

## Première Partie : découverte du *Novint Falcon*

1. Déterminez les dimensions de l'espace de travail.
2. Réalisez un ressort (+amortisseur) virtuel. Tracez les courbes position poignée, force générée et position vs. force sur une des 3 axes. Comment choisir une bonne valeur pour l'amortissement ?
3. Déterminez la masse et l'amortissement apparents du Falcon. Faites une identification dynamique par oscillation libre, en utilisant un raideur élevé, entre 2000 et 5000 N/m.

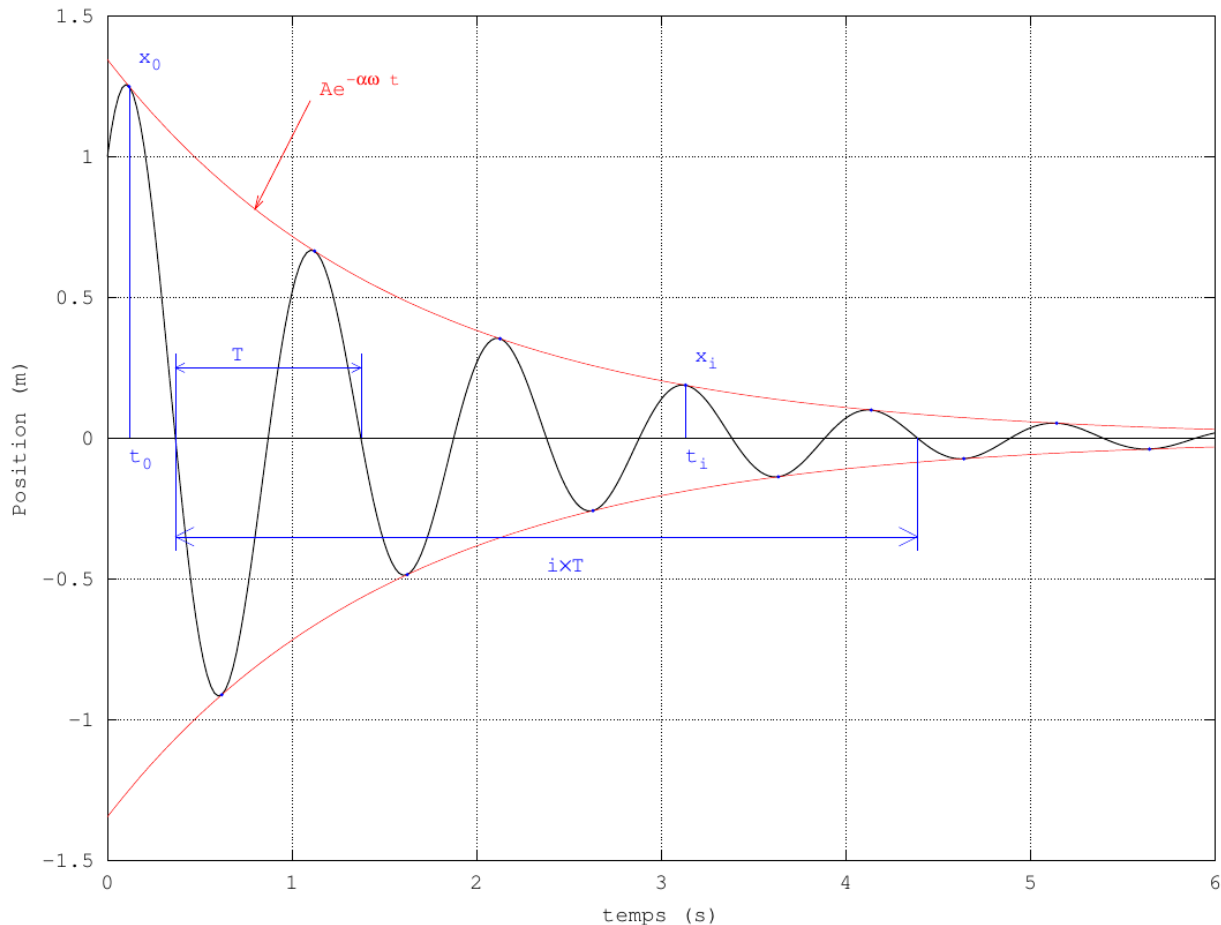


Figure 1: Rappel de la réponse libre d'un système masse-ressort+amortisseur.

$$m\ddot{x} + c\dot{x} + kx = 0 \Rightarrow \ddot{x} + 2\alpha\omega\dot{x} + \omega^2x = 0$$

$$T = 1/F ; \quad 2\pi F = \omega\sqrt{1-\alpha^2}, \quad \omega^2 = k/m, \quad \alpha = \frac{\ln \frac{x_0}{x_i}}{2i\pi}$$

4. Implémenter un mur virtuel pour le plan  $y < -0.02$  m. Expliquez et justifiez votre choix de la fonction  $F(y)$ , courbes à l'appui.
5. Implémenter une sphère puis un cube virtuels centrés sur l'espace de travail de Falcon. Justifier vos choix de  $F(x,y,z)$ .
6. Implémenter une glissière à clic sur l'axe  $y$ .

## Seconde Partie : Couplage à un simulateur physique


Ouvrez le fichier `falcon.blend` (Blender).

Il représente une scène virtuelle avec 2 objet immuables 'obstacles' (une sphère et un cube), et une sphère grise contrôlée par l'utilisateur, appelé « Handle ». Il contient aussi une sphère bleue appelée « Proxy », qui est géré comme un corps rigide par le simulateur.

Il contient 3 fichiers, 'init', 'couplageDirecte' et 'exit'.

- 'init' est exécuté au démarrage une seule fois. Il démarra la communication avec le Falcon et crée les listes qui contiendront des valeurs qu'on voudrait garder entre les pas de simulations, comme attributs global associés au *game*. (voir *game.posx* par exemple)
- 'exit' est exécuté une seule fois à la sortie de la simulation, déclenchée par 'ESC'. Il ferme la communication et effectue la sauvegarde des listes persistantes *game.XXX*.
- 'couplageDirecte' est exécuté du début à la fin, à chaque pas de simulation. Toutes les variables sont réinitialisées à chaque pas. Pour stocker des valeurs entre des exécutions consécutives, il faut utiliser des listes *game.XXX* créé dans 'init'. Ce script fait la lecture de la position de la poignée, et l'utilise pour déplacer le « Handle », puis renvoie des efforts au Falcon.

1. En utilisant les fonctions développées précédemment, recréer les retours haptiques au contact des obstacles.

2. Notez vos observation quand vous modifiez la fréquence d'exécution de la simulation (onglet World  ). Commentez sur les sensations subjectives pendant la manipulation, notamment sans et avec contact.

3. Remplacez le couplage directe par un couplage indirect, en insérant un ressort+amortisseur entre le proxy et le handle, pour prendre avantage de la détection de collision intégré au simulateur entre les objets rigides. Notez l'influence de la masse du proxy, et de la fréquence du simulateur. Ajouter éventuellement d'autres obstacles. Commentez la comparaison des 2 couplages, et les sensations haptiques.