Devoir 3

Date de remise : 18 novembre

Jeu de collision

Un industriel vous demande de concevoir un moteur physique pour son jeu vidéo qui devrait fonctionner de la façon suivante :

- Au temps $t = 0$, un bloc cubique de masse $m_{\text{bloc}} = 1.2$ kg et ayant des côtés mesurant $A_{\text{bloc}} = 8$ cm est éjecté par un canon. La position initiale du bloc lorsqu'il est lancé est $\vec{r}_{\text{bloc}}(0) = (3, 3, 1)^T$ m, ses arêtes étant parallèles aux axes. La vitesse de son centre de masse est initialement $\vec{v}_{\text{bloc}}(0)$ et sa vitesse angulaire initiale est $\vec{\omega}_{\text{bloc}}(0)$.

- Le concurrent lance la balle à son tour (temps t_l) et tente de toucher le bloc déjà en vol de façon à le faire dévier. Cette balle a un rayon $R_{\text{balle}} = 2$ cm et une masse $m_{\text{balle}} = 50$ g. La position initiale de la balle est $\vec{r}_{\text{balle}}(t_l) = (0, 0, 2)^T$ m et elle est lancée avec une vitesse initiale $\vec{v}_{\text{balle}}(t_l)$. La vitesse angulaire de rotation de la balle est initialement nulle.

La seule force qui agit sur la balle et le bloc cubique dont vous devrez tenir compte est la force gravitationnelle ($g = 9.8$ m/s^2). Le coefficient de restitution lors d'une collision entre la balle et le bloc cubique est $\epsilon_c = 0.8$. Vous pouvez négliger le frottement entre les deux objets et entre ces objets et le sol.

Le but de ce devoir est de programmer une fonction Matlab qui permet de :

- simuler le mouvement de la balle et du bloc cubique dans l'espace en fonction du temps ;
- détecter si des collisions entre les deux objets existent et dans cette éventualité, déterminer les vitesses de la balle et du bloc cubique (centre de masse et vitesse angulaire) après la collision et terminer la simulation;
- si la balle ou le bloc cubique touche le sol avant qu'une collision se produise, terminer la simulation et retourner les vitesses de la balle et du bloc cubique (centre de masse et vitesse angulaire) au moment où ils touchent le sol (tout juste avant la collision avec le sol).

Ce client vous demande donc de programmer un script Matlab pour effectuer ces simulations qui pourra être appelé directement par son module de contrôle. Le format d'appel qu'il vous impose est le suivant

```
[Resultat blocf ballef Post]=Devoir3(vbloci,avbloci,tl,vballei)
```

où les données d'entrée sont

- `vbloci`

  Un vecteur de 3 éléments contenant la vitesse linéaire initiale du centre de masse du bloc.

  - `vbloci(1:3)`=$\vec{v}_{\text{bloc}}(0)$

- `avbloci`

  Un vecteur de 3 éléments contenant la vitesse angulaire initiale du bloc autour de son centre de masse.

  - `avbloci(1:3)`=$\vec{\omega}_{\text{bloc}}(0)$

- `vballei`

  Un vecteur de 3 éléments contenant la vitesse linéaire initiale du centre de masse de la balle au temps t_l.

  - `vballei(1:3)`=$\vec{v}_{\text{balle}}(t_l)$

- `tl`

  Le temps t_l où le candidat a décidé de tirer la balle.

Les résultats produits par cette fonction Matlab sont

- `Resultat`

  Ici, `Resultat`=0 indique que la balle a touché le bloc, `Resultat`=-1 que la balle a touché le sol en premier et `Resultat`=1 que le bloc a touché le sol en premier.

- `blocf`

  Une matrice de 6×2 éléments contenant la vitesse linéaire et la vitesse angulaire du bloc tout juste avant la collision (temps t_f^-) et tout juste après la collision (temps t_f^+).

  - `blocf(1:3,1)`=$\vec{v}_{\text{bloc}}(t_f^-)$
  - `blocf(4:6,1)`=$\vec{\omega}_{\text{bloc}}(t_f^-)$
  - `blocf(1:3,2)`=$\vec{v}_{\text{bloc}}(t_f^+)$
  - `blocf(4:6,2)`=$\vec{\omega}_{\text{bloc}}(t_f^+)$

  Notez que dans le cas où `Resultat`$= \pm 1$,

  - `blocf(:,2)=blocf(:,1)`
  - `blocf(:,2)=blocf(:,1)`

- `ballef`

  Une matrice 6×2 contenant la vitesse linéaire et la vitesse angulaire de la balle tout juste avant la collision (temps t_f^-) et tout juste après la collision (temps t_f^+).

  - `ballef`(1:3,1)=$\vec{v}_{\text{balle}}(t_f^-)$
  - `ballef`(4:6,1)=$\vec{\omega}_{\text{balle}}(t_f^-)$
  - `ballef`(1:3,2)=$\vec{v}_{\text{balle}}(t_f^+)$
  - `ballef`(4:6,2)=$\vec{\omega}_{\text{balle}}(t_f^+)$

  Encore une fois, dans le cas où `Resultat`$= \pm 1$, vous utiliserez

  - `ballef`(:,2)=`ballef`(:,1)
  - `ballef`(:,2)=`ballef`(:,1)

- `Post`

  C'est une matrice contenant les points dans le temps et les positions du bloc et de la balle simulés avec

  - `Post`$(1, i) = t_i$
  - `Post`$(2:4, i) = \vec{r}_{\text{bloc}}(t_i)$
  - `Post`$(5:7, i) = \vec{r}_{\text{balle}}(t_i)$.

Les vitesses initiales de la balle et le temps du lancer pour les six tirs à simuler et à analyser ont donnés au tableau 1. La précision requise pour les résultats des simulations correspond des erreurs maximales sur les positions de la collision en x, y et z de ± 1 mm (positions du entre de masse de la balle et de la boîte).

Tableau 1: Vitesses initiales et temps du lancer de la balle pour les six tirs à simuler.

Tir	t_l (s)	$\vec{v}_{\text{bloc}}(0)$ (m/s)	$\vec{\omega}_{\text{bloc}}(0)$ (rad/s)	$\vec{v}_{\text{balle}}(0)$ (m/s)
1	0.545454	$(-2,-3,5)^T$	$(0,0,0)^T$	$(5,2,0.642424)^T$
2	0.545454	$(-2,-3,5)^T$	$(0,0,15)^T$	$(5,2,0.642424)^T$
3	0.071429	$(0,-6,3)^T$	$(0,0,0)^T$	$(7,0,0.40834)^T$
4	0.071429	$(0,-6,3)^T$	$(0,0,15)^T$	$(7,0,0.40834)^T$
5	0.6	$(-2,-3,5)^T$	$(-5,-5,0)^T$	$(5,2,0.642424)^T$
6	0.1	$(-2,-3,5)^T$	$(0,0,0)^T$	$(5,2,0.1)^T$