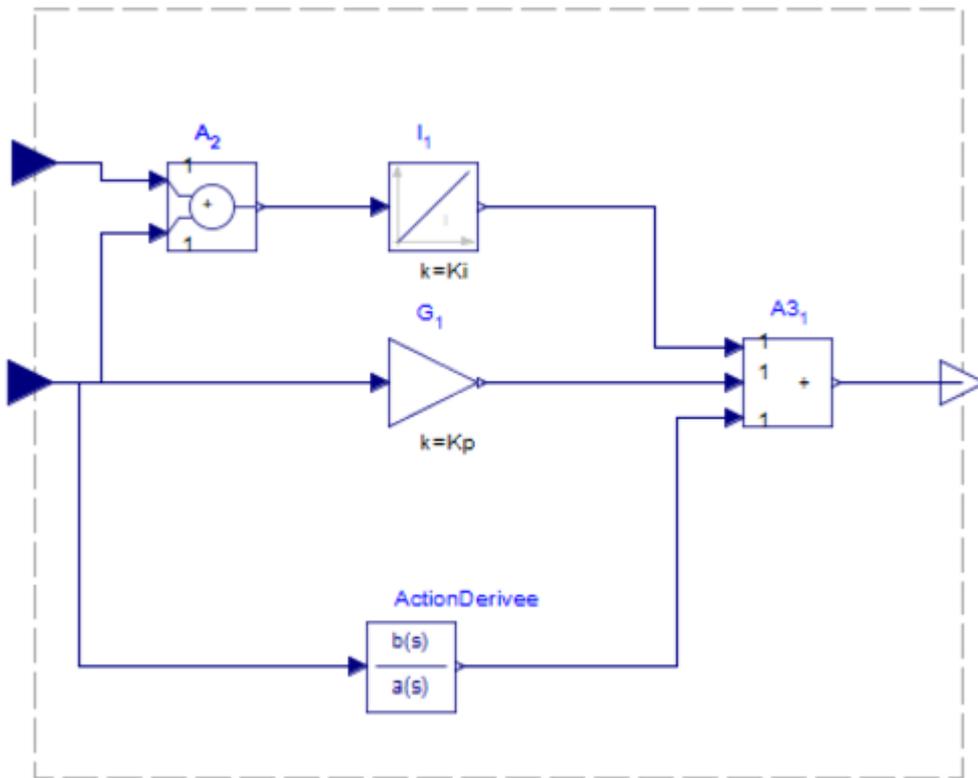
Le codeur avec interface SSI permet d'obtenir en sortie directement la valeur de l'angle, ce qui implique l'absence d'adaptateur entre l'entrée  $\gamma_c$  et le comparateur. On peut donc considérer un retour unitaire.

L'asservissement de position des axes est réalisé par un régulateur de type proportionnel-intégral-dérivé avec saturation de vitesse de consigne en sortie et dispositif d'anti-saturation de l'intégrateur (anti-windup).

Le régulateur a la forme générale suivante :



Ci-dessous, le dispositif d'anti-windup et le correcteur proportionnel-intégral-dérivé dans le sous-système PID.



Par rapport au PI implanté dans la boucle de vitesse, on a ajouté la fonction de transfert « ActionDerivee », soit donc la fonction de transfert du correcteur, sans saturation,

$$C(p) = K_p + \frac{K_i}{p} + \frac{K_d p}{T_f p + 1}$$

D'un point de vue mathématique, le dénominateur permet d'éviter que la fonction de transfert du PID soit « non-propre » (ordre du numérateur supérieur à l'ordre du dénominateur). D'un point de vue plus concret, il permet aussi de filtrer la dérivée. La valeur de  $T_f$  est 0,02.

Si la sortie est saturée, la relation entre l'entrée ( $u$ ), la sortie non saturée ( $y$ ) et la sortie saturée ( $y_{sat}$ ) est :

$$y_{sat} = \left( K_p + \frac{K_i}{p} + \frac{K_d p}{T_f p + 1} \right) u + \frac{K_i (y_{sat} - y)}{p K_p}$$

## IV.III Chaîne d'énergie

# 25 Verin électrique

## 25.1 Référence

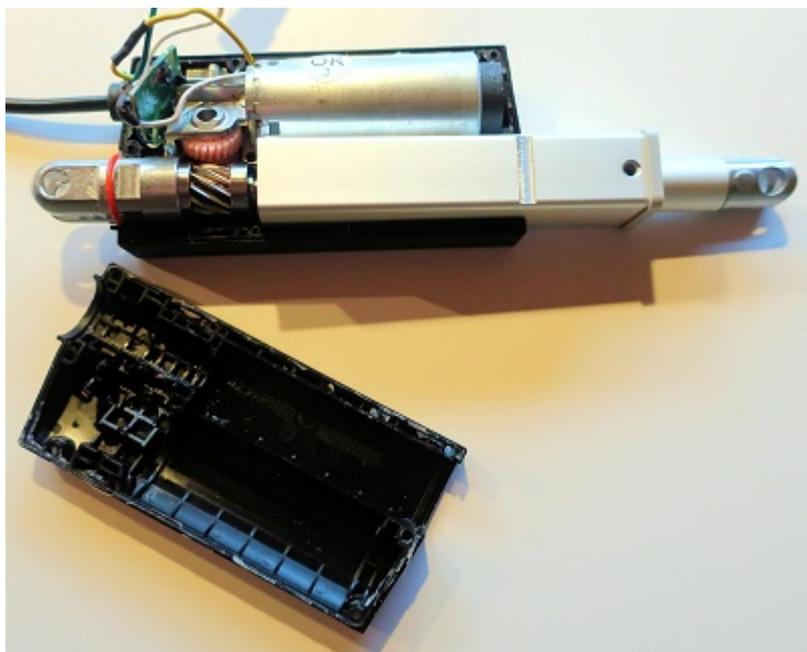
Le vérin électrique **TA38** est un produit conçu et fabriqué par la société **TIMOTION** (site).

[Document constructeur](#)

## 25.2 Constitution du vérin électrique

### 25.2.1 Vue d'ensemble

Le démontage de la partie supérieure du carter fait apparaître les pièces et les sous-ensembles du vérin (figure ci-dessous).

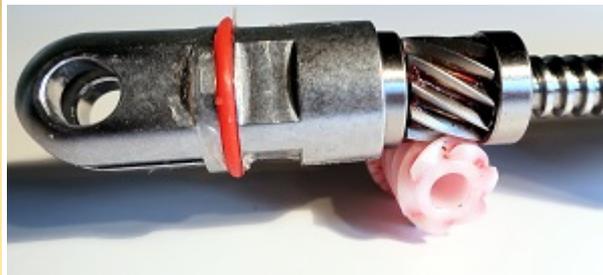
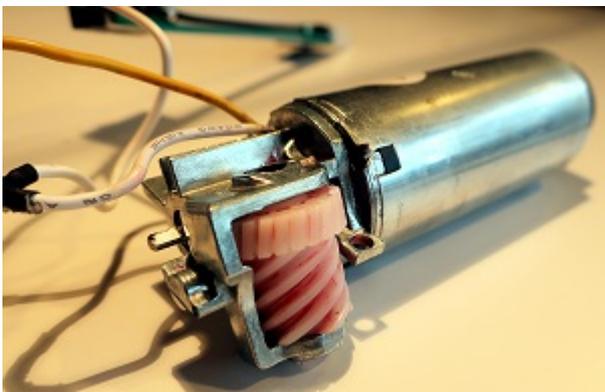
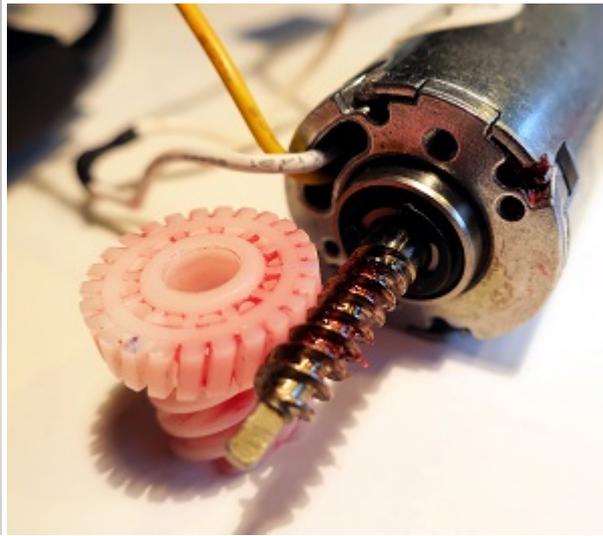
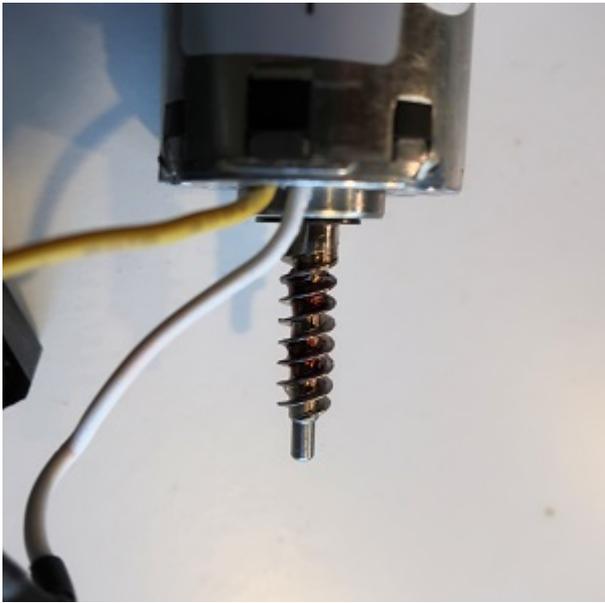


Dans le carter, en deux parties, se trouvent principalement :

- un **moteur électrique** ;
- une **carte électronique** ;
- un **système de transmission** du mouvement de rotation du moteur à un **système vis-écrou**.

### 25.2.2 Transmission du mouvement arbre/vis de manœuvre

Les figures ci-dessous présentent les détails de la transmission du mouvement depuis l'arbre moteur jusqu'à la vis de manœuvre. L'arbre moteur, **vis sans fin** à deux filets à droite, engrène avec une **roue hélicoïdale** de 25 dents. Cette dernière est liée complètement à une **roue hélicoïdale** de 6 dents qui engrène avec une **roue hélicoïdale** de 11 dents d'axe orthogonal, liée à la **vis de manœuvre** (pas de 3 mm à droite).



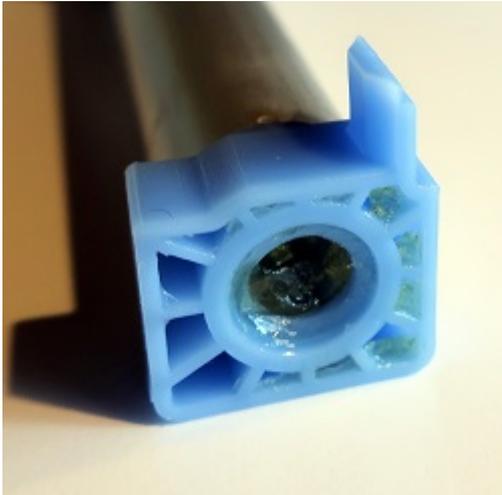
### 25.2.3 Système vis/écrou

Le système vis-écrou (figure ci-dessous) est principalement et classiquement constitué d'une **glissière** (tube carré), d'un **écrou** lié complètement à un tube cylindrique (s-e coulisseau) et d'une **vis de manœuvre** en liaison pivot par rapport au carter (figure ci-dessous).



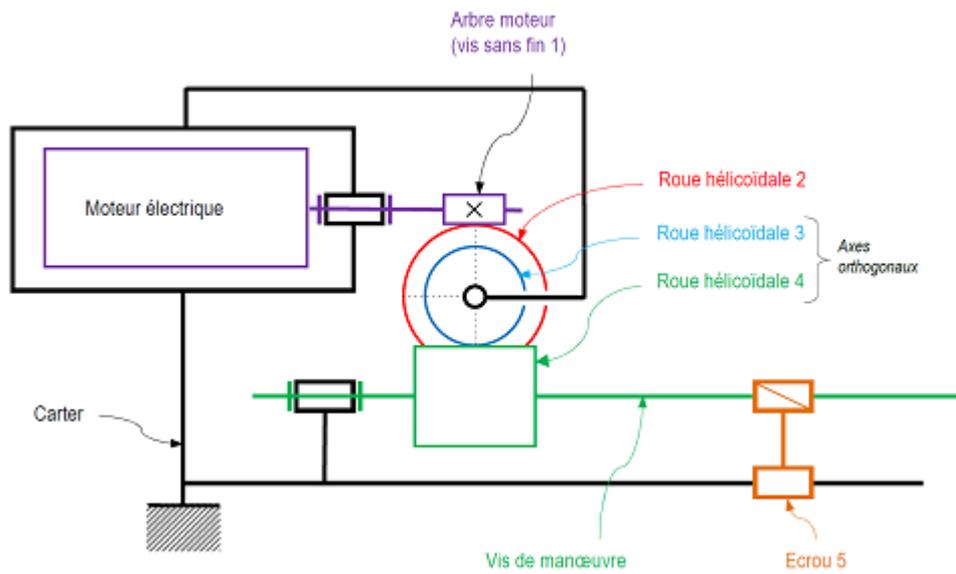
La liaison pivot (figure ci-dessous) est, plus précisément, réalisée entre les deux  $\frac{1}{2}$  chapes liés complètement au carter et la vis de manœuvre par deux roulements à billes et une butée à billes simple effet.

La glissière est munie de deux interrupteurs de fin de course actionnés par l'intermédiaire de l'ergot de l'écrou.



## 25.3 Schéma cinématique

L'examen des pièces démontées permet de tracer le schéma cinématique plan développé du vérin.



## 25.4 Maquette numérique

Se référer à la partie modèle volumique.

## 25.5 Rapport de transmission

## 25.6 Masses des pièces du vérin électrique

### ⚠ Matériaux inconnus

Les matériaux utilisés pour les différentes pièces du vérin électrique sont inconnus. La pesée des différentes pièces a permis d'obtenir la masse.

<u>SE Carter</u>	Masse (g)
Carter 1	47
Carter 2	31,9
Tube section rectangulaire	26,8
Embout	3,8
Chape fixe	16
Palier	32,7

<b><u>SE Carter</u></b>	<b>Masse (g)</b>
Axe palier	5,9
Vis torx l=25	1,1x7
Vis torx l=12	0,6x3
Vis (tube carré)	0,5
Vis (palier)	0,5x2

<b><u>SE Coulisseau</u></b>	<b>Masse (g)</b>
Tube section circulaire	21,5
Ecrou	5,2

<b><u>SE Vis de manœuvre</u></b>	<b>Masse (g)</b>
Vis de manœuvre	59,3
Billes+cage	1,3
Rondelles	1,3x2
Roulements	4,9x2
Entretoises	1,6x2

<b><u>SE Engrenages</u></b>	<b>Masse (g)</b>
Roue hélicoïdale	8,6
s-e Roues dentées	3,5
Ecrou freiné	3

<b><u>SE Moteur électrique</u></b>	<b>Masse (g)</b>
------------------------------------	------------------

<b>SE Moteur électrique</b>	<b>Masse (g)</b>
Moteur électrique	210 (mesuré)
Stator	50 (évaluation)
Arbre moteur	160 (évaluation)

<b>Total</b>	<b>503,1</b>
Capteur effort	100 (donnée constructeur)

## 26 Structure articulée

La transformation de mouvement, translation/rotation, de chaque axe est réalisée par la biais d'une structure articulée actionnée par le vérin (association en série de trois liaisons pivot et une liaison glissière). Pour le godet, un système 4 barres vient s'ajouter entre le godet et la structure articulée.

Voir la section [Modélisation géométrique plane](#) pour plus de détails.