

ROBOTS AUTONOMES ALEATOIRES

Matériels et Méthodes

Chloé Fougères, Duncan Gilbert, Jérémie Laydevant

Mai 2016

1 Matériels utilisés

- ❑ 4 Robots que nous avons construits *Figure 4* :
 - Support à 3 pieds réalisé par imprimante 3D grâce à l'atelier (*Figure 1*)
 - Composants *Figure 2* du circuit électrique *Figure 3* :
 - Moteur vibreur de téléphone (diamètre 10mm, tension 3V-4.5V, courant 60mA)
 - Résistance 2.2k Ω
 - Transistor NPN BC 33715
 - Photorésistance modèle LDR GL5528
 - 2 piles au lithium SAFT LR6 AA avec pattes à souder (3.6V, 2600mAh)
 - Interrupteur à glissière avec 3 pieds
 - Plaque époxy découpée circulairement à l'atelier puis impression du circuit en laboratoire d'électronique
 - Cache : disque noir en tissu
- ❑ Planche en bois vernie, mise à niveau horizontalement pour avoir une surface plate lors du déplacement des robots
- ❑ Une zone de confinement : 4 barres métalliques verticales d'environ 2 m soutenues par des barres métalliques horizontales et des pans de nappe noire en papier opaque (*Figure 5*)
- ❑ Système solidaire par du ruban adhésive placé en haut du montage (sur une échelle métallique) composé de :
 - Rétroprojecteur BenQ modèle MW529 (luminosité 3300 lumens)
 - Miroir orienté afin d'afficher au sol l'image projetée par le rétroprojecteur
 - Webcam USB 6-LED
- ❑ 2 ordinateurs reliés respectivement à la caméra USB et au rétroprojecteur

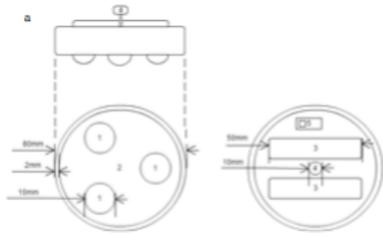


Figure 1 : squelette du robot



Figure 2 : composants du circuit électrique

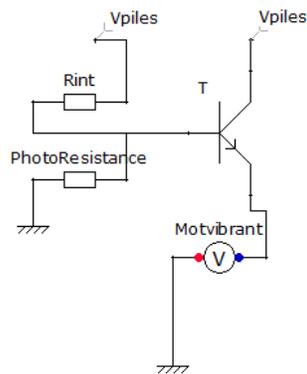


Figure 3 : schéma du circuit

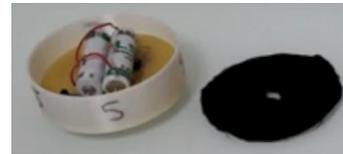


Figure 4 : robot monté



Figure 5 : montage expérimental

2 Méthodes

2.1 Expériences

L'objectif principal est la caractérisation du déplacement d'un robot vibrant selon l'intensité lumineuse reçue par la photorésistance.

- Nous créons sous le logiciel *Matlab* une image uniforme soit un niveau de gris fixe (Niveau de gris $\in [40, 45, 50]$) que nous projetons sur la planche de bois au sol via le système rétroprojecteur+miroir.

Nous plaçons un robot au centre de la planche et allumons le circuit via l'interrupteur. Nous le laissons se déplacer pendant 200 secondes.

- Nous réalisons l'acquisition numérique des positions du robot, au cours d'une expérience, par la caméra USB et un programme *Matlab* (cf Annexe 1).
Le pas de temps entre chaque image traitée est de 0.1 s. Le programme détecte les disques noirs (de diamètre entré en début de programme en mesurant sur la première image prise le disque noir porté par le robot) sur chaque image. Les positions du robot au cours du temps, sont stockées dans une matrice $2 * \tau$ avec τ le temps d'acquisition qui est un paramètre fixé au début du programme (200 s). Cette matrice des positions est finalement enregistrée dans un fichier texte.
- Nous mesurons ainsi la trajectoire de 4 robots successivement pour les 3 niveaux de gris. A niveau de gris et robot fixés, nous répétons 3 fois l'expérience.

2.2 Traitement numérique des données

Les fichiers textes contenant les positions des robots pour les différentes expériences sont traités sous *Matlab*.

- Calcul de $\langle r^2 \rangle_{\text{moy}}$ au cours du temps pour une expérience

Le programme fourni en Annexe 2 permet d'afficher l'évolution de l'écart moyenne à l'origine : $\langle r^2 \rangle (t)$. Après importation du fichier texte contenant les positions, la matrice est subdivisée en sous matrices de taille 10 représentant des expériences échantillons (de 1s) de l'expérience physique. Chaque échantillon est centré à l'origine nulle (en otant le premier terme) puis $\langle r^2 \rangle = \sum_{i=0}^j x_i^2 + y_i^2$ au temps j est implémenté dans une matrice 1*10 dans une boucle sur $0 < j < 10$. Cette matrice stocke temporairement l'évolution de $\langle r^2 \rangle$ pour l'expérience échantillon. Enfin $\langle r^2 \rangle_{\text{moy}}$ au cours de 1 seconde est calculé pour les 20 sous-expériences (par moyenne arithmétique des $\langle r^2 \rangle$ de chaque échantillon). Le graphe de $\langle r^2 \rangle_{\text{moy}}$ en fonction du temps (pour un robot, un niveau de gris, une expérience réelle de 200s) est affiché en sortie du programme

- Modélisation

Pour un robot et un niveau de gris, la courbe moyenne des $\langle r^2 \rangle_{\text{moy}}$ en fonction du temps sur les 3 expériences que nous avons réalisées, est calculée. Puis par l'outil *Curve fitting* de *Matlab* nous enregistrons la pente de régression linéaire de cette courbe, pour les trois niveaux de gris associés à un robot. En effet nous partons de la formule d'Einstein du déplacement d'une particule au cours du temps t dans un fluide :

$$\langle x^2 \rangle = t * \frac{2 * R * T}{N * w} \text{ où } w = \text{résistance visqueuse du fluide}$$

Nous supposons ainsi que notre robot a un mouvement brownien lors de son déplacement dans le champ lumineux d'intensité fixe.

Les 4 courbes de la pente en fonction du niveau de gris associées aux 4 robots sont tracées. Les étudier permet de valider ou non le modèle linéaire selon l'intensité de vibration des robots ainsi que de caractériser chaque robot et son coefficient de diffusion.

2.3 Simulations numériques

Par le langage *Java*, nous simulons la dynamique d'un grand nombre de particules soumises à un environnement d'intensités variables de vibration, le programme est fourni en Annexe 3.

Nous avons construit trois types de champ de vibration : gradient, fonction triangulaire et 4 puits de potentiel nul. A chaque instant, les déplacements horizontal et vertical des particules sont calculés comme le produit de la direction aléatoire $[-1, 0, 1]$ et de la longueur du saut selon la valeur de l'intensité de vibration à la position de la particule. La fenêtre *Java* affiche alors la dynamique des particules : pour tous les champs, au bout d'un certain temps les particules se piègent dans les zones de vibration nulle.

3 Perspectives

La poursuite du projet mène à effectuer plus d'expériences pour un champ uniforme donné, diminuant ainsi le bruit sur les données (pertes dans le circuit analogique, imprecisions sur la détection avec la caméra USB...). Il est aussi intéressant d'obtenir des données pour d'autres niveaux gris dans la place[40; 50] afin de confirmer ou infirmer le modèle linéaire du coefficient de diffusion du robot.