



HAL
open science

Co-conception de Robots Mobiles Commandés en réseaux sans fil

Amine Mechraoui, Jean-Marc Thiriet

► **To cite this version:**

Amine Mechraoui, Jean-Marc Thiriet. Co-conception de Robots Mobiles Commandés en réseaux sans fil. QUALITA 2011 - 9ème Congrès international pluridisciplinaire en Qualité et Sécurité de Fonctionnement, Mar 2011, Angers, France. pp.95. hal-00594372

HAL Id: hal-00594372

<https://hal.science/hal-00594372>

Submitted on 19 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Co-conception de Robots Mobiles Commandés en réseaux sans fil

Amine MECHRAOUI

Département d'automatique

GIPSA-lab, Université de Grenoble (UJF) 38402, France

Email : amine.mechraoui@gipsa-lab.inpg.fr

Jean-Marc THIRIET

Département d'automatique

GIPSA-lab, Université de Grenoble (UJF) 38402, France

Email : jean-marc.thiriet@gipsa-lab.inpg.fr

Résumé—Cet article traite le problème de la co-conception des systèmes commandés à travers un réseau sans fil (WNCS). L'intégration du réseau sans fil (WN) dans la boucle de commande influence la Qualité de Contrôle (QoC) du système en terme de la Qualité de Service (QoS) du réseau sans fil. Nous présentons une approche de co-conception basée sur des réseaux bayésiens distribués. Cette approche permet de prendre des décisions pour assurer une bonne QoC pour le robot et aussi faire en sorte que la QoS soit toujours suffisante pour maintenir une bonne QoC. Maintenir une QoS suffisante dépend de la méthode d'ordonnement utilisée et donc l'approche proposée vise également à déterminer une méthode d'ordonnement pour maintenir les performances du système.

I. INTRODUCTION

Les systèmes commandés en réseau (Networked Control Systems NCSs) entrent dans le cadre des systèmes distribués dans lesquels les informations entre le nœud contrôleur et le nœud système sont échangées via un réseau de communication filaire ou sans fil. Les NCSs ont trouvé leurs applications dans différents domaines tels que les réseaux de capteurs mobiles [1], la robotique et la robotique mobile [2], [3], les UAV (Unmanned Aerial Vehicles UAV) [4]. Cependant, l'intégration du réseau dans la boucle de commande introduit quelques problèmes difficiles à résoudre.

La commande des systèmes traditionnellement connus est basée sur l'hypothèse que toutes les informations transmises sont reçues sans erreur, sans pertes et sans retard, ce qui signifie que le lien entre les différents composants du système est dit "parfait". Par contre, dans les systèmes commandés en réseau, le réseau que ce soit filaire ou sans fil est "imparfait". C'est le réseau donc qui fait la différence entre un système commandé classique et un système commandé à travers un réseau et rend la modélisation et l'analyse des NCSs plus complexe [5]. L'influence du réseau sur la QoC du système est due généralement à : l'échantillonnage, le retard, les pertes de paquets et le débit du réseau utilisé. Beaucoup de travaux ont été faits récemment concernant les NCSs. Selon les différents aspects de l'analyse et la conception des NCSs, les travaux existants peuvent être généralement divisés en quatre catégories : l'analyse de stabilité [6], [7], l'estimation d'état [8], [9], la synthèse de commande [10], [11] et de la co-conception du contrôle et de la communication [12], [13], [14], [15].

L'approche proposée dans cet article traite le problème de

co-conception et est basée sur une méthode probabiliste vu le comportement stochastique des réseaux sans fil. Cette approche est basée sur les réseaux bayésiens distribués. Elle vise le contrôle de flux capteur pour garantir à la fois une bonne QoC et une bonne QoS.

L'article est organisé comme suit. Dans un premier temps, nous allons présenter le modèle du robot ainsi que l'architecture de commande via un réseau sans fil dans le cas d'atteinte de cible et évitement d'obstacles. Puis, le problème de co-conception sera discuté et l'approche proposée sera présentée. Ensuite, l'utilisation de cette approche pour contrôler le flux capteur sera présentée ainsi qu'une comparaison avec la méthode classique d'ordonnement sera illustrée avec deux scénarios. Enfin, une conclusion et des perspectives.

II. ARCHITECTURE DE LA COMMANDE VIA UN RÉSEAU SANS FIL

Cette section se focalise sur la présentation du système choisi, à savoir, le robot Khepera III (voir Fig. 1) et son modèle cinématique et dynamique. Le robot Khepera III est un véhicule autonome, muni de deux roues motrices indépendantes. Le couple est transmis par deux moteurs à courant continu identiques. Les moteurs sont équipés d'encodeurs magnétiques (12 impulsions par millimètre du déplacement du robot) pour donner la position angulaire et linéaire du robot par rapport à un repère fixe. Le robot est muni d'un processeur Motorola 68331 @25MHz et deux microcontrôleurs dsPIC et un PIC18F4431. L'OS du robot est un Kernel 2.6.25.7-kb1 et celui du Korebot un ARM Embedded Linux. Il contient 9 capteurs infrarouges et 5 ultrasons.

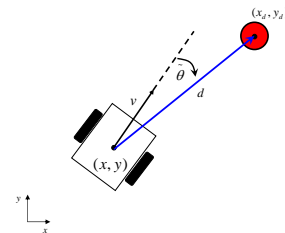


FIG. 1. Robot Khepera III et son modèle

A. Modèle du robot

1) *Modèle cinématique du robot*: Le robot étudié est unicycle. Il est actionné par deux roues indépendantes. Le schéma des robots de type unicycle est donné sur la Fig. 1. Le déplacement du robot est décrit par un vecteur $M = [x \ y \ \theta]^T$ où le couple (x, y) désigne les coordonnées du centre de masse du robot par rapport au repère $R(O, \vec{x}, \vec{y})$ et l'angle θ représente l'orientation du véhicule par rapport à l'axe (O, \vec{x}) de ce même repère.

On considère que l'axe de rotation passe par le centre de masse du robot. Le modèle cinématique est donné par les équations suivantes :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (1)$$

Le modèle cinématique du robot a deux entrées ω_{left} et ω_{right} , vitesses angulaires de la roue gauche et droite respectivement. Ces vitesses angulaires sont liées directement à la vitesse linéaire v et angulaire ω du robot.

2) *Modèle dynamique des moteurs du robot*: Le modèle dynamique du robot est caractérisé par les équations des deux moteurs ou leurs fonctions de transfert. La fonction de transfert est trouvée par identification et est une fonction du premier ordre donnée comme suit :

$$\frac{\omega_{(left, right)}}{U} = \frac{K}{\tau s + 1} \quad (2)$$

Où U représente la tension appliquée aux deux moteurs et $\omega_{(left, right)}$ représente la vitesse angulaire de chaque roue.

B. Architecture de commande via un réseau

Deux niveaux de commande sont implémentés. Le premier niveau concerne la commande des deux roues en vitesse angulaire en utilisant un contrôleur PI discret pour chacune des roues motrices. Le deuxième niveau de commande concerne la régulation en vitesse linéaire et angulaire du robot en utilisant les lois de commande non-linéaires proposées par [16] dans le cas d'atteinte de cible et [17] dans le cas d'évitement d'obstacles. Le réseau sans fil a été intégré dans la boucle de commande entre le contrôleur et le robot d'un côté et entre les capteurs et le contrôleur d'un autre côté comme présenté sur la Fig. 2.

L'outil de simulation des systèmes temps réel TrueTime [18] a été utilisé pour simuler le comportement du réseau sans fil. Deux tâches ont été programmées. La première est la tâche contrôleur qui génère un flux contrôleur et la deuxième concerne la tâche capteur qui, à son tour, génère le flux capteur. La tâche contrôleur est guidée par les événements, c'est-à-dire que le contrôleur calcule et envoie la commande $(\omega_{r_k}, \omega_{l_k})$ à condition qu'il ait reçu toutes les mesures qui sont envoyées par le capteur $(\hat{x}_k, \hat{y}_k, \hat{\theta}_k)$. Le mode d'accès au medium utilisé dans notre cas est le CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).

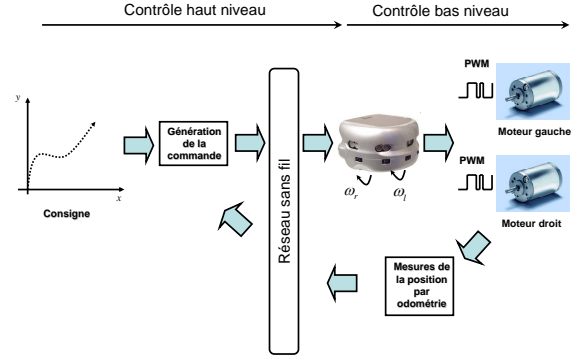


FIG. 2. Architecture de commande via un réseau sans fil

1) *Atteindre une cible*: Pour que le robot puisse rejoindre la position désirée définie par $[x_d \ y_d \ \theta_d]$, des erreurs sont calculées :

$$\tilde{x} = x_d - x, \quad \tilde{y} = y_d - y \quad (3)$$

l'erreur de suivi (Eq. 4a) et l'erreur d'orientation (Eq. 4b) sont calculées comme suit :

$$d = \sqrt{\tilde{x}^2 + \tilde{y}^2} \quad (4a)$$

$$\tilde{\theta} = \theta_d - \theta = \tan^{-1}(\tilde{y}/\tilde{x}) - \theta \quad (4b)$$

Selon [16], les lois de commande suivantes sont considérées :

$$v = \frac{v_{max}}{1 + |d|} d \cos(\tilde{\theta}) \quad (5a)$$

$$\omega = \frac{v_{max}}{1 + |d|} \cos(\tilde{\theta}) \sin(\tilde{\theta}) + K_{\tilde{\theta}} \tanh(k_{\tilde{\theta}} \tilde{\theta}) \quad (5b)$$

Avec v_{max} la vitesse linéaire maximale que le robot peut atteindre ($v_{max} = 0.3m/s$) et $K_{\tilde{\theta}}, k_{\tilde{\theta}}$ sont des constantes.

Les valeurs des commandes générées par le contrôleur ont été discrétisées en mettant un échantillon après le contrôleur. Comme les erreurs reçues par le contrôleur sont discrètes, un bloqueur d'ordre zéro est placé avant le contrôleur.

2) *Évitement d'obstacles*: Pour éviter un ou plusieurs obstacles, l'algorithme de cycle limite proposé dans [17] pour les robots unicycles a été utilisé.

On considère que le robot doit atteindre une cible avec comme coordonnées (X_{cible}, Y_{cible}) . On considère aussi qu'il y a un obstacle circulaire avec les coordonnées (X_{obs}, Y_{obs}) et un rayon de R_{sec} avec : $R_{sec} = R_{obs} + D_{sec}$. R_{obs} est le rayon réel de l'obstacle circulaire et D_{sec} représente la marge de sécurité (D_{sec} est égal au rayon du robot (65mm) augmentée d'une constante, dans notre expérimentation 100mm).

En fonction des données fournies par les différents capteurs du robot, la décision de commuter entre le contrôle pour atteindre une cible et celui pour l'évitement d'obstacle sera prise. Si $D_{robot-obstacle}$ (distance entre les centres de gravité du robot et de l'obstacle) est inférieure à R_{sec} le robot décide de commuter son contrôleur vers celui d'évitement d'obstacles. Les différentes étapes de l'algorithme d'évitement d'obstacles sont présentées dans [3].

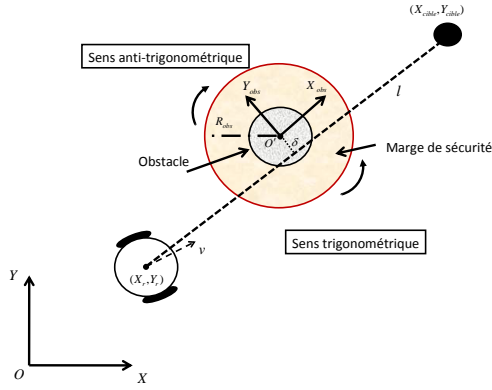


FIG. 3. Evitement d'obstacles

C. Influence de l'intégration du réseau dans la boucle de commande

Pour bien étudier l'influence de la charge du réseau sur la QoC du robot, on peut distinguer deux cas à savoir, l'atteinte d'une cible sans obstacle et avec obstacle.

1) *Atteinte d'une cible sans obstacle*: Dans ce cas de figure, le robot doit atteindre une cible $([X_d, Y_d] = (1, 1))$ sans qu'un obstacle soit sur sa trajectoire. On charge le réseau avec une tâche externe (un autre robot qui envoie différents types de trames ie. trame vidéo, mesures ...). La charge a varié, en changeant la longueur de la trame externe, entre 0% et 100%. Le retard de la trame capteur, dans le cas de 50% de charge, ne dépasse pas la période d'échantillonnage (entre 200 et 250ms), et le retard mesuré de la trame contrôleur est entre 15 et 20ms ce qui n'influence pas d'une façon remarquable la performance du système (le robot rejoint la cible avec la trajectoire souhaitée Fig. 4). Dans ce cas, le retard du flux capteur est dix fois plus grand que le retard du flux contrôleur car la tâche capteur est guidée par le temps et la tâche contrôleur est guidée par les événements ce qui influence le moment d'envoi de la trame contrôleur et donc le retard.

Dans le cas de 90% de charge, le retard du flux capteur a augmenté (entre 350 et 400ms) ce qui est normal parce que la tâche externe a mis beaucoup plus de temps à être acheminée. Pour la tâche contrôleur, le retard reste constant vu qu'il n'y a pas de concurrence avec les autres tâches (pas de collision). Au delà de 95% de charge (Fig. 4 et Fig. 5), on remarque que la station perd complètement le contrôle et le robot ne peut pas atteindre la cible. Le retard dans ce cas dépasse dans certains cas la période d'échantillonnage, que ce soit pour le flux capteur ou le flux contrôleur ce qui dégrade la QoC du robot.

2) *Atteinte d'une cible avec obstacle*: Dans cette partie, on suppose qu'il y a un obstacle entre la position initiale du robot et la cible à atteindre. En changeant la charge du réseau, l'influence de cette charge sur la performance de l'algorithme d'évitement d'obstacle a été étudiée. Cette performance est caractérisée par la distance entre le centre du robot et le centre de l'obstacle (le robot ne doit pas toucher l'obstacle). La Fig.6

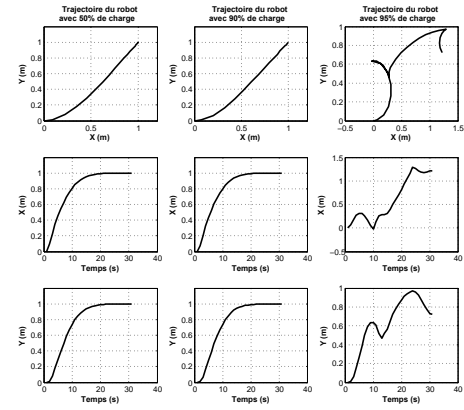


FIG. 4. Trajectoire du robot en simulation pour atteindre une cible en fonction de la charge du réseau

montre l'influence de la charge sur la distance entre le robot et l'obstacle. En effet, l'influence de la charge du réseau est beaucoup plus remarquable dans ce cas que dans le cas où il n'y a pas d'obstacle. Entre 0% et 54% de charge, le robot contourne l'obstacle sans y toucher (Fig.6). Par contre, à partir de 55% de charge, le robot n'a pu l'éviter (Fig.6). Cela est dû au retard qui est généré par la tâche externe.

III. PROBLÈME DE CO-CONCEPTION

Le problème de co-conception dans notre cas, est d'essayer d'adapter le réseau aux contraintes de la commande pour garantir une bonne QoC du robot, et aussi adapter la commande aux besoins du réseau pour assurer toujours une bonne QoS du réseau. Dans les travaux de [14], une solution a été proposée. En effet, une stratégie de changement de station (Handover) a été proposée pour adapter le réseau aux contraintes de la commande et une stratégie de commutation de contrôleur vers un contrôleur interne dégradé a été aussi proposée pour adapter la commande et assurer un minimum de QoS. Dans ce papier, nous proposons une stratégie d'ordonnancement qui prend en considération la QoC du robot et la QoS du réseau. Pour cela, on a besoin d'outil d'aide à la décision qui, en fonction des différents paramètres qui influencent le réseau, peut quantifier

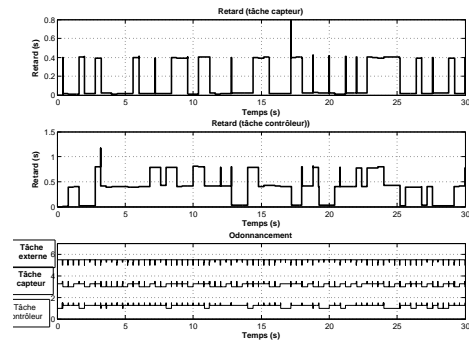


FIG. 5. Retards (capteur-contrôleur et contrôleur-actionneur) et ordonnancement des différentes tâches avec 95% de charge

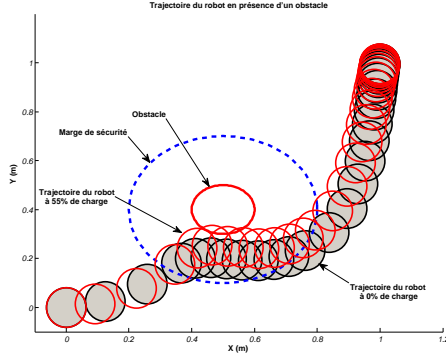


FIG. 6. Trajectoire du robot avec obstacle en fonction de la charge du réseau

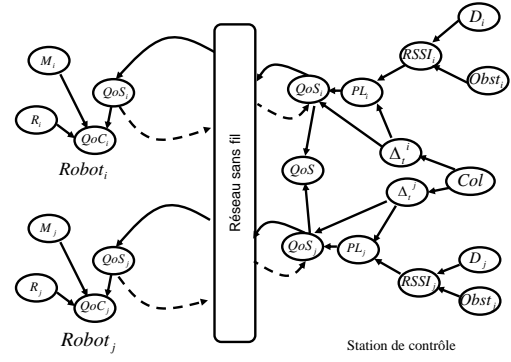


FIG. 7. Réseau Bayésien Distribué (RBD)

la QoS du réseau et aussi la QoC du robot en estimant leurs états, à savoir, bonne, dégradée ou mauvaise pour contrôler le flux capteur. Dans cette section, nous proposons d'utiliser les Réseaux Bayésiens Distribués (RBD) pour l'aide à la décision et le contrôle de flux des tâches capteurs.

A. Contrôle de flux capteur par RBD

Les systèmes embarqués autonomes sont limités en terme de mémoire, temps de calcul et aussi d'énergie. Pour cela, l'intégration des RB dans ce genre de systèmes n'est pas souhaitable vu la complexité en terme de calcul et de mémoire de ces derniers, ce qui entraîne une consommation d'énergie importante et ensuite diminue leur autonomie. Pour remédier à ce problème, une stratégie de distribution du RB proposé dans [19] a été étudiée. La Fig. 7 montre l'architecture de distribution du RB en question pour un environnement multi-robots (Tab.I montre la signification de chaque nœud du RBD et leurs modes). Seuls les nœuds qui sont observés par le robot (à savoir la QoC, QoS, l'état du robot et la mission) sont embarqués sinon le reste sera débarqué (dans la station de contrôle). Des messages entre chaque partie du RBD sont envoyées via le réseau sans fil sous forme de trame de contrôle (la trame la plus petite possible). Les modes et les probabilités conditionnelles a priori de chaque nœud sont trouvés par apprentissage de paramètres en utilisant la base de données provenant du comportement réel du robot Khepera III en utilisant le logiciel Bayesia-Lab. Tous les nœuds du RB proposé sont mesurable sauf la QoS et la QoC qui sont calculées en utilisant ces deux équations 6 pour la QoS et 7 pour la QoC :

$$QoS = \max\left(\frac{(1 - PL) * (nT_s - \Delta_t)}{nT_s}, 0\right) \quad (6)$$

PL étant la probabilité de pertes de paquets, elle est définie par un retour d'expérience sur le système réel. Δ_t est le retard entre le capteur et le contrôleur. T_e est la période d'échantillonnage. n est une constante qui caractérise le nombre maximal d'échantillons perdus que l'on peut tolérer pour que le système fonctionne parfaitement en débarqué. La constante n est définie pour que le véhicule mobile ne heurte pas un obstacle.

$$QoC = QoS * \kappa_j * \varrho_i \quad (7)$$

κ_j étant une variable qui dépend de l'état j du robot tel que $j \in \{\text{Bon, dégradé ou mauvais}\}$ et $\kappa_{bon} > \kappa_{deg} > \kappa_{mauvais}$; dans notre cas on a pris $\kappa_{bon} = 1, \kappa_{deg} = 0.5$ et $\kappa_{mauvais} = 0$. ϱ_i est une variable qui dépend de la mission i du système ($i \in \{\text{obstacle, cible}\}$) tel que $\varrho_{obstacle} | \varrho_{cible}$ ($\varrho_{cible} = 1$ et $\varrho_{obstacle} = 0.5$).

1) *Contrôle de flux capteur par RBD*: Après avoir proposé un modèle basé sur les RB, nous procédons maintenant à l'utiliser pour contrôler les flux capteurs en fonction des informations fournies par le RBD sur la QoS et la QoC.

En fonction de l'état de la QoC (donné par l'Algorithme 1) et l'état de la QoS $P(QoS | Observations)$ (donné par le RB proposé et en appliquant l'Algorithme 2), un signal est fourni pour contrôler le flux capteur.

B. Résultats

Pour vérifier l'efficacité de l'approche proposée en termes de décision et contrôle de flux capteur, deux scénarios sont présentés. Dans chaque scénario, le RBD et l'Algorithme 1 sont testés et comparés avec la stratégie d'ordonnancement guidée par le temps. L'utilisation de l'Algorithme 1 sans le RBD est comparée avec l'Algorithme 2. (voir Table II). Pour comparer les différentes stratégies de contrôle de flux

TAB. I
SIGNIFICATION DES DIFFÉRENTS NŒUDS DU RBD ET LEURS MODES

Nœuds	Signification	Modes
QoC	Qualité de Contrôle	Bonne, dégradée, mauvaise
QoS	Qualité de Service	Bonne, dégradée, mauvaise
R	Etat du robot	Bon, dégradé, défaillant
M	Mission	Cible, Obstacles, Trajectoire
PL	Perte de Paquets	$0\% \leq PL < 40\%$, $40\% \leq PL < 60\%$, $60\% \leq PL < 100\%$
RSSI	Request Signal Strength Indicator	$RSSI > -48dBm$, $RSSI < -48dBm$
Δ_t	Retard agissant sur la QoS	$\Delta_t < T_e$, $\Delta_t \geq T_e$
Obst	Obstacle	Oui, Non
D	Distance entre la station et le robot	
Col	Collision	≤ 0.8 , > 0.8
N	Nombre de nœuds	≤ 14 , > 14

Algorithm 1 Contrôle de flux capteur

```
if  $d = 0$  and  $\dot{d} = 0$  then
   $QoS = \text{good}$  and  $sensor \leftrightarrow controller$ 
else if  $\dot{d} < 0$  then

  if  $\dot{\theta}(k) \geq 0$  and  $\dot{\theta}(k-1) \geq 0$  then
     $QoS = \text{degraded}$  and  $sensor \leftrightarrow controller$ 
  else
     $QoS = \text{good}$  and  $sensor \leftrightarrow controller$ 
  end if
else
   $QoS = \text{bad}$  and  $sensor \leftrightarrow controller$ 
end if
```

\leftrightarrow : communication, \leftrightarrow : stop communication

Algorithm 2 Contrôle du flux capteur : BN + Algo 1

```
 $P_1 = P(QoS = \text{good} | \text{Observations})$ 
 $P_2 = P(QoS = \text{degraded} | \text{Observations})$ 
 $P_3 = P(QoS = \text{bad} | \text{Observations})$ 
if  $(P_1 > P_2)$  and  $(P_1 > P_3)$  then
   $sensor \leftrightarrow controller$ 
else
  Execute Algorithm 1
end if
```

capteur, des critères ont été proposés pour évaluer l'efficacité de chaque stratégie :

- 1) Le premier critère (Eq. 8) est la demande du réseau (Use Request factor (URF)). ce paramètre influence directement la QoS du réseau.

$$URF = \sum_{i=1}^N \frac{\xi_i}{\Gamma * T_s^i} \quad (8)$$

ξ_k est la longueur de la trame de la tâche i . Γ est le débit. N est le nombre de tâches.

- 2) Le second critère concerne le pourcentage de perte de paquets (Eq. 9).

$$\Phi = \frac{P_r}{P_s} \quad (9)$$

P_s est le nombre de paquets envoyés et P_r est le nombre de paquets reçus.

- 3) Le troisième critère est pour évaluer la QoC. Il est calculé en utilisant un critère quadratique (Integral of the Absolute Error (IAE)) (voir Eq.10).

$$IAE = \sum_{k=0}^p \epsilon_x(k)^2 + \epsilon_y(k)^2 \quad (10)$$

ϵ_x et ϵ_y sont les erreurs entre les points de la courbe de référence (à 0% de charge de réseau) et les points des autres courbes (en fonction de la charge du réseau) à l'instant t suivant les axes X et Y respectivement. k est le nombre de points représentant chaque courbe.

- 4) Le quatrième critère concerne l'arrivée à la position désirée ou non.
- 5) Le dernier est la collision avec un obstacle ou non.

C. Scénario 1

Dans le premier scénario, nous supposons que nous avons deux robots qui communiquent avec la station de contrôle. La position initiale du R1 est $(x_0, y_0, \theta_0) = (0, 0, 0)$ la position de la cible est $(x_d, y_d) = (1, 1)$. Les deux robots sont toujours dans la zone de couverture de la station, donc $P(D < D_{seuil}) = 1$ pour chaque robot.

Pour travailler dans des conditions d'un réseau chargé, Nous initialisons le débit à 8000 bits/s . Avec ce débit l'URF = 64%. Dans ce cas, le robot arrive à la cible mais il ne peut pas éviter un obstacle (voir Fig. 6). En appliquant l'Algorithme 1 pour contrôler le flux capteur, un signal binaire est généré ("0" pour l'envoi des mesures capteur et "1" pour arrêter la communication). On remarque dans ce cas, l'URF diminue de 64% à 34%, La QoS redevient bonne mais l'IAE augmente de 0.13 à 0.89 pour R1 et de 0.42 à 1.36 pour R2 (voir Table II). L'application de cet Algorithme dans ces conditions n'a pas influencé l'objectif d'atteindre la cible ou d'éviter les obstacles. Cependant, en appliquant l'Algorithme 2 (RBD+Algo1) a permis de diminuer l'IAE vu qu'il a pris en considération l'état de la QoS (suffisante pour envoyer).

D. Scénario 2

Dans ce deuxième scénario, les mêmes conditions du scénario 1 ont été reprises en changeant seulement le débit à 3000 bits/s pour que le réseau soit plus chargé, dans ce cas l'URF = 170%. l'IAE a augmenté de 0.13 à 66.35. Les pertes de paquets de 0 à 90% et donc le robot n'a pas pu atteindre la cible désirée (Fig. 8) et il n'a pas pu éviter l'obstacle (voir Table II). En revanche, en appliquant l'Algorithme 1, l'URF a diminué de 170% à 82%. En conséquent, le robot a pu rejoindre la cible mais il n'a pas pu éviter l'obstacle. En utilisant l'algorithme 2 le robot a pu atteindre la cible (voir Fig. 8) et éviter l'obstacle. La Fig. 9 montre l'amélioration du retard en appliquant l'approche proposée.

IV. CONCLUSION

Dans cet article, une nouvelle approche basée sur les RB distribuées a été proposée pour contrôler le flux capteur dans un système commandé en réseau sans fil. Cette approche prend en considération, dans un problème de co-conception, la QoS du réseau et la QoC des robots mobiles. Le RB a été construit en utilisant l'apprentissage de paramètres via une base de données. Cette base de données a été enregistrée du comportement réel du robot Khepera III utilisé comme Benchmark. En fonction de la situation, les probabilités fournies par le RBD génèrent un signal de contrôle du flux capteur pour contrôler l'envoi de mesure du capteur vers le contrôleur et ainsi garder une bonne QoC et aussi une bonne QoS. Cette stratégie d'ordonnancement a été ensuite comparée avec la stratégie classique (envoi chaque période d'échantillonnage) et les résultats sont satisfaisants.

TAB. II
COMPARISON

		Scénario1	Scénario2	
\overline{RBD} Algo1	URF	64%	170%	
	Φ	$R1$	0%	91%
		$R2$	0%	96%
		S	0%	91%
	IAE	$R1$	0.13	66.35
		$R2$	0.42	49.97
	Atteindre la cible		Oui	Non
	Collision		Oui	Oui
\overline{RBD} Algo1	URF	34%	82%	
	Φ	$R1$	0%	32%
		$R2$	0%	33%
		S	0%	64%
	IAE	$R1$	0.89	1.31
		$R2$	1.36	1.76
	Atteindre la cible		Oui	Oui
	Collision		Non	Oui
RBD Algo1	URF	64%	79%	
	Φ	$R1$	0%	33%
		$R2$	0%	30%
		S	0%	56%
	IAE	$R1$	0.21	0.87
		$R2$	0.59	1.02
	Atteindre la cible		Oui	Oui
	Collision		Non	Non

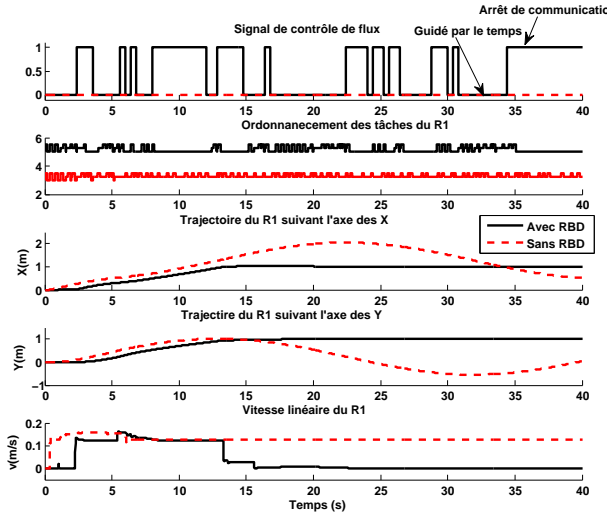


FIG. 8. Comparaison entre la trajectoire du R1 avec et sans RBD

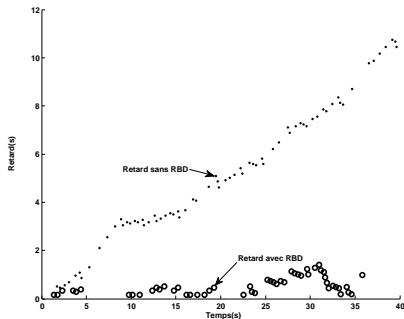


FIG. 9. Comparaison entre le retard de la tâche capteur du R1 avec et sans RBD

L'utilisation des réseaux bayésiens dynamiques est envisageable pour intégrer, en plus de l'aspect diagnostique et prise

de décision, un aspect pronostic (prévoir ou anticiper la dégradation de la QoS et de la QoC) pour éviter la dégradation de la QoC et la QoS avant qu'elle ne se produise. Le même réseau bayésien pourrait être utilisé en ajoutant quelques nœuds dynamiques dans deux ou plusieurs laps de temps.

RÉFÉRENCES

- [1] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis. Collective motion, sensor networks, and ocean sampling. In *Proceedings of the IEEE*, volume 95, pages 48–74, 2007.
- [2] T. Val, P. Fraisse, and D. Andreu. Vers l'utilisation de bluetooth pour la commande à distance de robots mobiles. In *Journal Européen des Systèmes Automatisés*, volume 37, pages 859–892, 2003.
- [3] A. Mechraoui, J.-M. Thiriet, and S. Gentil. Control of an intelligent robot system over a wireless network. In *Proceeding of the 23rd IAR Workshop on Advanced Control and Diagnosis, Coventry, United Kingdom*, 2008.
- [4] C. Berbra, S. Gentil, and S. Lescecq. Hybrid priority scheme for networked control quadrotor. In *17th Mediterranean Conference on Control and Automation, MED'09, Grèce*, 2009.
- [5] J.P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. In *Proceeding of IEEE special issue on Technology of Networked Control Systems*, volume 95, pages 138–162, 2007.
- [6] G. C. Walsh, O. Beldiman, and L. G. Bushnell. Asymptotic behavior of nonlinear networked control systems. In *IEEE Transaction on Automatic Control*, volume 46, pages 1093–1097, 2001.
- [7] W. Zhang, M.S. Branicky, and S.M. Phillips. Stability of networked control systems. In *IEEE Control Systems Magazine*, volume 21, pages 84–99, 2001.
- [8] N. Elia and J. Eisembeis. Limitation of linear control over packet drop networks. In *Proceedings of 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, volume 5, pages 5152–5157, 2004.
- [9] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry. Kalman filtering with intermittent observations. In *IEEE Transactions on Automatic Control*, volume 49, pages 1453–1464, 2004.
- [10] D. H. Varsakelis and L. Zhang. LQG control of networked control systems with access constraints and delays. In *International Journal of Control*, volume 81, pages 1266–1280, 2008.
- [11] P. Naghshtabrizi and J. P. Hespanha. Designing an observer-based controller for a network control system. In *Proceedings of the 44th Conference on Decision and Control, and the European Control Conference, Seville, Spain*, pages 848–853, 2005.
- [12] M. S. Branicky, S. M. Phillips, and W. Zhang. Scheduling and feedback co-design for networked control systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA*, 2002.
- [13] C. Aubrun, D. Simon, and Y.-Q. Song. *Co-design Approaches for Dependable Networked Control Systems*. 2010.
- [14] A. Mechraoui, Z. H. Khan, J.-M. Thiriet, and S. Gentil. Co-design for wireless networked control of an intelligent mobile robot. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO), Milan, Italy*, 2009.
- [15] Z. H. Khan, D. Genon-Catalot, and J.-M. Thiriet. Co-design in heterogeneous wireless networked control systems. In *International Multiconference on Computer Science and Information Technology, Mragowo, Pologne*, 2009.
- [16] J. M. Toibero, R. Carelli, and B. Kuchen. Switching control of mobile robots for autonomous navigation in unknown environments. In *Proceeding of IEEE International Conference on Robotics and Automation, Roma, Italy*, 2007.
- [17] D. Kim and J. Kim. A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. In *Robotics and Autonomous Systems*, volume 42, pages 17–30, 2003.
- [18] M. Anderson, D. Henriksson, and A. Cervin. Truetime 1.5 reference manual. Technical report, 2007.
- [19] A. Mechraoui, J.-M. Thiriet, and S. Gentil. Aide à la décision et diagnostic par réseaux bayésiens d'un robot mobile commandé en réseau. In *Sixième Conférence Internationale Francophone d'Automatique Nancy, France*, 2010.