

République Algérienne démocratique et Populaire

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université des Sciences et de la Technologies d'Oran « Mohamed Boudiaf »



FACULTE DE GENIE ELECTRIQUE

Département d'Automatique

Calcul de la vitesse d'un moteur par un
<<Capteur infrarouge>>
ET IDENTIFICATION DE SA REPONSE

Réalisé par :

- RAHMOUN LOKMANE NOUR EL ISLEM.
- KABOUCHE BACIM.

Année universitaire 2019/2020

Introduction générale

Dans nombreux domaines (industrielle, recherche scientifique, services, loisirs), on a besoin de contrôler de nombreux paramètres physiques par exemple (température, force, position, vitesse, luminosité,...). Le capteur est l'élément indispensable à la mesure de ces grandeurs physiques.

Son domaine d'utilisation est très vaste il intervient dans tout les domaines où il y a un recueil d'information en vue d'un traitement d'information. En effet tout les domaines d'activités nécessitent l'emploi de capteurs. Par exemples le domaine d'automobile, agriculture contrôle de la production...

Dans ce projet on est mené à calculer la vitesse d'un moteur a courant continue en utilisant un capteur infrarouge et une carte arduino.

Les principaux objectifs de ce projet sont :

- Aborder quelque concept sur les capteurs d'une façon générale.
- Lire la vitesse du moteur à partir du capteur infrarouge.
- Déterminer le modèle mathématique de notre moteur.
- Faire l'identification de notre système.

Notre travail est structuré de la manière suivante :

- **Premier Chapitre :**

Généralités sur les capteurs.

Généralités sur le moteur à courant continue.

- **Deuxième Chapitre :**

Réalisation d'un montage qui permet de détecter la vitesse du moteur à l'aide d'un capteur infrarouge.

- **Troisième chapitre :**

Identification de la réponse indicielle du moteur .

Premier chapitre

Généralités sur les capteurs

Dans ce chapitre on va aborder d'une façon générale quelque notion sur le fonctionnement du capteur et du moteur à courant continu.

1. Introduction :

C'est un organe de mesure qui transforme une grandeur physique analogique (variation continue dans le temps) en une grandeur électrique pouvant être traité par une unité de traitement.

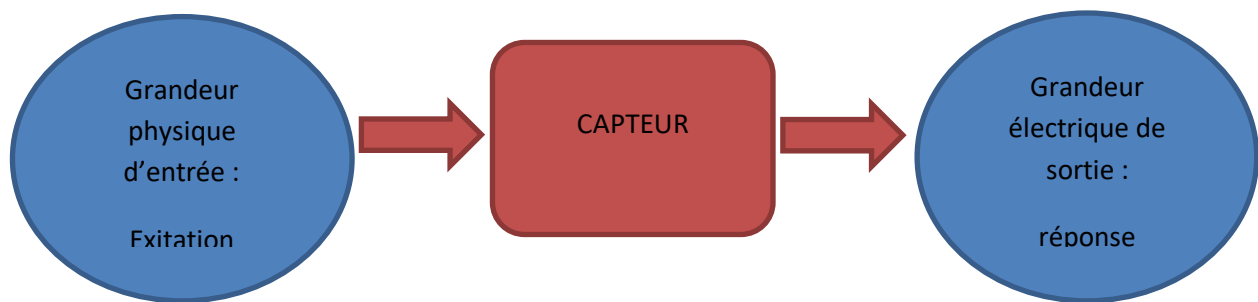


Figure 1.1 : Schéma simple d'un capteur.

2. Classification des capteurs selon leur principe de fonctionnement :

2.1 Le capteur actif (transducteur) :

Il assure une conversion en une énergie électrique, caractérisé par une tension un courant ou une charge électrique. La valeur de sortie étant faible il faudra l'amplifier.

Exemples :

Mesurande	Effet physique utilisé	Grandeur de sortie
Température	Thermoélectricité	Tension
Flux lumineux	Photoémission	Courant
	Pyroélectricité	Charge
	Effet Photovoltaïque	Tension
Force, Pression, accélération	Piézoélectricité	Charge
Position	Effet Hall	Tension
Vitesse	Induction EM	Tension

2.2 Le capteur passif :

Il se présente comme une impédance dont l'un des paramètres est sensible au mesurande. Il a besoin d'une source d'excitation pour fournir un signal électrique de mesure.

2.3 Le capteur intelligent :

C'est un dispositif communiquant qui correspond à l'intégration dans le corps du capteur un organe de calcul (un microcontrôleur) et une interface de communication.

3. Le Capteur infrarouge :

3.1 Introduction :

Le spectre infrarouge (IR) fait partie du spectre électromagnétique: il est défini comme la Région juste au-dessus des ondes visibles ($0,7 \mu\text{m}$) à millimètres ($1000 \mu\text{m}$). C'est la région où chaque objet ayant une température non nulle émet un «rayonnement thermique».

Les caractéristiques de ce rayonnement dépendent de la température et de la longueur d'onde.

Puisque l'œil humain répond bien à la lumière visible mais mal aux rayonnements infrarouges, presque toutes les informations codées dans le rayonnement infrarouge ne sont pas directement détecté par les yeux humains. Par conséquent, il est nécessaire de développer un appareil thermique si le rayonnement doit être détecté.

3.2 Définition d'un capteur infrarouge :

Un capteur infrarouge (IR) est un appareil électronique qui mesure et détecte le rayonnement infrarouge dans son environnement. Il existe deux types de capteurs infrarouges: actifs et passifs. Les capteurs infrarouges actifs émettent et détectent à la fois un rayonnement infrarouge. Les capteurs IR actifs se composent de deux parties: une diode électroluminescente (LED) et un récepteur. Lorsqu'un objet s'approche du capteur, la lumière infrarouge de la LED se reflète sur l'objet et est détectée par le récepteur. Les capteurs infrarouges actifs agissent comme des capteurs de proximité et sont couramment utilisés dans les systèmes de détection d'obstacles (comme dans les robots). Les capteurs infrarouges passifs (PIR) détectent uniquement les rayonnements infrarouges et ne les émettent pas à partir d'une LED. Les capteurs infrarouges passifs sont composés de:

- Deux bandes de matériau pyroélectrique (un capteur pyroélectrique)
- Un filtre infrarouge (qui bloque toutes les autres longueurs d'onde de la lumière)
- Une lentille de Fresnel (qui collecte la lumière sous plusieurs angles en un seul point)
- Un boîtier (pour protéger le capteur d'autres variables environnementales, telles que l'humidité).

Les capteurs PIR sont les plus couramment utilisés dans la détection basée sur le mouvement, comme les systèmes de sécurité à domicile. Lorsqu'un objet en mouvement qui génère un rayonnement infrarouge entre dans la plage de détection du détecteur, la différence des niveaux IR entre les deux éléments pyroélectriques est mesurée. Le capteur envoie alors un signal électronique à un ordinateur embarqué, qui à son tour déclenche une alarme.

Avantage :

- Bon marché et faciles à mettre en œuvre.

Inconvénients :

- Ils sont sensibles à la lumière ambiante.
- Ils sont difficiles à calibrer.
- Ils ne conviennent aussi que pour des courtes distances.

Dans cette réalisation on va utiliser un capteur infrarouge FC 51 IR.

3.3 Capteur FC 51 IR :

Le concept de base de la détection d'obstacles IR (infrarouge) est de transmettre le signal IR (rayonnement) dans une direction et un signal est reçu au niveau du récepteur IR lorsque le rayonnement IR rebondit sur une surface de l'objet.

Caractéristique:

- Il y a un obstacle, le voyant vert allume sur le circuit imprimé.
- Signal de sortie numérique.
- Distance de détection: 2 ~ 20 cm.
- Angle de détection: 35 ° degré.
- Puce de comparaison: LM393.
- Plage de distance de détection réglable via potentiomètre:
 - Sens horaire: Augmentez la distance de détection
 - Sens antihoraire: réduire la distance de détection

Spécifications:

- Tension de fonctionnement: 3 - 5V DC.
- Type de sortie: sortie de commutation numérique (0 et 1).
- Trous de vis de 3 mm pour un montage facile.
- Taille de la planche: 3,2 x 1,4 cm.

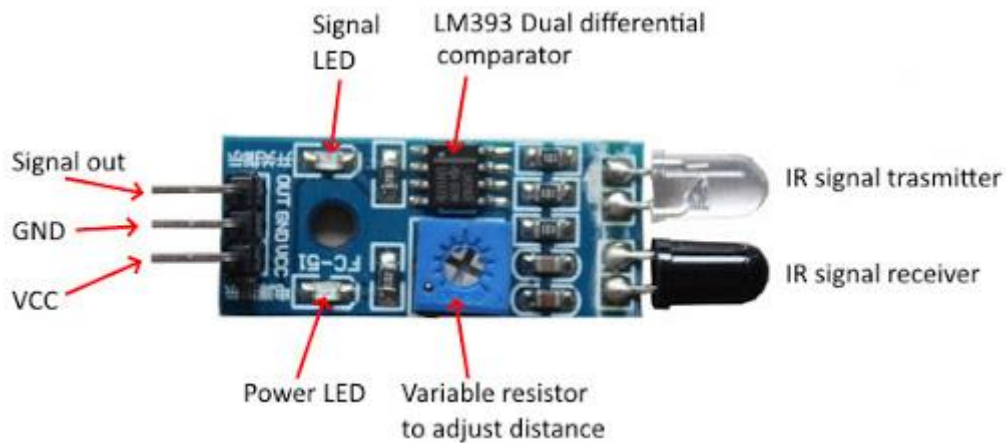


Figure 1.2 : Vue réel du capteur FC 51 IR.

4. Généralité sur le moteur à courant continu :

Dans cette partie on va comprendre le fonctionnement du moteur à courant continu et ses principales constitutions d'une façon très simple.

Le moteur à courant continu (mcc) est une machine tournante qui transforme l'énergie électrique en mécanique, Son fonctionnement est basé sur la loi de Laplace : soit un conducteur L traversé par un courant I et placé dans un champ magnétique B est soumis à une force F dont le sens est déterminé par la règle des trois doigts de la main droite.

$$F = B \cdot S \cdot I \cdot L \cdot \sin(\alpha)$$

Cette force va tourner le rotor et créer le couple moteur, la construction du moteur est basée sur ce principe.

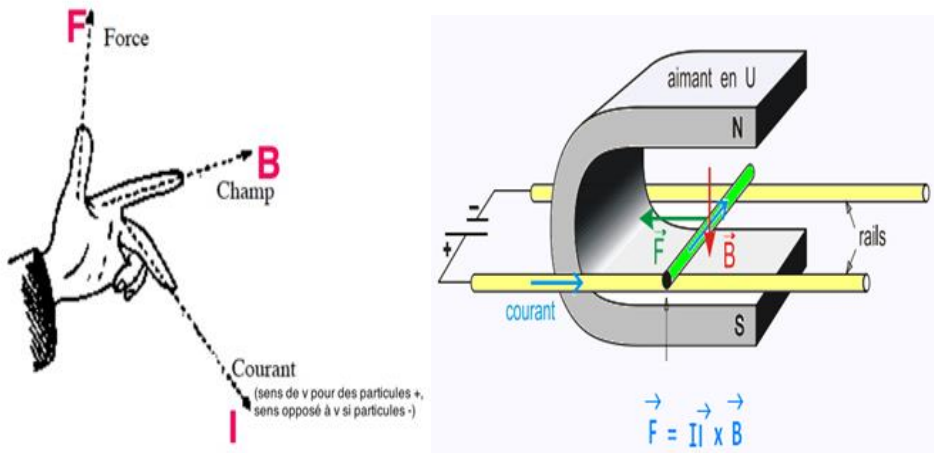


Figure 1.3: Principe de la force LAPLACE

4.1 Constitution d'un moteur à courant continu :

Le moteur à courant continu est constitué principalement par deux ensembles :

Stator (Inducteur) :

C'est l'élément fixe, son rôle est de créer un flux magnétique, cette fonction est assurée par un aimant permanent ou par un courant électrique circulant dans le bobinage.

Rotor (Induit) :

Composé d'un châssis métallique comprenant un certain nombre d'encoches, sur lequel sont placé un certain nombre de bobinage.

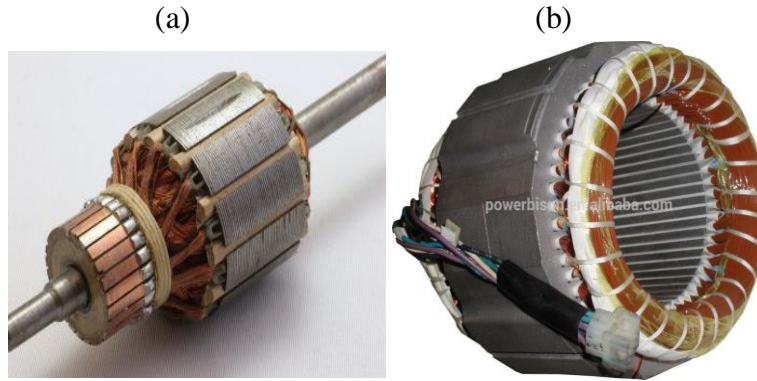


Figure 1.4 : (a) Illustration de l'inducteur, (b) Illustration de l'induit.

4.2 Les branchements des moteurs à courant continu :

a-excitation indépendante :

L'inducteur et l'induit sont alimentés de façons indépendantes.

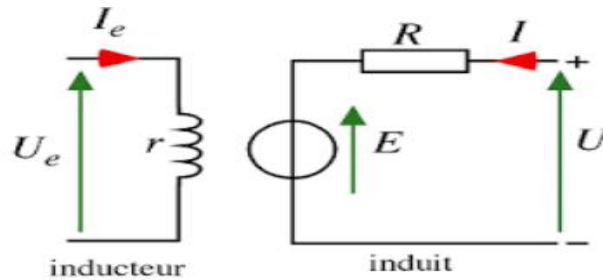


Figure 1.5 : Circuit électrique du moteur à excitation Séparée.

b-excitation série :

L'inducteur et l'induit sont connectés en série.

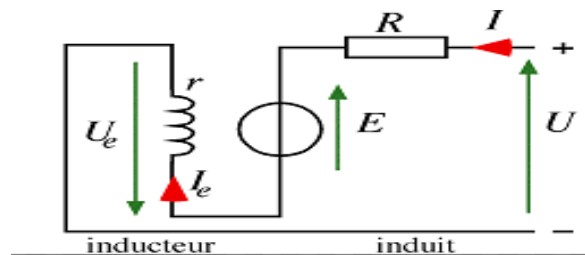


Figure 1.6 : Circuit électrique du moteur à excitation Shunt.

c-excitation parallèle(ou shunt) :

L'inducteur et l'induit sont connectés en parallèle.

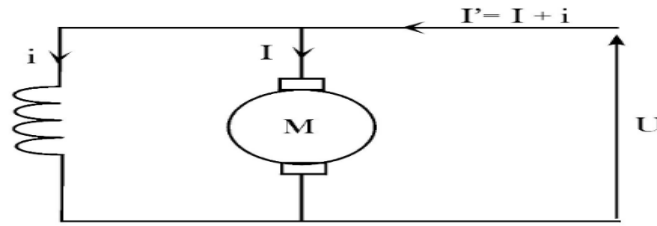


Figure 1.7 : Circuit électrique du moteur à excitation Série.

d-excitation composé :

Une partie du stator est raccordé en série avec le rotor et un autre est de type parallèle ou shunt. Ce moteur réunit les avantages des deux types de moteur : le fort couple à basse vitesse du moteur série et l'absence d'emballement (survitesse) du moteur shunt.

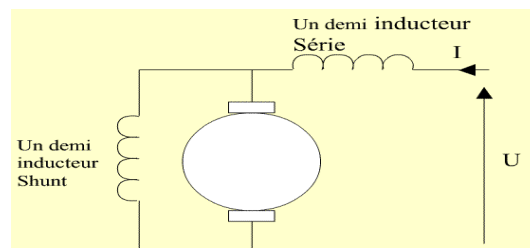


Figure 1.8 : Circuit électrique du moteur à excitation Composée.

On va utiliser un moteur MAXON A-max 391116 il est caractérisé par :

1. marque: Maxon A-max
2. Modèle: 391116
3. diamètre du moteur: 22mm
4. longueur du moteur: 32mm
5. diamètre de l'arbre de sortie: 2,0 mm (plat: 1,6 mm)
6. longueur de l'arbre de sortie: 6 mm (arbre D)
7. longueur totale: 39mm
8. tension: DC 12 V-24 V



Figure 1.9 : vue réel du moteur utilisé.

5. Carte arduino :

L'arduino c'est une plateforme électronique open source basé sur une simple interface entrée/sortie. Les cartes arduino peuvent lire les entrées (lumière sur un capteur, doigt sur un bouton ou message Twitter) et en faire une sortie: activer un moteur, allumer une LED, publier quelque chose en ligne. Pour ce faire, on utilise le langage de programmation arduino (basé sur le câblage) et le logiciel arduino (IDE) basé sur le traitement.

Dans cette présentation on a choisi d'utiliser une carte arduino.

5.1 Carte arduino uno :

Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328P. Il possède :

14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées en tant que sorties PWM).

6 entrées analogiques.

Un quartz 16 MHz.

Une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation.

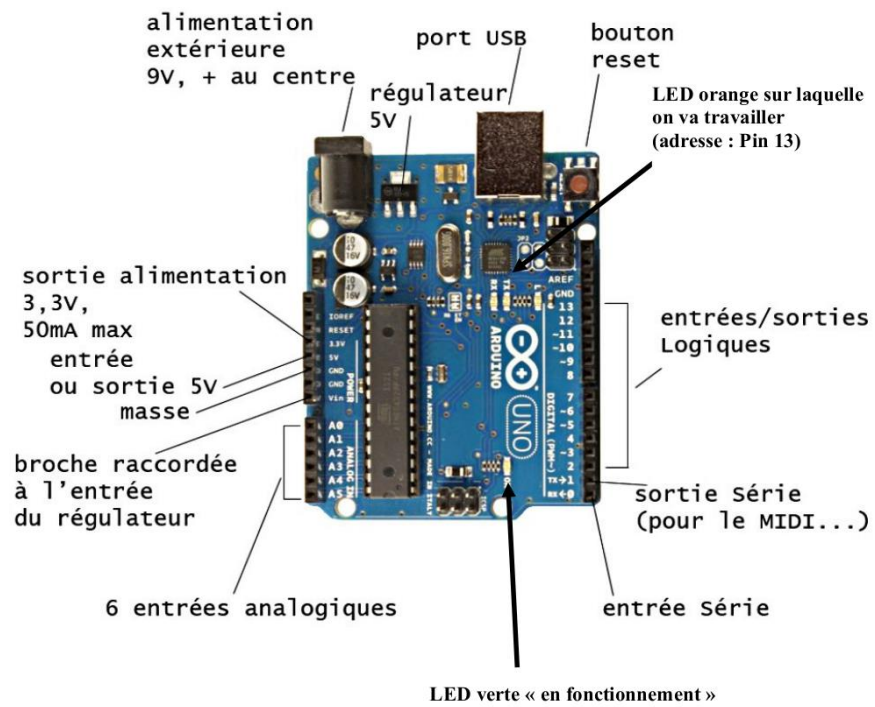


Figure 1.10 : Vue réelle sur la carte ARDUINO UNO

Deuxième chapitre :

Hypothèse :

Dans notre projet on a utilisé un moteur maxon mais le problème c'est que ce moteur ne contient plus de datasheet donc on a décidé d'utiliser une méthode pour déterminer les paramètres de notre moteur l'idée était ce que on peut déterminer les paramètres on utilise un capteur infrarouge.

1. Principe du Calcul de la vitesse :

Notre principale but de ce projet et de calculer la vitesse du moteur à l'aide du capteur infrarouge,

Pour cela on a utilisé trois composants principaux qui sont une carte arduino, un capteur et une hélice placée dans le rotor du moteur on va rapprocher l'hélice du capteur d'une façon à placer l'hélice en haut du capteur. Puisque on sait que l'infrarouge il ne détecte pas la couleur noire elle considéré comme passive donc on va couvrir les extrémités des dents de l'hélice par une couleur noire et on laisse une seule borne nommé x son couvert quand l'hélice va tourner il va détecter x donc une tour. La précision elle est liée par le nombre de dents utilisé dans le calcul.

Notre projet on peut le schématiser de la manière suivante :

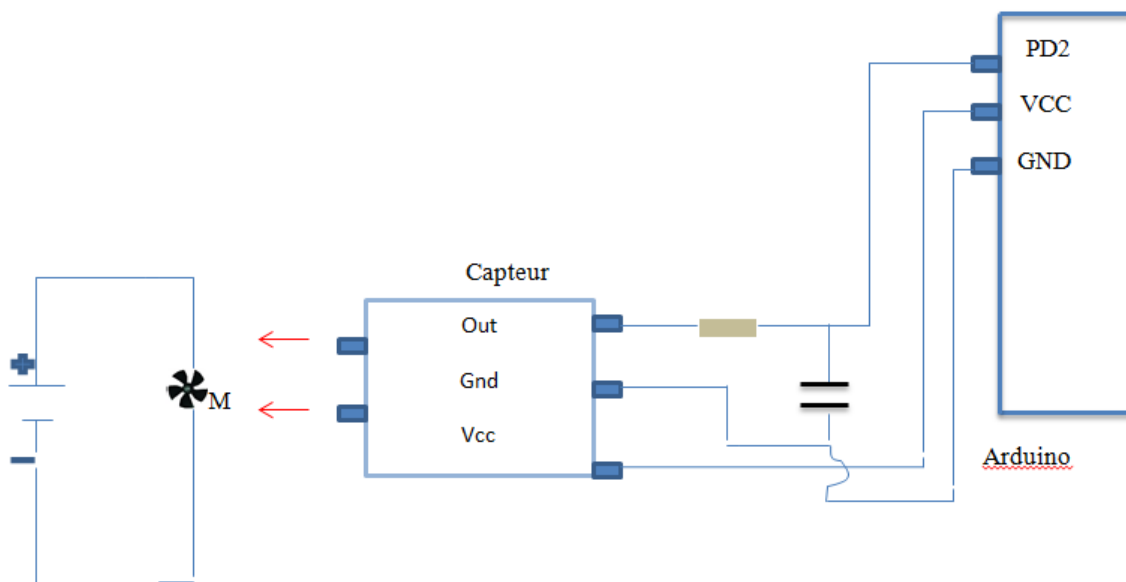


Figure 2.1: Schématisation simple du fonctionnement.

Le capteur est relié avec l'arduino de la manière suivante :

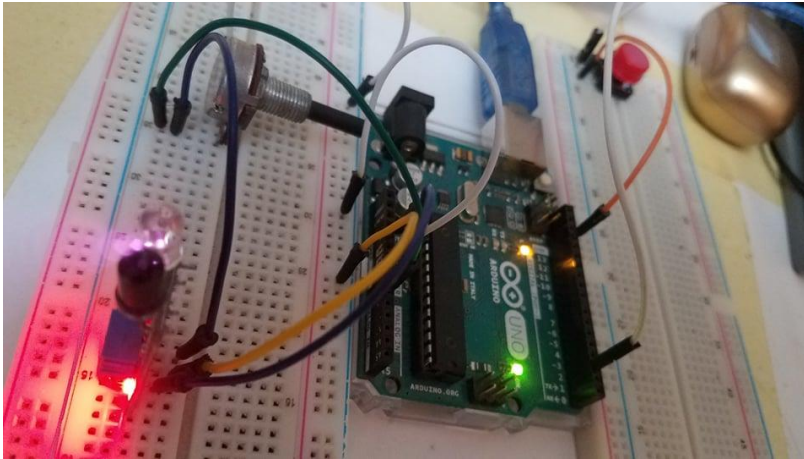


Figure 2.2: reliure du capteur avec arduino.

L'expérience fait est la suivante :

Puisque on sait que l'infrarouge il ne détecte pas la couleur noire elle considéré comme passive donc on a couvrez une partie avec le noir qui n'est pas détectable est la partie restante on la couvre avec un scotch jaune qui est détectable.

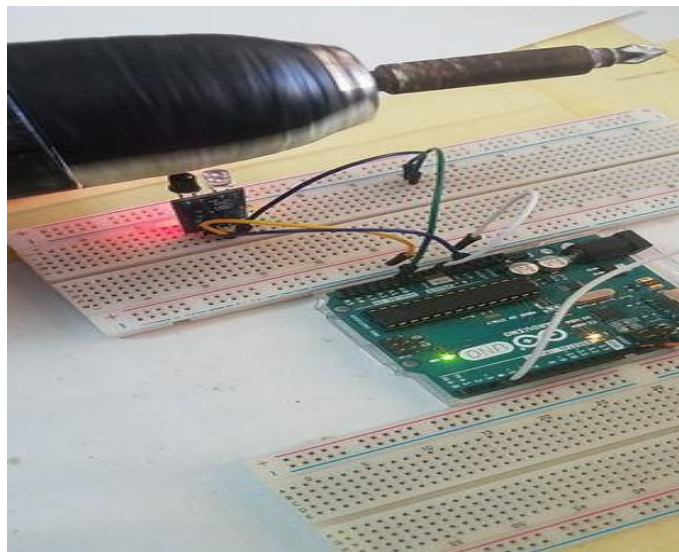


Figure 2.4: Montage du capteur avec arduino et visseuse.

Dans cette figure on peut bien remarquer que le capteur n'a pas pu détecter cette couleur noir, Par ce que elle est considéré comme passive. On voit bien que la led verte est éteinte.

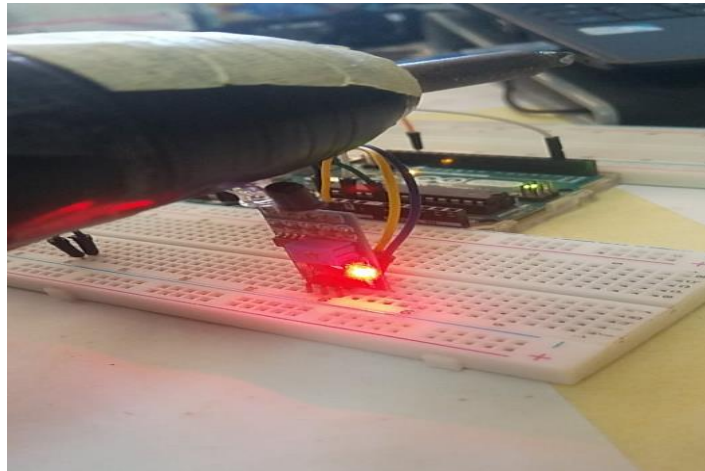


Figure 2.5: Cette figure montre que le capteur ne détecte pas la couleur noir.

Dans cette deuxième figure le capteur à détecter la couleur jaune Par ce que elle est considéré comme active. On voit bien que la led verte est allumé.

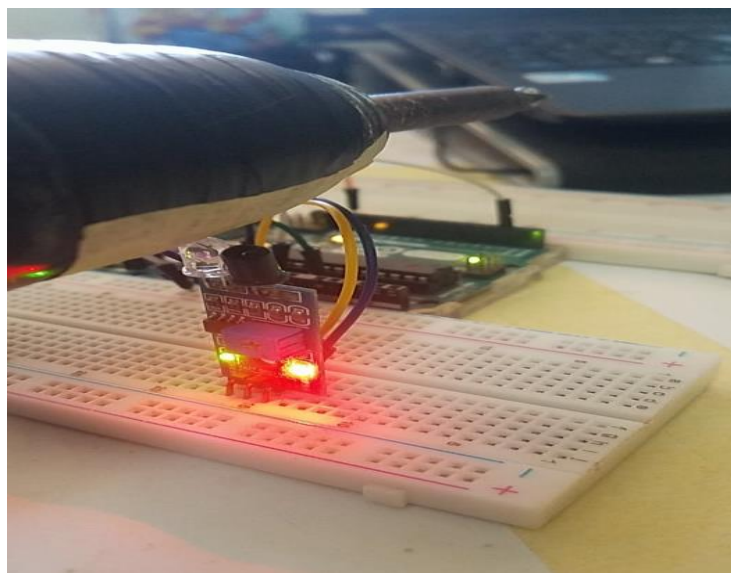
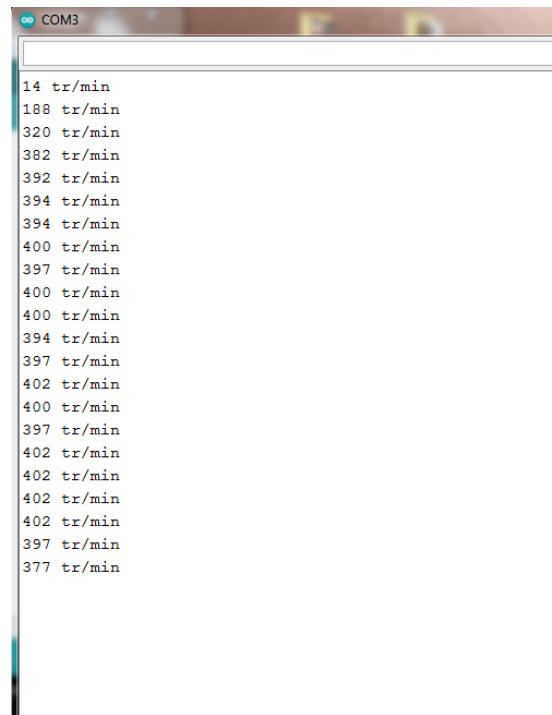


Figure 2.6: Cette figure montre que le capteur détecte la couleur jaune.

La vitesse détectée par le capteur est la suivante :



```
COM3
14 tr/min
188 tr/min
320 tr/min
382 tr/min
392 tr/min
394 tr/min
394 tr/min
400 tr/min
397 tr/min
400 tr/min
400 tr/min
394 tr/min
397 tr/min
402 tr/min
400 tr/min
397 tr/min
402 tr/min
402 tr/min
402 tr/min
402 tr/min
402 tr/min
397 tr/min
377 tr/min
```

Figure 2.7: Affichage de la 1^{ère} vitesse de la visseuse.



```
COM3
13 tr/min
170 tr/min
833 tr/min
1276 tr/min
1363 tr/min
1363 tr/min
1463 tr/min
1463 tr/min
1500 tr/min
1500 tr/min
1500 tr/min
1500 tr/min
1538 tr/min
1538 tr/min
1500 tr/min
1538 tr/min
1500 tr/min
1500 tr/min
1500 tr/min
1500 tr/min
1500 tr/min
1538 tr/min
1500 tr/min
1500 tr/min
1500 tr/min
1428 tr/min
1276 tr/min
1111 tr/min
967 tr/min
810 tr/min
```

Figure 2.8: Affichage de la 2^{ème} vitesse de la visseuse.

Alors maintenant on a pu confirmer le bon fonctionnement de notre capteur on va retourner à la réalisation principale.

Le montage utiliser est le suivant :

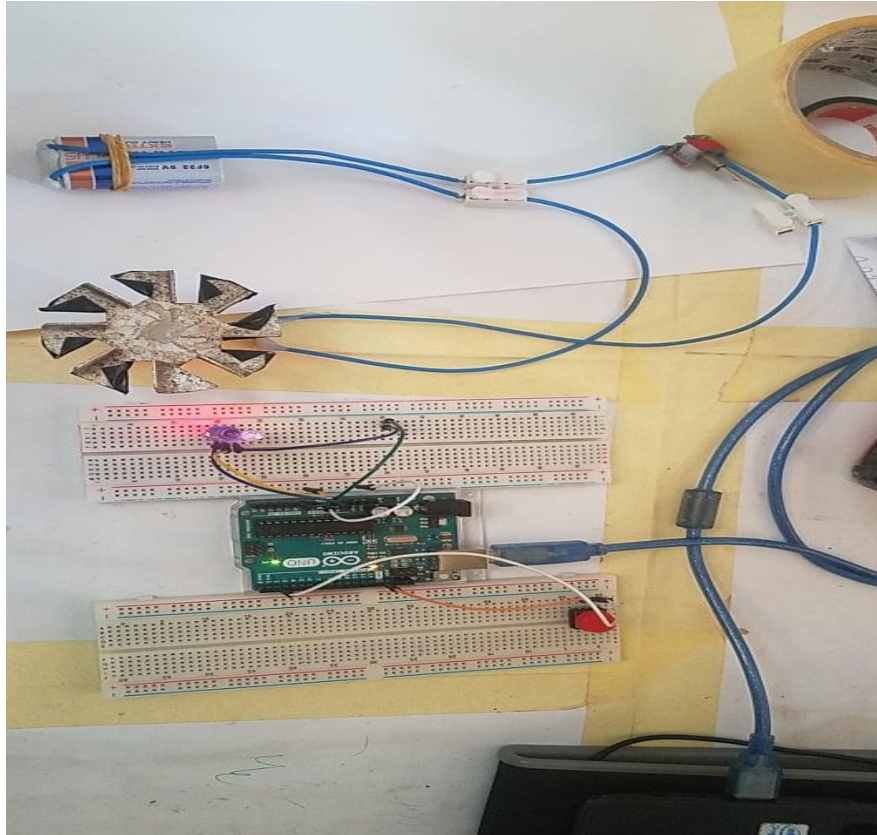


Figure 2.9: Montage qui contient moteur avec hélice, arduino et capteur.

Ce montage contient une alimentation 9v placé avec le moteur ensuite on a un capteur relié avec une carte arduino laquelle elle est reliée avec un ordinateur.

Maintenant on doit ajuster la largeur de la bande pour qu'on peut la détecter, plus on augmente la largeur de la bande plus on aura plus besoin d'une vitesse d'exécution du programme ni vitesse de mesure de l'ADC a chaque fois le flag de la comparaison du timer 0 déclenche on fait la mesure.

Si on fixe la a 10ms on veut déterminer l'angle de la prochaine mesure.

La comparaison se fait entre deux registres TENTO et OCROA chaque 0x09 représente 64us approximativement donc 0x9c pour 10ms.

La vitesse tourne de 2000 tr/min

La deuxième vitesse :

$$\begin{array}{lcl} 1\text{tr /min} & \longrightarrow & 2\pi \text{ rad/min} \\ 2000 \text{ tr/min} & \longrightarrow & 4000\pi \text{ rad/min} \end{array}$$

$$\begin{array}{lcl} 1\text{tr/min} & \longrightarrow & 66\pi \text{ rad/s} \\ 2000\text{tr/min} & \longrightarrow & 0.066 \pi \text{ rad/ms} \end{array}$$

$$\begin{array}{lcl} \pi & \longrightarrow & 180 \\ 0.066\pi & \longrightarrow & 11.88 \end{array}$$

Ça veut dire tourne 11.88°/ms donc 118.8°/10ms

Donc si on règle le timer 0 a 10ms il faut que la largeur de la bande de passe selon théorème de Shannon $118 \times 2 = 239^\circ$ ce qui est très large donc si on règle ce timer a 1ms donc 11°/ms la largeur de la bande ne peut pas dépasser 24°.

Pour que l'arduino puisse récupérer les informations du capteur et faire les calculs de la vitesse on va écrire un programme qui va faire les calculs.

Le programme charger dans l'arduino est :

```

1  /*
2  * prjt_infraroug.c
3  *
4  * Created: 12/11/2020 21:26:49
5  * Author : bacim
6  */
7
8  #define F_CPU 16000000UL // Clock Speed
9  #include <avr/io.h>
10 #include <util/delay.h>
11 #include <avr/interrupt.h>
12
13 #define BAUD 9600
14 #define MYUBRR F_CPU/16/BAUD-1
15
16 void USART_Init(unsigned int);
17 void USART_Transmit(unsigned char);
18 unsigned char USART_Receive();
19 void USART_Transmit_String(char*);
20 /**
21 void ADC_init();
22 unsigned int ADC_read(unsigned char canal);
23 /**
24 void init_TIO(void);
25 /**
26 char* intToString(int nbr);
27 /*****
28

```

```

28
29 volatile unsigned int adc_value = 0;
30 volatile int c = 0, cv = 0;
31 volatile int j = 0, jv = 0;
32 int val;
33
34 /**
35 int main(void)
36 {
37     DDRD &= ~(1<<DDD2);
38     PORTD |= (1<<PORTD2);
39     USART_Init(MYUBRR);
40     ADC_init();
41     /**
42     while(PIND & (1<<PIND2));
43     /**
44     _delay_ms(1000);
45     init_TIO();
46     while(PIND & (1<<PIND2))
47     {
48         if(cv != c)
49         {
50             val = 60000/(j-jv);
51             //USART_Transmit_String(intToString(j));
52             //USART_Transmit(' ');
53             USART_Transmit_String(intToString(val));
54             USART_Transmit_String(" tr/min ");
55             //USART_Transmit_String(intToString(c));

```

```

56     USART_Transmit('\n');
57     jv = j;
58     cv = c;
59 }
60 }
61 }
62 //*****
63 char* intToString(int nbr)
64 {
65     char cstr[16];
66     return itoa(nbr, cstr, 10);
67 }
68 //*****
69
70 void init_TIO(void)
71 {
72     TCCR0A |= (1<<WGM01);
73     TCCR0B = 0x05;
74     TIMSK0 |= (1<<OCIE0A);
75     OCR0A = 0x0F; // 10 ms == 0x9C // 1ms = 0x0F
76     sei();
77 }
78
79 //*****
80 ISR(TIMERO_COMPA_vect)
81 {
82     cli();
83     if(!(adc_value < 100 && ADC_read(0) < 100))

```

```

84 {
85     adc_value = ADC_read(0);
86     if(adc_value < 100)
87     {
88         c++;
89     }
90 }
91 j++;
92 sei();
93 }
94
95 //*****
96 void ADC_init()
97 {
98     ADMUX = (1<<REFS0);           // Selecting internal reference voltage
99     ADCSRA=(1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);    // Enable ADC also set Prescaler as 128
100 }
101
102 //*****
103 unsigned int ADC_read(unsigned char canal)
104 {
105     canal = canal & 0b00000111;    // select adc channel between 0 to 7
106     ADMUX |= canal;                // channel A0 selected
107     ADCSRA|=(1<<ADSC);              // start conversion
108
109     while(!(ADCSRA & (1<<ADIF)));    // wait for ADIF conversion complete return
110     ADCSRA|=(1<<ADIF);              // clear ADIF when conversion complete by writing 1
111     return (ADC);                  // return calculated ADC value
112 }
113
114 //*****
115 void USART_Init(unsigned int ubrr)
116 {
117     /* Set baud rate */
118     UBRR0H = (unsigned char) (ubrr>>8);
119     UBRR0L = (unsigned char)ubrr;
120     /* Set frame format: 8data, 1stop bit */
121     UCSRC = (3<<UCSZ00) | (0<<UMSEL00);
122     /* Enable receiver and transmitter */
123     UCSRB = (1<<TXEN0) | (1<<RXEN0);
124 }
125
126 //*****
127 void USART_Transmit(unsigned char data)
128 {
129     /* Wait for empty transmit buffer */
130     while (!(UCSR0A & (1<<UDRE0)));
131     /* Put data into buffer, sends the data */
132     UDR0 = data;
133 }
134
135

```

```

135
136 //*****
137 unsigned char USART_Receive( void )
138 {
139     /* Wait for data to be received */
140     while (!(UCSR0A & (1<<RXC0)));
141     /* Get and return received data from buffer */
142     return UDR0;
143 }
144 //*****
145 void USART_Transmit_String(char* sendString)
146 {
147     /* Send string */
148     while(*sendString != 0x00)
149     {
150         USART_Transmit(*sendString);
151         sendString++;
152     }
153 }

```

Le signal entré par l'arduino dans l'ADC0 est le suivant :

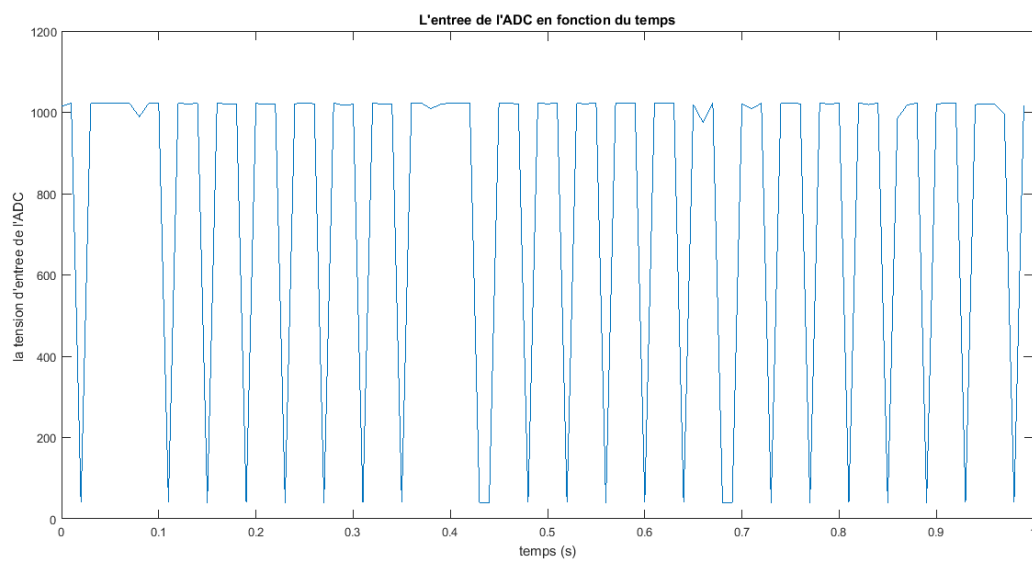
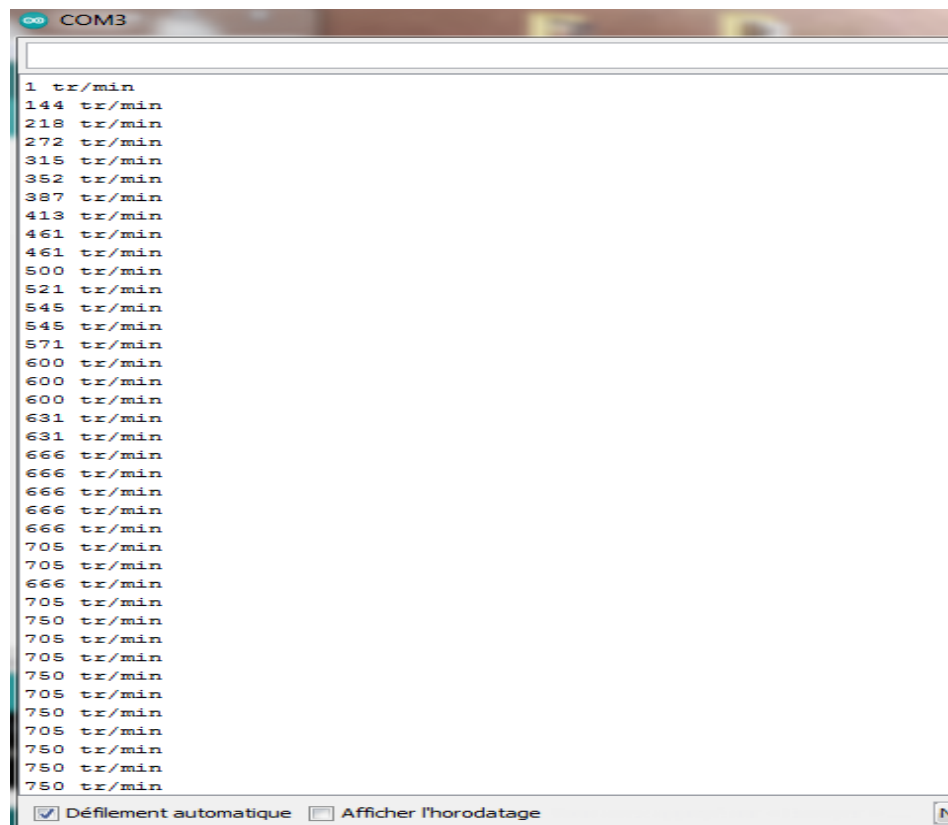


Figure 2.10: Signal d'entrée de l'ADC.

La vitesse du moteur détecté par le capteur avec alimentation est la suivante :



```
COM3
1 tr/min
144 tr/min
218 tr/min
272 tr/min
315 tr/min
352 tr/min
387 tr/min
413 tr/min
461 tr/min
461 tr/min
500 tr/min
521 tr/min
545 tr/min
545 tr/min
571 tr/min
600 tr/min
600 tr/min
600 tr/min
631 tr/min
631 tr/min
666 tr/min
666 tr/min
666 tr/min
666 tr/min
666 tr/min
705 tr/min
705 tr/min
666 tr/min
705 tr/min
750 tr/min
705 tr/min
705 tr/min
750 tr/min
705 tr/min
750 tr/min
705 tr/min
750 tr/min
750 tr/min
750 tr/min
750 tr/min
```

☒ Défilement automatique ☐ Afficher l'horodatage

Figure 2.11: Affichage de la vitesse du moteur avec alimentation.

La vitesse du moteur détecté par le capteur sans alimentation est la suivante :

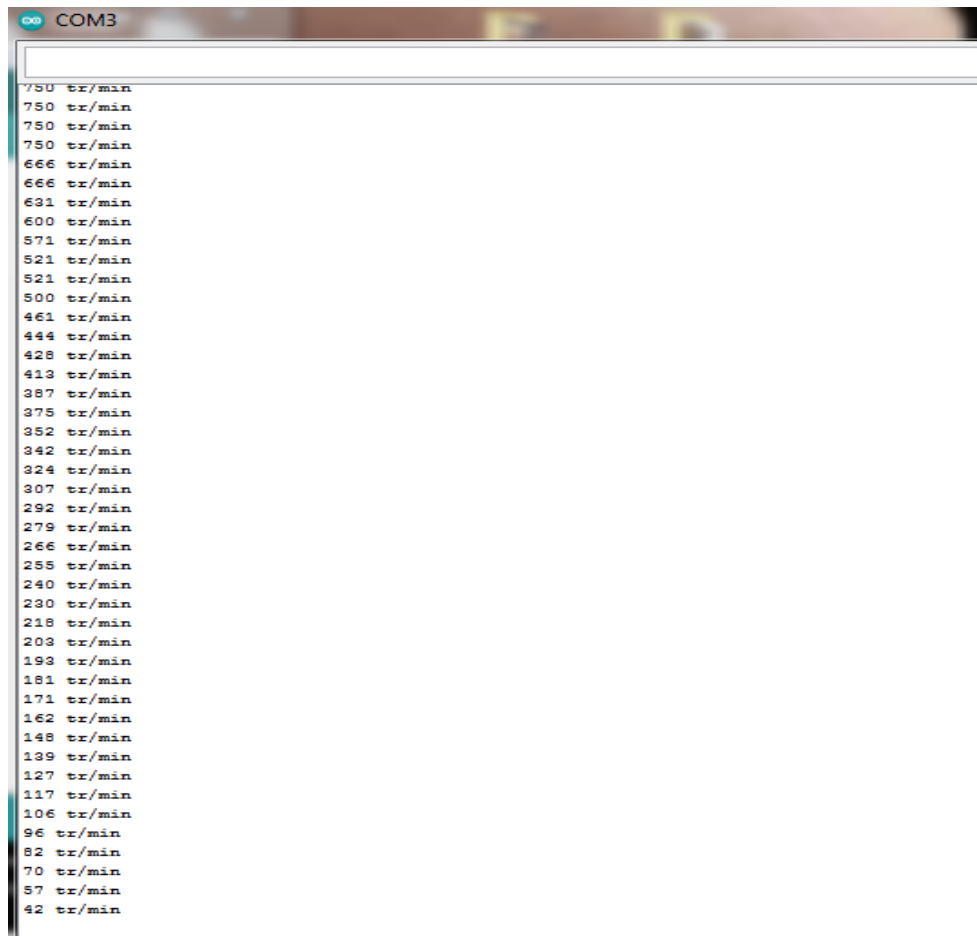


Figure 2.12: Affichage de la vitesse du moteur sans alimentation.

Troisième chapitre :

Ce chapitre est la suite du projet capteur intelligent , on va faire l'identification de la réponse indicielle de notre moteur.

La réponse obtenue est la suivante :

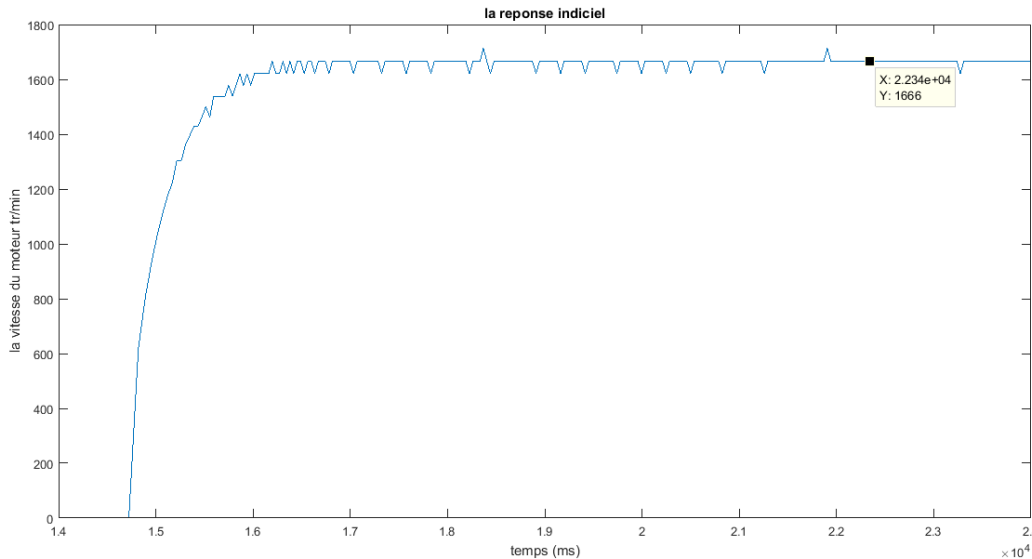


Figure 3.1: Réponse indicielle du moteur.

Pour cela on va utiliser la méthode de broïda.

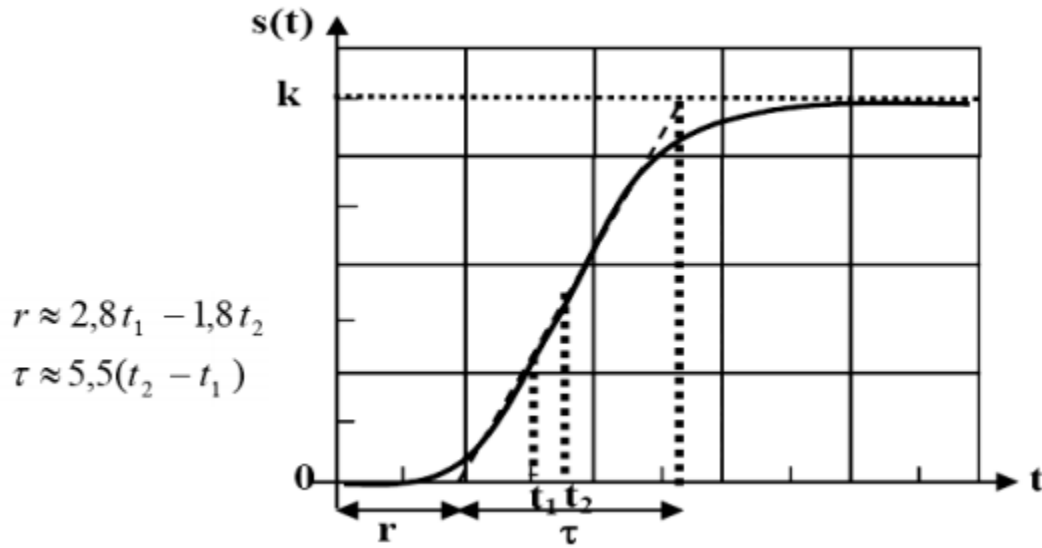
1. Identification d'un modèle avec retard : (Méthode de Broïda) :

Pour un système naturellement stable ou autorégulant Le modèle proposé pour approcher le comportement du système est un premier ordre plus retard pur. Sa FT est :

$$H(p) = \frac{k e^{-\tau p}}{1 + \tau p}$$

(1) : La première solution consiste à tracer la tangente de la courbe au point d'inflexion, puis à mesurer le retard r et la constante de temps τ .

(2) : Une autre approche, appelée méthode de Broïda, consiste à mesurer les instants t_1 et t_2 aux quels la réponse atteint respectivement 28% et 40% (voir Fig.1) de sa valeur finale. La constante de temps et le retard sont alors obtenus par les relations suivantes :



Le modèle de Broïda donne un modèle correct si $\tau > 4r$.

(3) : Une approche analogue consiste à considérer les instants t_3 et t_4 aux quels la réponse atteint respectivement 35,5% et 85,3% (voir Fig. 1) qui présente l'intérêt d'être plus nettement séparés que dans le cas précédent. On alors :

$$r \approx 1,3 t_3 - 0,29 t_4$$

$$\tau \approx 0,67 (t_4 - t_3)$$

Dans ce qui suit, nous allons établir un modèle mathématique de notre système étudié en utilisant la méthode d'identification empirique de Broïda.

Pour la détermination des paramètres τ et T caractéristiques de la fonction, cette méthode s'appuie sur deux points caractéristiques de la réponse du système. Les points t_1 et t_2 choisis correspondent aux points d'ordonnées égales à 28% et à 40% respectivement de la valeur finale de la sortie.

Pour effectuer cette opération on va écrire sur matlab le programme suivant :

```
function [K,tau,r,t1,t2]=broïda(t,y,ymax,a,ah)
K = ymax/a;           % Gain statique
tmax = max(t);

idx_28 = find(y<0.28*ymax,1,'last');    % indice de 28% de ymax
if exist('idx_28','var')

    idx_40 = find(y<0.40*ymax, 1, 'last'); % max(find(y<0.40*ymax));
    t1 = t(idx_28);           % t à 28% de ymax
    t2 = t(idx_40);           % t à 40% de ymax
    % calcul
    tau = 5.5*(t2-t1);         % constante du temps
    r = 2.8*t1 - 1.8*t2;       % retard
    if exist('ah','var')
        axes(ah);
        hold(ah,'off');
        plot(ah,NaN,NaN);      % vider la figure
        m=max(y);
        if m >= a
            set(ah,'ylim',[-0.2*ymax 1.1*m]);
        else
            set(ah,'ylim',[-0.2*ymax 1.1*a]);
        end
        set(ah,'xlim',[0 4*t2]);
        hold(ah,'on');
        plot(ah,[0 tmax],[a a],'r'); % tracé de la consigne
        plot(ah,t,y,'k','LineWidth',2); % tracé du signal y
        plot(ah,[t1 t1],[0.0 y(idx_28)],'--g'); % tracé verticale à t1
        plot(ah,[t2 t2],[0.0 y(idx_40)],'--r'); % tracé verticale à t2
        text(t1-0.04,-0.05*ymax, strcat('\uparrow t1=',num2str(t1),'s'));
        text(t2,y(idx_40), strcat('\leftarrow t2=',num2str(t2),'s'));
        legend(ah,'consigne','vitesse','t1','t2','Location','Best');
        plot(ah,[0 tmax],[0 0],'k'); % axe du temps en noir (y=0)
        title('Identification par la méthode de Broïda');
        xlabel(ah,'Temps (s)');
        ylabel(ah,'Vitesse (tr/min)');
        set(ah,'YGrid','on');
        hold(ah,'off');
    end
end
end
```

Le résultat obtenu est le suivant :

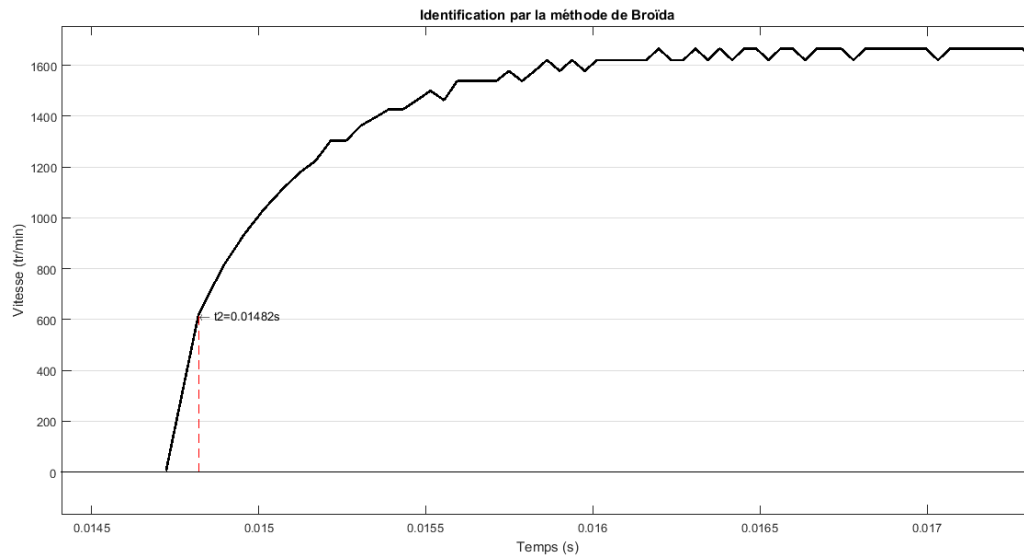


Figure 3.2: Identification par la méthode de broida.

Les paramètres obtenus sont les suivants :

$K = 1714.$

$\text{Tau} = 5.3350 \cdot 10^{-4}.$

$R = 0.0145.$

$t1 = 0.0147.$

$t2 = 0.0148.$

Donc notre système peut être écrit de la manière suivante :

$$G(s) = \frac{1714 * e^{0.00055s}}{1 + 5.335 * 10^{-4} * s}$$

Donc grâce à la méthode de broida on a pu identifier notre système et l'écrire sous forme une fonction de transfert $G(s)$.

Conclusion :

Dans notre projet, on présente la détection de la vitesse du moteur avec un capteur infrarouge et l'identification de sa réponse indicielle.

Dans le premier chapitre, nous avons rappelé les principaux éléments constituant la machine à courant continu, les différents types des capteurs ainsi leur classification.

Dans le deuxième chapitre on a pu détecter la vitesse du moteur à l'aide de la carte arduino et un capteur infrarouge.

Et enfin dans le troisième chapitre on a fait l'identification de la réponse indicielle du notre moteur.

Cette réalisation nous a permis de :

- Déterminer la vitesse du moteur à l'aide d'un capteur infrarouge.
- Identification de la réponse du moteur.

Un capteur peut être classifié selon leur principe de fonctionnement :

-capteur actif.

-capteur passif.

-capteur intelligent.

Un capteur est caractérisé selon plusieurs critères dont les plus courants sont :

- la grandeur physique observée ;
- son étendue de mesure (gamme de mesure) ;
- sa sensibilité ;
- sa résolution ;
- sa précision ;
- sa reproductibilité ;
- sa linéarité ;
- son temps de réponse ;
- sa bande passante ;
- son hystérésis ;
- sa gamme de température d'utilisation.

Pour utiliser un capteur dans les meilleures conditions, il est souvent utile de pratiquer un étalonnage et de connaître les incertitudes de mesure relatives à celui-ci.