Rapport de projet de fin d'étude

Projet "P05" IMA 5 2019/2020

Etude d'un système mécatronique piloté par Arduino

Hugo **Delbroucq**

IMA5

11/09/2019 - 17/12/2019

Tuteur: Florian **Chevalier**

Sommaire

Introduction

Dans le cadre de mon cursus de 5ème année, j'ai pu effectuer ce projet situé au cœur de la spécialité Informatique, Microélectronique et Automatique (IMA) de l'école d'ingénieur de Polytech'Lille. Ce projet est aussi la conclusion de mon cursus post bac qui a débuté en classe préparatoire aux grandes écoles au lycée Pierre d'Ailly de Compiègne. Mon choix pour la filière IMA s'est donc porté sur cette voie d'avenir à la suite de ces deux premières années. Pour donner suite à mes deux premières années, mon choix s'est porté sur la filière « Systèmes Autonomes » de la filière IMA à laquelle je me suis senti plus identifié. De plus, j'ai eu la chance de pouvoir effectuer un stage à Prague ayant pour thème les matériaux piézoélectriques ainsi que les microprocesseurs. Mon choix pour ce projet intitulé « Étude d'un système mécatronique piloté par Arduino » a donc été influé par ces différents éléments, me permettant d'étudier des aspects physique d'un système et en réappliquant mes connaissances mises en pratiques depuis ces dernières années comme la programmation de microprocesseurs, l'analyse énergétique de systèmes mécatroniques et de pouvoir les approfondir en vue de retourner à Prague par la suite en tant qu'ingénieur capteurs et transducteurs chez CTS Corp avec qui j'ai pu avoir un accord de CDI en fin de stage débutant en février.

Remerciements

Pour ce projet, je souhaiterais tout d'abord remercier mon tuteur Mr. Florian Chevalier pour sa disponibilité et son aide apportée tout au long de ce projet. Les discussions que j'ai pu avoir avec lui sur le sujet ont été très instructives et intéressantes. J'aurai aussi voulu remercier Mme Betty Semail qui m'a donné de précieux conseils et corrections concernant la représentation énergétique de mon système étudié. Pour finir je voudrais aussi remercier ma famille pour son soutien ainsi que mes quelques amis encore à Lille.

I. Contexte

1) Gestion du projet.

Ce projet s'est déroulé du 11 septembre 2019 au 17 décembre 2019 soit un peu plus de 3 mois. Le projet devait se réaliser seul par décision du chef de projet avec une soutenance intermédiaire à la demande des étudiants redoublants. En effet il nous a semblé plus sécurisant de pouvoir avoir un retour semi officiel à la moitié de la période de projet en plus de celui de notre tuteur. Mes rencontres avec Mr Chevalier ont été effectuées une fois par semaine, le plus souvent le mercredi après-midi dans son bureau afin de discuter de l'avancement du projet et des différents points à avancer.

	Nom de la tâche	Durée	Début	Fin
1	Complete project execution	14	11.09.19	14.12.2019
2	Ingénierie	10		
3	Documentation	2	11.09.19	29.09.19
4	Liste de matériel	1	25.09.19	02.10.19
5	Commande et réception du matériel	3	25.09.19	08.10.19
6	Liste des tests nécessaires pour la caractérisation	3	30.09.19	08.10.19
7	Design PCB et définition du système	3	25.09.19	12.10.19
8	Code	9	13.11.19	27.11.19
9	Représentation énergétique macroscopique	9	02.10.19	27.10.19
	Pratique	14		
10	Tests du matériel	3	02.10.19	16.10.19
11	Montage du véhicule électrique	2	18.09.19	25.09.19
12	Exécuter la liste de test pour caractériser le véhicule	3	02.10.19	30.10.19
13	Impression et soudure PCB	1	16.10.19	23.10.19
14	Tests électriques	1	23.10.19	30.10.19
15	Installation du module sur le véhicule	1	30.10.19	06.11.19
	Finalisation du projet et tâches annexes	14		
16	Tests en situation réelle	1	06.11.19	13.11.19
17	Utilisation du banc de caractérisation sur un autre système	2	13.11.12	26.11.12
18	Ajout du panneau solaire	1	27.11.19	04.12.19
19	Redaction du wiki	14	11.09.19	11.12.19
20	Schémas	0	24.05.12	29.05.12
21	Rapport et diapo de présentation	2	04.12.19	14.04.19

Illustration 1 : Diagramme de Gant du projet

Comme on peut le voir sur le diagramme illustration 1, la réalisation de la carte a été bien plus longue que prévu, certains points de la carte ont dû être modifiés au cours du temps et de même les commandes prenant du temps il m'a fallu quelques fois commander de moi-même dans le but d'accélérer les délais. En effet ces délais « fixes » sont relativement contraignants. En effet, après ce projet, j'ai pu observé une nécessité importante notamment durant un laps de temps aussi court de gérer ces délais « fixes » J'ai pu avancer sur les autres tâches en parallèle mais pas aussi rapidement qu'il aurait été nécessaire.

2) Objet du projet

À partir d'un assortiment robotique comprenant un motoréducteur et différentes roues et engrenages, le but de ce projet est de proposer un moyen de contrôler le fonctionnement d'un système mécatronique comme un véhicule électrique. Le travail consiste à piloter le moteur électrique à partir d'un Arduino en régulant par exemple la vitesse, à partir d'une consigne et des données des différents capteurs du système. Réaliser une carte électronique de puissance permettant de faire la gestion de différentes sources ici électriques. Faire une étude énergétique sur le fonctionnement de système. Le développement d'un banc de test contrôlé de manière indépendante, avec mesure de vitesse et contrôle du couple appliqué sur les roues du véhicule permettra d'en évaluer complètement les performances.

La base de ce système est donc une voiture à 4 roues entraînés par un moteur à courant continu et contrôlé par un microprocesseur ATMEGA328P. Les deux sources d'alimentation sont une batterie 9V source de tension et une cellule photovoltaïque source de courant. L'environnement de développement utilisé est l'IDE Arduino qui a pour avantage d'être open source et gratuit.

II. Réalisation du projet

1) Analyse du projet

Pour ce projet, nous avons tout d'abord commencé par définir les limites de celui-ci. Le module de test sera composé d'une carte électronique pouvant gérer deux sources d'alimentation différentes. La commande doit pouvoir être réalisée au niveau de la sélection du type d'alimentation que l'on désire avoir (batterie ou cellule photovoltaïque) ainsi qu'un contrôle en vitesse permettant de réguler le moteur à courant continu du robot. On peut donc voir le schéma fonctionnel du robot Illustration 2.

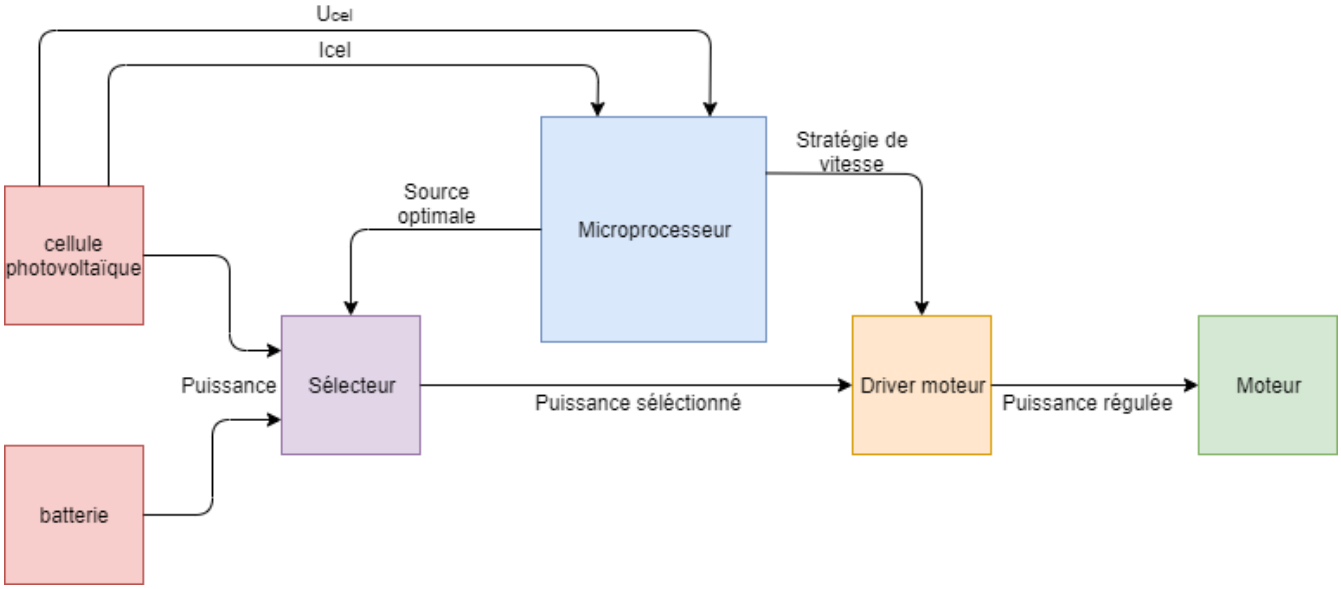

Illustration 2 Schéma fonctionnel simplifié du système

Nous avons décidé de partir avec deux différents types d'alimentation afin de pouvoir réaliser. L'intérêt est de pouvoir utiliser une énergie plus propre que l'énergie électrique fournie par la batterie lorsqu'on en a la possibilité ce qui pourra rajouter un intérêt au projet par la suite. Le microprocesseur quant à lui est alimenté par la batterie comme tous les autres composants actifs de la carte. La cellule photovoltaïque est utilisée dans le but d'économiser l'énergie de la batterie lorsque c'est possible.

En résumé, les besoins du projet sont :

- Utiliser deux sources d'alimentation
- Définir une stratégie de commande
- Définir des méthodes de calcul des différentes variables
- Réaliser un banc pouvant permettre l'évaluation de ces données.

2) Réalisation de la partie électronique

Par suite de la première réunion le 11 septembre 2019, nous avons été avertis de commander nos composants avant le début du mois d'octobre afin d'être sûr d'être livré dans les temps. J'ai donc directement commencé par définir les limites du sujet précisé plus haut pour réaliser la partie électronique du système.

a) Etude du moteur

Pour commencer, je me suis intéressé au moteur qui allait être alimenté par la carte et donc se situant au cœur du projet. Les références du moteur sont données par l'Illustration 3

MODEL	VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY						STALL TORQUE	
	OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	EFF		
			R.P.M.	A	R.P.M.	A	oz-in	g-cm	W	%	oz-in	g-cm
RE-140	1.5-3	1.5v Constant	8200	0.190	6250	0.62	0.089	6.4	0.400	44.2	0.375	27.0
RE-140	1.5-3	3v Constant	15300	0.26	12200	1.04	1.444	13	2.51	52.3	0.750	64.0
RE-140/1	3-9	6v Constant	9200	0.066	7071	0.219	0.11	8.1	0.589	44.79	0.48	35.1

Stall Current RE-140 at 1.5v = 1.6A RE-140 at 3v 3.0A RE-140/1 at 6v 0.7A

Illustration 3 : Références moteur RE-140

D'après ce tableau de valeurs données par le constructeur, on apprend que notre moteur à courant continue peut-être commandé par une tension de 1.5V à 3V. Sa vitesse maximale donnée est de 15 300 tours par minutes à 3V pour un courant de 0.26A. De plus, On connait les équations typiques d'un moteur à courant continu qui sont de la forme suivante :

$$E = K_\varphi * \omega \qquad (1)$$

E la force électromotrice en V

K_φ une constante (ici le flux est constant) en V.s/rad

ω la pulsation de rotation du moteur en rad/s

$$C = K_\varphi * I \qquad (2)$$

C le couple du moteur en N.m

K_φ une constante (ici le flux est constant) en N.m/A

I le courant à l'induit du moteur en A

En appliquant nos valeurs à l'équation (1), on obtient une estimation de notre K_φ

$$K_\varphi = 0.0018 \text{ V.s/rad}$$

Après avoir examiné ces valeurs données par le constructeur, j'ai décidé de travailler avec un courant maximal de 1A tout au long de mes essais. Cela me permettant de travailler sur les points d'efficacités maximaux du moteur.

b) Etude de la cellule photovoltaïque

La cellule photovoltaïque est une source de courant correspondant à notre source « propre » qu'il faudra utiliser lorsque les conditions le permettent.

La cellule reçue avec le kit de départ est une cellule composée de silicium mono cristallin reconnaissable par sa couleur bleu foncé ainsi que sa surface sans cristaux. Ce type de cellule a pour avantage de posséder un bon rendement de l'ordre de 20%. Il est à noter que plus la température est basse et meilleur sera ce rendement. Cette cellule peut délivrer une puissance de 3W (5,5V pour 0.540A).

Afin de pouvoir l'utiliser dans notre circuit, il va nous falloir un régulateur de type MPPT afin d'avoir un rendement maximal et transformer cette source de courant en source de tension.) à l'aide d'un élément capacitif. Le MPPT a pour objectif de réaliser une dichotomie de la puissance délivrée par la cellule en fonction de sa tension dans le but de rester constamment au maximum de puissance que celle-ci peut délivrer. Pour se faire, il agit exactement comme un hacheur avec un temps de cycle variable qui lui permet de faire varier la tension reçue. Il analyse ensuite le courant et avec un rapide calcul, compare si la puissance est plus grande ou non que celle calculé à l'instant t-1 Cela permet d'avoir un rendement maximal en temps réel délivré par notre système. On peut voir la courbe sur laquelle le MPPT se tourne pour effectuer sa dichotomie sur l'illustration 4

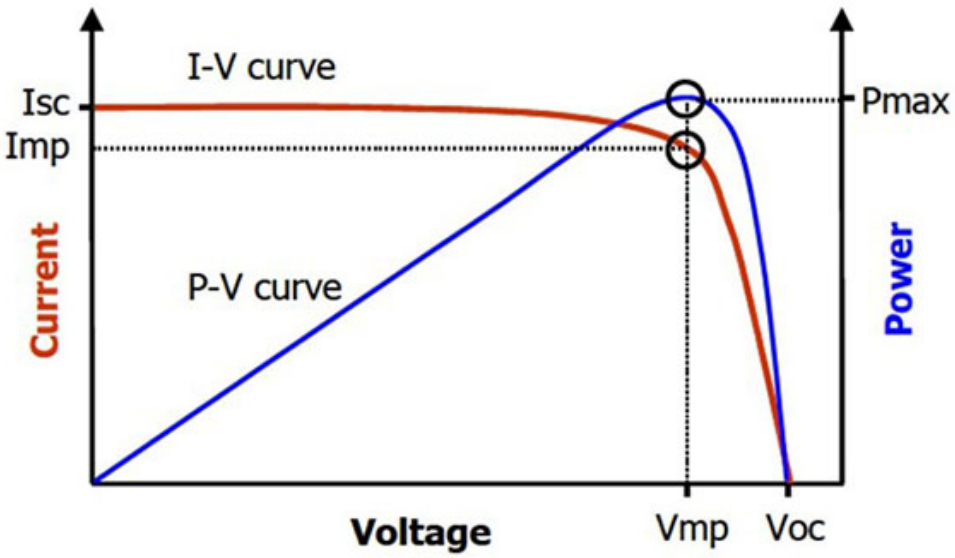

Illustration 4 : Caractéristique cellule photovoltaïque

En plus de cela, j'ai aussi fait des tests afin de rajouter des capteurs de tension et courant sur notre carte. Le capteur de tension est en fait un pont de tension relié à l'une des

broches de l'ATmega, il permet de connaitre la tension aux
bornes de la cellule. J'ai pu réaliser des tests sur LTSpice afin de vérifier la conception comme on peut le voir sur l'illustration 5.

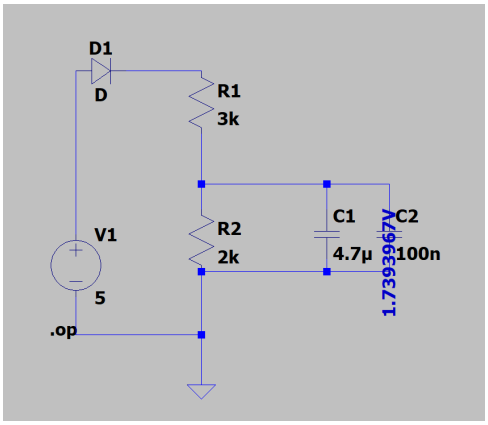

Illustration 5 : Test capteur de tension

J'ai choisi d'effectuer un pont diviseur de tension car la tension maximale aux bornes de la cellule peut être supérieur de 3V aux tensions normales aux bornes des broches analogiques de l'Atmega. On a ici le résultat suivant :

$$U_2 = V1 * \left(\frac{R2}{R1+R2}\right) \quad (3)$$

Au maximum, Ce pont peut donc « voir » une tension maximale de 8V ce qui est amplement suffisant pour notre cellule.

$$U_2 = V1 * \left(\frac{R2}{R1+R2}\right) = 8 * \left(\frac{3000}{5000}\right) = 5$$

J'ai ensuite voulu réaliser un capteur de courant à l'aide d'une résistance. Ce capteur a aussi été simulé sur LTSpice et est théoriquement fonctionnel comme on peut le voir sur l'illustration 6

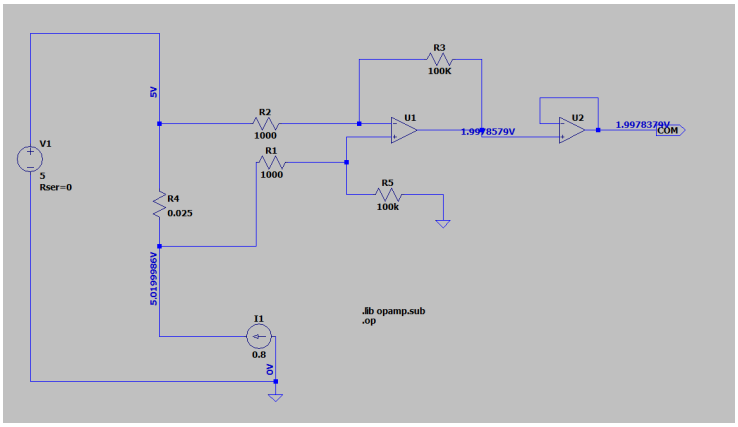

Illustration 6 : Test capteur de courant

Ce capteur a été démontré comme non utile lors de la séance en demi-soutenance du fait que la source de courant représentant le moteur et permettant le calcul de courant est coupé du système par le sélecteur. Néanmoins j'ai décidé de le conserver pour des cas où l'on test avec le sélecteur ouvert par exemple pour déterminer le courant sur un intervalle de temps réduit ou afin de réaliser des tests pourquoi pas et obtenir la relation Tension/Courant de la cellule (avec le MPPT) et ainsi déterminer le courant en fonction de la tension lue. De plus il est plus simple de couper une piste que de la rajouter par la suite sur une carte.

L'équation liée à ce schéma est la suivante :

$$V_{out} = (V1 - V2) * (\frac{R1}{R2}) \quad (4)$$

Avec (V1-V2) le potentiel au niveau de la résistance de mesure de courant.

c)Réalisation de la carte

J'ai ensuite poursuivi la réalisation d'un schéma sous Eagle en utilisant un Atmega328p comme processeur étant donné que le sujet portait sur un véhicule piloté par Arduino et que notre système ne nécessite pas de grosse force de calcul et ne nécessite donc qu'un 328p. Le seul détail inhabituel sur le reste de la carte concerne le quartz que j'ai reçu à la place du quartz classique car déjà disponible chez Mr Flamand qui est un oscillateur à quartz. Après réflexion j'ai pris la décision de le laisser car bien que plus encombrant il permet une isolation du signal de l'horloge et je pense pouvoir ainsi rendre ce signal plus stable.

La carte une fois réalisée est alors comme ci-dessous :

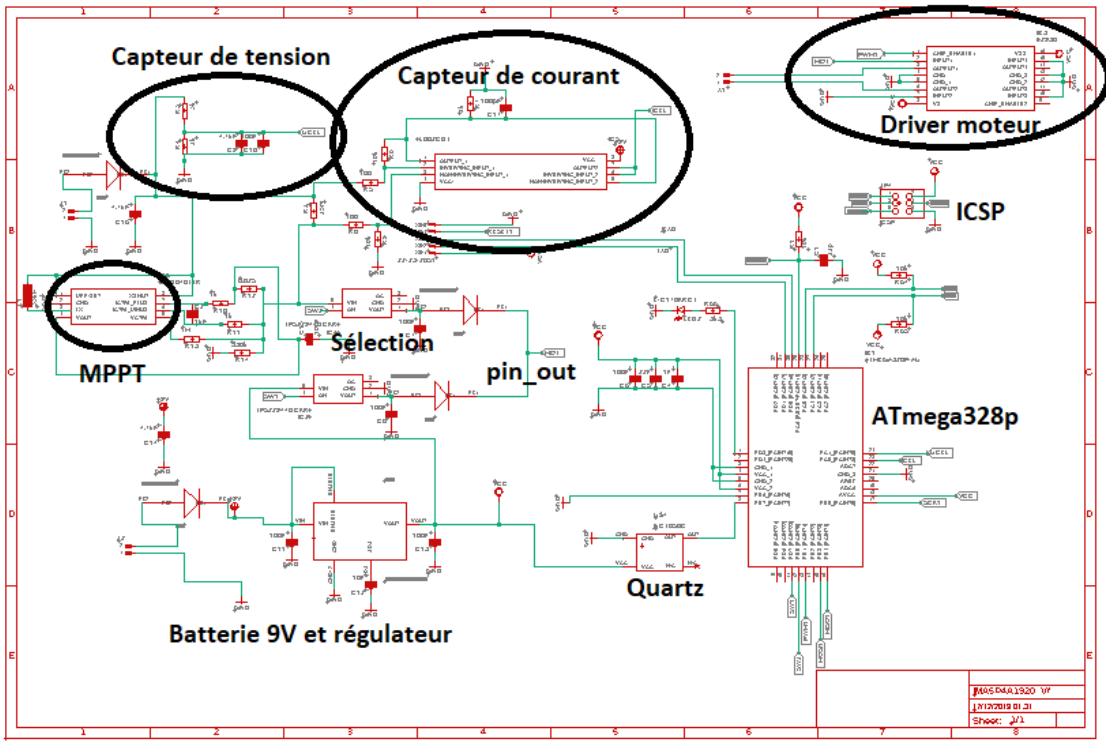

Illustration 7 : Schématique de la carte sous Eagle

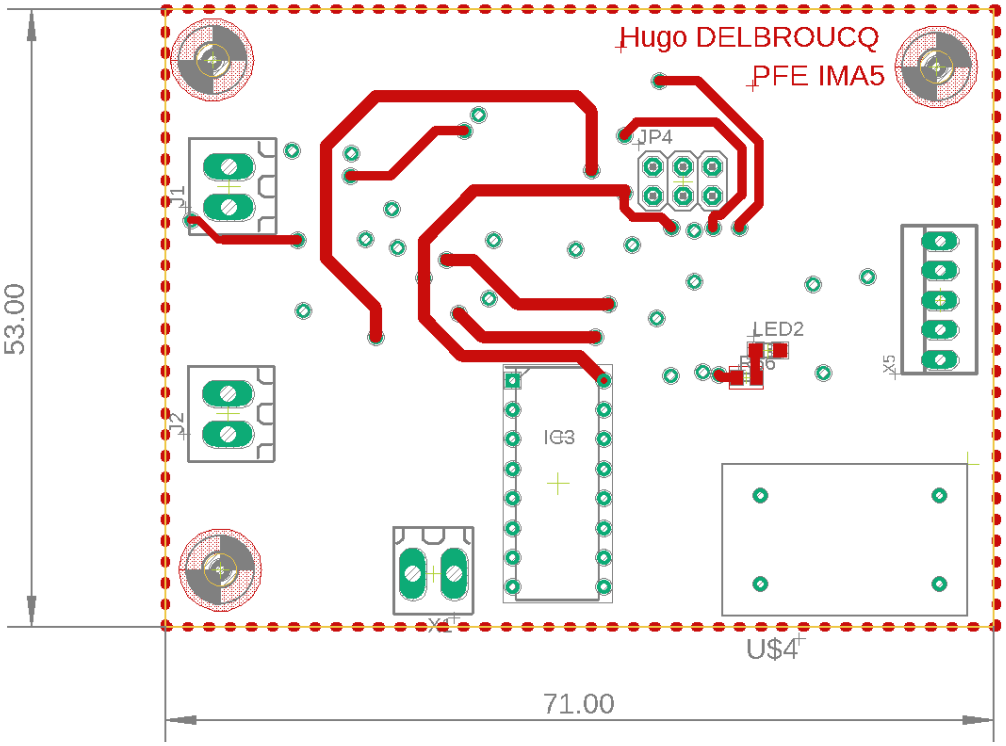

Illustration 9 : PCB Top side

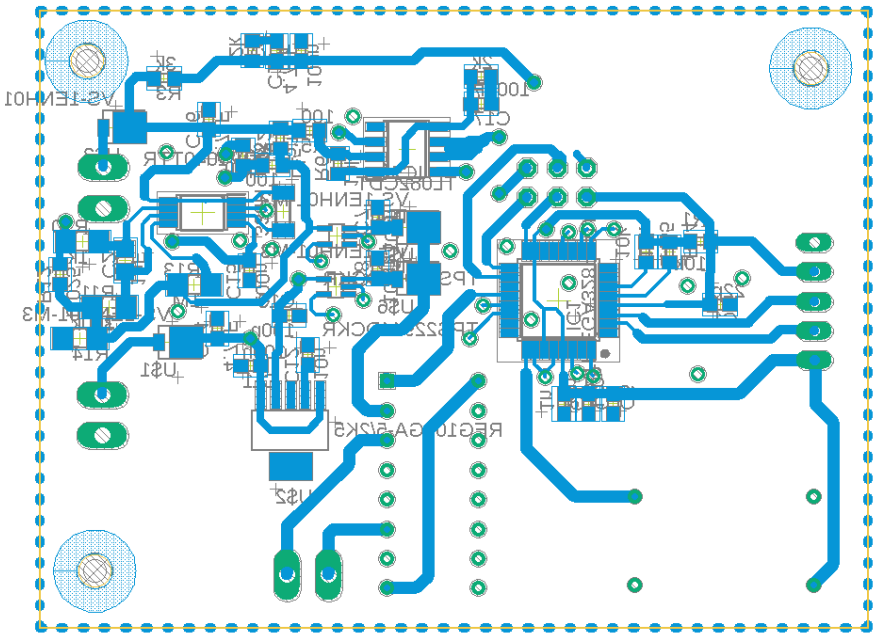

Illustration 8 : PCB Bottom side

Une fois l'impression réalisée j'ai pu commencer à souder à l'aide d'un fer à souder et de flux (sans réfléchir j'ai placé une via sous un composant, pour passer des composants au four il est nécessaire que rien ne soit soudé au préalable sur la carte. J'ai donc tout soudé au fer à souder ce qui m'a aussi permis de contrôler la continuité au fur et à mesure de la phase de soudage.

Par la suite, j'ai tenté de burn l'Atmega à l'aide de la communication ICSP, il s'est avérer que le microprocesseur était soudé à l'envers (une via était juste dans le coin de l'Atmega et sur un moment de mégarde j'ai dû prendre cette via comme le point de référence. J'ai donc changé l'ATmega et retenté de burn le bootloader sur la carte.

De même, impossible de réaliser le bootloading de la carte cette fois ci. Après avoir vérifié plusieurs fois la continuité de la carte notamment aux bornes de l'ATmega dont les pistes avaient été mal découpées la première fois (il a fallu gratter le cuivre à certains endroits où le plan de masse et certaines via étaient confondues) ainsi que les potentiels courts circuits, j'en suis venu à l'hypothèse que l'oscillateur à quartz pourrait être l'issue du problème. En réétudiant les connections de la carte, j'ai remplacé la connexion de l'horloge à xTAL2 par xTAL1 comme recommandé dans la datasheet du microprocesseur. L'information trouvée en ligne sur les différences entre modes de communications devait être pour un processeur particulier. J'ai pu réussir à téléverser un boot loader sur l'Arduino avant les vacances ce qui m'a permis de confirmer la mauvaise connexion de l'horloge au microprocesseur. Une fois rentré chez moi, j'ai néanmoins découvert que la connexion faite avec un fil de cuivre c'était légèrement déplacée et entrait en contact avec xTAL2 en même temps que xTAL1. Il m'a donc fallu réeffectuer la soudure et ensuite retester le bootloader cette fois ci avec un programme (http://www.gammon.com.au/forum/?id=11635), ce programme a pour avantage de permettre la vérification du boot et j'ai donc pu confirmer que l'impossibilité de téléverser des programmes sur la carte était dus au fait que le boot loader ne s'était pas installé avec succès. En ressoudant le fil de cuivre et en réeffectuant le boot de mon PCB, j'ai pu obtenir les résultats suivants :

```
Atmega chip programmer.
Written by Nick r ...
xFD
Lock byte = 0xCF
Clock calibration = 0xA5
Type 'L' to use Lilypad (8 MHz) loader, or 'U' for Uno (16 MHz) loader
...

Atmega chip programmer.
Written by Nick Gammon.
Version 1.38
Compiled on Dec 24 2019 at 15:35:07 with Arduino IDE 10809.
Attempting to enter ICSP programming mode ...
Entered programming mode OK.
Signature = 0x1E 0x95 0x0F
Processor = ATmega328P
Flash memory size = 32768 bytes.
LFuse = 0xFF
HFuse = 0xDE
EFuse = 0xFD
Lock byte = 0xCF
Clock calibration = 0xA5
Type 'L' to use Lilypad (8 MHz) loader, or 'U' for Uno (16 MHz) loader
...
Using Uno Optiboot 16 MHz loader.
Bootloader address = 0x7E00
Bootloader length = 512 bytes.
```

```
Type 'Q' to quit, 'V' to verify, or 'G' to program the chip with the
bootloader ...
Verifying ...
No errors found.
Done.
Programming mode off.
Type 'C' when ready to continue with another chip ...
```

On peut voir ucu que le boot a pu se faire avec succès.

Ensuite il m'a fallu connecter les broches de communication entre ma carte et l'Arduino UNO :

RxD→RxD; TxT→TxT ; VCC→VCC ; GND→GND ; RESET→RESET

Cette manipulation a pour but de se servir du 16u2 de l'Arduino afin de réaliser la communication USB ce qui nous éviter d'emporter trop de composants supplémentaires sur la carte.

Lors de la phase de tests, les programmes se téléchargeaient sans soucis mais il m'était impossible d'obtenir une tension au niveau du driver moteur. En remontant à la source, j'ai découvert que le problème venait certainement de mes deux sélecteurs de puissance qui permettent le changement de type d'énergie utilisé. En essayant de corriger le problème en isolant les broches (notamment la broche d'erreur qui touchait la masse du fait de la petite taille du composant), j'ai pu faire fonctionner la chaîne liée à la cellule photovoltaïque. La partie batterie quant à elle, le sélecteur ne renvoyant pas de signal d'erreur (signalant un surplus de courant ou un court-circuit), j'en ai conclu qu'il était détruit et en ai donc recommandé un autre. En changeant de composant et en plaçant un morceau de scotch sous la broche d'erreur (pour éviter le contact entre la masse et la broche) j'ai pu avec plaisir constater que l'ensemble de la carte fonctionnait.

c) Situation du projet

Actuellement nous avons donc une gestion de l'alimentation fonctionnelle sur la carte. Pour le moment, le programme consiste à vérifier que la puissance fournie par la cellule photovoltaïque n'est pas trop faible pour alimenter le moteur (la valeur a été déterminé par tests successifs) et l'on essaie de fixer la tension maximale acceptable par le moteur (3V) à son entrée. D'une part à l'aide du capteur de tension pour la cellule photovoltaïque et on réajuste si dépassement à 3V et d'autre part à l'aide d'un proportionnel pour la batterie de 9V et délivrant du 5V au système. Il est aussi possible de lire la tension et le courant fournis par la cellule photovoltaïque grâce à l'Arduino ce qui permet d'avoir une bonne estimation de la puissance fournie par la cellule photovoltaïque.

d) Programme Arduino

Le programme Arduino implémenté sous l'IDE Arduino est le suivant :

```
/*
  POLYTECH LILLE
```

```
  IMA 5 PROJECT 5 2019/2020
  DELBROUCQ HUGO
  TARGET : Réalisation d'un banc de caractérisation d'un système
mécatronique de type véhicule électrique
*/
#define BAUD_RATE 9600  //Serial communication baudrate
#define MIN_VOL  77     //1.5V
#define MAX_VOL 153       //3V max voltage for motor
//Voltage command
const int pin_PWM_CTRL_MOTOR = 9; //control du moteur
const int pin_DO_SWITCH_BATT = 7; //switch batterie (1 en marche, 0 en
arret
const int pin_DO_SWITCH_CELL = 8; //switch cellue photo ( 1 en marche, 0
en arret
const int pin_AI_I_CELL = A0; //capteur de courant de la cellule
const int pin_AI_U_CELL = A1; //capteur de tension de la cellule
const int pin_DO_LED = 3;
int PWM_CTRL_MOTOR;   //send a PWM signal to the motor driver
int DO_SWITCH_BATT;   //If 1, battery is selected
int DO_SWITCH_CELL;   //If 1, solar cell is selected
int DO_LED = HIGH;         //1 light is on
extern unsigned int AI_I_CELL;   //current value from the solar cell
extern unsigned int AI_U_CELL;   //voltage value from the solar cell


void setup() {
  Serial.begin(BAUD_RATE);
  pinMode(pin_PWM_CTRL_MOTOR, OUTPUT);
  pinMode(pin_PWM_CTRL_MOTOR, OUTPUT);
  pinMode(pin_DO_SWITCH_BATT, OUTPUT);
  pinMode(pin_DO_SWITCH_CELL, OUTPUT);
  pinMode(pin_DO_LED, OUTPUT);
  pinMode(pin_AI_I_CELL, INPUT);
  pinMode(pin_AI_U_CELL, INPUT);
  DO_SWITCH_BATT = 0;
  DO_SWITCH_CELL = 1;
  digitalWrite(pin_DO_LED, DO_LED);
  digitalWrite(pin_DO_SWITCH_BATT, DO_SWITCH_BATT);
  digitalWrite(pin_DO_SWITCH_CELL, DO_SWITCH_CELL);
  analogWrite(pin_PWM_CTRL_MOTOR, MAX_VOL);

}

void loop() {

  Serial.print("Tension cellule : ");
  float U_cell = (analogRead(pin_AI_U_CELL) *  5.0 / 1024.0) * (5.0 /
2.0) ; //produit en croix pour obtenir la tension en volt multiplié par
le pont diviseur de tension
  Serial.println(U_cell);
  Serial.print("Courant cellule : ");
  float I_cell = (analogRead(pin_AI_I_CELL) *  5.0 / 1024.0) / (200 *
0.025) ; //produit en croix pour obtenir le courant voir rapport p8
```

```
  Serial.println(I_cell);
  float P_cell = U_cell*I_cell;
  if (U_cell > 1.5 && P_cell > 3) {//si la cellule a suffisament de
puissance pour fonctionner, on passe par cette boucle
    DO_SWITCH_BATT = 0;
    DO_SWITCH_CELL = 1;
    digitalWrite(pin_DO_SWITCH_BATT, DO_SWITCH_BATT);
    digitalWrite(pin_DO_SWITCH_CELL, DO_SWITCH_CELL);
    analogWrite(pin_PWM_CTRL_MOTOR, int(3 * 256 / U_cell)); //place une
commande de tension de 3V au driver, peut être modifié si commande en
vitesse
  }
  else {//chaîne de puissance batterie on considère 5V fixe et on fixe 3V
sur le moteur
    DO_SWITCH_CELL = 0;
    DO_SWITCH_BATT = 1;
    digitalWrite(pin_DO_SWITCH_BATT, DO_SWITCH_BATT);
    digitalWrite(pin_DO_SWITCH_CELL, DO_SWITCH_CELL);
    analogWrite(pin_PWM_CTRL_MOTOR, MAX_VOL);
  }
  Serial.print(DO_SWITCH_BATT);
}
```

Ce programme est relativement succin et consiste à prendre les valeurs de tension et de courant de la cellule photovoltaïque afin de pouvoir décider si sa puissance est suffisante pour notre système. Si oui on applique un rapport cyclique au driver permettant d'envoyer 3V au moteur en prenant la valeur de tension de la cellule comme référence, sinon on utilise un rapport cyclique déjà calibré sur 3V en utilisant la batterie comme sa source est-elle considérée de 5V constant (avant la régulation en vitesse).

Pour trouver la valeur de U_cell, la valeur est lue sur 8 bits soit 1024 en décimal, 1023 signifiants 5V on réalise un produit en croix pour avoir la tension analogique en V. Puis on ajoute le 5/2 du pont diviseur de tension.

e) Améliorations futures

J'ai pu déduire une liste d'améliorations à la suite de ce premier prototype :

- D'écarter les via des composants afin de pouvoir souder celles-ci loin des autres zones « à risque » du système et d'ainsi créer des courts circuits.
- Retirer la via sous le microprocesseur afin de souder au four les composants CMS
- Ajouter deux ports afin d'ajouter la régulation en vitesse du système à l'aide d'un codeur incrémental ou de roues codeuses ou tout autre capteur de vitesse.
- On peut changer l'horloge pour une horloge plus classique (quartz et deux capacités de charge) afin d'avoir moins d'encombrement sur la carte.
- Sortir toutes les broches de l'ATMEGA328p afin de pouvoir ajouter des modules facilement en cas de modification.

- Faire le mode marche arrière en connectant
  la pin 7 du driver moteur à la même broche que la pin 2 et ajouter un contrôle afin de sélectionner la puissance (ajouter deux sélecteurs comme pour les chaînes d'alim afin de n'avoir qu'une seule entrée alimentée en tout temps.

3) Etude de la représentation énergétique macroscopique du système

a) La chaîne directe

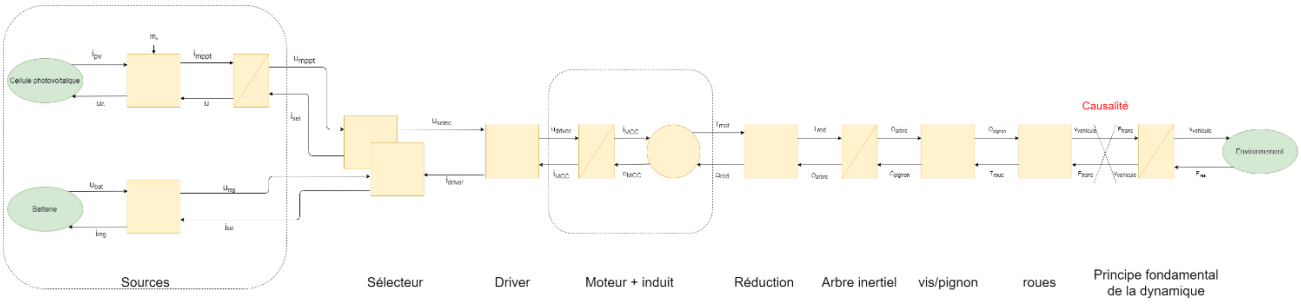

Illustration 10:REM sous sa première forme

La REM ci-dessus est la première version sans la modification de causalité du dernier bloc. Nous allons donc tout d'abord expliciter les équations du système sous cette forme puis avec l'ajustement des blocs.

1) la source de courant est caractérisée par l'équation

$$I_{PV} = I_0 * (1 - e^{-\frac{q(V_{CC}-U_c)}{kT}})$$

Avec I_0le courant de saturation ($I_0 = 0.8\ A$)

q la charge d'un électron ($q = 1.6022 * 10^{-19} C$)

k la constante de Boltzmann ($k = 1.3806 * 10^{-23} J.K^{-1}$)

T la température en K ($T = 291K$)

V_{CC}la tension de court-circuit valant 8.2V

Cette équation permet d'exprimer la relation entre la tension lue aux bornes du MPPT ainsi que le courant généré par la cellule.

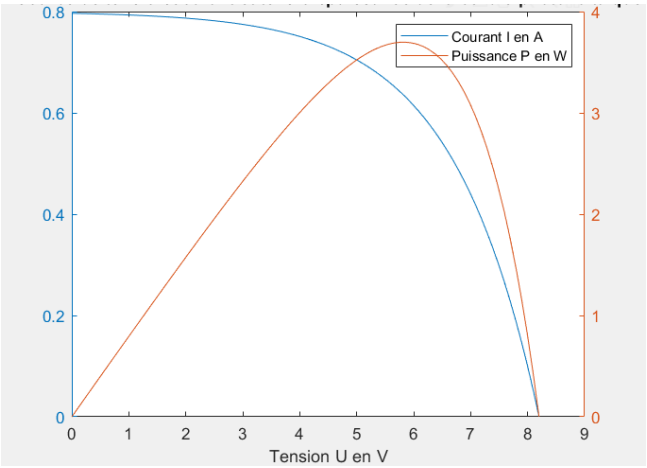

Illustration 11: Courbes de courant et puissance en fonction du temps

2) Le régulateur MPPT fonctionne comme un hacheur, son rapport cyclique va varier afin de faire varier la tension aux bornes de la cellule dans le but de trouver le maximum de puissance délivrée par celle-ci.

$I_{mppt} = I_{PV} * mh$ et $U_c = U_{mppt} * mh$

Avec mh le rapport cyclique du « hacheur » du MPPT déterminé par une stratégie de type dichotomie

3) On a ici la capacité située aux bornes du module MPPT qui va nous permettre de transformer la causalité de notre source (on aura donc une tension en sortie plutôt qu'un courant.

$$U_{mppt} = \frac{1}{C}\int I_{mppt} - I_{selecteur}$$

Avec C la capacité placée aux bornes du module MPPT qui a pour valeur 1uF

Pour la seconde source de tension, on considère qu'elle délivre une source de tension fixe de 9V il n'est donc pas nécessaire de simuler l'équation liant courant/tension pour notre exemple, simplement

$$U_{batt} = constante$$

Le régulateur de tension est ensuite exprimé. Son rapport va correspondre au rapport $\frac{U_{reg}}{U_{batts}}$.

On a alors les équations du régulateur suivantes :

$$U_{reg} = U_{batt} * reg_{ubat} \quad et \quad I_{batt} = I_{selecteur2} * reg_{ubat}$$

On passe maintenant au sélecteur, ce bloc permet simplement de sélectionner le type d'alimentation que l'on souhaite et apparait dans un bloc de couplage.

Nous avons donc 3 équations pour associer les 7 paramètres présents sur ce bloc :

$$U_{selecteur} = U_{mppt} * (1 - m_{selecteur}) + U_{batt} * m_{selecteur}$$

$$I_{sel1} = I_{driver} * (1 - m_{selecteur})$$

$$I_{sel2} = I_{driver} * m_{selecteur}$$

Avec $m_{selecteur}$notre paramètre de sélection avec $m_{selecteur} \in [0\,;1]$

Pour le driver, son but va être d'adapter la tension d'entrée au moteur à l'aide d'un rapport cyclique. Nous allons pour le moment fixer la valeur de tension voulue à 3V ($1.26 m.s^{-1}\ en\ sortie$). Dans le futur cette valeur sera déterminée par le retour de la chaîne de commande et la consigne donnée.

Le rapport cyclique lui aura comme valeur $\frac{U_{ref}}{U_{selecteur}}$

Le moteur est un moteur à courant continu, ça forme nous est bien connue mais nous allons la remettre en forme ici :

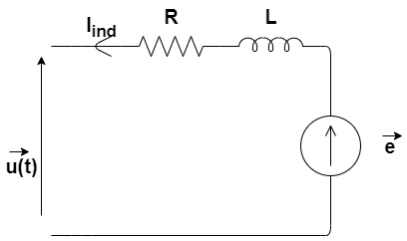

Illustration 12: partie électrique du moteur

$$u(t) = e(t) + Ri(t) + L\frac{d}{dt}i(t)$$

$$D'où\ en\ Laplace\ on\ obtient :$$

$$\frac{i(t)}{u(t) - e(t)} = \frac{\frac{1}{R}}{1 + \frac{L}{R}p} = \frac{\boldsymbol{K_1}}{\boldsymbol{1 + \tau_1 p}}$$

Par mesure, on obtient R =2.2Ω ce qui est tout à fait logique lorsque l'on observe les dimensions du moteur (plus un moteur est petit et plus sa résistance d'induit est grande.) Pour ce qui est de l'inductance, je me suis inspiré d'un des TP fait l'an précédent afin de sélectionner une valeur proportionnelle à la résistance présente dans le système.

La partie convertisseur du bloc moteur elle se traduit sous la forme de deux équations :

$$T_{MCC} = I_{ind} * K_{\varphi} \qquad et \qquad U_{MCC} = \Omega_{MCC} * K_{\varphi}$$

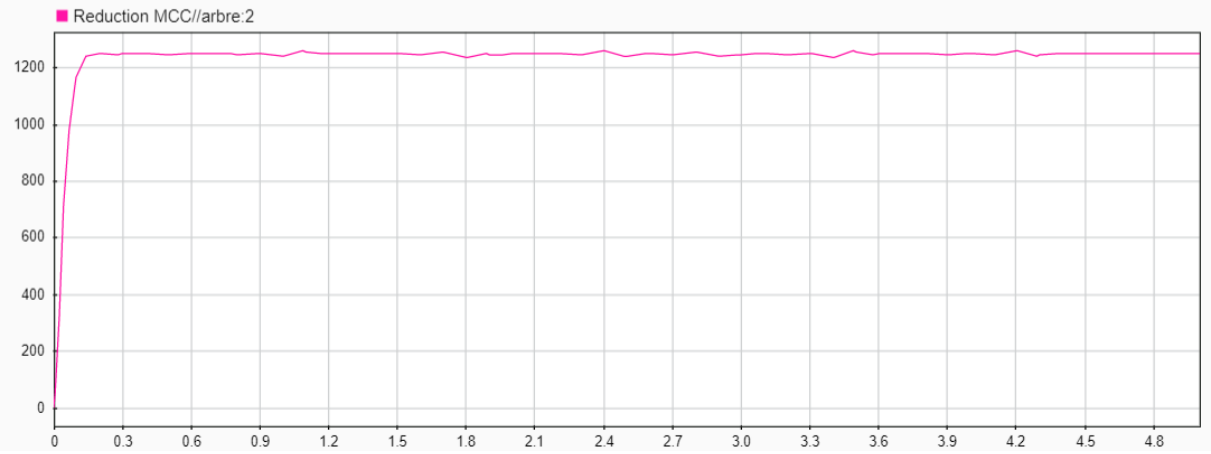

Illustration 13: Courbe de vitesse du moteur en rad/s en fonction du temps

Avec K_{φ} le rapport lié au flux induit du moteur (ici constant)

On a ensuite le rapport de réduction appliqué directement en sortie du moteur :

$$T_{red} = T_{MCC} * rap_{reduction} \qquad et \qquad \Omega_{MCC} = \Omega * rap_{reduction}$$

Ce rapport nous permet de réduire la vitesse avant de passer par l'arbre inertiel du moteur. Il est égal à $\left(\frac{12}{48}\right)^2$.

L'arbre inertiel qui lui va transférer l'énergie fournit par le moteur est traduit par la relation suivante :

$$J\frac{d}{dt}\Omega_{arb} + f * \Omega_{arb} = T_{red} - T_{pignon}$$

Avec J le moment d'inertie de l'arbre ($1.2 * 10^{-3}\ kg.m^2$)

f les forces de frottement au niveau de l'arbre (0.05 dans notre étude)

Sous transformée de Laplace :

$$\frac{\Omega_{arb}}{T_{red} - T_{pignon}} = \frac{\frac{1}{f}}{1 + \frac{J}{f}p} = \frac{\boldsymbol{K_2}}{\boldsymbol{1 + \tau_2 p}}$$

À la suite de cela nous avons donc une relation vis/pignon qui se traduit elle aussi par un simple rapport de réduction :

$$\Omega_{pignon} = \Omega_{arb} * k_{vis/pignon} \; et \; T_{pignon} = \eta * T_{roue} * k_{vis/pignon}$$

Avec $k_{vis/pignon} = 10/16$le rapport de réduction lié au système de vis/pignon (rapport du nombre de dents sur la roue menantes sur le nombre de dents sur la roue menée)

η un coefficient permettant de simuler les pertes de freinage dues au contact sol/roues (0.95 ici)

À la suite de cela, il est temps de simuler la roue qui va transformer la vitesse de rotation en en vitesse de translation, on peut symboliser ce phénomène par l'équation suivante :

$$v_{vehicule} = \Omega_{pignon} * R_{roue} \; et \; T_{roue} = F_{trans} * R_{roue}$$

Avec $R_{roue} = 0.0175\ m$ le rayon de la roue.

Avec un rapport de réduction total de $\left(\frac{12}{48}\right)^2 * (\frac{10}{16})$, On a environ :

$$V_{vehicule} = \left(\frac{12}{48}\right)^2 * \left(\frac{10}{16}\right) * \Omega_{arbre} * R_{roue} = 0.87 m.s^{-1}$$

Enfin il nous reste à simuler le principe fondamental de la dynamique à notre système :

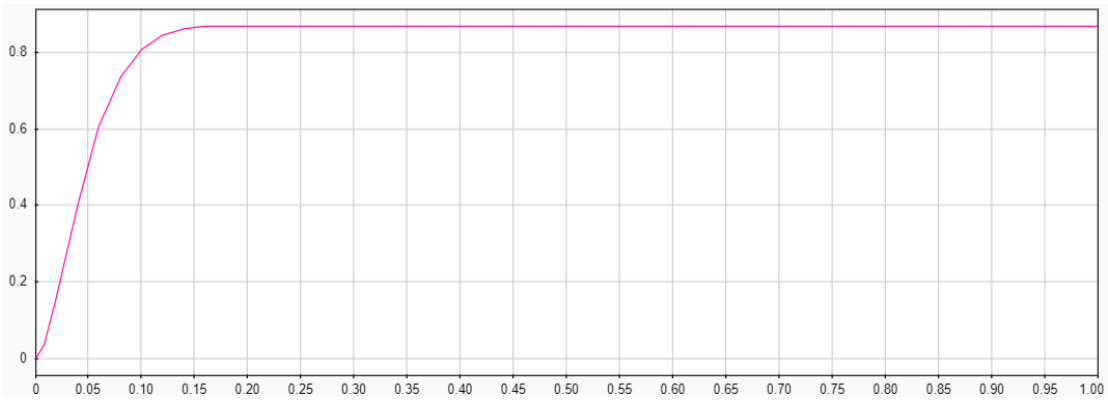

Illustration 14:Courbe de vitesse du véhicule en m/s en fonction du temps sous 3V

$$v_{vehicule} = \frac{1}{M} \int F_{trans} - F_{res}$$

Avec $M = 0.467kg$ la masse du véhicule

La relation au sein de l'environnement elle est définie comme ci-dessous :

$$F_{res} = F_{res_{pente}} + F_{res_{aéro}} + F_{res_{frottements}}$$

Avec $F_{res_{pente}} = M * g * grad(pente)$

$F_{res_{aéro}} = rho.Cx.A_{contact}.(V_{vehicule} + V_{vent}))/2.$ (Négligeable)

Rho la densité de l'aire, Cx le coefficient de traînée, A l'aire de contact, $V_{vehicule}$ la vitesse du véhicule et V_{vent} la vitesse du vent.

$F_{res_{frottements}} = k * M * g$

K le coefficient de frottement à la route, M la masse du véhicule, g la pesanteur

Maintenant, il nous faut donc remodeler notre REM sous la forme suivante afin de ne plus avoir de soucis de causalité. Tout d'abord il nous faut assembler les blocs « Roues » et « système vis/écrou » le but et de réexprimer nos fonctions précédentes en fonction de $v_{vehicule}$ et T_{red2} dans une équation puis F_{res} et $v_{vehicule}$ dans une seconde.

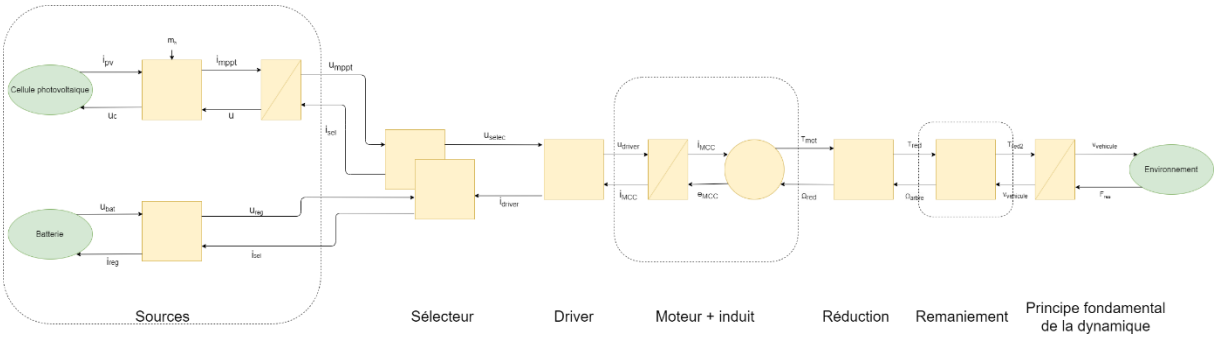

Illustration 15: REM sous sa forme corrigée

On note A le bloc réducteur vis/pignon et B le bloc roues pour obtenir le rapport combiné :

$$AB = k_{vis/_{pignon}} * R_{roue}$$

On a donc

$$C_{pignon} = \eta * AB * F_{trans} \quad et\ v_{vehicule} = AB * \Omega_{arb}$$

On réécrit l'équation liant les vitesses sous la forme suivante :

$$\Omega_{arb} = \frac{1}{AB} * v_{vehicule}\ et\ F_{trans} = \frac{1}{\eta * AB} * T_{pignon}$$

Cela nous permet alors de réinjecter ces formules dans l'arbre inertiel :

$$J\frac{d}{dt}\Omega_{arb} + f * \Omega_{arb} = T_{red} - T_{pignon}$$

$$\frac{J}{AB} * \frac{d}{dt}v_{vehicule} + \frac{f}{AB}v_{vehicule} = T_{red} - AB * \eta * F_{trans}$$

$$\frac{J}{\eta AB^2} * \frac{d}{dt}v_{vehicule} + \frac{f}{\eta AB^2}v_{vehicule} = \frac{1}{\eta AB}T_{red} - F_{trans}$$

De plus, on modifie notre PFD

$$v_{vehicule} = \frac{1}{M}\int F_{trans} - F_{res}$$

$$F_{trans} = M * \frac{d}{dt}v_{vehicule} + F_{res}$$

On réinjecte F_{trans} dans l'équation :

$$\frac{J}{\eta AB^2} * \frac{d}{dt}v_{vehicule} + \frac{f}{\eta AB^2}v_{vehicule} = \frac{1}{\eta AB}T_{red} - F_{trans}$$

$$\frac{J}{\eta AB^2} * \frac{d}{dt}v_{vehicule} + \frac{f}{\eta AB^2}v_{vehicule} = \frac{1}{\eta AB}T_{red} - (M * \frac{d}{dt}v_{vehicule} + F_{res})$$

$$(\frac{J}{\eta AB^2} + M) * \frac{d}{dt} v_{vehicule} + \frac{f}{\eta AB^2} v_{vehicule} = \frac{1}{\eta AB} T_{red} - F_{res}$$

On en déduit alors le couple $F_{red2} = \frac{1}{\eta AB} T_{red}$

On a alors un nouveau bloc avec le couple T_{red2}pris en compte.

Sous Laplace on obtient alors :

$$\frac{v_{vehicule}}{F_{red2} - F_{res}} = \frac{\frac{\eta * AB^2}{f}}{1 + (\frac{J}{f} + \frac{M * \eta * AB^2}{f})p} = \frac{\boldsymbol{K_3}}{\boldsymbol{1 + \tau_3 p}}$$

On remplace maintenant le facteur AB dans l'équation :

$$\frac{v_{vehicule}}{T_{red2} - F_{res}} = \frac{\frac{\eta * (R_{roue} * k_{vis/pignon})^2}{f}}{1 + (\frac{J}{f} + \frac{M * \eta * (R_{roue} * k_{vis/pignon})^2}{f})p}$$

L'identification des valeurs n'est pas encore complètement terminée mais pour le moment le code Matlab pose les différentes valeurs du système. Pour les variables d'environnement j'ai pris la liberté de reprendre celles fournies par Walter Lhomme qui sont des variables fournies en TP. Le reste a été évalué ou en cours de test sur le système pour ajuster le modèle.

Le programme Matlab contenant toutes les différentes valeurs est situé ci-dessous.

```
%% Matlab program: InitIMA5A1920P5.m
%Matlab init file for REM IMA5A1920P5
clc; clear

%%Batterie
BAT.U = 9 ;%tension de la pile en V


%%Regulateur 5V

RED.UBAT = 5/9; % regulateur de tension 5V


%%Cellule photovoltaïque

PV.I0 = 0.8; %courant de saturation, fluctue avec l'irradiance sur la
cellule
PV.VT = 8.2; %tension en court circuit
PV.k = 1.3806e-23; %constante de Boltzmann en J.K-1
PV.q = 1.6022e-19; %charge d'un electrion en C
PV.TC = 18;
PV.T = PV.TC + 273.15; %température en Kelvin
%% Panneaux Photovoltaïques
tension_pv = linspace(0,PV.VT,PV.VT*10);
courant_pv = ones(PV.VT*10,1);
puissance_pv = ones(1,PV.VT*10);
for n = 1:PV.VT*10
    courant_pv(n,1)=PV.I0*(-exp(-PV.q*(PV.VT-
tension_pv(n))/(PV.k*PV.T*60))+1);
    puissance_pv(1,n)=tension_pv(1,n)*courant_pv(n,1);
end

plotyy(tension_pv,courant_pv,tension_pv,puissance_pv,'plot')% plot de
la courbe courant/puissance en fonction de la tension
xlabel('Tension U en V')
legend('Courant I en A','Puissance P en W')

title('Courbe de tension/courant et tension/puissance de la cellule
photovoltaïque')

%%Moteur courant continu
    % --- MCC Parameters ---
        % --- Description ---

    MCC.U = 3;                                          % Nominal voltage
of the armature winding (V)
    MCC.I = 1.04;                                       % Nominal Current
```

```
of the armature winding (A)
    MCC.N = 12200;                                                  %
Nominal speed (rpm)
    MCC.W = MCC.N*pi/30;                                % Nominal speed
(rad/s)
    MCC.P = MCC.U*MCC.I;    % Nominal electrical power of the armature
winding (W)
    MCC.P_util = 0.4;                           % Nominal useful power
(W)
    MCC.tho = 0.003;
    MCC.R=2.2;
    MCC.L=8.54e-2;
          % --- Elec1romechanical conversion ---
     MCC.K  = (MCC.U-MCC.R*MCC.I)/(MCC.W);
      % Torque Coefficient via the excitation Flux

    MCC.tho = 0.0014;
%%Véhicule

VEH.R = 0.0175;%rayon de la roue en m

VEH.M = 0.467; %masse du mobile en kg

VEH.MPLUS = 0; %masse en charge sur le véhicule en kg

VEH.MTOT = VEH.M + VEH.MPLUS; %masse totale du véhicule

VEH.J = 1.2e-3; %inertie du véhicule à définir

VEH.f = 0.05; %frottements


%%Reduction
RED.ARBRE = (12/48)*(12/48); %rapport de réduction lié à l'arbre du
moteur

RED.VP = 10/16; % rapport de réduction lié au système engrenage/pignon

RED.pertes = 0.3;



      % --- Environment parameters ---
            % /// Aerodynamic Resistance ///
            % Far = (rho.Cx.A.(V+Vwind))/2
            ENV.g = 9.81;                           % gravity (m/s^2)
            ENV.A = 9e-5;                              % Frontal aera
(m^2)
            ENV.Cx = 0.35;                          % Drag coefficient
            ENV.rho = 1.223;                        % Density of the air
with 20°C under 1013 mbar (kg/m^3)
            ENV.kaero = ENV.rho*ENV.Cx*ENV.A/2;    % Constant for the
```

```
resistance force to aerodynamics
            ENV.Vwind = 0;                          % Velocity of the
wind (m/s)

            % --- Grade Resistance ---
            % Fgr = m.g.sin(alpha)
            % As the grade alpha is often small the relationship is
generally writen Fgr = m.g.gradient with sin(alpha) = tan(alpha) =
gradient = height / lenght
            ENV.gradient =  0;                                  % Gradient
of the road (%)
            ENV.kgr = VEH.MTOT*ENV.g*ENV.gradient/100;       % Constant
gain for the grade relationship

            % --- Rolling Resistance ---
            % Frr = k1.M.g.cos(alpha)
            % As the grade alpha is often small the relationship is
generally writen Frr = k1.m.g because cos(alpha) =
sin(alpha)/tan(alpha) = 1
            ENV.k1_road = 0.017;                    % Constant resistance
coefficient of the rolling for the roadway
            ENV.krr = ENV.k1_road*VEH.MTOT*ENV.g;  % Constant gain for
the rolling relationship


%% Control parameters
% Maximal Control Structure:
% - all variable are considered ideal
% - all sensors are considered ideal
%%REM transformation

AB = RED.VP * VEH.R;
AB2 = AB*AB;
AB_INV = 1/AB;
AB_INV2 = AB_INV*AB_INV;

num = 428*(RED.pertes*AB2)/VEH.f;
den =  [(VEH.J/VEH.f)+(VEH.MTOT*RED.pertes*AB2)/VEH.f 1];

transfert=tf(num,den);
```

I) Chaîne de retour

Maintenant nous allons nous intéresser à la boucle de retour de notre système. Cette partie servira notamment en cas d'ajout d'un capteur de vitesse au système (type codeur incrémental ou encore roue codeuse). Aux vues de l'avancé du projet, cette partie restera pour le moment purement théorique.

Nous allons donc contrôler le driver moteur à l'aide de la vitesse de sortie du véhicule.

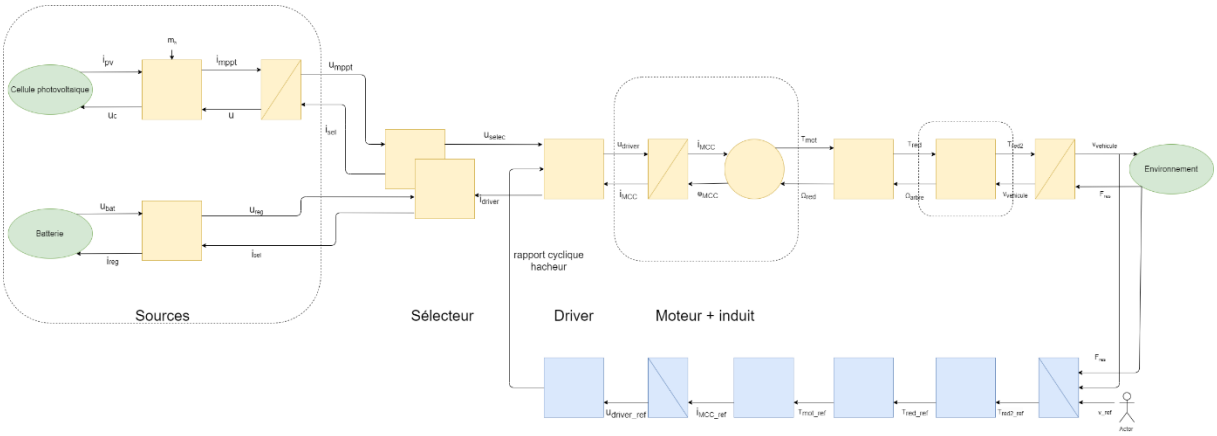

Illustration 16: Chaine de retour de la REM

Les blocs de conversion sont simulés de manière similaire à ceux en chaîne direct, le gain interne est simplement le gain inverse à ceux de la chaine directe.

Pour ce qui est des blocs d'accumulation, ces blocs sont des blocs de type intégrateur/proportionnel (IP) dont la forme est la suivante

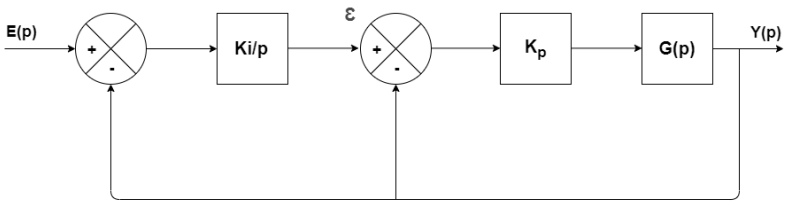

Figure 1: Représentation graphique du régulateur IP

La méthode utilisée pour définir les paramètres Ki et Kp est la méthode de placement de pôles qui nous permet de choisir nos paramètres système.

Nos deux fonctions de transfert G(p) sont les deux fonctions provenant des blocs d'accumulation vus plus haut.

On les nommera comme suivi :

$$G_1(p) = \frac{K_1}{1+\tau_1 p} \quad et\ G_2(p) = \frac{K_3}{1+\tau_3 p}$$

On commence par exprimer l'équation en fonction de $\varepsilon\ et\ Y(p)$

$$Y(p) = (\varepsilon - Y(p))K_p \frac{K}{1+\tau p}$$

On exprime cette équation en fonction de ε et on obtient:

$$\varepsilon = Y(p) * (1 + \frac{1+\tau p}{K_p K})$$

Maintenant on exprime $\varepsilon \; et \; E(p)$

$$\big(E(p) - Y(p)\big) * \frac{K_i}{p} = \; \varepsilon$$

On injecte ensuite cette expression dans l'expression précédente pour obtenir :

$$\frac{Y(p)}{E(p)} = \frac{1}{1 + \frac{1+K_p K_i}{K_p K_i K} p + \frac{\tau}{K_p K_i K} p^2} = \frac{K}{1 + \frac{2\xi}{\omega_0} p + \frac{1}{\omega_0^2} p^2}$$

Par identification on obtient:

$$\omega_0 = \sqrt{\frac{K_p K_i K}{\tau}}$$

$$\xi = \frac{1}{2}(\frac{1+K_p K_i}{\tau\sqrt{K_p K_i K}})$$

- $z > 1$: régime apériodique
- $z = 1$: régime apériodique critique
- $z < 1$: régime pseudopériodique

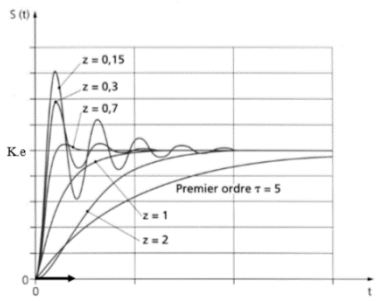

Illustration 17: Réponse à un échelon pour une fonction de transfert de second ordre

Pour ce qui est de la détermination de nos valeurs, on pose $\xi = 0.7$ qui est la valeur optimale pour un coefficient d'amortissement d'après la figure 6. Pour ce qui est de notre pulsation propre, elle va nous permettre de régler l'efficacité de notre système. On a notamment la formule $t_1 = \frac{\pi}{\omega_0\sqrt{1-\xi^2}}$ avec t_1 le temps t à lequel on a le premier dépassement. On doit alors sélectionner nos pulsations propres pour pouvoir opérer à l'identification.

On obtient alors 4 équations composées de 4 inconnues ce qui nous permet de déterminer nos 4 paramètres $K_{p_1}, K_{i_1}, K_{p_3}, K_{i_3}$.déterminés à l'aide d'un solveur.

4) Mise en œuvre des expérimentations

a) Système étudié

Le système a été réalisé avec le kit de robotique fournit en début de projet. Le système est assez simple, j'ai commencé par couper la barrette métallique perforée en deux afin de réaliser mon châssis, j'ai ensuite fait passer les essieux dans ces barrettes. Le support est la plaque métallique qui m'a été fourni, j'ai pu fixer à celui-ci le moteur qui entraine un système de pignon/vis et l'espace supérieur du module est prévu pour augmenter la charge que le module supportera. Néanmoins, l'arbre de rotation était trop long et la force s'appliquant sur les roues non suffisantes, j'ai donc dû ajouter quelques modifications au niveau de l'arbre entraîné par le moteur. On obtient alors le résultat suivant :

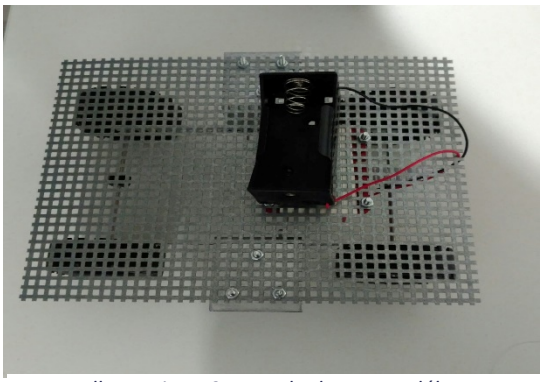

Illustration 19 : Vue de dessus au début

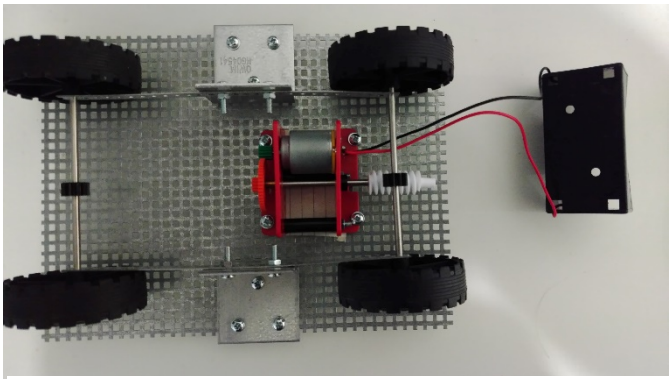

Illustration 18 : Vue de dessous au début

Le système possède un moteur de puissance relativement faible, il a donc dû être nécessaire de réduire les forces de frottement au maximum. Les tests ayant été fait au début du projet avec la pile de 1.5V fourni avec le kit. Il a fallu attendre le driver moteur arrivant avec les premiers composants pour pouvoir appliquer des commandes différentes au moteur.

Avant la soutenance, l'arbre inertiel avait du mal à entraîner les roues du moteur et après de multiples tentatives d'agencement, j'ai finalement inversé le lieu de contact pignon/engrenage afin d'éviter que le poids de l'arbre puisse créer une force sur le châssis et freiner le système. J'ai aussi agrandi les trous au niveau des petites barrettes métalliques afin de ne plus avoir de frottement entre les barres métalliques et les barrettes qui rendait même l'utilisation de la pile fournie difficile.

Je disposais d'engrenages 12/48 pour la grande majorité, avec un seul le système tournait très vite et ne possédait que peu de couple. En passant ensuite à deux, le système s'est avéré être d'une vitesse convenable comme on peut le calculer ici pour la vitesse nominale :

$$V_{nom} = \Omega_{mot} * red_{arbre} * red_{vis/pignon} * \frac{12200 * 2 * \pi}{60} * \left(\frac{12}{48}\right)^2 * \frac{10}{16} * 0.0175 = 0.87 m.s^{-1} = 3.14 km.s^{-1}$$

Cette vitesse est la plus adaptée pour le moment et nous permet d'avoir un couple au niveau des roues plus important.

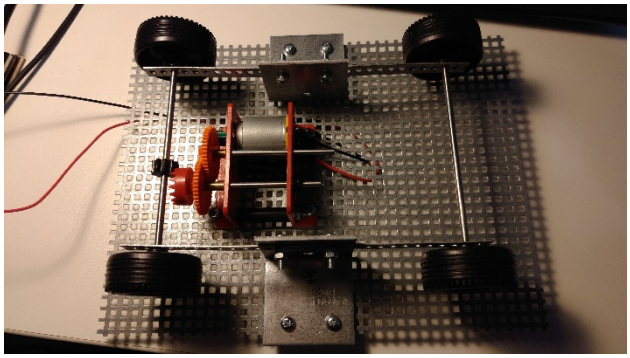

Illustration 20: Vue de dessous après soutenance

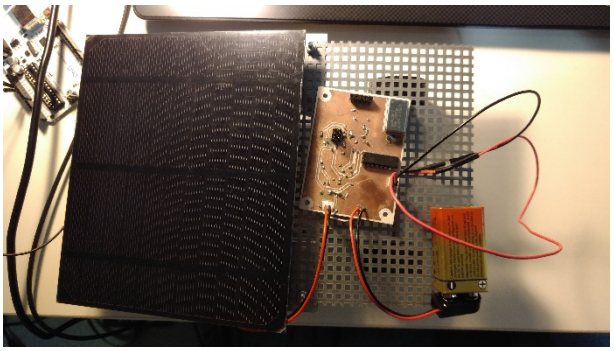

Illustration 21: Vue de dessus après soutenance

b) Mise en œuvre des expériences

- La première expérience consiste à mesurer la vitesse du module.

  Pour se faire, nous allons faire parcourir à notre véhicule électrique une distance de 10m avec une consigne en vitesse choisis par l'utilisateur. On mesure ensuite le temps que le véhicule met pour parcourir ces 10m (sans prendre en compte le démarrage cat cette manière ne permet que de calculer une valeur de vitesse moyenne. Il est donc nécessaire de commencer la mesure en régime établi.

  La première mesure se fera à vitesse minimale (1.5V constant appliqué au moteur)

  La seconde sera à vitesse maximale (3V constant appliqué au moteur).

La formule pour calculer la vitesse ici sera simplement :

$$v = \frac{d}{t} \qquad (.)$$

$$avec\ d\ la\ distance\ parcourue\ en\ régime\ établi\ en\ m$$

$$t\ le\ temps\ en\ s$$

Et $v\ la\ vitesse\ moyenne\ en\ m/s$

L'incertitude se fera sur le nombre d'essais réalisés. Pour se faire nous allons réaliser 5 mesures dans les mêmes conditions. Ensuite faire la moyenne de ces mesures et utiliser la formule d'incertitudes suivante :

$$\sqrt{\sum_{n=1}^{5} \frac{(x_n - \bar{x})^2}{n}} \qquad (.)$$

$$n\ le\ numéro\ de\ l'essai$$

$$x_n\ la\ vitesse\ de\ l'essai\ n\ en\ m/s$$

$$\bar{x}\ la\ vitesse\ moyenne\ en\ m/s$$

- La seconde expérience consiste à mesurer le couple des roues du véhicule.

Pour se faire, nous allons utiliser la technique du bras de levier afin de calculer le moment de celui-ci et ainsi déterminer le couple des roues. Il faut tout d'abord installer une fixation sur l'axe des roues comme sur l'Illustration 13

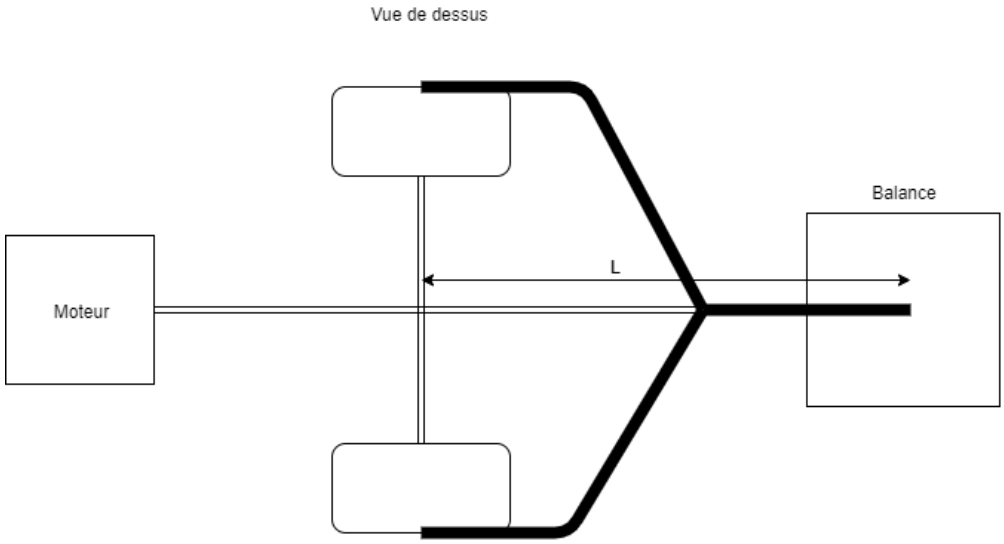

Illustration 22: Schéma d'expérimentation couple roues vue de dessus

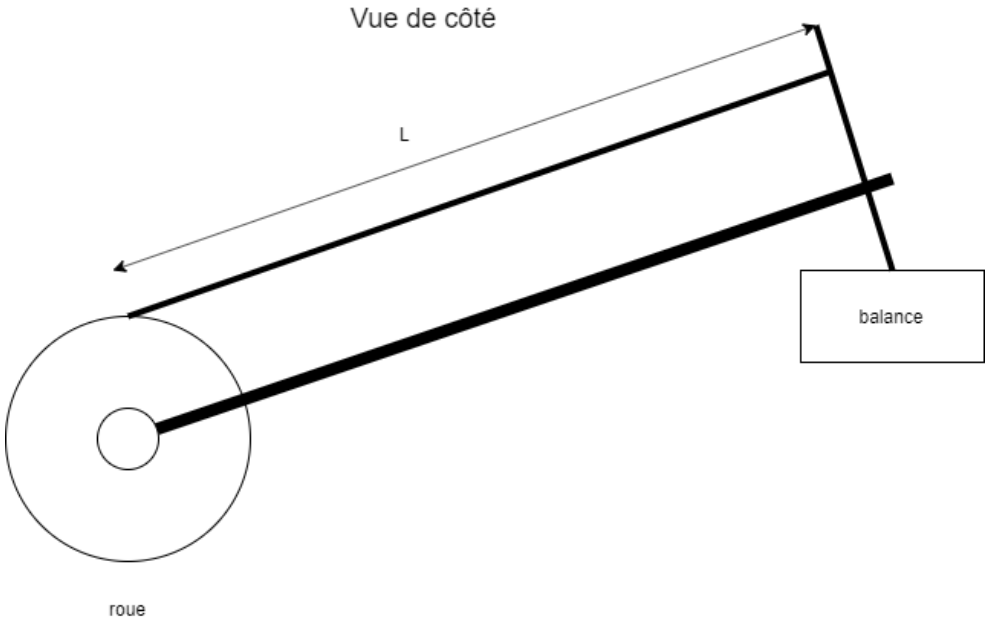

Illustration 23 : Schéma d'expérimentation couple roues vue de côté

Soit L la distance du bras de levier en m

On utilise alors le principe de la dynamique qui nous donne

$$\vec{F} = m * \vec{g}$$

On a alors en valeurs absolues $F = m * g$

Avec F la force appliquée sur le bras au niveau du capteur de masse

M la masse en kg

$g = 9{,}81 m.s^{-2}$

De plus, on a avec le bras de levier un moment qui s'exprime sous la forme suivante :

$$T = F * L$$

En prenant la mesure de masse avant d'actionner la roue avec le moteur et pendant qu'on applique cette force on obtient alors une différence de masse. Cette différence de masse, nous donne alors « m » que l'on remplace dans la formule. Soit $T_{roue} = \Delta m * g * L$.

La balance à ma disposition possède une précision de 1 gramme.

La 3ème expérimentation porte sur le couple de freinage. Afin de calculer celui-ci, on peut placer le système sur un tapis roulant.

Conclusion

Ce projet multidisciplinaire m'a permis de pouvoir tester et utiliser mes connaissances acquises à Polytech en cette fin de période d'études en grande autonomie. Certains points restent encore à être conclus et améliorés du fait de quelques retards de ma part et de quelques problèmes qui sont apparus au cours du projet et ont donc ralentis la réalisation de celui-ci. Toutefois la plupart des tâches sont à un stade très avancé et certaines notamment la représentation énergétique du système à laquelle il ne manque que deux valeurs à identifier. Cette représentation pourra ensuite permettre lorsque la carte sera opérationnelle de transcrire les équations réalisées sous un code arduino pouvant faire office de régulateur et pouvant être contrôlé par n'importe quelle valeur de la chaine direct (dans notre cas la vitesse qui peut réguler le cycle périodique du hacheur par exemple. De plus, la carte est elle aussi presque opérationnelle mais il faudrait certainement changer son microprocesseur pour effectuer des tests plus poussés sur le véhicule. La partie du banc de test est actuellement présentée sous forme d'expérimentations réalisables sur le module. On pourra par la suite faire le banc de test pensé pour ces expériences.

J'ai pu poursuivre ce projet après la soutenance officielle afin de faire fonctionner avec succès les différents éléments de ce projet et plus notamment la gestion de puissance du module. Cela permet d'avoir un résultat plus décent et de partir en ayant rendu un projet qui fonctionne au moins en partie. Les autres éléments auraient été la régulation en vitesse prête à être finalisée d'un point de vue simulation et le banc de tests sur lequel j'ai pu faire une ébauche du type d'expérimentations à effectuer sans pour autant les mettre en œuvre.