

Remerciements

Je tiens à exprimer mes plus vifs et sincères remerciements à Roger Pissard-Gibollet ingénieur de recherche à l'INRIA Rhône-Alpes pour m'avoir accueilli au sein du projet Cycab et qui m'a conseillé, aidé et soutenu tout au long de mon stage.

Je souhaite également remercier Philippe Garnier post-doctorant pour sa collaboration dans mon travail et pour les précieuses réponses qu'il a donné à mes questions.

Je remercie aussi les autres membres du service moyen robotique pour leur gentillesse et pour l'ambiance sympathique et chaleureuse dans laquelle j'ai travaillé.

Sommaire

Introduction.....	3
I Description des modèles et des algorithmes	4
I.1 Modélisation du véhicule électrique	4
I.2 Le suivi de trajectoire.....	8
I.2.1 Le générateur de trajectoire.....	8
I.2.2 Algorithme de KANAYAMA	8
I.3 Evitement d'obstacle	10
I.4 Le suivi de véhicule.....	13
I.5 Le parking automatique par créneau	14
II Programmation logicielle	18
II.1 Méthodologie générale de programmation.....	18
II.2 La ressource physique	21
II.3 Les tâches robotiques.....	23
III Tests et résultats.....	25
III.1 Tests et résultats avec la simulation	25
III.1.1 Le suivi de trajectoire	25
III.1.2 Evitement d'obstacle.....	26
III.1.3 Le suivi de véhicule.....	27
III.1.4 Le parking par créneau.....	29
IV Partie économique.....	31
IV.1 Raisons du choix du sujet fait par l'INRIA.....	31
IV.2 Coût de l'investissement depuis le début du stage.....	31
IV.2.1 Investissement matériel.....	31
IV.2.2 Investissement humain.....	31
IV.3 Impact économique du projet Cycab.....	32
IV.4 Moyen mis en œuvre pour promouvoir la vente de la voiture électrique	32
IV.5 Tableau récapitulatif	33
Conclusion.....	34
Indexe	35
Bibliographie	36

Introduction

La robotique mobile existe depuis longtemps dans les laboratoires. L'idée d'appliquer les recherches faites dans ce domaine au transport est apparue récemment. De nombreux projets nationaux, européens (Prometheus, Praxitèle, Chauffeur ou Path..) ont ainsi vu le jour. Tous ces projets ont pour but d'apporter soit une aide à la conduite soit des manoeuvres automatiques. En France, des nouveaux concepts ont fleuri suite à ces projets comme celui de TUPI (Transport Urbain Public individuel) du projet Praxitèle ou encore celui de TULIP (Transport Urbain Libre Individuel et Public) de PSA. Dans le cadre du projet Praxitèle et du concept TUPI, un nouveau véhicule électrique, le Cycab est réalisé à l'INRIA Rhône-Alpes. Le service des moyens robotique l'a doté d'un contrôleur logiciel, ORCCAD. Le travail restant à faire est de lui spécifier des missions complexes et réalisables. Quelque soit le scénario, une mission complexe peut se décomposer en différentes tâches élémentaires.

Ces tâches élémentaires ont été identifiées et sont :

- le suivi de trajectoire nominale ;
- le suivi de trajectoire avec évitement d'obstacles ;
- le parking automatique par créneau ;
- le suivi d'un véhicule ;

Pour spécifier, vérifier et implémenter ces missions sur la voiture électrique, nous adapterons et utiliserons les outils du système ORCCAD. Ces tâches robotiques seront testées avec une simulation développée sous le logiciel SIMPARC.

Ce deuxième tome s'articule selon quatre parties qui sont les suivantes :

.la première partie expose le calcul du modèle de la voiture et les algorithmes utilisés.

.la deuxième partie a pour but de montrer la façon dont ces algorithmes sont programmés avec le logiciel ORCCAD.

.la troisième partie montrera les tests et résultats obtenus.

.et enfin une quatrième partie présentera mon stage d'un point de vue économique.

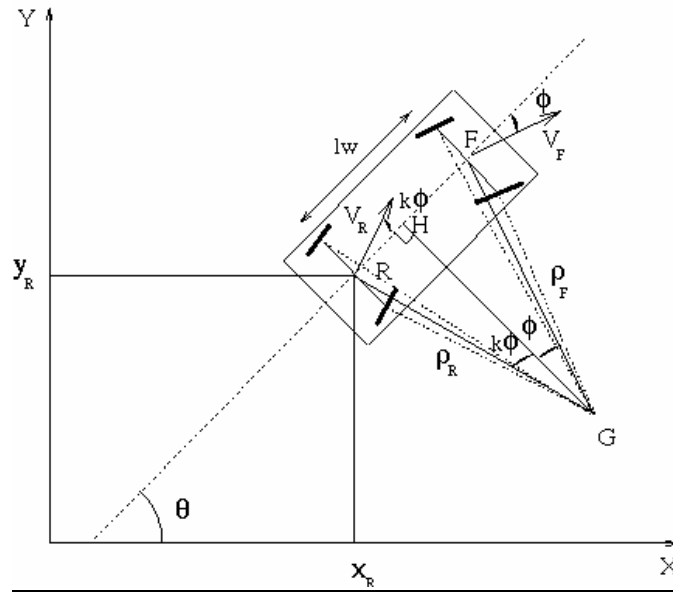
I Description des modèles et des algorithmes

L'objet de ce chapitre est de présenter le modèle mathématique du véhicule électrique utilisé et de décrire les algorithmes utilisés pour la programmation des tâches élémentaires.

I.1 Modélisation du véhicule électrique [10]

Notations :

Soient $F=(x_F, y_F)$ (resp. $R=(x_R, y_R)$) les coordonnées du centre de l'essieu avant (resp. arrière). De même, δ_F (resp. δ_R) représente le braquage moyen des roues de l'essieu avant. lw correspond à l'empattement du véhicule. θ caractérise l'angle que fait l'axe longitudinal du véhicule avec l'axe des abscisses du repère lié à l'environnement. Enfin, ρ_F (resp. ρ_R) est le rayon de giration instantané associé au point F (resp. R).



Les roues avant et arrière sont directrices avec un coefficient de proportionnalité entre les deux braquages : si le braquage avant est de ϕ , le braquage arrière est de $k \phi$ (avec $k = 0.69$). On introduit le point H défini comme la projection orthogonale du centre de giration instantanée G sur l'axe longitudinal du véhicule. La distance GH peut être exprimée de la manière suivante :

$$GH = \frac{RH}{|\tan(k * \Phi)|} = HF \frac{\tan(k * \Phi)}{\tan(\Phi)} \quad (1)$$

ce qui permet d'obtenir :

$$RH = HF * \frac{\tan(k\Phi)}{\tan(\Phi)} \quad (2)$$

De plus :

$$RH + HF = l_w$$

On en déduit les expressions de RH et HF en fonction de l_w et ϕ :

$$\left\{ \begin{array}{l} RH = \frac{l_w}{1 + \frac{\tan(\Phi)}{\tan(k\Phi)}} = l_w * \frac{\cos(\Phi) * \sin(k\Phi)}{\sin(\Phi + k\Phi)} \\ HF = \frac{l_w}{1 + \frac{\tan(k\Phi)}{\tan(\Phi)}} = l_w * \frac{\cos(k\Phi) * \sin(\Phi)}{\sin(\Phi + k\Phi)} \end{array} \right. \quad (4)$$

Les rayons de braquage ρ_R et ρ_F sont donnés par :

$$\left\{ \begin{array}{l} \rho_R = \frac{RH}{|\sin(k\Phi)|} = l_w * \frac{\cos(\Phi)}{|\sin(\Phi + k\Phi)|} \\ \rho_F = \frac{HF}{|\sin(\Phi)|} = l_w * \frac{\cos(k\Phi)}{|\sin(\Phi + k\Phi)|} \end{array} \right. \quad (5)$$

La vitesse de rotation instantanée peut s'écrire :

$$\mathbf{q}' = \frac{v_R}{\mathbf{r}_R} = \frac{v_F}{\mathbf{r}_F} \quad (6)$$

A partir de (5) et (6), on en déduit :

$$v_R = v_F * \frac{\mathbf{I}_R}{\mathbf{I}_F} = v_F * \frac{\cos(\mathbf{f})}{\cos(k * \mathbf{f})} \quad (7)$$

Les équations du mouvement en R sont obtenues de manière géométrique :

$$\left\{ \begin{array}{l} x'_R = v_R * \cos(\mathbf{q} + k * \mathbf{f}) \\ y'_R = v_R * \sin(\mathbf{q} + k * \mathbf{f}) \\ \mathbf{q}'_R = v_R * \frac{\sin(\mathbf{f} + k * \mathbf{f})}{l_W * \cos(\mathbf{f})} \end{array} \right. \quad \text{ou} \quad \left\{ \begin{array}{l} x'_R = v_F * \frac{\cos(\mathbf{f})}{\cos(k * \mathbf{f})} * \cos(\mathbf{q} + k * \mathbf{f}) \\ y'_R = v_F * \frac{\cos(\mathbf{f})}{\cos(k * \mathbf{f})} * \sin(\mathbf{q} + k * \mathbf{f}) \\ \mathbf{q}' = v_F * \frac{\sin(\mathbf{f} + k * \mathbf{f})}{l_W * \cos(\mathbf{f})} \end{array} \right. \quad (8)$$

$$\left\{ \begin{array}{l} x'_F = v_R * \frac{\cos(k * \mathbf{f})}{\cos(\mathbf{f})} * \cos(\mathbf{q} + \mathbf{f}) \\ y'_F = v_R * \frac{\cos(k * \mathbf{f})}{\cos(\mathbf{f})} * \sin(\mathbf{q} + \mathbf{f}) \\ \mathbf{q}' = v_F * \frac{\sin(\mathbf{f} + k * \mathbf{f})}{l_W * \cos(k * \mathbf{f})} \end{array} \right. \quad \text{ou} \quad \left\{ \begin{array}{l} x'_F = v_R * \frac{\cos(k * \mathbf{f})}{\cos(\mathbf{f})} * \cos(\mathbf{q} + \mathbf{f}) \\ y'_F = v_R * \frac{\cos(k * \mathbf{f})}{\cos(\mathbf{f})} * \sin(\mathbf{q} + \mathbf{f}) \\ \mathbf{q}' = v_F * \frac{\sin(\mathbf{f} + k * \mathbf{f})}{l_W * \cos(k * \mathbf{f})} \end{array} \right. \quad (9)$$

De même, on déduit les équations du mouvement en F :

$$\left\{ \begin{array}{l} x'_F = v_F * \cos(\mathbf{q} + \mathbf{f}) \\ y'_F = v_F * \sin(\mathbf{q} + \mathbf{f}) \\ \mathbf{q}' = v_R * \frac{\sin(\mathbf{f} + k * \mathbf{f})}{l_W * \cos(\mathbf{f})} \end{array} \right.$$

I.2 Le suivi de trajectoire

Le but de cette application est de pouvoir asservir le véhicule électrique afin qu'il suive une trajectoire nominale générée par un générateur de trajectoire.

I.2.1 Le générateur de trajectoire

Un véhicule mobile est repéré dans un repère cartésien par trois coordonnées qui sont (x, y, θ) où x , y et θ représentent respectivement l'abscisse, l'ordonnée et l'angle d'orientation du véhicule. Grâce aux intentions de l'utilisateur qui peuvent être :

- Aller tout droit d'une certaine distance et à une certaine vitesse
- Tourner d'un certain angle, d'un certain rayon giration et à une certaine vitesse

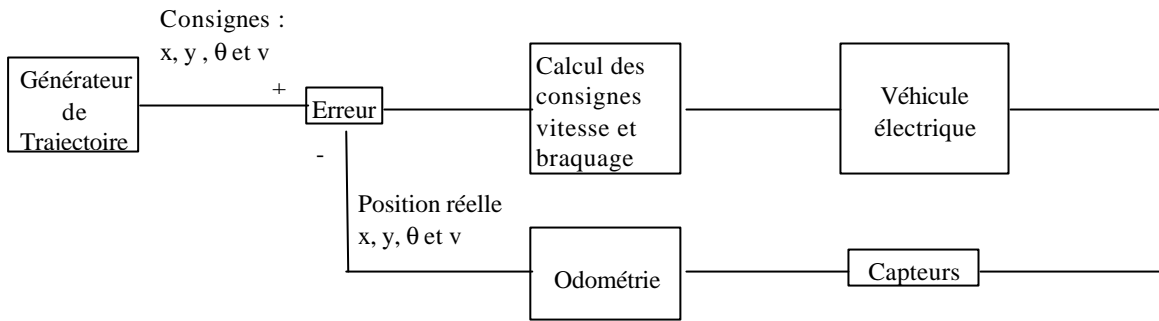
Le générateur traduit ces intentions en trajectoire en tenant compte des contraintes cinématiques du véhicule électrique. Chaque point de cette trajectoire contient alors quatre paramètres qui sont x , y , θ et la vitesse de la voiture.

I.2.2 Algorithme de KANAYAMA[4]

Cet algorithme a pour but d'amener un véhicule mobile d'une configuration donnée de départ à une autre configuration donnée finale. Une configuration d'un véhicule mobile est définie par ces coordonnées x , y , θ et sa vitesse v . Tout véhicule mobile est contrôlé par sa vitesse et l'angle de braquage de ces roues.

Pour le cycab, la vitesse est commandée par une consigne en vitesse angulaire sur chaque roue et pour le braquage des roues par une consigne de position.

Le schéma fonctionnel de l'algorithme est le suivant :



L'odométrie permet de calculer la position (x, y, θ) du véhicule grâce aux données fournies par les capteurs et en utilisant les équations du modèle (9) du Cycab.

L'algorithme peut être décomposé en 5 points :

- 1) le prochain point $(x, y, \theta$ et $v)$ à atteindre est fourni par le générateur de trajectoire
- 2) l'état actuel de la voiture $(x, y, \theta$ et $v)$ est calculé grâce à l'odométrie
- 3) Calcul de l'erreur entre le point réel de la voiture et le point nominal : $(x_e, y_e, \theta_e$ et $v_e)$
- 4) A partir des erreurs, on calcule les consignes de vitesses et de braquage suivant les équations suivantes :

$$\left\{ \begin{array}{l} \mathbf{q}' = \mathbf{q}'_{ref} + v_{R, ref} * ((k_y * y_e + k_\theta * \sin(q_e)) \quad (10) \\ v_R = v_{R, ref} * \cos(q_e) + k_x * x_e \\ \mathbf{f} = \frac{\mathbf{q}' * l_w}{v_{R, ref} * 0,76} \quad \text{linéarisation de } \mathbf{q}' = v_R * \frac{\sin(\mathbf{f} + k * \mathbf{f})}{l_w * \cos()} \quad \text{sur } \left[\frac{-p}{4}, \frac{-p}{4} \right] \end{array} \right.$$

Où

→ $q_e = (x_e, y_e, \mathbf{q}_e)$ représente l'erreur entre la configuration de référence et la

configuration courante

- q'_{ref} et v_{ref} sont respectivement les vitesses de référence d'orientation et linéaire de la voiture.
- k_x, k_y et k_q sont des constantes positives à choisir de façon empirique

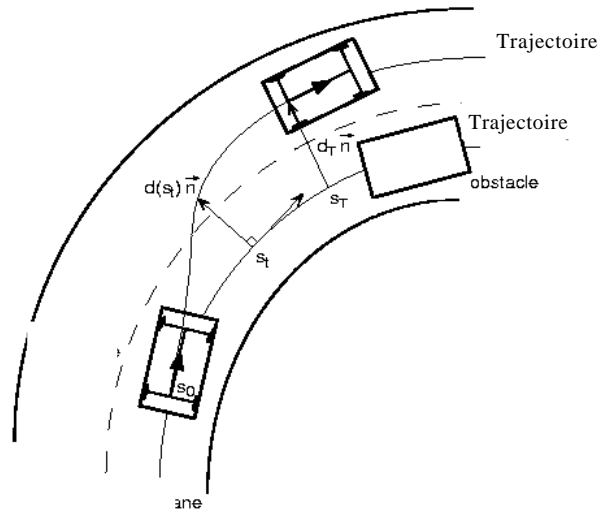
5) on revient au point 1) et on itère jusqu'au dernier point de la trajectoire.

I.3 Evitement d'obstacle [3]

L'évitement d'obstacle consiste à suivre une trajectoire nominale. Lorsque la véhicule détecte un obstacle sur sa trajectoire nominale, il génère une nouvelle trajectoire en translatant la trajectoire nominale vers sa droite ou sa gauche. Une fois l'obstacle dépassé, le véhicule reprend sa trajectoire nominale.

Lorsque la voiture détecte un obstacle, elle suit l'algorithme suivant qui se résume en 4 points :

- 1) Génération d'une trajectoire locale qui connecte la trajectoire nominale à une trajectoire 'parallèle'
- 2) Suivi de cette deuxième trajectoire jusqu'à dépassement de l'obstacle : la trajectoire 'parallèle' est atteinte
- 3) Génération d'une autre trajectoire locale qui connecte la trajectoire 'parallèle' à la nominale.
- 4) Suivi de cette trajectoire locale jusqu'à atteindre la trajectoire nominale.



Translation de la trajectoire nominale

Une trajectoire de raccordement possible entre les deux trajectoires peut suivre l'équation d'un polynôme du cinquième degré suivant :

$$d(s) = d_r * (10 * (\frac{S}{S_T})^3 - 15 * (\frac{S}{S_T})^4 + 6 * (\frac{S}{S_T})^5) \quad (11)$$

Où d_r est la distance entre les deux trajectoires pendant le dépassement, d_r doit être fixée avant le lancement de l'algorithme d'évitement, S_T étant la longueur curviligne nécessaire pour effectuer le changement de trajectoire, $S = S_T$ l'abscisse curviligne $S=0$ au départ de la manoeuvre de changement.

La distance d_r est supposée connue. La valeur minimale pour S_T est estimée par la relation :

$$S_{T, \min} = \frac{p * \sqrt{k * d_r}}{2 * C_{\max}} \quad (12)$$

Où C_{\max} est la courbure maximum permise.

$$C_{\max} = \min \left\{ \frac{\tan(\mathbf{f}_{\max})}{L}, \frac{\mathbf{g}_{\max}}{v_{R,ref}^2} \right\} \quad (13)$$

\mathbf{g}_{\max} est l'accélération latérale maximum autorisée et k une constante empirique ($k=1.17$)

Quand un obstacle est détecté, la valeur $S_{T, \min}$ est calculée selon la relation (2). Une comparaison est alors faite entre la distance voiture - obstacle et $S_{T, \min}$. Si la distance voiture - obstacle est supérieure à $S_{T, \min}$ alors la manoeuvre évitement est engagé sinon la voiture s'arrête.

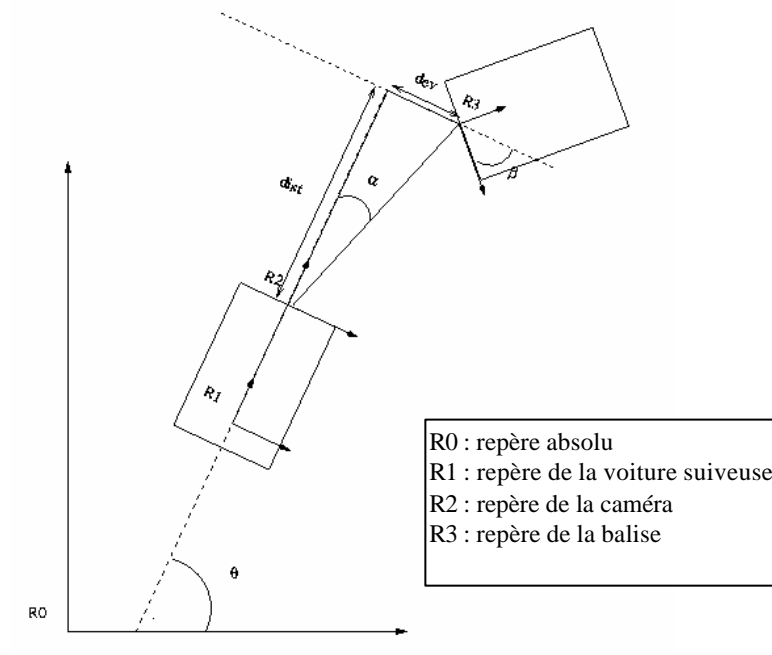
Dans le cas où la manoeuvre d'évitement d'obstacle est engagé, à chaque instant t depuis le début de la manoeuvre, la position de référence P_{ref} est translatée selon le vecteur $d(S_t) * n$ où n est un vecteur unité normale au vecteur vitesse. L'orientation de référence \mathbf{q}_{ref} devient

alors $\mathbf{q}_{ref} + \arctan\left(\frac{\|d\|}{\|s\|}(S_t)\right)$ et la vitesse de référence $v_{R, ref}$,

$$v_{R, ref}(t) = \frac{dist(P_{ref}, P_{ref}(t + \Delta t))}{\Delta t} \quad (14)$$

Où $dist$ est la distance euclidienne.

I.4 Le suivi de véhicule



Cette tâche consiste en un suivi d'un véhicule conduit manuellement par un véhicule automatique. Le procédé d'accrochage est immatériel. Chaque voiture possède une caméra linéaire CCD et une balise infrarouge à trois points. La caméra [12] permet de repérer la balise et par un calcul par triangulation(cf annexe), elle fournit dans le repère de la caméra R2 les données suivantes :

- Balise hors/dans champ de la caméra

si la balise est dans le champ de la caméra :

- dist : ordonnée de la balise dans le repère de la caméra.
- dev : abscisse de la balise dans le repère de la caméra
- β : angle de rotation entre R2 et R3

Grâce à ces données fournies par la caméra, nous sommes en mesure de calculer à chaque instant la configuration $(x, y, \theta$ et v) exacte de la balise dans $R0$ et $R1$, son orientation et également sa vitesse. Soit dans $R0$ $(x0, y0, \theta0)$ la configuration de la voiture et $(xb0, yb0, \thetab0)$ la configuration de la balise à déterminer. La caméra nous renvoie dans le repère $R2$, $xb2 = dev$, $yb2 = dist$ et $\theta b2 = \beta$. Soit yc l'ordonnée de la caméra dans $R1$.

Passage de $R2$ à $R1$:

$$\begin{cases} xb1 = xb2 - yc \\ yb1 = yb2 \\ qb1 = qb2 \end{cases}$$

Passage de $R1$ à $R0$:

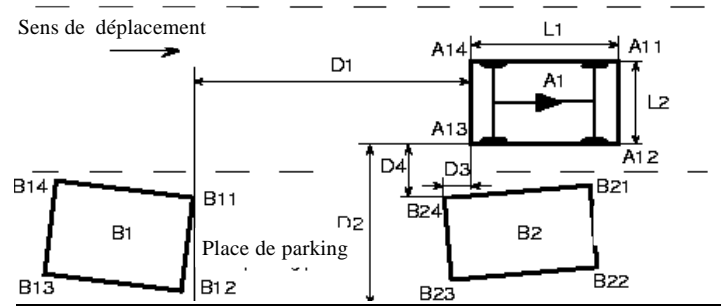
$$\begin{cases} xb0 = x + \cos(\mathbf{q} + \mathbf{a1}) * \sqrt{xb1^2 + yb1^2} \\ yb0 = y + \sin(\mathbf{q} + \mathbf{a1}) * \sqrt{xb1^2 + yb1^2} \\ qb0 = \mathbf{q} + qb1 \end{cases}$$

avec $\mathbf{a1} = a \tan\left(\frac{yb1}{xb1}\right)$

Nous pouvons alors appliquer l'algorithme de Kanayama.

La configuration suivante à atteindre n'est plus donnée par le planificateur de trajectoire mais par un calcul sur les données fournies par la caméra.

I.5 Le parking automatique par créneau [5] [3]



Au départ le conducteur doit placer son véhicule de façon assez parallèle de la file de voiture puis lance la tâche de recherche de place libre . Cette tâche utilise les ultrasons latéraux pour se positionner vis à vis de la file de véhicules. Si la place est occupée, les ultrasons renvoient une distance moins grande que si la place est libre .Si il y a une place de libre , en utilisant le temps pendant la détection de la place libre et connaissant la vitesse du véhicule, la longueur de la place peut-être déduite. Pour la largeur de la place libre, nous la supposons connue d'avance.

Une fois cette recherche faite, le véhicule aura mesuré les distances D1, D2, D3, D4.

Sur la figure, nous distinguons le véhicule automatique A1, les deux autres voitures qui sont déjà garées B1 et B2.

Les commandes en vitesse v et en direction ϕ à envoyer à la voiture suivent les équations suivantes :

$$\begin{cases} f(t) = f_{\max} \cdot k_f \cdot A(t) & 0 \leq t \leq T \\ v(t) = v_{\max} \cdot k_v \cdot B(t) & 0 \leq t \leq T \end{cases} \quad (15)$$

où $f_{\max} \geq 0$ et $v_{\max} \geq 0$ sont les consignes maximums de vitesse et de braquage des roues , $k_f = \pm 1$ correspond à un choix de braquage des roues à droite (+1)ou à gauche(-1) , $k_v = \pm 1$ commande la marche avant (+1)/ arrière(-1).

$$A(t) = \begin{cases} 1 & 0 \leq t < t' \\ \cos\left(\frac{\mathbf{p}^*(t-t')}{T^*}\right) & t' \leq t \leq T-t' \\ -1 & T-t' < t \leq T \end{cases} \quad (16)$$

$$B(t) = 0.5 * (1 - \cos\left(\frac{4 * \mathbf{p}^* t}{T}\right)) \quad (17)$$

où T est la durée de la manoeuvre , $t' = \frac{T-T^*}{2}$, $T^* < T$

$$T^*_{\min} = \mathbf{p} \max \left\{ \frac{\mathbf{f}_{\max}}{\mathbf{f}'_{\max}}, \sqrt{\frac{\mathbf{f}_{\max}}{\mathbf{f}'_{\max}}} \right\} \quad (18)$$

où \mathbf{f}_{\max} et \mathbf{f}'_{\max} sont la vitesse et l'accélération maximales de braquage des roues admissibles.

T^*_{\min} donne la durée

pour tourner les roues de $-\mathbf{f}_{\max}$ à \mathbf{f}_{\max} . On peut initialiser $T^* = T^*_{\min}$

La valeur T est calculée selon les contraintes de vitesse v_{\max} , d'accélération v'_{\max} et de la relation $T^* < T$.

$$T_{\min} = \max \left\{ \frac{2 \cdot v_1(D1)}{v'_{\max}}, T^* \right\} \quad (19)$$

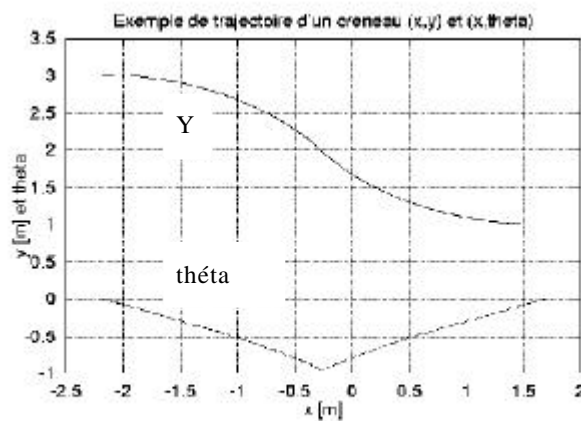
où $v_1(D1) \leq v_{\max}$ est une fonction empirique.

La recherche de T et de \mathbf{f}_{\max} permet d'obtenir des valeurs optimales afin de satisfaire les deux conditions suivantes :

$$\begin{cases} |(x_T - x_0) \cdot \cos(\mathbf{q}_0) + (y_T - y_0) \cdot \sin(\mathbf{q}_0)| < D1 \\ |(x_0 - x_T) \cdot \sin(\mathbf{q}_0) + (y_T - y_0) \cdot \cos(\mathbf{q}_0)| < D2 \end{cases} \quad (20)$$

L'algorithme se résume aux quatre points suivants :

- 1) Obtenir les valeurs D1 et D2 en utilisant les capteurs ultrasoniques
- 2) Recherche des valeurs T et f_{\max} en évaluant le modèle (9) avec les contrôles de vitesse et de braquage (15) jusqu'à que les conditions (20) soient satisfaites .
- 3) Envoyer les contrôles (15) à la voiture tout en vérifiant avec les capteurs, les risques de collisions
- 4) Obtenir la position de la voiture par rapport aux autres objets de l'environnement. Si la voiture est bien garée alors la tâche est arrêtée sinon revenir au point 1).



La courbe (x, y) dépend des deux paramètres T et f_{\max} : l'allure ne change pas mais le choix de T et f_{\max} peut en modifier son amplitude en X et en Y.

II Programmation logicielle

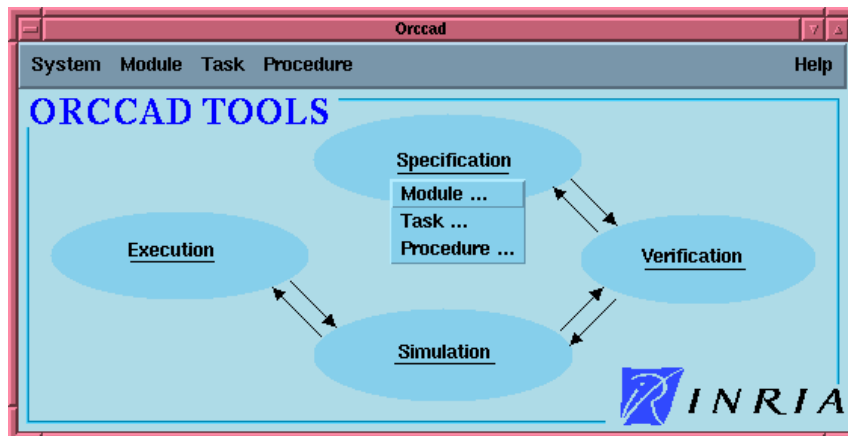
II.1 Méthodologie générale de programmation

(Cette méthodologie a été présentée de façon plus détaillée dans le Tome1)

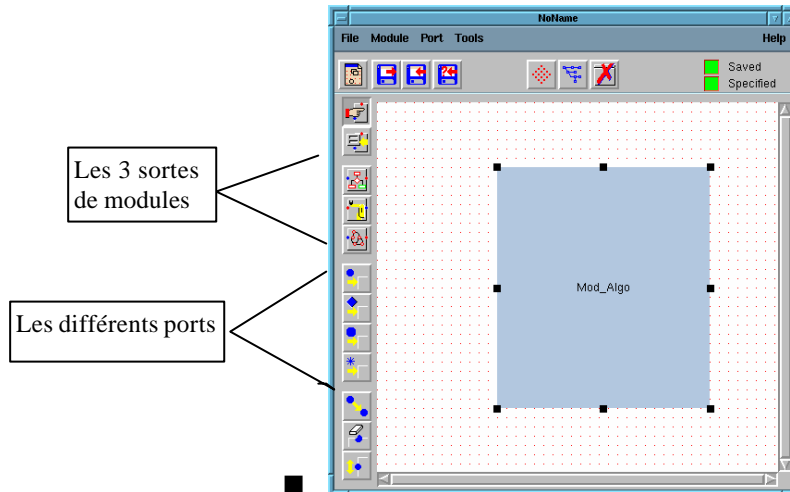
Pour l'implantation le logiciel ORCCAD est utilisé. ORCCAD permet de programmer des tâches robotiques qui sont toutes constituées de trois types de modules différentes :

- les modules algorithmiques
- les modules automates qui gèrent les interruptions de lancement et d'arrêts des tâches
- les modules ressources physiques qui représentent les entrées et sorties des robots. Ils ont aussi pour rôle l'initialisation du robot. Pour notre cas, la ressource physique sera la voiture électrique.

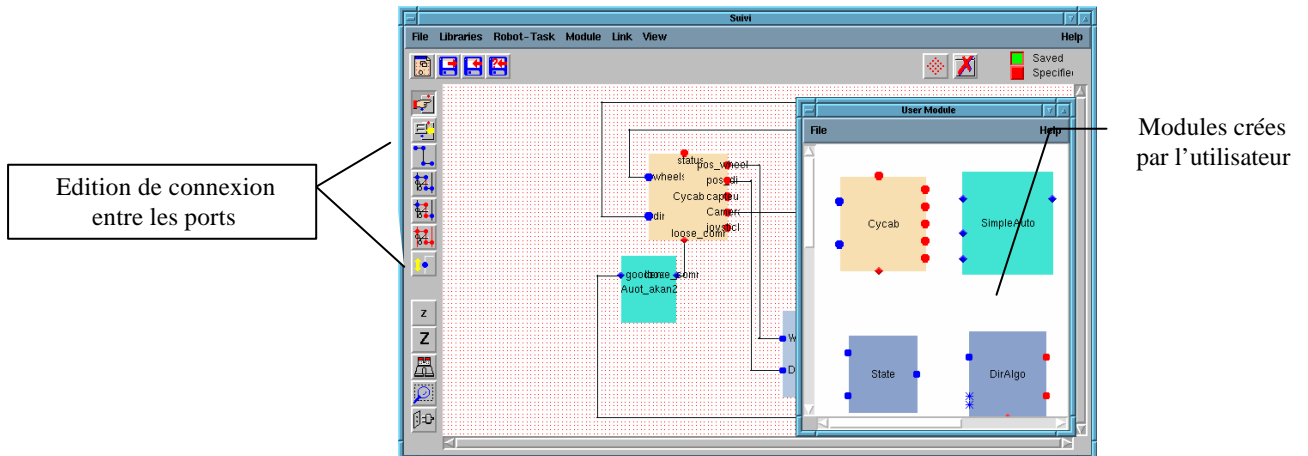
La méthodologie générale de programmation sous ORCCAD est la suivante :



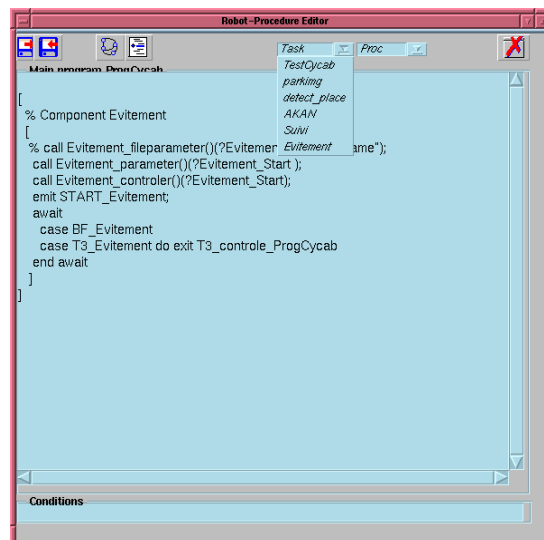
■ Création des modules nécessaires à la tâche robotique :



■ Création de la tâche robotique en assemblant les modules créés :



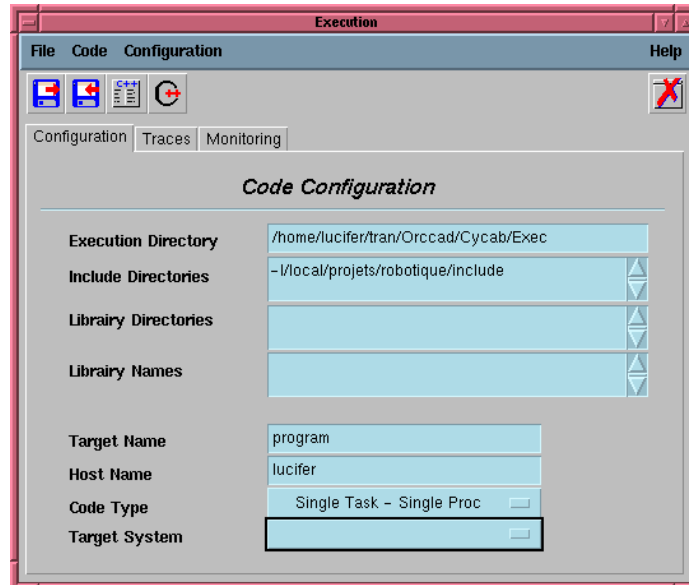
- Génération du code ESTEREL (Procedure) : la génération se fait de façon automatique



```
Robot-Procedure Editor
Task: TestCycab
Proc: parking
detect_place
AKAN
Suivi

Main program ProgCycab
[
% Component Evitement
[
% call Evitement_fileparameter()(?Evitement_ame");
call Evitement_parameter()(?Evitement_Start);
call Evitement_controler()(?Evitement_Start);
emit START_Evitement;
await
case BF_Evitement
case T3_Evitement do exit T3_controler_ProgCycab
end await
]
]
]
Conditions
```

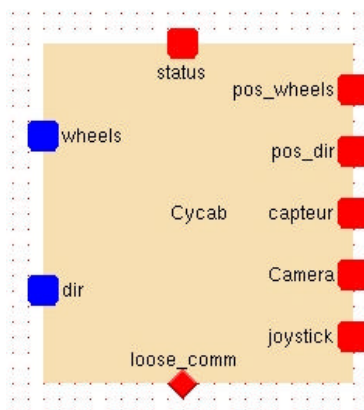
- Compilation : la compilation peut se faire pour différents systèmes



II.2 La ressource physique

La première chose à faire pour pouvoir programmer des tâches sur un robot est de connaître ses entrées et ses sorties. Donc avant toute chose, il faut développer la ressource physique qui sera utilisée dans toutes les tâches développées.

Le module ressource physique une fois développée pour le Cycab:



La ressource physique représente les accès robot: toutes les entrées et sorties doivent être présentes. Pour notre voiture électrique , nous distinguons

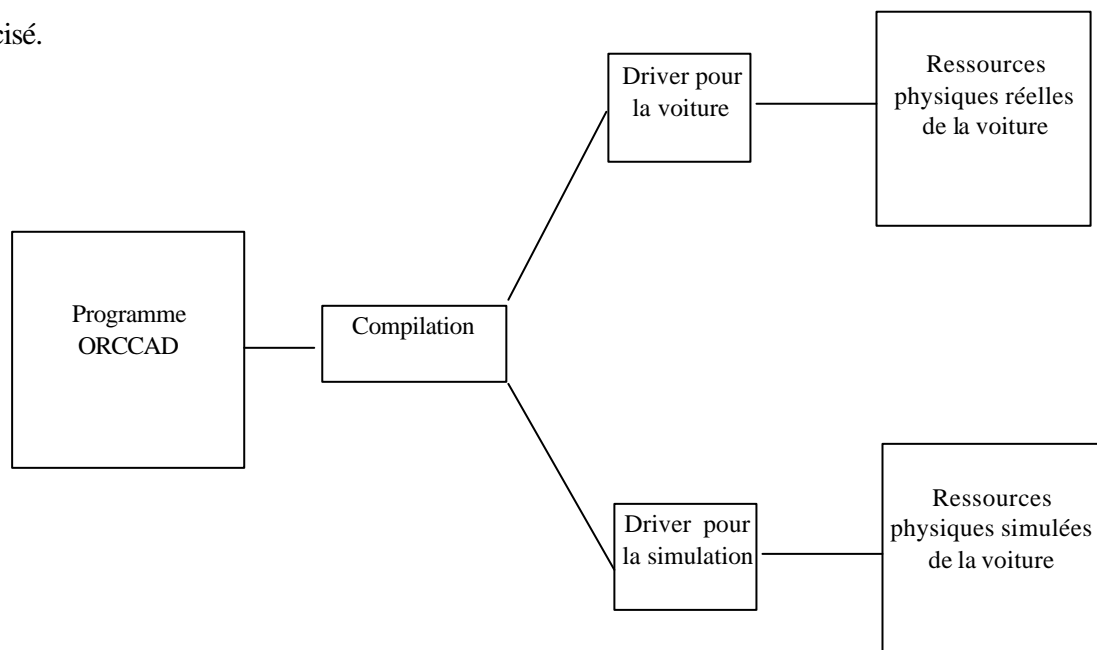
en entrée :

- La commande en position de braquage des roues (dir)
- Les commandes indépendantes en vitesse des quatre roues (wheels)

en sortie :

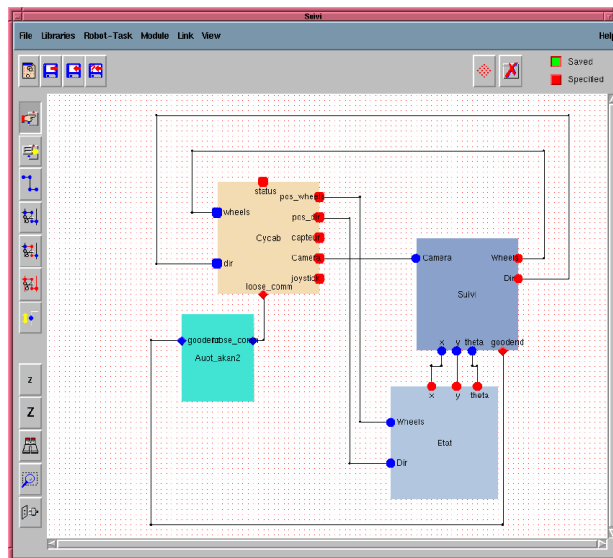
- La position angulaire de chaque roue (pos_wheels)
- La position angulaire du braquage des roues (pos_dir)
- les seize capteurs ultrasonores (capteur)
- La caméra linéaire CCD (Caméra)
- l'état de la voiture (status)
- l'état de la communication sur le bus CAN.

Chaque port d'entrée et de sortie est associé à une fonction qui est le lien entre la ressource physique réelle de la voiture et celle de ORCCAD. Chaque port doit également avoir une fonction pour la partie simulation C'est lors de la compilation que le choix de la fonction est précisé.



II.3 Les tâches robotiques

Les tâches que nous avons développées suivent le schéma suivant :



Le schéma ci-dessus est constitué :

- de la ressource physique décrit dans le paragraphe précédent (Cycab)
- du module automate (auto_kan2)
- Et de deux modules algorithmique :
 - Etat : calcul la position du véhicule par odométrie
 - Suivi : calcul des commandes à envoyer selon les algorithmes qui ont été présentés précédemment : consigne vitesse et de braquage des roues.

Chacun de ces modules est associe à plusieurs fichiers :

inc.h : code contenant les includes

var.c : code de déclaration des variables

init.c : code d'initialisation du module, exécuté qu'une seule fois au début de la tâche.

Compute.c : code où se trouve l'algorithme de calcul

end.c : code exécuté lorsque la tâche se termine.

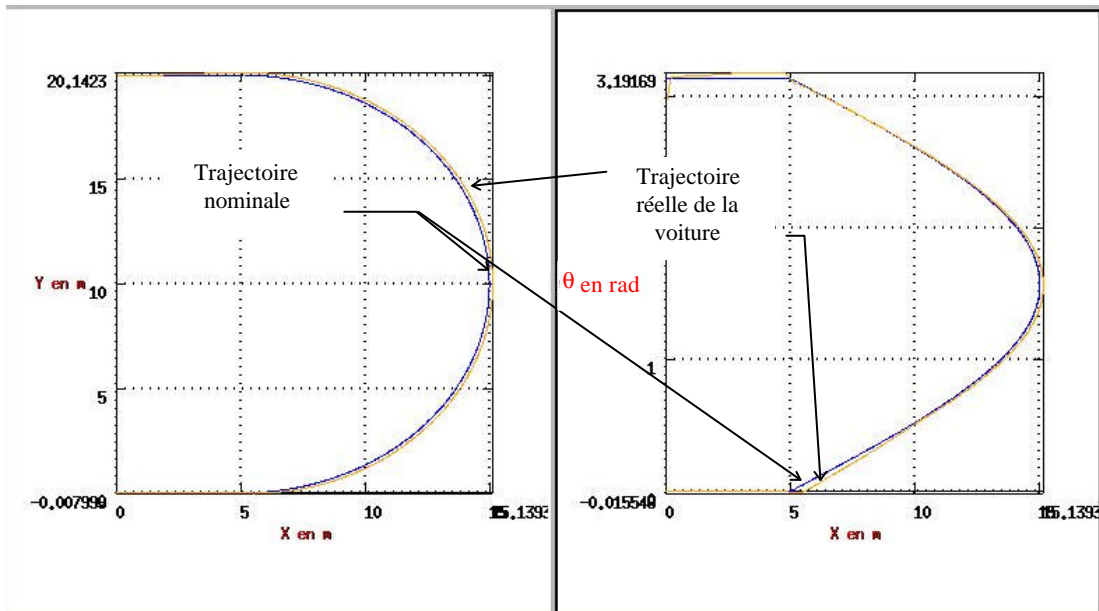
III Tests et résultats

III.1 Tests et résultats avec la simulation

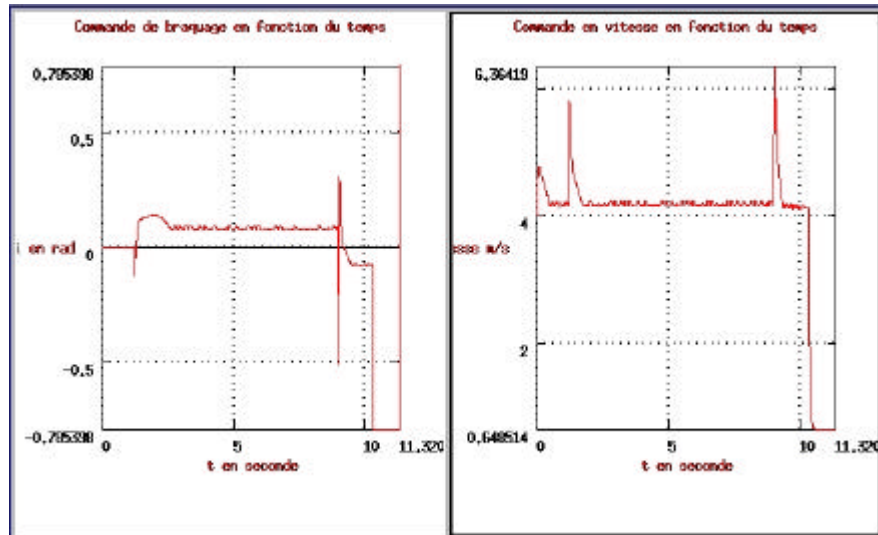
Nous avons mis en place une simulation de la voiture électrique avec le logiciel SIMPARC. Cette simulation est décrite dans le Tome1. Elle permet de tester les programmes écrits sous ORCCAD. En effet ORCCAD et SIMPARC ont été développés de façon à être compatibles. Ainsi avant de tester les tâches robotique développées sous ORCCAD sur le véhicule réel, nous pouvons les tester avec la simulation. Ceci permet de corriger rapidement les éventuelles erreurs. En effet le même code est spécifié avec ORCCAD est utilisé pour la simulation et pour le véhicule réel.

III.1.1 Le suivi de trajectoire

Les mesures suivantes sont faites avec : $k_x=10$, $k_y=1.118034$, $k_\theta=2.5$



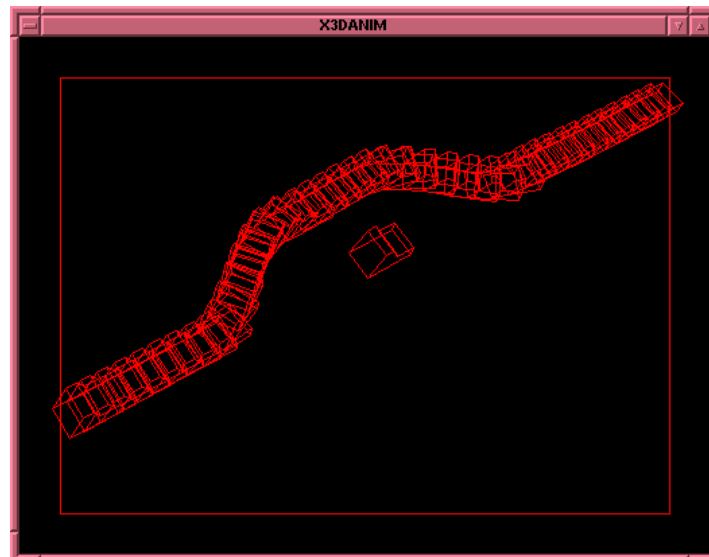
Les performances de suivi dépend du choix des constantes k_x , k_y et k_θ .



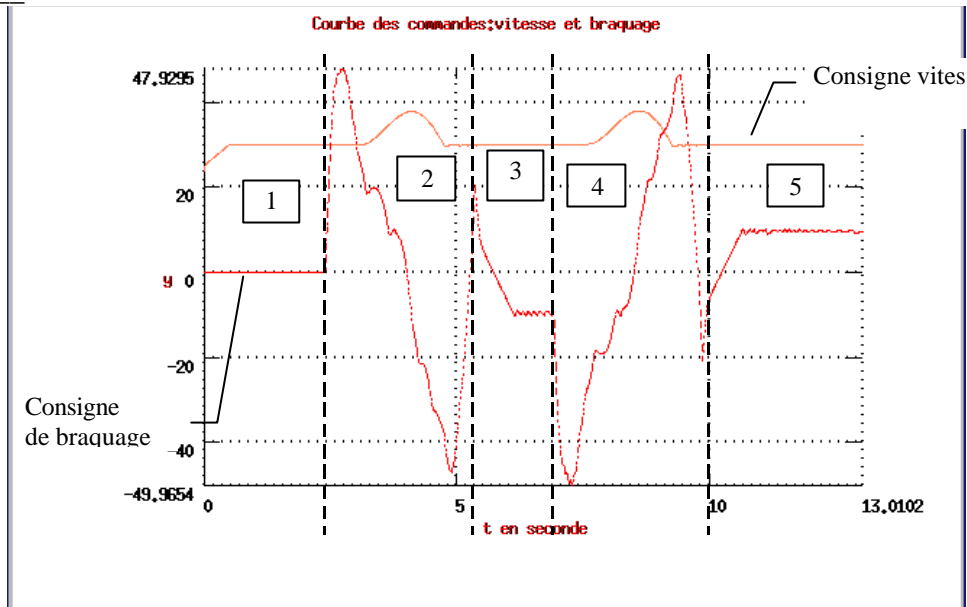
III.1.2 Evitement d'obstacle

Pour tester cette tâche , nous avons fait suivre la voiture une trajectoire de référence et sur cette trajectoire nous avons mis un obstacle.

Un aperçu d'une simulation d'évitement d'obstacle :



Visualisation de la scène grâce à x3d



1 différentes phases de l'évitement d'obstacle sont les suivantes :

- 2 : Suivi d'une trajectoire rectiligne nominale
- 3 : Déboîtement pour atteindre la trajectoire 'parallèle'
- 4 : Dépassement de l'obstacle
- 5 : Rabattement
- 6 : reprise du suivi de la trajectoire nominale

L'unité de l'axe des ordonnées est en pourcentage des valeurs maximums de vitesses (8m/s) et de l'angle de braquage ($\pi/4$).

Ces mesures ont été faites avec $k_x=2.5$, $k_y=1$ et $k_\theta=2$.

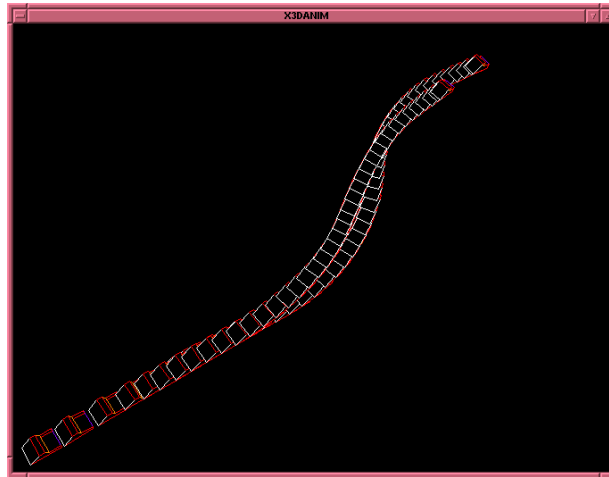
La courbe de l'angle de braquage des roues possède des pics. Ils peuvent être atténué pour un choix de k_θ moins élevé.

III.1.3 Le suivi de véhicule

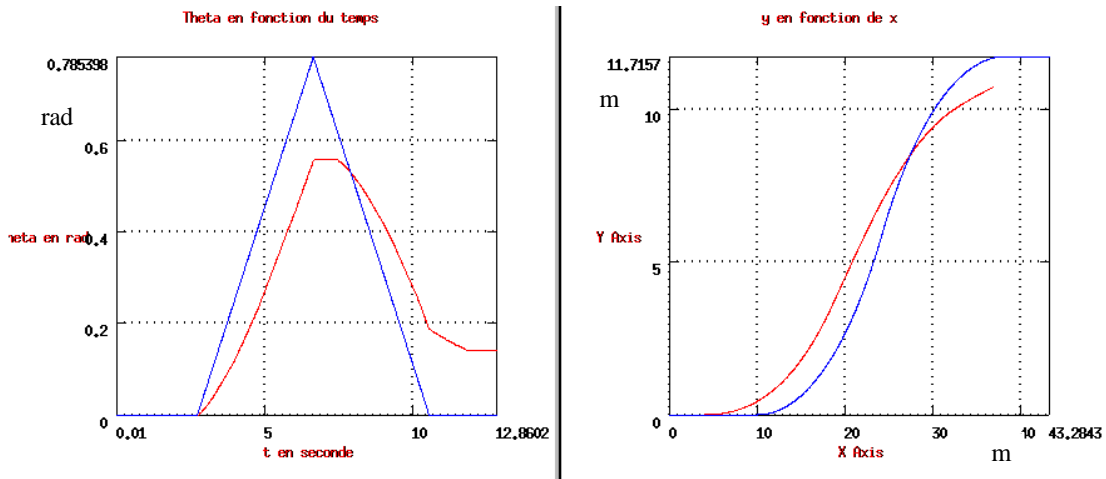
Pour simuler cette application, une caméra a été ajouté à notre voiture ainsi qu'une deuxième voiture qui porte la balise. Cette deuxième voiture suit une trajectoire nominale.

Grâce aux données renvoyées par la caméra, nous pouvons situer exactement la position de la balise dans le repère de la voiture.

Le simulation suivante est faite avec les paramètres suivantes : $k_x=4$, $k_y=0.0064$ et $k_\theta=0.16$.



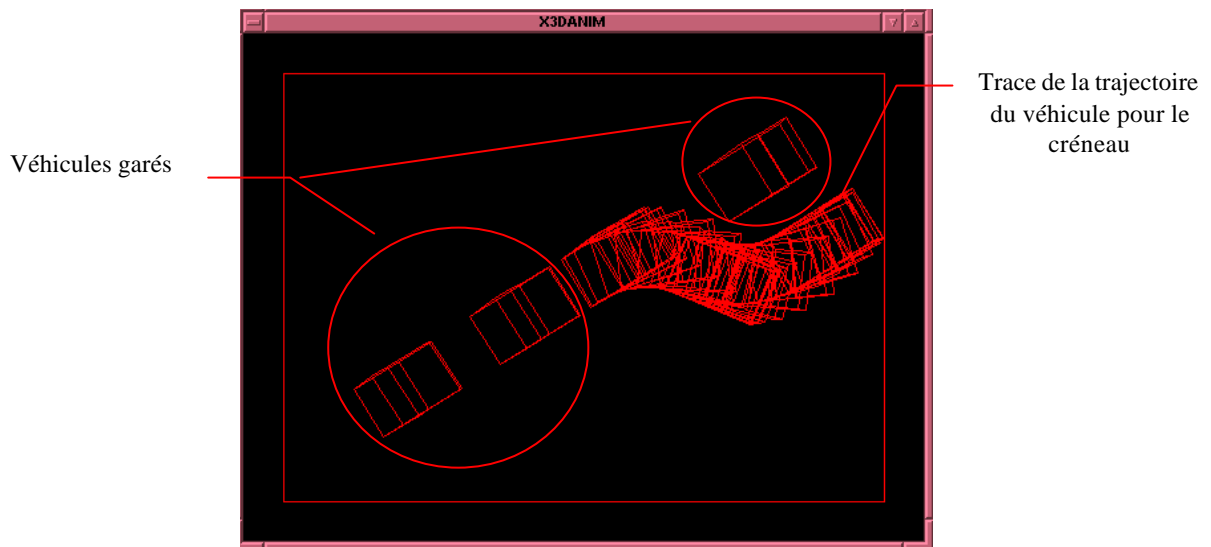
Visualisation avec x3d



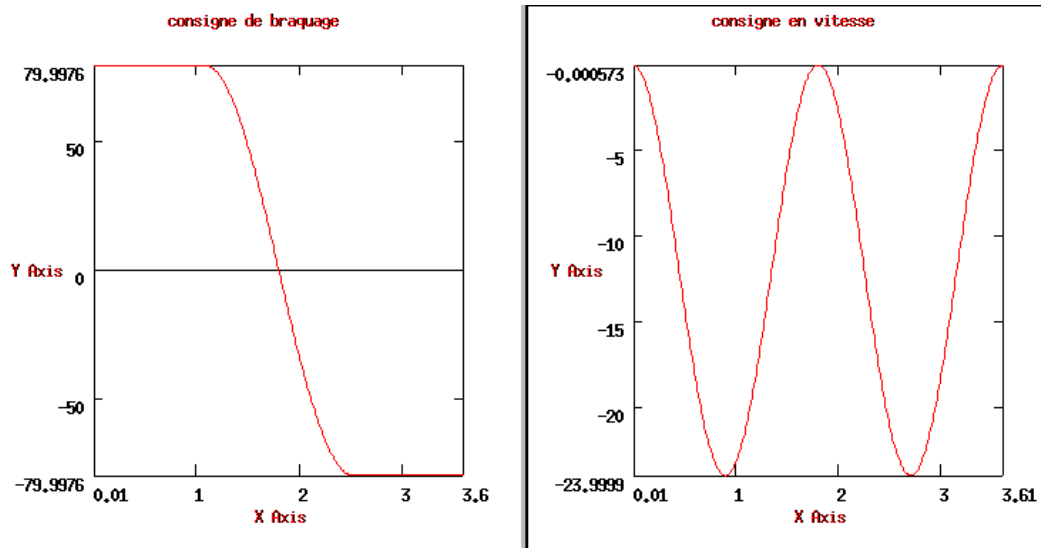
L'application de l'algorithme de KANAYAMA donne d'assez bon résultat mais un décalage existera toujours entre la trajectoire effectuée par la balise et la voiture suiveuse. Ceci s'explique par le fait que l'asservissement de la configuration (x, y, θ) de la voiture se fait sur chaque configuration (x, y, θ) de la balise. Une solution à ce problème est d'enregistrer les configurations de la balise et d'asservir sur ces configurations. Le problème de cette solution

est le risque de perdre la balise du champ de la caméra à moins d'avoir un angle d'ouverture de la caméra assez grande.

III.1.4 Le parking par créneau



Visualisation avec x3d



L'unité des abscisses des courbes sont en seconde et les ordonnées sont en pourcentage des valeurs maximums des consignes qui sont pour la vitesse 8.33 m/s et le braquage des roue $\pi/4$.

III.2 Test et résultat avec la voiture électrique le Cycab

Il a été prévu au début de mon stage de tester les tâches développées sur le véhicule réel, mais celui-ci était indisponible pour des problèmes matériels, nous n'avons pas pu l'utiliser. Aussi pour pallier ce petit imprévu, notre travail s'est focalisé sur la mise en place d'une simulation avec le logiciel SIMPARC. Ceci nous a permis de développer plus de tâches robotiques que prévu. Cette simulation pourra être par la suite ré-utilisée pour d'autres applications développées sous ORCCAD pour le Cycab.

IV Partie économique

IV.1 Raisons du choix du sujet fait par l'INRIA

Le service robotique de l'INRIA de Rhône-alpes est chargé de la mise en œuvre des outils et logiciels pour les expérimentations robotiques des projets de recherche du site. Un véhicule électrique réalisé à l'INRIA est disponible et le service doit maintenant le doter d'un contrôleur logiciel .

Le but de ce stage est de spécifier des applications complexes de déplacement d'un véhicule électrique automatique dans un parking.

Ce stage répond donc à un travail exploratoire en vue d'une conduite automatisée de véhicule électrique.

IV.2 Coût de l'investissement depuis le début du stage

IV.2.1 Investissement matériel

Ce projet n'a pas nécessité d'investissement supplémentaire.

IV.2.2 Investissement humain

La liste suivante présente la fonction et le coût des différentes personnes ayant participées à ce projet :

1 post-doctorant : 3H/semaine

3 ingénieurs de recherche : 10 H/semaine (total)

soit sur 26 semaines, 338 H. Ca représente environ 2,2 mois d'ingénieur.

La valeur de l'ingénieur dit "habillé" (charges, frais de fonctionnement) est d'environ 430000F par an, donc on peut estimer le coût de travail de ces personnes à :

$430000/12 \times 2,3 \cong 78\ 800\ F$

IV.2.3 Coût de mon travail :

- Durant ce projet, l'indemnité de stage s'élève à 4000 francs par personne, soit pour l'ensemble du projet : $4000 \text{ francs} \times 6 \text{ mois} = 24000\text{F}$.
- Charge : chaque personne travaillant dans les locaux de l'INRIA reçoit une subvention affectée à la prise de repas. Nous avons donc bénéficié de cette subvention s'élevant à :

$$26,5 \text{ francs} \times 5 \text{ jours/semaine} \times 26 \text{ semaines} = 3445 \text{ francs.}$$

Soit un coût total de 27445 francs

IV.3 Impact économique du projet Cycab

Les retombées industrielles à long terme dans le domaine de la recherche ne sont pas évaluables.

IV.4 Moyen mis en œuvre pour promouvoir la vente de la voiture électrique

Ce projet appartenant principalement à la recherche fondamentale, il n'existe donc pas de moyens de mise en œuvre pour la vente, mais un appel d'offre a été lancé pour la fabrication de 8 Cycab.

IV.5 Tableau récapitulatif

Coût de l'investissement	matériel	humain	total
ingénieurs / doctorant		78800 F	
stagiaires		24000 F	
divers	0 F	3445 F	
total	0 F	84645 F	84645 F

Conclusion

Le travail durant le stage s'est décomposé en trois parties , la première était la mise en place de la simulation avec le logiciel SIMPARC. Cette simulation peut être maintenant utilisée comme un moyen de test et de validation des programmes écrits avec ORCCAD avant de les transférer sur le véhicule réel.

La deuxième était une partie de programmation des tâches robotique et enfin la dernière était une phase de tests et de validation de ces tâches avec la simulation. Une fois le véhicule électrique sera opérationnel, ces tâches pourront y être transférées, en effet le même code ORCCAD est utilisé pour la simulation et pour le véhicule réel.

Sur le plan personnel, le travail effectué a été très intéressant et ceci pour plusieurs raisons.

Tout d'abord le sujet du stage m'a contraint à utiliser de nouveaux outils :

- environnement Unix
- environnement temps réel
- un outil de spécification d'application robotique : ORCCAD
- un outil de simulation de système robotique : SIMPARC.

Ce stage m'a aussi permis de connaître le monde de la robotique mobile et ses principaux algorithmes (suivi de trajectoire, évitement d'obstacle ...).Ce domaine exige à la fois des connaissances en automatique mais également en informatique.

Et enfin ceci m'a permis d'avoir une vision des méthodes de travail dans un laboratoire de recherche.

Index

créneau.....	2, 3, 13, 25
Cycab.....	1, 3, 8, 21, 27, 29
Evitement d'obstacle	2, 9, 23
KANAYAMA	2, 7, 25
odométrie	8, 21
<i>ORCCAD</i>	33
robots.....	16
SIMPARC	3, 22, 31, 33
simulation.....	2, 3, 20, 22, 23, 25, 26, 31, 33
suivi.....	2, 3, 7, 12, 22, 24, 31
tâches robotique.....	2, 3, 16, 20, 22, 27, 31
trajectoire	2, 3, 7, 8, 9, 10, 13, 22, 23, 24, 25, 31
ultrason.....	15, 19, 25

Bibliographie

- [1] J.J. Borrelly, E. Coste-Manière, B. Espiau, K. Kapellos, R. Pissard-Gibollet, D. Simon, N. Turro, *The ORCCAD Architecture*, The International Journal of Robotics Research, Sage Science Press, Avril 1998.
- [2] <http://www.inrialpes.fr/iramr/pub/Orccad/>
- [3] Ph. Garnier, Th. Fraichaed/C. Laugier, I. Paromtchick and A. Scheuer (1997) Motion autonomy through sensor-guided.
- [4] Kanayama Y, Kimura Y , Miyazaki F. , Noyuchi T (1991), A stable control method for a non-holonomic mobile robot.IEEE/RSJ int. Work-shop on Intelligent robots and systems, Osaka , Japan, November 1991, pp1236-1241
- [5] R. Murray and S. Sastry, « Steering non-holonomic systems using sinusoids », Proc. Of the IEEE int. Conf. on decision and control, Dec. 1990, pp2097-2101
- [6] L. Guiraud , Editeurs spécialisés pour le logiciel de simulation SIMPARC, stage DESS ISI septembre 1995.
- [7] S. Abdou, INPG/INRIA, thèse , Spécification, vérification et implémentation de missions appliquées à des véhicules automatiques, octobre 1997
- [8] J. J. BORRELLY, INRIA, SIMPARC V1.0 User manual , novembre 1991
- [9] D. Djian, DEA simulation de systèmes robotique ayant des interactions avec l'environnement
- [10] P. Garnier, INRIA, Cinématique des véhicules Ligier et Cycab, février 1998
- [11] Rapport annuel 1997, INRIA
- [12] N. Cardiles, INRIA , rapport de stag 1997.