



ÉCOLE  
POLYTECHNIQUE  
DE BRUXELLES



UNIVERSITÉ LIBRE DE BRUXELLES

ELEC-H311 - PROJET INTÉGRÉ

---

# Rapport projet multidisciplinaire

---

*Auteurs :*

DENGUIR Anass  
NOEL Luca  
PHAN Trong Quy

*Professeurs :*

MILOJEVIC Dragomir  
NONCLERCQ Antoine  
OSEE Michel  
ROBERT Frédéric

22 mai 2017

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Description du projet . . . . .	4
1.2	Partie communication . . . . .	4
1.3	Partie propulsion . . . . .	6
<b>2</b>	<b>Communication</b>	<b>7</b>
2.1	Description et cahier des charges . . . . .	7
2.2	Prototype pour la communication audio . . . . .	8
2.3	Traitement analogique . . . . .	8
2.3.1	Microphone . . . . .	8
2.3.2	Amplification . . . . .	9
2.3.3	Filtrage . . . . .	10
2.3.4	Choix des composants des étages à amplification . . . . .	11
2.3.5	Choix des composants du filtre analogique . . . . .	14
2.3.6	Simulation . . . . .	16
2.3.7	Validation de la chaîne . . . . .	18
2.4	Traitement numérique . . . . .	19
2.4.1	Échantillonnage . . . . .	19
2.4.2	Filtrage numérique . . . . .	19
2.4.3	Détecteur de crête . . . . .	26
2.4.4	Décodeur . . . . .	26
<b>3</b>	<b>Propulsion</b>	<b>29</b>
3.1	Description et cahier des charges . . . . .	29
3.2	Prototype de la base roulante . . . . .	29
3.3	Régulation du robot . . . . .	30
3.3.1	Régulation en position . . . . .	30
3.3.2	Simulations . . . . .	34
3.3.3	Résultats et corrections . . . . .	35
<b>4</b>	<b>Conclusion</b>	<b>37</b>
	<b>Bibliographie</b>	<b>37</b>

# Table des figures

1.1	Schéma-bloc du système . . . . .	5
1.2	Diagramme de fonctionnement . . . . .	6
2.2	Équivalent de Thévenin du microphone . . . . .	9
2.1	Schéma électrique du microphone . . . . .	9
2.3	montage inverseur . . . . .	11
2.4	Principe de superposition avec diviseur résistif . . . . .	12
2.5	Principe de superposition avec signal de microphone . . . . .	12
2.6	Schéma électrique de la chaîne d'acquisition . . . . .	14
2.7	Filter Wizard . . . . .	15
2.8	Simulation chaîne d'acquisition . . . . .	16
2.9	Étude du bruit en sortie de montage . . . . .	17
2.10	Sortie de la chaîne d'acquisition pour un signal à 1500 Hz . . . . .	18
2.11	Sortie de la chaîne d'acquisition pour un signal à 7000 Hz . . . . .	18
2.12	Schéma de principe du traitement numérique . . . . .	19
2.13	Schéma de fonctionnement du filtre RII . . . . .	21
2.14	Interface graphique de filterDesigner . . . . .	21
2.15	Allure des filtres passe-bande centrés en les fréquences utiles . . . . .	22
2.16	Simulation du filtre numérique centré en 900 Hz avec en entrée un signal sinusoïdal . . . . .	23
2.17	Vérification du filtre numérique centré en 900 Hz avec en entrée un si- gnal sinusoïdal . . . . .	24
2.18	Simulation du filtre numérique du dsPIC centré en 900 Hz avec en entrée un signal sinusoïdal . . . . .	25
2.19	Vérification du filtre numérique du dsPIC centré en 900 Hz avec en en- trée un signal sinusoïdal . . . . .	26
2.20	Vérification des filtres numériques implémentés . . . . .	27
2.21	Machine d'état du décodeur . . . . .	28
3.1	Schéma de régulation proportionnelle . . . . .	30
3.2	Réponses statiques des roues . . . . .	31
3.3	Réponses indicielles des roues du robot . . . . .	33
3.4	Lieux d'Evans de $H(p)$ . . . . .	33
3.5	Courbes de Bode des roues . . . . .	34
3.6	Schéma de régulation de la roue droite . . . . .	34

3.7 Réponse de la roue droite à une consigne de $1m$ . . . . .	35
--	----

# Chapitre 1

## Introduction

### 1.1 Description du projet

Le cours d'ELEC-H309 - *Projet intégré* permet aux étudiants de la section électronique et télécommunications de mettre en application les différents concepts vus durant leur cursus de bachelier. En effet le sujet principal de ce cours consiste en l'élaboration d'un robot roulant et commandé via un canal de communication audio. Ce rapport a pour but de décrire en détail toutes les étapes de conception et justification des choix entrepris tout au long du développement du prototype. Une découpe de la conception du robot en deux grandes parties a été faite et ont été étudiées de manière indépendante : *la communication* et *la propulsion du robot*. Le matériel fourni permet de faire cette séparation car deux dsPIC différents ont été utilisés, l'un pour la partie communication et l'autre pour la partie déplacement du robot. Par conséquent, une communication série sera implémentée entre ces deux microcontrôleurs pour transmettre des informations telles que les ordres décodés par le  $\mu c$  s'occupant de l'analyse audio. Un schéma-bloc du système entier est représenté à la Figure 1.1 et est expliqué dans les sections suivantes.

### 1.2 Partie communication

Comme décrit à la section précédente, le robot sera muni d'un microphone et devra être commandé par un signal audio modulé en fréquence, qui sera généré au moyen d'un ordinateur ou d'un smartphone. Deux types de commande seront possibles :

- "Déplace-toi en ligne droite de X cm".
- "Effectue une rotation sur place de Y degrés".

Pour mettre en œuvre la communication, plusieurs modules devront être implémentés tels que l'acquisition et l'amplification du signal audio, le filtrage analogique de celui-ci suivi d'un filtrage numérique, la détection de crête pour identifier les fréquences perçues et finalement le décodage du message.

Le développement complet de la communication sera fait au chapitre 2.

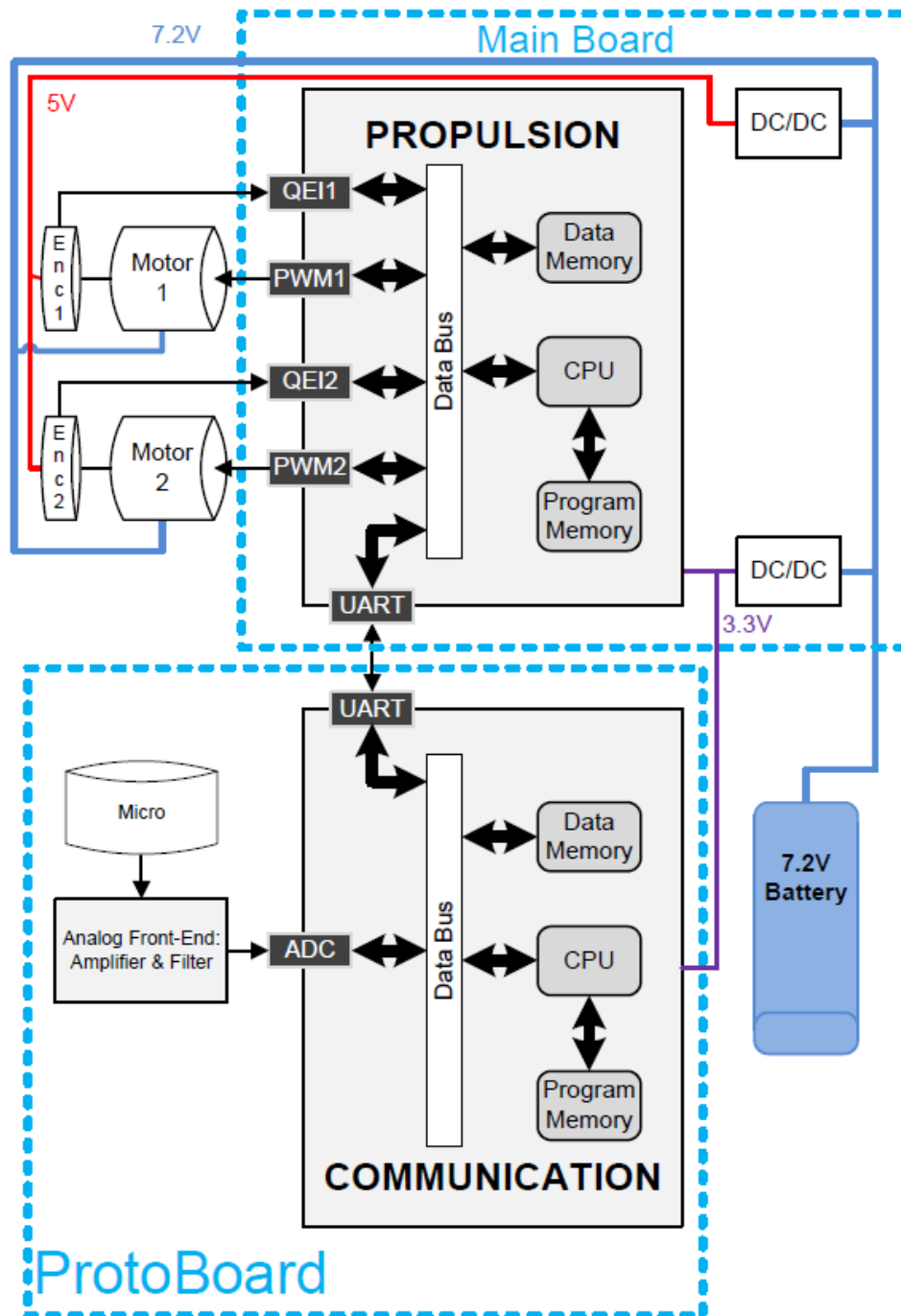


FIGURE 1.1 – Schéma-bloc du système

### 1.3 Partie propulsion

Le robot se déplace au moyen de moteurs DC commandés en PWM et permet de calculer sa distance parcourue grâce à des encodeurs en quadrature. Deux type de déplacement différents lui sont possibles :

- avancer ou reculer en suivant une ligne droite,
- faire une rotation à gauche ou à droite d'un certain angle.

Le déplacement du robot devra se faire de manière précise et en respectant la commande audio envoyée par l'ordinateur. Pour ce faire, une régulation sera donc nécessaire pour gérer la vitesse et la direction du robot. Une description mécanique du robot et une étude détaillée de la régulation de ses moteurs seront faite au chapitre 3.

Le diagramme à la Figure 1.2 résume les différentes étapes de fonctionnement du projet en entier.

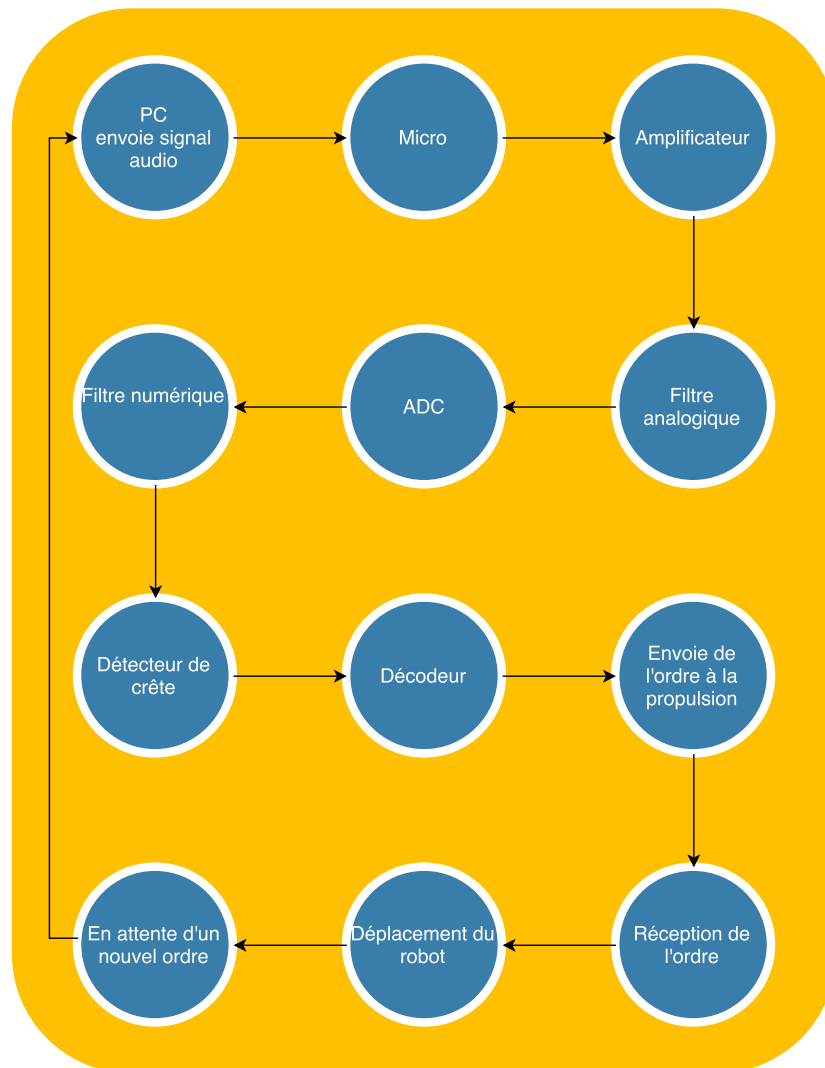


FIGURE 1.2 – Diagramme de fonctionnement

# Chapitre 2

## Communication

### 2.1 Description et cahier des charges

Comme mentionné dans l'introduction, la communication des commandes de contrôle se fait au moyen de la modulation d'un signal audio. En effet, la commande qui correspond à une trame de bits est envoyée au robot et est codée selon un protocole bien précis. Elle est composée d'un préambule, l'ordre ainsi que le paramètre lié à cet ordre et est terminée par un postambule. Le protocole de communication est détaillé dans les tableaux 2.1, 2.2 et 2.3.

Les bits qui composent le message à envoyer sont transformés en un signal audio par une modulation FSK (*Frequency Shifting Modulation*). En effet le signal audio est modulé selon la relation suivante :

$$\text{signal modulé} = \begin{cases} A_0 \sin(2\pi(f_p - \Delta f)t) & \Leftrightarrow \text{signal modulant} = 0 \\ A_0 \sin(2\pi(f_p + \Delta f)t) & \Leftrightarrow \text{signal modulant} = 1 \end{cases} \quad (2.1)$$

Dans le cadre de notre projet, la fréquence porteuse du signal  $f_p$  est de  $1kHz$ , le déplacement de fréquence  $\Delta f$  est de  $100Hz$  et la durée d'un bit est de  $100ms$ . Le programme permettant l'envoi de la commande audio est fourni.

Du côté de la réception, cette partie est à concevoir et plusieurs modules doivent donc être implémentés ce qui est le sujet principal de ce chapitre et sera détaillé dans les sections suivantes.

Ordre	Code
Avance	0b00
Tourne à droite	0b01
Tourne à gauche	0b10

TABLE 2.1 – Codage de l'ordre



Ordre	paramètre
Avance	distance à parcourir en cm. Le signe donne la direction
Tourne à droite	angle à parcourir, non signé, en degrés
Tourne à gauche	angle à parcourir, non signé, en degrés

TABLE 2.2 – Construction du message

Préambule	Données		Postambule	
Start Bit	Ordre	Paramètre	Parité	Stop bit
0	2 bits	8 bits	1 bit	0

TABLE 2.3 – Format de la trame de bits

## 2.2 Prototype pour la communication audio

Pour analyser le signal audio, le *dsPIC33FJ128GP802* configuré lors du laboratoire sur la *Communication série asynchrone* est utilisé pour échantillonner le signal audio et ce dernier est capté au moyen d'un microphone de type piézoélectrique. Cependant, un traitement analogique est indispensable avant l'échantillonnage par l'ADC du micro-contrôleur et sera détaillé dans la section 2.3. Il est à noter que le robot doit être capable de fonctionner dans un environnement calme, à l'intérieur et sans bruit de fond.

## 2.3 Traitement analogique

Comme mentionné précédemment, un traitement analogique du signal est primordial pour analyser le signal audio le plus efficacement possible. D'autant plus que ce conditionnement est imposé dans le cahier des charges. En effet, la chaîne d'acquisition audio doit être composée d'un amplificateur et d'un filtre de garde. Le signal audio converti en signal électrique sera échantillonné par un convertisseur analogique/numérique et les données recueillies par le CAN pourront être interprétées par le robot. Il est à noter que la fréquence d'échantillonnage du CAN est limitée justifiant l'utilisation d'un filtre de garde. La résolution du CAN justifie l'usage d'un amplificateur.

### 2.3.1 Microphone

Le signal audio transmis au robot est reçu grâce à un microphone piézoélectrique. Ce microphone capte les fréquences audibles situées entre 20 Hz et 20 kHz. Il sera donc nécessaire de filtrer les signaux afin d'assurer le transfert du signal vers le CAN. La

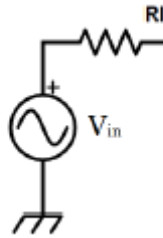


FIGURE 2.2 – Équivalent de Thévenin du microphone

*datasheet* fourni avec le microphone informe qu'un signal en tension est émis en polarisant le dispositif avec une résistance de  $2,2k\Omega$ . Ce micro se comporte comme une source de courant.

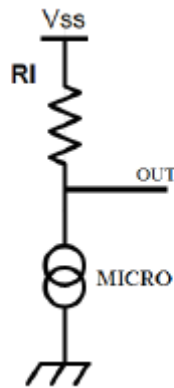


FIGURE 2.1 – Schéma électrique du microphone

Le signal d'entrée de la chaîne d'acquisition est représenté en utilisant un équivalent de Thévenin. Cet équivalent est déterminé en appliquant le théorème de superposition. En annulant la source de courant, la tension d'entrée est égale à la tension d'alimentation. Afin d'éviter que cette tension n'affecte le signal émis, l'usage d'un condensateur en série est requis. Il n'est donc pas nécessaire de prendre en compte la tension d'alimentation. En prenant en compte la contribution de la source de courant, la résistance se trouve en parallèle avec celle-ci. Ceci donne un équivalent de Norton du circuit. En convertissant cet équivalent en équivalent de Thévenin, le schéma résultant est le suivant (Figure 2.2).

La tension alternative d'entrée délivré par le microphone est de 1mV d'amplitude.

### 2.3.2 Amplification

Le signal électrique ne peut être traité par le CAN que si son amplitude est suffisamment grande. En effet, le CAN possède une plage de fonctionnement qui est de 0 à 3,3

V dans ce cas-ci. Le codage des données numériques se fait sur 10 bits dans ce projet, la résolution du CAN est donc de :

$$\frac{3,3}{2^{10}} = 3,22mV \quad (2.2)$$

La résolution est proche de l'amplitude du signal émis par le microphone. Il faut donc que le signal envoyé au CAN soit amplifié et supérieur à 0. Durant le dimensionnement de la chaîne d'acquisition, il a été décidé d'amplifier le signal émis d'un gain de 1500. Un signal de 1,5 V d'amplitude oscillant autour d'un offset de 1,65V assure une marge de sécurité dans le cas d'un niveau sonore trop élevé et ainsi que le signal entrant dans le CAN soit positif constamment.

La création de l'offset est effectuée en utilisant un diviseur résistif à l'entrée non inverseur d'un amplificateur opérationnel étant donné l'accès à une tension de 3,3 V. Chaque amplificateur opérationnel sera alimenté entre 0 V et 3,3 V afin d'éviter une surcharge du CAN.

Étant donné la création d'un offset grâce à l'entrée non-inverseur de l'amplificateur opérationnel, l'entrée inverseur de l'amplificateur est utilisé afin de traiter le signal du microphone. Il est à noter que le condensateur mentionné précédemment sera placé avant l'entrée inverseur.

### 2.3.3 Filtrage

Un filtre de garde est réalisé afin d'éviter tout phénomène de repliement spectral. Ce filtre est rajouté à la chaîne d'acquisition après les étages à amplification. Le type de filtre ainsi que l'ordre du filtre sont imposés par le cahier des charges. Le filtre analogique doit être de type "Butterworth" et d'ordre 2. L'atténuation maximal des fréquences utiles est de 0,99 et l'atténuation minimale des fréquences repliées est de 0,05.

#### Dimensionnement du filtre :

L'équation d'un filtre de "ButterWorth" est porté par le module de la fonction de transfert :

$$||H(j\omega)|| = \sqrt{\frac{1}{1 + (\frac{\omega}{\omega_c})^{2n}}} \quad (2.3)$$

Cette équation permet de trouver la fréquence de coupure du filtre de garde ainsi que la fréquence d'échantillonnage du CAN.

$$f_c = \sqrt[n]{\frac{1}{||H(j\omega)||^2 - 1}} \cdot f_p = 2925kHz \quad (2.4)$$

où  $||H(j\omega)|| = 0,99$  et  $f_p = 1100\text{Hz}$ . En utilisant cette équation une seconde fois avec  $||H(j\omega)|| = 0,05$  et en utilisant la fréquence de coupure déterminée précédemment, la fréquence d'échantillonnage est trouvée

$$f_s = 13072\text{Hz} \quad (2.5)$$

Cette fréquence est la borne inférieure de la plage de fréquence atténuée de, au moins, 0,05. Cette fréquence d'échantillonnage devra être prise en compte lors de la création de filtre numérique.

### 2.3.4 Choix des composants des étages à amplification

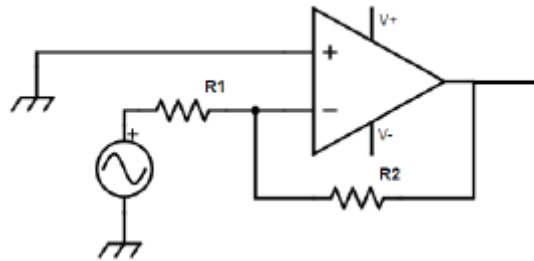


FIGURE 2.3 – montage inverseur

Il a été dit précédemment que le montage inverseur a été choisi afin de créer un offset de 1,65 V afin d'assurer l'acquisition du signal par le CAN. Etant donné que l'envoi de message se repose sur une modulation FSK, le déphasage du signal n'est en rien important.

Cependant, l'adaptation d'impédance pose un léger problème puisque la résistance  $R_1$  forme l'impédance d'entrée de l'étage à amplification.

Le gain de ce montage s'exprime ainsi :

$$G = -\frac{R_2}{R_1} \quad (2.6)$$

Il est difficile de créer un gain élevé tout en maintenant la résistance  $R_1$  élevée pour assurer l'adaptation d'impédance. Durant le projet, la résistance de polarisation du microphone est considérée comme suffisamment grande pour l'adaptation en impédance. Un montage suiveur n'est donc pas nécessaire.

La tension de sortie de l'étage à amplification est obtenue en appliquant le théorème de superposition comme mentionné auparavant.

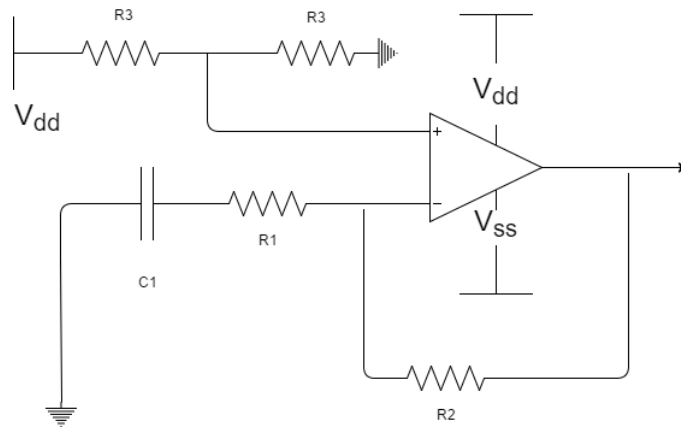


FIGURE 2.4 – Principe de superposition avec diviseur résistif

Le condensateur peut être remplacé par un circuit ouvert puisque l'on travaille avec une tension continue. Le diviseur résistif multiplie la tension d'entrée par un facteur  $\frac{1}{2}$  grâce aux résistances ayant toutes deux la même valeur. La tension accessible valant 3,3 V, on déduit que la tension d'entrée vaut 1,65 V ainsi que la tension de sortie puisque la résistance R2 est en court-circuit.

$$V_{out,DC} = 1,65V \quad (2.7)$$

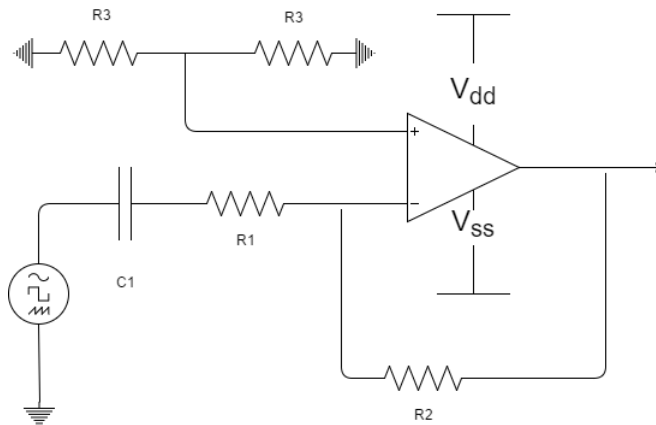


FIGURE 2.5 – Principe de superposition avec signal de microphone

Dans ce cas-ci, le condensateur est considéré comme un court-circuit en raison d'une haute fréquence. La tension de sortie est donc la multiplication de la tension d'entrée par le gain défini en (2.6).

$$V_{out,AC} = -\frac{R_2}{R_1} V_{in} \quad (2.8)$$

La fonction de transfert en prenant en compte le condensateur s'écrit :

$$H_1(\omega_{in}) = -\frac{R_2}{R_r + \frac{1}{j\omega_{in}C}} \quad (2.9)$$

Cette fonction représente un filtre passe haut. Le montage doit être fonctionnel pour une fréquence au plus de 900 Hz. La fréquence de 900 Hz doit être considéré comme une haute fréquence.

$$\frac{1}{\omega C} \ll R_1 \quad (2.10)$$

Avec  $\omega = 2\pi \cdot 900$  rad/s. La capacité doit être fortement supérieure à une capacité calculée de 18 nF. En pratique, un ou deux ordre de grandeur au dessus suffisent à respecter ce critère. La capacité sélectionnée vaut 180 nF.

La tension total est donc la somme des deux tensions calculées ci-dessus.

$$V_{out} = 1,65 - \frac{R_2}{R_1} V_{microphone} \quad (2.11)$$

Avant de sélectionner des résistances  $R_1$  et  $R_2$ , le nombre d'étage à amplification doit être déterminé.

Les amplificateurs utilisés sont de type UA748C. Ceux-ci peuvent être alimentés par une tension maximale de 22 V et minimale de -22 V La datasheet précise que le produit gain bande passante est de 1MHZ. La bande passante en raison de la présence du filtre de garde est de 2,9 kHz.

$$G = \frac{10^6}{2900} = 344,82 \quad (2.12)$$

Un seul amplificateur ne peut donc pas suffire à amplifier les signal du microphone. Un alignement de deux étages à amplification similaire est utilisé. Les condensateurs sont déterminés mais il reste à définir les valeurs des résistances à exploiter. Le schéma des étages à amplification ressemble à ceci.

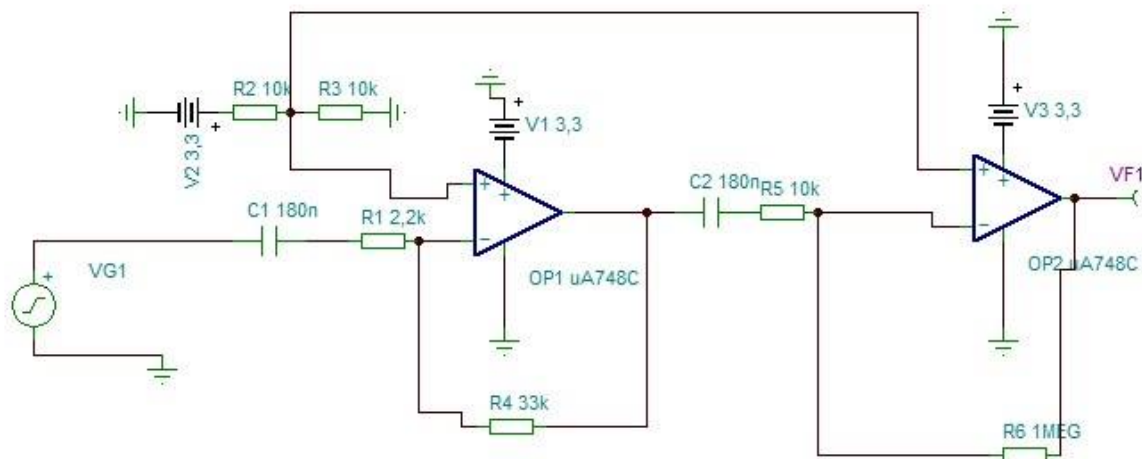


FIGURE 2.6 – Schéma électrique de la chaîne d'acquisition

La résistance de polarisation du premier étage permet de choisir une résistance de  $33k\Omega$  afin de produire un gain de -15. Il reste donc à multiplier le signal sortant du premier étage par un gain de - 100. Le signe négatif est justifié par l'utilisation de la borne non-inverseuse pour la création d'un offset. Afin de garder des courants suffisamment petits, des résistances de l'ordre de  $10k\Omega$  sont utilisées. Une résistance de  $1 M\Omega$  et une résistance de  $10k\Omega$  sont utilisées afin de créer le gain de -100.

### 2.3.5 Choix des composants du filtre analogique

Lors de la conception du filtre analogique, le logiciel en ligne fiable, "Filter Wizard", a été utilisé. Les montages électroniques n'ont pas été développés en détails lors des cours de cette année. Cela a donc mené à l'utilisation de logiciel créer par le groupe "Analog Devices".

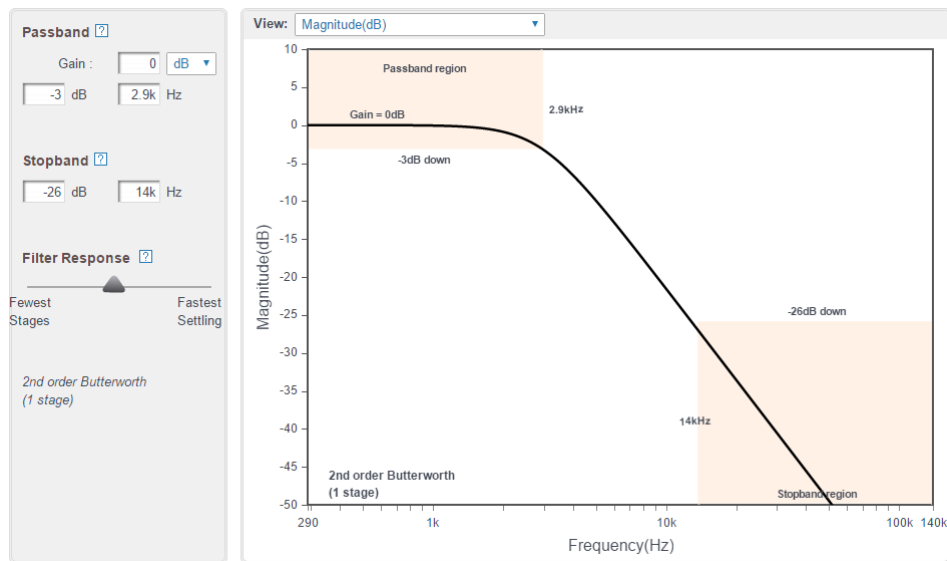
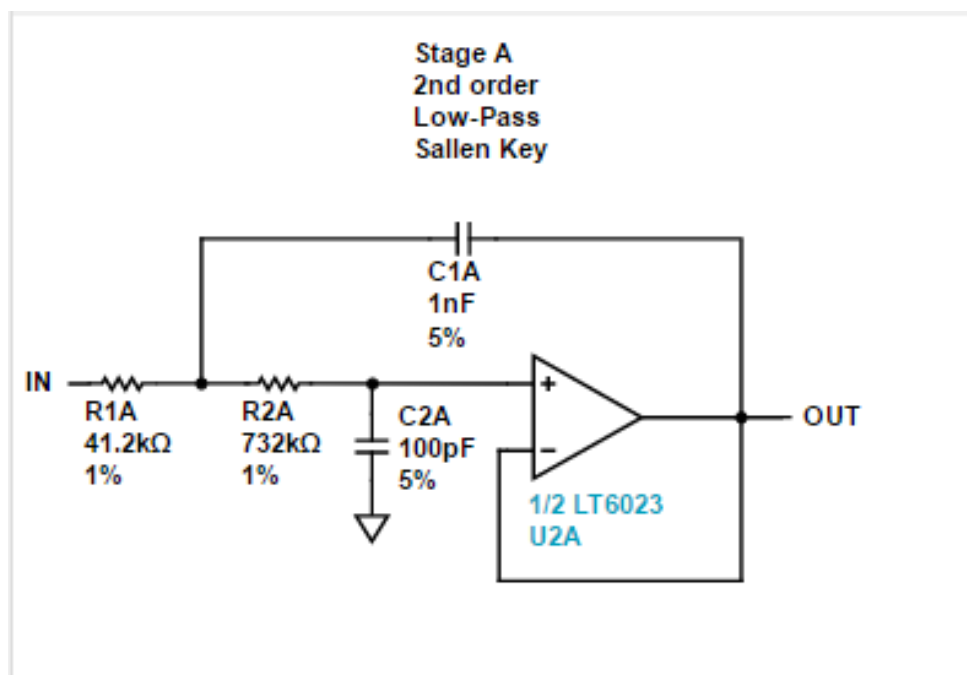


FIGURE 2.7 – Filter Wizard

Le filtre analogique proposé est le suivant :



La fréquence d'échantillonnage trouvée auparavant est de 13 kHz. Il s'agit de la borne inférieure des fréquences repliées. Cela permet de choisir une fréquence d'échantillonnage plus grande afin de produire un filtre "Butterworth" d'ordre 2. La fréquence de 14kHz est choisie et sera utilisée pour la conception de filtre numérique.

Il est à noter que l'amplificateur opérationnel choisi est le même que ceux des étages à amplifications pour des raisons qui seront citées plus tard.



### 2.3.6 Simulation

La simulation des étages à amplifications est effectuée sur le logiciel "Tina Spice" proposé par le professeur du cours *instrumentation* Antoine NONCLERCQ.

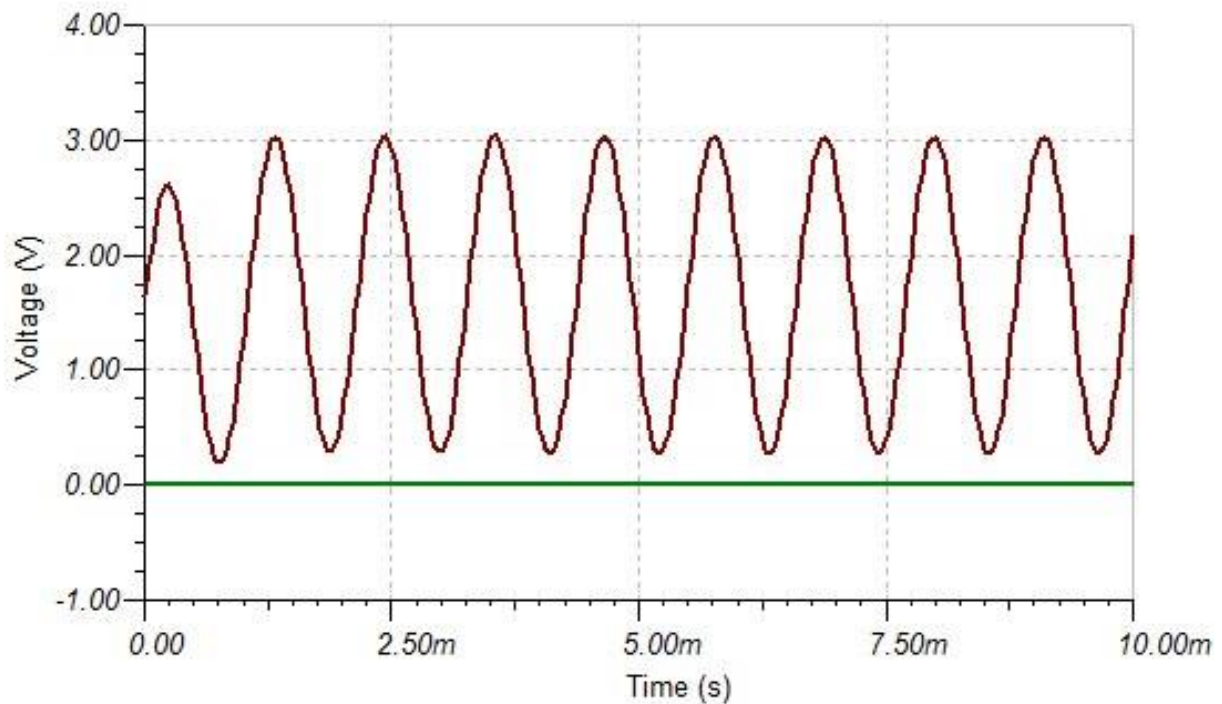


FIGURE 2.8 – Simulation chaîne d'acquisition

Le signal entrant au CAN est bien centré autour d'une tension de 1,65 V pour une amplitude de 1,5 V. On peut tout de même observer la phase de transition au temps équivalent à 0 seconde avant d'arriver à un signal en régime.

L'étude du bruit en sortie de montage est fournie par le même logiciel.

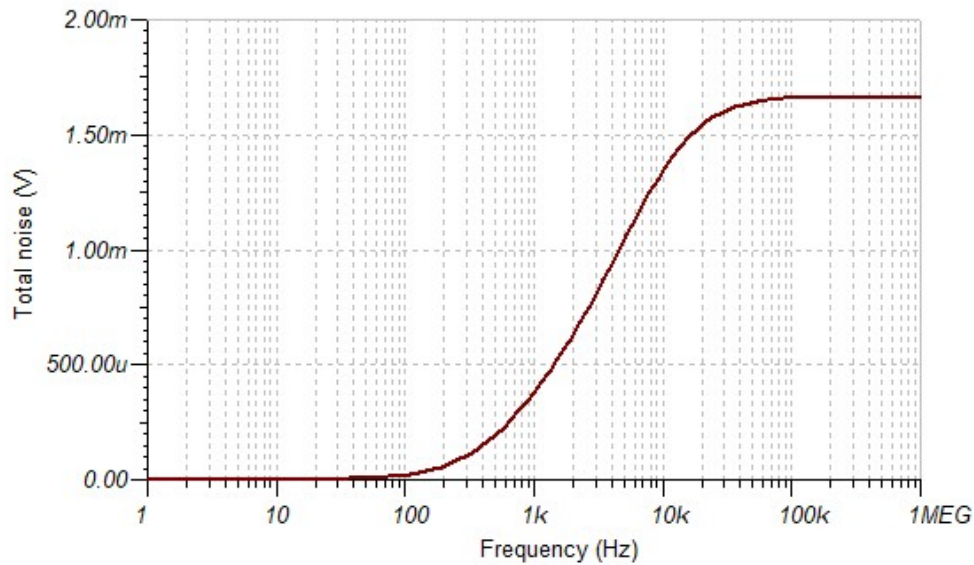


FIGURE 2.9 – Étude du bruit en sortie de montage

On remarque que pour une fréquence de 900 Hz le bruit possède une valeur en "root mean square" inférieure à  $500\mu\text{V}$ . La valeur du bruit est légèrement au dessus de  $500\mu\text{V}$ . Cette valeur du bruit est largement inférieure à l'amplitude du signal résultant. Ceci justifie l'utilisation des amplificateurs opérationnels sélectionnés antérieurement. On peut donc utiliser l'amplificateur uA748C pour la conception du filtre de garde.

### 2.3.7 Validation de la chaîne

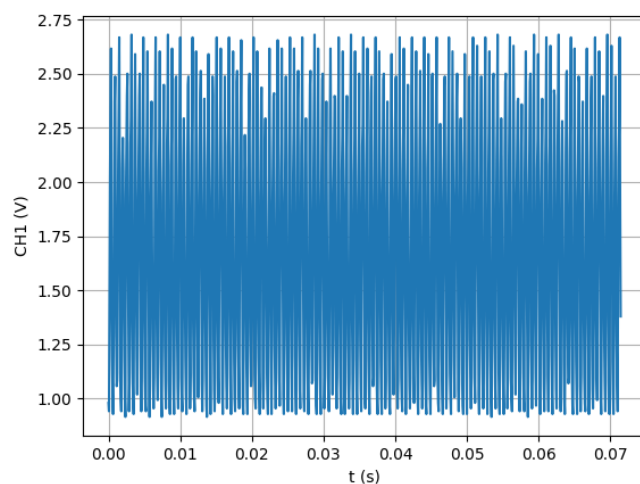


FIGURE 2.10 – Sortie de la chaîne d’acquisition pour un signal à 1500 Hz

Le filtre analogique est validé en appliquant un signal inférieur à la fréquence de coupure de 2,9 kHz. Pour une fréquence de 1500 Hz, la tension se situe entre 1 V et 2,75 V. Cette tension est moins élevée que prévue mais satisfaisante puisque qu’il est possible de démoduler ce signal. L’amplitude observée peut s’expliquer par le fait que le niveau sonore capté par le microphone n’était pas élevé lors de l’acquisition.

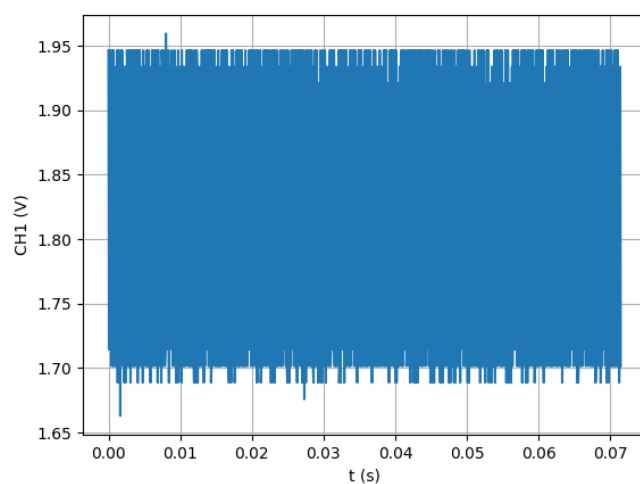


FIGURE 2.11 – Sortie de la chaîne d’acquisition pour un signal à 7000 Hz

Pour une fréquence de 7000 Hz, on remarque une forte atténuation du signal puisque le signal oscille entre 1,95 V et 1,70 V.

On peut désormais conclure que la chaîne d'acquisition est opérationnelle et la suite du projet peut être développée.

## 2.4 Traitement numérique

### 2.4.1 Échantillonnage

Une fois le signal traité analogiquement, ce dernier doit maintenant subir un traitement numérique dont le schéma de principe est décrit à la Figure 2.12. Grâce au filtre de garde, les hautes fréquences de bruit ont été atténuées et l'ADC peut donc échantillonner le signal sans craindre un repli spectral. La fréquence d'échantillonnage minimale a été établie à la section précédente. Par conséquent le choix du cadencement de l'ADC a été choisi à  $14\text{kHz}$  et celui-ci est cadencé par le *TIMER3* du dsPIC. Le microcontrôleur analyse donc la tension à l'entrée de sa patte analogique *AN0* et stocke cette valeur sur 10 bits dans son buffer. Il est également possible de stocker la tension sur 12 bits mais une précision de 10 bits était suffisante dans le cadre de ce projet. De ce fait, la tension en aval du montage analogique comprise entre 0 et  $3,3\text{V}$  est quantifiée en binaire dans une plage de valeurs comprises entre 0 et 1023. Le signal oscillant autour de l'offset de  $1,65\text{V}$  oscillera donc autour de la valeur 512 dans le dsPIC.

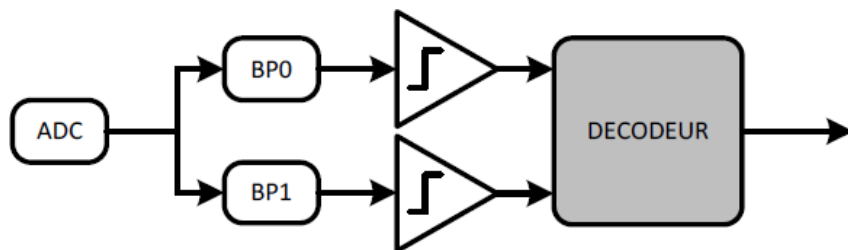


FIGURE 2.12 – Schéma de principe du traitement numérique

### 2.4.2 Filtrage numérique

#### Détermination du type de filtre

Les échantillons recueillis par le convertisseur analogique-numérique doivent ensuite passer par deux filtres numériques passe-bande comme illustré sur la Figure 2.12. Ces deux filtres devront être centrés chacun sur une des fréquences du signal modulé :  $900\text{Hz}$  et  $1100\text{Hz}$ . Ainsi donc le filtre numérique atténuera toutes les autres fréquences autres que les fréquences utiles et permettra de faire la correspondance fréquence-bit dans la suite grâce aux détecteurs de crête.

Un cahier des charges bien précis est imposé pour le filtre numérique :

- Les fréquences utiles se situent dans une bande de  $\pm 1,5\%$  autour de la fréquence centrale

- Les fréquences situées au-delà de  $\pm 3,5\%$  de la fréquence centrale doivent être coupées
- Atténuation maximale des fréquences utiles :  $H1 = 0,9$
- Atténuation minimale des fréquences coupées :  $H2 = 0,1$

La première étape dans la conception du filtre numérique a été la détermination du type de filtre à implémenter. En effet deux grandes familles de filtre existent :

1. les filtres à réponse impulsionnelle finie ou *RIF*
2. les filtres à réponse impulsionnelle infinie ou *RII*

La principale différence entre ces deux catégories de filtres est que celui à RIF est un filtre non récursif tandis que RII est un filtre récursif. Cette différence n'est pas anodine, en effet l'implémentation d'un filtre non récursif nécessitera d'avoir un ordre largement supérieur pour avoir les mêmes caractéristiques que le filtre récursif en terme de module. Le nombre de calcul pourrait également être supérieur avec un RIF ce qui risque d'être déroutant car le signal doit être filtré en continu et rapidement. Néanmoins le RII possède également des désavantages car celui-ci est sujet aux instabilités dues à la récursivité qui est susceptible d'accumuler des erreurs mais cela peut être anticipé si le choix des coefficients est fait de manière correcte. Le choix d'un filtre récursif est donc plus adapté pour ce projet. L'équation aux différences décrivant le RII ainsi qu'un schéma (Figure 2.13) le décrivant sont les suivants :

$$y[n] = \sum_{i=0}^N b_i \cdot x[n-i] - \sum_{j=1}^M a_j \cdot y[n-j] \quad (2.13)$$

où :

- $x$  représente l'entrée du filtre
- $y$  représente la sortie du filtre
- $b_i$  et  $a_j$  les coefficients du filtre

Et sa fonction de transfert :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_N \cdot z^{-N}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_M \cdot z^{-M}} \quad (2.14)$$

A titre d'information, les filtres non récursifs sont décrits par la même équation 2.13 mais dont les coefficients  $a_j$  sont tous nuls.

### Détermination des coefficients des filtres

Afin d'exploiter la relation 2.13, il est donc nécessaire d'établir les coefficients  $b_i$  et  $a_j$  du filtre. Heureusement, *MATLAB* possède un outil puissant permettant de designer

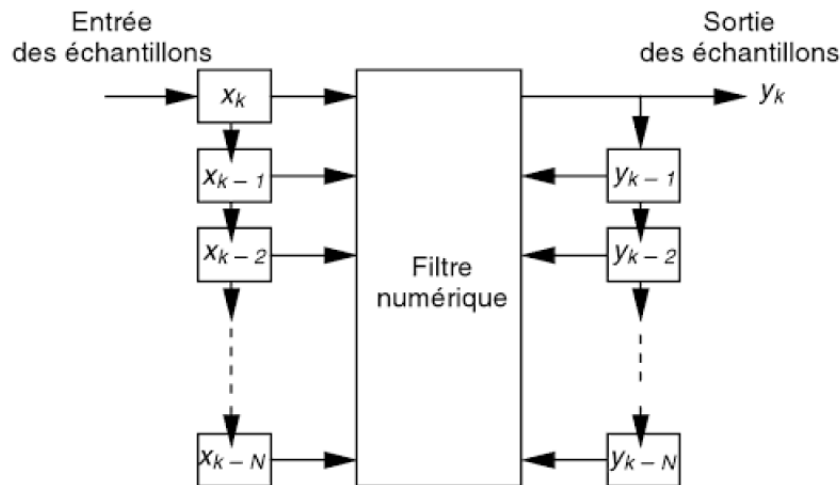


FIGURE 2.13 – Schéma de fonctionnement du filtre RII

toutes sortes de filtres aisément et avec une bonne précision. Il s'agit de *filterDesigner*. Cet outil possède une interface graphique facilitant son utilisation et la création de filtre numérique. Une image de cette interface est représentée à la Figure 2.14.

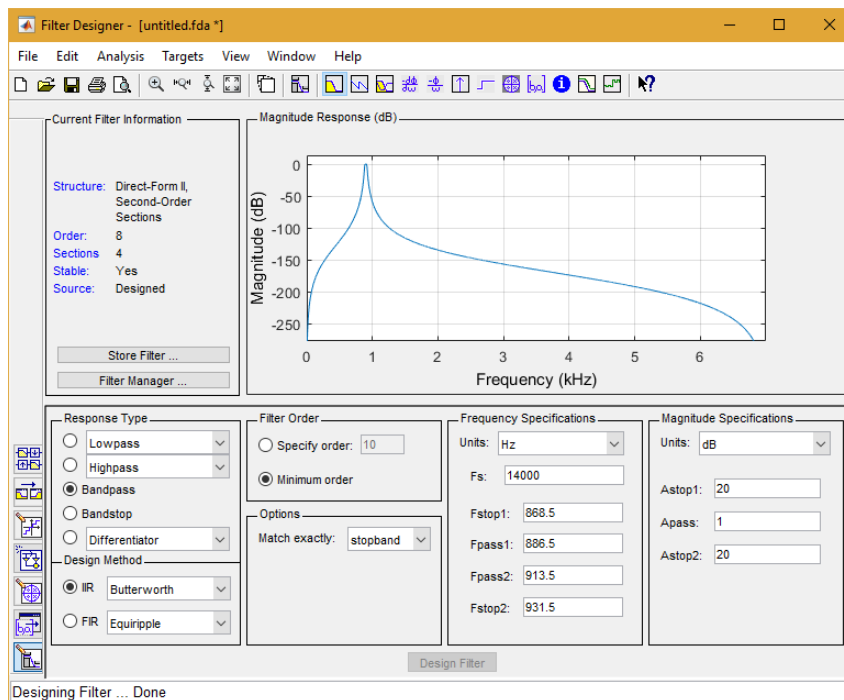
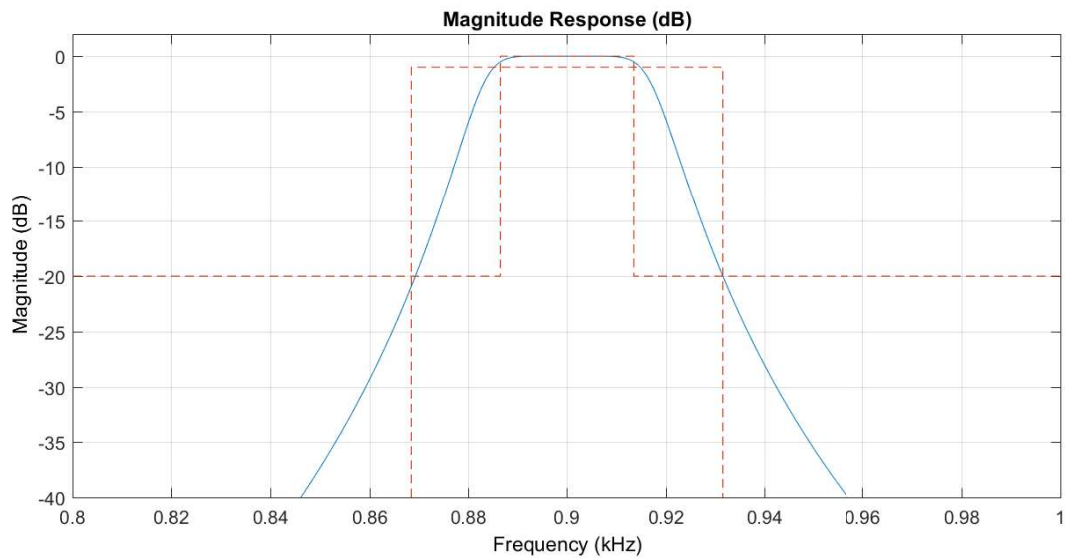


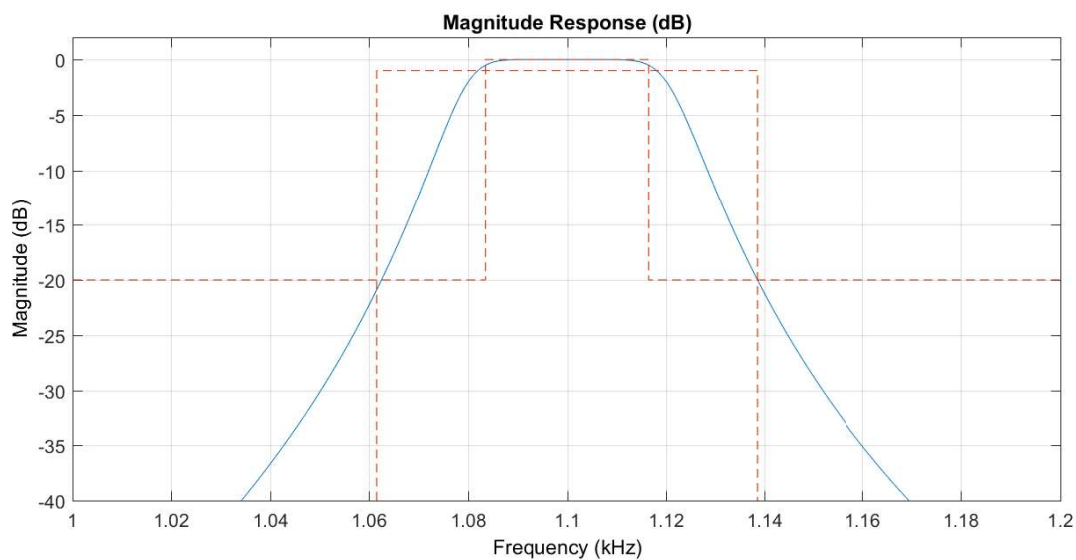
FIGURE 2.14 – Interface graphique de filterDesigner

Les caractéristiques du filtre doivent respecter le cahier des charges détaillé précédemment et sont introduit tout simplement dans *filterDesigner*. Celui-ci conçoit ensuite deux filtres numériques passe-bande centrés en 900Hz et 1100Hz d'ordre 8. De plus

une indication montre si les filtres sont stables ou pas et dans ce cas-ci, ils le sont. L'allure de ces filtres est représentée à la Figure 2.15 et répond bien aux spécifications demandées.



(a) 900 Hz



(b) 1100 Hz

FIGURE 2.15 – Allure des filtres passe-bande centrés en les fréquences utiles

Les filtres obtenus sont constitués de 4 sections d'ordre 3 qui seront mises en cascade, la sortie de la première section correspondra à l'entrée de la seconde section et ainsi de suite.

## Simulation des filtres numériques

Maintenant que les coefficients des filtres ont été trouvés, une simulation et validation basée sur l'équation 2.13 des filtres avec *MATLAB* est nécessaire pour s'assurer de leur bon fonctionnement. Les résultats de la simulation du filtre centrée en 900Hz sont repris à la Figure 2.16 et ont été faits avec une fréquence d'échantillonnage correspondant à celui de l'ADC (14kHz). Plusieurs signaux sinusoïdaux de fréquence différente ont été appliqués en entrée (Figure 2.17) afin de vérifier si le filtre respecte bien les caractéristiques imposées par le cahier des charges. Ces graphes confirment bien l'efficacité du filtre et la simulation du filtre centré en 1100Hz a montré des résultats similaires mais ne sont pas explicités dans ce rapport.

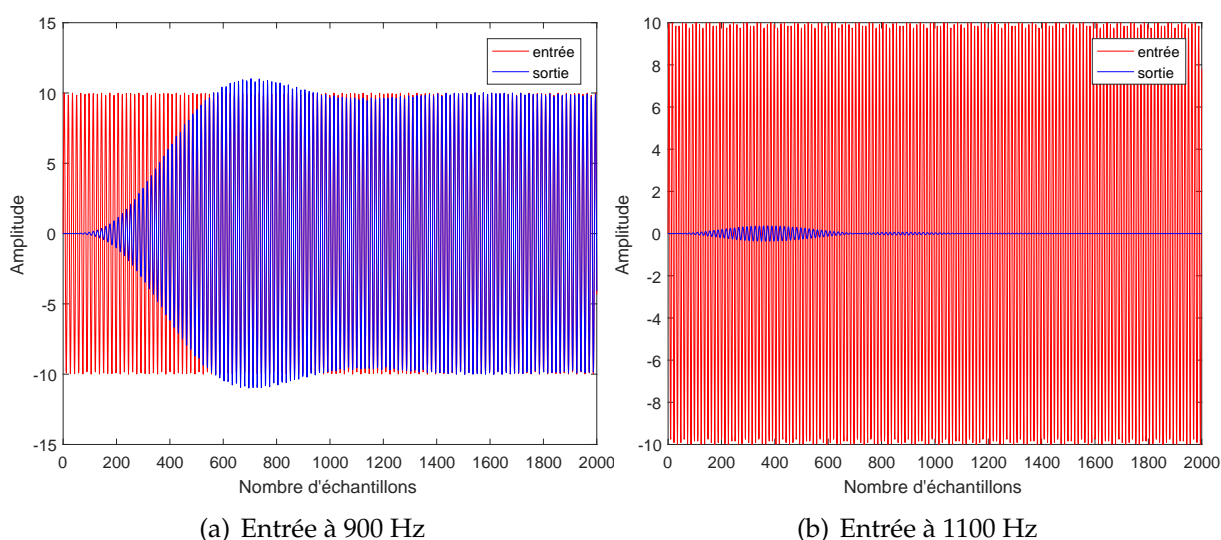


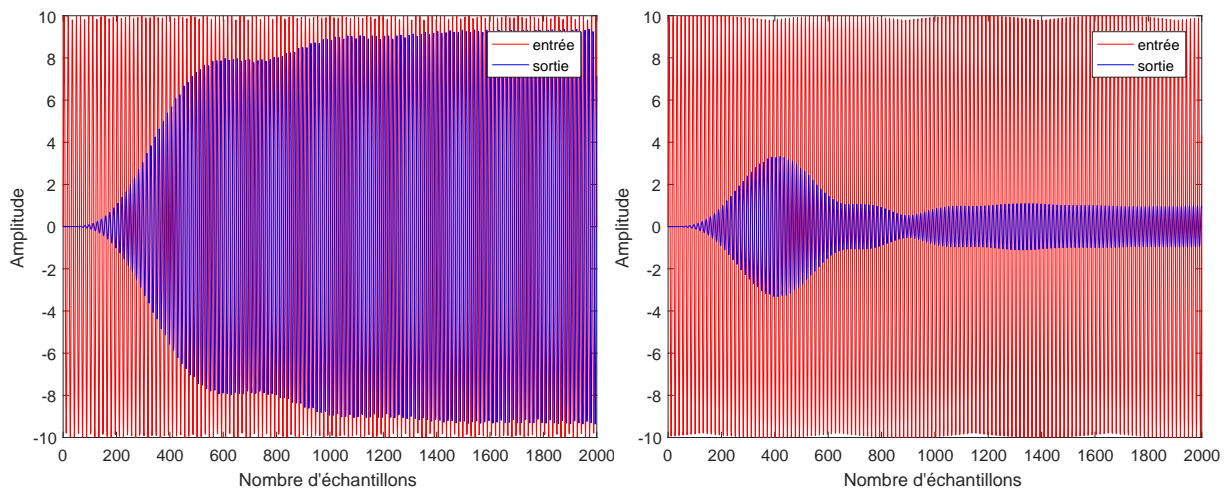
FIGURE 2.16 – Simulation du filtre numérique centré en 900 Hz avec en entrée un signal sinusoïdal

Le lecteur pourra remarquer sur ces graphes un phénomène transitoire dans le signal en sortie des filtres. En effet cela est dû au fait que le filtrage d'un échantillon se base sur les échantillons antérieurs à celui-ci. Il est donc nécessaire de récolter un certain nombre d'échantillons avant d'avoir le signal d'amplitude constante en sortie.

## Implémentation des filtres numériques

Une première implémentation sur le dsPIC des filtres numériques simulés précédemment a été faite et très vite abandonnée. En effet, bien que les calculs en nombre flottant soient possibles et que le microcontrôleur travaille avec une fréquence de cycle





(a) Entrée 913,5 Hz, atténuation d'au plus 0,915 dB (b) Entrée 931,5 Hz, atténuation d'au moins 20 dB

FIGURE 2.17 – Vérification du filtre numérique centré en 900 Hz avec en entrée un signal sinusoïdal

de 40MHz, les opérations en nombre flottants sont assez coûteuse en temps. Ceux-ci prennent nettement plus de temps que pour des calculs avec des nombres entiers. Cette première implémentation ne permettait donc pas de filtrer les échantillons de l'ADC assez rapidement par rapport à la fréquence d'échantillonnage du convertisseur. Il a donc été nécessaire de trouver une solution pour pallier à ce problème.

La solution a en fait été élémentaire, les coefficients ont tout simplement été multipliés par une constante afin que leur partie décimale passe dans la partie entière. Par exemple, 0,91345 devient 91345 si la constante utilisée vaut 100000. Il suffit ensuite de diviser le résultat par cette même constante pour obtenir la valeur recherchée permettant ainsi au calculateur du PIC d'effectuer des opérations mathématiques avec des nombres entiers. Cependant cette astuce possède ses limites si la partie décimale des coefficients est toujours présente après la multiplication par la constante, cela conduit à une perte de précision en échange d'un gain de temps sur les calculs. Cette perte de précision sera étudiée afin de déterminer si les coefficients entiers sont assez précis pour être implémentés sur le dsPIC.

De plus le choix de la constante peut sembler à première vue arbitraire mais cela n'est pas le cas. Effectivement si la constante est une puissance de 2, par exemple  $2^{16}$ , lors de la division par la constante pour obtenir le résultat recherché, l'unité arithmétique et logique (ALU) du  $\mu c$  pourra simplement procéder à un décalage des bits vers la droite de 16 bits car un décalage binaire de  $n$  bits vers la droite correspond à une division par  $2^n$ . En outre, la constante ne doit pas être trop grande afin de ne pas arriver en *overflow* lors des calculs car les données sont stockées dans des variables de type long codées sur 32 bits et pouvant donc prendre des valeurs comprises entre -2 147 483 648 et 2 147 483 647. Une constante de  $2^{16}$  semble être bonne, apportant une bonne précision tout

en laissant une assez grande marge pour ne pas atteindre l'*overflow*.

Une simulation précise des calculs du dsPIC a donc été faite et les résultats sont représentés à la Figure 2.18 et 2.19. Afin que cette simulation soit la plus proche du comportement du PIC, le signal en entrée est une sinusoïde de fréquence variable qui oscille autour de 512, valeur correspondant à l'offset en entrée de l'ADC. Cette sinusoïde est échantillonnée à  $14\text{kHz}$  et stockée dans des entiers. Les coefficients en nombre flottant ont été multipliés par la constante  $2^{16}$  puis convertis en *int32* abandonnant donc leur partie décimale et le décalage binaire a été effectué en MATLAB au moyen de la fonction *bitshift*.

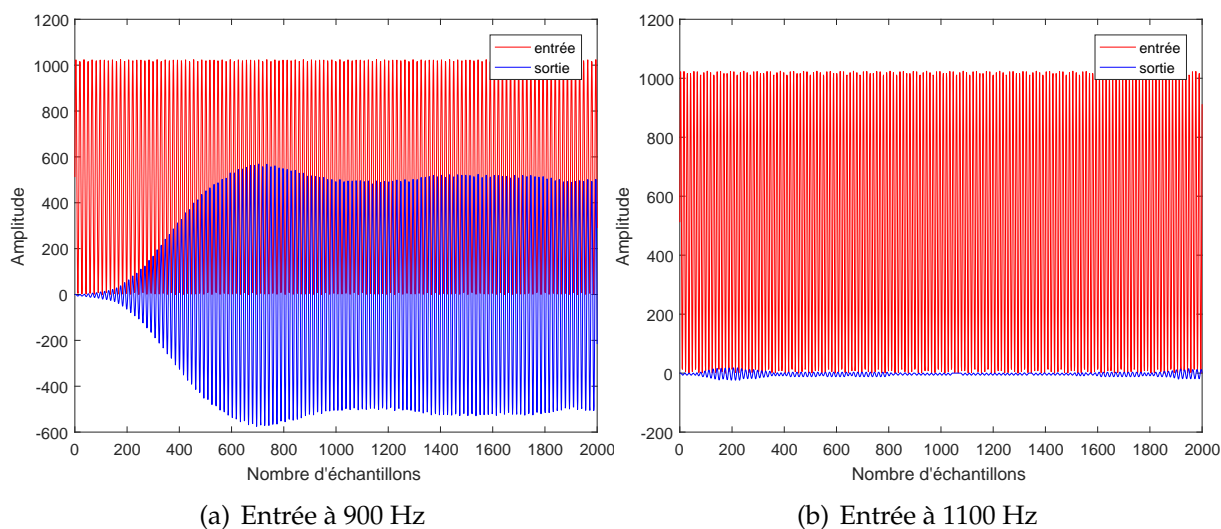
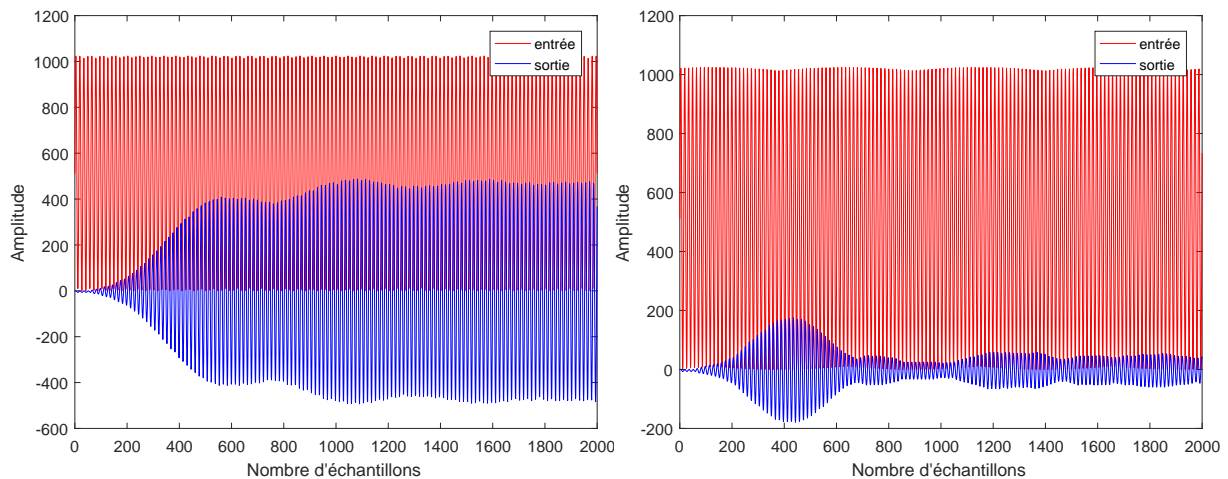


FIGURE 2.18 – Simulation du filtre numérique du dsPIC centré en 900 Hz avec en entrée un signal sinusoïdal

On remarque sur les Figures 2.18 et 2.19 que le filtre recentre le signal autour de 0 dû au fait que la composante continue de l'offset est atténuée par le filtre. Ceci n'est pas problématique du tout mais influencera sur la méthode choisie pour le détecteur de crête décrit à la section suivante. Encore une fois la simulation pour le filtre centré en  $1100\text{Hz}$  n'a pas été explicité mais les résultats étaient forts similaires à celui de  $900\text{Hz}$ . La simulation en nombre entier étant très proche de la simulation en nombre flottant, ceci valide donc l'efficacité des coefficients entiers et son implémentation sur le dsPIC. La Figure 2.20 représente la sortie des filtres numériques du microcontrôleur pour plusieurs signaux de fréquences différentes en entrée et valide les filtres numériques implémentés, l'ordre de grandeur des graphes de droite est beaucoup faible. Ces graphes ont été réalisés grâce au code oscilloscope du premier laboratoire.



(a) Entrée 913,5 Hz, atténuation d'au plus 0,915 dB (b) Entrée 931,5 Hz, atténuation d'au moins 20 dB

FIGURE 2.19 – Vérification du filtre numérique du dsPIC centré en 900 Hz avec en entrée un signal sinusoïdal

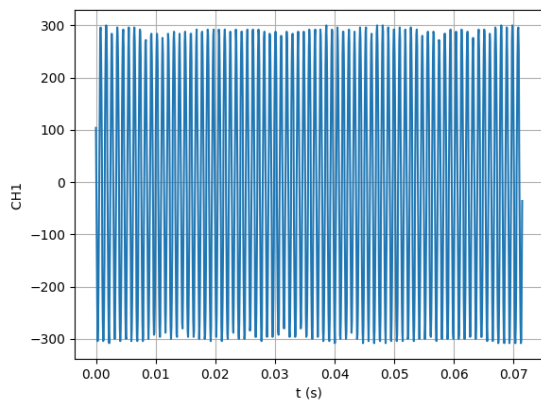
### 2.4.3 Détecteur de crête

Une fois le signal filtré numériquement, celui entre dans le détecteur de crête détectant ainsi la présence d'un signal à 900Hz ou 1100Hz comme illustré à la Figure 2.12. Le fonctionnement du détecteur est assez simple, ce dernier stocke en mémoire un certain nombre d'échantillons afin d'observer au moins une période du signal et renvoie un 1 si parmi ces échantillons la valeur crête à crête est supérieur à une valeur seuil déterminée expérimentalement. Si ce n'est pas le cas, le détecteur renvoie 0. La détection de la présence d'un signal à 900Hz ou 1100Hz permet donc la détection d'un BIT 0 ou d'un BIT 1.

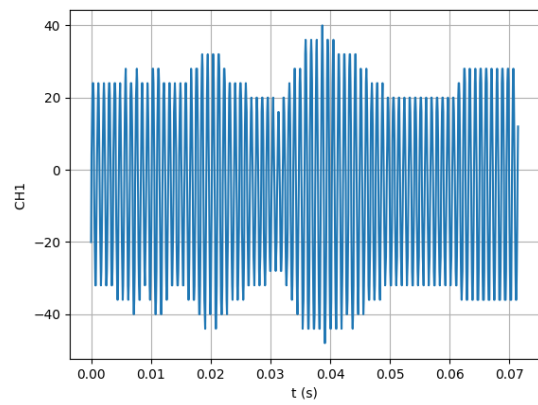
### 2.4.4 Décodeur

Le décodeur permettant de reconstituer le message d'origine est fourni afin de faciliter le développement du projet. Il est implémenté sous la forme d'une fonction *fskDetector()* prenant comme argument la sortie des deux détecteurs de crête et renvoyant le message décodé seulement si celui-ci est bien formaté selon le protocole de communication détaillé à la section 2.1. Le décodage d'une trame de bits étant un processus séquentiel, la fonction emploie donc une machine d'état détaillée à la Figure 2.21.

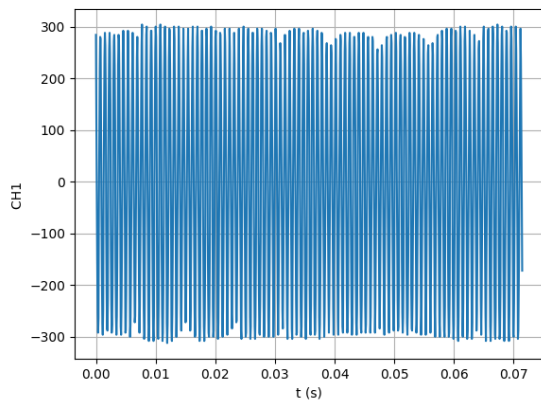
Une fois le message reconstitué, celui-ci est envoyé au dsPIC gérant la propulsion via une communication série.



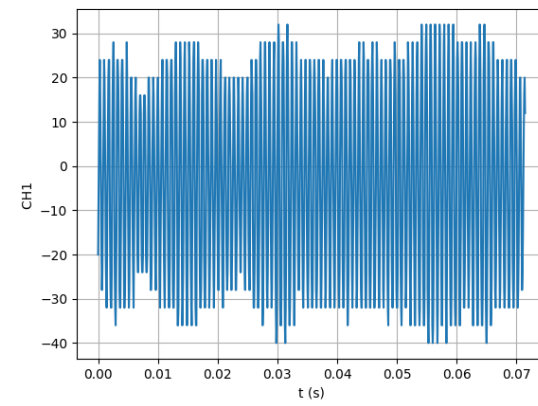
(a) Filtre 900Hz, entrée 900Hz



(b) Filtre 900Hz, entrée 931Hz



(c) Filtre 1100Hz, entrée 1100Hz



(d) Filtre 1100Hz, entrée 1138Hz

FIGURE 2.20 – Vérification des filtres numériques implémentés

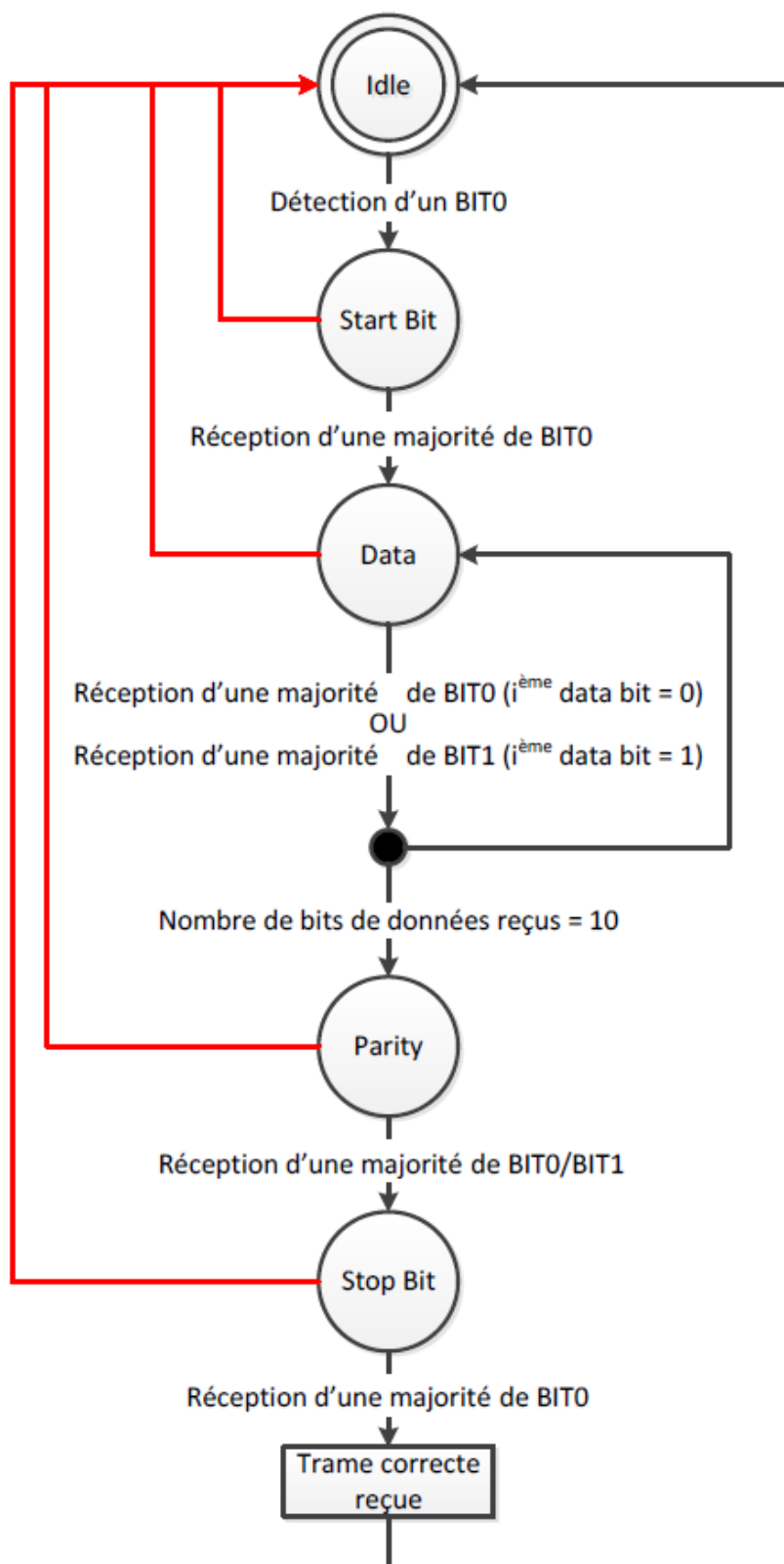


FIGURE 2.21 – Machine d'état du décodeur

# Chapitre 3

## Propulsion

### 3.1 Description et cahier des charges

Comme mentionné dans le chapitre d'introduction, le robot devra se déplacer selon des critères bien précis imposés. Celui-ci doit se déplacer de manière autonome en fonction des ordres qu'il aura reçus provenant du dsPIC de communication étudié au chapitre précédent. Deux mouvements lui sont possibles : une translation de  $X$  cm ou une rotation d'un certain angle. Afin de pouvoir effectuer ces mouvements avec précision, il est impératif d'implémenter une régulation pour contrôler la vitesse du robot. En effet le cahier des charges impose des contraintes concernant la propulsion du robot :

- se déplacer en ligne droite de  $X$  cm ;
- effectuer une rotation sur place de  $Y$  degrés ;
- être capable de se déplacer sur un terrain solide, plat et horizontal ;
- la vitesse de translation du robot doit être telle que celui-ci puisse être facilement "maîtrisé" par les étudiants si nécessaire durant les tests de leurs programmes, indépendamment de la manière dont les moteurs sont commandés ;
- l'accélération au démarrage devra être limitée afin d'éviter le patinage des roues au sol.

### 3.2 Prototype de la base roulante

La partie mécanique de la base roulante a été réalisée à l'avance dans le but de focaliser les étudiants sur la programmation du robot. Elle a été conçue le plus simplement possible et consiste en un tricycle équipé de deux roues motrices indépendantes, actionnées chacune par un moteur à courant continu ainsi que d'une bille assurant le troisième appui tout en influençant au minimum les mouvements du robot. Afin de permettre au robot de se repérer dans l'espace, celui-ci est équipé de deux capteurs de rotation montés sur l'axe des roues pour mesurer leur déplacement angulaire.

Pour contrôler les moteurs et relever la position, un *dsPIC33FJ128MC804* a été utilisé

car c'est un processeur spécialisé dans le contrôle de moteurs. Il possède 2 périphériques d'interface pour encodeur en quadrature et un périphérique *Output Compare* capable de générer 4 PWM, idéal donc pour ce projet. Le robot contrôlera donc les deux moteurs DC indépendamment à l'aide de deux signaux PWM générés par le *Output Compare* du dsPIC. Le détail des signaux de commande des moteurs est donné dans le document fourni *Etude du déplacement du robot*. Les encodeurs quant à eux fonctionnent simplement grâce aux QEI du PIC qui incrémentent un compteur en fonction du sens de rotation de la roue ce qui permet de relever la variation angulaire de cette dernière. Il suffit ensuite de calculer la position du robot en connaissant cette variation angulaire et le rayon des roues.

Le robot devant être autonome, une batterie de 7,2V comme illustrée à la Figure 1.1 est utilisée afin d'alimenter tous les composants du robot.

### 3.3 Régulation du robot

#### 3.3.1 Régulation en position

Afin de réguler le robot en position, il a fallu en premier lieu déterminer les réponses statiques des roues du robot, celles-ci caractérisent l'évolution de leur vitesse en fonction du rapport cyclique appliqué au moteur. Les caractéristiques statiques de chacune des roues sont représentées à la Figure 3.2. On observe sur ces courbes une différence notable entre leur zone morte. En effet, on remarque que la roue gauche nécessite une tension minimale plus importante que la roue droite pour imposer un mouvement de la roue. Comme l'on souhaite une relation linéaire entre le rapport cyclique et la vitesse, une vitesse nominale de  $0.3m/s$  a été imposée au robot. La régulation se fera donc en-deçà de ce point de fonctionnement, là où la caractéristique est quasi-linéaire.

Il a été décidé de concevoir un régulateur proportionnel. Le régulateur recevra en entrée la position de consigne et retournera une vitesse de consigne directement proportionnelle à l'erreur sur la position. Celle-ci sera convertie en rapport cyclique grâce à la caractéristique statique du moteur. Le principe de la régulation est résumé dans le schéma de la Figure 3.1.

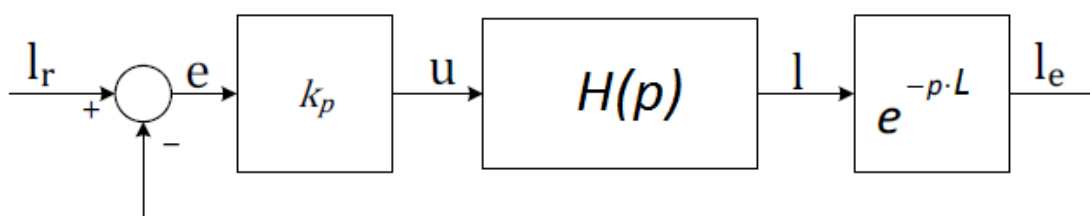


FIGURE 3.1 – Schéma de régulation proportionnelle

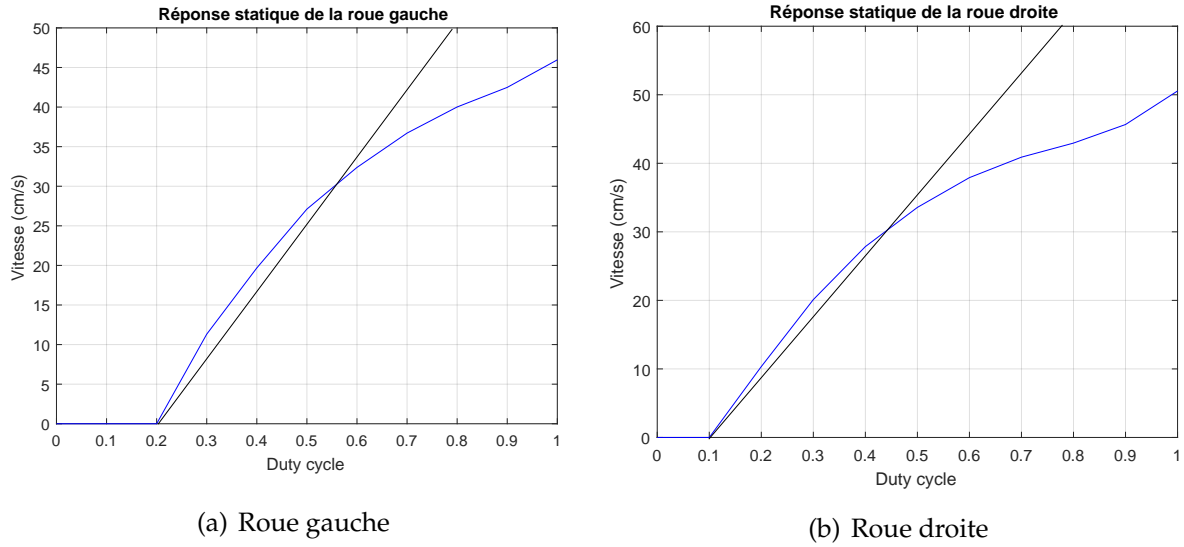


FIGURE 3.2 – Réponses statiques des roues

Par après, il a fallu relever les réponses indicielles des roues à un échelon de tension. Cet échelon de tension a été choisi autour du point de fonctionnement du robot correspondant à la vitesse nominale choisie pour ce projet, à savoir  $0.3m/s$ . Ce point est déterminé à partir de la réponse statique des moteurs. Les réponses indicielles pour chacune des deux roues ont été mesurées et sont représentées à la Figure 3.3. Afin de supprimer le bruit de quantification présent sur ces courbes, les données recueillies ont été filtrées à l'aide d'un filtre passe-bas dont la fréquence de coupure est de  $10Hz$ . A partir de ces réponses indicielles, il a été possible de modéliser le système en question par une fonction de transfert du premier ordre :

$$G(p) = \frac{V_{out}(p)}{U(p)} = \frac{A_0 K_v}{(1 + p\tau)} e^{-pL} \quad (3.1)$$

Le gain statique  $A_0$ , la constance de temps  $\tau$  ainsi que le retard  $L$  de chacune des roues sont résumés dans le Tableau 3.1. Le paramètre  $K_v$  représente la pente de l'approximation linéaire de la caractéristique statique (cf : Tableau 3.1). Il s'agit de la constante qui lie directement le rapport cyclique à la vitesse de consigne par la formule suivante :

$$V = \begin{cases} K_v(\delta - \delta_0) = K_v u & si \quad \delta > \delta_0 \\ K_v(\delta + \delta_0) = K_v u & si \quad \delta < -\delta_0 \\ 0 & si \quad -\delta_0 \leq \delta \leq \delta_0 \end{cases} \quad (3.2)$$



	Roue gauche	Roue droite
$A_0[m/s]$	1	1
$\tau[ms]$	61,5	70
$L[ms]$	80	100
$K_v[m/s]$	0.86	0.91
$\delta_0$	0.2	0.1

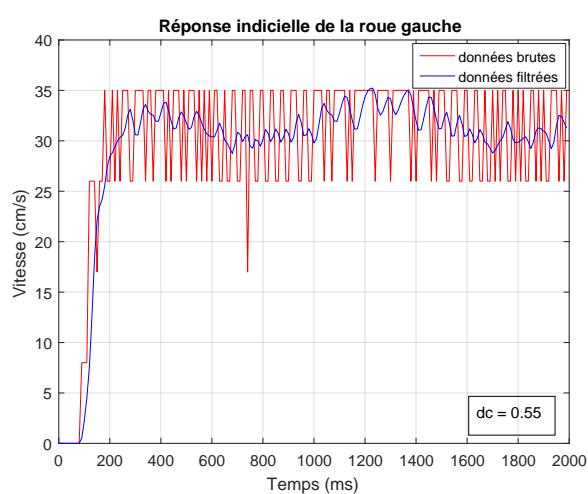
TABLE 3.1 – Données relatives aux réponses indicielles des roues

Le régulateur proportionnel doit asservir la position du robot. C'est pourquoi on introduit un pôle intégrateur à la réponse indiciale du robot modélisée en (3.1) :

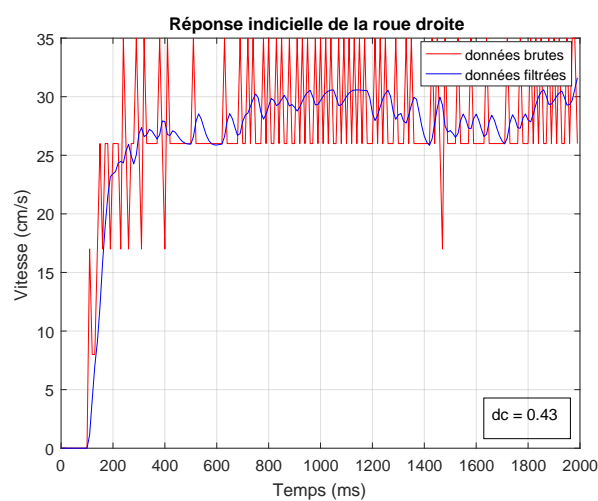
$$H(p) = \frac{G(p)}{p} = \frac{A_0 K_v}{(1 + p\tau)p} e^{-pL} \quad (3.3)$$

Le dimensionnement du gain proportionnel du régulateur se réalise grâce aux lieux d'Evans de la fonction de transfert  $H(p)$  pour chacune des deux roues. Ceux-ci sont représentés à la Figure 3.4. Le critère de stabilité de la boucle fermée est respecté en choisissant un pôle à partie réelle négative. Afin d'éviter toute oscillation autour de la position d'équilibre, il a été décidé de choisir le point du lieu à partie imaginaire nulle, c'est à dire dont le facteur d'amortissement est maximal. Une fois le gain  $K_p$  déterminé pour chaque roue, les courbes de Bode du produit  $K_p H(p)$  ont été établies (voir Figure 3.5). Elles présentent toutes deux une marge de gain supérieure à  $6dB$  ainsi qu'une marge de phase supérieure à  $30^\circ$ , qui sont les valeurs de bonne pratique de régulation.

Il est à noter que la régulation en translation et en rotation sont gérées de la même manière, elles font donc appel à la même fonction de régulation. Seuls les paramètres de cette fonction changent en fonction du type de mouvement voulu. Par exemple, si le robot doit tourner d'un angle de  $\alpha$  radians vers la droite, la commande équivalente correspondra à une translation de la roue droite de  $R\alpha$  ainsi qu'une translation de la roue gauche de  $-R\alpha$  où  $R$  est la demi-distance entre les deux roues.

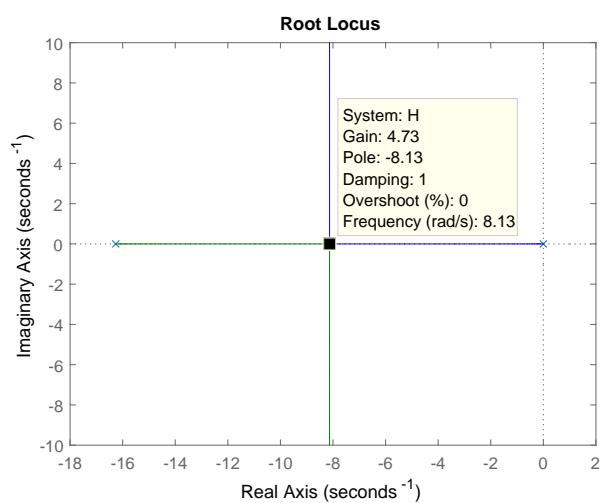


(a) Roue gauche

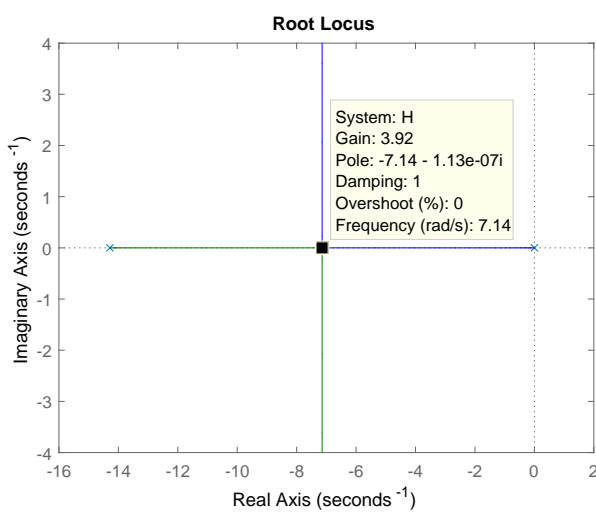


(b) Roue droite

FIGURE 3.3 – Réponses indicielles des roues du robot



(a) Roue gauche,  $Kp = 4.73$



(b) Roue droite,  $Kp = 3.92$

FIGURE 3.4 – Lieux d'Evans de  $H(p)$

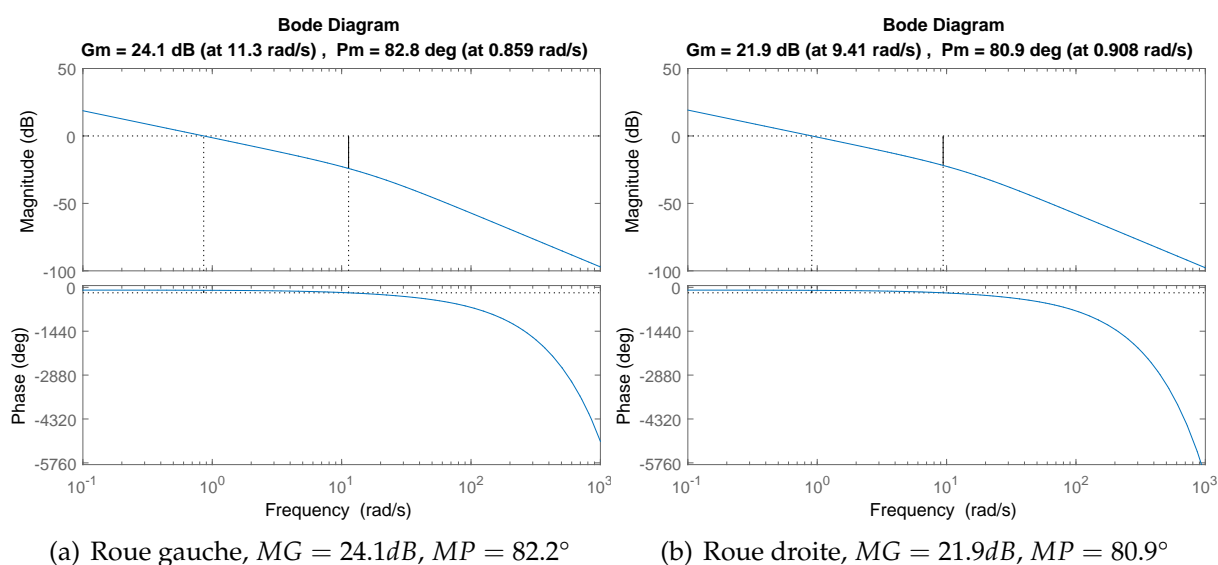


FIGURE 3.5 – Courbes de Bode des roues

### 3.3.2 Simulations

Une simulation du régulateur implémenté ci-dessus a été faite à l'aide de Simulink. Le schéma bloc de la roue droite ainsi que la réponse indicielle associée sont représentées respectivement aux Figures 3.6 et 3.7. Une réponse similaire a été observée pour la roue gauche, cependant elle n'est pas représentée pour éviter tout encombrement visuel.

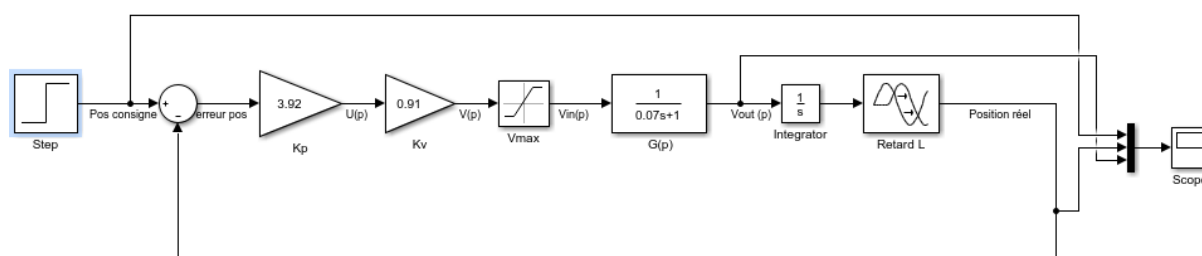


FIGURE 3.6 – Schéma de régulation de la roue droite

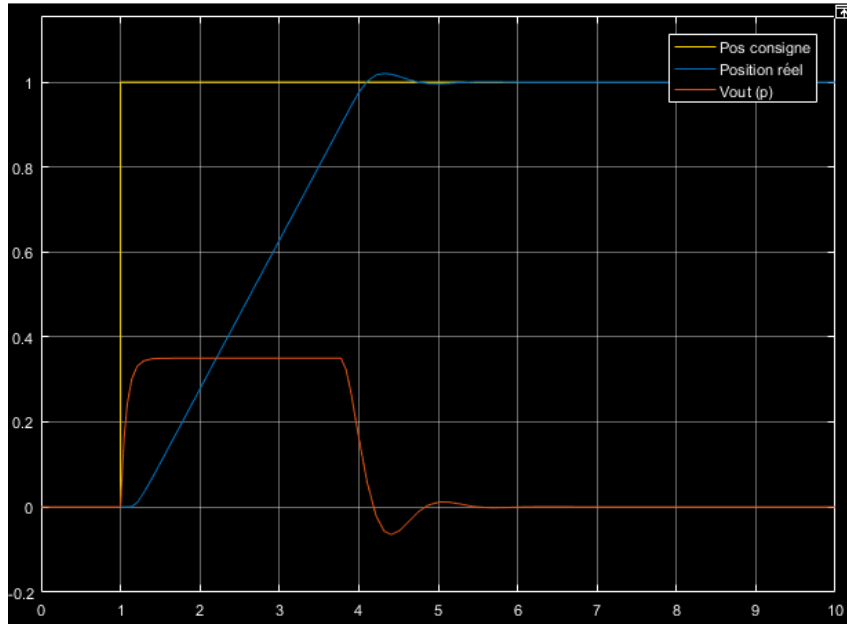


FIGURE 3.7 – Réponse de la roue droite à une consigne de 1m

### 3.3.3 Résultats et corrections

Après avoir implémenté la régulation proportionnelle du robot, il ressort des différentes expériences menées par le groupe que ce dernier n'adopte pas tout à fait une trajectoire rectiligne lorsqu'il roule sur des sols différents de ceux sur lesquels il a été designé. En effet, ce dernier dévie à son départ de sa trajectoire. Ceci s'explique par le fait que la zone morte change avec les différents types de sol.

Pour pallier à ce problème, un régulateur polaire a été implémenté numériquement. Celui-ci évalue à chaque boucle de la régulation la distance parcourue par chacune des roues. Une moyenne de la distance parcourue par les deux roues est ensuite calculée. Une erreur de position est alors calculée de la façon suivante :

$$\begin{cases} errorPosRight = averagePos - |posRight| \\ errorPosLeft = averagePos - |posLeft| \end{cases} \quad (3.4)$$

La vitesse de chacune des roues est ensuite corrigées en additionnant à la consigne de vitesse du régulateur de position un terme proportionnel à l'erreur précédemment calculée. Ainsi, si la roue droite a parcourue une plus grande distance que la roue gauche, celle-ci verra sa vitesse diminuer tandis que la roue gauche augmentera en vitesse. Cet algorithme agit alors comme un régulateur polaire car il corrige à chaque itération (donc toutes les 10 ms) l'erreur sur la distance parcourue par chacune des roues grâce à la commande suivante :

$$velocityMotors(speedRight + p * errorPosRight, speedLeft + p * errorPosLeft) \quad (3.5)$$

Où :

- *velocityMotors* est la commande qui actionne les moteurs droit et gauche à partir d'une vitesse de consigne en entrée pour chacune de ces roues
- *errorPosX* est l'erreur de la roue *X* sur la moyenne des position calculée en 3.4
- *p* est un facteur correctif constant déterminé expérimentalement par essai erreur

Une fois la régulation terminée, le robot est prêt à recevoir une nouvelle commande.

# Chapitre 4

## Conclusion

En conclusion, la conception du robot a été divisée en 2 grandes parties, la communication d'un signal audio et la régulation de position du robot. Il a été possible de réaliser et dimensionner les éléments nécessaires au fonctionnement du robot en respectant le cahier des charges. Ce projet a permis de mettre en pratique chaque point théorique vus dans les différents cours. Il a également montré l'importance de la répartition d'un projet volumineux en plusieurs problèmes indépendants. Ces éléments ont du être étudiés de manière rigoureuse afin d'assurer le bon déroulement du montage final. Il est également notable que le passage à la pratique permet de voir que les phases de test et d'études sont nécessaire pour valider un processus et ainsi préparer les membres de l'équipe à leur fonction d'ingénieur.

# Bibliographie

- [1] Analog Devices. Analog filter wizard. <http://www.analog.com/designtools/en/filterwizard/>. Site web sur internet. Consulté le 15 mars 2017.
- [2] Antoine NONCLERCQ. *Instrumentation*. Presses du Cercle Polytechnique, 2017.
- [3] Christophe LE LANN. Le PID utilisé en régulation de position et/ou de vitesse de moteurs électriques, 2006.
- [4] Mathworks. IIR Filter. [http://nl.mathworks.com/help/hdlfilter/iir-filter.html?s\\_tid=srchtitle](http://nl.mathworks.com/help/hdlfilter/iir-filter.html?s_tid=srchtitle). Site web sur internet. Consulté le 29 avril 2017.
- [5] MathWorks. tf2sos. <https://nl.mathworks.com/help/signal/ref/tf2sos.html>. Site web sur internet. Consulté le 29 avril 2017.
- [6] Michel Kinneart. *Automatique*. Presses du Cercle Polytechnique, 2017.
- [7] Michel Osée. ELECH309 - la fonction fskDetector(), 2014.
- [8] Michel Osée. ELECH309-Etude du déplacement du robot, 2014.
- [9] Michel Osée. ELECH309-Analyse du projet, 2016.
- [10] Michel Osée. ELECH309-Etude de la communication audio, 2016.
- [11] Niel STOREY. *Electronics : a system approach*. Pearson, 2013.
- [12] Texas instruments. uA748c, uA748m GENERAL-PURPOSE OPERATIONAL AMPLIFIERS, October 1990.