



Soutenance de Projet Long

2016 - 2017

Yizhe FAN Si CHEN
Master Informatique
Université Paris Diderot



Index

- Introduction
- Des fonctionnalités
- Architecture, Conception et Gestion
- Programmation
- Conclusion

Introduction

Suiveur de ligne

Nous intéressons une technique de l'automatisation des processus robotiques qui est suiveur de ligne. Ces robots ne sont pas seulement des robots qui servent lors de tournois, mais servent aussi dans l'industrie ou encore dans les transports en commun.

Développer sur un robot mindstorms pour:

- réaliser un algorithme intelligent qui suit les chemin prédéfini.
- choisir les chemin par couleur différentes

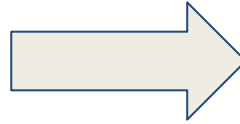
Nous utilisons un certain nombre de techniques pour suiveur de ligne:

- Capteur couleur traite données de couleur ainsi que il distingue couleur semblable.
- Technique de écriture & lecture fichier fait conservation permanente des données.
- Algorithme PID(proportionnel, intégrateur, dérivateur) exécute suiveur de ligne.
- Un algorithme choisit les chemin.

Introduction



Lego Mindstorms NXT



Google Driverless Car

Des possibles usages:

- [L'industrie](#)
- [Transports en commun \(bus sans chauffeur\)](#)
- [Livrer ou récupérer des colis aux dépôts fixes.](#)
- [Guider des personne aveugle.](#)

Présentation des fonctionnalités

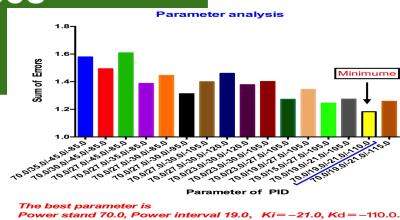
Étalonnage



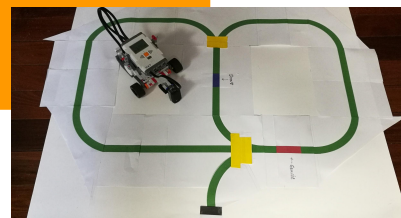
Suivre des différentes lignes



Stocker des datas à analysés

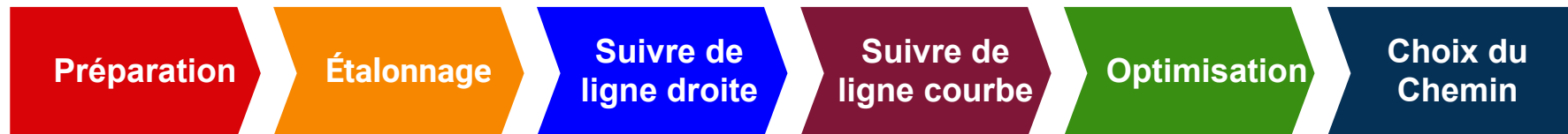


Choix du Chemin



Architecture, Conception et Gestion de Projet

Plan du Développement



Durée Total : 12 Séances

1 Séance

Monter le robot

Test les moteurs

2 Séances

Tester le capteur

Lire les couleurs

Stocker les
couleurs

Reconnaître les
couleurs

2 Séances

Suivre la ligne
droite par un
capteur

3 Séances

Suivre la ligne
courbe par deux
capteur

Suivre la ligne
courbe par un
capteur avec PID

2 Séances

Optimisation
des paramètres du
PID

2 Séances

Choix du chemin
selon la couleur

Architecture, Conception et Gestion de Projet

Les compétences techniques nécessaires

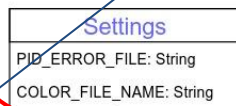
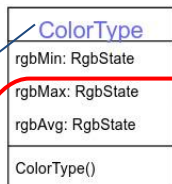
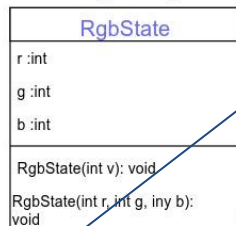
- Programmation avec Java.
- Connaissance et utilisation de librairie Lejos.
- Utiliser des outils pour gérer des codes.
- Travailler en l'équipe.
- Des connaissances de Mathématiques pour traiter les couleurs.

Les Dépendances



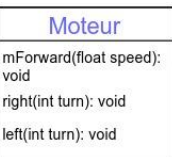
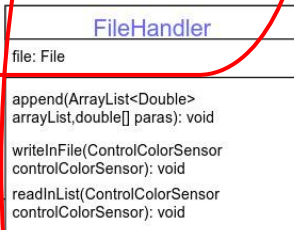
Architecture, Conception et Gestion de Projet

RgbState et ColorType sont types des données pour stocker variable de rouge, vert et bleu



Package:
indi.fan_chen.pl.model

FileHandler utilise nom de fichier

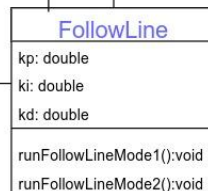


contrôler les moteur



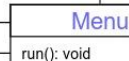
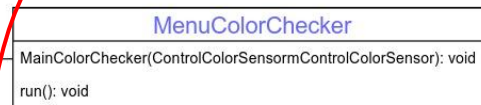
verifier couleur

écrire et lire données dans le fichier



algorithme de
suivre la ligne

Package:
indi.fan_chen.pl.view



Package:
indi.fan_chen.pl.controller

Architecture, Conception et Gestion de Projet

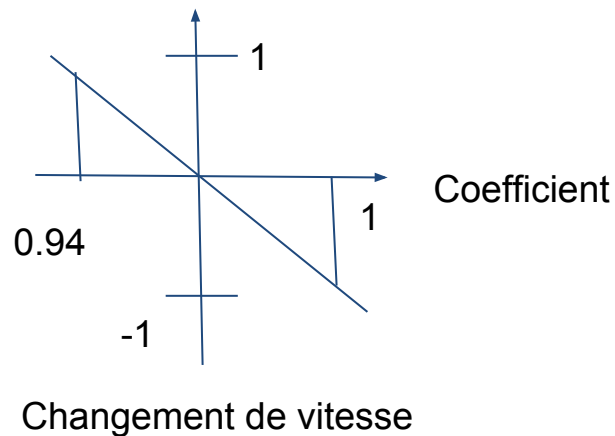
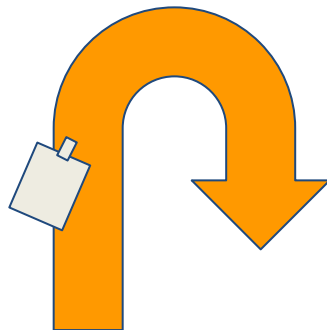
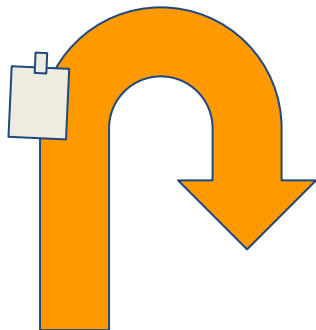
La difficulté technologique rencontrée



Le suiveur de ligne ne pouvait pas suivre le bord de la ligne courbe précisément.



Contrôleur PID



Architecture, Conception et Gestion de Projet

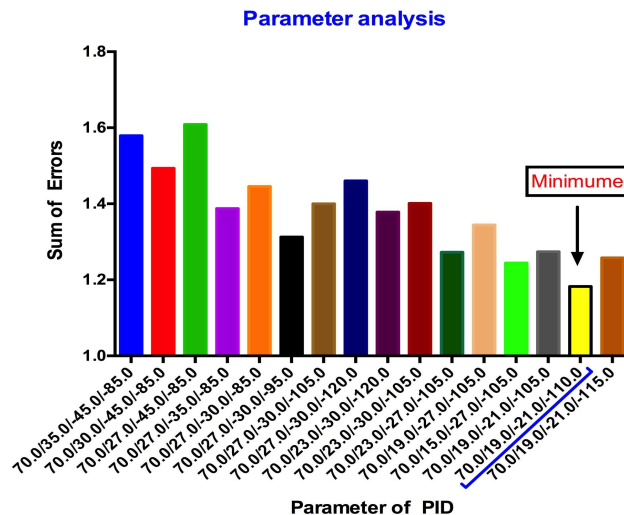
La difficulté technique rencontrée



Chercher les meilleurs paramètres de PID pour un cadre fixe.



Calculer la somme des erreurs



The best parameter is

Power stand 70.0, Power interval 19.0, $K_i = -21.0$, $K_d = -110.0$.

Programmation

```

1
2  /*
3   PID Controller
4  */
5  private void pid(ColorSensor.Color vals){
6      //calculer le nouveau coefficient de la couleur que le capteur détecte dans la couleur de la ligne.
7      double corNew = calculCor(vals.getRed(), vals.getGreen(), vals.getBlue(), line, lineColorSqrt);
8
9      //calculer la distance entre ce nouveau coefficient et le coefficient offset.
10     // le coefficient offset est le médium du coefficient calculé par la couleur fonde et la couleur de la ligne.
11     double newError = corNew - offSet;
12
13     //calculer la dérivée
14     //dérivée est la tendance de changement de distance entre la capteur et le bord de la ligne.
15     double derivative = newError - error;
16
17     //si la distance entre nouveau coefficient et offset a le signe contraire.
18     //cela veut dire la capteur a bougé dans l'autre côté du bord de la ligne.
19     //Donc on met la valeur de intégrale à 0, et on la re-augmente.
20     if ( (newError * error) < 0 ){
21         integral = 0;
22     }
23
24     //noté ce nouveau coefficient pour comparer avec le coefficient la prochaine fois.
25     error = newError;
26
27     //augment l'intégrale
28     integral += error;
29
30     //avec le proportion, l'intégrale et la dérivée. on calculer le changment de la vitesse de les deux moteurs
31     double turn = powerVal * kp * error + ki * integral + kd * derivative;
32
33     //mise à jours des vitesse de les deux moteurs.
34     double powerA = powerStandard - turn;
35     double powerB = powerStandard + turn;
36     changePower(powerA, ma);
37     changePower(powerB, mb);
38 }

```

Conclusion

Des connaissances apprises

- Programmation sur un robot.
- Traiter des données des couleurs.
- Théorie et pratique du contrôleur du PID.
- La façon pour analyser des données.

Pour Version 2

- Algorithme plus intelligent pour choisir le chemin au carrefour.
- Une fonctionnalité pour mémoriser la trace du robot à dessiner un plan.



