

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET
POPULAIRE**
**Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique**



Ecole Nationale Polytechnique
Département d'Automatique



Mémoire en vue de l'obtention du complément Master

Thème :

**Etude comparative entre l'asservissement visuel 2D
basé sur le formalisme de fonction de tâche et
l'asservissement visuel 2D basé sur la commande
par logique floue**

Réalisé par
LOUNES Selman Ahmed Amin

Soutenu le 2 juillet 2015 devant le jury :

Pr BERKOUK
Pr CHKIREB
M. Omar STIHI

Président
Examineur
Rapporteur

Remerciements

Au terme de ce travail,

Nous adressons nos remerciements à l'ensemble des enseignants de l'École Nationale Polytechnique spécialement ceux du département du Génie Automatique, pour leur encadrement, appuie scientifique, disponibilité ainsi que pour tout le savoir qu'ils nous ont transmis durant tout au long de notre formation.

Nous remercions aussi Monsieur E.CHKIREB, enseignant à l'École Nationale Polytechnique, d'avoir accepté d'examiner et évaluer ce modeste travail.

Nous tenons également à remercier Monsieur A.BERKOUK, enseignant à l'École Nationale Polytechnique, de nous avoir fait l'honneur de présider ce jury.

Nous remercions également tous ceux qui nous ont soutenus et ont contribué de loin ou de près à la réalisation de ce travail.

TABLE DES MATIÈRES

Introduction	1
Application de la Logique Floue pour le suivie de ligne	2
Fuzzificationn	4
Énoncé des règles	6
Inférence	12
Degré d'activation	12
Agrégation	13
Defuzzification	13
L'asservissement visuel 2D basé sur le formalisme de fonction de tâche	14
Comparaison des résultats des deux méthodes	16
Conclusion	23
Bibliographie	24

Introduction :

Depuis la nuit des temps l'homme a toujours essayé de réduire ses efforts en exploitant autrui, les circonstances et les faits historiques ainsi que sa curiosité lui ont permis d'inventer, de développer et ainsi exploiter des machines.

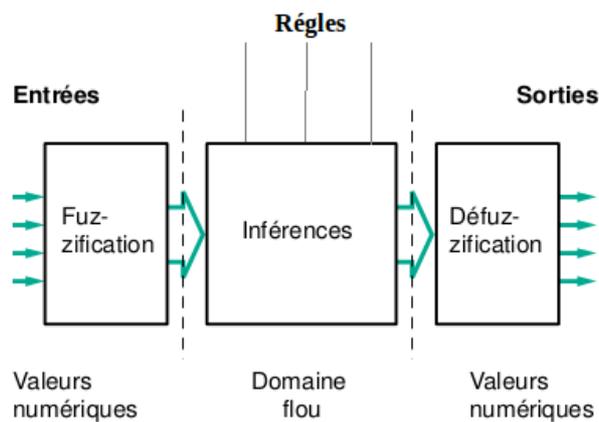
Dans un précédent travail [1] nous avons essayé de réduire les efforts tant mentales (décisions) que physiques (conduite) d'un conducteur, en développant une commande de suivie de trajectoire (marquage routier) détectée via une camera, pour cela nous nous sommes orientés à la commande par logique floue car elle nous permet d'implémenter le raisonnement et ainsi les bons réflexes qu'un être humain aurait eu lors d'une situation déterminée.

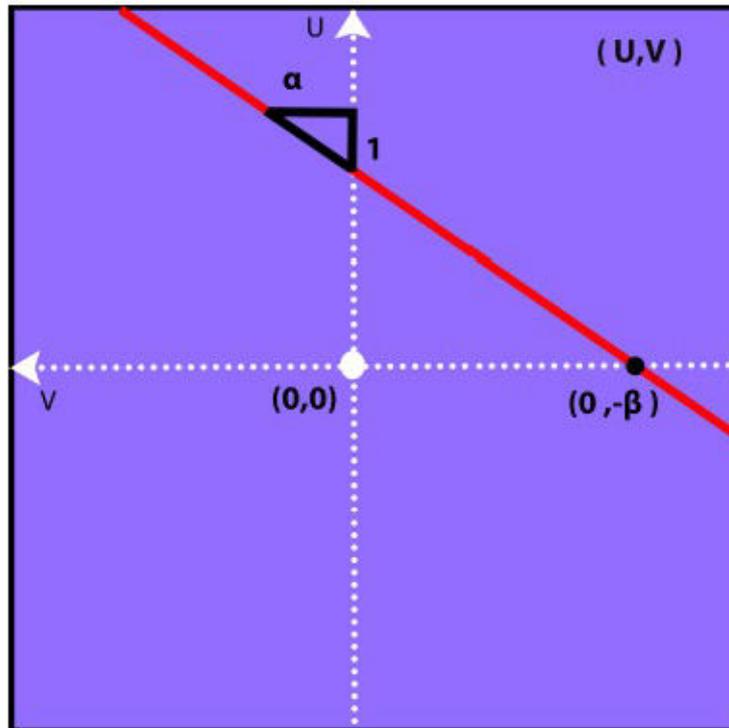
Dans ce travail nous allons essayer de faire une étude comparative entre les résultats que nous avons obtenu de l'asservissement visuel 2D basé sur la commande par logique floue et les résultats que deux anciens polytechniciens ont obtenu [2] de l'asservissement visuel 2D basé sur le formalisme de fonction de tâche réalisés sur la même plate-forme mobile « Robucar »

Application de la Logique Floue pour le suivie de ligne

L'être humain résout souvent des problèmes complexes à l'aide de données approximatives la précision des données est, donc, souvent inutile. Ce qui fait que plutôt que de modéliser le système, il est souvent intéressant de modéliser le comportement d'un opérateur humain face au système et plutôt que par des valeurs numériques précises, le fonctionnement doit être décrit par des qualificatifs globaux traduisant l'état approximatif des variables.

Un traitement ou se fait suivant le schéma suivant :





Extraction des caractéristiques de la droite détectée à partir de l'image

Beta représente la « distance » en Pixels entre l'origine et l'intersection de la droite détectée avec l'axe (OV)

Alpha représente la tangente de la droite détectée par rapport à l'axe (OU)

Les grandeurs numériques d'entrées que nous avons utilisées dans notre cas sont alpha, beta, dalpha et dbeta où :

alpha et beta nous permettent de savoir la position actuelle de la ligne détectée.

dalpha et dbeta sont respectivement les variations de alpha et de beta, elles nous permettent d'avoir une estimation sur la position future de la ligne détectée et de pouvoir ainsi anticiper et éviter qu'elle s'éloigne de la position désirée

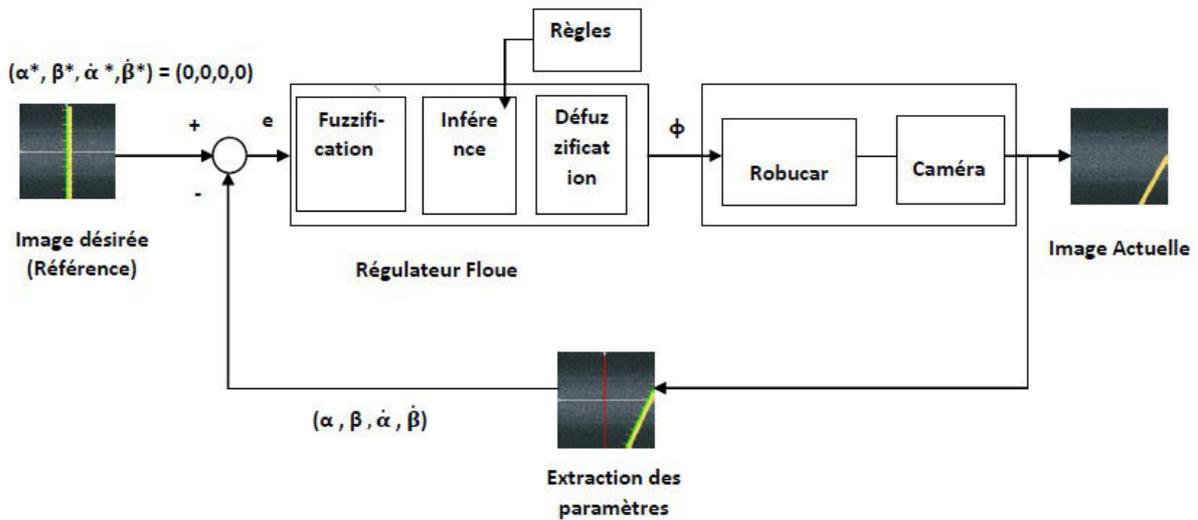
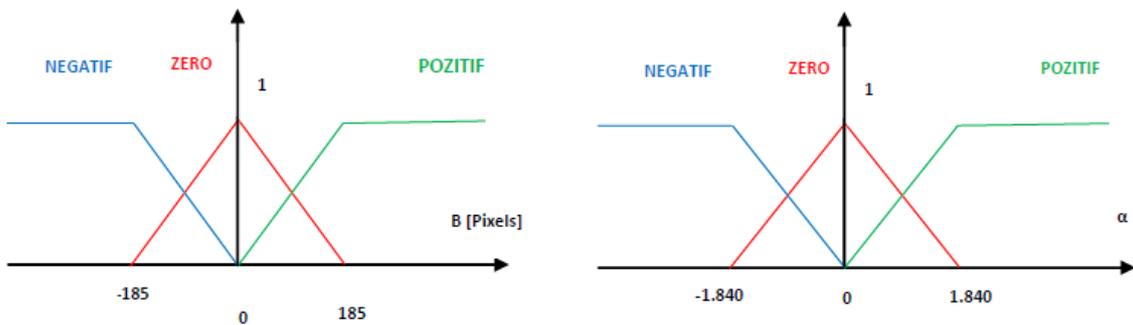


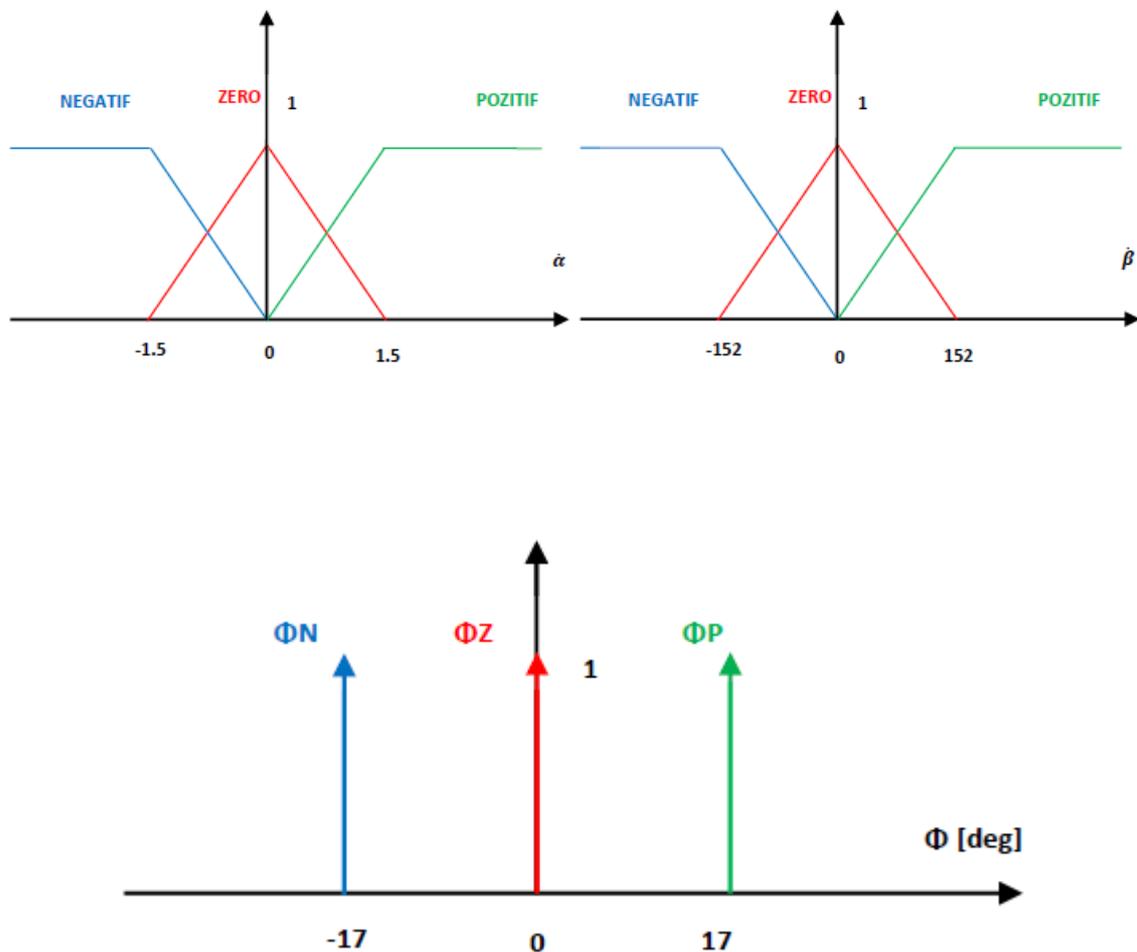
Schéma fonctionnel du suivi de ligne

Fuzzification

La fuzzification consiste à évaluer les fonctions d'appartenance qui vont nous permettre d'établir une relation entre la grandeur d'entrée et le degré de vérité de la variable floue correspondante ceci est illustré par les graphes des fonctions d'appartenances.

Les fonctions d'appartenances choisit pour le suivi de ligne sont les suivants :





Où $\phi = 17\text{deg}$ représente l'angle de braquage maximal.

Les fonctions d'appartenances ont été choisies triangulaires et symétriques pour simplifier le calcul de la commande et ainsi permettre de réduire le temps d'exécution du programme mais ceci a été fait sous les hypothèses suivantes :

- les quarts roues du robucar ont le même rayon (parfaitement identiques).
- la vitesse appliquée aux deux roues avant est exactement la même.
- l'angle maximum ainsi que la vitesse angulaire de braquage des deux roues avant sont exactement les mêmes.

Si ces hypothèses ne sont pas prises en considération ne devrions enlever la symétrie présente dans les fonctions d'appartenance et ceci pour ne pas favoriser une direction du robot par rapport à une autre.

Énoncé des règles :

Les systèmes à logique floue nécessitent une certaine expertise et connaissance en ce qui concerne le fonctionnement du système à commander exprimé sous forme d'une base de règles du type : Si ... (prémises) ... Alors ... (conclusion).

Une bonne énonciation des règles est une étape cruciale du bon fonctionnement de la commande par logique floue car cette étape constitue la partie « raisonnement » du robot, c'est pourquoi le choix des règles doit être minutieux.

Pour cela nous nous sommes basés sur les décisions prises par un "bon" conducteur de voiture qui essaye de rejoindre la trajectoire à suivre et de maintenir son véhicule sur cette dernière tout en nous inspirant des travaux menés au Département of Intelligent Mechanical System Japan [3]

Règles qui favorisent un angle de braquage positif (vers la Gauche) :

-RP1 Si (α est POSITIF et β est POSITIF) alors ϕ est POSITIF

En effet dans ce cas le conducteur doit braquer à gauche afin de rejoindre la ligne et d'éviter qu'elle soit perdue du champ visuel.

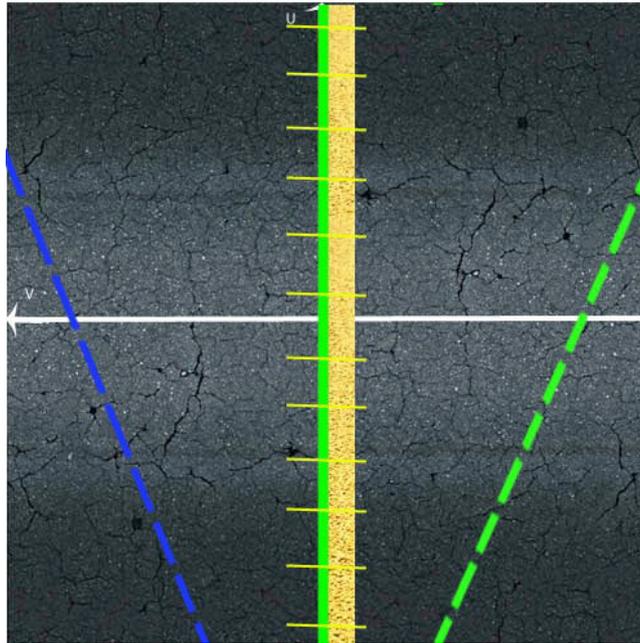
Même chose pour les deux règles qui suivent.

-RP2 Si (α est POSITIF et β est ZERO) alors ϕ est POSITIF

- RP3 Si (α est ZERO et β est POSITIF) alors ϕ est POSITIF

- RP4 Si (α est ZERO et β est ZERO et $\delta\alpha$ est POSITIF et $\delta\beta$ est POSITIF) alors ϕ est POSITIF

Dans ce cas malgré que la ligne détectée soit pratiquement confondue avec la ligne désirée, on doit prendre en considération les variations $\delta\alpha$ et $\delta\beta$ qui nous permettent d'avoir une estimation sur l'emplacement futur de la trajectoire détectée (représenté dans la figure ci-dessous), qui s'éloigne de la position désirée, et de pouvoir ainsi anticiper et appliquer une commande (braquer à gauche) afin d'éviter cet emplacement futur de la ligne.



Remarque :

- ————— : Position désiré de la ligne détectée (consigne).
- ————— : Position actuelle de la ligne détectée (à l'instant t).
- - - - - - : Position de la ligne détectée a l'instant t-1.
- - - - - - : Position estimée de la ligne a l'instant t+1.

Même chose pour les deux règles qui suivent.

-RP5 Si (α est ZERO et β est ZERO et $d\alpha$ est POSITIF et $d\beta$ est ZERO)
alors ϕ est POSITIF

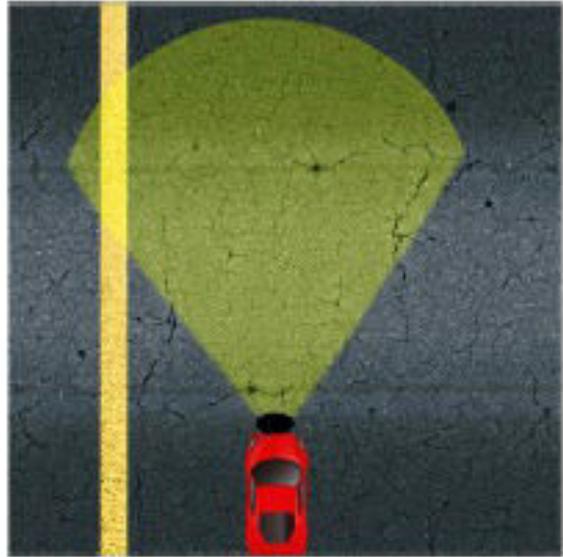
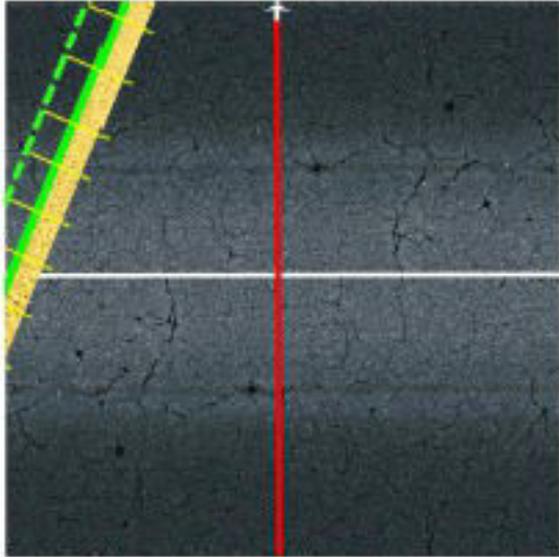
-RP6 Si (α est ZERO et β est ZERO et $d\alpha$ est ZERO et $d\beta$ est POSITIF)
alors ϕ est POSITIF

-RP7 Si (α est NEGATIF et β est POSITIF et $d\alpha$ est ZERO et $d\beta$ est ZERO)
alors ϕ est POSITIF

Cette règle a été ajoutée pour remédier au problème représenté dans la figure ci-dessous dû à la déformation de la camera, pour savoir qu'il s'agit bien d'une ligne parallèle au robucar, on vérifie (via $d\alpha$ et $d\beta$) si il n'y pas de variation de α

8

et de β malgré l'avancement du robot si c'est le cas le conducteur doit braquer à gauche afin de rejoindre la ligne.



Règles qui favorisent un angle de braquage négatif (vers la droite)

-RN1 Si (α est NÉGATIF et β est NÉGATIF) alors ϕ est NÉGATIF

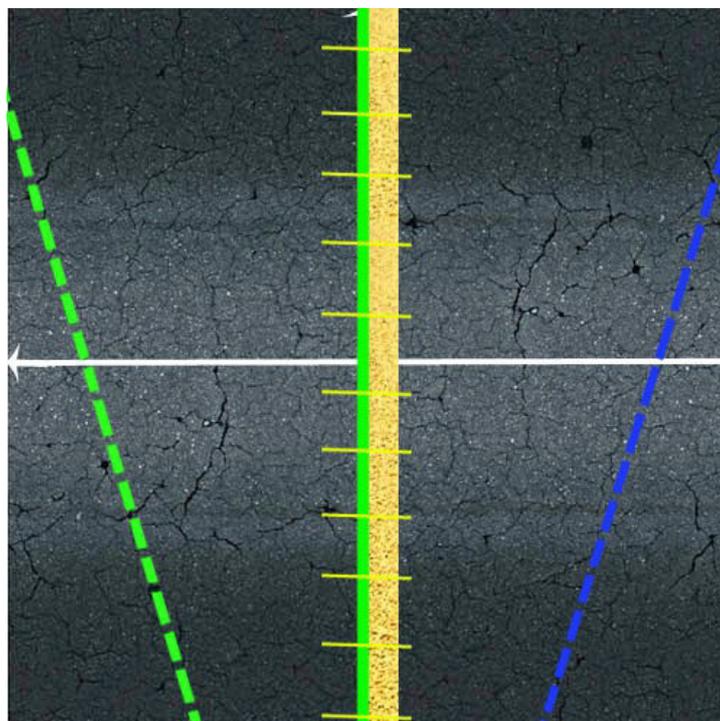
En effet dans ce cas le conducteur doit braquer à droite afin de rejoindre la ligne et d'éviter qu'elle soit perdue du champ visuel.

Même chose pour les deux règles qui suivent

-RN3 Si (α est ZERO et β est NEGATIF) alors ϕ est NEGATIF.

- RN4 Si (α est ZERO et β est ZERO et $d\alpha$ est NEGATIF et $d\beta$ est NEGATIF) alors ϕ est NEGATIF

Dans ce cas malgré que la ligne détectée soit pratiquement confondue avec la ligne désirée, on doit prendre en considération les variations $d\alpha$ et $d\beta$ qui nous permettent d'avoir une estimation sur l'emplacement futur de la trajectoire détectée (représenté dans la figure ci-dessous), qui s'éloigne de la ligne désiré, et de pouvoir ainsi anticiper et appliquer une commande (braquer à droite) afin d'éviter cette emplacement futur de la ligne.

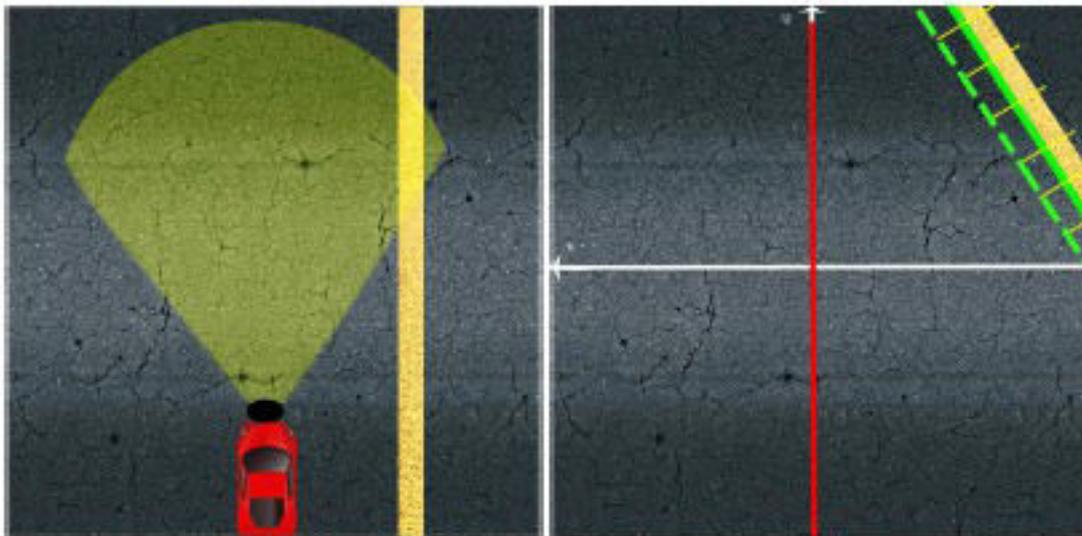


Même chose pour les deux règles qui suivent

-RN6 Si (α est ZERO et β est ZERO et $d\alpha$ est ZERO et $d\beta$ est NEGATIF) alors ϕ est NEGATIF

- RN7 Si (α est POSITIF et β est NEGATIF et $d\alpha$ est ZERO et $d\beta$ est ZERO) alors ϕ est NEGATIF

Cette règle a été ajoutée pour remédier au problème représenté dans la figure ci-dessous dû à la déformation de la camera, pour savoir qu'il s'agit bien d'une ligne parallèle au robucar, on vérifie (via $d\alpha$ et $d\beta$) si il n'y pas de variation de α et de β malgré l'avancement du robot si c'est le cas le conducteur doit braquer à droite afin de rejoindre la ligne.



Règles qui favorisent un braquage nul (avancer tout droit) :

-RZ1 Si (α est ZERO et β est ZERO et $d\alpha$ est NEGATIF et $d\beta$ est POSITIF) alors ϕ est ZERO

-RZ2 Si (α est ZERO et β est ZERO et $d\alpha$ est ZERO et $d\beta$ est ZERO) alors ϕ est ZERO

-RZ3 Si (α est ZERO et β est ZERO et $d\alpha$ est POSITIF et $d\beta$ est NEGATIF) alors ϕ est ZERO

- RZ4 Si (α est POSITIF et β est NEGATIF et $d\beta$ est POSITIF) alors ϕ est ZERO

- RZ5 Si (α est POSITIF et β est NEGATIF et $d\alpha$ est POSITIF) alors ϕ est ZERO

RZ4 et RZ5 nous permettent de savoir que le marquage routier est bien incliné par rapport au robucar et qu'il ne s'agit pas de la déformation de la caméra, ce qui nous permet de prendre la décision de continuer tout droit pour rejoindre la trajectoire.

Contrairement à la règle RN7 qui nous permet de savoir que le marquage routier est en fait parallèle au robucar et qu'il s'agit de la déformation de la caméra ce qui nous permet de prendre la décision de braquer à gauche pour rejoindre la trajectoire.

-RZ6 Si (α est NEGATIF et β est POSITIF et $\delta\beta$ est NEGATIF) alors ϕ est ZERO

- RZ7 Si (α est NEGATIF et β est POSITIF et $\delta\alpha$ est NEGATIF) alors ϕ est ZERO.

RZ6 et RZ7 nous permettent de savoir que le marquage routier est bien incliné par rapport au robucar et qu'il ne s'agit pas de la déformation de la caméra ce qui nous permet de prendre la décision de continuer tout droit pour rejoindre la trajectoire.

Contrairement à la règle RP7 qui nous permet de savoir que le marquage routier est en fait parallèle au robucar et qu'il s'agit de la déformation de la caméra ce qui nous permet de prendre la décision de braquer à droite pour rejoindre la trajectoire.

On précise que dans la commande par logique floue toutes les règles sont appliquées en même temps, cependant certaines règles seront plus représentatives de l'état actuel du robot par rapport à la trajectoire, elles auront donc un degré d'influence plus important sur la commande.

Le tableau suivant résume les règles citées précédemment :

n	Règles	α	β	$\hat{\alpha}$	$\hat{\beta}$	ϕ
1	RP1	P	P	-	-	P
2	RP2	P	Z	-	-	P
3	RP3	Z	P	-	-	P
4	RP4	Z	Z	P	P	P
5	RP5	Z	Z	P	Z	P
6	RP6	Z	Z	Z	P	P
7	RP7	N	P	Z	Z	P
8	RN1	N	N	-	-	N
9	RN2	N	Z	-	-	N
10	RN3	Z	N	-	-	N
11	RN4	Z	Z	N	N	N
12	RN5	Z	Z	N	Z	N
13	RN6	Z	Z	Z	N	N

7	RP7	N	P	Z	Z	P
8	RN1	N	N	-	-	N
9	RN2	N	Z	-	-	N
10	RN3	Z	N	-	-	N
11	RN4	Z	Z	N	N	N
12	RN5	Z	Z	N	Z	N
13	RN6	Z	Z	Z	N	N
14	RN7	P	N	Z	Z	N
15	RZ1	Z	Z	N	P	Z
16	RZ2	Z	Z	Z	Z	Z
17	RZ3	Z	Z	P	N	Z
18	RZ4	P	N	-	P	Z
19	RZ5	P	N	P	-	Z
20	RZ6	N	P	-	N	Z
21	RZ7	N	P	N	-	Z

Inférence :

Le mécanisme d'inférence le plus couramment utilisé, et que nous allons utiliser tout au long de notre travail, est celui dit « de Mamdani ». Il représente une simplification du mécanisme plus général basé sur « l'implication floue » et le « modus ponens généralisé »

Degré d'activation :

Le degré d'activation d'une règle est l'évaluation du prédicat de chaque règle par combinaison logique des propositions du prédicat. Le « ET » est réalisé en effectuant le minimum entre les degrés de vérité des propositions

En effet une conclusion ne peut pas être « plus vraie » que les conditions qui la génèrent ceci peut être expliqué à travers cet exemple :

La puissance d'une chaîne, constituée par plusieurs maillons, est définie par la puissance du maillon le plus faible.

Agrégation :

L'ensemble flou global de sortie est construit par agrégation des ensembles flous obtenus par chacune des règles concernant cette même sortie. On considère que les règles sont liées par un « OU » logique et on calcule donc le maximum entre les fonctions d'appartenance résultantes pour chaque règle ce qui correspond à la règle la plus représentative de l'état actuel c'est à dire qu'on calcule le maximum:

- des degrés d'activation des règles POSITIF (RP) qui sera noté $M\Phi_P$.
- des degrés d'activation des règles NEGATIF (RN) qui sera noté $M\Phi_N$.
- des degrés d'activation des règles ZERO (RZ) qui sera noté $M\Phi_Z$.

Defuzzification :

A la fin de l'inférence, le braquage (commande floue) est déterminé mais il n'est pas directement utilisable pour commander le braquage des roues. Il est nécessaire de passer du « monde flou » au « monde réel », c'est la défuzzification. Il existe plusieurs méthodes, la plus souvent utilisée est celle du calcul du « centre de gravité » de l'ensemble flou, cependant pour simplifier les calculs nous utilisons une autre méthode qui est aussi très pratique, cette méthode s'intitule Moyenne pondérée.

$$\Phi = \frac{-\Phi_{max}M\Phi_N + 0 \times M\Phi_Z + \Phi_{max}M\Phi_P}{M\Phi_N + M\Phi_Z + M\Phi_P}$$

L'asservissement visuel 2D basé sur le formalisme de fonction de tâche :

Notre but est de synthétiser une loi de commande permettant la réalisation d'une tâche de navigation référencée vision. Pour ce faire, nous exploitons le formalisme des fonctions de tâches que nous venons brièvement de présenter. Classiquement, un asservissement visuel permettant d'amener $s(q)$ vers le motif désiré s^* est défini par la fonction de tâche eAV suivante [CHAUMETTE, 1990] :

$$e_{AV} = \mathcal{C}(s(q) - s^*)$$

La matrice d'interaction se détermine selon le type de l'information visuelle choisit, des méthodes sont proposées pour calcul analytique de la matrice d'interaction pour différentes primitives géométriques simples (des points, des lignes, sphères, ellipses...).

Dans le cas d'une primitive de type droite la matrice d'interaction est donnée sous la forme suivante

$$\begin{bmatrix} -\lambda_\theta \sin\theta & \lambda_\theta \cos\theta & -\lambda_\theta \rho & \rho \sin\theta & -\rho \cos\theta & -1 \\ -\lambda_\rho \sin\theta & \lambda_\rho \cos\theta & \lambda_\rho \rho & (1 + \rho^2) \cos\theta & (1 + \rho^2) \sin\theta & 0 \end{bmatrix}$$

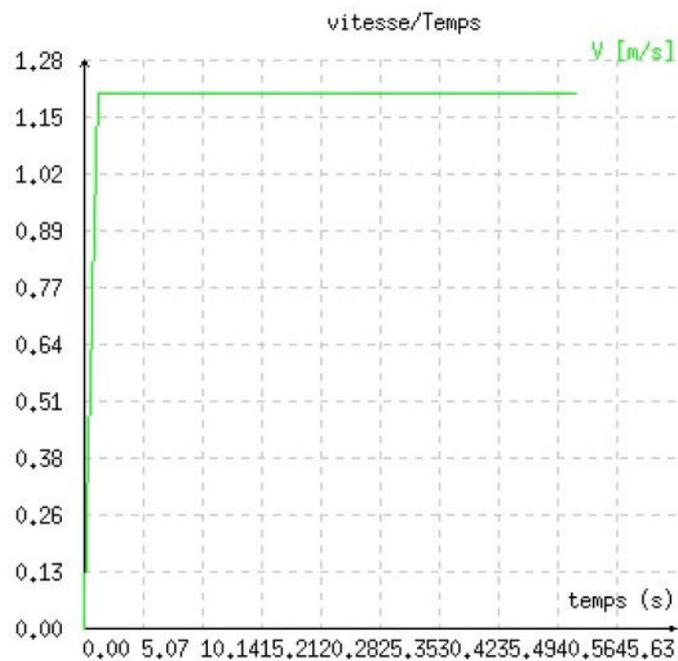
Avec :

$$\begin{cases} \lambda_\theta & = \frac{1}{d_i} \cdot (a_i \cdot \cos\theta + b_i \cdot \sin\theta) \\ \lambda_\rho & = \frac{1}{d_i} \cdot (a_i \cdot \rho \cdot \cos\theta + b_i \cdot \rho \cdot \sin\theta + c_i) \end{cases}$$

Le calcul de la matrice d'interaction $L(s,z)$ nécessite généralement une mesure ou un modèle de la profondeur z . Toutefois, cette information n'étant pas toujours disponible, il est nécessaire, dans ce cas, d'utiliser un modèle ou une approximation de la matrice d'interaction, notée $\hat{L}[\cdot]$ où cette dernière représente la pseudo inverse de la matrice d'interaction. Cette approximation pourrait en effet fausser le calcul de la commande, entraîner des oscillations et réduire la vitesse optimale du suivie.

Comparaison des résultats des deux méthodes :

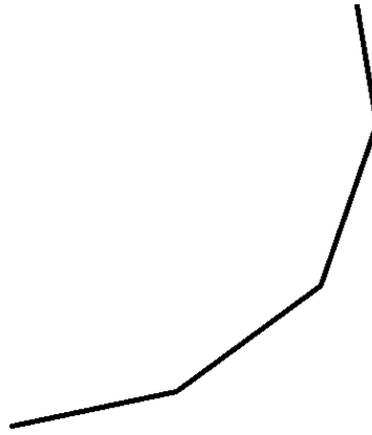
Résultat de l'asservissement visuel 2D basé sur la commande par logique floue :
Dans cet essai pratique nous avons utilisé une vitesse fixe de 1.2 m/s



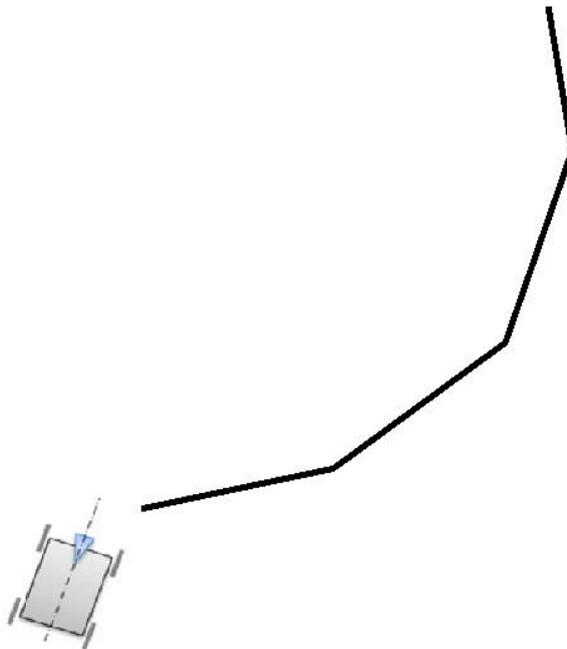
Remarque :

Nous avons augmenté la vitesse de 0 m/s à 1.2 m/s graduellement pour éviter le démarrage brusque du robot

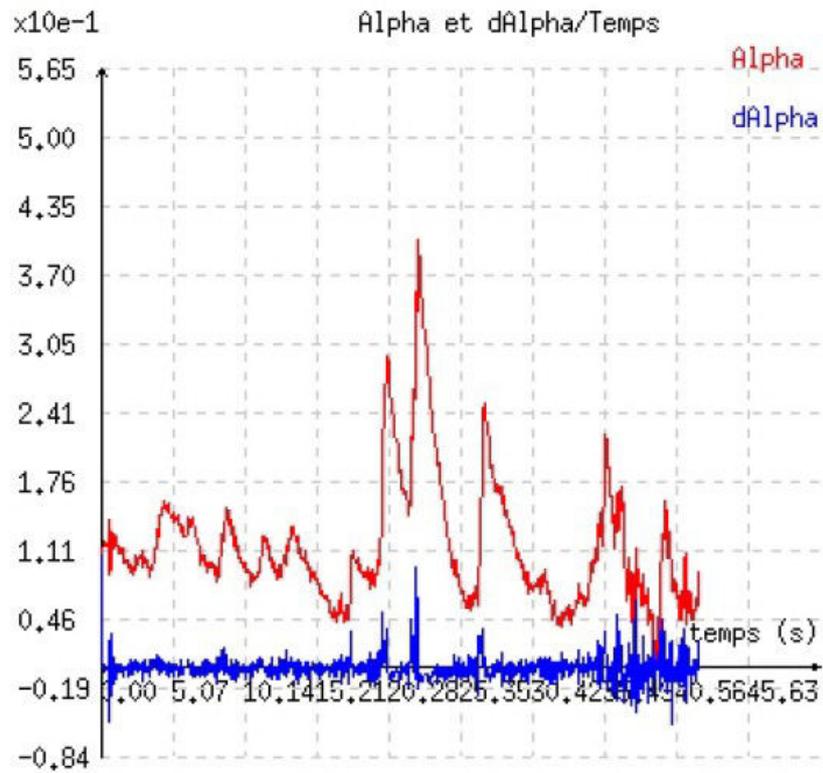
La trajectoire à suivre a la forme suivante :



Positionnement initiale du robot par rapport à la trajectoire :

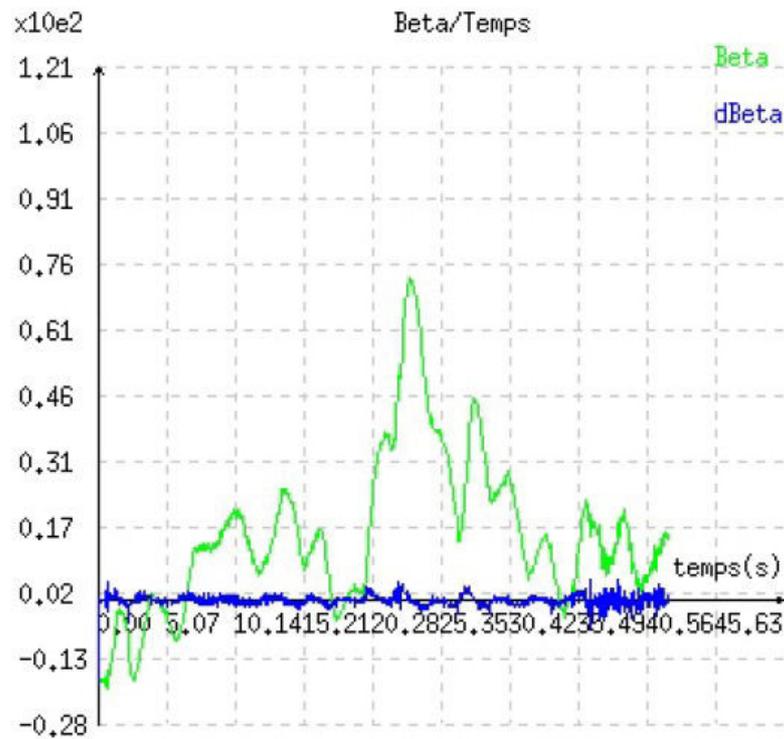


Variation de α et $d\alpha$ (et ainsi leurs erreurs par rapport à leur références car $\alpha_{Ref}=d\alpha_{Ref}=0$) en fonction du temps :



Alpha est de l'ordre de 10^{-1}

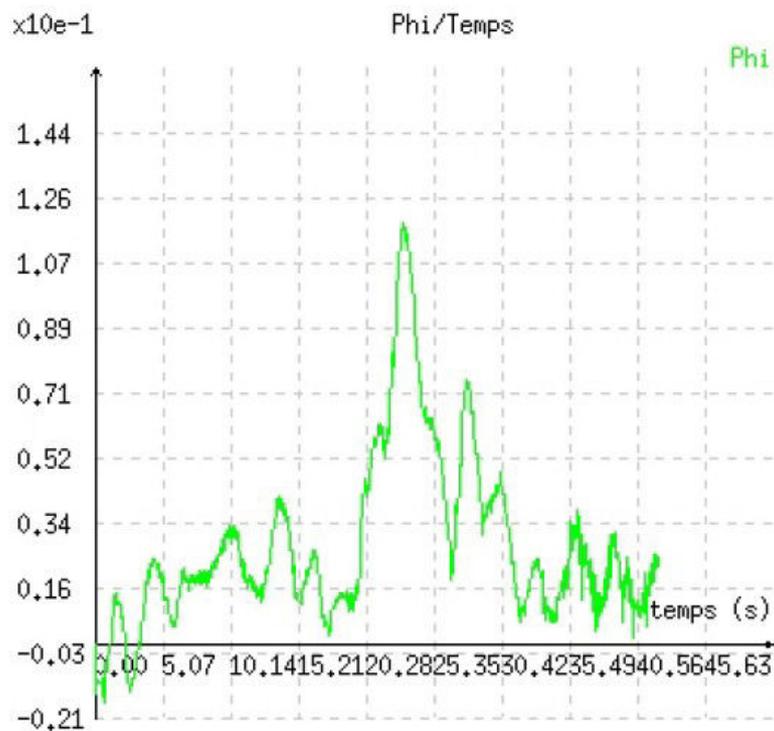
Variation de beta et dbeta (et ainsi leurs erreurs par rapport à leur références car $\text{betaRef}=\text{dbetaRef}=0$) en fonction du temps :



Rappelons que Beta représente la « distance » en Pixels entre l'origine et l'intersection de la droite détectée avec l'axe (OV) et Alpha représente la tangente de la droite détectée par rapport à l'axe (OU)

Les pics présents dans les graphes sont en grande partie dus à la variation brusque de l'orientation de la trajectoire.

La commande de braquage :



Φ est de l'ordre de 10^{-1} rad

On remarque que la commande prend des valeurs négatives uniquement lors du positionnement initial du robot sur la trajectoire, la commande ne prend par la suite que des valeurs positive ceci est expliqué par la forme de la trajectoire et démontre qu'il n'y a pratiquement aucune oscillation autour de la trajectoire ce qui permet une certaine conformabilité et une préservation de l'actionneur (vérin) de braquage

Le pic présent dans le graphe est en grande partie dû à la variation brusque de l'orientation de la trajectoire.

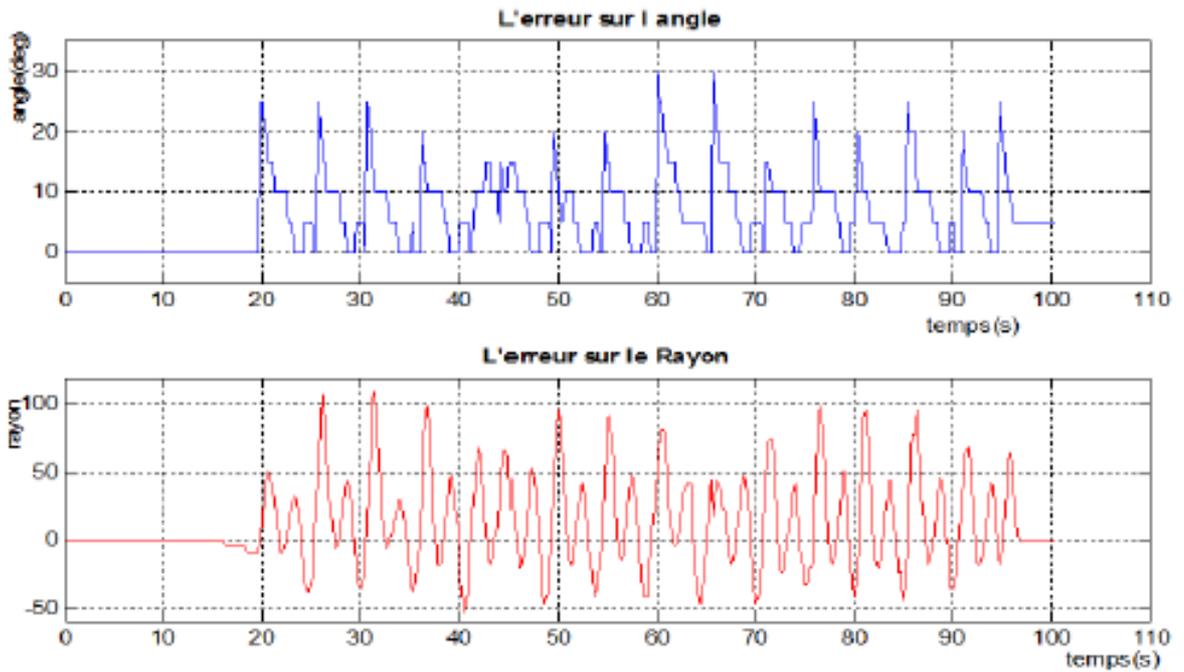
Rappelons que la synthèse de la commande a été faite sous les hypothèses suivantes :

- _ Les roues du robot sont parfaitement commandables en vitesse et en braquage.
- _ Les roues du robucar ont le même rayon (parfaitement identiques).
- _ La vitesse appliquée aux deux roues avant est exactement la même.
- _ L'angle maximum ainsi que la vitesse angulaire de braquage des deux roues avant sont exactement les mêmes.

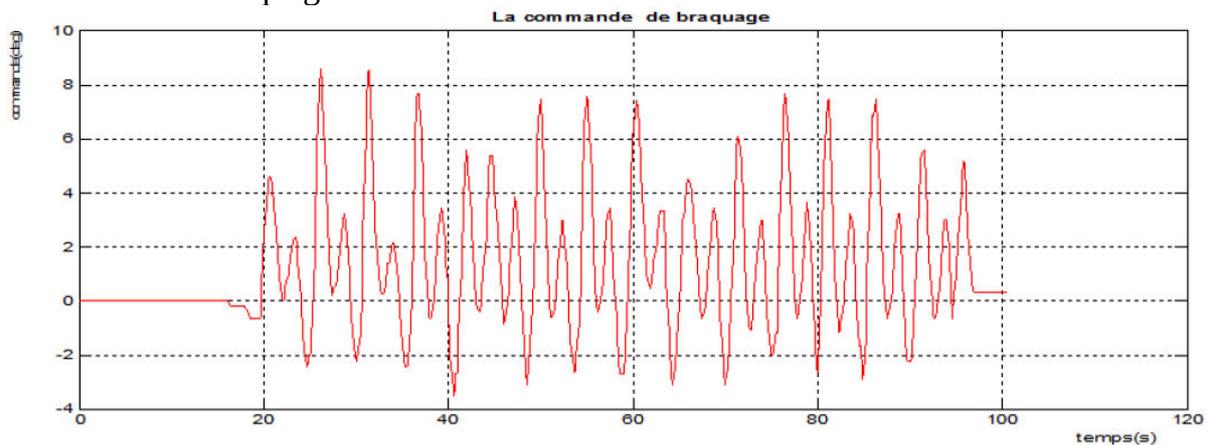
On en conclue que les résultats obtenus nous permettent d'augmenter la vitesse de suivie et atteindre la vitesse maximale du robot (5m/s), ils démontrent encore une fois la robustesse de la commande par logique floue.

Résultat de l'asservissement visuel 2D basé sur le formalisme de fonction de tâche

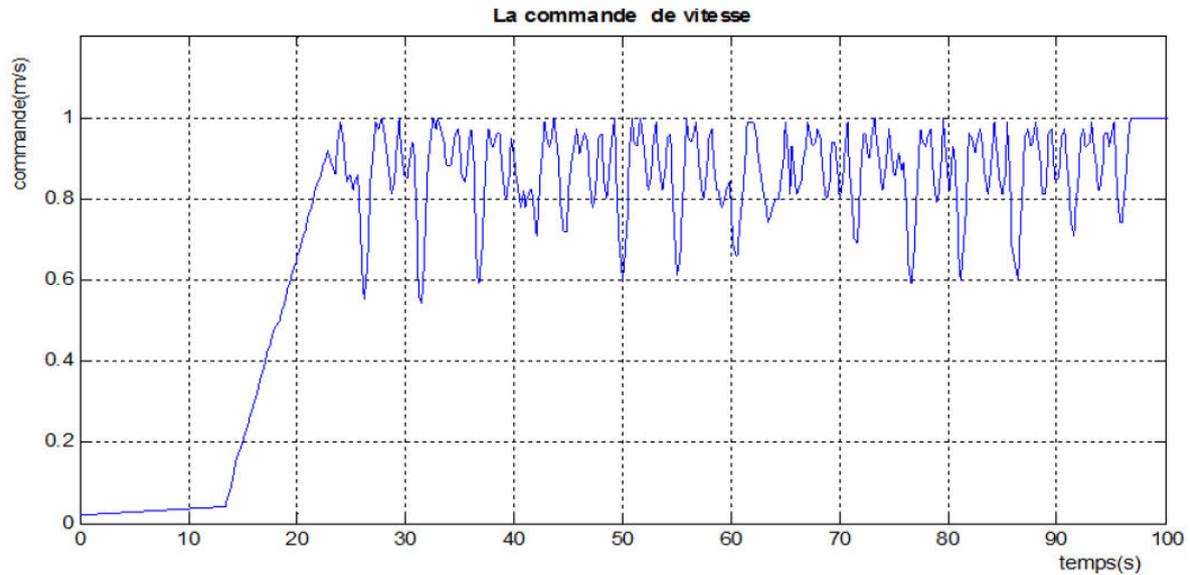
Dans ce cas une régulation de la vitesse a été nécessaire pour éviter la perte de la droite dans l'image pour des trajectoires avec une forte courbure.



Commande de braquage :



Commande de la vitesse :



On remarque que malgré la commande de vitesse qui fait varier la vitesse entre 0.57m/s et 1 m/s les résultats ci-dessus montre de fortes oscillations durant tout le trajet du robot se qui conduit à l'endommagement des actionneurs (vérin) de braquages.

Ceci-est dû principalement à l'approximation de la matrice d'interaction a cause de la non-disponibilité du model de profondeur.

Conclusion générale :

Dans ce travail nous avons présenté la synthèse d'une commande par logique floue pour garantir le suivi de trajectoire

Nous avons fait une étude comparative entre l'asservissement visuel 2D basé sur le formalisme de fonction de tâche et l'asservissement visuel 2D basé sur la commande par logique floue.

Les résultats nous ont démontrés encore une fois l'utilité, l'efficacité, et la robustesse de la commande par logique floue.

Bibliographie :

[1] Selman Ahmed Amin LOUNES et Abderraouf BOUSRI Navigation référencée vision et contrôle d'un Robot mobile de type voiture, Ecole Nationale Polytechnique, June 2015.

[2] AMMOUR Rabah et BOUSSIF Abderraouf. COMMANDE REFERENCEE VISION DU ROBOT MOBILE ROBUCAR. PhD thesis, Ecole Nationale Polytechnique (ENP), Juin 2012.

[3] SICK AG. LMS5xx Laser Measurement Sensors. Sick Sensor Intelligence, Erwin-Sick-Str. 1 79183 Waldkirch Germany, fevrier 2015.

[4] François Chaumette. De la perception a l'action : l'asservissement visuel, de l'action a la perception : la vision active. PhD thesis, Universite de Reennes 1 Institut de formation Superieure en Informatique et en Communication, Janvier 1998.

[5] F. Chevré and F. Guey. La logique 30, Mars 1998.

[6] Bertrand Florat. Chapitre 1 l'orienté objet. <http://www.florat.net/tutorial-java/chapitre01.html>, Consulté en Mai 2015.

[7] David FOLIO. Stratégies De Commande Référencées Multi-Capteurs Et Gestion De La Perte Du Signal Visuel Pour La Navigation D'un Robot Mobile. PhD thesis, Université Paul SABATIER de Toulouse III, Sept 2007.

[8] Open Source Robotics Foundation. Documentation ros. <http://wiki.ros.org/fr>, Consulté en Mai 2015.

[9] Shoichi Maeyama Keigo Watanabe et Tatsuya Kato. Image-based fuzzy trajectory tracking control for four-wheel steered mobile robots. Proc. of the 37th Annual Conference of the IEEE Industrial Electronics Society, Mars 2012.

[10] Shoichi Maeyama Keigo Watanabe et Tatsuya Kato. Obstacle avoidance for mobile robots using an image-based fuzzy controller. Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE, 69 :6392 { 6397, March 2013.

[11] Aaron Martinez and Enrique Fernandez. Learning ROS for Robotics Programming A practical, instructive, and comprehensive guide to introduce yourself to ROS, the top-notch, leading robotics framework. PACKT, Birmingham B3 2PB, UK, 2013.
76 Références Bibliographiques 77

[12] Visual Servoing Platform VISP. Biblioth#eque multi plate-forme modulaire pour l'asservissement visuel. <http://www.irisa.fr/lagadic/visp/visp.html>, Consulté en Janvier 2015.