



NAUTILUS - RAPPORT

Dylan Bideau, Julien Turpin, Pierre Bogrand, Guillaume Vincenti

10 Avril 2018

Sommaire

| | | |
|----------|-------------------------------------------|-----------|
| 1 | Introduction | 2 |
| 2 | Présentation du projet | 3 |
| 3 | Cahier des charges | 4 |
| 3.1 | Analyse Fonctionnelle | 4 |
| 3.1.1 | Structure | 4 |
| 3.1.2 | Commandabilité | 4 |
| 3.1.3 | Milieu d'utilisation | 4 |
| 3.1.4 | Energie | 5 |
| 3.1.5 | Motorisation | 5 |
| 3.1.6 | Acquisitions | 5 |
| 4 | Motorisation et énergie | 6 |
| 4.1 | Moteurs brushless et ESC | 6 |
| 4.2 | Calibration et commande des ESC | 8 |
| 4.3 | Alimentation et montage | 11 |
| 5 | Références | 12 |

Chapitre 1

Introduction

Les fonds marins réunissent aujourd'hui de nombreux secteurs et enjeux, tant professionnels que particuliers. On y retrouve entre autre l'exploration sous-marine, la surveillance et maintenance d'installations professionnelles, ainsi que la cartographie des fonds marins. Tout ces domaines demandent le développement de solutions techniques plus rentables et pratiques qu'une intervention humaine. Notre projet propose ainsi un ROV (Remotely Operated Vehicle) polyvalent et simple d'utilisation à cet effet.

Chapitre 2

Présentation du projet

Un ROV est un robot sous-marin contrôlé à distance et permettant une acquisition d'informations, visuelles ou à partir de capteurs. Notre projet de ROV filoguidé, Nautilus, sera transportable et pilotable à l'aide d'un ordinateur portable. Il permettra d'observer facilement des installations ou des fonds marins à l'aide de caméras. Disposant également de fonctions avancées, le Nautilus sera en mesure de recréer le fond marin d'une zone géographique déterminée par l'utilisateur à partir d'une batterie de photographies prises lors de la phase d'exploration. Les différentes fonctionnalités du Nautilus en font ainsi un outil polyvalent, permettant exploration, maintenance et cartographie des fonds.

Chapitre 3

Cahier des charges

3.1 Analyse Fonctionnelle

3.1.1 Structure

Facilement transportable et peu encombrant.

Contraintes :

- Poids : 2-3kg
- Dimension : 300*200*150mm
- Etanche de norme IP 68

3.1.2 Commandabilité

Commandé à distance par une liaison filaire.

Contraintes :

- Câble : 15m
- Carte intégrée dans le ROV
- FPV (First Person View)
- Piloté au clavier

3.1.3 Milieu d'utilisation

Adapté aux contraintes imposées par son environnement.

Contraintes :

- Eau non salé (moins de 1 g de sels dissous par kilogramme d'eau)
- Eau translucide (transmittance de la lumière entre 75% et 95%)
- Lieu : Piscine, lac
- Ecoulement laminaire
- Courant marin inférieur à 2 noeuds
- Profondeur de 10m (résistant à 2 bars)

3.1.4 Energie

Etre entièrement autonome.

Contraintes :

- Autonomie de 20 minutes

3.1.5 Motorisation

Etre mobile une fois immergé.

Contraintes :

- Propulsion électrique
- Déplacement horizontal (Vitesse maximale de 1m/s)
- Déplacement vertical (Vitesse maximale de 0.5m/s)
- Direction droite/gauche à 360 degrés

3.1.6 Acquisitions

Acquérir et transmettre l'information.

Contraintes :

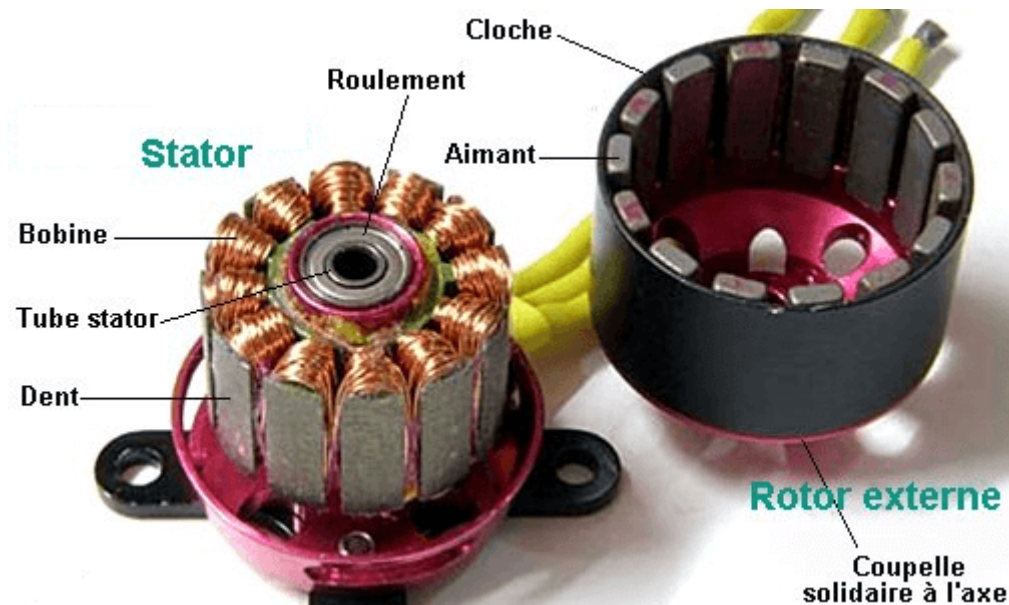
- Acquisition et retransmission d'un signal vidéo
- Acquisition et stockage de photographies
- Mesure de la pression
- Mesure de la position relative avec signaux GPS

Chapitre 4

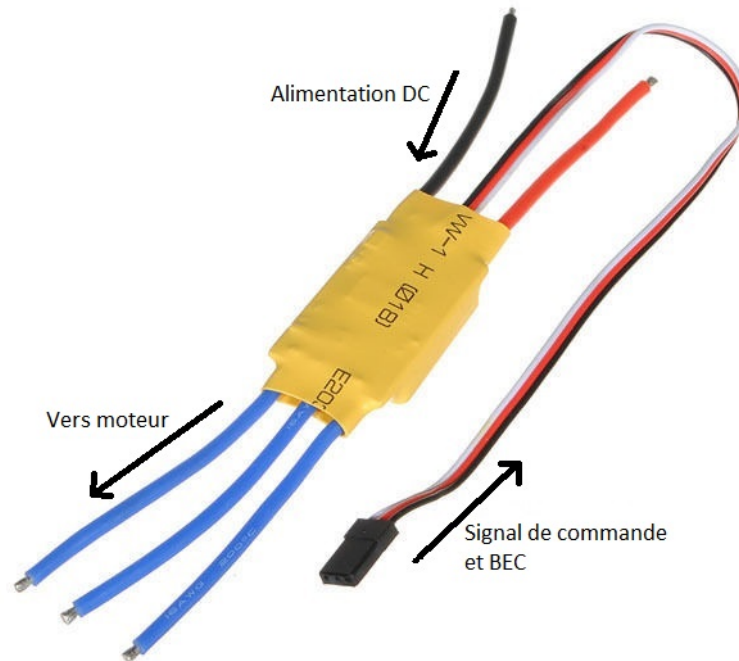
Motorisation et énergie

4.1 Moteurs brushless et ESC

Dans un premier temps, il a été question de la technologie des moteurs à utiliser. Après une étude des différentes solutions disponibles, nous avons finalement choisi des moteurs brushless (Référence 1). En effet, les moteurs brushless sont des machines synchrones auto-pilotées à aimants permanents et donc sans balais.



Le principal avantage de ces moteurs est qu'ils peuvent être utilisés immergés dans l'eau sans aucun traitement particulier au préalable. En revanche, un système électronique de commande doit assurer la commutation du courant dans les enroulements statoriques : les ESC, ou Electronic Speed Controllers. Un ESC transforme un signal d'alimentation continu, dans notre cas issu d'une batterie, en un signal triphasé envoyé ensuite au moteur brushless. Pour contrôler la vitesse de rotation du moteur, on envoie à l'ESC un signal de commande, généralement créneau, et dont le rapport cyclique définit la vitesse du moteur.

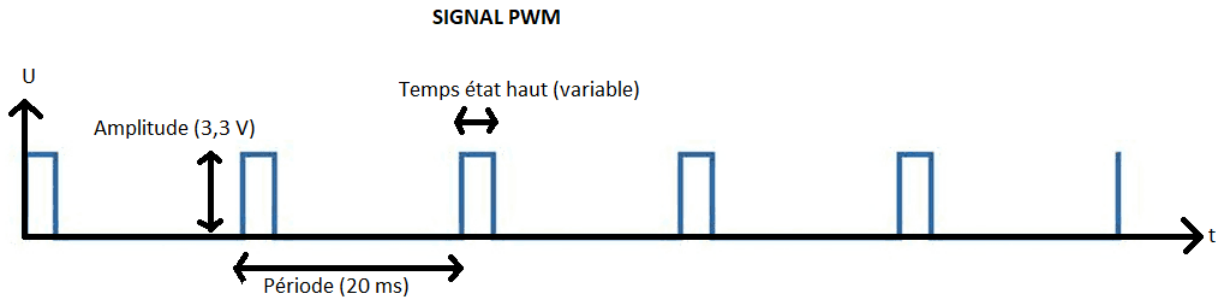


Les trois ESC utilisés dans un premier temps pour nos moteurs (Référence 2) sont également équipés d'un circuit éliminateur de batterie, ou BEC, permettant de générer un signal d'alimentation constant de 5V et 3A maximum. Ce dernier permet d'alimenter un autre composant, comme une carte Raspberry Pi dans notre cas, sans avoir à recourir à une seconde batterie.

Cependant, ces ESC ne permettent la rotation du moteur que dans un seul sens. Le seul moyen de modifier le sens de rotation du moteur dans ce cas est d'échanger deux des trois signaux déphasés envoyés au moteur. La direction droite/gauche étant assurée par les deux moteurs de propulsion arrière, cette particularité n'est pas problématique : la rotation plus rapide d'un des deux moteurs arrière par rapport à l'autre permet de diriger le ROV à gauche ou à droite. En revanche, le moteur vertical devant assurer la propulsion verticale doit pouvoir tourner dans les deux sens. Un second modèle d'ESC a donc été nécessaire pour permettre au moteur de tourner dans les deux sens. Ce dernier (Référence 3) possède ainsi un mode "reverse" permettant au moteur de tourner dans les deux sens, ainsi qu'un BEC, et sera donc attribué au moteur vertical du ROV.

4.2 Calibration et commande des ESC

Traditionnellement, le signal de contrôle envoyé à l'ESC est un signal PWM de fréquence 50 Hz environ, un certain écart de cette valeur étant accepté. Dans notre cas, le signal est généré par une Raspberry Pi 3, et l'amplitude du signal est donc de 3,3 V environ.



La caractéristique la plus importante de ce signal est la largeur de chaque impulsion. C'est cette dernière, généralement entre 0.5 ms et 2.5 ms, qui commande directement l'amplitude du signal envoyé aux moteurs et donc leur vitesse de rotation. Notre carte doit donc être capable de générer un signal créneau à 50 Hz, et de rapport cyclique variable sur commande. Pour générer un tel signal, on utilise le programme ServoBlaster, disponible sur Github. En effet ce dernier a été conçu pour piloter des servomoteurs à partir des pins GPIO de la Raspberry Pi, et est donc le plus adapté pour générer un tel signal. Pour installer ServoBlaster sur la carte, on rentre les commandes suivantes sur le terminal de Raspbian :

```
1 sudo apt-get install git
2 git clone https://github.com/richardghirst/PiBits.git
3 cd PiBits/ServoBlaster/user
4 nano init-script
5 # Supprimer la valeur --idle-timeout=2000 des options par défaut
6 nano servod.c
7 # Modifier la ligne 960 pour avoir "else if (strstr(modelstr, "BCM2709") ||
  strstr(modelstr, "BCM2835"))"
8 make
9 sudo make install
```

Une fois cette installation effectuée, on peut générer le signal à l'aide de la commande, toujours dans le terminal de Raspbian, "echo x=y > /dev/servoblaster", où x représente le GPIO que l'on utilise et y la valeur du temps d'état haut, en dizaine de us (par exemple y=100 correspond à un temps d'état haut de 1 ms). Les valeurs de y peuvent aller de 50 à 250, correspondant respectivement à un temps haut de 0.5 ms et 2.5 ms.

Les valeurs de x désignent les GPIO suivant :

| Servo number | GPIO number |
|--------------|-------------|
| 0 | 4 |
| 1 | 17 |
| 2 | 18 |
| 3 | 21/27 |
| 4 | 22 |
| 5 | 23 |
| 6 | 24 |
| 7 | 25 |

On obtient ainsi des commandes comme ci-dessous :

```
1 echo 0=200 > /dev/servoblaster
2 echo 0=100 > /dev/servoblaster
```

Avec chaque nouvelle valeur de y modifiant la valeur du temps haut du signal sortant.

Toutefois, avant leur utilisation, les ESC nécessitent d'être calibrés. En effet, en fonction des radiocommandes ou autre dispositifs de commande utilisés, l'intervalle des valeurs du temps d'état haut du signal de commande envoyé à l'ESC peut différer. L'ESC doit donc être calibré pour que la valeur maximale de temps d'état haut corresponde à la vitesse maximale de rotation du moteur. De même pour une valeur neutre (moteur à l'arrêt), et une valeur minimale correspondant à la vitesse de rotation inverse maximale, disponible uniquement pour l'ESC "Reverse", et donc le moteur vertical associé.

La calibration ne peut pas être effectuée à l'aide des commandes issues de ServoBlaster, l'ESC nécessitant une variation douce du temps d'état haut pour cela. C'est le cas d'un joystick de télécommande, mais pas du programme ne permettant que des variations abrupts de la valeur du temps d'état haut. La calibration des ESC a donc été effectuée à l'aide d'une radiocommande disponible à l'ENSEA (Référence 4). Pour cela, on place le joystick de la manette au point neutre, on branche l'ESC à l'émetteur de la radiocommande et on allume dans l'ordre la radiocommande, puis l'ESC. Lorsque ce dernier s'allume, il associe ainsi automatiquement le signal en train d'être reçu au point neutre, donc moteur immobile. On effectue la même opération avec le joystick au maximum pour calibrer la valeur maximale de temps d'état haut et l'associer à la vitesse maximale de rotation du moteur. Une fois cette calibration effectuée, l'ESC en marche ne fonctionnera qu'après avoir reçu le signal correspondant à l'état neutre pendant un certain temps. Cela permet d'éviter entre autres un démarrage intempestif des moteurs.

Une fois la calibration effectuée, on observe à l'oscilloscope le signal de commande envoyé par la manette pour déterminer le temps d'état haut correspondant au point neutre, à la vitesse maximale, et à la vitesse maximale en rotation inverse pour le moteur vertical.

On obtient les résultats suivants :

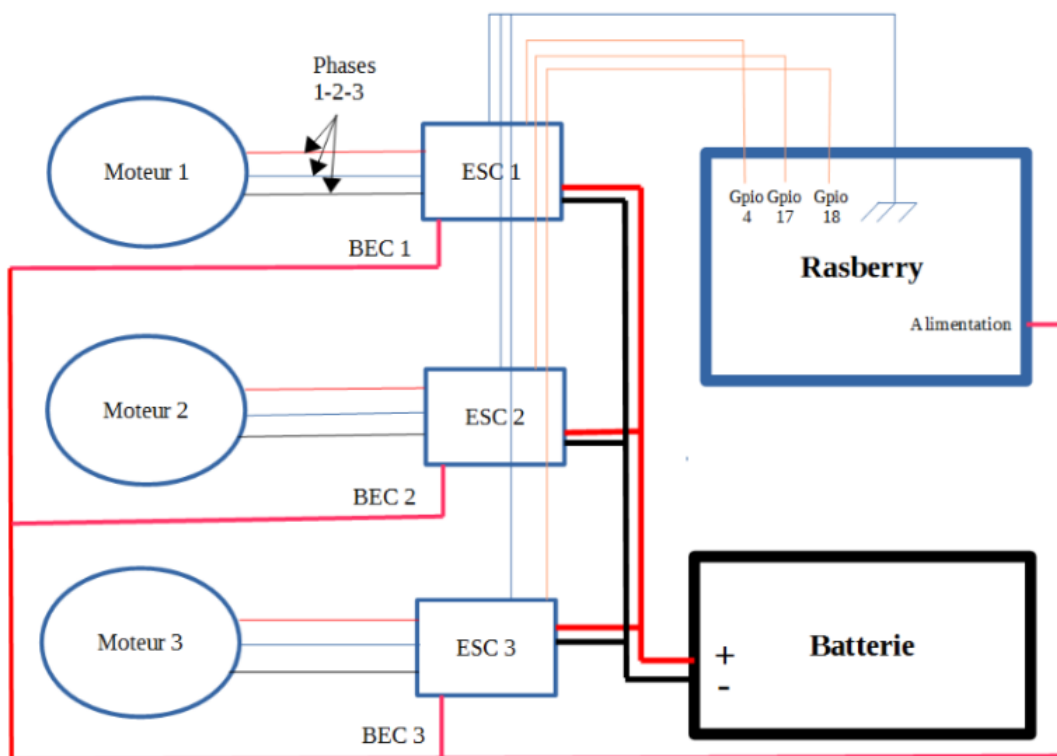
| Temps état haut Type d'ESC | Point neutre (valeur de y) | Vitesse maximale (valeur de y) | Vitesse maximale inverse (valeur de y) |
|-----------------------------------------------------|-------------------------------|-----------------------------------|-------------------------------------------|
| ESC 20A (propulsion) | 1,2 ms (120) | 1,8 ms (180) | Pas de mode Reverse |
| ESC 30A (moteur vertical) | 1,5 ms (150) | 1,9 ms (190) | 1,4 ms (140) |

Après calibration et relevé de ces valeurs, on peut bien commander les 3 moteurs à l'aide des commandes de ServoBlaster.

4.3 Alimentation et montage

On utilise pour l'alimentation des ESC et des moteurs une batterie NiMh, de capacité 3000 mAh et délivrant une tension de 7,2 V (Référence 5). L'alimentation de la Raspberry est elle assurée par les BEC des ESC : ces derniers, mis en parallèle, délivrent une alimentation constante d'environ 5V et 3A, suffisante à alimenter la carte. On obtient donc le schéma fonctionnel suivant :

Architecture de la motorisation du ROV



Lorsque les 3 moteurs sont à leur vitesse maximale, avec la carte fonctionnelle, le courant prélevé sur la batterie est de 2A maximum. L'autonomie théorique dans cette configuration est donc d'environ 90 minutes. Toutefois cette dernière risque de diminuer avec l'utilisation et donc l'alimentation des différents capteurs, ainsi que la résistance de l'eau en condition.

Chapitre 5

Références

Motorisation et énergie :

- **Référence 1** : Moteur d'avion électrique brushless ROXXY 315079 chez Conrad (x3)
- **Référence 2** : ESC Suppo Multirotor 20A M20A chez RobotShop (x3)
- **Référence 3** : ESC HobbyKing 30A avec Reverse (x1)
- **Référence 4** : Radiocommande Graupner MX-20 (disponible à l'ENSEA)
- **Référence 5** : Batterie d'accumulateurs (NiMh) 7.2 V 3000 mAh Conrad (x1)