CONFERE'16
7 ET 8 JUILLET 2016, PRAGUE, REP. TCHEQUE.

ROBOTS COLLABORATIFS POUR L'USINE DU FUTUR - INTELLIGENCE EMBARQUÉE

Soufian ZEROUALI, M'hammed SAHNOUN, David TRIMOULET, Fabrice DUVAL, David BAUDRY, Anne LOUIS
CESI-Centre Nord-Ouest, 1 rue Marconi – CS 30285 – 76137 Mont Saint Aignan Cedex

Résumé :
L'usine du futur est une usine connectée, économique et comportant plusieurs éléments autonomes et intelligents tels que les machines et les robots de transport. Ces derniers devraient être capables de se repérer dans l'atelier avec précision et d'effectuer des tâches collaboratives avec un coût abordable.
Le but de ce travail est de concevoir un robot mobile autonomecommunicant à moindre coût et consommation pour effectuer une cartographie intramuros d'un bâtiment. Ce robotest programmé sous ROS (Robot Operating System), l'un des middlewares open-source les plus utilisés dans la communauté robotique de nos jours. Cependant, ROS est souvent implémenté sur des ordinateurs disposant d'importantes capacités de calcul et de stockage, ce qui induit une consommation énergétique élevée, un coût non négligeable et une taille conséquente pour une utilisation sur des petits robots.
Ce papier propose de palier à ce problème par l'utilisation d'une carte Raspberry Pi 3 avec un processeur ARM qui répond parfaitement aux applications que nous allons utiliser. Il expose aussiune présentation rapide du système ROS, son implémentation sur le robot que nous avons développé, l'algorithme d'évitement d'obstacles, l'algorithme de cartographie et le modèle de déplacement du robot ainsi que les résultats expérimentaux.

Mots clés : Robot mobile, ROS, Cartographie, Odométrie, Usine du futur.

1 INTRODUCTION

L'utilisation des robots mobiles pour assurer le transport représente une problématique qui intéresse des chercheurs issus de plusieurs domaines tels que la robotique, l'intelligence artificielle, le génie industriel et l'informatique. La nouvelle génération des ateliers de production sera équipée de plusieurs types de robots mobiles de différentes tailles pour assurer plusieurs tâches telles que la surveillance, l'inventaire et le transport,…[1].Ces systèmes de production représentent des systèmes complexes difficiles à contrôler.

En effet, les méthodes traditionnelles centralisées de gestion de tel système ont montré leurs limites face à la complexité et le haut niveau d'interaction entre les différentes parties du système[2], [3]. Plusieurs chercheurs s'intéressent au pilotage des systèmes de production et notamment les systèmes de transport en introduisant les notions de l'intelligence distribuée, des systèmes autoreconfigurables et des systèmes collaboratifs [4]. Ces systèmes distribués sont basés sur des agents robotiques artificiels autonomes, intelligents et collaboratifs. La première difficulté pour ces robots c'est la localisation et l'interaction sécurisée et précise dans leur environnement.

En dépit de la croissance exponentielle dans le domaine de la robotique, des difficultés sontrencontrées au niveau de développements logiciels compatibles pour des robots distincts. Cependant, il existe des systèmes tels que Robot Operating System (ROS), largement utilisés[5], [6]et permettant de fournir des bibliothèques et des outils pour aider les développeurs de logiciels à créer des applications robotiques.

En outre, plusieurs robots mobiles tels que MiR, Eddie Bot, PMB-2, Lego Mindstorm NXT et Mecanum Bot intégrant le ROS et permettant de faire la localisation et la cartographie ont été développés [7], [8].Plusieurs critères sont utilisés pour évaluer ces robots mobiles tels que le coût, lesdi-

mensions physiques, la consommation énergétique et la capacité de communication. Cependant, aucun d'eux ne les possède tous à la fois.

Le but de ce travail est de concevoir un robot mobile autonomecommunicant qui répond aux caractéristiques listées ci-dessus pour effectuer une cartographie de l'intérieur d'un bâtiment. L'article est organisé comme suit : l'historique ainsi que les principes de fonctionnement du ROS sont présentés dans la section 2. Le modèle proposé regroupant l'architecture matérielle et logicielle du robot sont abordées dans la section 3.Aussi, les résultats des expérimentations menées, ainsi que leurs discussions sont présentés dans la section 4. Finalement, nous terminons avec une conclusion présentée dans la section 5.

2 ROBOT OPERATING SYSTEM (ROS)

ROS est développé et maintenu par la société Willow Garage en 2008, sa philosophie se résume aux principes suivants[6]:

- Peer to Peer.
- Basé sur des outils.
- Multi langages.
- Léger.
- Gratuit et open source.

Le concept fondamental du ROS est de faire fonctionner en parallèle un grand nombre d'exécutables (Nodes) qui doivent pouvoir échanger de l'information de manière synchrone (service) ou asynchrone (Topic)(Figure 1).

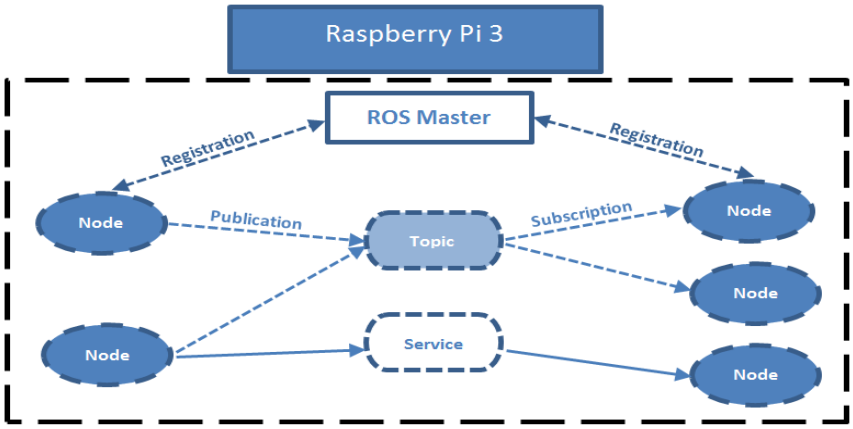

Figure 1. La structure de communication ROS.

ROS débute par l'exécution du maître (Master) qui permet à tous les exécutables ROS (Nodes) de se retrouver et de communiquer entre eux [6].

3 CONCEPTION DU ROBOT MOBILE

3.1.Architecture Matérielle

Comme le montre la Figure 2,les principales options envisageables pour répondre aux besoins du robot sont l'utilisation d'un ordinateur embarqué à petite dimension, faible coûtavec capacité de communication et qui supporte le ROS. Donc nous proposons la carte Raspberry Pi 3 qui répond parfaitement à ces exigences.

Pour interfacer les différents capteurs et actionneurs, nous proposons la carte Arduino utilisée par une grande communauté, à faible coût et surtout pouvant être interfacée avec ROS en utilisant la pile ROSSERIAL.

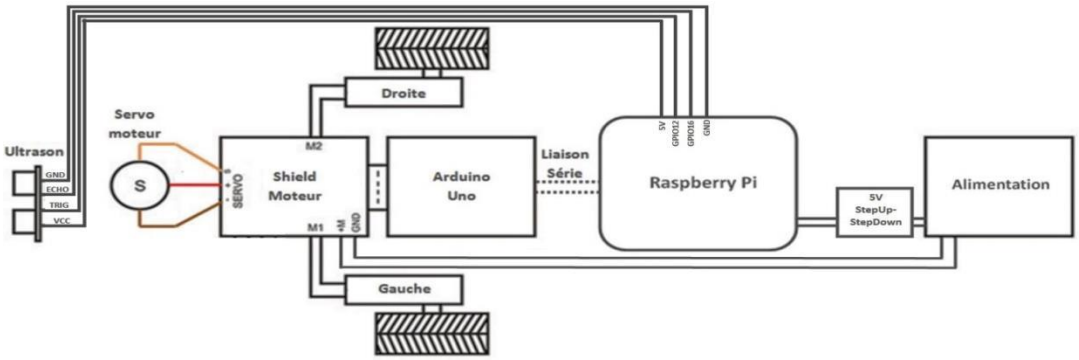

Figure2. Architecture matérielle du robot mobile.

La Figure 3 présente le robot mobile à faible coût que nous avons réalisé en respectant l'architecture matérielle proposée dans la Figure 2.

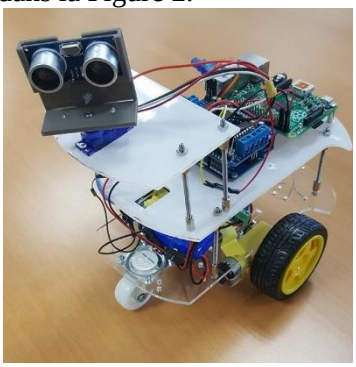

Figure 3. Le robot mobile.

3.2. Architecture Logicielle

Parmi les versions dusystème d'exploitation existantespour le processeur ARM de la Raspberry Pitelles que Rasbian, Pidora, Arch Linux, OSMC,OpenELEC, RISC OS et Windows 10[9], Rasbian JESSIE est le plus adaptéau ROS. Aussi, la pile ROSSERIAL permet d'établir la communication entre ROS sur la Raspberry Pi [10]et la carteArduino[7].

Figure 4. Diagramme d'états et d'actions du robot.

Afin de rendre le robot autonome dans l'évitement des obstacles et le suivi des murs, nous avons développé un algorithme dans un nœud control_node (Figure 4). Sur la partie gauche de la figure 4, le 1 représente une distance supérieure à la distance de sécurité par rapport à l'obstacle et le 0 représente une distance inférieure à la distance de sécurité par rapport à l'obstacle.

L'odométrie est une technique permettant d'estimer la position d'un robot mobile en mouvement. Cette mesure de bas niveau se fait grâce au capteur ultrason qui permet de mesurer les distances entre le robot et son environnement. En partant d'une position initiale connue et en intégrant les déplacements mesurés.

Pour estimer la position du robot et sa trajectoire à partir des mesures odométriques, il est nécessaire de disposer d'un modèle décrivant le déplacement du robot.

Le tableau 1 résume les différentes variables utilisés dans le modèle proposé :

À l'instant t (Figure 5-6)	**À l'instant t+1 (Figure 5-6)**
x_k : La position du robot selon l'axe x. y_k: La position du robot selon l'axe y. Φ_k : L'angle de rotation du robot. d1 : la distance latérale gauche entre le robot et l'obstacle. d2 : La distance longitudinale entre le robot et l'obstacle. d3 : la distance latérale droite entre le robot et l'obstacle.	x_{k+1} : La position du robot selon l'axe x. y_{k+1}: La position du robot selon l'axe y. Φ_{k+1} : L'angle de rotation du robot. d1' : la distance latérale gauche entre le robot et l'obstacle. d2' : La distance longitudinale entre le robot et l'obstacle. d3' : la distance latérale droite entre le robot et l'obstacle.

Tableau 1. Définition des variables utilisés.

Modèle de translation :

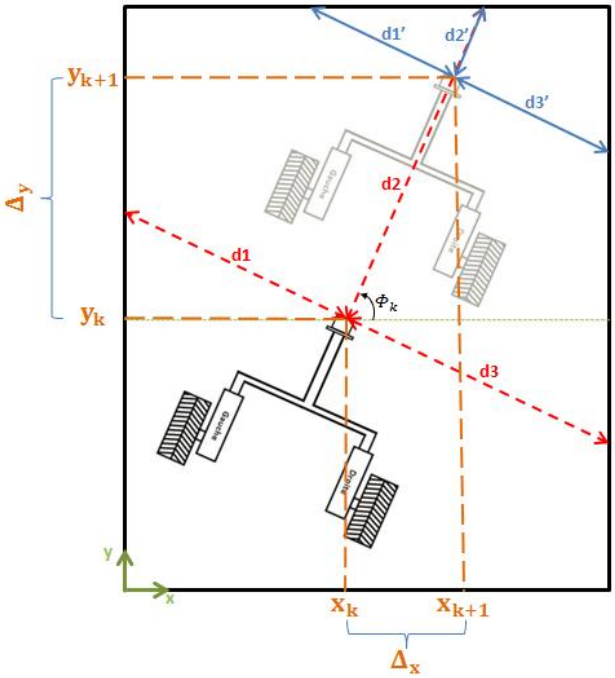

Figure 5. Le modèle de translation.

La position initiale du robot est définie par :

$$\begin{cases} x_k = 0 \\ y_k = 0 \\ \Phi_k = 0 \end{cases} \quad (1)$$

La position du robot à l'instant t+1 est :

$$\begin{cases} x_{k+1} = x_k + \Delta_x \\ y_{k+1} = y_k + \Delta_y \\ \Phi_{k+1} = \Phi_k \end{cases} \quad (2)$$

Avec :

$\Delta_x = (d2 - d2') \times cos(\Phi_k)$: Le déplacement du robot selon l'axe x.
$\Delta_y = (d2 - d2') \times sin(\Phi_k)$: Le déplacement du robot selon l'axe y.

Modèle de rotation :

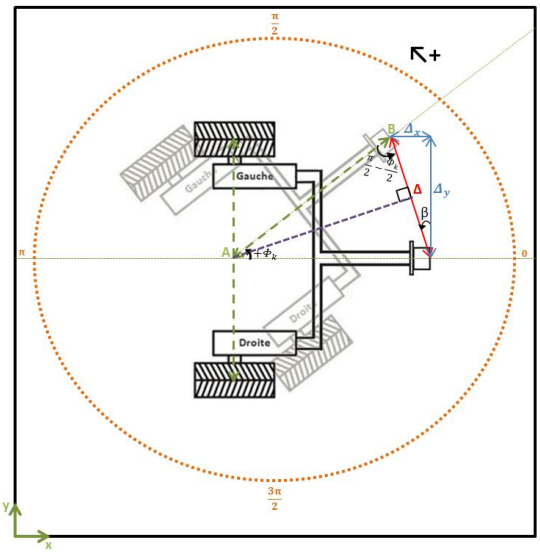

Figure 6.Le modèle de rotation.

$$\begin{cases} \Delta_x = -\Delta \times \cos(\Phi_k - \frac{\theta}{2}) \\ \Delta_y = -\Delta \times \sin(\Phi_k - \frac{\theta}{2}) \end{cases} \quad (3)$$

Avec θ : l'orientation du robot.

Afin de tourner à droite ou à gauche, le robot se base sur les équations (4) et (5) respectivement.

Tourner à droite :

$$\begin{cases} x_{k+1} = x_k + \Delta_x \\ y_{k+1} = y_k + \Delta_y \\ \Phi_{k+1} = \Phi_k + \theta \end{cases} (4)$$

Tourner à gauche :

$$\begin{cases} x_{k+1} = x_k - \Delta_x \\ y_{k+1} = y_k - \Delta_y \\ \Phi_{k+1} = \Phi_k + \theta \end{cases} (5)$$

La Figure 7 montre le schéma de connexion des nœuds.LaRaspberry Pi 3 comporte 2 nœuds l'un dédié au contrôle et l'autre à la connexion. Le nœud, `serial_node`, dédié àétablir la connexion entre la Raspberry Pi 3 et l'Arduino est critique, car il permet la traduction des données entre le port série et le nœud dédié au calcul.Ainsi, il permet de transmettre les informations aux actionneurs (moteurs et servomoteur). Le nœud `control_node`, dédié au contrôle, est en charge de la gestion des obstacles, la cartographie et la gestion de la trajectoire du robot.

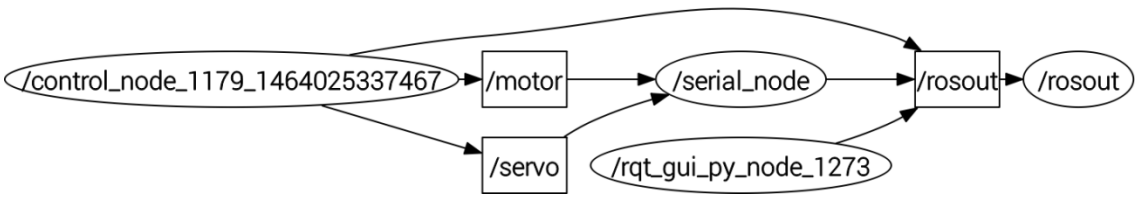

Figure 7.La surveillance en temps réel des nœuds actifs ainsi que les connexions topics des différents nœuds.

4 RÉSULTATS EXPÉRIMENTAUX

Pour démontrer les capacités de la plate-forme développée et l'intégration du logiciel mis au point, deux cas d'études ont été abordés. Dans les deux cas étudiés le robot se localise, parcourt, et construit simultanément une carte de son environnement. Ces tests peuvent être considérés comme une première étape dans l'apprentissage des techniques de SLAM[11].

La Figure 8 représente deux scénarios du parcourt qu'effectue le robot sur un environnement bien définit et la Figure 9 montre respectivement les résultats obtenusen terme de cartographie de l'environnement et estimation de la trajectoire sans passer par des encodeurs au niveau des roues. Dans les deux cas le robot a réussi à cartographier son environnement (Figure 9) et à estimer sa position avec précision. Toutefois, nous avons constaté des limites dans la cartographie traduit par les phénomènes (Phéno 1 et Phéno 2). Comme le montre la Figure 9, la cartographie des coins manque de précision,détecte un point isolé (phéno 2). En effet notre robot se déplace en deux étapes, premièrement il effectue un balayage de 180° de son environnement avec le capteur ultrason en prélevant 3 points (0°, 90°, 180°). Puis il avance en fonction de ce qu'il vient de détecter en suivant l'algorithme de la Figure 4. Et il repasse au balayage. Le phénomène (phéno 2) est dû au pas du servomoteur que nous avons fixé à 90° et aussi au champ du capteur ultrason qui arrive à détecter un mur sur le côté malgré qu'il l'ait déjà dépassé. Aussi, nous avons constaté des erreurs dues au capteur ultrason, qui se sont traduites par une non linéarité du balayage du robot voir (Phéno 1).

Figure 8.Scénarios de la cartographie.

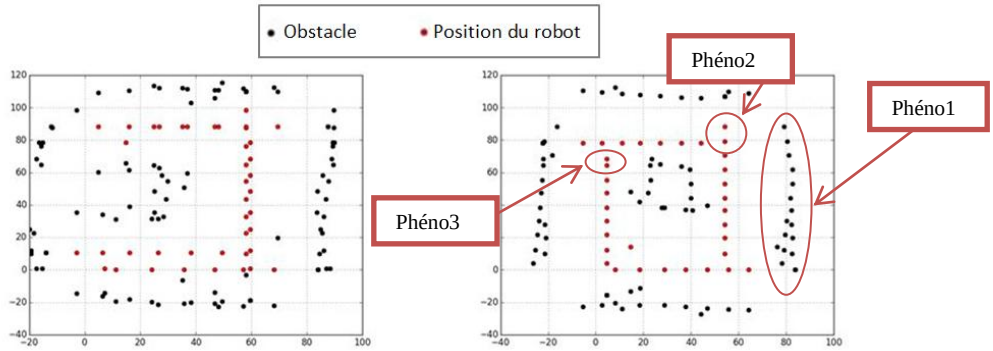

Figure 9.Résultats des scénarios de la cartographie.

La Figure 9, présente aussi un rapprochement entre les points de déplacement du robot (Phéno 3)qui est due à un problème purement technique, celui de la batterie. En effet, quand la charge de la batterie diminue, le moteur manque de puissance qui se répercute sur son déplacement, notamment la rotation qui nécessite plus de puissance. Quand la batterie n'arrive pas à déplacer le robot avec le même pas habituel, l'algorithme arrive à détecter correctement ce petit déplacement.

Afin de remédier aux problèmes de Phéno3, nous allons développer un régulateurde tension qui permettra de réguler le pas du déplacement du robot. Nous proposons aussi l'augmentation du-

nombre de prélèvements faite par le capteur (diminution du pas de servomoteur). Cela va augmenter le nombre de points dans notre carte et augmenter ainsi la précision de la cartographie et de la trajectoire.

Nous avons effectué un chiffrage du robot que nous avons développé et nous l'avons comparé à d'autres robots de la même gamme avec des capacités similaires. Le tableau 2 positionne notre robot par rapport aux robots mobiles bas coûts existants. Et démontre que nous avons proposé une solution intéressante en termes de coût pour des fonctionnalités comparables.

				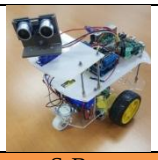
	TraxBot	Lego MindstormNXT	RoombaiCreate	S-Bot
Coût	485,00€	290,00€	180,00€	90,00€
ROS	✓	✓	✓	✓
Communication	- USB - Zigbee	- Bluetooth - USB	- RS-232 - USB	- WiFi - Bluetooth - USB
Capteur	- 3 × Ultrason	- Accéléromètre - Tactile - LDR - Son - Ultrason	- Temperature - IR - Micro-interrupteurs à détection falaise.	- Ultrason
Batterie	12V	9V	14.8 V	12V

Tableau 2. Comparaison des robots mobiles.

5 CONCLUSION

Dans ce papier nous avons présenté le développement d'un robot mobile basé sur le ROS, pour la cartographie d'un bâtiment. Nous avons démontré par expérimentation qu'il est possible de réaliser la cartographiegrâce à la combinaison des distances et des anglesdonnés par le capteur ultrason et le servomoteur qui le guide.Le robot réagit de manière autonomeavec son environnement et permet d'obtenir des cartes assez précises en utilisant uniquement un capteur ultrason.

Toutefois, notre approche comporte des lacunes auxquelles nous allons répondre tout en tenant compte de nos exigences de départ qui sont le coût, les dimensions physiques, la consommation énergétique et la capacité de communication. Plusieurs méthodes peuvent être utilisées pour l'amélioration de la cartographie, notamment l'utilisation d'une couche de filtrage de la carte pour éliminer les points inutiles ou l'utilisation d'une méthode de fusion de données telle que les réseaux bayésiens pour l'amélioration de la position du robot.

La suite du travail consiste à l'utilisation de plusieurs robots pour la cartographie mais avec des capacités différentes. L'idée sera de faire corriger la position d'un robot par les indications des autres robots qui arrivent à le voir.

REMERCIEMENTS :

Ces travaux se déroulent dans le cadre du projet XTERM soutenu par le fonds FEDER et la Région Normandie.

REFERENCES

[1] G. K. Kraetzschmar, N. Hochgeschwender, W. Nowak, F. Hegger, S. Schneider, R. Dwiputra, J. Berghofer, and R. Bischoff, "Robocup@ work: competing for the factory of the future," in *RoboCup 2014: Robot World Cup XVIII*, Springer, 2014, pp. 171–182.

[2] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[3] P. Leitão, J. Barbosa, and D. Trentesaux, "Bio-inspired multi-agent systems for reconfigurable manufacturing systems," *Eng. Appl. Artif. Intell.*, vol. 25, no. 5, pp. 934–944, 2012.

[4] D. Trentesaux, "Distributed control of production systems," *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 971–978, 2009.

[5] A. Martinez and E. Fernández, *Learning ROS for robotics programming*. Packt Publishing Ltd,

2013.
[6] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, 2009, vol. 3, no. 3.2, p. 5.
[7] A. Araújo, D. Portugal, M. S. Couceiro, and R. P. Rocha, “Integrating Arduino-based educational mobile robots in ROS,” *J. Intell. Robot. Syst.*, vol. 77, no. 2, pp. 281–298, 2015.
[8] M. Karimi, A. Ahmadi, N. K. Korghond, E. Babaians, and S. S. Ghidary, “ReMoRo; A mobile robot platform based on distributed I/O modules for research and education,” in *Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on*, 2015, pp. 657–662.
[9] E. G. Sierra and G. A. R. Arroyave, “Low cost SDR spectrum analyzer and analog radio receiver using GNU radio, raspberry Pi2 and SDR-RTL dongle,” in *2015 7th IEEE Latin-American Conference on Communications (LATINCOM)*, 2015, pp. 1–6.
[10] L. Joseph, *Mastering ROS for Robotics Programming*. Packt Publishing Ltd, 2015.
[11] M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *Robot. Autom. IEEE Trans.*, vol. 17, no. 3, pp. 229–241, 2001.

Contact principal: Soufian ZEROUALI

Coordonnées : szerouali@cesi.fr