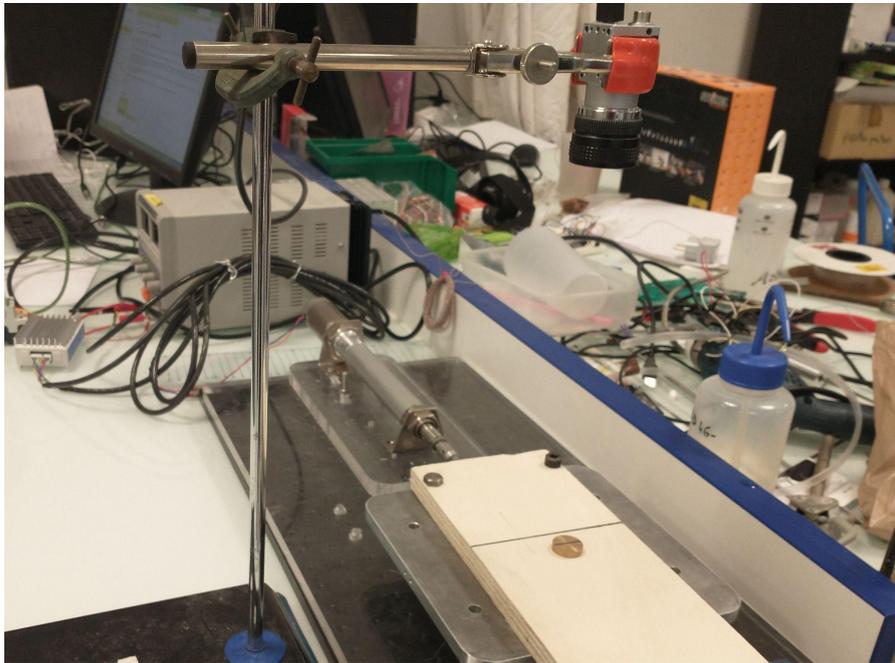


Mise en mouvement d'une pièce par vibrations asymétriques

Timothée Gautry
Louis-Victor Barrois

Méthodes et protocoles

I- Montage expérimental



Matériel pour expérience :

- plaque en PLA (20x40x2 cm)
- pièces en laiton (20x5 mm, 25x3.2, 30x2.2, 35x1.6, 40x1.25,...)
- piston électromagnétique
 - moteur
 - controllino
- caméra (100 ips)

II- Mise en marche du piston

Voilà un exemple de code arduino pour les conditions 100_75 : le piston recule pendant 100 ms et avance pendant 75 ms.

PIN_1 est responsable de la mise en mouvement du piston : HIGH → mouvement, LOW → arrêt

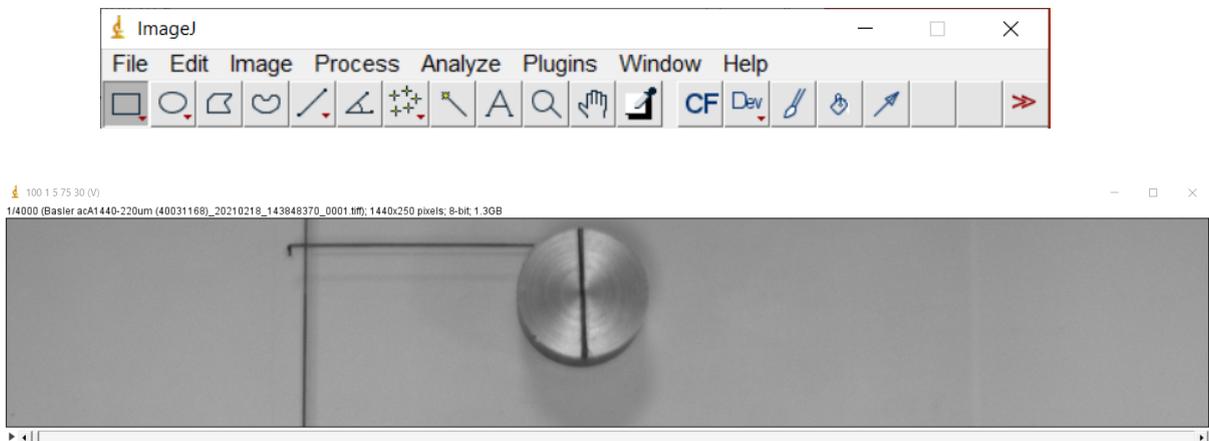
PIN_2 est responsable de la direction du déplacement : HIGH → vers la gauche, LOW → vers la droite

```
void () {  
  
  digitalWrite(PIN_2, LOW);  
  
  while (t<100) {  
    digitalWrite(PIN_1, HIGH);  
    delay(5);  
    digitalWrite(PIN_1, LOW);  
    t = t+1; }  
  
  digitalWrite(PIN_2, HIGH);  
  digitalWrite(PIN_1, HIGH);  
  delay(75)  
  
  digitalWrite(PIN_1, LOW); }
```

III- Expérience et analyse

On met en marche le piston puis on fait une capture vidéo de notre montage pendant une durée de 20s à 100ips.

On étudie le déplacement de la pièce avec imageJ avant de traiter les données récupérées grâce à un code Matlab :



```

1 - clear all
2
3 - Nd=2;
4
5 - T_exp=2000; %temps total de l'expérience en centième de seconde
6 - x=zeros(T_exp,Nd);
7 - t=[0:1999];
8
9 - L= {'100 1 5 75 5 XY.txt','100 1 5 75 5 XY_plaque.txt'} ;
10
11 - for k=(1:Nd)
12
13 -     S = load(L(k));
14
15 -     X = S(:,1) ;
16 -     T = S(:,2) ;
17
18
19 -     for i = 1:T_exp
20 -         ind=find(T==i-1) ; %pixel correspondant au temps i-1
21 -         pos=ceil(length(ind)/2) ; %pixel central
22 -         if ~isempty(ind)
23 -             x(i,k)=X(ind(pos)) ;
24 -         else %correction des sauts de frames
25 -             %ind1=find(T==i-2);
26 -             %ind2=find(T==i);
27 -             %x(i,k)= (X(ind1(end))+X(ind2(1)))/2;
28 -             x(i,k)=x(i-1,k);
29 -         end
30 -     end
31

```

```

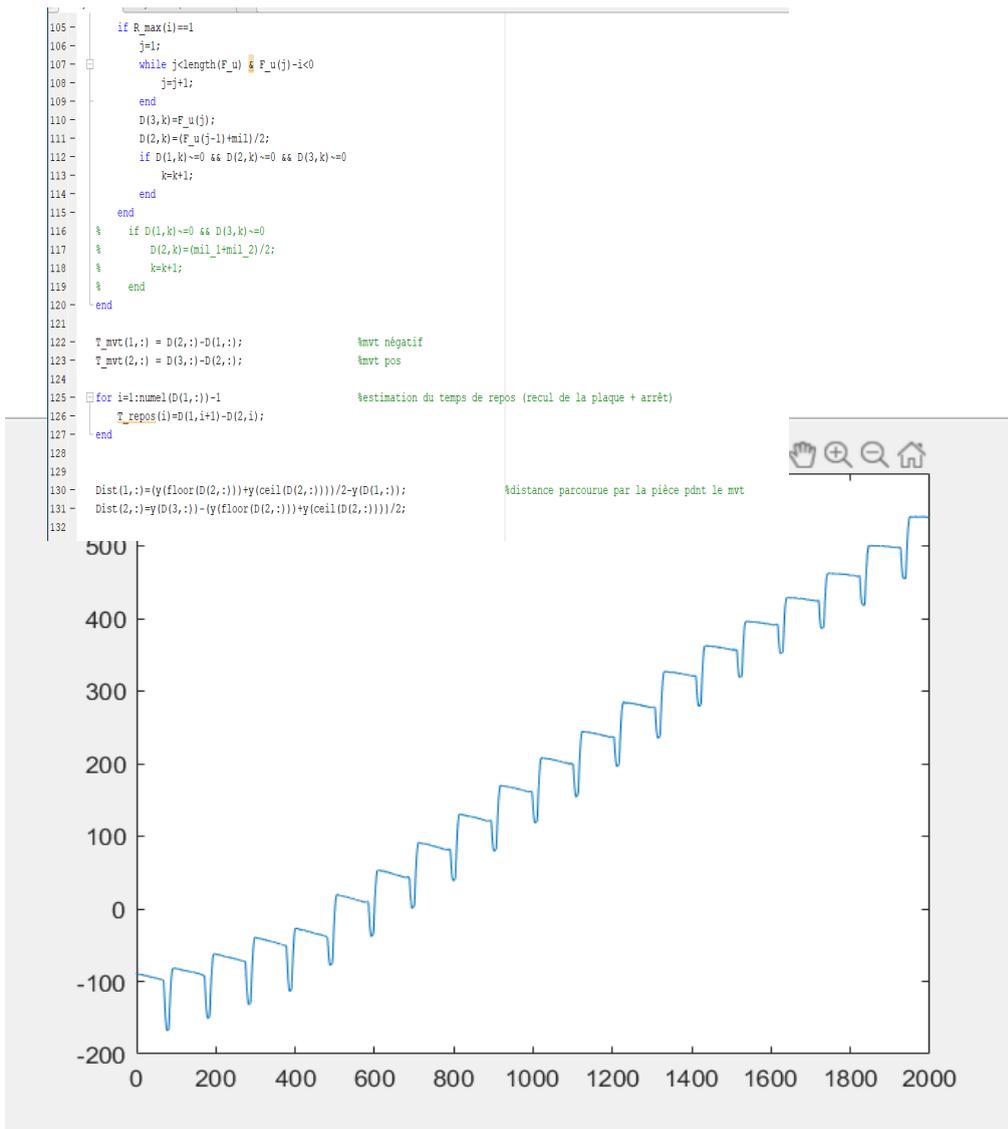
43
44 - figure(k)
45 - plot(t,x(:,k))
46 - title(L(k))
47 - xlabel('Temps en frame')
48 - ylabel('Position en pixel')
49 - end
50
51 - y = x(:,1) - x(:,2);
52 - Y=diff(y)./diff(t');
53 - figure, plot(Y)
54 - figure, plot(t,y)
55
56
57 - R_max=islocalmax(Y); %détection des maximums locaux (en vitesse)
58 - R_min=islocalmin(Y);
59 - k=0;
60 - for i=1:numel(Y) %tri des extremums locaux
61 -     %attention les R sont décalés de 1
62 -     if R_max(i)==1
63 -         if Y(i)<10
64 -             R_max(i)=0;
65 -         else
66 -             k=k+1;
67 -         end
68 -     end
69 -     if R_min(i)==1
70 -         if abs(Y(i))<6
71 -             R_min(i)=0;
72 -         end
73 -     end
74 - end
75

```

```

76 - D=zeros(3,k); %matrice contenant les temps (mise en mouvement, passage vitesse négative positive et arrêt)
77
78 - for i=1:numel(Y)-1 %détection des changements de signes en vitesse (pour repérer les zéros)
79 -     F(i)=0;
80 -     if Y(i+1)*Y(i)<=0
81 -         F(i)=1;
82 -         % if abs(Y(i))<abs(Y(i+1)) %à affiner en précision (i ou i+1)
83 -         F(i)=1;
84 -     else
85 -         F(i+1)=1;
86 -     end
87 - end
88
89 - F_u=find(F==1); %indices i de changement de signe de Y
90
91 - k=1;
92 - for i=1:numel(Y) %remplissage de D
93 -     if R_min(i)==1
94 -         j=1;
95 -         while j<length(F_u) && F_u(j)-i<0
96 -             j=j+1;
97 -         end
98 -         D(1,k)=F_u(j-1);
99 -         mil=F_u(j);
100 -         if D(1,k)~=0 && D(2,k)~=0 && D(3,k)~=0
101 -             k=k+1;
102 -         end
103 -     end
104 - end
105 - if R_max(i)==1

```



Ce code nous permet d'obtenir différentes valeurs d'intérêts (temps de déplacement, distances parcourues ...). Mais également d'afficher les courbes de déplacement de la pièce et de la plaque au cours de nos expériences.

