Rapport de la séance 15

2 avril 2024

WARTSKI NARANJO Daniel

Robotique

Travail réalisé

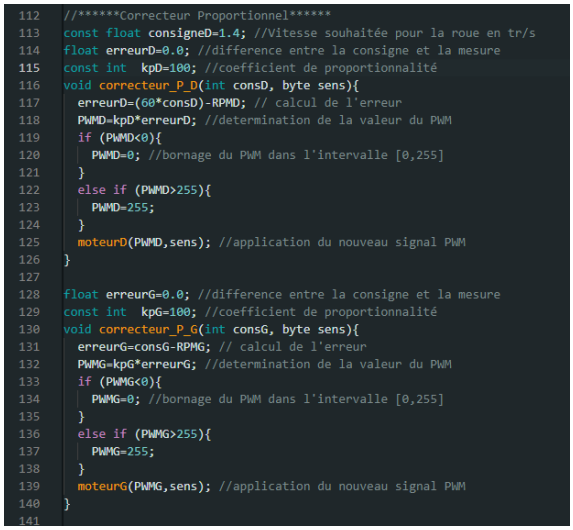
Lors de cette séance, une grande partie de mon temps a été consacrée à résoudre un problème de connexion qui empêchait l'exportation du programme. Après une longue recherche, j'ai identifié un simple problème de faux contact lors du soudage du Shield de la carte Mega. Une fois ce problème de faux contact réglé, j'ai réinstallé le Shield dans la carte Mega et le problème a été résolu.

Ensuite, j'ai poursuivi avec le développement du programme PID pour mon robot. Après avoir apporté des modifications au programme, j'ai réussi à obtenir les valeurs de vitesse de ma roue en tours par seconde. Cependant, j'ai remarqué qu'en modifiant la constante de proportionnalité, la vitesse ne variait pas comme prévu et n'atteignait pas la vitesse attendue. Par conséquent, je vais devoir continuer à travailler sur le programme afin de trouver et de corriger ce problème, ainsi que pour déterminer les constantes de proportionnalité, d'intégration et de dérivabilité appropriées. Les changements apportés au programme PID peuvent être consultés dans les images ci-jointes.

```
int distance;
int v=120;
String a="s";

int PWMD=0;
int PWMG=0;
unsigned int compteurD=0;
unsigned int compteurG=0;
unsigned long int deltatemps=200;
float const Mult=1/(deltatemps*0.001);
float RPMD;
float RPMG;

void setup() {
  // put your setup code here, to run once:
  Timer1.initialize(deltatemps*1000);
  Timer1.attachInterrupt(CalculD);
  //Timer1.attachInterrupt(CalculG);
  Serial.begin(115200);
  pinMode(M1A,OUTPUT);
  pinMode(M2A,OUTPUT);
  pinMode(M1B,OUTPUT);
  pinMode(M2B,OUTPUT);
  pinMode(IR1,INPUT);
  pinMode(IR2,INPUT);
  pinMode(IR3,INPUT);
  pinMode(echo,INPUT);
  pinMode(trig,OUTPUT);
  attachInterrupt(digitalPinToInterrupt(IRPIDR),changeD,RISING);
  attachInterrupt(digitalPinToInterrupt(IRPIDL),changeG,RISING);
  stop();
  delay(1000);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
correcteur_P_D(1.4, 1);
}

void stop(){···
}

//Calcul de la vitesse des moteur en tr/s
void CalculD(){
  RPMD=Mult*compteurD/8;
  //Serial.println(millis());
  //Serial.print(" ");
  Serial.println(RPMD/60,3);
  compteurD=0;
}

void CalculG(){
  RPMG=Mult*compteurD/8;
  Serial.println(millis());
  Serial.print(" ");
  Serial.println(RPMG/60,3);
  compteurG=0;
}

//******Interruptions******
void changeD(){
  compteurD++;
}

void changeG(){
  compteurG++;
}
```

```
//******Fonction pour le moteur droit******
void moteurD(int v, byte sens){
  if (sens==0){
    analogWrite(M2A,0);
    analogWrite(M2B,v);
  }
  else{
    analogWrite(M2B,0);
    analogWrite(M2A,v);
  }
}

//******Fonction pour le moteur gauche******
void moteurG(int v, byte sens){
  if (sens==0){
    analogWrite(M1A,0);
    analogWrite(M1B,v);
  }
  else{
    analogWrite(M1B,0);
    analogWrite(M1A,v);
  }
}
```



```
//******Correcteur Proportionnel******
const float consigneD=1.4; //Vitesse souhaitée pour la roue en tr/s
float erreurD=0.0; //difference entre la consigne et la mesure
const int  kpD=100; //coefficient de proportionnalité
void correcteur_P_D(int consD, byte sens){
  erreurD=(60*consD)-RPMD; // calcul de l'erreur
  PWMD=kpD*erreurD; //determination de la valeur du PWM
  if (PWMD<0){
    PWMD=0; //bornage du PWM dans l'intervalle [0,255]
  }
  else if (PWMD>255){
    PWMD=255;
  }
  moteurD(PWMD,sens); //application du nouveau signal PWM
}

float erreurG=0.0; //difference entre la consigne et la mesure
const int  kpG=100; //coefficient de proportionnalité
void correcteur_P_G(int consG, byte sens){
  erreurG=consG-RPMG; // calcul de l'erreur
  PWMG=kpG*erreurG; //determination de la valeur du PWM
  if (PWMG<0){
    PWMG=0; //bornage du PWM dans l'intervalle [0,255]
  }
  else if (PWMG>255){
    PWMG=255;
  }
  moteurG(PWMG,sens); //application du nouveau signal PWM
}
```

Objectifs pour la prochaine séance

- Poursuivre la mise en place du programme pour intégrer le correcteur PID.
- Continuer le développement du programme du robot afin qu'il réponde aux exigences spécifiées.