

Kilobot

Implémentation d'algorithmes pour les cohortes de robots

Benjamin Bielle
Arnaud Guermont

Université Pierre et Marie Curie

12 mai 2015

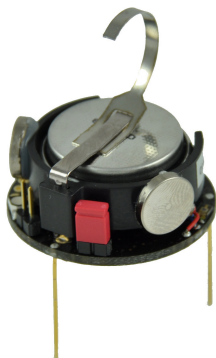


Vue d'ensemble

- 1 Spécifications des kilobots
- 2 Simulateurs
- 3 Phase I
 - Firmware
- 4 Phase II
 - Algorithme du phototaxis
 - Algorithme du gradient
- 5 Phase III
 - CORDA
 - 1^{ère} Approche
 - 2^e Approche
 - Pistes Expérimentales

Spécifications des kilobots

La plate-forme Kilobot est définie par les caractéristiques suivantes :

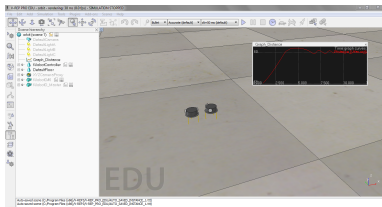


- Communication par infrarouge
- Déplacement par vibration
- Mesure de la lumière ambiante
- Essaim contrôlable par un seul opérateur
- Faible coût de fabrication

Simulateurs

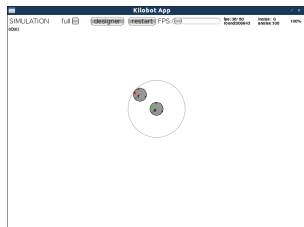
V-Rep

- ✓ Polyvalent
- ✓ Programmation par scripts LUA
- ✓ Moteur physique avancé
- ✗ Ne supporte pas la mise à l'échelle



KbSim

- ✓ Simulateur dédié
- ✓ Programmation Python
- ✓ Léger
- ✓ Simulation d'un grand nombre de kilobots



Phase I

Objectifs

- ▶ Recherche documentaire
- ▶ Prise en main

Phase I

Prise en main

Algorithme de l'orbite

Un robot dessine une orbite autour d'un autre et maintient sa distance de message

Algorithme de synchronisation

Chaque robot envoie son horloge et l'ajuste en fonction des messages reçus

Caractéristiques des firmwares disponibles :

| | K-Team | Kilobotics |
|--------------------|--------|------------|
| Compatible Linux | ✗ | ✓ |
| Open source | ✗ | ✓ |
| Fonctions avancées | ✗ | ✓ |

Flashage

Flashages des kilobots et du contrôleur via l'outil AVRDUDE

Phase II

Objectifs

- ▶ Implémentation de bio-algorithmes
 - Algorithme du phototaxis
 - Algorithme du gradient

Phase II

Algorithme du phototaxis

Pourquoi ?

Propose un bon exemple de comportement de groupe (d'insectes) observé dans la nature

Phase II

Algorithme du phototaxis

Pourquoi ?

Propose un bon exemple de comportement de groupe (d'insectes) observé dans la nature

Spécifications

Chaque robot capte la source lumineuse et ajuste ses déplacements vers celle-ci

Phase II

Algorithme du phototaxis

Pourquoi ?

Propose un bon exemple de comportement de groupe (d'insectes) observé dans la nature

Spécifications

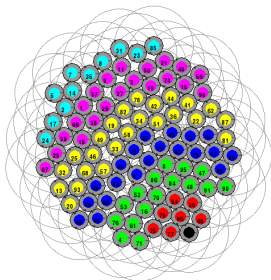
Chaque robot capte la source lumineuse et ajuste ses déplacements vers celle-ci

Contraintes

- ▶ Environnement à lumière ambiante réduite
- ▶ Source de lumière dirigée

Phase II

Algorithme du gradient

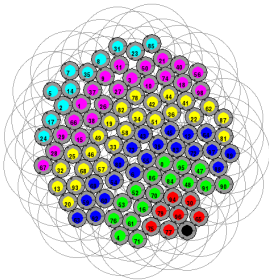


Pourquoi ?

Bio-algorithme préambule à la phase III

Phase II

Algorithme du gradient



Pourquoi ?

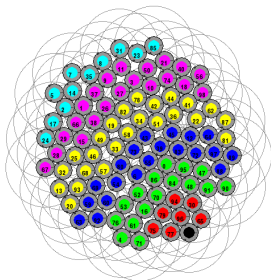
Bio-algorithme préambule à la phase III

Spécifications

Chaque robot calcule la distance qui le sépare de la balise et affiche une couleur en fonction de celle-ci

Phase II

Algorithme du gradient



Pourquoi ?

Bio-algorithme préambule à la phase III

Spécifications

Chaque robot calcule la distance qui le sépare de la balise et affiche une couleur en fonction de celle-ci

Contraintes

Plusieurs robots ont le rôle de balise

Phase III

Objectif

Implémentation d'un modèle robotique : le modèle **CORDA**

Pourquoi ?

Il est largement utilisé pour les algorithmes répartis dans le domaine de la robotique

Description

Le modèle CORDA comprend un cycle de 3 phases : Voir, Calculer, Agir

Contraintes

| | CORDA | Kilobot |
|--------------------|-------|---------|
| Communication | ✗ | ✓ |
| Vision | ✓ | ✗ |
| Repère orthonormé | ✓ | ✗ |
| Déplacement précis | ✓ | ✗ |

Phase III

1^{ère} Approche

Résultat attendu

Localisation des robots

Résultat attendu

Localisation des robots

Comment ?

Utilisation de la triangulation à l'aide de robots fixes ayant le rôle de balise

Résultat attendu

Localisation des robots

Comment ?

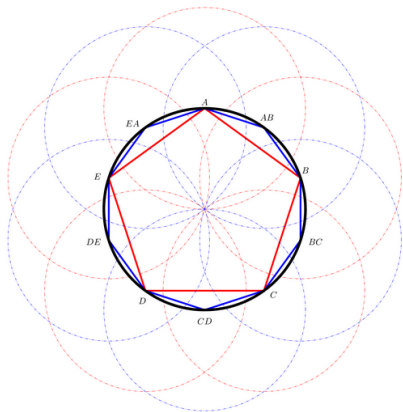
Utilisation de la triangulation à l'aide de robots fixes ayant le rôle de balise

Contrainte

La distance d'émission des kilobots est limitée à 7cm

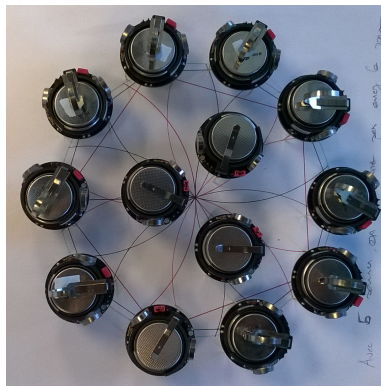
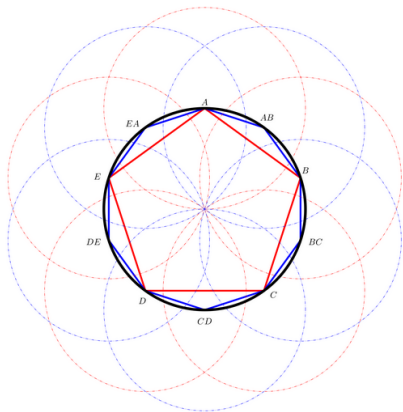
Phase III

Résultat obtenu



Phase III

Résultat obtenu



Résultat attendu

Localisation des robots avec un nombre réduit de balises tout en augmentant la portée d'émission

Comment ?

Utilisation de la méthode du gradient associée à la méthode vue dans la première approche

API

Les primitives envisagées sont :

API

Les primitives envisagées sont :

- ▶ **getPosition** qui implémente la seconde approche

API

Les primitives envisagées sont :

- ▶ **getPosition** qui implémente la seconde approche
- ▶ **getVision** qui permet de connaître la position de ses voisins

API

Les primitives envisagées sont :

- ▶ **getPosition** qui implémente la seconde approche
- ▶ **getVision** qui permet de connaître la position de ses voisins
- ▶ **toPosition** qui permet de se rendre à une position donnée

API

Les primitives envisagées sont :

- ▶ **getPosition** qui implémente la seconde approche
- ▶ **getVision** qui permet de connaître la position de ses voisins
- ▶ **toPosition** qui permet de se rendre à une position donnée

API

Les primitives envisagées sont :

- ▶ **getPosition** qui implémente la seconde approche
- ▶ **getVision** qui permet de connaître la position de ses voisins
- ▶ **toPosition** qui permet de se rendre à une position donnée

Problème

- ▶ Consommation mémoire pour la détection d'un cycle trop importante
- ▶ Implémentation des autres primitives menacées par le manque d'espace mémoire

- ▶ Travail réalisé
 - Prise en main des kilobots
 - Implémentation d'algorithmes démonstratifs
 - Implémentation de bio-algorithmes
- ▶ Travail à long terme
 - Recherches de solutions pour l'implémentation du modèle CORDA