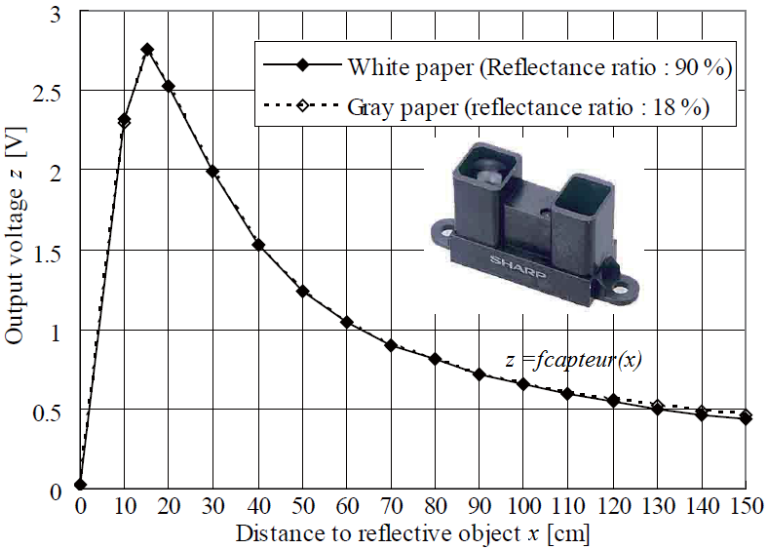
Introduction à la robotique mobile **GLO-4001/7021**
Philippe Giguère A14

Laboratoire Nº 2
Capteurs infrarouges Sharp

Matériel : règle de 1 mètre en aluminium, règle de 30 centimètres, boite (de Kinect), ruban adhésif, feuille blanche et carton noir.

Partie 1 Courbe de calibration d'un capteur

Dans cette première partie du laboratoire, vous allez voir comment on peut calibrer un capteur, particulièrement s'il est non-linéaire. L'idée de base consiste à placer une cible à différents endroits (x en cm), et à mesurer la sortie du capteur (z en Volt). Ainsi, vous allez échantillonner manuellement la fonction $z=f_{capteur}(x)$. Bien que le manufacturier vous fournisse une telle fonction (voir la figure ci-contre), les aléas de la fabrication font en sorte que chaque capteur est différent. Le capteur à utiliser est celui à longue portée, illustrée à la figure ci-contre.



Pour vous connecter sur la plateforme robotisée et sélectionner les bons capteurs sur le convertisseur numérique analogue, utilisez le code suivant dans matlab :

```
clc; clearvars;
addpath('RoombaControl')
addpath('ros4mat/wrapper')
IP = 'addresse ip de votre robot, genre 10.240.198.55';
ros4mat('connect', IP, 1);
sensorsFlag = 1; % (1)IR_20_150 (2)IR_10_80 (16)GyroZ (32)Gyro4Z (64)GyroX (128)Gyro4X
acquisitionFrequency = 140; % Hz (max.2000-10000HZ, IR=25Hz, Gyro=140Hz)
pollingFrequency = acquisitionFrequency; % Hz
bufferDuration = 1; % seconds
bufferSize = acquisitionFrequency*bufferDuration;
ros4mat('subscribe','adc', acquisitionFrequency, sensorsFlag, bufferSize,
pollingFrequency)
```

Détail très important : les mesures que vous faites vont être bruitées, ce qui signifie qu'en réalité c'est plutôt $z=N(f_{capteur}(x),\sigma_z^2)$ que vous capturez avec le convertisseur analogue-numérique, où $N(\mu,\sigma^2)$ est une distribution normale centrée sur μ avec variance σ^2. Pour réduire l'erreur due au bruit, les mesures seront moyennées. Utilisez donc les deux lignes de code suivantes pour faire la prise de données à chaque position x :

```
[data, timestamps] = ros4mat('adc');
meanValue = mean(data(1,:))
```

À la fin de vos expériences, n'oubliez pas de vous désinscrire du capteur avec la commande
`ros4mat('unsubscribe', 'adc')`

La page suivante décrit les manipulations physiques à faire.
Commencez par placer le capteur de façon verticale sur la tourelle en plexiglass du robot, à une distance minimum de 10 cm de la table, afin de limiter les mauvaises réflexions qui pourraient fausser les valeurs. Placez l'extrémité 0 cm de la règle métallique vis-à-vis le capteur infrarouge. Au besoin, utilisez du ruban adhésif pour immobiliser la règle sur la table (ça évitera bien des frustrations). Utilisez une boîte en carton comme cible (genre boîte vide d'une Kinect recouverte d'une feuille blanche) que vous placez aux positions telles que dans la table ici-bas. Notez la valeur moyenne de voltage *z* à chaque distance *x*. Après avoir pris toutes les données, tracez la courbe du capteur dans matlab, avec vos données x, z. Comparez-là visuellement avec la figure du manufacturier. Prenez le temps d'observer la non-linéarité de la réponse du capteur, ainsi que le fait qu'elle n'est pas une fonction bijective.

Distance *x* (cm)	Mesure *z* (V)
5	
7.5	
10	
12.5	
15	
17.5	
20	
25	
30	
35	
40	
45	
50	
60	
70	
80	
90	
100	
110	
120	
130	

2 Testez la courbe de calibration

Maintenant que vous avez une table de calibration, vous pouvez faire un télémètre infrarouge bon marché! Pour cet exercice, vous devrez convertir une mesure de voltage du capteur (en Volt) en une distance (en cm). Il s'agit ici d'inverser la fonction de capteur pour avoir $x=f^{-1}_{capteur}(z)$. Or, comme la fonction $f^{-1}_{capteur}(x)$ n'est pas bijective, l'inverse n'existe pas. Il vous faut donc scinder en deux votre table de calibration, de sorte que la première table ne contient que les données de voltage croissant avec la distance, et la deuxième table les données de voltage décroissant avec la distance. Par exemple, si vous avez les données suivantes :

x	5	10	15	20	25	30	40	50	60	80	100	120	140	150
z	1.5	2.2	2.5	3	2.7	2.4	2.2	1.9	1.7	1.6	1.5	1.4	1.3	1.25

La première table de calibration contiendra les valeurs suivantes

x	5	10	15	20
z	1.5	2.2	2.5	3

Et la deuxième table de calibration sera

x	20	25	30	40	50	60	80	100	120	140	150
z	3	2.7	2.4	2.2	1.9	1.7	1.6	1.5	1.4	1.3	1.25

Le code correspondant pour dans matlab (pour les tables de calibration en exemple ci-haut) serait donc :

```
x1 = [5   10   15 20];
z1 = [1.5 2.2 2.5 3];
x2 = [20 25  30  40  50  60  80 100 120 140 150];
z2 = [3 2.7 2.4 2.2 1.9 1.7 1.6 1.5 1.4 1.3 1.25];
```

Ainsi, vous avez maintenant deux fonctions bijectives. Pour pouvoir convertir une valeur de z quelconque en x (i.e. inverser la fonction), il suffit de faire un table-lookup et d'interpoler. La fonction `interp1` dans matlab permet de faire la chose facilement. Lorsque vous faites une mesure de voltage `zmesure`, vous devrez donc faire :

```
xProche = interp1(z1,x1,zmesure);
xLoin = interp1(z2,x2,zmesure);
```

Affichez les résultats à l'écran de la conversion voltage en deux distances. Idéalement, faites une boucle qui

- prend une mesure moyennée (voir code de la question 1);
- fait les deux conversions avec `interp1` pour trouver `xProche` et `xLoin`;
- les affiche: `display(sprintf('Proche=%.2f cm  Loin=%.2f cm',xProche,xLoin));`

Vous pourrez ainsi voir votre télémètre infrarouge en action en temps réel pendant que vous déplacez la cible! Regardez voir si la mesure de distance à l'écran correspond bien à la position de la boite indiquée par la règle. Vous devriez remarquer que l'erreur est plus grande lorsque la cible est loin. Vous devriez remarquer que les valeurs fluctuent beaucoup plus quand la cible est loin (120 cm) que lorsque la cible est proche (30 cm).

3 Réponse transversale du capteur

Note :

1. la taille du faisceau infrarouge augmente en fonction de la distance par rapport au capteur;
2. l'intensité lumineuse diminue au fur et à mesure qu'on s'éloigne du centre du rayon.

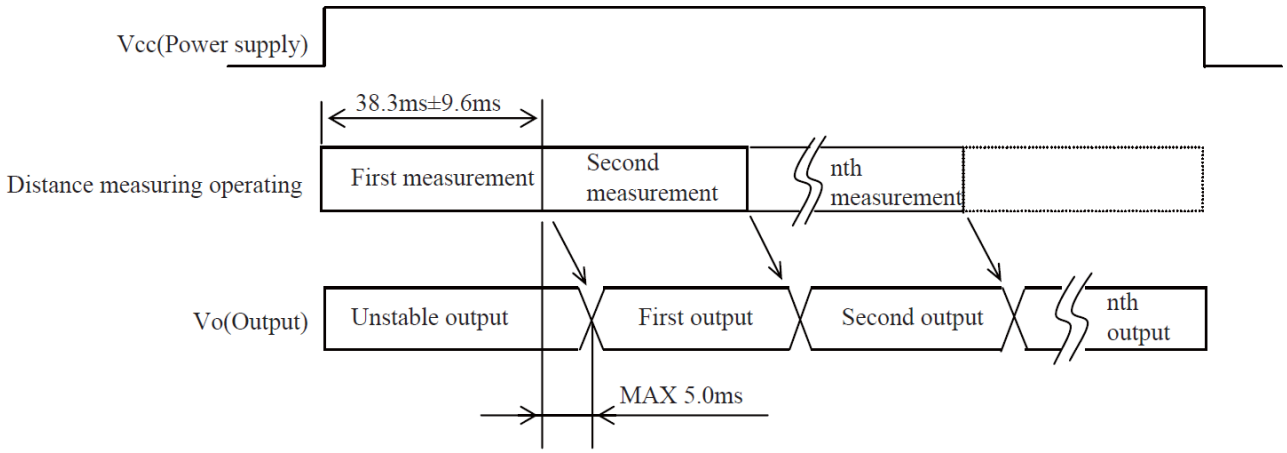
Pour observer la réponse transversale du capteur, il faut déplacer une petite boîte rectangulaire de 8-10 cm de largeur à une distance de 70 cm du capteur, perpendiculairement à l'axe du faisceau. Il faut qu'à la position de départ de la boîte, le *faisceau ne soit pas en contact* avec la boîte (voltage très bas). Identifiez cette position de départ en regardant le voltage lorsque vous déplacez la boîte de façon latérale : quand il commence à monter, c'est que la boîte pénètre dans le faisceau. Déplacez cette boîte perpendiculairement à l'axe du faisceau par incréments de 0.5 cm. Assurez-vous que la règle perpendiculaire soit bien fixée à la table avec du scotch tape pour ne pas qu'elle se déplace pendant la saisie des données. Que pouvez-vous conclure de l'apparence de la boîte carrée, telle que perçue par ce capteur?

Position (cm)	Voltage (V)
0	
0.5	
1	
1.5	
2	
2.5	
3	
3.5	
4	
4.5	
5	
5.5	
6	
6.5	
7	
7.5	
8	
8.5	
9	
9.5	
10	
10.5	
11	
11.5	
12	
12.5	
13	

4 Échantillonnage du capteur

Le capteur lui-même ne mesure pas la distance en continue. La documentation officielle du fabriquant montre le timing du signal produit par le capteur infrarouge. En particulier, le fabriquant indique que le voltage devrait être stable pour 38.3 ms, +- 9 ms.

SHARP **Fig. 1 Timing chart** **GP2Y0A02YK0F**



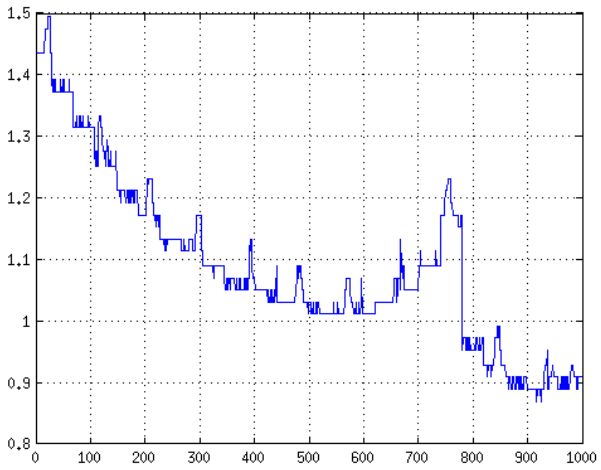
Changez la fréquence de capture dans ros4mat à 1000 Hz. Si vous n'avez pas unsubscribé du capteur, faites d'abord

```
ros4mat('unsubscribe', 'adc')
```

Pour modifier la fréquence de capture, faites les commandes suivantes :

```
sensorsFlag = 1; % (1)IR_20_150 (2)IR_10_80 (16)GyroX (32)GyroZ (64)Gyro4X (128)Gyro4Z
acquisitionFrequency = 1000; % Hz (max.2000-10000HZ, IR=25Hz, Gyro=140Hz)
pollingFrequency = acquisitionFrequency; % Hz
bufferDuration = 1; % seconds
bufferSize = acquisitionFrequency*bufferDuration;
ros4mat('subscribe','adc', acquisitionFrequency, sensorsFlag, bufferSize,
pollingFrequency)
```

En déplaçant **brusquement** la cible et en capturant le signal pendant ce déplacement selon l'axe du capteur (par exemple de x=40 cm à x=60 cm), vous devriez voir le voltage changer par paliers stables (timing similaire au graphe ci-haut). Vous devriez observer, par exemple, le résultat ci-contre (Au besoin, faire un zoom avec l'utilitaire de matlab):



Est-ce que la largeur des paliers (en *ms*) est comparable à la documentation du fabriquant? Quel impact ce phénomène aura, si le capteur balaye très rapidement l'environnement?

5 Dimension de la cible

Mettez la fréquence d'échantillonnage à 20 Hz en modifiant les commandes de la question précédentes. Placez un objet de faible largeur (1-2 cm) à 30, 40, 60 et 90 cm du capteur. En déplaçant la cible transversalement, trouvez la valeur maximale du voltage (qui correspondra à la cible placée au centre du faisceau). Comparez les valeurs de voltage de ce pic à celles préalablement obtenues avec la grosse boîte de carton pour les mêmes distances. D'après vous, pourquoi les valeurs sont-elles différentes? En particulier, y aura-t-il une erreur d'estimation de la distance pour ces petites cibles?

Position (cm)	Voltage (V)
30	
40	
60	
90	