



SolidWorks et Meca3D

1 | Barrière Sinusmatic

1.1 Présentation

1.1.1 Objectif

L'objectif de ce TP est de se familiariser avec l'utilisation de Solidworks et meca3D dans le but de faire de la simulation numérique.

Les attendus en fin de TP sont :

- Avoir une idée de la réalisation d'une pièce sous SolidWorks.
- Réaliser des contraintes (correctement) pour assembler un mécanisme.
- Sous Méca3D, créer les liaisons, animer le système et tracer la loi entrée/sortie.
- Récupérer les données sous python et tracer la loi entrée/sortie.

1.2 Support

Le système Sinusmatic est un système de lève-barrière adaptable en dimension et en vitesse à la plupart des applications. Sa particularité résulte de la cinématique brevetée de son renvoi d'angle. Le système est constitué de 5 ensembles cinématiques présentés sur la Figure 1.

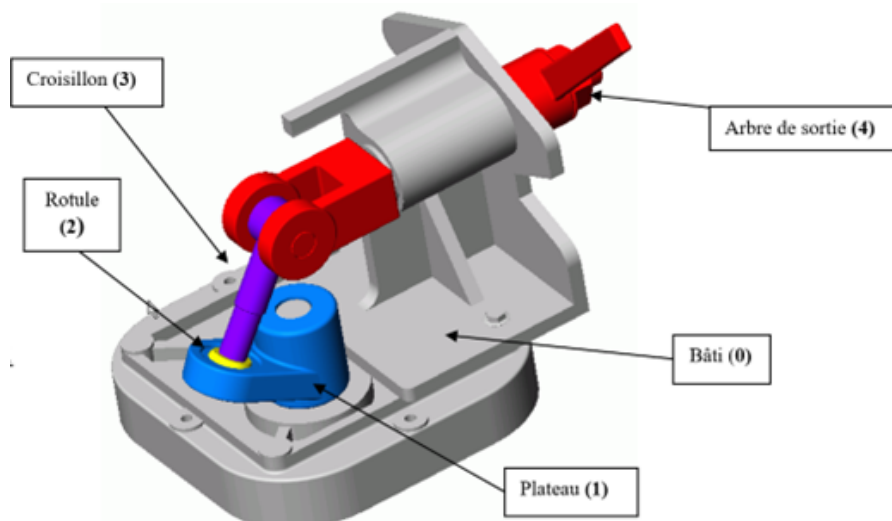


FIGURE 1 – Ensembles cinématiques du système Sinusmatic

1.3 Création d'une pièce

L'objectif de cette première partie est de dessiner/modéliser la pièce "Croisillon", le plan est donné dans le fichier **croisillon.pdf**.

🔗 Ouvrir ce pdf et s'assurer de bien comprendre la pièce à produire.

Cette pièce peut être dessinée en 5 étapes :

1. *Gros* cylindre de la base
2. *Grand* cylindre de l'arbre
3. *Petit* cylindre de l'arbre
4. Alésage (trou) dans le cylindre de la base
5. Finition (congé et chanfrein)

Il serait malin de penser à **sauvegarder** votre fichier **de temps en temps**. **Ne pas** le sauvegarder sous le nom croisillon (choisir monCroisillon par exemple).

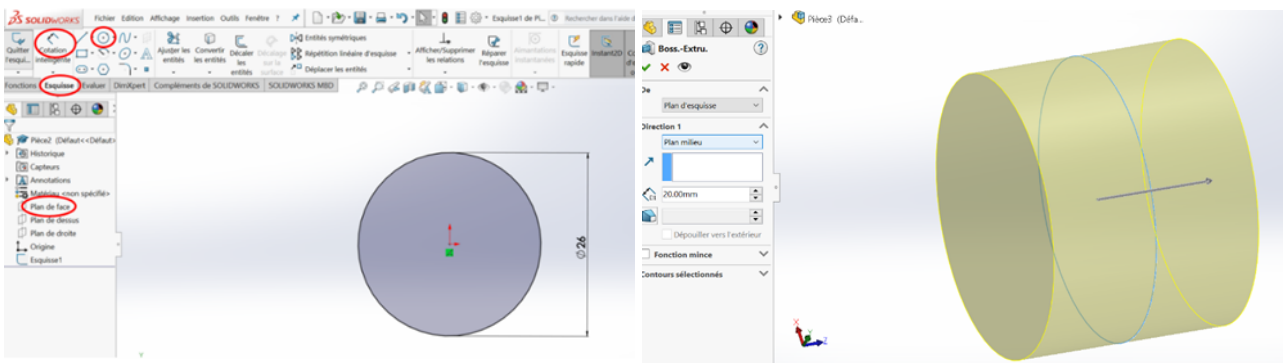
 Ouvrir Solidworks (pas Solidworks 2017)

 Créer une nouvelle pièce (**Ctrl+N**, ou )

Solidworks fonctionne par **Esquisse (2D)**. Une fois les formes dessinées en 2D, les **Fonctions** permettent de traiter la 3D.

Toutes les opérations effectuées dans un fichier Solidworks sont repérées dans l'arborescence en gauche. On y remarque que chaque pièce possède, une origine ainsi que trois plans (face, dessus, droite).

1.3.1 Gros cylindre de la base



 Cliquer sur **Plan de face**

 Dans l'onglet **Esquisse**, cliquer sur **Esquisse** pour en créer une nouvelle

 Choisir le **cercle** (en haut dans l'onglet Esquisse)

 Cliquer sur l'origine du repère puis faire bouger la souris et cliquer n'importe où. On ne s'intéresse pas à la dimension à ce stade (**ne pas** rentrer le rayon non plus).

 Cliquer sur **Cotation Intelligente** puis cliquer sur le cercle. La cotation s'affiche, y rentrer le diamètre de ce cylindre : 26mm d'après le plan.

Remarque : Les cotations **intelligentes** permettent de garder la cotation même si d'autres paramètres sont amenés à changer. On utilisera **TOUJOURS** les **cotations intelligentes**.

 Maintenant que la forme est fermée et de bonne dimension, il faut ajouter de la profondeur. Pour cela, dans l'onglet **Fonction**, cliquer sur **Base/Bossage extrudé**.

 Dans **Direction 1** choisir dans le menu déroulant **Plan milieu** et renseigner la cote de 20mm.

 La vue peut être changée avec les flèches du clavier ou en maintenant le clic molette de la souris.

 Si la fonction vous convient, **valider**.

1.3.2 Grand et petit cylindres de l'arbre

Un cylindre a été dessiné à partir du plan de face. D'après le plan, les cylindres de l'arbre sont orthogonaux à ce premier cylindre. Ainsi, nous allons choisir un autre plan (orthogonal au plan de face).

-  Choisir le **Plan de dessus**.
-  Réaliser un cylindre de diamètre 17mm et de hauteur 45mm conformément aux plans. Pour cela, sélectionner toujours l'origine comme centre du cercle. Dans la fonction **Base/Bossage extrudé**, choisir **Borgne** (et pas Plan milieu).
-  Faire de même pour le cylindre de diamètre 15mm et de hauteur 75mm .

1.3.3 Alésage

A ce stade, la pièce devrait prendre forme avec ses trois cylindres. Il reste à percer le cylindre de la base.

-  Choisir un plan adéquat (soit une face de ce cylindre, soit le plan de face) pour créer une nouvelle **Esquisse**.
-  Y dessiner un cercle de 22mm de diamètre, toujours avec l'origine comme centre (avec **Cotation intelligente** bien sûr).
-  Dans l'onglet **Fonction**, sélectionner **Enlèvement de matière extrudé** et choisir les bons paramètres en fonction de l'aperçu 3D. Penser à changer **Borgne** en **A travers tout les deux** pour la pièce soit creusée dans les deux directions.

1.3.4 Finitions

Sur les plans, on peut remarquer qu'entre les deux cylindres orthogonaux il y a un **Congé**, c'est à dire une surface arrondie, l'arrête n'est pas *aiguë*.

-  Sélectionner l'arête entre les deux cylindres puis cliquer sur **Congé** et renseigner le rayon de 1mm

- Aussi, au bout du grand cylindre, il y a un **Chanfrein**, c'est à dire que l'arrête est tronquée.
-  Sous l'icône congé, cliquer sur la flèche noire et sélectionner **Chanfrein** puis renseigner l'angle et la distance tels qu'affiché sur les plans.

Enfin, enregistrer votre pièce et fermer le fichier.

1.4 Assemblage des composants

L'objectif de cette partie est de construire le système en assemblant les composants. Le système est composé des 5 ensembles cinématiques présentés sur la Figure 2.

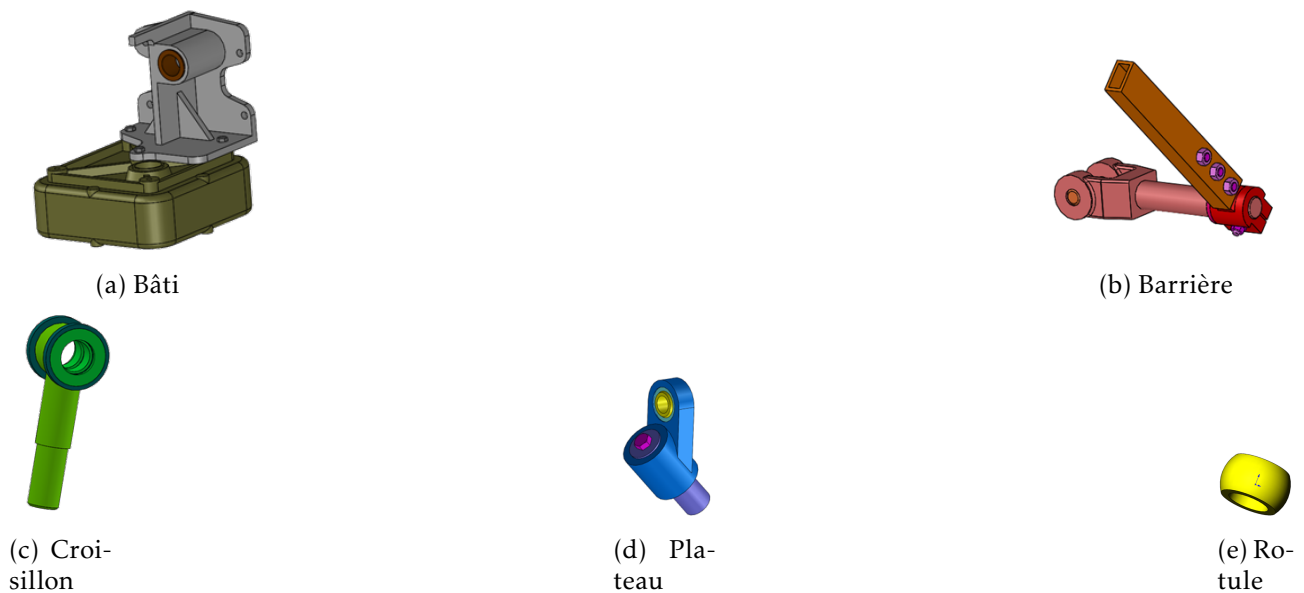
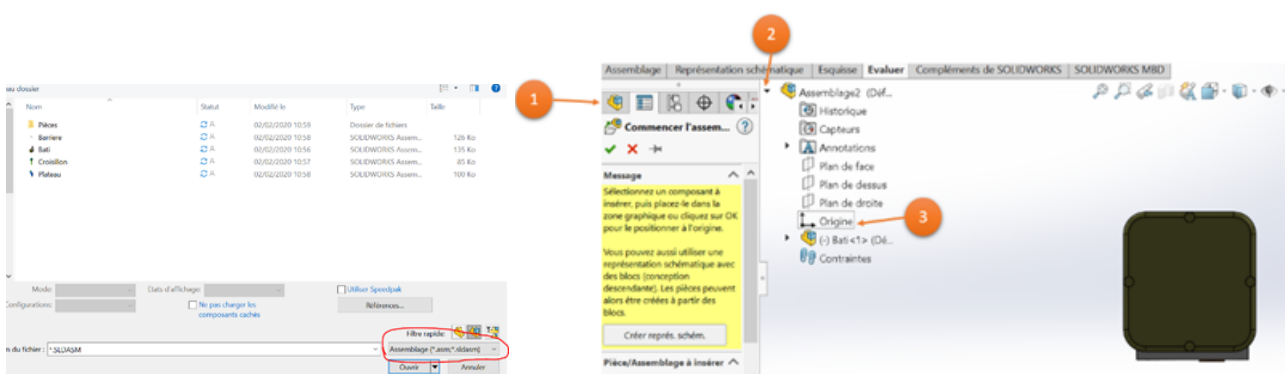


FIGURE 2 – Ensembles cinématiques du système Sinusmatic

Certains de ces ensembles sont des pièces, d'autres sont déjà des assemblages.

1.4.1 Création de l'assemblage et insertion du bâti

- Créer un nouvel **Assemblage** (*Ctrl+N*, ou)
- Dans le dossier *sunismat* choisir le **bâti** et cliquer sur **Ouvrir**.
- L'objectif est de positionner le bâti correctement dans l'assemblage, il faut sélectionner l'origine via l'arborescence de l'assemblage. Pour cela, cliquer sur pour faire apparaître l'arbre d'assemblage puis cliquer sur la flèche noire pour le développer. Enfin, cliquer sur **Origine** pour que le repère lié au bâti soit confondu avec le repère de l'assemblage du système.



1.4.2 Insérer des composants

Afin d'insérer les autres composants :

- Sélectionner l'onglet **Assemblage** puis **Insérer des composants**
- Choisir les bons fichiers (3 assemblages et la pièce **Rotule BI**)
- Ouvrir** puis cliquer n'importe où dans l'espace de l'assemblage pour y déposer les composants

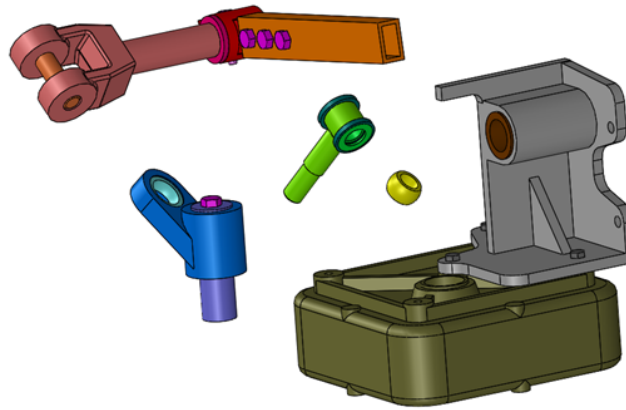


FIGURE 3 – État actuel de l'assemblage

1.4.3 Ajouter des contraintes aux composants

L'objectif est de **correctement** lier les composants entre deux afin que le système fonctionne comme souhaité, y compris dans l'*add-on* Meca3D.

Meca3D est un complément de Solidworks qui permet de réaliser des animations, des études cinématiques, statiques, dynamiques... Celui effectue des **calculs numériques à partir des modèles de liaisons** telles que nous les connaissons. Afin de limiter le travail dans Meca3D et que celui-ci puisse créer automatiquement toutes les liaisons, il faut définir convenablement les contraintes dans Solidworks.

Les contraintes à imposer aux composants sont des **contraintes élémentaires**, notamment plan-plan, cylindre-cylindre et sphère-sphère.

Ainsi, pour réaliser une liaison pivot, d'après la fiche des liaisons, il faut une contrainte plan/plan **ET** une contrainte cylindre/cylindre.

Il est rappelé que les liaisons sont définies par rapport aux géométries de contact, dans les cas rencontrés sur Solidworks, il ne s'agit que de surfaces. **Pour réaliser les contraintes, il faut sélectionner des surfaces et non pas des arêtes**, sinon Meca3D ne pourra construire automatiquement les liaisons (et il faudra modifier...).

Avant de se lancer dans les contraintes sur Solidworks, voici le graphe de liaison du système est présenté sur la Figure 4.

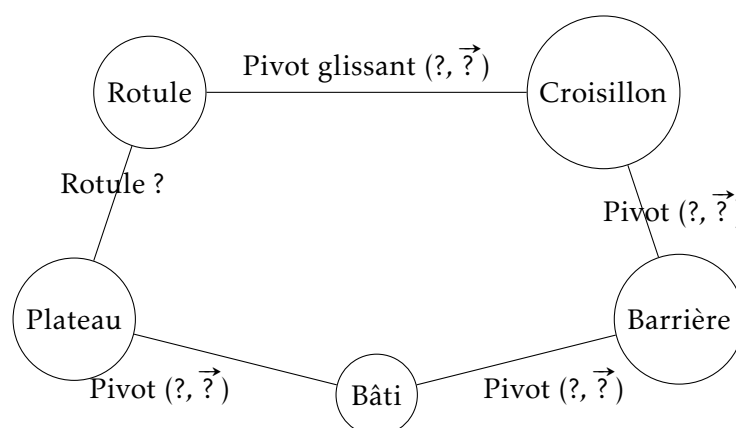


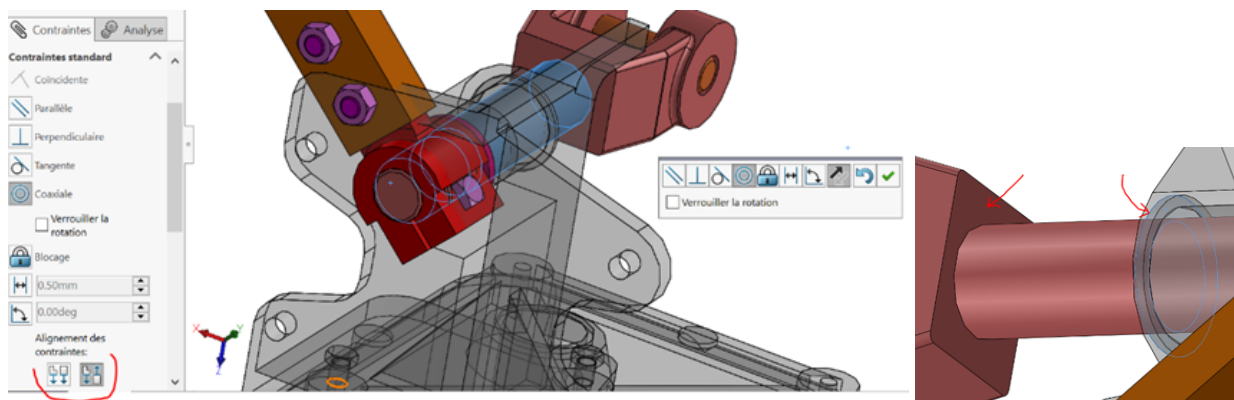
FIGURE 4 – Graphe de liaison du système

Dans un premier temps, voici comment réaliser la liaison pivot entre la barrière et le bâti.

-  Dans l'onglet **Assemblage**, cliquer sur **Contrainte** 
-  Cliquer sur la surface cylindrique de la barrière, puis sur la surface cylindrique l'alésage dans le bâti. Par défaut, le type de contrainte est **coaxialité**, c'est à dire le même axe pour les deux cylindres. Il s'agit de la bonne contrainte.
-  Si la barrière se positionne dans le mauvais sens (la barrière dans le mécanisme et le mécanisme à l'extérieur), il suffit d'**inverser le sens** à l'aide du bouton avec les flèches.

Pour l'instant cela revient à une liaison pivot glissant. Il faut ajouter une contrainte plan/plan pour bloquer la translation et obtenir une liaison pivot.

-  Refaire une contrainte en prenant un plan sur la barrière et le plan sur lequel il doit être en contact sur le bâti. Une contrainte **coïncidente** est proposée, c'est à dire que les deux plans sont confondus. **Valider**, car il s'agit de la bonne contrainte.
-  En utilisant le même principe, finir l'assemblage du système.



Une fois toutes les contraintes définies, l'assemblage doit ressembler à celui présenté sur la Figure 5. En déplaçant le plateau (rester appuyer sur le clic gauche et bouger), la barrière devrait entrer en rotation.

 Enregistrer l'assemblage.

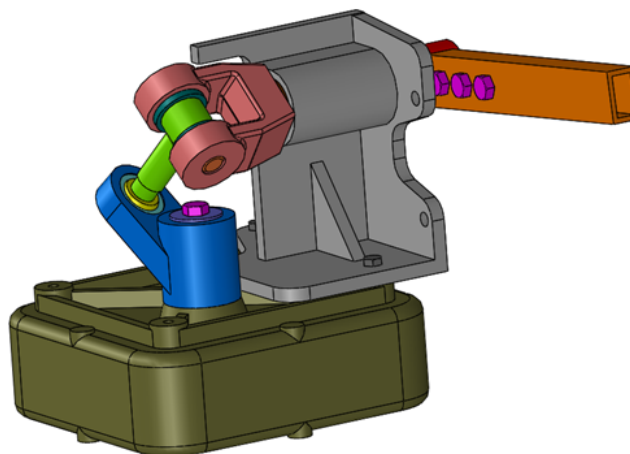


FIGURE 5 – Assemblage terminé

1.5 Meca3D

Dans un premier temps, il faut activer Meca3D. Cette opération sera à répéter lors des TP (en fonction des postes, des sessions...).

- ⚙ Dans le menu en haut, cliquer sur la flèche noir à droite du symbole de roue dentée (engrenage, pour option).
- ⚙ Aller dans **Compléments**, puis tout en bas, activer la version la plus récente de Meca3D en cochant les **deux cases** (une de chaque côté).
- ⚙ Un message s'affiche disant que le complément sera actif à la prochaine ouverture d'un assemblage. Fermer votre assemblage, **pas Solidworks**, puis le rouvrir.
- ⚙ L'icône Meca3D  doit s'ajouter à la fin du menu qui commence par 

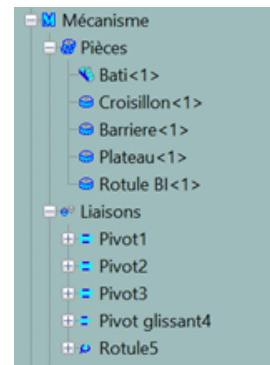
1.5.1 Création des pièces et des liaisons

 Cliquer sur l'icône Meca3D 

 Cliquer droit sur **Mécanisme** puis **Construction automatique**

Si tout est bien défini, en dessous de **Mécanisme**, doit apparaître 5 pièces (les ensembles cinématiques) et les 5 liaisons correspondantes au graphe de liaison (Figure 4).

Si ce n'est pas le cas, les contraintes ont été mal définies sur l'assemblage (contrainte en double, oubli, sélection d'arête au lieu de surface...). Il faut corriger avant de continuer puis refaire une construction automatique dans Meca3D.



1.5.2 Animation

 Cliquer droit sur **Analyse** puis **Calcul mécanique** La fenêtre doit annoncer 2 degrés de mobilité. En effet, la rotation de la rotule autour du croisillon est autorisée même si le système est à l'arrêt. C'est une mobilité interne.

Dans la fenêtre suivante, il y a deux mouvements à fixer. Pour choisir le mouvement il faut double-cliquer la liaison (dans la fenêtre), un menu déroulant s'affiche, ensuite, le cas échéant, faire de même pour le degré de liberté de la liaison.

-  Pour un premier mouvement, choisir la **liaison pivot glissant** et fixer à 0 tr.min^{-1} la vitesse de rotation.
-  Pour le second mouvement, choisir la **liaison pivot entre le plateau et le bâti** et fixer à 1 tr.min^{-1} la vitesse de rotation.
-  Plus bas, sélectionner **Etude cinématique**.
-  Choisir une durée de simulation de 60s (dans l'objectif de réaliser un tour).
-  Choisir un nombre de point de 60, il s'agit du nombre de fois où le logiciel fera le calcul au cours du mouvement.
-  Lancer le calcul en appuyant sur le **bouton start vert** puis fermer la fenêtre avec le bouton **en bas** à droite.
-  Cliquer droit sur **Résultats** puis **Simulation** (ou cliquer sur l'icône  en haut). La barrière doit fonctionner.

1.5.3 Loi entrée/sortie

La loi entrée/sortie (E/S) est la relation entre :

- le paramètre de sortie, ici l'angle de la barrière ;
- le paramètre d'entrée, celui qui est commandé, ici l'angle du plateau la pièce bleue.

Afficher la courbe de la loi entrée/sortie

 Cliquer droit sur **Courbes** puis **Ajouter**.

Il y a trois possibilités :

1. **La courbe simple** : il s'agit de l'évolution d'un paramètre par rapport au temps.
2. **La courbe paramétrée** : il s'agit de l'évolution d'un paramètre par rapport à un autre.
3. **La courbe multiple** : il s'agit de l'évolution de plusieurs paramètres par rapport au temps.

 Choisir **Paramétrée** car la loi entrée/sortie représente l'évolution d'un paramètre par rapport à un autre.

Avant de continuer, il convient de comprendre ce que Meca3D a fait.

Meca3D crée un repère pour chaque liaison (en rouge) en plus des repères associés aux ensembles cinématiques (en bleu). Chacun de ses repères est bien sûr composé de trois vecteurs. Au bout de la représentation de ces vecteurs, il y a 1 cône, ou 2 cônes ou 3 cônes.

 En abscisse, choisir l'angle de la liaison pivot entre le plateau et le bâti. Pour cela, cliquer sur la liaison dans l'**arborescence** Meca3D. Le paramètre souhaité une **position**. Il reste d1, d2, d3, norme, a1, a2, a3 ; il s'agit des distances (d) ou des angles (a) pour le vecteur représenté avec 1, 2 ou 3 cônes. Il est aussi possible d'utiliser l'aperçu pour conforter son choix.

 En ordonnée, choisir l'angle de la liaison pivot entre la barrière et le bâti. Il est possible de changer la pièce de référence, **garder le bâti**.

 Une fois validée, il est possible d'afficher la courbe en faisant un clic droit dessus puis **Afficher**.

Il est possible que la courbe obtenue ne soit pas exactement celle présentée sur la Figure 6 et quelle soit décalée suivant l'axe des abscisses. La position initiale n'est peut être pas la même pour le calcul.

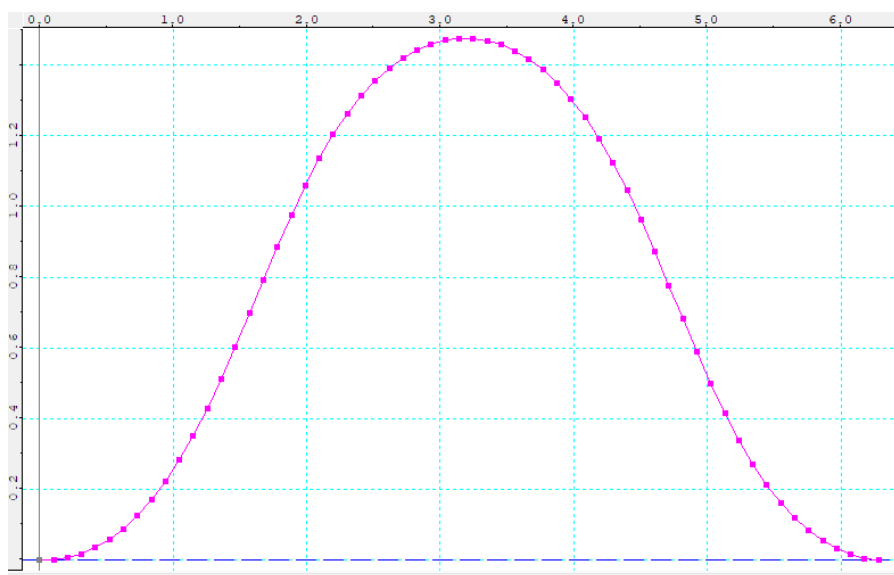


FIGURE 6 – Courbe obtenue sous Meca3D

Changer la position initiale

 Bouger le système dans la position souhaitée, ou imposer une contrainte pour que le positionne-

ment soit parfait, par exemple une contrainte parallèle ou perpendiculaire (**Bouger à la main et ne pas ajouter de contrainte pour ce système**).

-  **Reconstruire** l'assemblage en appuyant sur le bouton  dans le menu tout en haut.
-  Il est possible (conseillé) de supprimer la contrainte ajoutée. Pour cela, tout en bas de l'arborescence, dans **Contraintes**, cliquer droit sur la contrainte ajoutée puis l'icône  **Supprimer** qui permet de la réactiver au besoin.
-  Il faut alors relancer le calcul, la position initiale ayant changée.

Exporter les données

L'objectif sera de **comparer** ces **données simulées** aux **données expérimentales** et **théoriques** en les affichant sur une même figure Python.

 Pour exporter les données, afficher la courbe puis cliquer dans la zone bleue (avec le tableau de données) puis **Enregistrer les données**.

 **Attention à bien choisir le format .txt**

1.6 Tracer la courbe avec Python

 Supprimer les trois dernières lignes du fichier .txt qui génèrent l'interprétation Python.

 Ouvrir le fichier **sinusmat.py**

 En vous aidant au besoin du TP d'initiation à Python pour la SI, faire en sorte d'obtenir la courbe présentée sur la Figure 7. *Noter que le code est volontairement génèreux.*

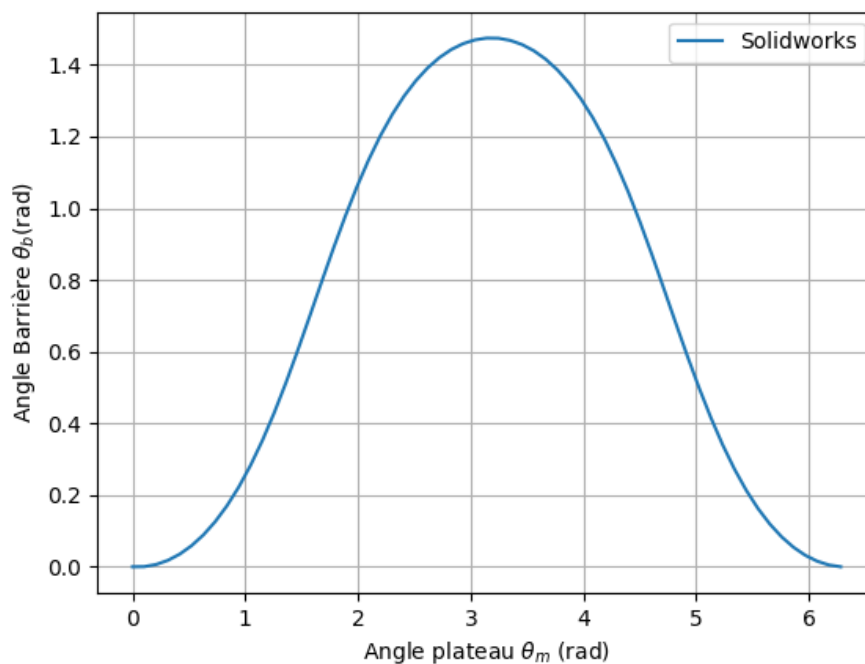


FIGURE 7 – Courbe Python des données simulées

2 | Micromoteur

2.1 Objectif

L'objectif de ce TP est de mettre en pratique, en étant moins guidé, la réalisation d'une simulation Meca3D.

2.2 Présentation

Il s'agit du micromoteur qui a été, ou sera *en fonction du groupe* traité en TD. Ce moteur de modélisme est un moteur thermique monocylindre qui équipe des maquettes de bateaux. Il s'agit d'un type de moteur dont le cycle de fonctionnement n'a que deux temps. En effet, l'admission et le refoulement des gaz se fait en même temps et ce moteur ne possède pas de soupape.

Il est constitué de 4 ensembles cinématiques, présentés sur la Figure 8.

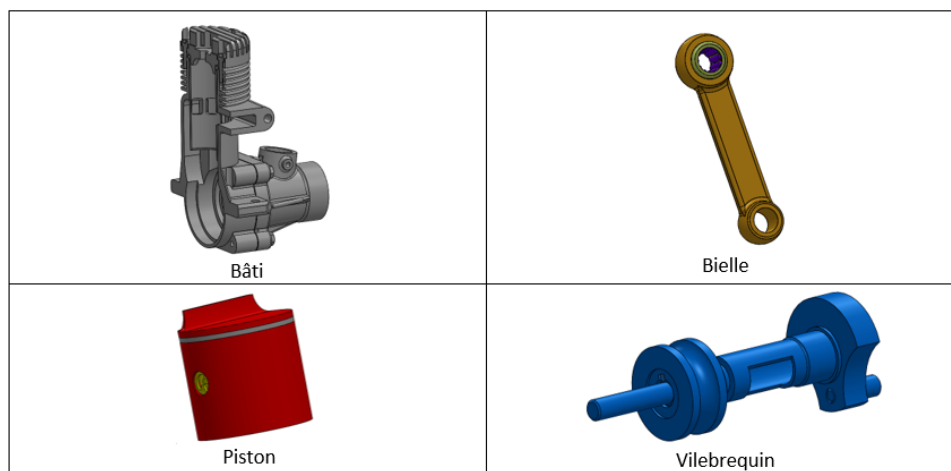


FIGURE 8 – Ensembles cinématiques du micromoteur

2.3 Assemblage des composants

 Créer un nouvel assemblage et réaliser l'assemblage du système.

 A l'ouverture de certains fichiers, il est possible d'avoir des pop-up demandant des mises à jours, **accepter**. Aussi, la bielle peut avoir des erreurs mais il n'y a aucun impact (problème lié aux versions).

Remarque : La contrainte de coaxialité entre la bielle et le piston suffit, **ne pas ajouter** une contrainte plan-plan.

L'assemblage terminé est présenté Figure 9.

2.4 Meca3D

 Sous Meca3D, réaliser la **Construction automatique**, le système comprend 4 pièces et 4 liaisons (2 pivots et 2 pivots glissants).

 Réaliser le calcul pour une **étude cinématique**. Pour cela, fixer une vitesse de rotation à la liaison entre le bâti et l'arbre d'entrée puis donner la durée de la simulation en faisant en sorte de ne faire qu'un tour de vilebrequin. Enfin, avant de lancer le calcul, renseigner un nombre de points de calculs.

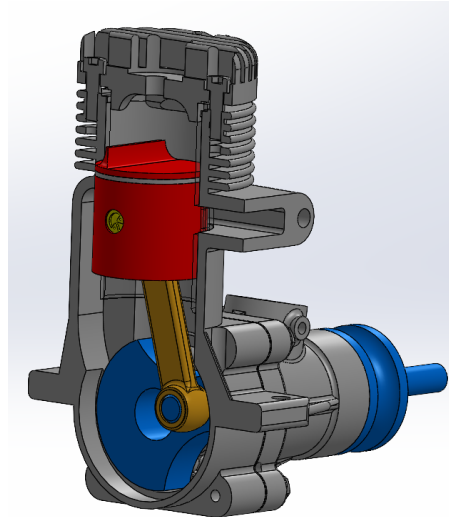


FIGURE 9 – Assemblage du micromoteur terminé

2.4.1 Loi entrée/sortie

L'objectif est de tracer l'évolution du déplacement du piston par rapport au bâti en fonction de l'angle de rotation moteur par rapport au bâti.

 Créer une courbe et modifier ai besoin la position initiale afin d'obtenir une courbe similaire à celle de la Figure 10.

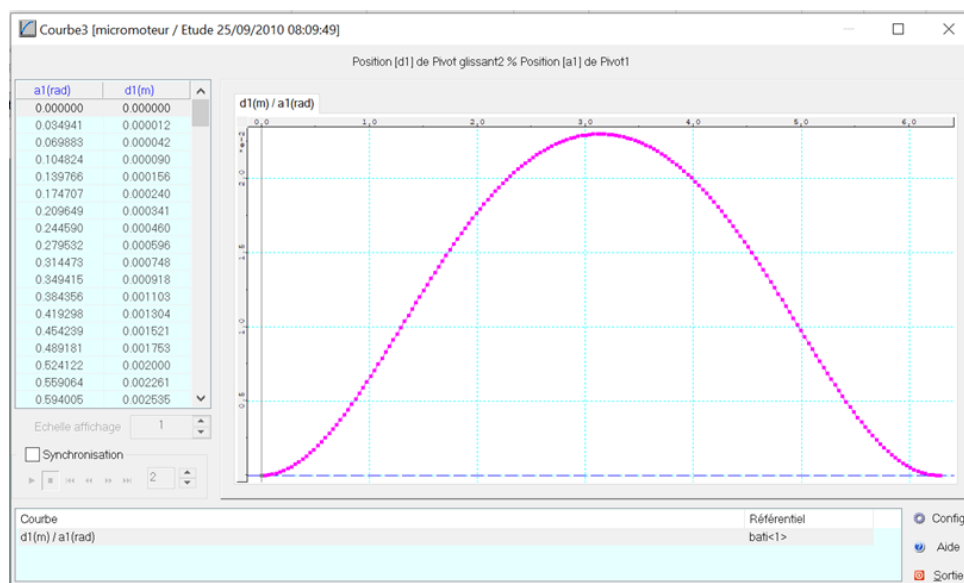


FIGURE 10 – Courbe obtenue sous Meca3D

 Exporter les données en .txt puis supprimer les trois dernières lignes du fichier.

2.5 Étude théorique

Le schéma cinématique du micromoteur est donné Figure 11, avec $\vec{OA} = R\vec{x}_1$, $\vec{OB} = x(t)\vec{x}_0$ et $\vec{AB} = -L\vec{y}_2$.

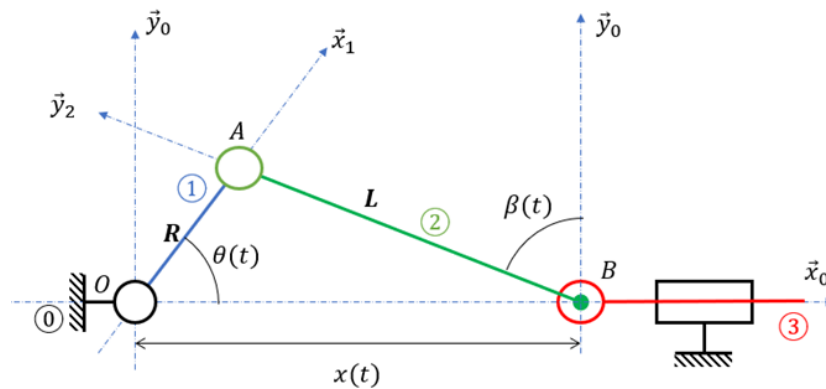


FIGURE 11 – Schéma cinématique du micromoteur

L'étude en TD donne $x(t) = R \cos \theta + \sqrt{L^2 - R^2 \sin^2 \theta}$.

2.6 Exploitation sous Python

L'objectif est de comparer les résultats simulés (obtenus sur Solidworks) aux résultats théoriques.

- 🔗 Ouvrir le fichier **Micromoteur-Etudiant.py**
- 🔗 Vérifier le code (la fonction *donneesSldW(NomFichier)*, son appel...) pour tracer les données simulées, notamment le nom de votre fichier texte.
- 🔗 Exécuter le programme et comparer la courbe obtenue à celle sur Meca3D, elles devraient être identiques.
- 🔗 Remplir la partie analytique (théorique) en utilisant la fonction *linspace* de *numpy*.
- 🔗 Compléter la fonction *CalculPosition(theta)* qui calcul $x(t)$ pour un θ donné.
- 🔗 Compléter la partie **Calcul Analytique**.
- 🔗 Enfin, décommenter la ligne pour tracer la courbe théorique et exécuter le script.
- 🔗 Comparer ces courbes, si elles ne sont pas confondues, émettre des hypothèses, identifier les problèmes éventuels et les résoudre.

2.7 Retour sur la partie théorique

Pour établir l'expression de $x(\theta)$, une fermeture géométrique a été établie, puis par projection sur la base $(\vec{x}_0, \vec{y}_0)$ le système d'équation suivant est obtenu :

$$\begin{cases} x - L \sin \beta - R \cos \theta = 0 \\ L \cos \beta - R \sin \theta = 0 \end{cases}$$

Où x , β et θ dépendent du temps.

Supposons que nous ne sachions pas résoudre ce système sur papier ; comment avoir une résolution numérique ?

Pour résoudre numériquement un système, il existe la fonction *fsolve* de module *optimize* de la bibliothèque *scipy*. Ainsi, pour la charger, on peut écrire en entête :

```
from scipy.optimize import fsolve
```

La fonction *fsolve* est définie de la manière suivante :

```
fsolve(système, initialisation, arguments)
```

- 🔗 Se renseigner sur la fonction *fsolve* de sorte à résoudre le système avec Python.