

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Le projet Swarmanoid et l'intégration du <i>cart-bot</i></b>	<b>7</b>
2.1	Projet Swarmanoid . . . . .	7
2.1.1	Foot-bot . . . . .	8
2.1.2	Hand-bot . . . . .	9
2.1.3	Eye-bot . . . . .	11
2.1.4	Développement actuels . . . . .	12
<b>3</b>	<b>Cahier des charges, contraintes de réalisation</b>	<b>13</b>
3.1	Fonctionnalités à mettre en œuvre . . . . .	14
3.2	Compatibilité avec les autres robots du <i>swarmanoid</i> . . . . .	14
3.3	Utilisation aisée . . . . .	16
3.4	Contraintes de réalisation . . . . .	17
<b>4</b>	<b>Solutions envisagées</b>	<b>18</b>
4.1	Être transportable . . . . .	19
4.1.1	Système d'arrimage . . . . .	19
4.1.2	Élimination des frottements . . . . .	20
4.1.3	Sécurisation du contenu . . . . .	21
4.2	Regroupement de plusieurs <i>book-bots</i> . . . . .	22
4.2.1	Stockage . . . . .	22
4.2.2	Chargement et déchargement . . . . .	25
4.3	Communiquer . . . . .	27

4.4	Utiliser des outils de développement déjà maîtrisés . . . . .	28
4.5	Être réutilisable rapidement . . . . .	28
<b>5</b>	<b>Stratégie retenue et conception du <i>cart-bot</i></b>	<b>30</b>
5.1	Stratégie retenue . . . . .	30
5.2	Stockage . . . . .	32
5.3	Élévation . . . . .	33
5.4	Chargement/déchargement . . . . .	34
<b>6</b>	<b>Conclusion</b>	<b>37</b>

# Table des figures

2.1	Le <i>foot-bot</i> . . . . .	8
2.2	Le <i>hand-bot</i> . . . . .	9
2.3	<i>hand-bot</i> transporté par des <i>foot-bots</i> . . . . .	10
2.4	Prototype de développement du <i>eye-bot</i> . . . . .	11
2.5	Prototype de développement de la tourelle de vision du <i>eye-bot</i> . . . . .	11
4.1	Système d'arrimage sur les <i>foot-bots</i> . . . . .	19
4.2	Pince du <i>hand-bot</i> montée sur parallélogramme. . . . .	23
4.3	Principe du système de chargement/déchargement : le chariot élevateur. . . . .	26
4.4	Capteur de couleur de chez TAOS. . . . .	27
4.5	<i>e-puck</i> , base d'inspiration pour l'électronique du <i>cart-bot</i> . . . . .	29
5.1	le système d'élévation dans le <i>cart-bot</i> . . . . .	31
5.2	actions d'un bras du mécanisme de chargement/déchargement. . . . .	32
5.3	principe de fonctionnement du mécanisme de chargement/dé- chargement de <i>book-bots</i> . . . . .	33
5.4	composants du système d'élévation. . . . .	34
5.5	le bloc moteur fixé à la base du <i>cart-bot</i> . . . . .	35
5.6	système de freinage pour bloquer la base à la hauteur souhaitée. . . . .	35
5.7	système de chargement/déchargement. . . . .	36

# Chapitre 1

## Introduction

L'intelligence en essaim ou *swarm-intelligence* est un type d'intelligence artificielle inspiré des comportements d'animaux sociaux [1]. Elle est basée sur des systèmes multi-agents où chaque agent est relativement simple, indépendant, et avec une vue limitée du système. Le fonctionnement en est décentralisé : chaque agent prend ses décisions de manière autonome. C'est la collaboration entre ces agents qui permet de résoudre des problèmes complexes. L'intelligence en essaim, grâce à ces caractéristiques, permet une mise à l'échelle des solutions. C'est à dire que le système peut être adapté à une augmentation de la taille du problème par l'ajout de nouveaux agents.

La robotique en essaim ou *swarm-robotics* est une application de ces concepts à la robotique. Elle utilise la collaboration de robots simples et peu volumineux, mais en grandes quantités, pour accomplir des tâches qu'un robot seul ne pourrait pas réaliser. Il s'agit d'un domaine de recherche relativement récent, initié au début des années 80. Les caractéristiques que l'on souhaite exploiter découlent de celles de l'intelligence en essaim, à savoir une plus grande fiabilité, robustesse et adaptabilité qu'en robotique classique et une capacité de mise à l'échelle. La fiabilité est améliorée par l'utilisation de robots plus petits et plus simples, ce qui permet aussi une diminution des coûts. Les gains en robustesse et adaptabilité découlent de l'utilisation de plusieurs robots interchangeables et du contrôle décentralisé. Ils permettent une adaptation dynamique à tout



(a)



(b)

FIG. 1.1: Transport collaboratif d'un objet lourd par (a) un groupe de fourmis, et par (b) des *s-bots* du projet *Swarm-bots*.

changement local de l'environnement et une tolérance à la défaillance d'un ou plusieurs individus.

Le transport collaboratif est un problème classique et bien étudié en robotique collective [14, 19, 23, 8, 2]. Il correspond au transport d'un objet par un groupe de robots et est requis lorsque l'objet est trop lourd ou trop volumineux pour être déplacé par un seul robot. De nouveau, la résolution de ce type de problème tire son inspiration des systèmes biologiques et en particulier des colonies d'insectes [1, 12, 13]. La figure 1.1 illustre le transport collaboratif par des fourmis, et appliqué en robotique en essaim.

Des études sur les fourmis ont montré une plus grande efficacité dans le transport collaboratif d'une certaine masse que dans le transport individuel de la même masse répartie après découpe [15, 22]. Après le développement d'un modèle de transport collaboratif basé sur le comportement de ces insectes sociaux [12], des expériences ont démontré la validité de ce modèle appliqué à de vrais robots [11]. Des expériences supplémentaires ont alors pu montrer une croissance super-linéaire dans les performances de transport avec l'augmentation du nombre de robots impliqués. La masse qu'un groupe de robots peut transporter est supérieure à la somme des masses que chaque robot peut transporter individuellement. Il faut cependant noter qu'au delà d'un certain nombre de robots collaborant, l'efficacité chute drastiquement. Les problèmes de coor-

dination entre les robots deviennent trop complexes.

Lorsqu'un ensemble de petits objets est à transporter, il mobilise généralement un robot par objet, même si cet objet est de poids nettement inférieur à la capacité de transport du robot. De nombreux robots se retrouvent alors monopolisés pour une tâche non collaborative, ce qui entraîne une perte d'efficacité du système distribué.

Nous proposons avec ce travail une approche pour améliorer les performances d'un groupe de robots collaboratifs dans le transport d'un ensemble d'objets peu volumineux par rapport aux dimensions des robots. Les objets à transporter sont rassemblés sur une palette de transport intelligente qui peut ensuite être transportée en tant que simple entité.

Ce travail s'intègre dans le cadre du projet *Swarmanoid*[4]. Dans ce projet, les fonctionnalités sont distribuées entre groupes de robots spécialisés. Un essai, appelé *swarmanoid*, est constitué de *eye-bots*, robots volants capables d'explorer une scène et de localiser des objets ; de *hand-bots*, robots de manipulation capables de grimper en hauteur et d'attraper des objets ; et de *foot-bots*, robots capable de rouler sur de nombreux types de terrains et de transporter des objets ou d'autres robots. Nous présentons ici le développement d'un quatrième type de robots dans le respect des concepts et des contraintes liés au projet *Swarmanoid*, le *cart-bot*. Les fonctions de celui-ci sont de recevoir, stocker, et restituer un certain nombre d'objets, ainsi que de se faire transporter efficacement par le moins possible de *foot-bots*.

La suite de ce document est divisée en 5 parties. Le chapitre 2 présente le contexte de cette recherche, c'est à dire le projet *Swarmanoid*, et l'intégration du *cart-bot* dans ce projet. Dans le chapitre 3 nous dressons un cahier des charges pour la conception du *cart-bot*. Le chapitre 4 nous permet d'exposer les différentes solutions envisagées pour son développement, afin de mieux justifier la stratégie retenue. Dans le chapitre 5 nous présentons cette stratégie, ainsi que la conception du *cart-bot* qui en découle. Dans le dernier chapitre nous concluons.

# Chapitre 2

## Le projet Swarmanoid et l'intégration du *cart-bot*

### Sommaire

---

<b>2.1</b>	<b>Projet Swarmanoid</b>	<b>7</b>
2.1.1	Foot-bot	8
2.1.2	Hand-bot	9
2.1.3	Eye-bot	11
2.1.4	Développement actuels	12

---

### 2.1 Projet Swarmanoid

Swarmanoid est un projet de recherche en intelligence artificielle destiné à étudier les interactions d'agents hétérogènes. Les agents utilisés sont des robots spécifiquement conçus et développés pour le projet. Ces robots sont au nombre de trois et sont spécialisés pour des tâches particulières (transport, vision, manipulation). Les différents robots de Swarmanoid sont conçus dès le départ pour résoudre des problèmes différents, ils sont donc développés avec des morphologies et des capacités différentes.

Les robots sont de trois types, ce sont les *Foot-bot*, *Hand-bot* et *Eye-bot*. Le *Foot-bot* est spécialisé dans le *transport* d'autres robots et de certains objets spécialement préparés. Le *Hand-bot* est spécialisé dans la *prise et la manipulation d'objets*. Finalement, le *Eye-bot* est spécialisé dans les problèmes de repérage et d'évaluation du terrain. Ces trois robots sont décrits dans la suite.

### 2.1.1 Foot-bot

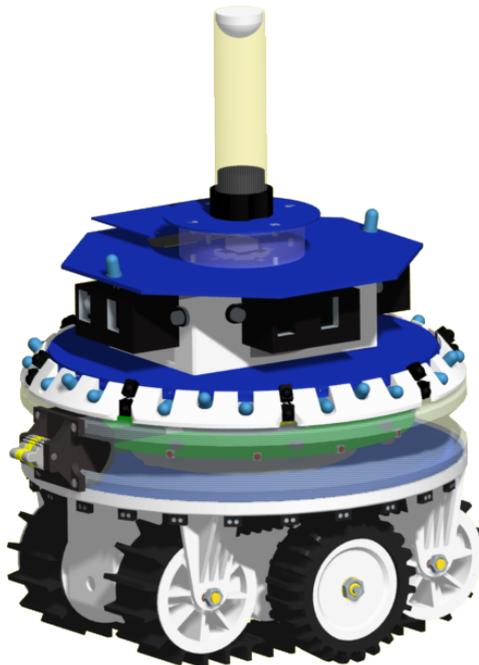


FIG. 2.1: Le *foot-bot*.

Le *Foot-bot* (figure 2.1) est un nouveau robot conçu sur des bases similaires au robot du projet qui a précédé *Swarmanoid*. Il est constitué de trois parties modulaires : motricité, interaction et capteurs. Ce robot a profité de l'expérience de la génération de robot précédente et propose des fonctionnalités particulièrement robustes.

La partie motrice utilise le principe des *treels* développés sur les *S-bot*. Ce système est constitué d'une chenille et d'une roue mûent le même moteur. La

base du *Foot-bot* comporte deux *treels* avec des moteurs indépendants et les batteries fournissant l'énergie au robot.

Ce système moteur donne au *Foot-bot* la capacité de se déplacer sur des terrains accidentés. Etant donné que les autres robots du projet n'ont pas de mobilité propre, et sont transportés par des *Foot-bot*, ils acquièrent les capacités du *Foot-bot* lors des déplacements. Ceci illustre les avantages apportés par des robots différenciés basé sur les principes de la robotique en essaim.

La partie intermédiaire du robot est constituée d'un anneau servant à l'assemblage inter-robot : le *docking-ring*, et d'un mécanisme d'attache permettant la connection aux autres robots : le *gripper*. Ces deux interfaces ont été importante pour le développement du *Cart-bot* car nous avons du fournir un système compatible dans le nouveau robot.

Finalement, la partie supérieure contient la carte mère, les cartes d'interfaces, et de nombreux capteurs du robot. Cette partie permet au *Foot-bot* de détecter son environnement et de communiquer avec d'autres robots.

## 2.1.2 Hand-bot

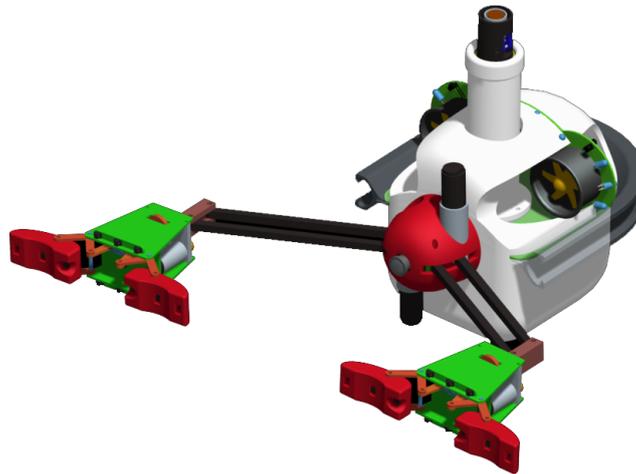


FIG. 2.2: Le *hand-bot*.

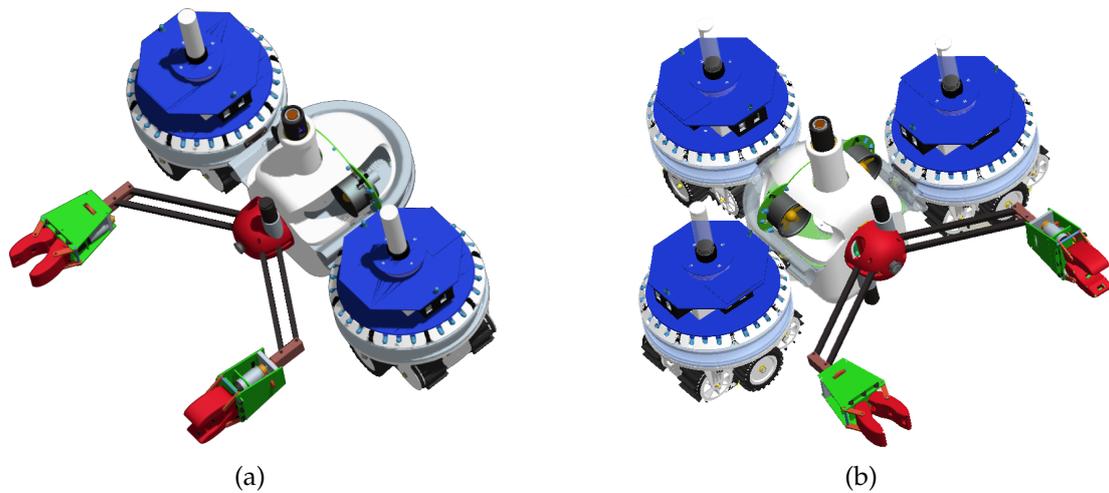


FIG. 2.3: *hand-bot* transporté par des *foot-bots*.

Le *Hand-bot* (figure 2.2) est une conception originale développée pour *Swarmanoid*. Il s'agit d'un robot complètement spécialisé à la manipulation des objets qui seront disposés dans l'environnement de *Swarmanoid*.

Le corps du robot contient l'électronique, les capteurs et les mécanismes nécessaires à la manipulation d'objets. On notera que le robot est aussi équipé d'un *docking ring* qui permet à des *Foot-bot* de s'y attacher. Le *Hand-bot* n'a en effet pas la possibilité de se mouvoir par lui-même, il repose sur les *Foot-bot* pour le transport.

La partie avant contient deux "bras" qui sont terminés par des manipulateurs, ceux-ci sont utilisés par le robot pour attraper et manipuler des objets de forme arbitraire. Ces bras constituent l'interface entre le *Hand-bot* et le *Cart-bot*, il a donc été essentiel de prendre en compte les spécifications des manipulateurs dans la conception du nouveau robot.

Le *Hand-bot* a la capacité de se déplacer verticalement afin de pouvoir atteindre des objets se trouvant en hauteur. Ceci est possible grâce à un filin déployé par le robot et le long duquel il peut se déplacer. Le filin est tiré par le *Hand-bot* en direction d'un plafond métallique, et s'accroche à l'aide d'un aimant.

### 2.1.3 Eye-bot



FIG. 2.4: Prototype de développement du *eye-bot*.

Le *Eye-bot* (figure 2.4) est un nouveau type de robot conçu spécifiquement pour *Swarmanoid*. Ce robot est spécialisé dans la vision et dans la reconnaissance du terrain. Le *Eye-bot* est un robot volant, ce qui lui donne la particularité de se déplacer dans les trois dimensions de l'arène.

Le *Eye-bot* doit pouvoir repérer facilement les autres robots présents dans l'environnement. Pour ce faire, il est doté d'une caméra haute résolution et les autres robots sont équipés d'une LED de grande puissance, ce qui les rends visible à de grande distance.

Le *Eye-bot* possède une tourelle mobile dotée d'une caméra et d'un pointeur laser (figure 2.5). Ce mécanisme lui permet de scruter l'environnement d'expérimentation et de pointer un élément pour les autres robots.

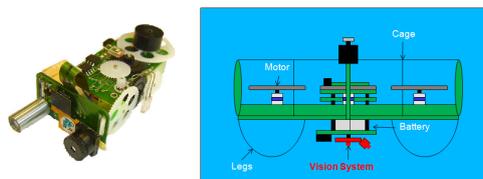


FIG. 2.5: Prototype de développement de la tourelle de vision du *eye-bot*.

## 2.1.4 Développement actuels

Les spécifications matérielles des trois robots de *Swarmanoid* ne sont pas complètement fixées, tous trois sont toujours en développement à l'Ecole Polytechnique Federale de Lausanne. Le *Foot-bot* est à un stade de développement avancé et une grande partie des spécifications nous ont été accessible pour le développement du *Cart-bot*. Les *Eye-bot* et *Hand-bot* sont à des states plus précoce de leur développement et les informations les concernant ne sont encore que partielle.

Le développement des robots de *Swarmanoid* a eu des implications sur le développement du *Cart-bot* car l'ensemble les spécifications ne sont pas toujours fixées. La conception du nouveau robot a pris cela en considération en conservant des degrés de liberté sur les parties qui dépendent des autres robots.

# Chapitre 3

## Cahier des charges, contraintes de réalisation

### Sommaire

---

3.1	Fonctionnalités à mettre en œuvre . . . . .	14
3.2	Compatibilité avec les autres robots du <i>swarmanoid</i> . . . . .	14
3.3	Utilisation aisée . . . . .	16
3.4	Contraintes de réalisation . . . . .	17

---

Dans ce chapitre, nous dressons un cahier des charges que doit respecter le *cart-bot*. L'intégration du *cart-bot* au scénario du projet Swarmanoid, exposée au chapitre précédent, met en évidence les fonctionnalités essentielles à implémenter.

En plus de ces fonctionnalités à mettre en œuvre, l'intégration du *cart-bot* au projet Swarmanoid entraîne certaines contraintes qui vont se répercuter sur sa conception et sa réalisation. Cette intégration concerne à la fois la compatibilité du *cart-bot* avec les robots en interaction avec lui, et l'utilisation aisée que doivent pouvoir en faire les chercheurs afin de mener efficacement leurs expériences.

Finalement, un dernier type de contraintes est dû aux moyens à disposition pour la réalisation.

### 3.1 Fonctionnalités à mettre en œuvre

Une première fonctionnalité caractérise l'intérêt même de créer le *cart-bot*. Il s'agit du regroupement de plusieurs *book-bots* en vue de leur transport. Cette fonctionnalité consiste donc à pouvoir charger et restituer des *book-bots*, ainsi qu'à pouvoir les stocker dans le *cart-bot*. Les *book-bots* en question sont au format livre de poche, c'est à dire 180 x 110 x 20 mm environ. Le *cart-bot* doit avoir des dimensions suffisantes pour pouvoir stocker un certain nombre de ces *book-bots*.

Afin de rendre le transport efficace et prétendre tirer avantage de l'utilisation du *cart-bot*, il est important de limiter ou d'éliminer les frottements au sol durant son transport par des *foot-bots*.

Enfin la mise en place d'une communication est nécessaire à la collaboration du *cart-bot* avec d'autres robots : avec un ensemble constitué d'un *hand-bot* tenu par des *foot-bots* (figure 2.3) pour l'acquisition et la restitution de *book-bots*, et avec des *foot-bots* pour le transport. Cette communication permet au *cart-bot* de signaler son état, de donner des indications de position ou de guidage, et de recevoir des informations des robots à proximité.

### 3.2 Compatibilité avec les autres robots du *swarmanoid*

Trois types d'interaction peuvent survenir entre le *cart-bot* et d'autres robots du Swarmanoid : le transport par des *foot-bots*, l'apport ou le retrait d'un *book-bot* par un *hand-bot*, et la communication avec les robots à proximité. Les deux premiers types d'interaction impliquent avant tout une compatibilité mécanique. Le dernier type nécessite une compatibilité dans les médiums et les protocoles de communication.

Pour un transport possible par les *foot-bots*, le *cart-bot* doit posséder un rail d'arrimage au profil adapté à leur système de fixation, c'est à dire leur pince inversée (voir figure 4.1a). De plus, comme on peut le voir sur la figure 2.1, la

pince des *foot-bots* n'est pas réglable en hauteur, le rail d'arrimage doit donc être à bonne hauteur lorsque les *foot-bots* tentent de s'y attacher. Le diamètre du corps du *cart-bot* sera probablement une fois et demi à deux fois plus grand que celui des *foot-bots*, qui est d'environ 20 cm. Afin de répartir au mieux la charge portée entre les *foot-bots*, le rail doit couvrir au maximum la circonférence du *cart-bot*. En effet, les *foot-bots* peuvent alors s'attacher en tout point du *cart-bot*, ce qui leur permet de se répartir le plus équitablement possible autour de lui.

Quoiqu'il en soit le *cart-bot* devra être le plus léger possible. On peut estimer le poids des *foot-bots* (encore en cours de prototypage) aux alentours de 1 kg, et leur pince d'arrimage peut reprendre de fortes charges. Le poids du *cart-bot* ne devrait pas dépasser 4 à 5 kg pour permettre un transport fiable en monopolisant le moins de *foot-bots* possible, sachant que trois *foot-bots* au minimum sont nécessaires pour garantir un transport stable (principe du trépied).

En intégrant un projet de robotique coopérative, le *cart-bot* se doit de pouvoir communiquer avec les autres robots du projet. Du moins ceux avec lesquels il aura des interactions directes. Nous l'avons vu au chapitre précédent, au sein d'un swarmanoid les communications se font principalement par couleurs et par signaux infrarouges (IR). Pour communiquer par les couleurs, le *cart-bot* doit lui aussi utiliser un anneau de LEDs RGB, dont la lumière est diffusée par une optique adéquate. Il lui faut également un moyen de décoder les signaux de couleurs des robots en interaction direct et donc à proximité immédiate. Concernant la communication par infrarouges, elle s'effectue sur les *foot-bots* et *hand-bots* à travers leurs capteurs de proximité et leur système de *range and bearing*. Le *cart-bot* doit donc également mettre en œuvre des modules émetteur et récepteur infrarouges s'il doit communiquer par ce médium. Ces couples émetteur/-récepteur infrarouges servent également de capteurs de proximité permettant au *cart-bot* de déterminer la position des autres robots dans son entourage immédiat.

Le *cart-bot* devra bien sûr éviter de créer des interférences dans les communications entre les autres robots. L'autonomie du *cart-bot* doit être suffisante pour la durée des expériences les plus longues. La gestion de sa batterie doit être telle que même en cas de faible niveau d'énergie dans la batterie le *cart-bot* soit en-

core capable de se faire transporter et redéposer à l'objectif, quitte à refuser des *book-bots* pour préserver l'énergie restante.

### 3.3 Utilisation aisée

Des expériences avec les robots peuvent être répétées un grand nombre de fois à la suite, en changeant simplement un ou deux paramètres. Le *cart-bot*, de même que les autres robots, doit présenter une grande fiabilité et une bonne reproductibilité des comportements. Cette fiabilité est notamment garantie par la qualité des pièces utilisées et de celle réalisées pour la conception. Mais elle est aussi garantie dans la durée par une maintenance et un entretien aisés. Pour cela les mécanismes doivent être faciles à démonter et à réparer. Ils doivent donc être au possible indépendants, de manière à pouvoir être isolés facilement. Les pièces quant à elles, doivent être choisies de manière à être faciles à remplacer, et donc à se procurer.

Pour assurer une rapide prise en main du *cart-bot* par les chercheurs qui mèneront les expériences, il doit utiliser au maximum les outils de développement déjà disponibles, et surtout déjà maîtrisés. Les méthodes de programmation du contrôleur du *cart-bot* doivent donc être similaires à celles des autres robots, ou à défaut, à certaines méthodes de projets précédents. Le choix de l'architecture électronique principale est donc idéalement une architecture avec laquelle le groupe de recherche est déjà familier.

Il se peut que de nouveaux besoins apparaissent au cours du temps. Le *cart-bot* a tout intérêt à être conçu de manière à favoriser le développement continu, en vue de nouvelles fonctionnalités. Ce pourrait-être par exemple la lecture d'étiquettes RFID placées sur les *book-bots* et contenant un identifiant. Il faut alors que le *cart-bot* présente une certaine modularité, permettant d'ajouter ou de modifier des extensions facilement.

### 3.4 Contraintes de réalisation

La conception du *cart-bot* doit tenir compte des moyens disponibles pour la réalisation. Un partenariat avec le service BEAMS permet d'accéder à des machines de productions simples, telles que tours et fraiseuses, à travers le savoir faire et l'expérience du technicien du service. Il ne s'agit cependant pas de machines à commande numérique (CNC), on ne peut donc pas parler de prototypage rapide comme dans le cas des autres robots, réalisés à l'EPFL (Ecole Polytechnique Fédérale de Lausanne). L'utilisation de techniques et d'outils conventionnels facilite cependant le remplacement de pièces défectueuses.

Les divers composants et matériaux constituant le *cart-bot* sont aussi choisis dans l'optique d'en faciliter la construction et la maintenance. La conception du *cart-bot* doit chercher à minimiser le nombre et la complexité des pièces à réaliser. Il faut cependant éviter d'utiliser des composants difficiles à se procurer en petite quantité, et favoriser les fournisseurs classiques tels que Farnell ou RS-Components. Et s'assurer ainsi d'un remplacement possible de ces composants dans un délai et pour un budget raisonnables.

# Chapitre 4

## Solutions envisagées

### Sommaire

---

<b>4.1 Être transportable . . . . .</b>	<b>19</b>
4.1.1 Système d'arrimage . . . . .	19
4.1.2 Élimination des frottements . . . . .	20
4.1.3 Sécurisation du contenu . . . . .	21
<b>4.2 Regroupement de plusieurs <i>book-bots</i> . . . . .</b>	<b>22</b>
4.2.1 Stockage . . . . .	22
4.2.2 Chargement et déchargement . . . . .	25
<b>4.3 Communiquer . . . . .</b>	<b>27</b>
<b>4.4 Utiliser des outils de développement déjà maîtrisés . . . . .</b>	<b>28</b>
<b>4.5 Être réutilisable rapidement . . . . .</b>	<b>28</b>

---

Le cahier des charges dressé au chapitre précédent fait notamment ressortir les besoins auxquels doit répondre le *cart-bot* pour être fonctionnel :

- permettre le regroupement de plusieurs *book-bots*, c'est à dire pouvoir stocker et restituer plusieurs *book-bots*
- être facilement transportable par les *foot-bots*
- disposer de capacités de communication et de guidage pour les autres robots
- Utiliser des outils de développement déjà maîtrisés

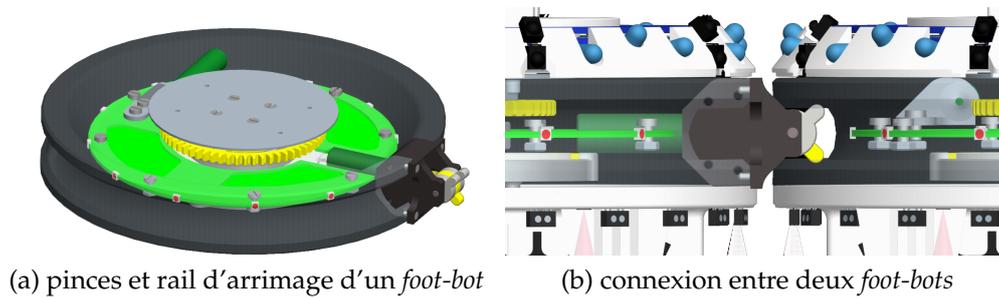


FIG. 4.1: Système d'arrimage sur les *foot-bots*.

A cela s'ajoute un dernier besoin, celui d'être réutilisable rapidement pour une autre expérience.

## 4.1 Être transportable

### 4.1.1 Système d'arrimage

Pour rappel, le *cart-bot* ne dispose pas de moyen de locomotion propre, mais est transporté par des *foot-bots*. Les *foot-bots* sont conçus pour pouvoir s'attacher entre eux et à d'autres objets prévus à cet effet. Ces objets, ainsi que les *foot-bots* eux-mêmes, possèdent un rail d'arrimage circulaire ou *docking ring*. Pour se connecter à ces *docking ring*, les *foot-bots* sont équipés d'une pince inversée (figure 4.1a) qui une fois insérée dans la rainure du rail d'arrimage s'écarte et assure ainsi la liaison (figure 4.1b). Cette liaison peut-être souple ou rigide selon le degré d'écartement de la pince.

Pour permettre aux *foot-bots* de s'arrimer à lui, le *cart-bot* doit donc posséder un rail d'arrimage à la bonne hauteur, la pince des *foot-bots* étant fixe. Cependant comme on peut le voir sur les figures 2.2 et 2.3, le pivot des bras des *hand-bots* est également situé à la hauteur des rails d'arrimage, et donc de cette pince. Le rail d'arrimage du *cart-bot* risque donc de se retrouver dans le chemin du *book-bot* apporté par un *hand-bot*.

Nous avons donc envisagé 3 solutions :

- le rail n'est pas un anneau continu mais présente une ouverture pour laisser passer le *book-bot*
- le rail ou une partie de la structure du robot supportant le rail peut s'ouvrir
- le rail ou une partie de la structure du robot supportant le rail peut être élevé ou abaissé, dégagant ainsi le passage

La première solution est la plus simple à mettre en œuvre mécaniquement. Mais elle présente l'inconvénient, de part la large ouverture nécessaire au passage aisé du *book-bot*, de ne pas offrir la totalité de la circonférence du *cart-bot* comme prise aux *foot-bots*. Les deux autres solutions impliquent la réalisation d'un mécanisme supplémentaire. L'abaissement du rail a cependant l'avantage sur un mécanisme d'ouverture d'une partie de la structure de ne pas modifier la surface occupée au sol par le *cart-bot*. Mais surtout, elle peut aussi être utilisée dans l'autre sens, c'est à dire faire monter la partie mobile par rapport au rail lorsque celui-ci est soutenu par les *foot-bots*. Elle répond alors également au problème exposé dans la section suivante.

#### 4.1.2 Élimination des frottements

Une fois les *foot-bots* physiquement connectés au *cart-bot*, celui-ci est toujours en contact avec le sol. Afin d'assurer un transport de qualité, il est préférable de limiter les frottements au sol, d'autant plus que le *cart-bot* présente une large surface au sol par rapport aux *foot-bots* et que son poids n'est pas négligeable. Dès lors on peut envisager soit de limiter ces frottements par l'adjonction de billes encastrées, soit d'éliminer tout contact en soulevant la structure.

La technique utilisée sur le *hand-bot* pour qu'il ne touche plus le sol lorsqu'il est transporté consiste à placer le rail d'arrimage légèrement plus bas que la pince des *foot-bots*. Ainsi lors de l'introduction de la pince dans le rail, celui-ci est progressivement soulevé. Cette technique ne permet qu'une très faible garde au sol et n'est pas adaptée au *cart-bot* dont le poids et la surface sont supérieurs à ceux du *hand-bot*.

L'utilisation de billes nécessite un système de blocage pour immobiliser le

*cart-bot* lorsqu'il n'est pas tenu par des *foot-bots*. De plus, elle entraîne également une mauvaise garde au sol.

La solution consistant à retirer le contact au sol paraît donc la plus prometteuse. Elle peut-être réalisée très simplement à l'aide de pieds rétractables. Chacun des pieds nécessite un mécanisme, mais il peut s'agir d'un simple servomoteur qui fait basculer le pied en position haute ou basse. Une autre manière de mettre en oeuvre cette solution est de soulever la partie du *cart-bot* en contact avec le sol par rapport au rail lorsque celui-ci est soutenu par les *foot-bots*. Cette mise en oeuvre requiert un mouvement relatif d'une partie de la structure du *cart-bot*, incluant les contacts au sol, par rapport au rail d'arrimage. Comme on l'a vu, le même mécanisme qui sert à dégager le rail du chemin du *book-bot* en l'abaissant peut alors être utilisé. Cette technique s'avère donc très intéressante pour minimiser le nombre de mécanismes. Nous voyons section 4.2.2 que son mouvement de translation verticale peut également être utilisé pour atteindre tous les *book-bots* stockés, ce qui fait d'elle la solution de premier choix.

### 4.1.3 Sécurisation du contenu

Pour éviter tout problème en cas de collision, ou tout autre imprévu, les *book-bots* doivent être sécurisés dans leur conteneur. Dans le cas contraire, ils risqueraient de tomber ou de bloquer des mécanismes, empêchant ainsi le bon déroulement de l'expérience en cours.

Les options pour sécuriser les *book-bots* diffèrent selon le type de stockage choisi. Si le stockage se fait dans des slots verticaux (voir les solutions possibles de stockage, section 4.2.1), alors le maintien des *book-bots* semble assuré. Du moins aussi longtemps que le *cart-bot* n'est pas renversé ou soumis à un choc important pouvant faire bondir des *book-bots* hors de leur slot. Si par contre, l'accès au stockage se fait horizontalement, un système de blocage doit être installé pour empêcher les *book-bots* de ressortir.

Deux systèmes ont été étudiés. Le premier, est un clapet basculant pour fermer l'accès aux slots horizontaux lorsqu'ils ne sont pas occupés à recevoir un

*book-bot*. Ce système s'est avéré complexe et fastidieux à réaliser (un mécanisme par slot), mais également fragile. Il a donc été abandonné. Le second est un film ou une cloison, mobile, possédant une ouverture déplacée à l'endroit où le *book-bot* sera introduit. la mise en œuvre d'un tel film est décrite dans la section présentant la réalisation mécanique du *cart-bot*.

## 4.2 Regroupement de plusieurs *book-bots*

Le type de systèmes de chargement et de déchargement de *book-bots* dépendront, s'il sont nécessaires, du système de stockage choisi. Ce dernier est donc analysé en premier.

### 4.2.1 Stockage

Pour un simple stockage, un simple empilement des *book-bots* dans une sorte de conteneur semble la solution la plus simple. Elle risque cependant de poser des problèmes pour une récupération individuelle des *book-bots*. A cette fin, il est préférable d'avoir une séparation physique entre les *book-bots* ce qui revient à les placer dans des slots. Quatre configurations sont alors envisageables et analysées dans la suite.

#### Slots verticaux

Dans la première configuration, les *book-bots* sont logés dans des slots verticaux. Ils sont donc stockés verticalement les uns à côté des autres. Pour l'insertion des *book-bots*, deux choix se présentent : une insertion par le haut avec des slots ouverts sur le dessus, et une insertion frontale avec des slots ouverts sur un des côtés, en plus de éventuellement sur le dessus. Cette configuration peut être constitué uniquement de parois droites.

Elle a le mérite d'être mécaniquement simple, mais présente deux inconvénients majeurs. Le premier est que le nombre de *book-bots* qu'il est possible de

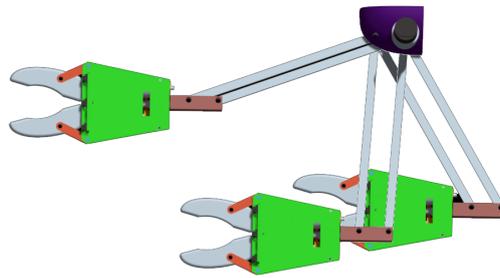


FIG. 4.2: Pince du *hand-bot* montée sur parallélogramme.

stocker influence directement la surface occupée par le *cart-bot* au sol. Plus ce nombre sera important et plus le *cart-bot* sera large et encombrant.

Le deuxième gros inconvénient est qu'à moins de systèmes de chargement et de déchargement, le *hand-bot* doit venir placer lui-même précisément le *book-bot* dans le slot adéquat. Or comme le montre la figure 4.2, ses pinces manipulatrices étant montées sur des bras constitués d'un parallélogramme, elles ne peuvent être orientées autrement que dans l'axe longitudinal du *hand-bot*. Celui-ci doit donc être parfaitement aligné dans l'axe du slot par les *foot-bots* qui le transportent. La gestion du dépôt du *book-bot* par le *hand-bot* est donc complexe, ce qui nécessite un travail supplémentaire important au niveau du contrôleur du *hand-bot*. Ceci n'est pas désirable.

L'option de l'insertion par le haut empire encore les choses avec la nécessité de soulever le *book-bot* puis de l'abaisser précisément dans le slot. Elle est donc à proscrire.

### Slots horizontaux

Dans la deuxième configuration, les *book-bots* sont logés dans des slots horizontaux formant une étagère. Ils sont donc stockés horizontalement les uns au dessus des autres. L'insertion est nécessairement frontale.

Cette solution est tout aussi simple mécaniquement que la configuration en slots verticaux. Elle ne partage cependant pas ses inconvénients. Dans ce cas ci, les slots s'accumulent verticalement et leur nombre n'a donc pas d'influence

sur la circonférence du *cart-bot*. Quant à l'insertion direct de *book-bots* dans leur slot, elle offre une plus grande marge de manœuvre aux *hand-bots* avec des slots légèrement plus large que les *book-bots* à y introduire. Il suffit alors en effet que le *hand-bot* relâche son étreinte sur le *book-bot* une fois celui ci partiellement introduit dans le slot, et finisse de le pousser. Le *book-bot* se retrouve guidé au fond du slot par les parois. Par contre, la solution d'insérer directement les *book-bots* dans les slots par les *hand-bots* limite le nombre de slots utilisable, de par la hauteur que les pinces des *hand-bots* peuvent atteindre.

Ces deux premières configurations imposent, à moins d'utiliser des systèmes de chargement et de déchargement supplémentaires, de venir déposer les *book-bots* directement dans les slots vides. Les *hand-bots* doivent déterminer par eux-mêmes, ou sur indication du *cart-bot*, quelles slots sont encore vides, et se positionner correctement pour les atteindre. Les deux configurations suivantes permettent aux *hand-bots* de venir insérer ou récupérer les *book-bots* en un seul et même endroit pour atteindre tous les slots, et ce sans l'aide de mécanisme supplémentaires. Elles réalisent cela à l'aide de slots mobiles.

## **Barillet**

La troisième configuration consiste en un barillet. Les slots sont disposés de manière radiale autour d'un axe de rotation. Une utilisation verticale d'un tel barillet n'améliore en rien la marge de manœuvre des *hand-bots* dans l'insertion des *book-bots*, et n'est donc pas intéressante.

L'utilisation de ce barillet, même horizontale, a le désavantage de présenter un rapport de volume utile au stockage sur volume occupé faible. Cela est dû à la présence de "zones mortes" entre les slots. De plus il faut entourer ce barillet d'un guide pour s'assurer que les livres restent bien dans les slots durant la rotation.

## Chenille de convoyage

La dernière configuration est une variante du barillet horizontal. Il s'agit d'une chenille de convoyage/stockage. Il y a toujours une rotation, mais qui entraîne un mouvement de translation des slots. Le long de cette translation les slots sont parallèles et le problème des "zones mortes" est éliminé.

Ce système présente tous les avantages des slots horizontaux fixes, sans leur limitation en nombre puisque les slots les plus haut peuvent être descendus pour être chargés d'un *book-bot* avant d'être remonté.

Le stockage se caractérise comme une pile FILO (First In Last Out, ou premier entré dernier sorti) et ne permet donc pas de choisir l'ordre de restitution des *book-bots*. Cette fonctionnalité n'a cependant pas été requise dans le cahier des charges.

Un des principaux points faibles de cette configuration est la difficulté de réaliser une telle chenille, qui n'a pu être trouvée toute faite. Un autre est le fait que le système d'élévation ne peut être utilisé pour dégager le rail d'arrimage du chemin d'insertion des *book-bots*, ou pour éliminer les frottements au sol durant le transport. C'est pourtant le cas avec le système que nous avons mis au point et qui sera présenté au chapitre 5.

### 4.2.2 Chargement et déchargement

On vient de le voir, pour certains types de stockage, le chargement et le déchargement des *book-bots* sont assurés directement par le *hand-bot*. Pour les configurations avec slots fixes, Il faut que l'accès à la totalité des slots soit libre. Si l'accès aux slots se fait par le dessus ce n'est pas un problème, mais dans le cas contraire, cela implique une large interruption dans le rail d'arrimage, ce qui n'est pas souhaitable.

Dans certaines configurations, le *hand-bot* ne peut pas déposer directement les *book-bots* dans les slots. Il faut alors prévoir un système capable d'amener le *book-bot* au bon slot, de l'y faire entrer, et de l'en faire ressortir. on peut voir

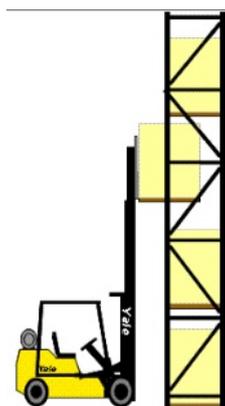


FIG. 4.3: Principe du système de chargement/déchargement : le chariot élévateur.

se système comme l'équivalent d'un chariot élévateur chargeant une étagère (figure 4.3).

Le rôle de ce mécanisme de chargement/déchargement est de tenir le *book-bot* une fois que le *hand-bot* l'a lâché et l'amener au lieu de stockage. Il permet au *hand-bot* de toujours déposer le *book-bot* au même endroit, et enlève ainsi la limitation sur le nombre de slots horizontaux fixes. Il permet en effet d'atteindre ceux qui seraient trop hauts pour le *hand-bot*.

Plusieurs manières de concevoir ce mécanisme ont été étudiées. Une première consiste en un tapis roulant et implique l'utilisation d'un autre système pour faire sortir le *book-bot* du slot. Probablement aussi pour l'y faire entrer complètement. La seconde manière est la mise en place de deux bras latéraux qui viennent pousser le *book-bot* pour le faire entrer et viennent le pincer pour le tirer hors du slot. Ce mécanisme semble assez dure à réaliser.

Une dernière option envisagée consiste en une hybridation des solutions précédentes afin de tirer parti de la simplicité mécanique du tapis roulant et de l'efficacité des bras latéraux ne nécessitant pas de mécanisme supplémentaire. C'est la solution qui a été retenue grâce à une innovation présentée dans la section 5.4.



FIG. 4.4: Capteur de couleur de chez TAOS.

### 4.3 Communiquer

De part la définition même de la robotique en essaim, le *cart-bot* doit pouvoir interagir avec les autres robots. On peut cependant supposer que le *cart-bot* n'aura pas à interagir avec des *eye-bots*, car étant transporté par des *foot-bots*, il n'a aucun besoin d'être lui même guidé.

Les *foot-bots* et les *hand-bots* peuvent communiquer par couleurs et infra-rouges (voir description des robots au ??). Pour la lecture des messages de couleur, ils utilisent une ou plusieurs caméras. Ce type de capteur nécessite une importante puissance de calcul et la mise au point d'un traitement performant de l'image.

Pour alléger les calculs, une possibilité à étudier est d'utiliser de simples capteurs de couleurs, tels que les TCS230 de chez TAOS (Texas Advanced Optoelectronic Solutions)(figure 4.4). La portée de reconnaissance est très limitée, mais la communication qui nous intéresse est celle avec les *foot-bots* déjà attachés ou en cours de l'être, et avec le *hand-bot* en train de déposer le *book-bot*, donc tout proche.

On peut envisager d'utiliser le rail d'arrimage transparent dépoli comme optique. C'est déjà le cas pour diffuser la lumière colorées provenant des LEDs RGB. Une telle utilisation devrait raccourcir la distance de lecture, mais élargir le champ de vision et donc permettre l'utilisation de moins de capteurs.

Une communication infrarouge peut se faire à l'aide de capteurs infra-rouges de proximité (couples émetteur/récepteur) servant également à repérer la proximité d'autres robots ou d'obstacles. Cette solution à déjà été mise en œuvre avec succès sur des *e-pucks*.

## 4.4 Utiliser des outils de développement déjà maîtrisés

Réutiliser une architecture déjà développée pour un autre projet présente l'avantage d'avoir un système ayant déjà fait ses preuves, et donc à priori fiable. Cela permet également d'importants gains de temps et d'argent en évitant un certain nombre d'impasses et de mauvaises surprises lors de la conception.

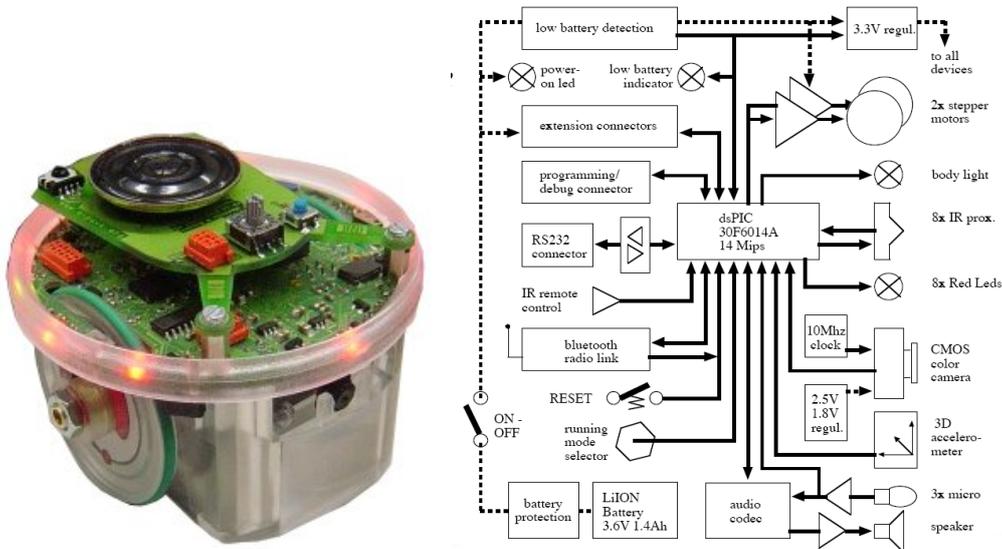
Les architectures électroniques des projet Swarmanoid et Swarmbots sont respectivement à base de processeur ARM11 et ARM9. De telles architectures permettent de travailler avec un linux embarqué, mais la réalisation de l'électronique est délicate. Ces processeurs 32 bits travaillent à haute fréquence (533 MHz et 200 MHz) et présentent de nombreuses entrées/sorties, et donc une grande densité de pattes, ce qui le routage du circuit imprimé critique.

Le *cart-bot* est un robot plus simple que les autres, il ne requiert pas une telle puissance de calcul. L'utiliser tout de même complexifierait la conception de l'électronique, au détriment de la fiabilité justement. Une autre architecture bien maîtrisée est celle du projet open hardware *e-puck* développé par l'EPFL (Ecole Polytechnique Fédérale de Lausanne), partenaire dans le projet *Swarmanoid*. Il s'agit d'une architecture à base d'un microcontrôleur 16 bits, le *dsPIC30F* de chez Microchip, bien plus simple à mettre en œuvre.

## 4.5 Être réutilisable rapidement

Que le *cart-bot* soit réutilisable rapidement signifie qu'il doit pouvoir être mis rapidement dans un état initial pour une nouvelle expérience. Or une nouvelle expérience peut impliquer un nouveau type de comportement, et donc un nouveau contrôleur. Le processeur principal du *cart-bot* devra alors être reprogrammé. L'idéal en cas de nombreux robots à reprogrammer souvent est d'utiliser une connexion sans-fil, par exemple de type Bluetooth.

Notre cas est moins critique : le nombre de *cart-bots* utilisé, de par leur rôle



(a) robot *e-puck* développé par l'EPFL (b) architecture électronique du *e-puck*

FIG. 4.5: *e-puck*, base d'inspiration pour l'électronique du *cart-bot*.

de regroupement, devrait-être restreint voir unitaire pour la plupart des expériences. Il suffit donc d'assurer un accès aisé au connecteur du programmeur pour ce processeur.

Un autre point important pour une réutilisation rapide du robot est le niveau de charge des batteries. Au début d'une expérience, elles doivent être pleines ou suffisamment chargées pour assurer le bon déroulement de l'expérience avec une certaine marge. L'expérience pourrait en effet durer plus longtemps ou les batteries se décharger plus vite que prévu. La recharge de batteries étant généralement assez longue, il est préférable de pouvoir changer facilement les batteries usées par des pleines. Le *cart-bot* peut alors être immédiatement réutilisé pendant que les batteries vidées se rechargent.

# Chapitre 5

## Stratégie retenue et conception du *cart-bot*

### Sommaire

---

5.1	Stratégie retenue . . . . .	30
5.2	Stockage . . . . .	32
5.3	Élévation . . . . .	33
5.4	Chargement/déchargement . . . . .	34

---

### 5.1 Stratégie retenue

La stratégie retenue est d'utiliser le moins de mécanismes possibles, et de faire en sorte que ceux utilisés soient les plus simples possible. Nous avons ainsi développé un système permettant à la fois d'assurer l'élévation du système de chargement/déchargement des *book-bots*, de positionner le rail d'arrimage à la bonne hauteur au moment du transport sans qu'il soit dans le chemin d'insertion des *book-bots*, et enfin d'éliminer tout contact avec le sol durant le transport.

Ce système, comme on peut le voir figure 5.1a, est constitué d'une étagère de slots horizontaux reposant au sol, entourée d'une base circulaire mobile com-

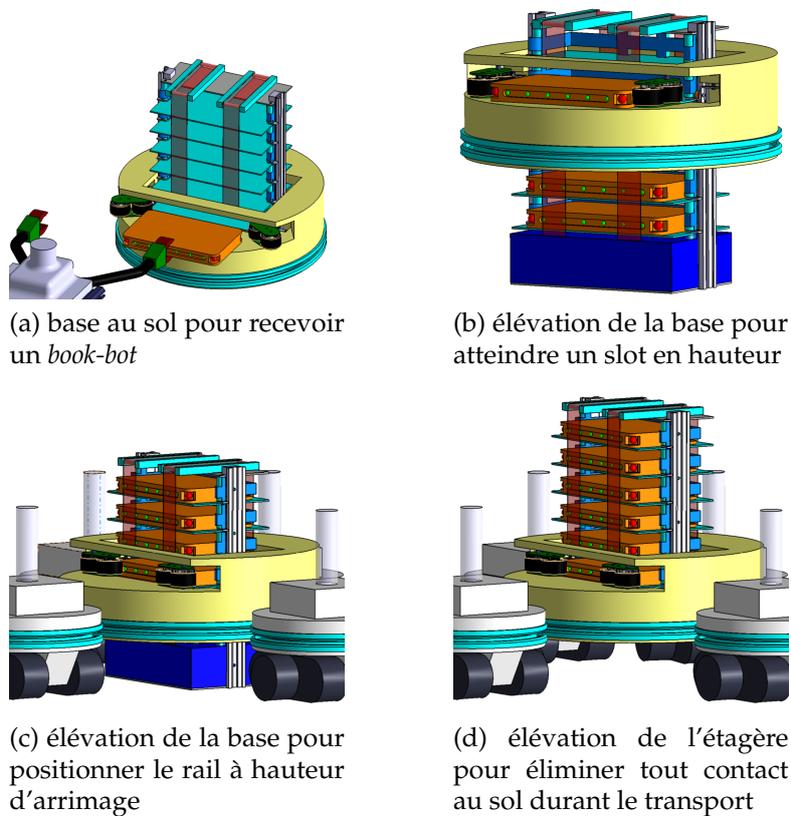


FIG. 5.1: le système d'élévation dans le *cart-bot*.

prenant le rail d'arrimage et le mécanisme de chargement/déchargement. L'étagère et la base peuvent coulisser verticalement l'une par rapport à l'autre, et le rail d'arrimage est fixé de manière à être plus bas que la pince des *foot-bots* lorsque la base repose au sol. Il suffit de faire monter la base le long de l'étagère pour mettre le rail à la bonne hauteur (figure 5.1c) ou pour atteindre les slots les plus hauts (figure 5.1b). Pour éliminer le contact au sol de l'étagère, il suffit de la faire remonter à travers la base lorsque celle-ci est tenue par des *foot-bots* (figure 5.1d).

Le système de chargement/déchargement que nous avons mis au point permet aussi de limiter le nombre de mécanismes. Il s'agit de bras latéraux portant une roue à chaque extrémité. Le fonctionnement en est illustré à la figure 5.3. Une rotation des bras (figures 5.2a et 5.3a) et vient caler les roues des extrémités

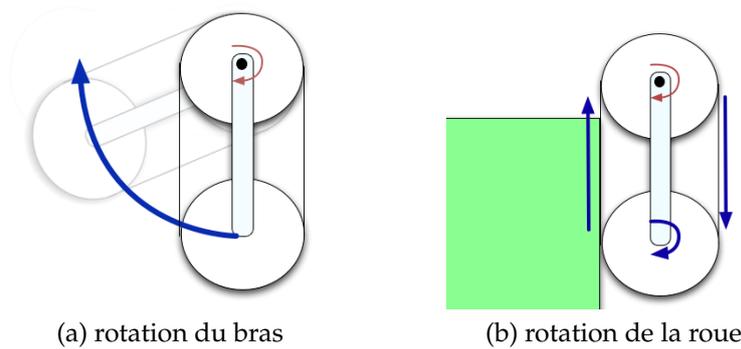


FIG. 5.2: actions d'un bras du mécanisme de chargement/déchargement.

libres contre le *book-bot* à charger, assurant ainsi une bonne accroche. Les roues peuvent alors faire avancer le *book-bot* en roulant dessus, à la manière d'un tapis roulant (figures 5.2b et 5.3b). Lorsque le *book-bot* dépasse les roues, les bras se remettent à tourner dans la même direction pour finir de le pousser complètement dans le slot. Il suffit de faire tourner les bras dans le sens inverse pour aller chercher un *book-bot* dans un slot. Le système est d'autant plus innovant qu'il n'utilise qu'un simple moteur par côté pour réaliser cela.

Le *cart-bot* est constitué de modules les plus indépendants possible afin de faciliter son assemblage et sa maintenance. Cette modularité permet également de rendre le *cart-bot* plus souple à de futur évolutions possible. L'ensemble des modules ont été conçus de manière à être les plus simples possible à réaliser. Excepté la forme circulaire de grand diamètre de la base, ils ne nécessitent pas d'usinage compliqué. Les composants ont été choisis de manière à devoir être le moins retravaillés possible. Les sections qui suivent décrivent la conception de chaque module.

## 5.2 Stockage

Le module de stockage correspond à l'étagère associé au film qui retient les *book-bots* dans leur slot. L'étagère est faite d'aluminium et de Plexiglas afin d'être la plus légère possible tout en offrant une certaine résistance. Elle est constituée

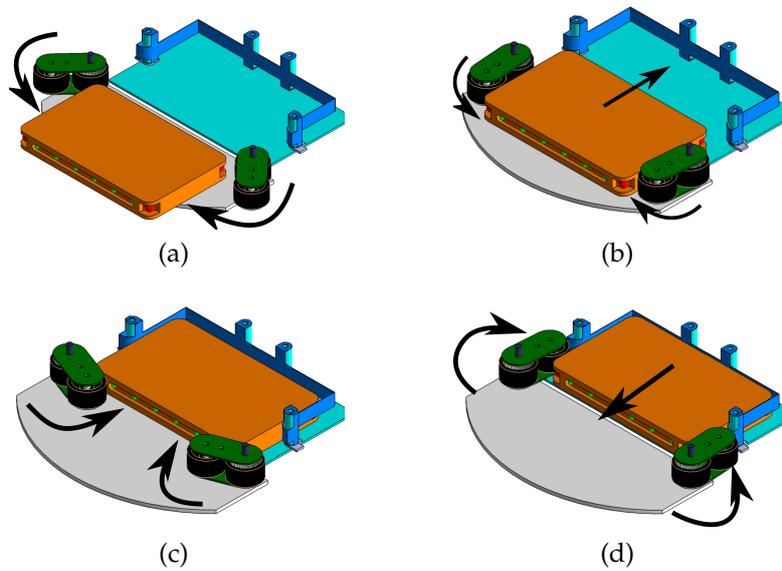


FIG. 5.3: principe de fonctionnement du mécanisme de chargement/déchargement de *book-bots*.

uniquement de plaques percées, de barres et de tubes, ainsi que de film souple pour les parois des slots. Elle supporte néanmoins deux rails de guidage linéaire de la marque IGUS. Ces rails garantissent une bonne précision dans la translation rectiligne et une quasi absence de jeu. Ceci est important pour avoir un alignement parfait de la surface sur laquelle repose le *book-bot* avec la paroi inférieure du slot dans lequel il doit glisser.

Le film de sécurité empêchant les *book-bots* de ressortir des slots est simplement un film souple présentant une ouverture et formant une boucle autour de l'étagère. L'ouverture de ce film est fixé en face du système de chargement/déchargement. Quand celui-ci se déplace le long de l'étagère, le film coulisse simplement autour d'elle.

### 5.3 Élévation

Le module d'élévation est constitué d'un bloc moteur fixé à la base du *cart-bot* (figure 5.5), et d'une courroie crantée coupée dont les extrémités sont fixés en

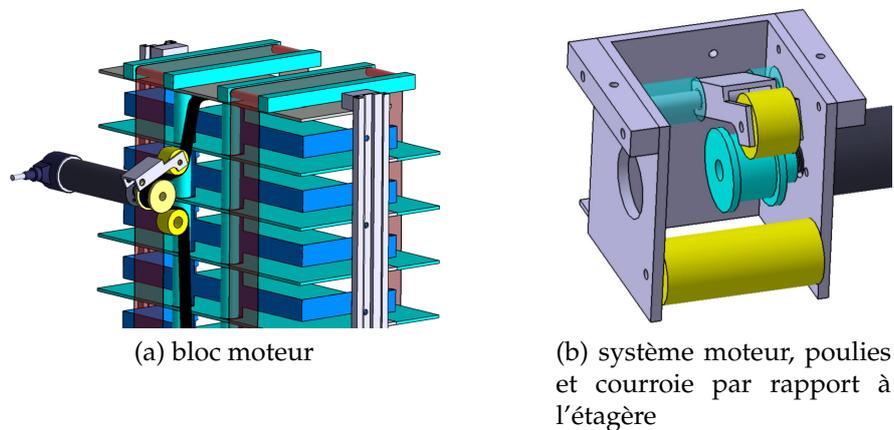


FIG. 5.4: composants du système d'élévation.

haut et en bas de l'étagère, à l'arrière de celle-ci. Le bloc moteur est conçu pour pouvoir facilement enlever ou remettre la courroie autour de la poulie crantée, sans devoir rien faire d'autre que de desserrer le tendeur. La figure 5.4a illustre le moteur les poulies et la courroie par rapport à l'étagère, mais sans représenter la base pour plus de visibilité. La régulation d'un tel système s'apparente très fort à celle d'un ascenseur et est donc bien documentée. Sa réalisation n'en sera que plus aisée.

Pour empêcher la base de redescendre par gravité après avoir été élevée, il a fallu mettre au point un système de freinage. Nous avons choisi de le faire par rapport à l'étagère puisque c'est l'alignement de celle-ci qui est critique (figure 5.6a). D'ailleurs les positions bloquées correspondent à cette alignement. Ce système de freinage est réalisé de manière très simple à l'aide de servomoteurs et ne nécessitent que peu de pièces supplémentaires. Ils permettent un réglage fin de la position bloquée à l'aide d'une vis.

## 5.4 Chargement/déchargement

Le système de chargement/déchargement complet est constitué des bras latéraux ainsi que de la base, servant à soutenir le livre et à la monter jusqu'au slot

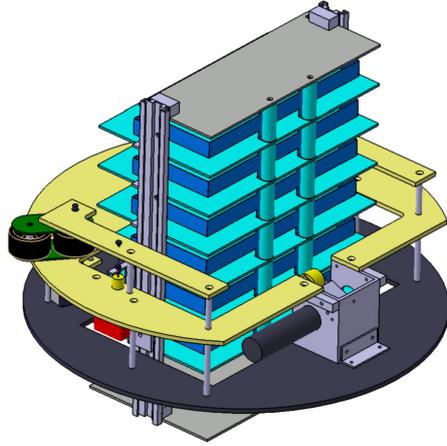


FIG. 5.5: le bloc moteur fixé à la base du *cart-bot*.

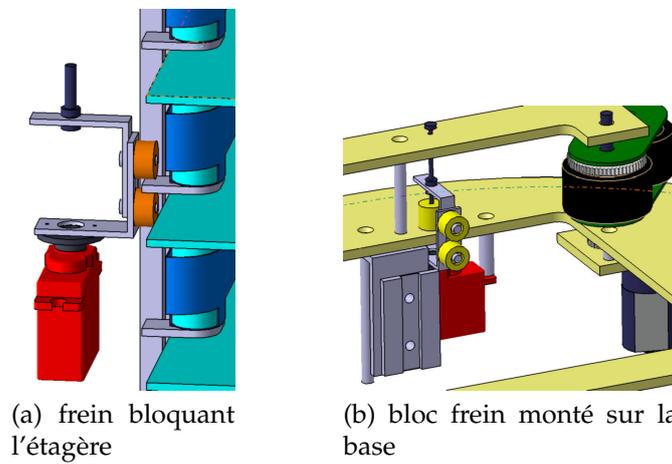
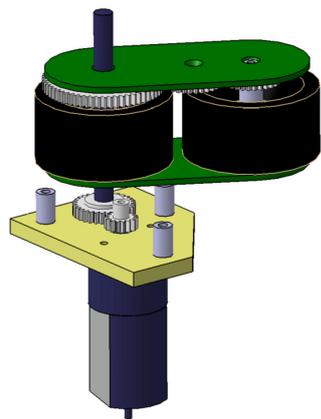
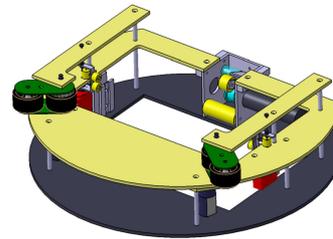


FIG. 5.6: système de freinage pour bloquer la base à la hauteur souhaitée.



(a) bloc bras de chargement avec son moteur



(b) bras de chargement montés sur la base, constituant ainsi le système de chargement/déchargement complet

FIG. 5.7: système de chargement/déchargement.

adéquat (figure 5.7). Ils ont cependant été conçus comme deux modules indépendants pour permettre un entretien et un remplacement aisé des bras, pièces critiques du *cart-bot*. Les bras peuvent en effet se retirer très facilement et très rapidement.

L'entraînement à la fois de la rotation du bras et de celle des roues est assuré par un jeu d'engrenages de démultiplication par deux. La rotation des roues est deux fois plus rapide, mais offre deux fois plus de résistance. C'est donc le bras qui sera entraîné tant que rien ne vient le bloquer. La régulation de ce système est très simple et peut s'apparenter à celle d'un servo-moteur avec en information angulaire la position du bras, ce qui ne nécessite qu'un encodeur ou même simplement un potentiomètre sur l'axe de rotation du bras.

# Chapitre 6

## Conclusion

Nous projet participe à l'étude du transport collaboratif de multiples objets par un essaim de robots collaboratifs. Dans l'approche adoptée, plutôt que de transporter de petits objets un à un, ils sont rassemblés sur une palette qui est ensuite transportée par un groupe de robots. Avec cette approche, les performances de transport croissent de manière super-linéaire, avec une limitation supérieure due aux problèmes de coordination d'un très grand groupe de robots. Nous avons étudié le problème du transport d'objets multiples dans le cadre du projet *Swarmanoid*. Nous avons en particulier décrit les fonctionnalité et la conception d'un robot appelé *cart-bot*, qui est capable de stocker des objets, les *book-bots*, et de faciliter son propre transport par d'autres robots, les *foot-bots* qui sont spécialisés dans cet tâche.

Le *cart-bot* améliore la robustesse apportée par le matériel et le contrôle distribué inhérents à la robotique en essaim. Il sécurise les *book-bots* stockés pour éviter de les perdre durant le transport, même en cas de collision ou de terrain difficile. Sa large base permet aux *foot-bots* de s'y accrocher aisément tout autour, ce qui résulte en un assemblage très stable. De plus l'assemblage ainsi créé est peu sensible au type de terrain, grâce à une bonne garde au sol obtenue par l'élévation de l'étagère une fois accroché aux *foot-bots*.

Pour assurer ce transport robuste, le *cart-bot* est conçu de manière simple et fiable. Durant le transport il peut-être vu comme une unique et solide en-

tité. Il n'a en effet pas de parties fragiles extérieures. L'utilisation des mêmes mécanismes pour plusieurs fonctionnalités réduit la complexité du *cart-bot* et minimise son poids. Ainsi des bras latéraux rotatifs sont utilisés pour le chargement et le déchargement des *book-bots*, et un système d'élévation est utilisé pour atteindre différents slots de l'étagère de stockage, pour régler la hauteur de l'anneau d'arrimage, et enfin pour éliminer les frottements durant le transport. Enfin le *cart-bot* est conçu principalement avec des composants courants, ce qui le rend facile à maintenir.

# Bibliographie

- [1] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence : from natural to artificial systems*. Oxford University Press, Inc., New York, NY, USA, 1999.
- [2] A. Campo, S. Nouyan, M. Birattari, R. Groß, and M. Dorigo. Negotiation of goal direction for cooperative transport. In *Proc. of the 5th Int. Workshop on Ant Colony Optimization and Swarm Intelligence, ANTS 2006*, volume 4150 of *Lecture Notes in Computer Science*, pages 191–202. Springer Verlag, Berlin, Germany, 2006.
- [3] Anders Christensen. Efficient neuro-evolution of hole-avoidance and phototaxis for a swarm-bot. Technical Report TR/IRIDIA/2005-014, IRIDIA, Université Libre de Bruxelles, October 2005.
- [4] M. Dorigo, E. Tuci, R. Groß, V. Trianni, T.H. Labella, S. Nouyan, C. Ampatzis, J.-L. Deneubourg, G. Baldassarre, S. Nolfi, F. Mondada, D. Floreano, and L.M. Gambardella. The SWARM-BOTS project. In E. Şahin and W. Spears, editors, *Proceedings of the First International Workshop on Swarm Robotics at SAB 2004*, volume 3342 of *Lecture Notes in Computer Science*, pages 31–44. Springer Verlag, Berlin, Germany, 2004.
- [5] Marco Dorigo, Eric Bonabeau, and Guy Theraulaz. Ant algorithms and stigmergy. *Future Gener. Comput. Syst.*, 16(9) :851–871, 2000.
- [6] R. Groß. *Self-assembling robots*. PhD thesis, Université Libre de Bruxelles (ULB), Brussels, Belgium, October 2007.

- [7] R. Gross, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in mobile robotics. 2006. In press.
- [8] R. Groß and M. Dorigo. Cooperative transport of objects of different shapes and sizes. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, editors, *Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004*, volume 3172 of *Lecture Notes in Computer Science*, pages 107–118. Springer Verlag, Berlin, Germany, 2004.
- [9] R. Groß and M. Dorigo. Group transport of an object to a target that only some group members may sense. In *Parallel Problem Solving from Nature – 8th International Conference (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science*, pages 852–861. Springer Verlag, Berlin, Germany, 2004.
- [10] Owen Holland and Chris Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5(2) :173–202, 1999.
- [11] Ronald C. Kube and Eric Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1/2) :85–101, 2000. ISSN : 0921-8890.
- [12] Ronald C. Kube and Hong Zhang. Collective robotics : from social insects to robots. *Adaptive Behaviour*, 1994, 2(2) :189–218, 1992.
- [13] A. Martinoli and F. Mondada. Collective and cooperative group behaviours : Biologically inspired experiments in robotics, 1995.
- [14] M. Mataric, M. Nilsson, and K. Simsarian. Cooperative multi-robot box-pushing, 1995.
- [15] M. W. Moffett. Cooperative food transport by an asiatic ant. *National Geog.*, pages 386–394, 1988.
- [16] F. Mondada, M. Bonani, A. Guignard, S. Magnenat, Studer C., and D. Floreano. Superlinear physical performances in a swarm-bot. In *Proceedings of the VIIIth European Conference on Artificial Life, Lecture Notes in Artificial Intelligence LNAI3630*, pages 282–291. Springer, Berlin, Germany, 2005.

- [17] F. Mondada, G. C. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. De-neubourg, S. Nolfi, L.M. Gambardella, and M. Dorigo. Swarm-bot : a new distributed robotic concept. *Autonomous Robots*, 17(2–3) :193–221, 2004.
- [18] R. O’Grady, R. Groß, A.L. Christensen, F. Mondada, and M. Dorigo M. Bonani. Performance benefits of self-assembly in a swarm-bot. In *Proceedings of the 2007 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS 2007)*, 2007. To Appear.
- [19] Guilherme A. S. Pereira, Vijay Kumar, John Spletzer, Camilo J. Taylor, and Mario F. M. Campos. Cooperative transport of planar objects by multiple mobile robots using object closure. In *Proceedings of the 8th International Symposium on Experimental Robotics (ISER’02)*, Sant’Angelo d’Ischia, Italy, July 2002.
- [20] Carlo Pinciroli. Object retrieval by a swarm of ground based robots driven by aerial robots. Master’s thesis, Université Libre de Bruxelles, Brussels, Belgium, August 2007. Mémoire de DEA.
- [21] K. Sugawara, M. Sano, I. Yoshihara, K. Abe, and T. Watanabe. Foraging behavior off multi-robot system and emergence of swarm intelligence. In *IEEE Internation Conference on Systems, Man and Cybernetics Vol. 3.*, pages 257–262, 1999.
- [22] J. F. A. Traniello and S. N. Beshers. Maximization of foraging efficiency and ressource defense by group retrieval in the ant *Formica schaufussi*. *Behav. Ecol. Sociobiol.*, 29 :283–289, 1991.
- [23] Z. Wang and V. Kumar. Object closure and manipulation by multiple co-operative mobile robots, 2002.