Compte Rendu : Outils numériques

Boris Baudel – ENS Rennes

Simulations numériques : Simscape et MultyBody

Introduction

Le but de ce Tp est de s'initialiser à la modélisation Matlab-Simulink de divers systèmes, en comparant par rapport aux solutions exactes de leurs équations différentielles. De même, nous allons étudier la construction mécanique de divers Solides d'un système composée de plusieurs Solides.

Par suite de cela, nous allons étudier un système masse ressort qui sera modéliser sur le logiciel Matlab-Simulink. On effectuera une comparaison entre les solutions exactes de la position et vitesse.

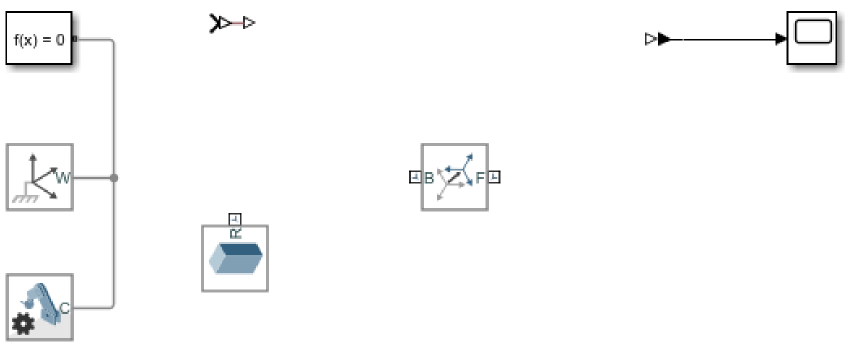
Après, nous allons effectuer la modélisation d'un pendule simple à l'aide de Simulink, que nous allons comparer à la solution exacte.

Finalement, nous allons réaliser une étude permettant de vérifier le dimensionnement de trois moteurs du Robot Scara qui sera modélisé grâce aux compléments Matlab Simscape Multybody. Un robot dont la structure mécanique nous a été fournie sur le logiciel SolidWorks.

Activité I : construction d'un système mécanique

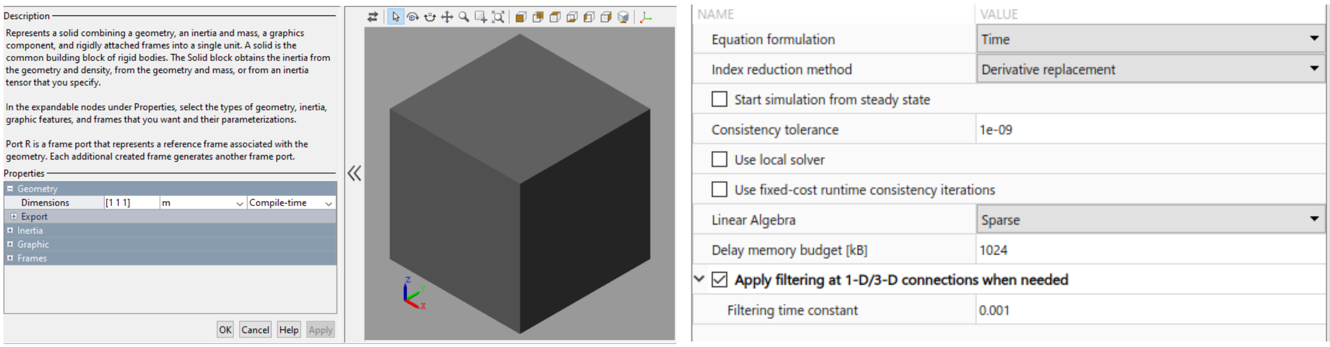
Question 1 :

Tout d'abord, pour créer un système sur le logiciel Simulink, il convient de comprendre les éléments de bas permettant de pouvoir modéliser le système.



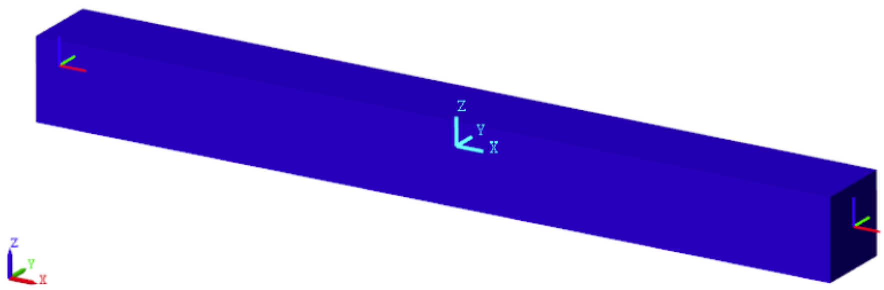
Par exemple, le bloc « solver configuration » sert à déterminer la solution avec une précision définie. Le deuxième permet de définir un repère. Le troisième bloc « mechanism » configuration permet de définir une gravité dans le système. Le « scope » permet de relever des valeurs à un endroit que l'on définit du système (vitesse, position, accélération, etc..) . Le bloc « Solid » permet de générer des volumes définis par ses caractéristiques, l'inertie, masses volumiques, dimensions qui le définisse.

Le bloc «rigid transform » permet de définir une rotation, translation. Il est indispensable lorsque l'on veut effectuer des liaisons entre des solides.



Question 2 :

Nous modifions les paramètres du solide pour avoir le paralepipède donnée de largeur 10 fois plus petite que sa longueur et en couleur bleue. De plus, on place les repères aux extrémités, à l'aide de « rigide transform ». Par ailleurs, il nous faudra aussi, modifier les dimensions de la pièce de manière successive à l'aide d'un script.

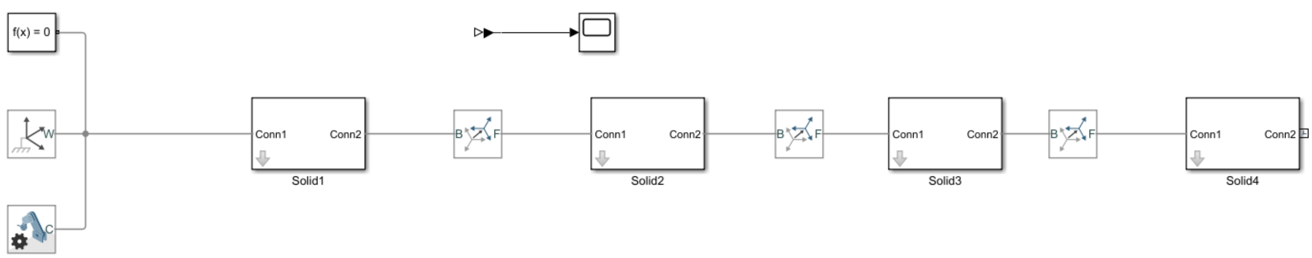


```
Solid1.L = 100 ;
Solid1.W = Solid1.L/10 ;
Solid1.H = 10 ;
Solid1.rgb = [0 0 1] ;

Solid2.L = 50 ;
Solid2.W = Solid2.L/10 ;
Solid2.H = 10 ;
Solid2.rgb = [1 0 0] ;

Solid1.rho = 1 ;
Solid2.rho = 1 ;
```

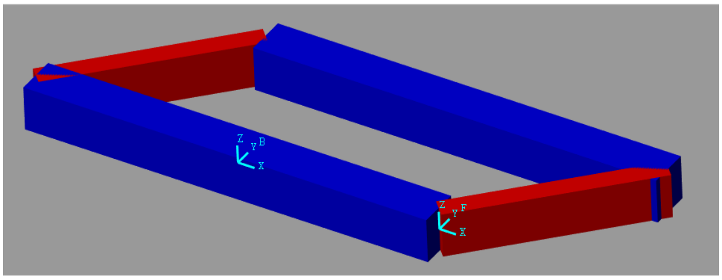
Les repères d'entrée et de sortie du système sont donc ceux situés aux extrémités de la pièce « Solid 1 »

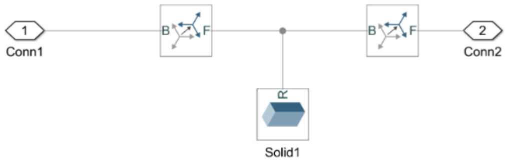
I Positionner une pièce par rapport à une autre

Nous devons maintenant dupliquer les pièces successivement, en créer la pièce rouge qui permettra de faire le pont du parallélogramme. Pour positionner une pièce par rapport à une autre, on effectue une série de groupes Solid alternée de « rigide transform » pour permettre la rotation de différents solides de 45°. Dans chaque bloque « Solid »nous allons placer deux « rigid transform » avec des translations de L/2 afin de pouvoir bien définir plus facilement les rotations à effectuer. En effet, les solides vont donc former une structure carrée par les positions successives que les solides vont obtenir.



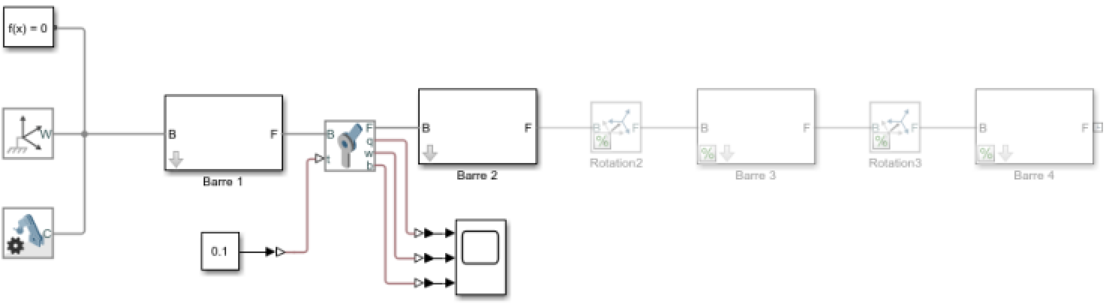
Dans chaque bloque, Solid1, Solid2, Solid3, Solid4, nous devons placer deux « rigid transform » et le solide en l'occurrence. Puis on les place en série comme il est montré, dans la parie précédente. À la fin, on obtient donc la structure suivante :





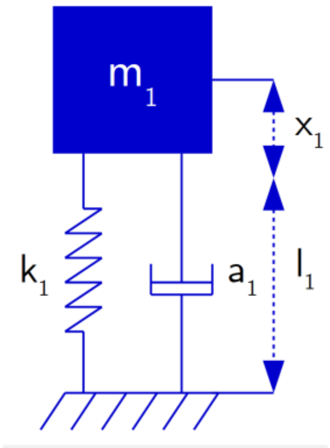
Question 3 :

Le bloc « revolute joint » permet d'établir une liaison pivot, et de générer une orientation angulaire d'une pièce, il pourra modifier l'angle d'une pièce tout en maintenant les contraintes du parallélogramme précèdent. Dans le système du « revolute joint » nous avons la possibilité d'implémenter des caractères tels qu'un ressort en torsion, une raideur, un facteur de frottement s'il s'agit d'une liaison pivot. De même, on peut paramétrer les valeurs d'entrée s'il s'agit d'un signal et on peut mesurer à l'aide de « sensing » toutes les valeurs de la liaison (vitesse, angle, position, accélération etc…)



Activité II : modélisation de systèmes mécaniques simples

I Modélisation d'un système ressort-amortisseur



$m_1 = 10\,kg,$

$c = 0.1\,m,$

$k_1 = 1000\,N.m^{-1},$

$l_1 = 0.1\,m,$

$a_1 = 0.1\,N.m^{-1}.s^{-1},$

position initiale de la masse $x_{1_0} + l_1 = 1\,m,$

vitesse initiale de la masse $v_{1_0} = 0\,m.s^{-1}.$

Question 4 :

On applique le théorème de la résultante dynamique et selon l'axe x aux solides de masse m :

$$m\frac{dx^2}{dt^2} = -kx - mg - a\frac{\mathrm{d}x}{\mathrm{d}t} \Leftrightarrow m\frac{dx^2}{dt^2} + a\frac{\mathrm{d}x}{\mathrm{d}t} + kx = -mg$$

Le polynôme caractéristique et le déterminant associée sont donnés par :

$$\chi = r^2 + \frac{a}{m}r + \frac{k}{m} \qquad \Delta = \frac{a^2}{m^2} - 4\frac{k}{m} < 0$$

Les racines du polynôme caractéristique sont données par :

$$x2 = -\frac{a}{2m} - i\frac{\sqrt{(a^2 - 4km)}}{2m} \qquad x1 = -\frac{a}{2m} + i\frac{\sqrt{(a^2 - 4km)}}{2m}$$

On a donc :

$$x(t) = \lambda e^{-\frac{at}{2m}} cos(\frac{\sqrt{a^2 - 4kmt}}{2m}) + \mu e^{-\frac{at}{2m}} sin(-\frac{\sqrt{a^2 - 4kmt}}{2m}) + Sp$$

On a donc A l'aide des conditions initiales et le fait que la solution particulière est une constante, on a :

$$Sp = -\frac{mg}{k}$$

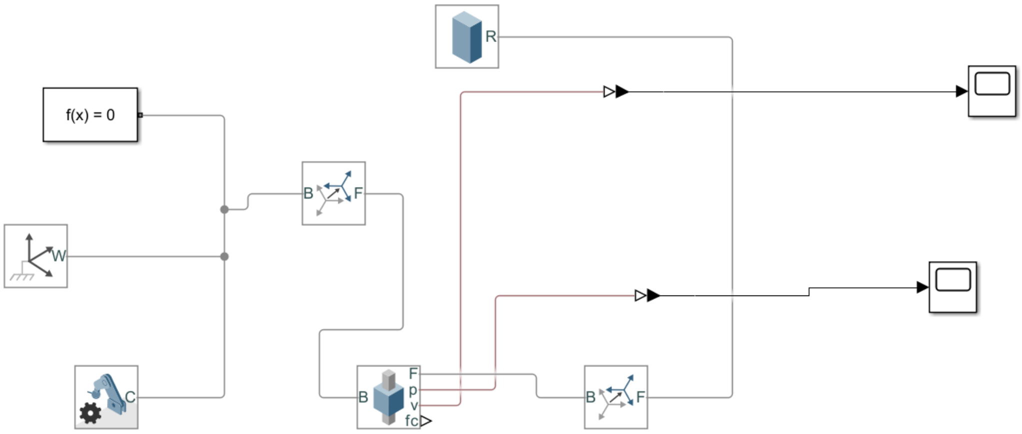
$$\lambda = (xo + l + \frac{mg}{k}) \qquad \mu = \frac{a((xo + l + \frac{mg}{k}))}{\sqrt{(a^2 - 4km)}}$$

La solution finale de l'équation différentielle est donc :

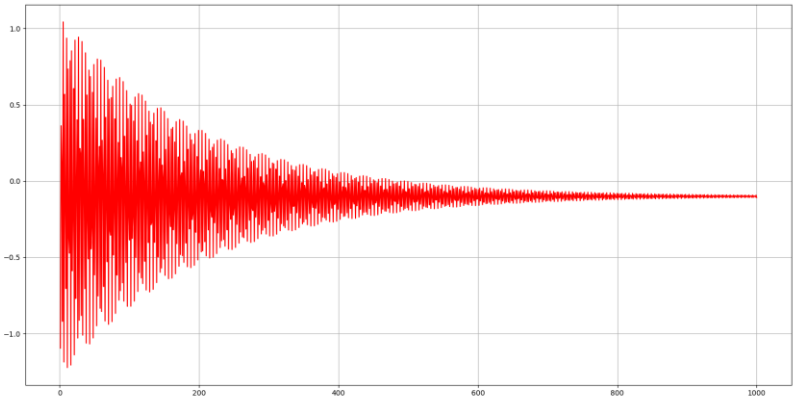
$$x(t) = (xo+l+\frac{mg}{k})e^{-\frac{at}{2m}} cos(\frac{\sqrt{a^2 - 4kmt}}{2m})+\frac{a(xo + l + \frac{mg}{k})}{\sqrt{(a^2 - 4km)}} e^{-\frac{at}{2m}} sin(-\frac{\sqrt{a^2 - 4kmt}}{2m}) - \frac{mg}{k}$$

Question 5 :

Pour créer le modèle Matlab Simulink, il est plus simple de faire une liaison glissière en ajoutant une certaine raideur sur le ressort puis un solide. On obtient donc la figure présentée ci-dessous :

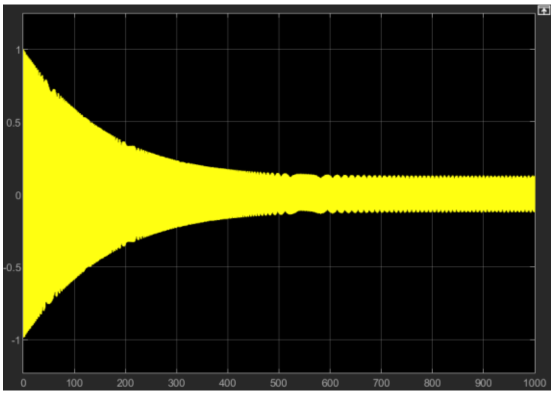


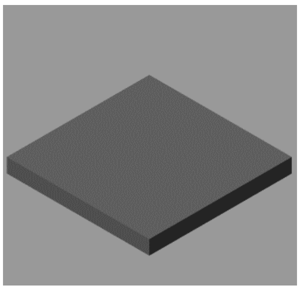
Tout d'abord, on peut calculer la solution exacte de l'équation différentielle donnée précédemment, à l'aide d'un simple programme python :



Puis, avec la simulation Simulink, on obtient, alors des résultats, quasi-identique, à l'aide de la structure présenter précédemment :

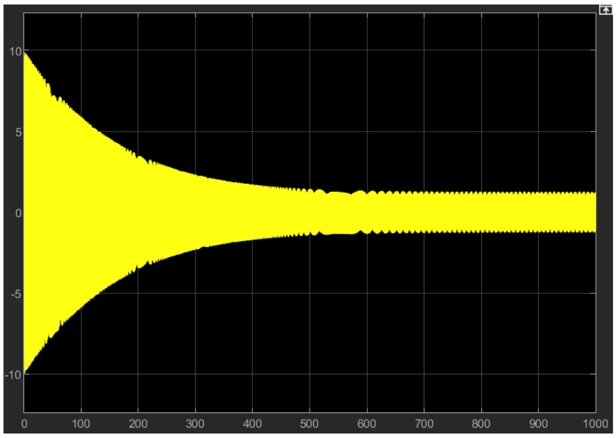
Position (m) en fonction du temps(s)





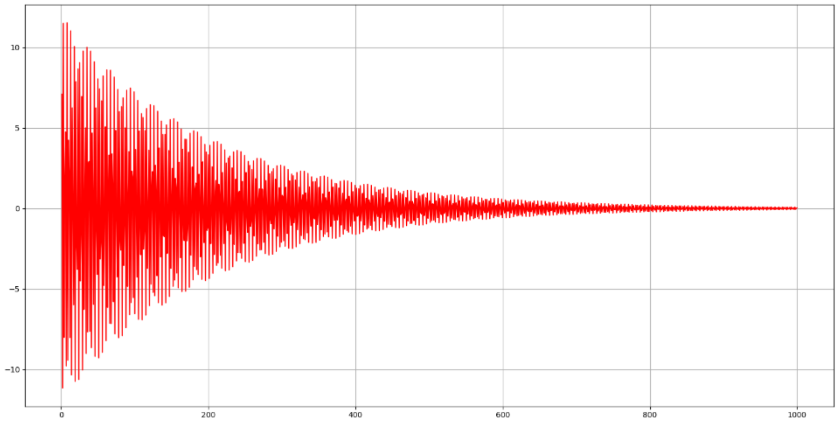
On observer la tendance d'oscillateur amortie par la solution exacte ainsi que par la solution proposée par Matlab-Simulink. En effet, le programme affiche plusieurs harmoniques qu'il serait intéressant de filtrer pour obtenir le premier harmonique, qui donnerait d'autres informations sur le système. Sinon, les deux équations ont les mêmes amplitudes et affiche les mêmes tendances d'oscillateur amortie. Lorsque l'on étudie les vitesses associées aux systèmes, on a, pour la simulation Simulink :

Vitesse (m/s) en fonction du temps(s)



La solution exacte du système en dérivant la position est donnée par la représentation python suivante :

Vitesse (m/s) de la masse m en fonction du temps(s)

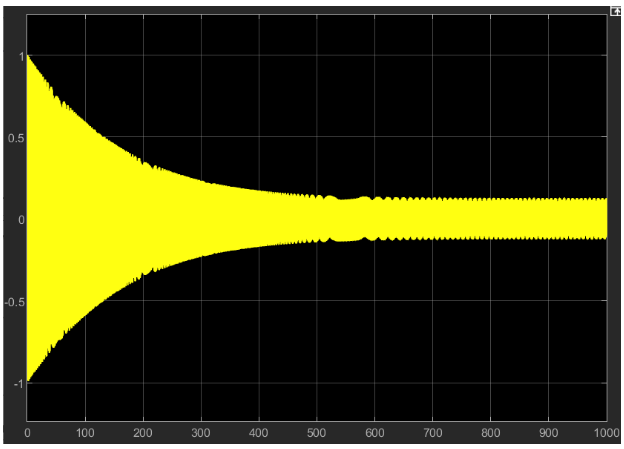


On remarque qu'alors que Simulink donne une représentation quasi la même que celle du programme python. La légère différence se trouve dans le fait que mon programme python utilise une fonction « dérive » et n'est pas exactement exacte.

Question 6 :

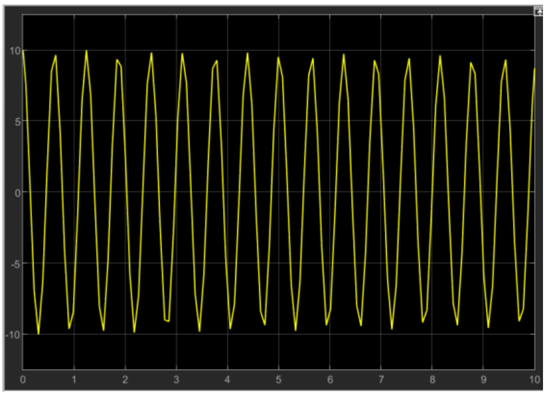
Maintenant, nous allons modifier les positions et les vitesses initiales du système à $x(0) = 0.1$ et on va rajouter une vitesse initiale de 10 mètres par seconde. On effectue ceci sur le logiciel Simulink et on obtient :

Position m de la masse m en fonction du temps(s)

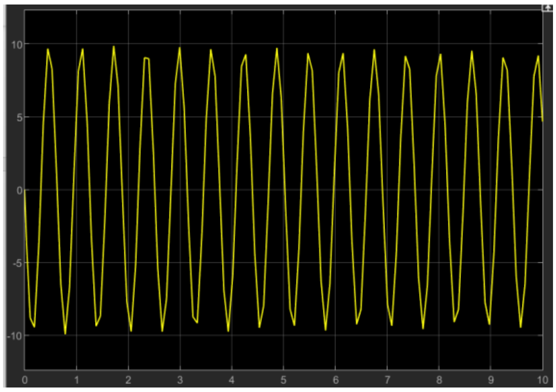


La position du système globale ne varie pas dans la structure globale. Cela dit, la présence d'une vitesse initiale cela pourrait accélérer les oscillations proches du temps 0 ainsi que modifier la vitesse. De la même manière, la structure de la vitesse ne se modifie pas avec ce changement. Sur un temps d'échelle plus petit, on observe bien la différence entre le premier cas et le second, le sens dans lequel sera la première vitesse :

Vitesse (m/s) de la masse m en fonction du temps(s) N 2

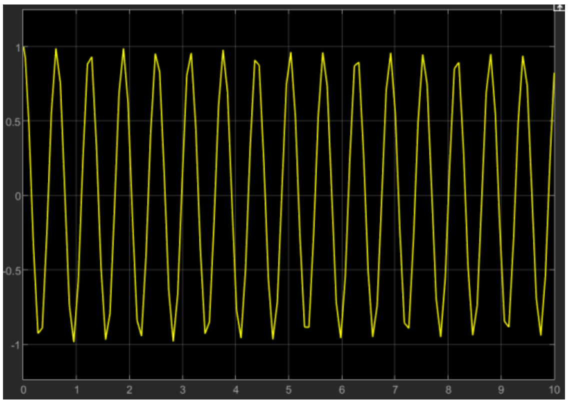


Vitesse (m/s) de la masse m en fonction du temps(s) N 1

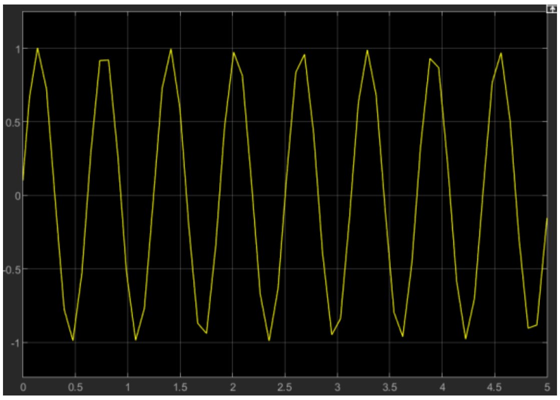


On peut aussi évaluer la différence de position, proche d'un temps initial, on observe bien les différences dans les conditions initiales qui ont été imposées. En revanche, le système prendra, par suite de cela, un déphasage de différence.

Position (m) de la masse m en fonction du temps(s) N 1



Position (m) de la masse m en fonction du temps(s) N 2



Activité II : modélisation d'un pendule

Question 7 :

Le modèle du pendule simple est donné par :

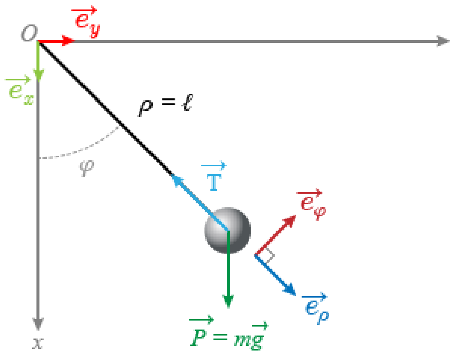


Fig. 2 : Pendule simple

- $m = 10\,kg$,
- $d = 0.1\,m$,
- $l = 2\,m$,
- $\theta = 15°$.

D'après-principe fondamental de la dynamique, par projection, on a :

$$ml\ddot{\theta}\vec{\tau} + ml\dot{\theta}^2\vec{n} = -mg\sin\theta\vec{\tau} - mg\cos\theta\vec{n} + T\vec{n}$$

$$\begin{cases} l\ddot{\theta} = -g\sin\theta \\ T = ml\dot{\theta}^2 + mg\cos\theta \end{cases}$$

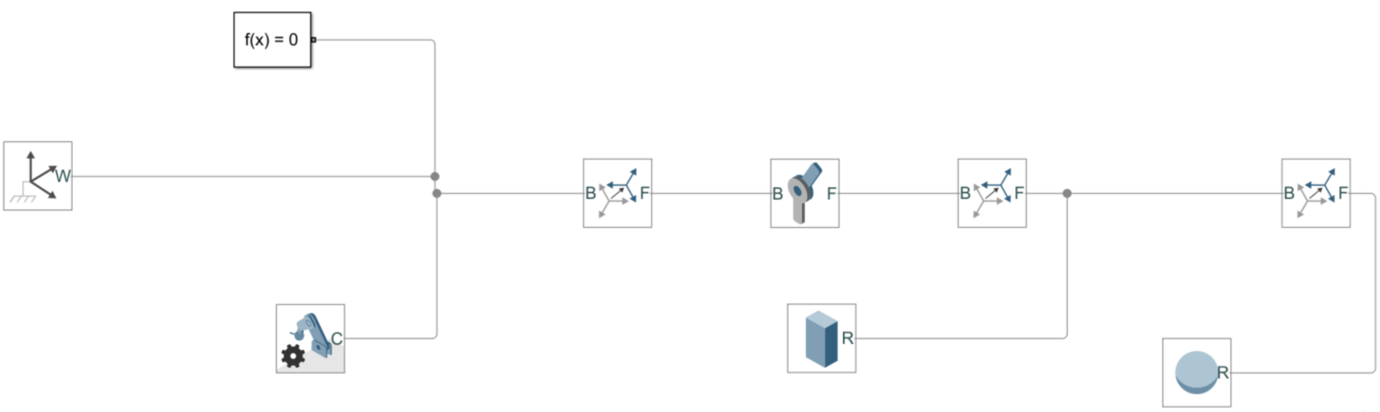
Il vient comme les angles sont supposés petit :

$$\ddot{\theta} + \omega_0^2\sin\theta = 0 \qquad \ddot{\theta} + \omega_0^2\theta = 0 \qquad \omega_0^2 = \frac{g}{l}$$

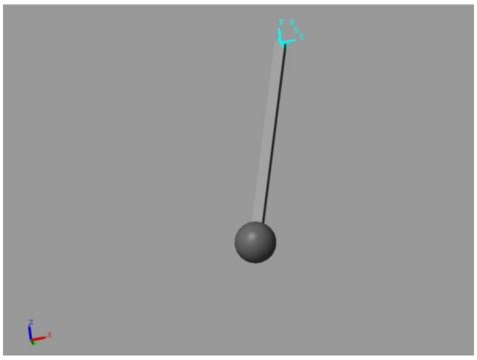
La solution est donc : $\theta(t) = \theta_0 cos(wt)$

Question 8 :

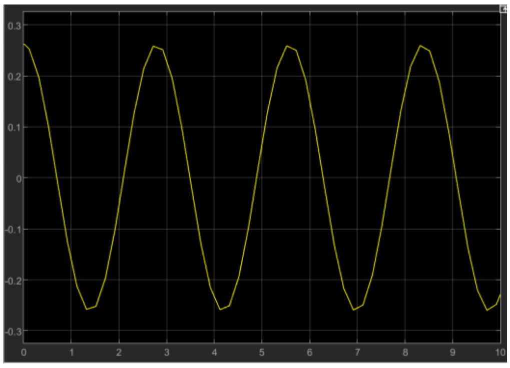
Le système utilisé pour modéliser le pendule simple est composé d'une liaison pivot et de deux solides, une tige et la sphère. Nous effectuons le modèle Simulink suivant qui permet de le modéliser :



On prend les différentes dimensions et on obtient la trajectoire du pendule ci-dessous :

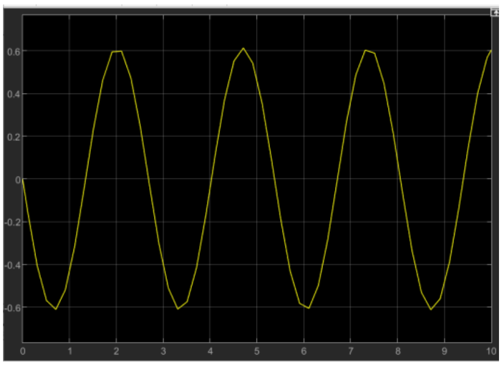


Position (rad) en fonction du temps(s)



On peut relever la vitesse angulaire associée :

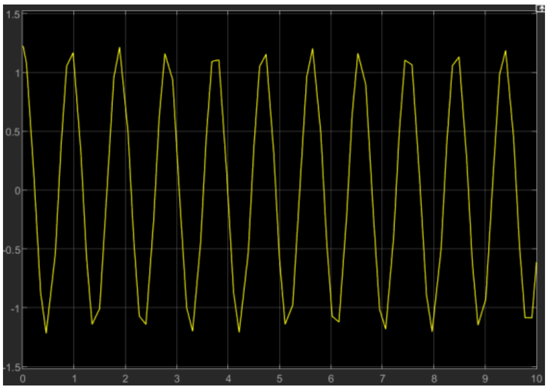
Vitesse angulaire (rad/s) en fonction du temps (s)



Question 9 :

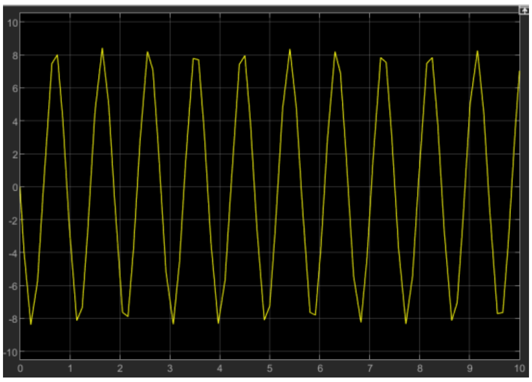
On effectue maintenant une nouvelle modélisation avec des nouvelles valeurs. On relève respectivement la position et la vitesse angulaire, pour les nouvelles valeurs montrer ci-dessous :

Position (m) en fonction du temps(s)



$m = 10\,kg,$
$d = 0.1\,m,$
$l = 0.2\,m,$
$\theta = 70°.$

Vitesse angulaire (rad/s) en fonction du temps (s)

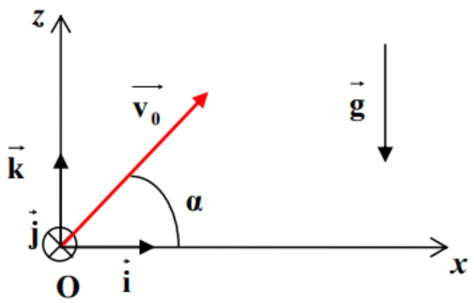


Comme on a diminué la longueur il est normal d'observer que la fréquence d'oscillation augmente d'après ce que nous avons établi dans les équations précédentes. De même, l'amplitude varie étant donné la condition initiale changeante à 1,1 rad.

Activité III : catapulte à frein magnétique

Question 10 :

En utilisant le modèle suivant, on a, à une hauteur h, d'après le principe fondamental de la dynamique :



$$\overrightarrow{v(t)} \left| \begin{array}{l} v_x = v_{0x} = v_0 \cdot \cos(\alpha) \\ v_y = 0 \\ v_z = -g \cdot t + v_{0z} = -g \cdot t + v_0 \cdot \sin(\alpha) \end{array} \right.$$

$$\overrightarrow{OG} \left| \begin{array}{l} x(t) = v_{0x} \times t = (v_0 \cdot \cos(\alpha)) \times t \\ y(t) = 0 \\ z(t) = -\frac{1}{2} \cdot g \times t^2 + v_{0z} \times t = -\frac{1}{2} \cdot g \times t^2 + (v_0 \cdot \sin(\alpha)) \times t \end{array} \right.$$

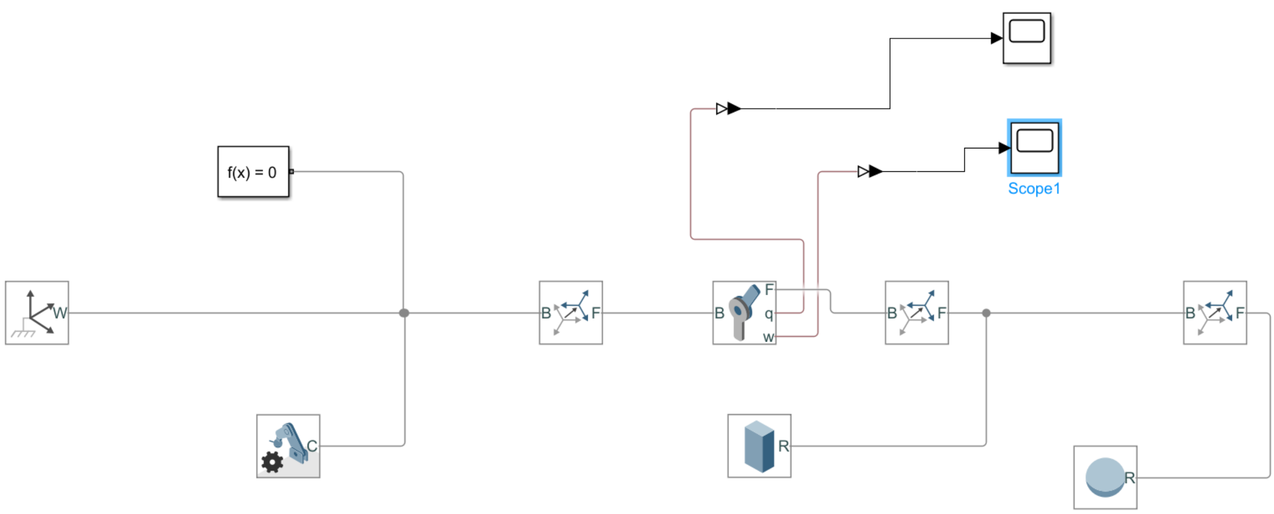
On se rappelle que l'on se situe à une hauteur h et on a, pour obtenir la vitesse initiale :

$$0 = -g \frac{l^2}{2Vo^2 cos(\alpha)} + ltan(\alpha) + h + hsin(\alpha)$$

$$Vo = \sqrt{g \frac{l^2}{2 * cos(\alpha)} * \left(\frac{1}{l * tan(\alpha) + h + hsin(\alpha)}\right)} = 7,5 m/s$$

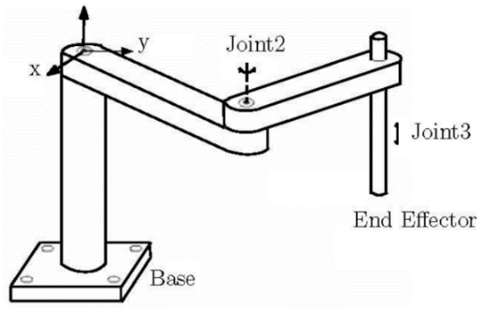
Question 11 :

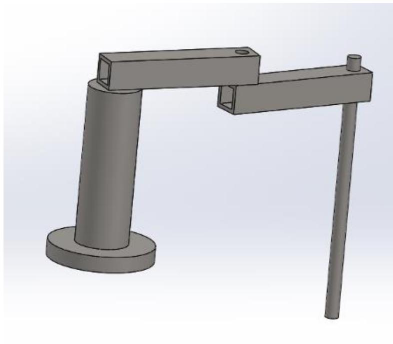
Le système associé à la catapulte est donné par le système ci-dessous :



Activité III : modélisation d'un robot Scara

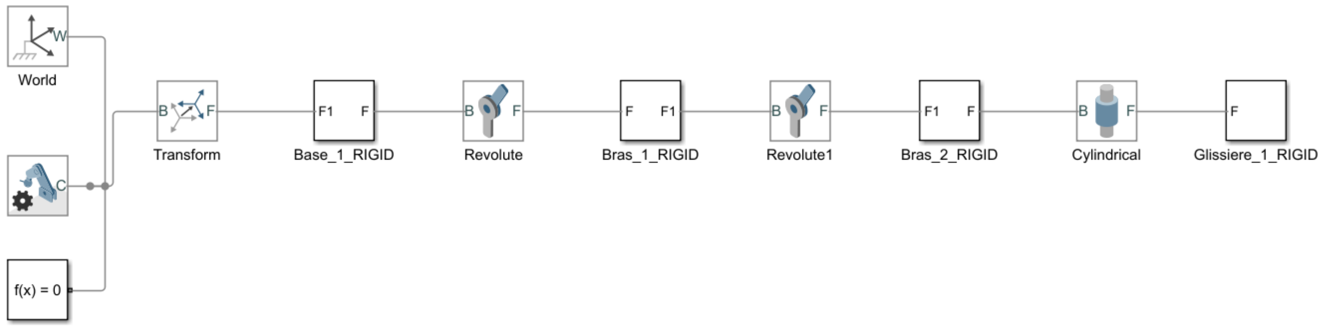
Le but maintenant est de pouvoir dimensionner les actionneurs d'un robot scara qui sera modélisé sur Matlab Simscape. Voici à quoi ressemble le robot :





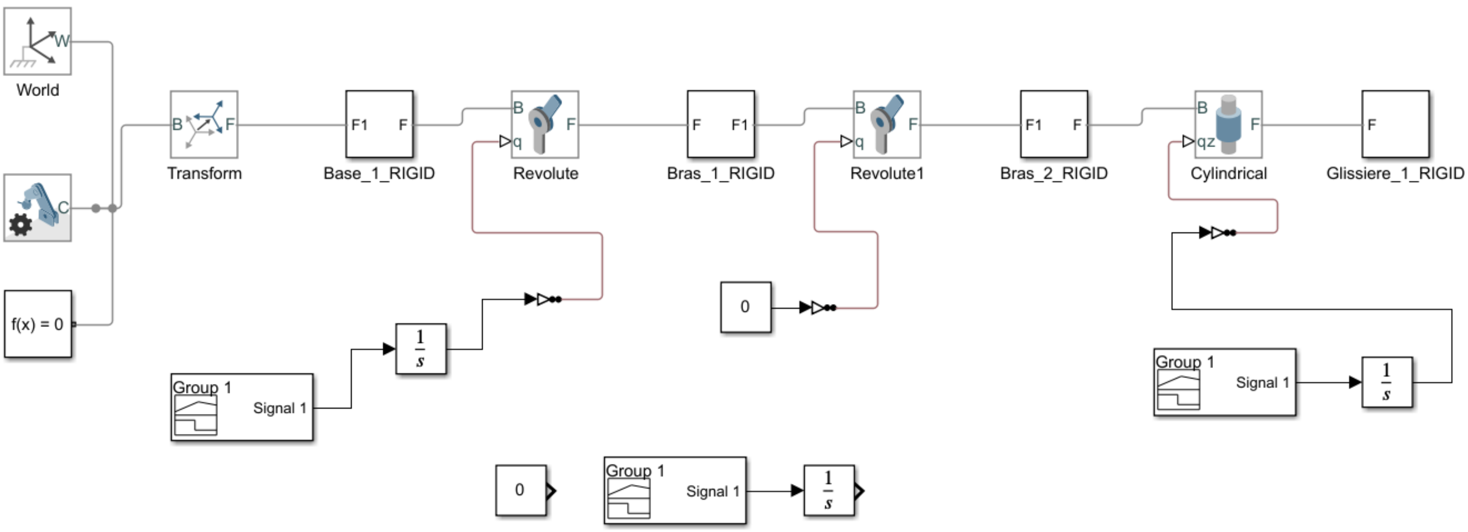
Question 11 :

Nous devons maintenant importer le modèle suivant SolidWorks dans Simscape. Pour cela, on installe le plug-in de Simscape (s'il ne sont pas installer) et on exporte le modèle SolidWorks grâce à la barre « Outils » en vérifiant que Simscape est bien activé. À la suite de cela, on utilise la commande « smimport(Robot) » pour obtenir ainsi le modèle Simscape associé aux Robot.



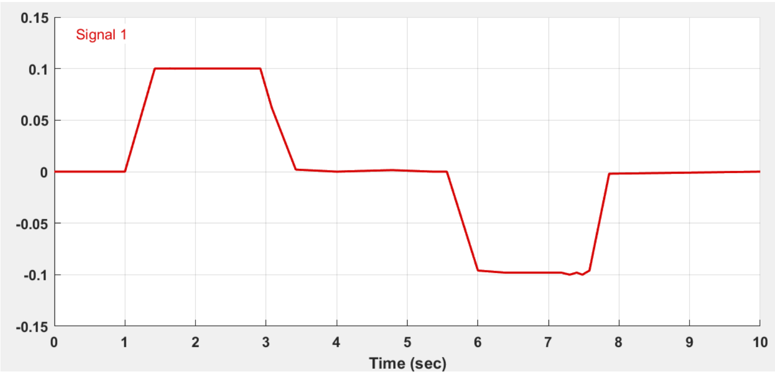
Question 13 :

Pour générer le mouvement associé aux phases 1,2,3 du système nous allons d'abord implémenter des créateurs de signal suivis des intégrateurs. En effet, les blocs « revolute » et « cylindrical » reçoive une position. Par ailleurs, il est important de mettre un couple en mode « automatique » pour que le système puisse marcher.

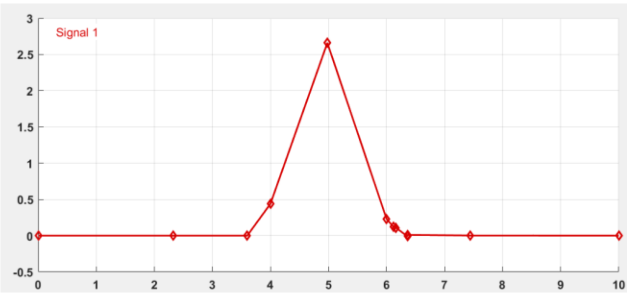


Nous devons maintenant respecter les mouvements du robot qui nous est imposé. Nous choisissons alors des signaux tels que l'air sous la courbe, la distance, soit égale aux déplacements souhaités. Par ailleurs, il faut que la pivot du milieu soit à 0°. C'est pour cela que l'on utilise la fonction nulle en entrée du « Revolute 1 »

Pour le premier déplacement, on a une élévation de 2 s du troisième bras d'une hauteur de 20 cm, puis la même chose dans le sens inverse après deux secondes d'attente. On choisit alors deux triangles disposés dans la partie « cylindrical » sous la forme suivante :

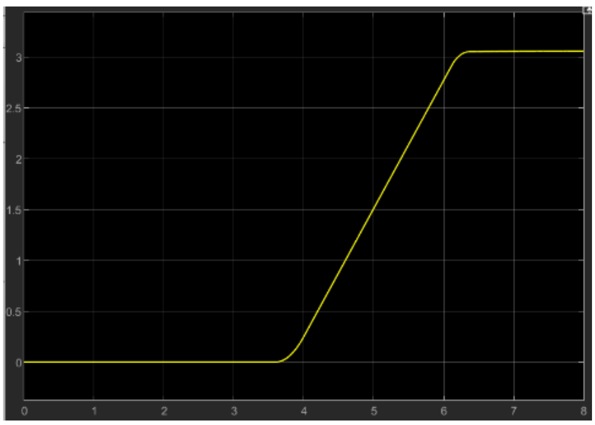


De la même manière, sur la première liaison pivot, nous devons faire un déplacement de 2 s de la position (x = 0 m, y = 0, 5 m, z = −0, 2 m) à la position (x = 0 m, y = −0.5 m, z = −0.2 m). Ainsi, nous devons faire une rotation de façon à passer du côté négatif. On ne touche donc pas à la deuxième liaison pivot. On propose donc le signal de la forme suivante, qui permet bien la rotation de 180°.

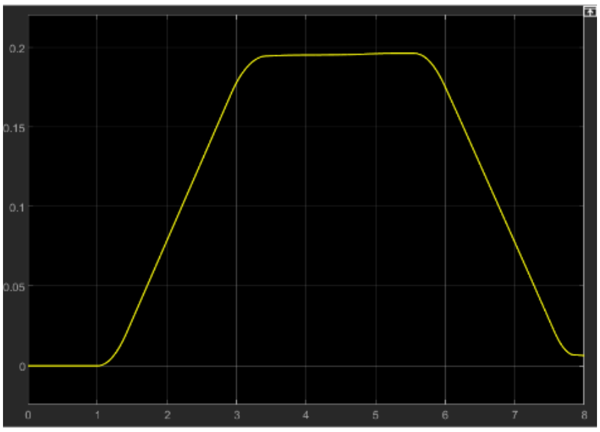


On peut vérifier facilement vérifier les trajectoires des différentes pièces et que ceci respecte bien le cahier des charges sur les déplacements. On observe ci-dessous l'angle de rotation en radians et à gauche les déplacements du solide 3 selon z en mètres.

<u>**Angle de 1 (rad) en fonction du temps(s)**</u>

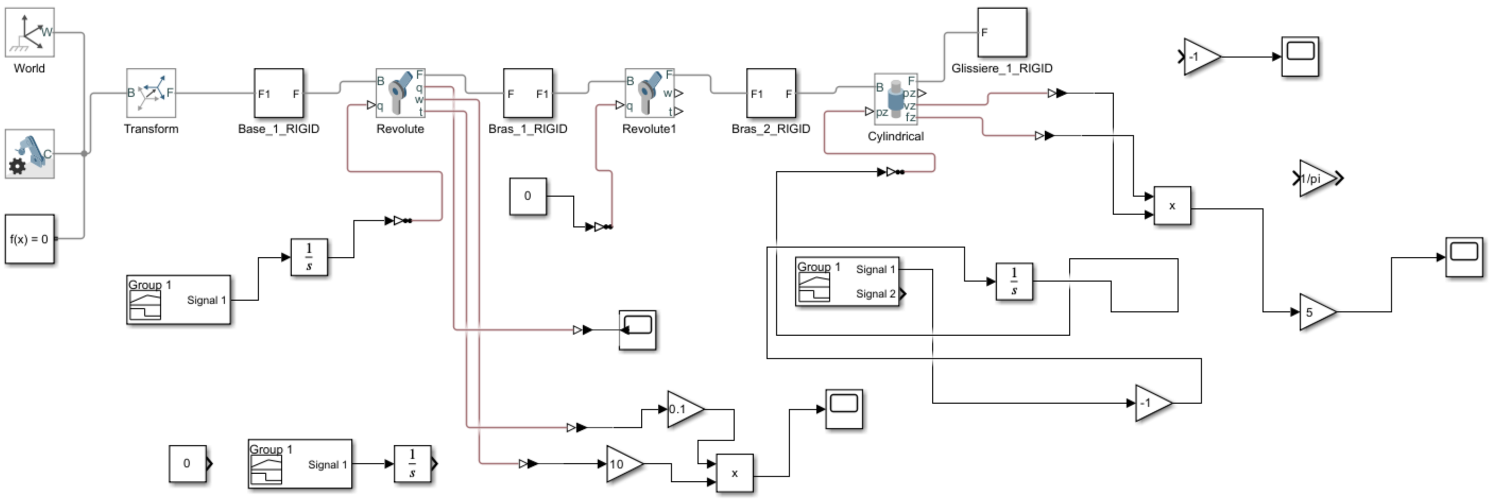


<u>**Position 3 (m) en fonction du temps(s)**</u>



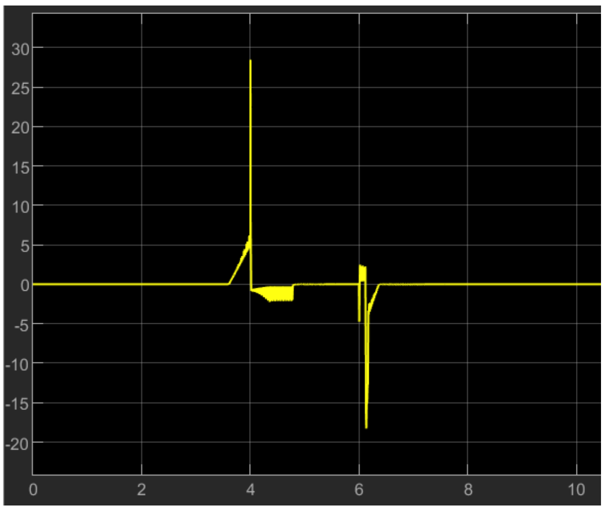
Question 14 :

Pour dimensionner les moteurs pas à pas nécessaire, nous pouvons mesurer les puissances respectives de différents moteurs à l'aide d'un scope dans lequel on enverra le produit du signal du couple multipliée par la vitesse de rotation, pour observer la puissance nominale nécessaire et donc déduire le moteur à choisir. On obtient :



En utilisant le scope pour en relever la puissance, on obtient alors, une puissance nécessaire de 29 Watt pour le premier moteur.

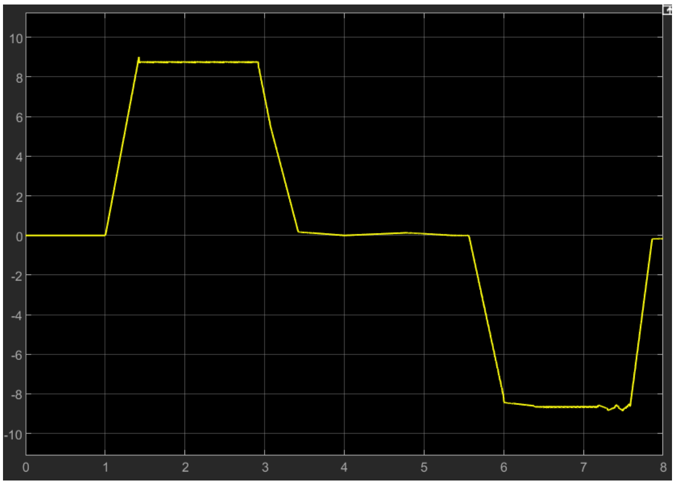
<u>Puissance (m) en fonction du temps(s)</u>



De la même manière, les autres moteurs, pas à pas, seront dimensionnés à 6 Watt. Le moteur adapté pourrait donc être le moteur pas à pas HT24-108. De la même manière, ce moteur semble adapté à la puissance des autres moteurs nécessaire du système.

On peut alors relever le schéma de la puissance du moteur associé à la translation cylindrique. On en déduit que la puissance nominale du troisième moteur est de 10 W. On en déduit que la puissance nominale du troisième moteur est de 10 W. En effet, on peut observer ceci dans la puissance nominale requise pour la partie glissière.

Puissance (W) en fonction du temps(s)



Le deuxième moteur possède la même structure que le premier il a donc besoin d'une puissance de 29 W. On en déduit alors que les moteurs à choisir sont :

- Nema 42 5042-022-2 (30W)

- Nema 42 5042-022-2 (30W)

- Nema 34 HT34-485 – 2 (10W)

Conclusion

Pour conclure, on peut dire que ce TP m'a permis de découvrir l'univers du logiciel Matlab-Simulink, il a été très intéressant de modéliser des systèmes mécaniques de difficulté croissante. Effectivement, nous avons modélisé une structure de parallélogramme avec un assemblage de Solides, un pendule simple, un système amortisseur-ressort et un robot Scara pour son dimensionnement. Nous avons comparé ainsi, les valeurs des solutions exactes aux équations différentielles et comparer aux valeurs du logiciel, qui sont quasi les mêmes.

Finalement, la manipulation des différents objets et blocs du système a été très importante pour me permettre de plonger dans l'ingénierie des systèmes complexes et leurs modélisations à travers des logiciels.