المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique d'Alger

Département d'Automatique

ojet de Fin d'Etudes en vue de l'obtention des diplômes
d'Ingénieur d'état et de Master en Automatique

Thème

Optimisation par Reinforcement Learning et implémentation d'une technique V-SLAM(2D) sur un robot mobile

Présenté par :
Oussama DEROUICHE
El Hacene CHABANE

Dirigé par :
Pr. Mohamed TADJINE
Mr. Zeryab MOUSSAOUI

24 Juin 2018

Plan de travail

1. Introduction
2. Techniques utilisées
3. Navigation autonome
4. Mise en œuvre pratique
5. Conclusion et perspectives

Présentation du sujet

ECE PARIS
ÉCOLE D'INGÉNIEURS

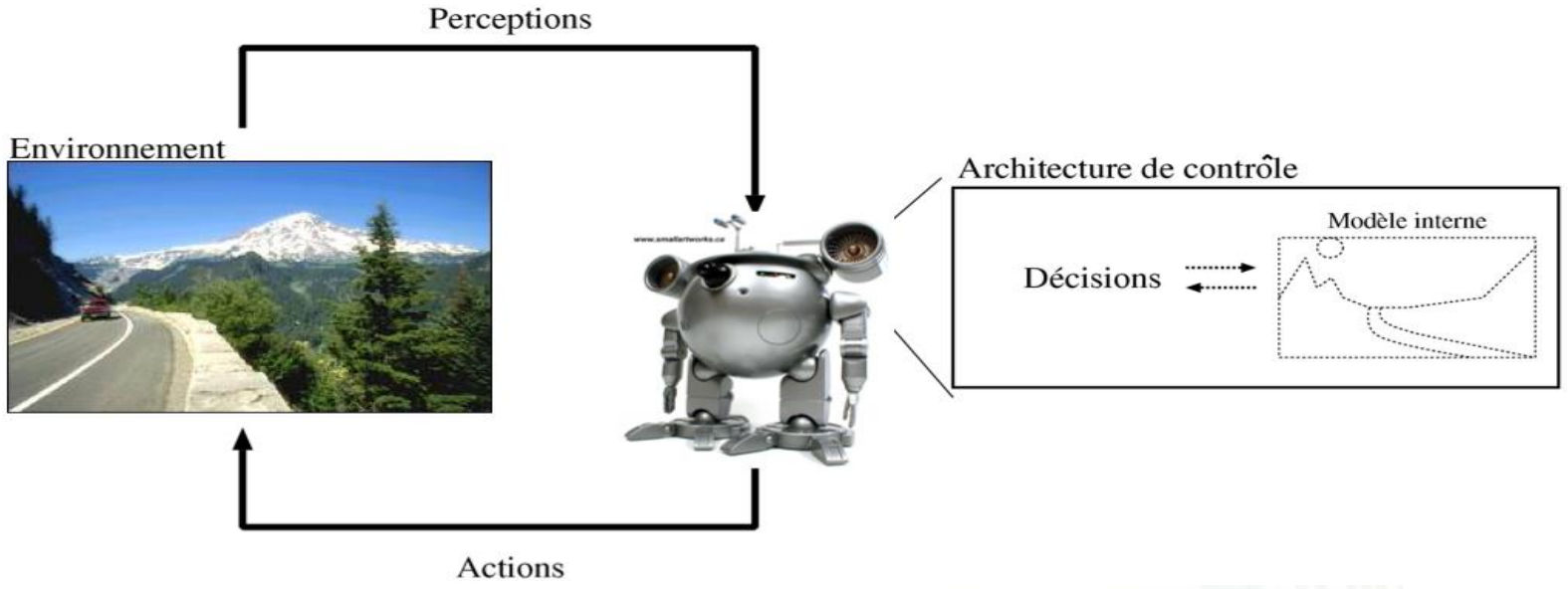
المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

ECEBORG

مخبر التحكم في العمليات التصنيعية
LCP
Laboratoire de Commande des Processus

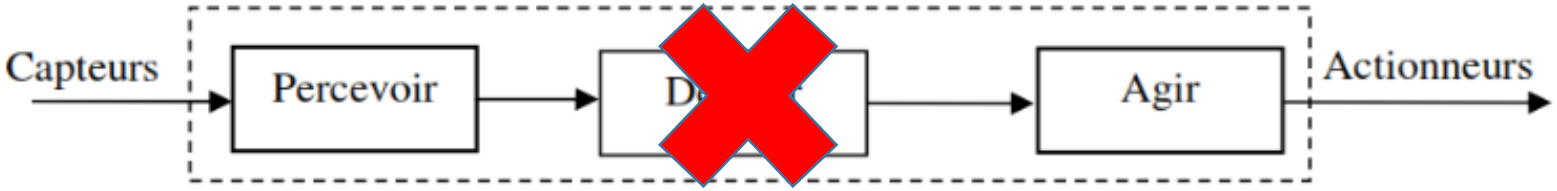
Robotique mobile

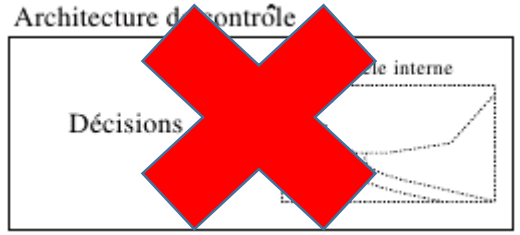
- On distingue robotique de manipulation et robotique mobile.
- Un robot mobile est une machine intelligente capable de se mouvoir dans un environnement.



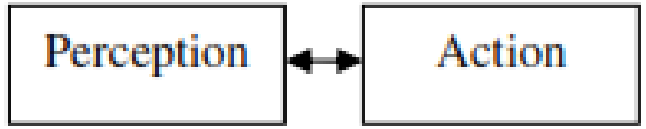
Robotique mobile

- Architecture hiérarchique





- Architecture réactive



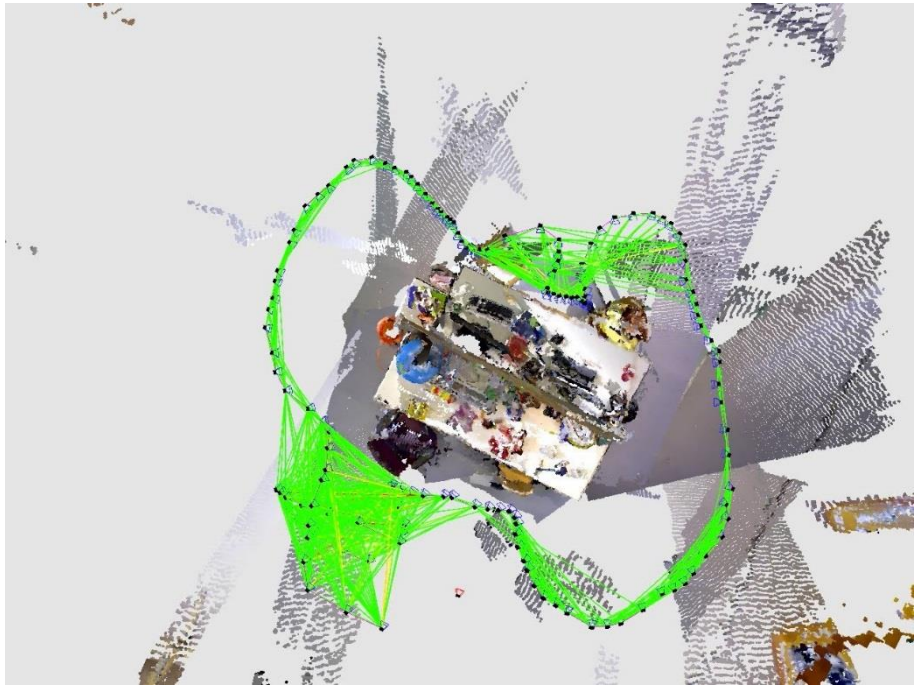
Filliat. 2016

Problématiques

- La Navigation, problématique principale de la robotique mobile.
- La Localisation, problématique secondaire mais essentielle.
- Solution retenue : Navigation Floue
- Optimisation par Reinforcement Learning.
- Localisation : incrémentale ou par fermeture de la boucle.
- Implémentation du VSLAM pour gain de précision.

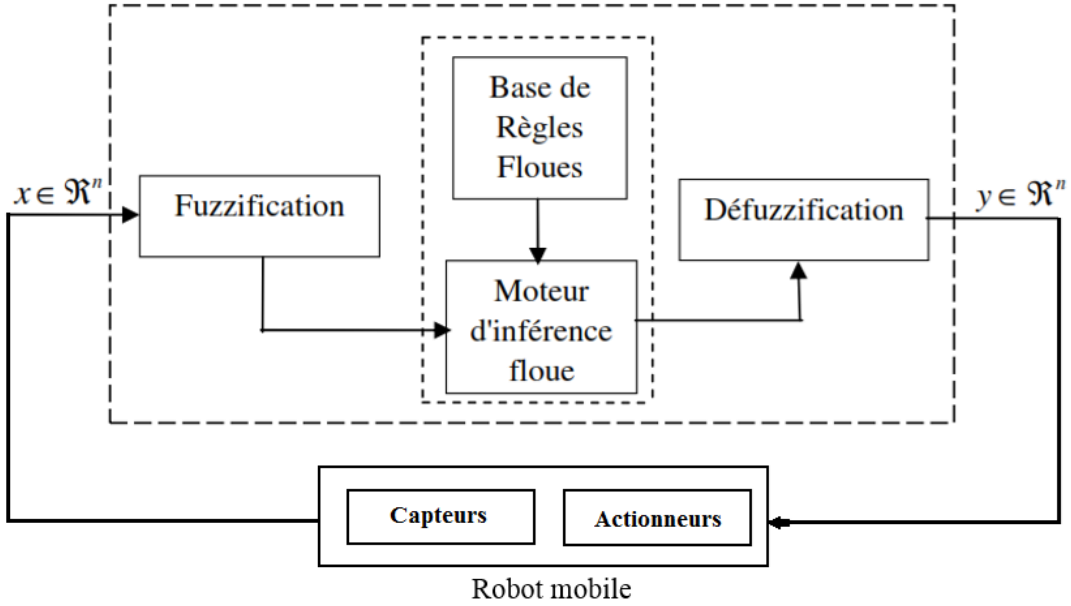
Visual SLAM

- ***S**imultaneous **L**ocalisation **A**nd **M**apping*, avec (au moins) une caméra.
- Plus fiable et plus robuste que l'Odométrie.
- Monoculaire : ORB-SLAM2 (2017), LSDSLAM (2015) MonoSLAM (2007), PTAM(2007)
- RGBD : ORB-SLAM2(2017), RGB-D SLAM (2014)
- Solution retenue : ORB-SLAM2
- ORB SLAM2 est récent, polyvalent et open-source.



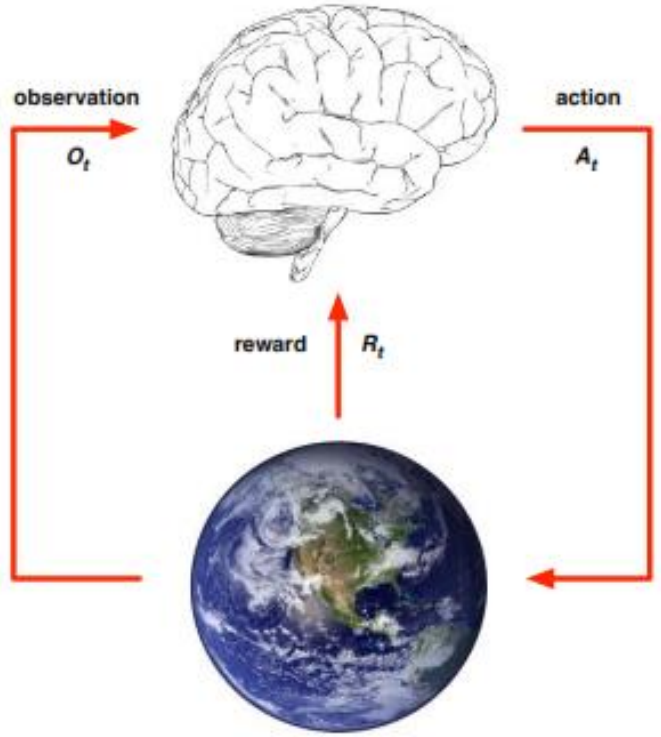
Logique floue

- Extension de la logique, introduite par le Professeur iranien Lotfi Zadeh en 1965.
- Application : Automatique, Traitement Images, …
- Choix : Régulateur Mamdani



Apprentissage par renforcement

- Branche du *Machine Learning*.
- Interactions d'un agent avec l'environnement.
- Formalisé par *Markov Decision Process* $< \mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R}, \gamma >$
- Recherche d'une politique optimale maximisant les récompenses.
- Fonctions de décision $Q(s,a)$ ou $V(s)$ à estimer :



$$V_\pi(s) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right]$$

$$Q_\pi(s,a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

Sutton. 1998

Apprentissage par renforcement

Méthodes de la programmation dynamique :

Policy Iteration

$$V_{k+1}(s) = \sum_{a \in \mathbb{A}(s)} \pi(a \mid s) \left(\mathbb{R}_s^a + \gamma \sum_{s' \in \mathbb{S}} \mathbb{P}_{s,s'}^a V_k(s') \right)$$

Value Iteration

$$V_{k+1}(s) = \max_{a \in \mathbb{A}(s)} Q_\pi(s,a) = \max_{a \in \mathbb{A}(s)} \left(\mathbb{R}_s^a + \gamma \sum_{s' \in \mathbb{S}} \mathbb{P}_{s,s'}^a V_k(s') \right)$$

Méthode de Monte Carlo :

$$V_{k+1}(s) = V_k(s) + \frac{1}{k+1} \left[G_{k+1}(s) - V_k(s) \right] \quad \text{ou} \quad Q_{k+1}(s,a) = Q_k(s,a) + \alpha \left[G_{k+1}(s) - Q_k(s,a) \right]$$

Méthodes de *Temporal difference learning :*

Q Learning

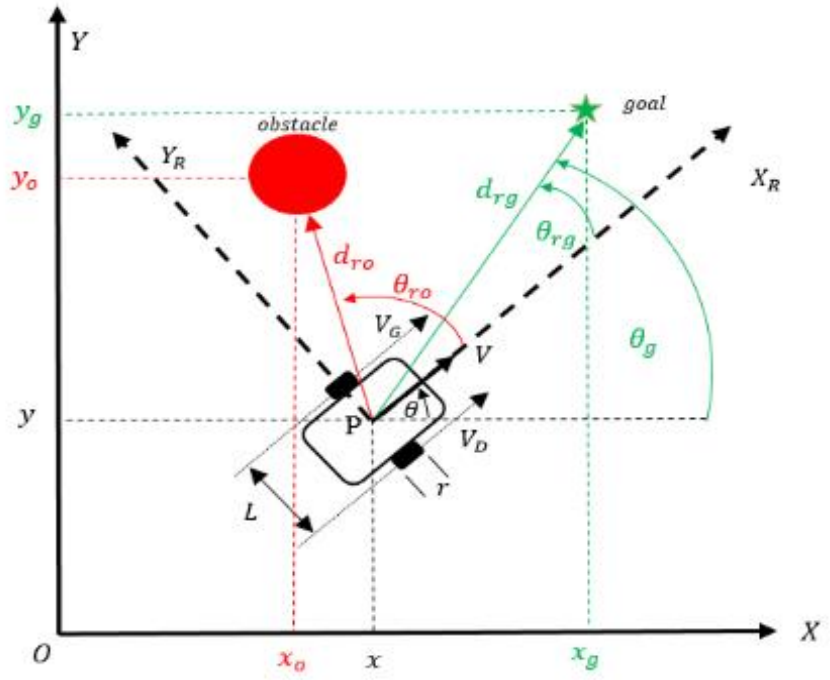
$$Q(s_t,a_t) \leftarrow (1-\alpha) Q(s_t,a_t) + \alpha \left[R_t + \gamma \max_{a \in \mathbb{A}(s_{t+1})} Q(s_{t+1},a) \right]$$

SARSA

$$Q(s_t,a_t) \leftarrow (1-\alpha) Q(s_t,a_t) + \alpha \left[R_t + \gamma Q(s_{t+1},a_{t+1}) \right]$$

Sutton. 1998

Modélisation



Configuration

Odométrie :

$$x(k+1) = x(k) + r\frac{T}{2}(\omega_D(k) + \omega_G(k))\cos(\theta(k))$$

$$y(k+1) = y(k) + r\frac{T}{2}(\omega_D(k) + \omega_G(k))\sin(\theta(k))$$

$$\theta(k+1) = \theta(k) + r\frac{T}{2}(\omega_D(k) - \omega_G(k))$$

But :

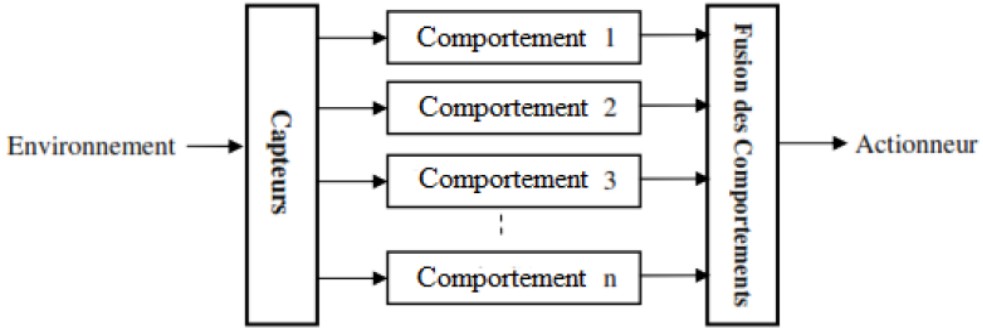
$$d_{rg} = \sqrt{(x_g - x)^2 + (y_g - y)^2}$$

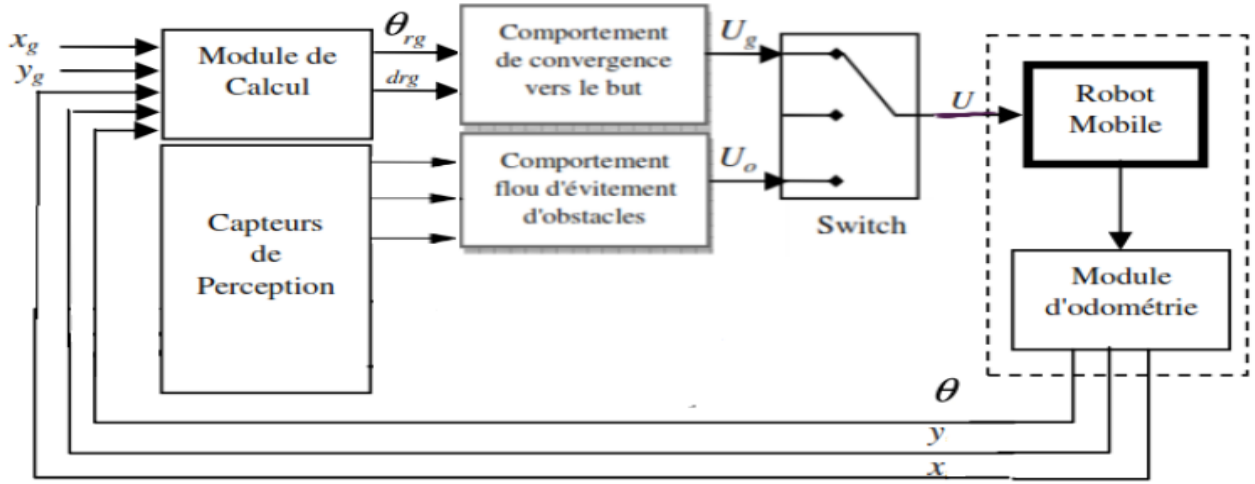
$$\theta_{rg} = \arctan(\frac{y_g - y}{x_g - x}) - \theta$$

Synthèse des régulateurs flous

Architecture de commande

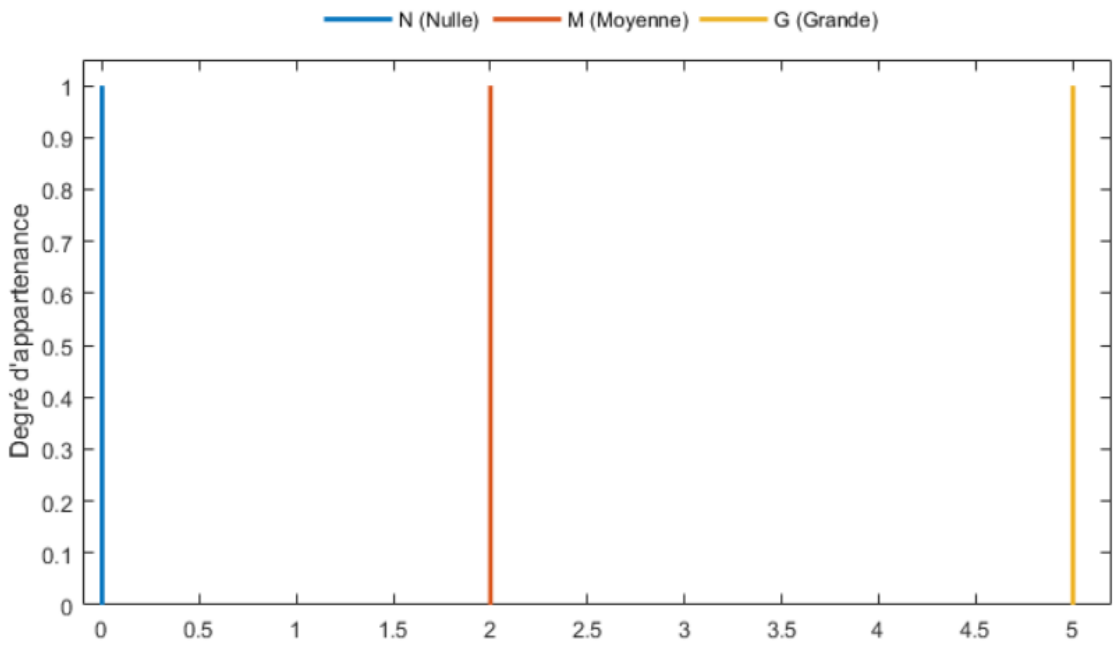
- Navigation réactive par logique floue.
- Comportement de convergence vers un but.
- Comportement d'évitement d'obstacles.





Synthèse des régulateurs flous

Régulateur de convergence vers un but (FLC20)



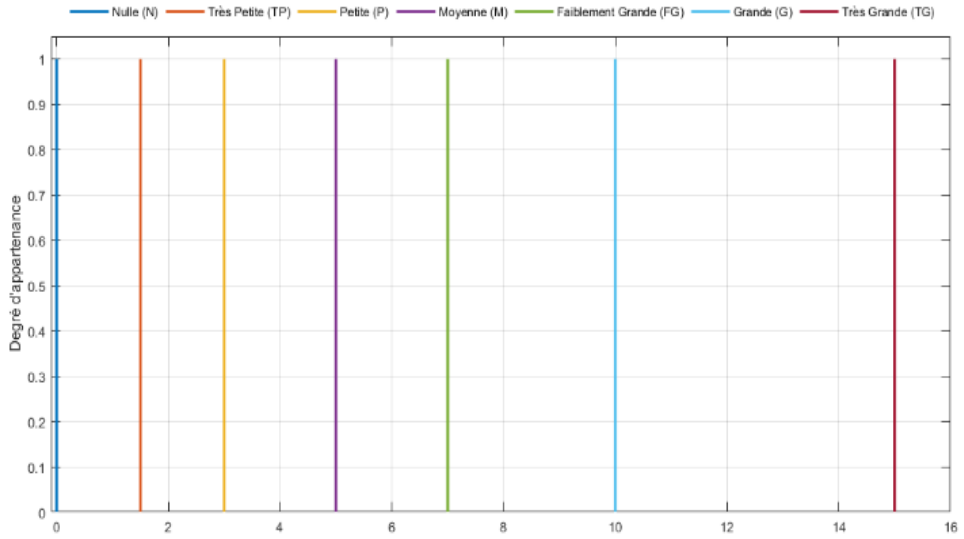
Fonctions d'appartenance des sorties de FLC20 (Convergence)

d_{rg} \ θ_{rg}		**N**	**NP**	**Z**	**PP**	**P**
N	ω_D	*M*	*M*	*N*	*N*	*N*
	ω_G	*N*	*N*	*N*	*M*	*M*
P	ω_D	*G*	*G*	*M*	*M*	*M*
	ω_G	*M*	*M*	*M*	*G*	*G*
M	ω_D	*G*	*G*	*M*	*M*	*M*
	ω_G	*M*	*M*	*M*	*G*	*G*
G	ω_D	*G*	*G*	*M*	*M*	*M*
	ω_G	*M*	*M*	*G*	*G*	*G*

Les règles de FLC20

Synthèse des régulateurs flous

Régulateur de convergence vers un but (FLC49)



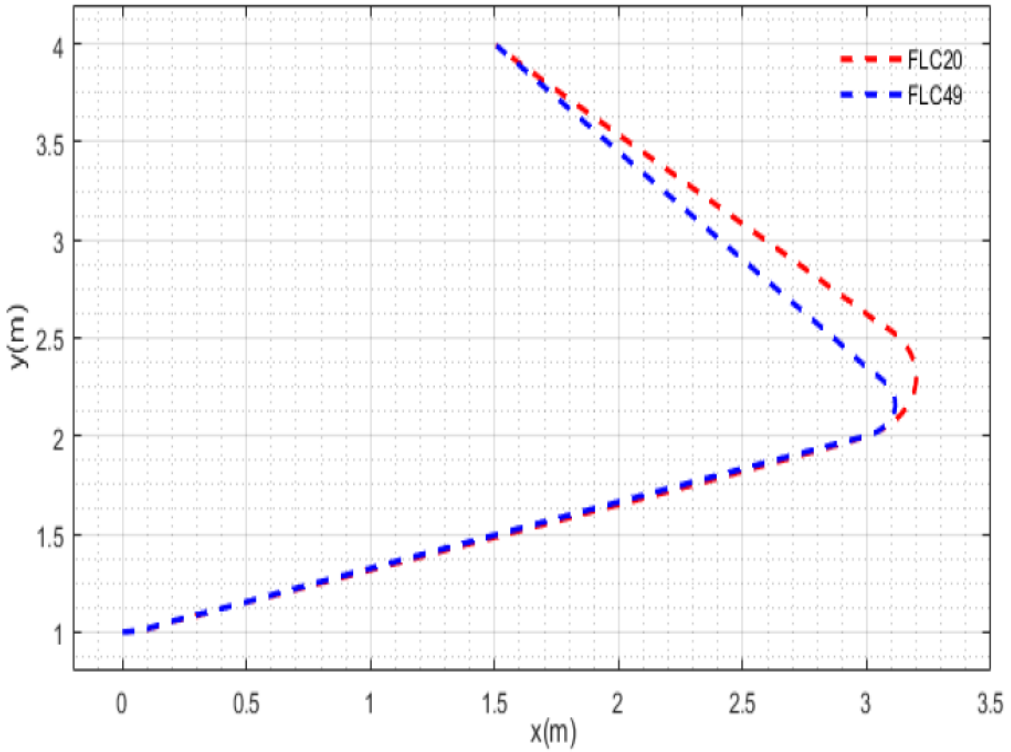
Fonctions d'appartenance des sorties de FLC49 (Convergence)

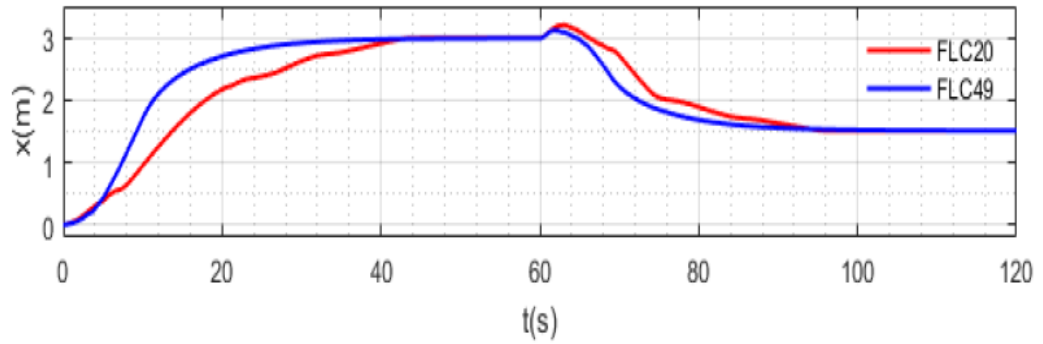
d_{rg} \ θ_{rg}		NG	NM	NP	Z	PP	PM	PG
N	ω_D	N	N	N	N	P	P	M
	ω_G	M	P	P	N	N	N	N
TP	ω_D	TP	TP	N	TP	M	FG	G
	ω_G	G	FG	M	TP	N	TP	TP
P	ω_D	TP	TP	TP	P	FG	G	TG
	ω_G	TG	G	FG	P	TP	TP	TP
M	ω_D	TP	TP	TP	M	G	G	TG
	ω_G	TG	G	G	M	TP	TP	TP
MG	ω_D	TP	TP	P	FG	FG	G	TG
	ω_G	TG	G	FG	FG	P	TP	TP
G	ω_D	TP	TP	M	G	FG	G	TG
	ω_G	TG	G	FG	G	M	TP	TP
TG	ω_D	TP	TP	P	TG	FG	G	TG
	ω_G	TG	G	FG	TG	P	TP	TP

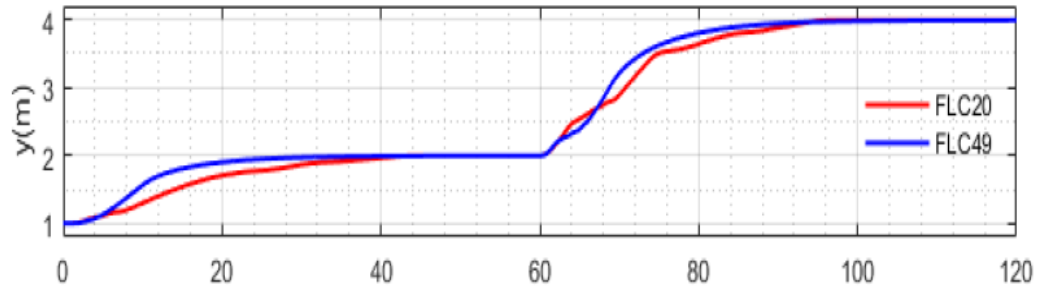
Les règles de FLC49

Synthèse des régulateurs flous

Résultats de comparaison (convergence vers un but)



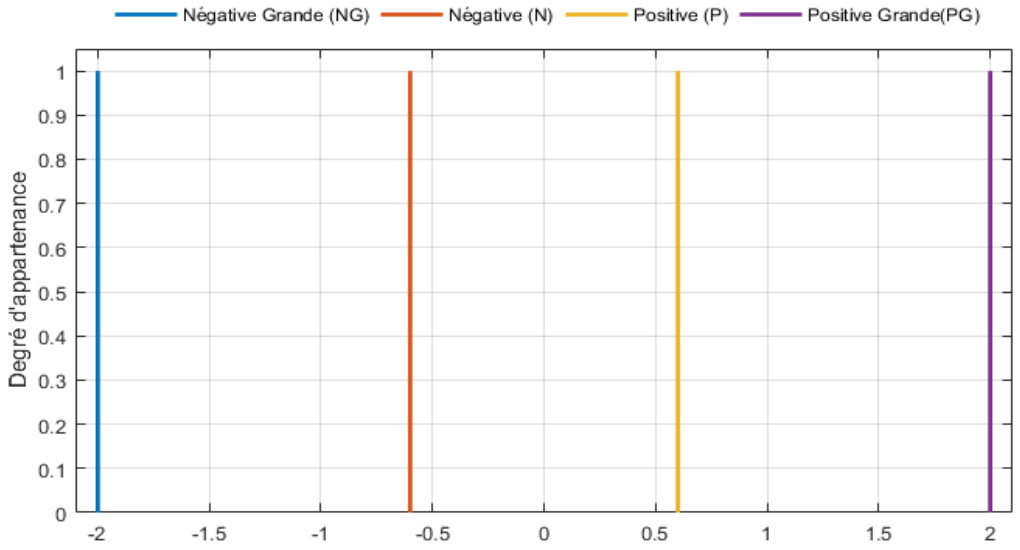




Comparaison

Synthèse des régulateurs flous

Régulateur d'évitement d'obstacles



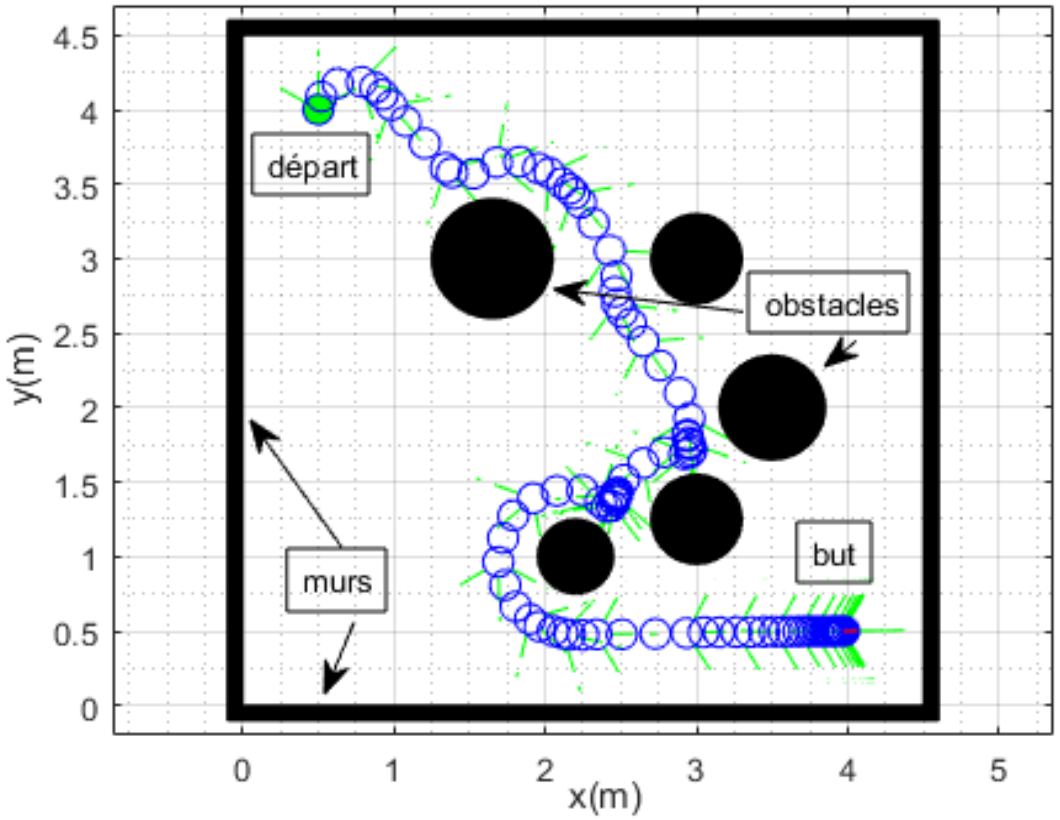
L	P	P	NG	PG
L	P	M	NG	PG
L	P	L	PG	NG
L	M	P	N	PG
L	M	M	P	PG
L	M	L	P	PG
L	L	P	N	PG
L	L	M	P	PG
L	L	L	P	P

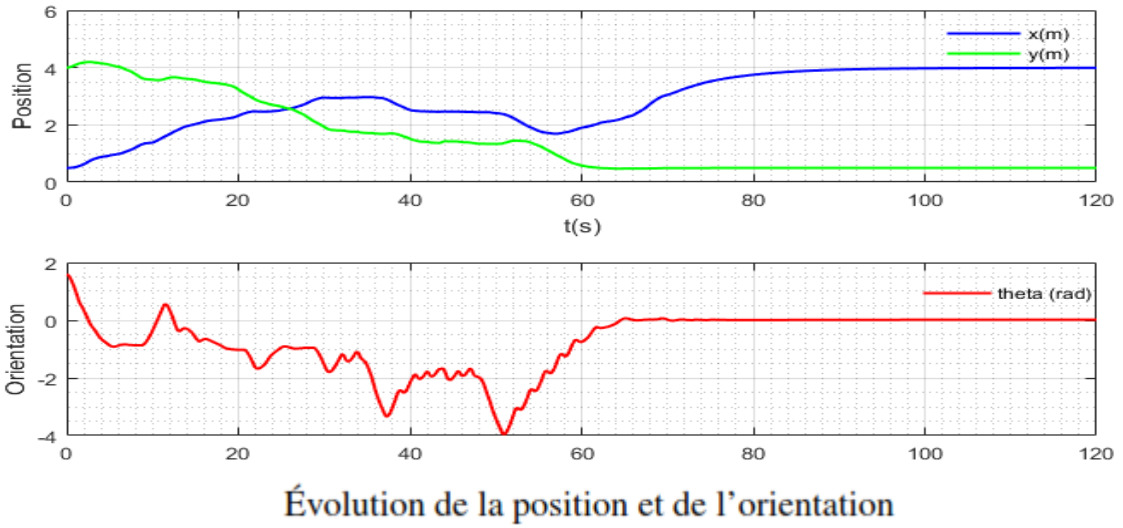
Entrées			Sorties	
d_{ro}^{G}	d_{ro}^{A}	d_{ro}^{D}	ω_G	ω_D
P	P	P	NG	NG
P	P	M	PG	NG
P	P	L	PG	NG
P	M	P	NG	NG
P	M	M	PG	N
P	M	L	PG	N
P	L	P	NG	NG
P	L	M	P	N
P	L	L	P	N

M	P	P	NG	PG
M	P	M	NG	PG
M	P	L	PG	NG
M	M	P	N	PG
M	M	M	NG	NG
M	M	L	PG	N
M	L	P	N	PG
M	L	M	P	P
M	L	L	PG	P

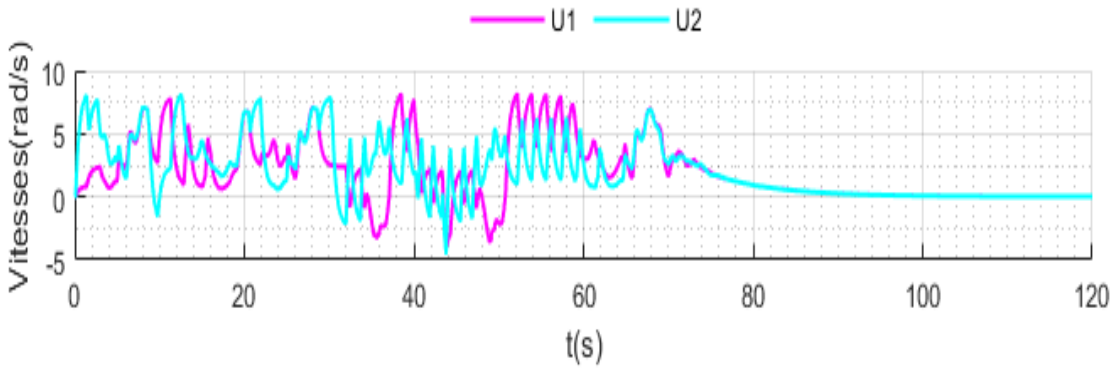
Synthèse des régulateurs flous

Résultats de simulation (évitement d'obstacles)





Évolution de la position et de l'orientation



Commandes appliquées

Synthèse des régulateurs RL flous

Apprentissage des règles par Fuzzy Q-Learning

Pour chaque épisode d'apprentissage

Pour chaque règle i

$$a_i = argmax_k q[i,k] \text{ with probability } 1-\epsilon$$
$$a_i = random\{a_k, k = 1, 2, \cdots, J\} \text{ with probability } \epsilon$$

$$a = \sum_{i=1}^{N} \alpha_i(s(t))\, a_i$$

$$Q(s(t), a) = \sum_{i=1}^{N} \alpha_i(s) \times q[i, a_i]$$

$$V(s(t+1)) = \sum_{i=1}^{N} \alpha_i(s(t+1)).max_k(q[i, q_k]).$$

$$\Delta Q = r(t+1) + \gamma \times V_t(s(t+1)) - Q(s(t), a)$$

$$q[i, a_i] = q[i, a_i] + \eta \cdot \Delta Q \cdot \alpha_i(s(t))$$

ε : Coeff exploitation
η : Taux d'apprentissage
α : Degré d'activation
γ : facteur de réduction

Synthèse des régulateurs RL flous

Optimisation Convergence vers un but (RL-FLC20)

Convergence vers but

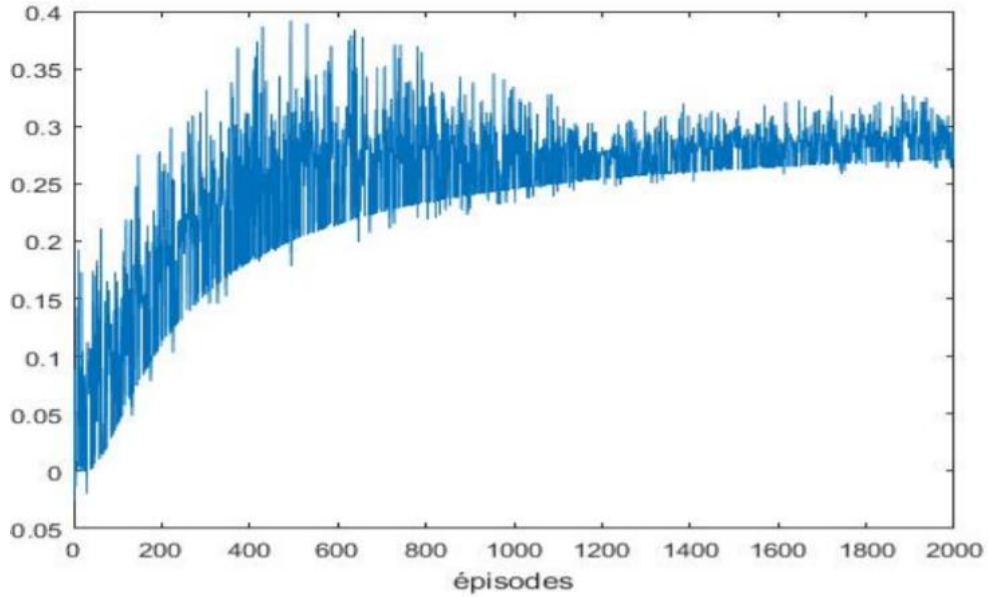
Si "d_{rg} est N et θ_{rg} est N" Alors $\omega_{Di} = \omega_{Di1}$ et $\omega_{Gi} = \omega_{Gi1}$ avec $Q_i = Q_{i11}$

ou $\omega_{Di} = \omega_{Di1}$ et $\omega_{Gi} = \omega_{Gi2}$ avec $Q_i = Q_{i12}$

$\vdots$ $\vdots$

ou $\omega_{Di} = \omega_{Di3}$ et $\omega_{Gi} = \omega_{Gi3}$ avec $Q_i = Q_{i33}$

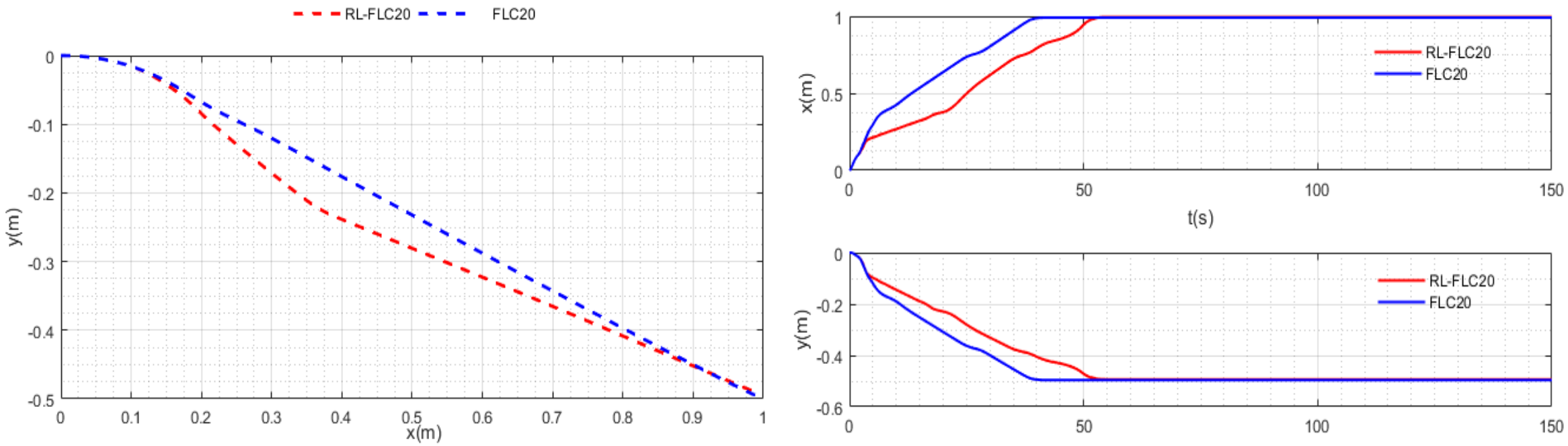
Comportement du robot	**Renforcement reçu**
Le robot s'approche du but	1
Le robot s'éloigne du but	−1
Le robot atteint le but	10



Renforcements reçus par épisode (Convergence vers un but)

Synthèse des régulateurs RL flous

Comparaison de convergence vers but (FLC20 et RL-FLC20)



Comparaison

Synthèse des régulateurs RL flous

Régulateur de d'évitement d'obstacle

Evitement d'obstacles

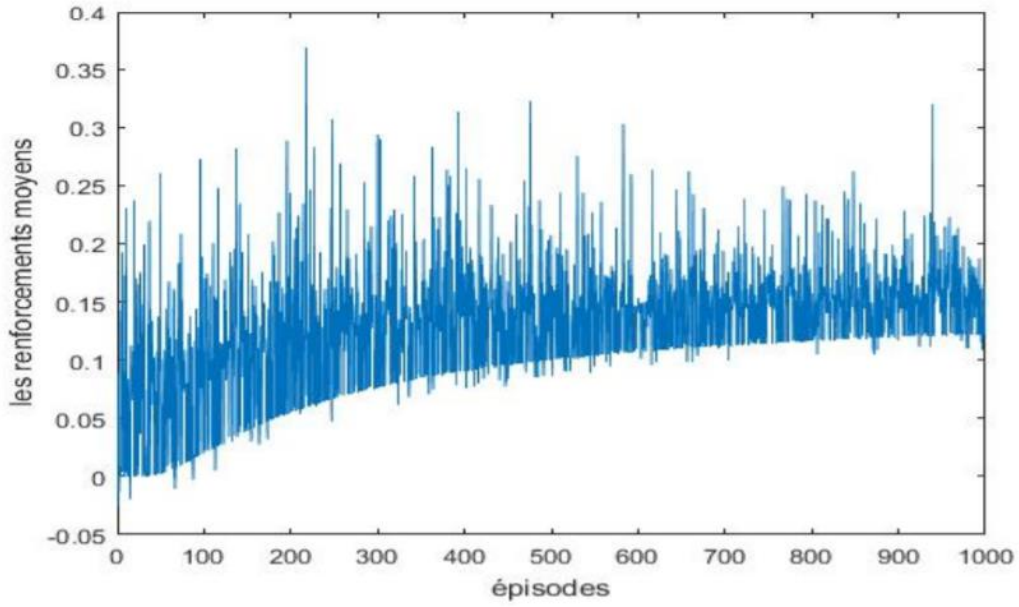
Si "o et O_i" Alors $\omega_{Di} = \omega_{Di1}$ et $\omega_{Gi} = \omega_{Gi1}$ avec $Q_i = Q_{i11}$

ou $\omega_{Di} = \omega_{Di1}$ et $\omega_{Gi} = \omega_{Gi2}$ avec $Q_i = Q_{i12}$

$\vdots$ $\vdots$

ou $\omega_{Di} = \omega_{Di4}$ et $\omega_{Gi} = \omega_{Gi4}$ avec $Q_i = Q_{i44}$

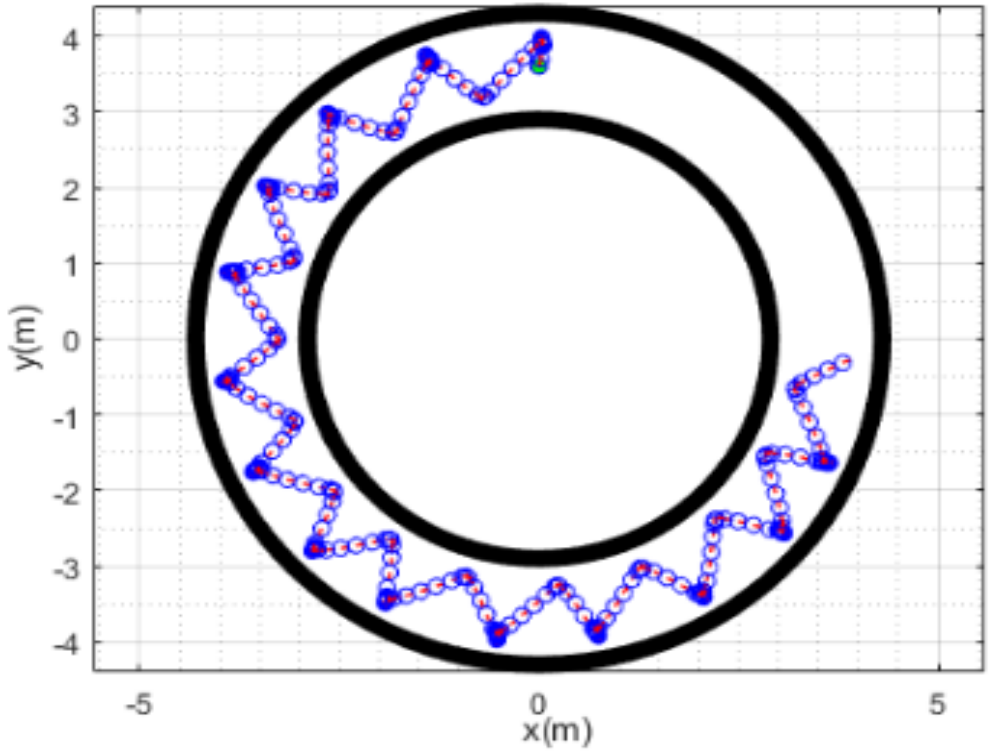
Comportement du robot	**Renforcement reçu**
Le robot s'éloigne de l'obstacle	1
Le robot s'approche de l'obstacle	-1
Le robot percute un obstacle	-10



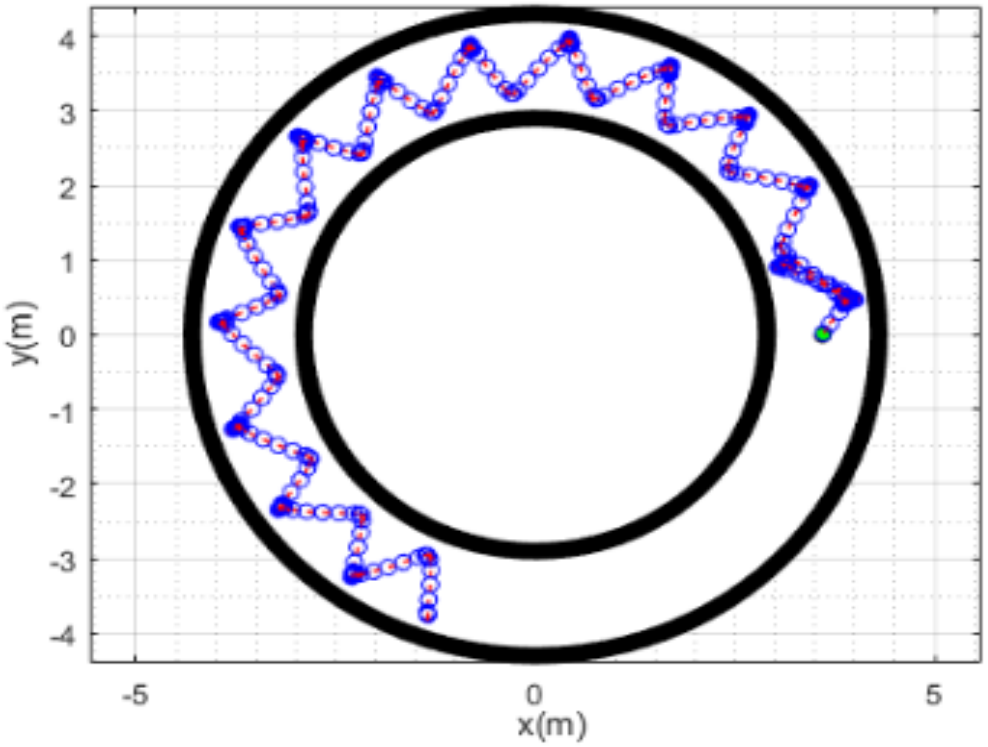
Renforcements reçus par épisode (évitement d'obstacles)

Synthèse des régulateurs renforcement-flous

Résultats de simulation et de comparaison (évitement)



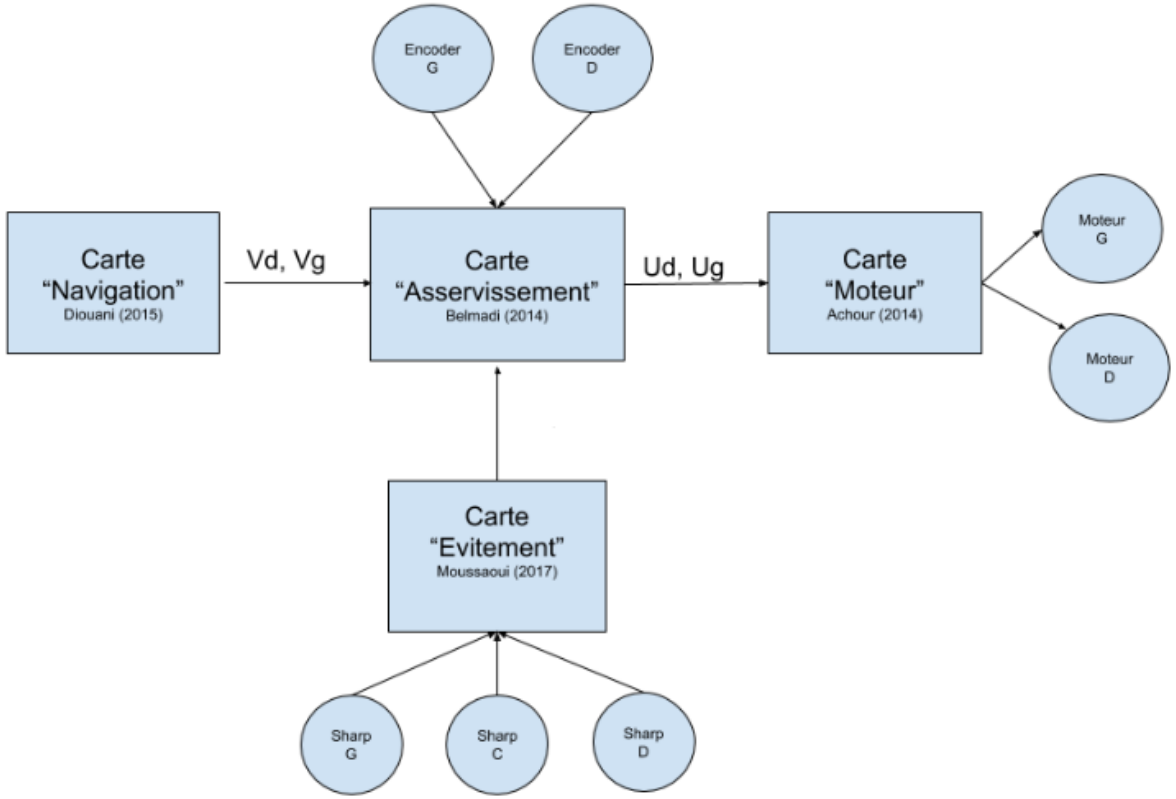
(a) Régulateur flou



(b) Régulateur renforcement-flou

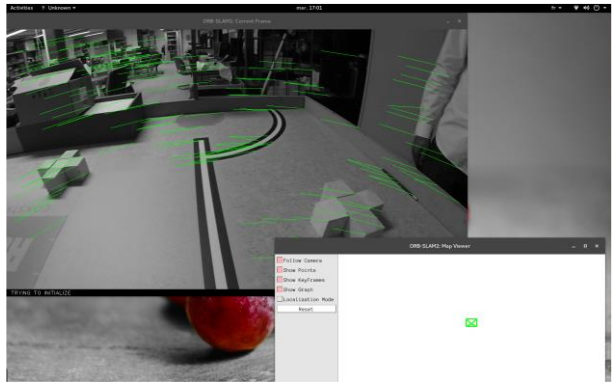
Matériel

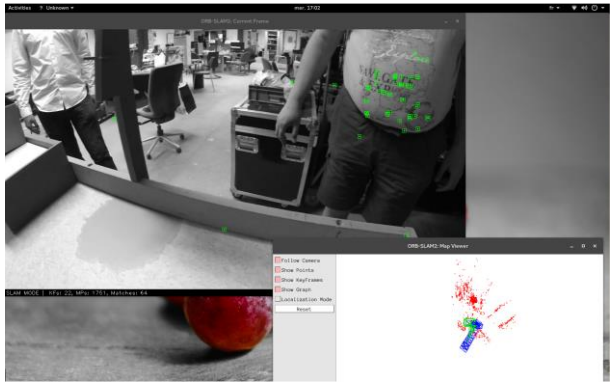

1080-60
00:01
2H00
HERO3+
GoPro

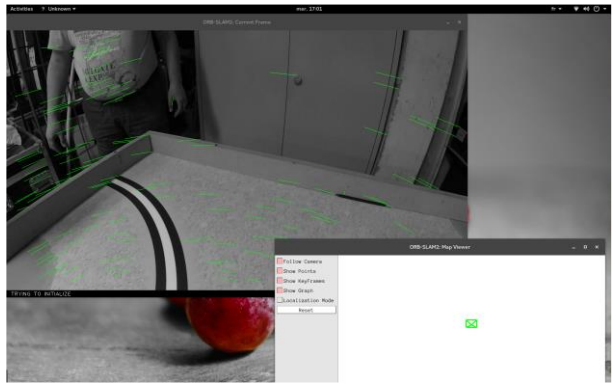

Encoder G
Encoder D
Carte "Navigation"
Diouani (2015)
Vd, Vg
Carte "Asservissement"
Belmadi (2014)
Ud, Ug
Carte "Moteur"
Achour (2014)
Moteur G
Moteur D
Carte "Evitement"
Moussaoui (2017)
Sharp G
Sharp C
Sharp D

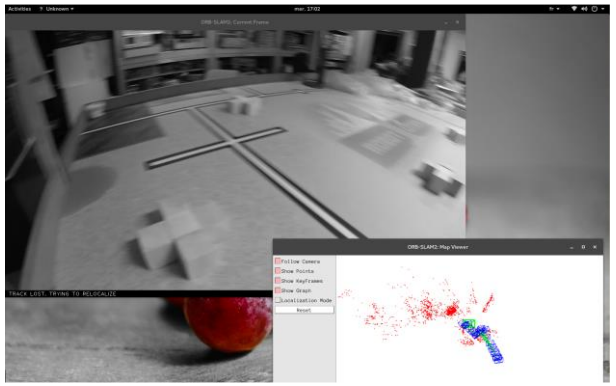
Localisation par V-SLAM

Test sur le MonoSLAM



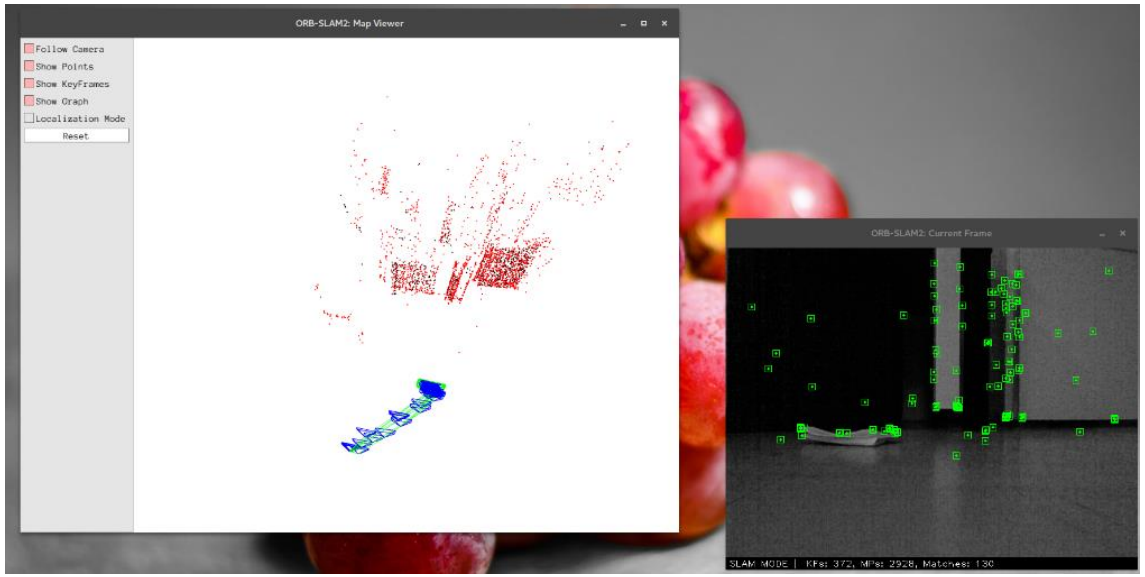






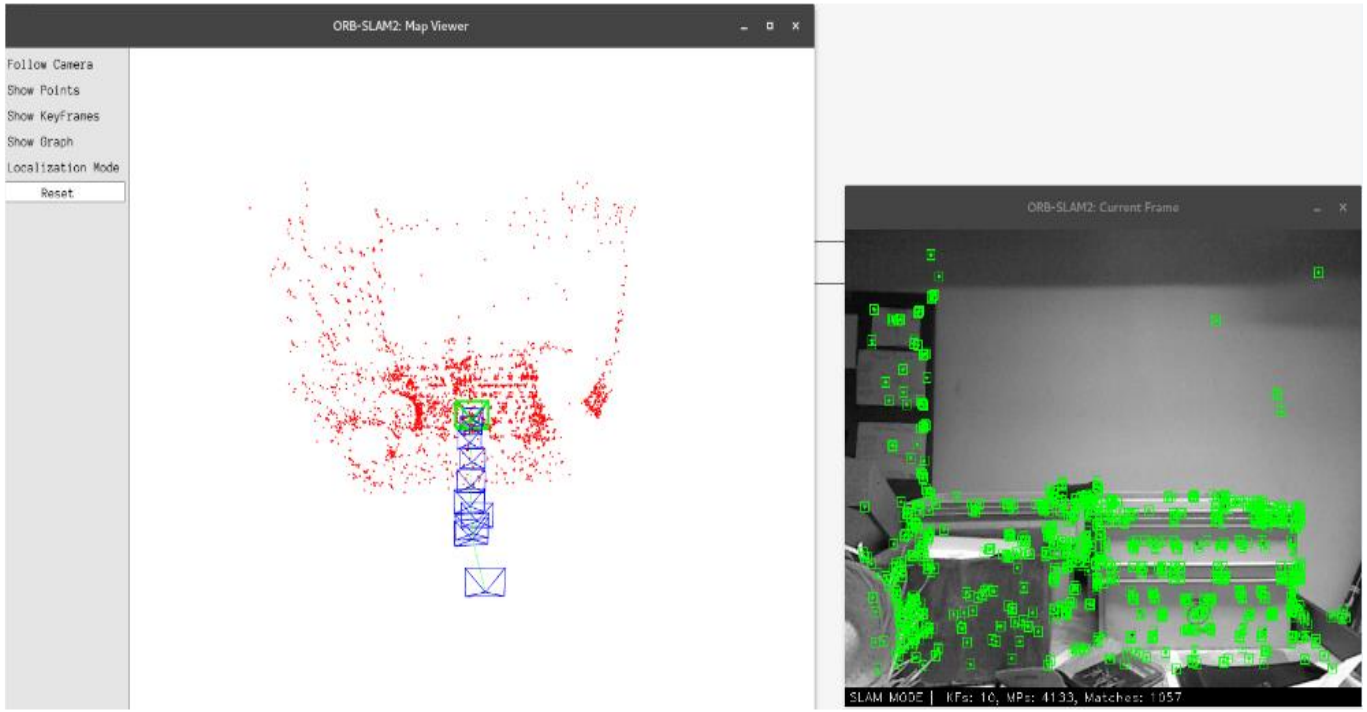
Localisation par V-SLAM:

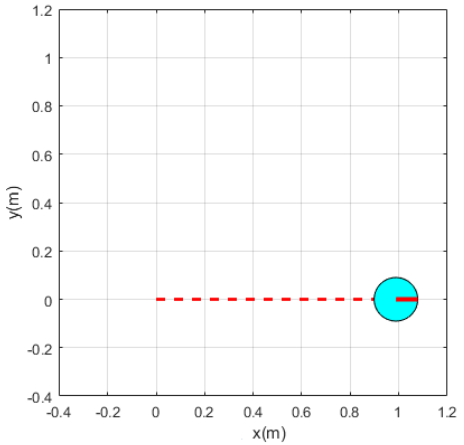
Test sur le RGB-D SLAM



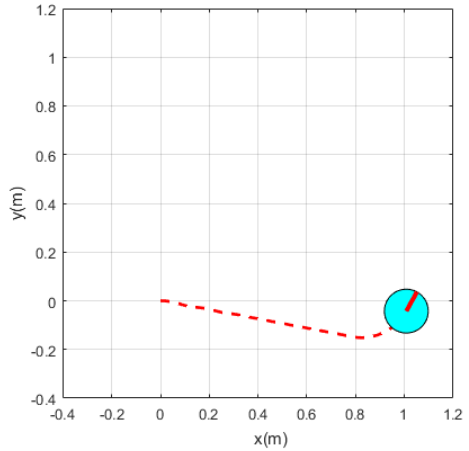
Valeurs réelles			RGB-D SLAM			Odométrie (Cyborg 2017)		
x	y	θ	x	y	θ	x	y	θ
0	0	0	−0.0370	0.0200	-1.7094^{-4}	0	0	0
40	40	0	39.9858	40.6389	−0.0975	40.53	42.87	0.01
0	0	0	−0.5027	−0.6823	−0.0111	−2.06	2.71	0.01
40	40	0	40.8477	40.5749	−0.0064	38	43.76	0
0	0	0	0.0313	−4.2988	−0.0264	−4.75	5.96	−0.01
40	40	0	40.5742	41.2686	−0.0954	37.01	46.09	−0.02

Navigation floue par V-SLAM





Navigation floue avec odométrie



Navigation floue avec V-SLAM

Navigation renforcement -floue par V-SLAM (vidéo)

Conclusion

- Navigation floue efficace.
- Navigation renforcement-floue par apprentissage.
- Efficacité de ORB-SLAM2 pour la localisation.
- Navigation floue par ORB-SLAM2 en mode SLAM.

Perspectives

- Extension aux environnements dynamiques.
- Hybridation de la logique floue et les réseaux de neurones.

Merci pour votre attention