

Rapport de stage

Brieuc Desoutter

29 Septembre 1999



Localisation par odometrie

INRIA Rhone-Alpes

Moyens robotiques

Du 1 juillet au 15 septembre 1999

Table des matières

Introduction	3
I Présentation	4
1 L'INRIA Rhône-Alpes	4
1.1 L'UR de Montbonnot	4
1.2 Le service robotique	4
2 Les outils	5
2.1 Le Cycab	5
2.2 ORCCAD	6
2.3 VxWorks	6
II Déroulement du stage et travail effectué	7
3 Validation du gyroscope	7
3.1 Introduction	7
3.2 Les tests	7
3.2.1 Description du "banc de mesure"	7
3.2.2 Traitements sur les données	7
3.2.3 Types d'essais	8
3.3 Résultats	8
3.3.1 Comportement général	8
3.3.2 Caractérisation de la zone linéaire	8
3.4 Conclusion	13
4 Le Cycab	13
4.1 Le modèle cinématique du Cycab	13
4.2 La commande séparée des 4 roues	15
4.2.1 les calculs	15
4.2.2 Implémentation d'une procédure ORCCAD	16
5 L'odométrie	20
5.1 Les équations	20
5.2 La localisation par filtrage de Kalman	21
5.2.1 Le principe	21
5.2.2 Application	22
5.2.3 Résultats	24
5.3 Localisation par filtrage de Kalman (suite)	25
6 L'interface utilisateur	26
6.1 Structure du programme	26
6.2 Résultat et problèmes connus	26

III Impressions personnelles	27
7 Intérêts pratiques	27
8 Intérêts personnels	27

Table des figures

1	Vue éclatée du Cycab	5
2	essai en saturation	9
3	essai axes alignés	10
4	essai axes à 45 degrés	11
5	essai axes perpendiculaires	12
6	schema	13
7	modèle du Cycab	14
8	Creation interactive d'un module algorithmique	17
9	Spécification d'une tâche robot	18
10	Spécification d'une procédure robot	19
11	Modele odometrique	20
12	Algorithme du filtre de Kalman	23
13	Critère de Janteau	25

Introduction

Dans le cadre de mon stage de deuxième année, j'ai passé deux mois et demi à l'INRIA, l'Institut Nationale de Recherche en Informatique et Automatique dans l'UR de Montbonnot du 1 juillet au 15 septembre. Durant cette période, j'ai travaillé au sein du service robotique (service qui regroupe les équipements de différents projets de recherche) sur le cycab, un véhicule électrique automatisé qui sera décrit plus en détail ultérieurement. J'ai d'abord choisi ce stage car il me permettait de découvrir un domaine qui m'attire : la robotique, et ensuite parce qu'ainsi je pouvais de l'intérieur découvrir le monde de la recherche. Dans ce document je présenterai brièvement l'INRIA Rhône-Alpes et son service robotique ; puis je décrirai le déroulement du stage et les travaux que j'ai effectué durant ces deux mois et demi ; enfin dans une dernière partie on trouvera une conclusion personnelle sur ce stage et ce que j'ai pu en retirer.

Première partie

Présentation

1 L'INRIA Rhône-Alpes

1.1 L'UR de Montbonnot

L'INRIA Rhône-Alpes est la plus récente des 5 UR de l'INRIA. Créée en décembre 1992 à Montbonnot dans l'agglomération Grenobloise, elle accueille aujourd'hui 8 projets de recherche en partenariat avec des établissements comme l'IMAG, le CNRS ou des entreprises privées (Bull, HP, ...). Les domaines de recherche concernent 4 grands thèmes :

- Informatique communicante : réseaux, télécommunications, internet, systèmes et applications réparties, systèmes temps réels ;
- Machines intelligentes : imagerie et vision, réalités virtuelles, véhicules autonomes, robot bipède ;
- calcul parallèle : système d'exploitation et calcul intensif ;
- modélisations mathématiques et statistiques.

En dehors de ses projets, l'INRIA est engagé dans des actions de développement comme PRAXITELE (projet de transport public individuel basé sur des voitures électriques) qui est arrêté aujourd'hui et des expérimentations à caractère régional :

- dans le domaine des autoroutes de l'information, l'internet et les entreprises communicantes ;
- autour des bibliothèques virtuelles.

Les activités de recherche et de développement s'appuient sur différents services dont le service robotique.

1.2 Le service robotique

Le service robotique au sein duquel j'ai effectué mon stage regroupe tout le matériel nécessaire aux différents projets tel qu'un robot 6 axes RX90, un robot bipède ou encore une voiture électrique. Le but est de fournir un ensemble d'outils performants pour la validation des travaux de recherche. De plus 4 ingénieurs de recherche sont en poste afin d'implémenter ces travaux sur les robots. Ils travaillent en collaboration avec les chercheurs afin de mettre au point des applications de démonstration.

2 Les outils

2.1 Le Cycab

Dans le cadre de mon stage, j'ai travaillé sur le Cycab. Il s'agit d'un petit véhicule électrique automatisé devant servir de plate-forme expérimentale pour les équipes de robotique et vision de l'unité de recherche. Il a été construit sur la base d'un châssis tubulaire Andruet accourci qui supporte une coque conçue par Sylvain Fauconnier et dont les moules appartiennent à l'INRIA. Les caractéristiques générales du véhicule sont les suivantes :

- Longueur hors-tout : 1,90 m
- Largeur hors-tout : 1,20 m
- Poids total avec batteries : 300 kg
- motorisation : 4 moteurs de 1 kW
- 4 roues motrices et directrices
- vitesse théorique maximale 20 km/h
- autonomie : 2 heures d'utilisation continue
- capacité d'accueil : 2 personnes avec bagages
- conduite automatique ou manuelle
- recharge automatique par induction

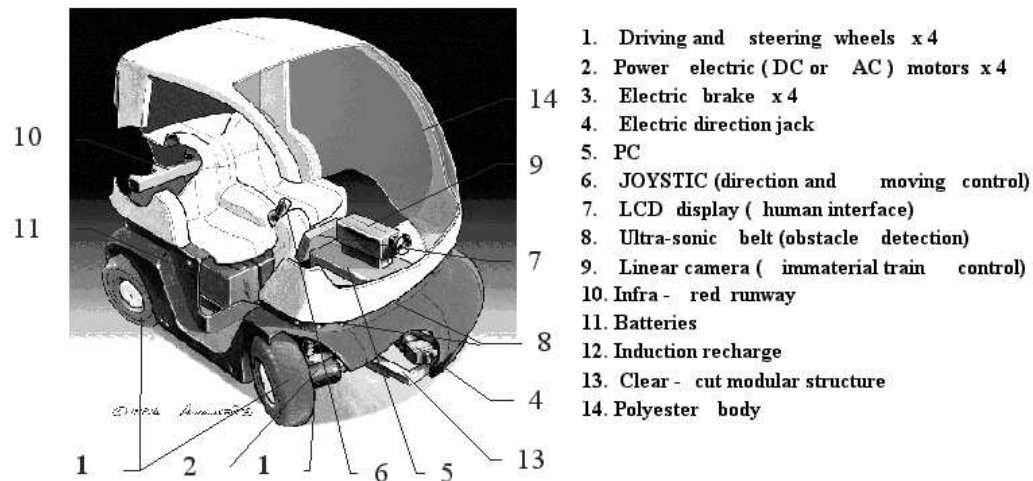


FIG. 1 – *Vue éclatée du Cycab*

Pour contrôler l'ensemble des 9 moteurs (4 de traction, 1 de direction et 4 de frein) une architecture matérielle distribuée a été choisie. Elle est

constituée de 3 noeuds intelligents pouvant gérer chacun 2 moteurs : le premier s'occupe du contrôle du vérin de direction et du joystick de pilotage manuelle, deux autres, un par train, gèrent les 4 moteurs de traction tandis que la commande de frein est unique. Un PC industriel et une carte Motorola MVME162 sous VxWorks complètent cette architecture. Les trois noeuds, le PC et la carte Motorola communiquent entre eux via un bus de terrain Controller Area Network (CAN). Sur ce bus sont connectés les capteurs ultrasonores et la caméra linéaire. Le Cycab peut être piloté manuellement grâce à un joystick ou bien il peut effectuer certaines tâches seul comme poursuivre une cible ou se garer. Les contraintes temps-réel étant très fortes, les tâches aussi bien la conduite manuelle qu'automatique sont programmées à l'aide du logiciel ORCCAD©.

2.2 ORCCAD

ORCCAD (Open Robot Controller Computer Aided Design) est un environnement pour la spécification, la validation par des méthodes formelles et par simulation, et d'implémentation systématique pour les applications robotiques, utilisant l'approche synchrone appropriée pour les systèmes réactifs comme les robots. La philosophie d'ORCCAD est la définition formelle d'une action robotique. Ceci est basé sur les trois principes suivants :

- la plupart des actions physiques accomplies par un robot peuvent être ramenées à des problèmes de contrôle-commande et peuvent être résolues en temps-réel ;
- l'action physique n'est pas suffisante pour définir complètement une action robotique : les instants de début et de fin doivent être considérés comme des réactions à des événements significatifs observés durant l'exécution de la tâche ;
- pour une exécution temps-réel sûre, un intérêt particulier sera donné à la spécification et à la validation.

Ces principes mènent le concepteur à définir des entités simples qu'il compose ensuite pour spécifier des tâches complexes. Un cas sera développé plus loin.

2.3 VxWorks

VxWorks est un OS temps-réel de WindRiver Systems utilisé dans le Cycab. C'est lui qui fait "tourner" les applications et ordonne les tâches. Il a l'avantage d'être très proche d'Unix dont il partage certains mécanismes comme les sockets et les sémaphores par exemple. Cependant il diffère d'Unix par la façon dont sont ordonnancées les tâches : en effet dans une application temps-réel les contraintes temporelles sont très fortes et on recherche un comportement déterministe dans l'enchaînement des tâches, c'est pourquoi l'ordonnanceur de VxWorks n'a pas la même stratégie.

Deuxième partie

Déroulement du stage et travail effectué

Durant le stage, j'ai effectué différents travaux : le premier fut de valider la caractéristique d'un gyroscope à résonance avant de le monter sur le Cycab ; puis j'ai travaillé sur l'odométrie du Cycab en collaboration avec un chercheur ; et enfin j'ai écrit une interface en Java pour contrôler le Cycab à distance.

3 Validation du gyroscope

3.1 Introduction

Afin d'améliorer l'odométrie du Cycab (processus qui consiste à calculer la configuration du véhicule dans le plan par rapport à une position initiale) qui est assez sensible et peu précise dans le calcul de l'angle θ , une idée consiste à utiliser un gyroscope. Or nous disposons d'un gyroscope mais avec une documentation technique correspondant à un modèle plus récent du même constructeur. Aussi pour l'exploiter correctement, il fallait vérifier que la caractéristique $V_{out} = f(\dot{\theta})$ dont nous disposions était valable. Nous avons donc procédé à une série de test décrits ci-dessous.

3.2 Les tests

3.2.1 Description du "banc de mesure"

Le gyroscope était fixé sur un support solidaire d'un axe de rotation. Un codeur incrémental permet de lire la position angulaire de l'axe et d'en déduire la vitesse de rotation. Pour faire les acquisitions, un programme C tournant sous VxWorks a été adapté; il permet de réaliser l'acquisition de la sortie du gyroscope et du codeur tout les $\frac{1}{500}$ de seconde.

3.2.2 Traitements sur les données

Les données brutes sont la sortie du gyroscope (en Volts) et les nombre de tops codeur depuis le début des mesures. Pour en déduire la vitesse angulaire et tracer la caractéristique, on applique différents traitements:

- La vitesse angulaire est calculée par une approximation classique de la dérivée :

$$\dot{\theta}_k = \frac{NbCodeur_k - NbCodeur_{k-1}}{\Delta t} resCodeur$$

- Les couples vitesse/tension sont triés par ordre croissant pour le tracé;
- Une approximation de la caractéristique par la méthode des moindres carrés est calculée ;

- Les points qui s'éloignent trop de cette caractéristique sont supprimés (on a isolé une source importante de bruit due aux variateurs qui ne sont mis sous tension que pour pouvoir utiliser les codeurs) ;
- Une deuxième approximation est calculée, on vérifie alors que les points supprimés sont non significatifs : la pente ne change pas et l'erreur diminue fortement.
- L'ensemble des données est copié dans des fichiers et tracé avec Xprism.

3.2.3 Types d'essais

Quatre types d'essais ont été réalisés :

- Axe du gyroscope et axe de rotation alignés, essai en saturation.
- Axe du gyroscope et axe de rotation alignés, essai dans la zone linéaire.
- Axe du gyroscope et axe de rotation faisant un angle d'environ 45 degrés.
- Axe du gyroscope et axe de rotation perpendiculaires.

3.3 Résultats

3.3.1 Comportement général

Le gyroscope a un comportement linéaire dans une plage de vitesse angulaire donnée : approximativement -80 à +80 deg/sec. Au-delà, il sature comme le montre l'essai 1 :

3.3.2 Caractérisation de la zone linéaire

Avec cet essai, on essaie de caractériser la zone de fonctionnement en régime linéaire du gyroscope ; on cherche en particulier à déterminer la pente ou facteur d'échelle de la caractéristique et les variations de celle-ci en fonction de l'angle entre l'axe du gyroscope et l'axe de rotation. Pour cela trois essais ont été réalisés :

- Axes alignés.
- Axes à 45 degrés.
- Axes perpendiculaires.

Essai axes alignés :

Avec cet essai, nous avons retrouvés la pente de la caractéristique en régime linéaire dans une configuration normale de fonctionnement. Les résultats obtenus avec les moindres carrés sont les suivants (essai avec 2000 acquisitions) :

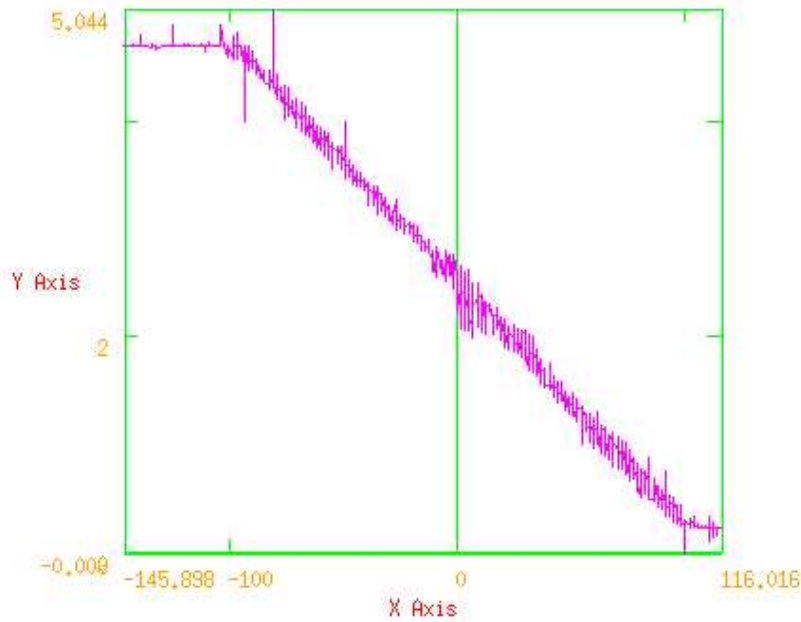


FIG. 2 – *essai en saturation*

segment de:	-73.827	4.168
a:	68.553	0.969
barycentre:	2.550	2.452
direction:	0.999748	-0.022462
erreur:	13.486	

et après remplacement de 21 point(s) sur 1998:

segment de:	-73.827	4.172
a:	68.553	0.968
barycentre:	2.550	2.453
direction:	0.999747	-0.022498
erreur:	7.614	

On trouve donc une pente de -22.4 mV/deg/s et un offset de 2.5 V . Le signe - n'est pas correcte mais il s'agit d'un simple problème de convention: une tension positive correspond à une rotation dans le sens des aiguilles d'une montre par rapport à l'axe du gyroscope (une marque est nécessaire).

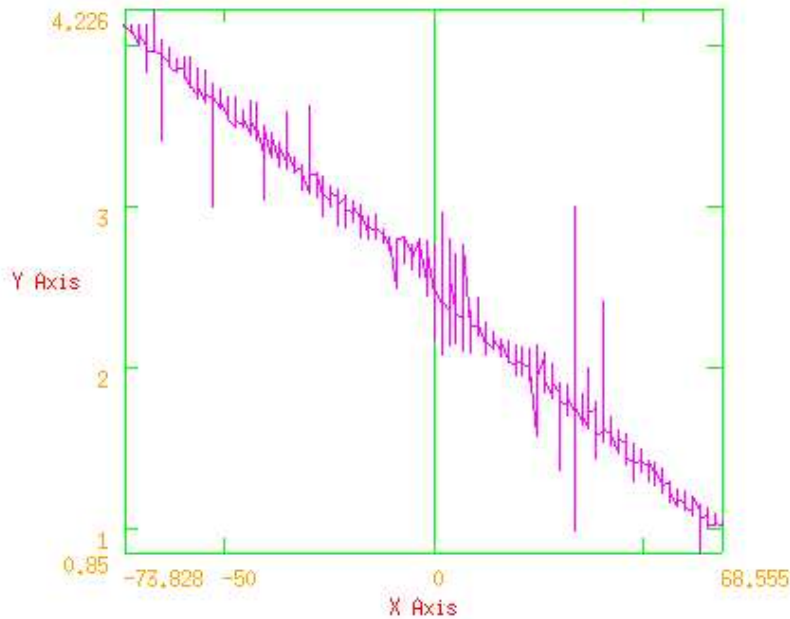


FIG. 3 – *essai axes alignés*

Essai axes à approximativement 45 degrés :

Le but de cet essai est de déterminer l'influence de l'angle entre l'axe du gyroscope et l'axe de rotation, en particulier sur la pente de la caractéristique. Les résultats obtenus sont les suivants(essai effectué avec 2500 acquisitions):

segment de:	-82.616	4.056
a:	58.007	1.420
barycentre:	-4.867	2.599
direction:	0.999824	-0.018742
erreur:	9.957	

et après remplacement de 20 point(s) sur 2498:

segment de:	-82.616	4.054
a:	58.007	1.420
barycentre:	-4.867	2.598
direction:	0.999825	-0.018724
erreur:	4.460	

On observe une variation de la pente qui vaut maintenant -18.7 mV/deg/s .
 Ce qui correspond à une multiplication par $\cos 35$.

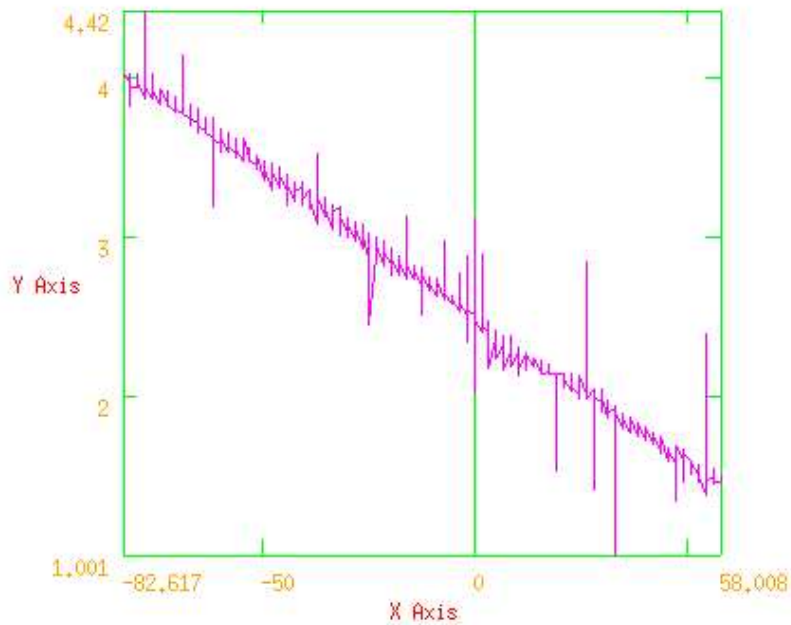


FIG. 4 – *essai axes à 45 degrés*

Essai Axes perpendiculaires:

Les résultats :

segment de:	-108.984	2.562
a:	96.680	2.462
barycentre:	1.562	2.508
direction:	1.000000	-0.000488
erreur:	2.320	

et après remplacement de 6 point(s) sur 1998:

segment de:	-108.984	2.562
a:	96.680	2.462
barycentre:	1.562	2.508
direction:	1.000000	-0.000488
erreur:	0.543	

On observe que la pente est quasiment nulle (les axes ne sont pas tout à fait perpendiculaires) ; cette situation confirme les hypothèses concernant la caractéristique dans la zone linéaire.

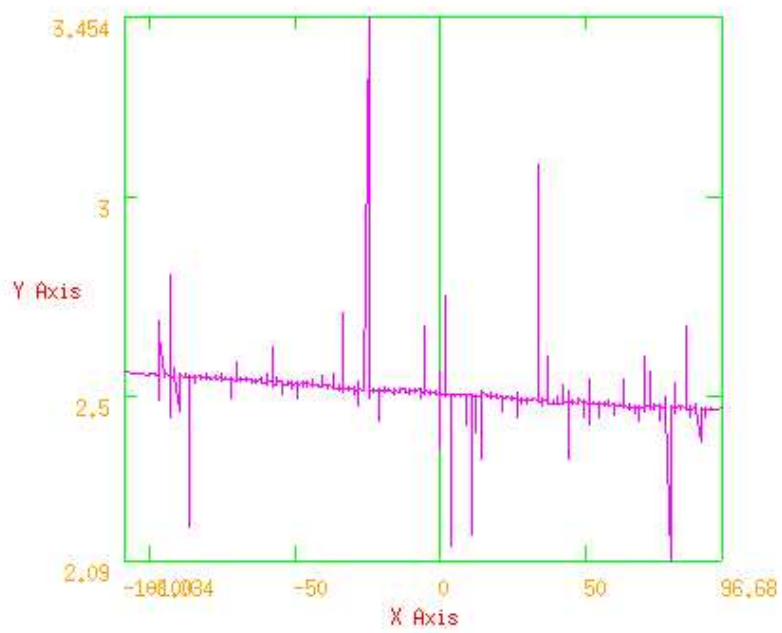


FIG. 5 – *essai axes perpendiculaires*

3.4 Conclusion

On a donc caractérisé le gyroscope avec les essais effectués. La caractéristique dans la zone linéaire peut s'écrire: $V_{out} = V_0 + 0.224 * \cos \theta * \omega$.

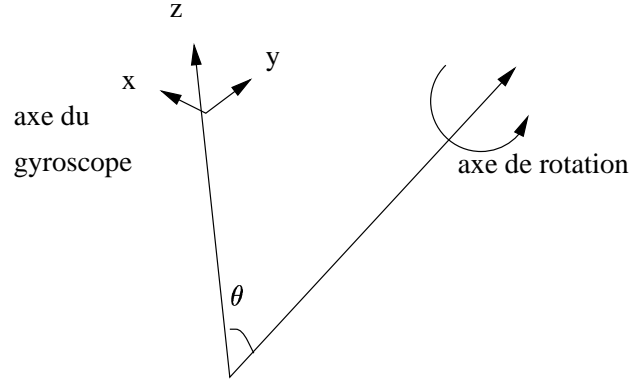


FIG. 6 – *schema*

4 Le Cycab

Une fois le gyroscope validé, l'étape suivante consiste à l'intégrer dans des applications dont l'odométrie. Nous avons donc travaillé sur le Cycab et en particulier sur la commande manuelle et l'odométrie du véhicule. Dans cette section, nous allons voir en détail le modèle cinématique du véhicule, puis la commande séparée des 4 roues (on détaillera ici l'utilisation d'ORCCAD) et enfin on étudiera les équations de l'odométrie et un moyen utilisé pour tenter d'améliorer la précision de celle-ci.

4.1 Le modèle cinématique du Cycab

Le modèle du Cycab est présenté sur la figure suivante. On calcule alors le modèle cinématique; un calcul géométrique (on ne prend pas en compte le signe des angles) donne:

$$RH = \frac{l_w}{1 + \frac{\tan(\phi)}{\tan(k\phi)}} = l_w \frac{\cos(\phi) * \sin(k\phi)}{\sin(\phi + k\phi)}$$

et

$$HF = \frac{l_w}{1 + \frac{\tan(k\phi)}{\tan(\phi)}} = l_w \frac{\cos(k\phi) * \sin(\phi)}{\sin(\phi + k\phi)}$$

Les rayons de braquages ρ_R et ρ_F sont donc:

$$\rho_R = \frac{RH}{|\sin(k\phi)|} = l_w \frac{\cos(\phi)}{|\sin(\phi + k\phi)|}$$

D'autre part dans le repère inertiel, les vitesses de R et de F s'écrivent en tenant compte des angles orientés :

$$\begin{cases} \dot{x}_R &= v_R * \cos(\theta - k\phi) \\ &= v_F * \frac{\cos(\phi)}{\cos(k\phi)} * \cos(\theta - k\phi) \\ \dot{y}_R &= v_R * \sin(\theta - k\phi) \\ &= v_F * \frac{\cos(\phi)}{\cos(k\phi)} * \sin(\theta - k\phi) \end{cases}$$

$$\begin{cases} \dot{x}_F &= v_F * \cos(\theta + \phi) \\ \dot{y}_F &= v_F * \sin(\theta + \phi) \end{cases}$$

D'où :

$$\begin{pmatrix} v_F * \left(\cos(\theta + \phi) - \frac{\cos(\phi)}{\cos(k\phi)} * \cos(\theta - k\phi) \right) \\ v_F * \left(\sin(\theta + \phi) - \frac{\cos(\phi)}{\cos(k\phi)} * \sin(\theta - k\phi) \right) \\ 0 \end{pmatrix} = \begin{pmatrix} -l_w \dot{\theta} * \cos(\theta) \\ l_w \dot{\theta} * \sin(\theta) \\ 0 \end{pmatrix}$$

On en déduit alors :

$$\dot{\theta} = v_F * \frac{\sin(\phi + k\phi)}{l_w * \cos(k\phi)}$$

On peut ensuite en déduire la vitesse à appliquer aux 4 roues en fonction des consignes de vitesse et de direction fournies par le joystick.

4.2 La commande séparée des 4 roues

Jusqu'à présent les 4 roues reçoivent la même consigne de vitesse à tout instant. Cette situation entraîne des glissements importants qui ne sont pas pris en compte dans l'odométrie. Calculer la vitesse de chaque roue suivant le rayon de braquage théorique du véhicule et la consigne de vitesse permet de diminuer les glissements et à améliorer la précision de l'odométrie.

4.2.1 les calculs

La conduite manuelle se fait à partir d'un joystick qui fournit une consigne de vitesse de rotation à la roue virtuelle avant soit ψ_F , et une consigne de direction ϕ dont le signe positif est défini par le sens trigonométrique direct (inverse des aiguilles d'une montre) ce qui assure que ϕ et θ sont du même signe. Les consignes sont donc :

$$\psi_F = \frac{v_F}{r} \text{ et } \phi$$

On en déduit directement :

$$\psi_R = \psi_F * \frac{\cos(\phi)}{\cos(k\phi)}$$

Les sorties sont les vitesses de rotation des 4 roues que nous allons calculer. Dans le repère lié au cycab, on a :

$$\overrightarrow{v_{RL}} = \overrightarrow{v_R} + \overrightarrow{L_R R} \wedge \overrightarrow{\Omega}$$

où

$$\overrightarrow{L_R R} = \begin{pmatrix} 0 \\ -\frac{L}{2} \\ 0 \end{pmatrix} \text{ et } \overrightarrow{v_R} = \begin{pmatrix} v_R * \cos(-k\phi) \\ v_R * \sin(-k\phi) \\ 0 \end{pmatrix}$$

D'où :

$$\overrightarrow{v_{RL}} = \begin{pmatrix} v_R * \cos(-k\phi) - \frac{L\dot{\theta}}{2} \\ v_R * \sin(-k\phi) \\ 0 \end{pmatrix}$$

et

$$v_{RL} = \sqrt{v_R^2 + \left(\frac{L\dot{\theta}}{2}\right)^2 - L\dot{\theta} * v_R * \cos(k\phi)}$$

D'où :

$$|\psi_{RL}| = \sqrt{\psi_R^2 + \left(\frac{L\dot{\theta}}{2r}\right)^2 - \frac{L\dot{\theta}}{r} * \psi_R * \cos(k\phi)}$$

De même :

$$v_{RR} = \sqrt{v_R^2 + \left(\frac{L\dot{\theta}}{2}\right)^2 + L\dot{\theta} * v_R * \cos(k\phi)}$$

et :

$$|\psi_{RR}| = \sqrt{\psi_R^2 + \left(\frac{L\dot{\theta}}{2r}\right)^2 + \frac{L\dot{\theta}}{r} * \psi_R * \cos(k\phi)}$$

En remplaçant v_R , ψ_R , $\cos(k\phi)$ respectivement par v_F , ψ_F , $\cos(\phi)$ dans les formules de ψ_{RL} et de ψ_{RR} on obtient les formules de ψ_{FL} et de ψ_{FR} . Le signe des vitesses de chaque roue est déterminé par le signe de la consigne de vitesse : + pour la marche avant, - pour la marche arrière.

Resultats :

La commande des quatre roues séparées permet une meilleure conduite et améliore sensiblement l'odométrie ; en effet les roues sont asservies en vitesse ainsi les erreurs de modélisation (non-unicité du centre de rotation) se traduisent par du glissement et donc par des déplacements non pris en compte dans l'odométrie, une commande unique pour les quatres roues entraîne un glissement important qu'une commande séparée permet de réduire.

4.2.2 Implémentation d'une procédure ORCCAD

Maintenant nous allons implémenter une procédure robot de conduite manuelle utilisant notre loi de commande à l'aide d'ORCCAD. Pour cela il faut créer un module algorithmique avec ses ports d'entrée/sortie et éditer ses propriétés. Sur la capture d'écran suivante on remarque les différentes propriétés d'un module algorithmique et d'un port. Pour le module, il faut définir le nom puis différents fichiers qui serviront ensuite à générer le code :

- def.h : précise les fichiers d'en-tête sont nécessaires ;
- var.c : dans ce fichier on déclare les variables comme on le ferait dans un fichier source en C ;

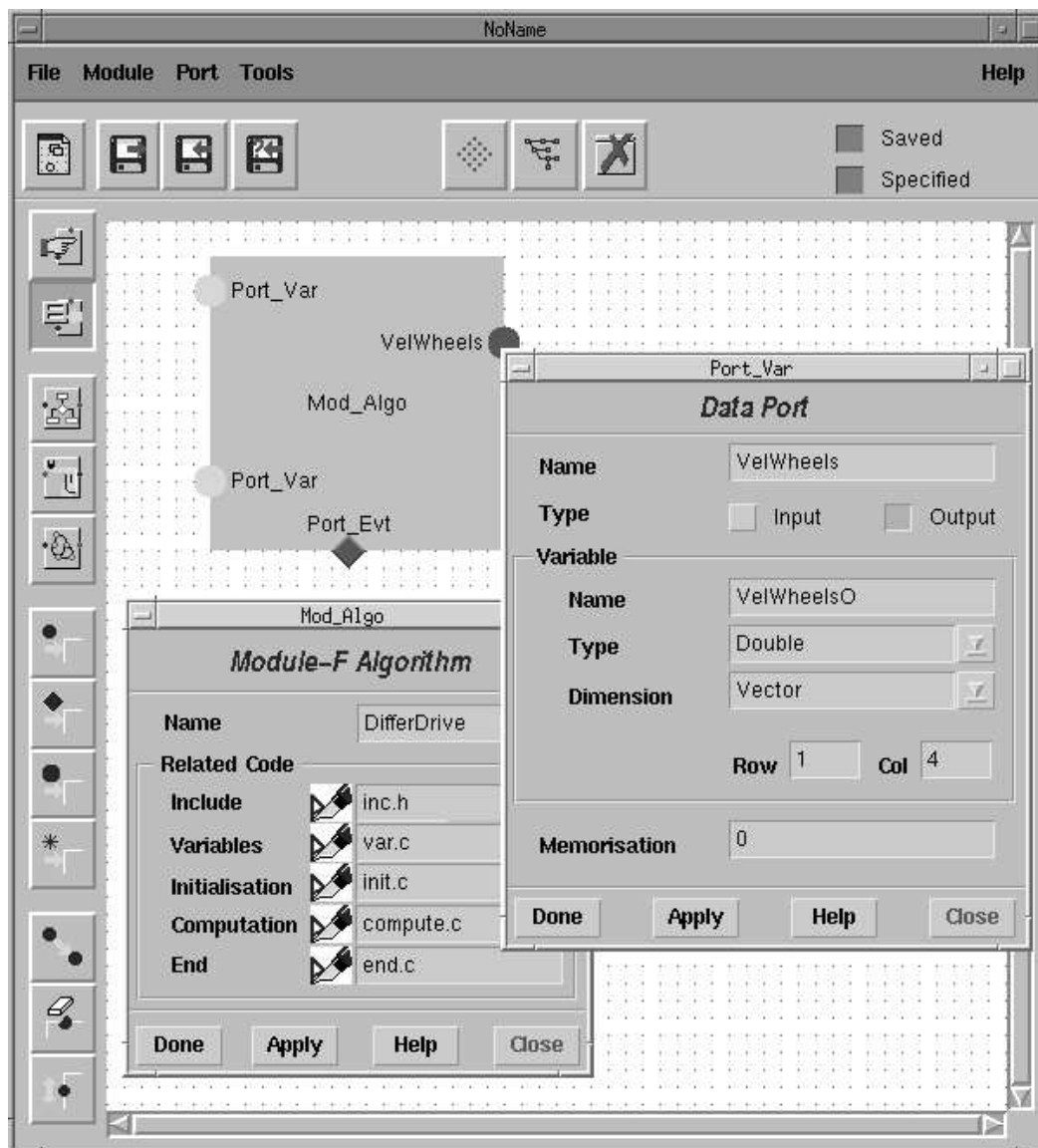


FIG. 8 – *Creation interactive d'un module algorithmique*

- init.c : initialisation des variables précédemment déclarées ;
- compute.c : dans ce fichier se trouve le code de calcul de l'algorithme : dans notre cas le calcul des vitesses angulaires des quatres roues en fonction des consignes.
- end.c : code à exécuter à la fin de la tâche.

Pour les ports il faut préciser si il s'agit d'une entrée ou d'une sortie, le nom, le nom de la variable associée, le type de la donnée (float, double, ...) et son format (array, vector ou scalar) et si nécessaire la taille. Nous allons maintenant spécifier une tâche robot. Sous ORCCAD une tâche robot est "la spécification complète et paramétrable d'une action élémentaire d'asservissement, c'est à dire de l'activation d'une loi de commande de structure invariante sur la durée de la tâche robot, et du comportement logique associé à un ensemble de signaux concernant l'action élémentaire d'asservissement et susceptible d'être émis préalablement à et pendant son exécution." La loi de commande a été spécifiée précédemment ; le comportement logique est implémenté dans un module spécifique : un automate à états finis qui réagit aux signaux logiques en considérant des pré- et post-conditions et en générant éventuellement des exceptions. Les modules employés sont ceux de la biblio-

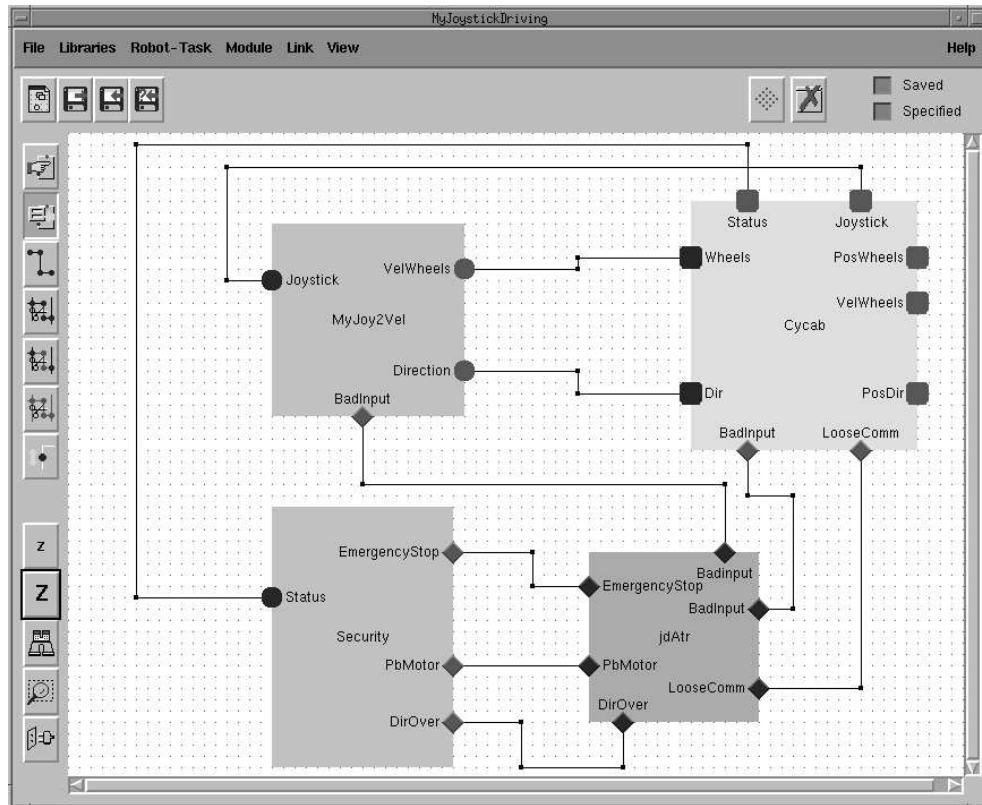


FIG. 9 – Spécification d'une tâche robot

thèques systèmes. Nous allons maintenant pouvoir spécifier une “procédure robot”. Une procédure robot est la spécification complète et paramétrée :

- d’un programme principal défini comme un arrangement logique et temporel de tâches robot (TRs) ou de procédures robot (PRs).
- d’un ensemble de triplet (événement, traitement, assertions) permettant de définir les conditions dans lesquelles enchaîner les TRs ou les PRs précédentes ;
- d’un comportement logique rythmé par des événements.

Notre procédure ne se compose que de la TR que l’on vient de définir. Pour cela il suffit de sélectionner la TR et ORCCAD génère automatiquement le code Esterel de la procédure. Il ne reste plus qu’à générer le code de



```
Robot-Procedure Editor
FILE TOOLS Ctrl HELP
Task Proc
Main program ProcMyJoy
[
% Component MyJoystickDriving
[
% call MyJoystickDriving_fileparameter()(?MyJoystickDriving_Start, "filename");
call MyJoystickDriving_parameter()(?MyJoystickDriving_Start);
call MyJoystickDriving_controler()(?MyJoystickDriving_Start);
emit START_MyJoystickDriving;
await
  case BF_MyJoystickDriving
  case T2_DirOverE do emit L_T2_DirOverE
  case T3_MyJoystickDriving do exit T3_controler_ProcMyJoy
end await
]
]
]
Conditions
```

FIG. 10 – Spécification d’une procédure robot

notre procédure et à le compiler pour la cible : le Cycab. Il faut ensuite ce programme dans le Cycab et le lancer grâce à un shell VxWorks.

5 L'odométrie

L'odométrie est un procédé de localisation qui permet à un robot mobile de connaître sa position dans l'espace des configurations $(x \ y \ \theta)$ par rapport à un repère du monde donné ; dans le cas d'un robot de type voiture on mesure son déplacement en intégrant la position angulaire des roues arrières et en effectuant quelques calculs. La section ci-dessous explique plus précisément les calculs à effectuer.

5.1 Les équations

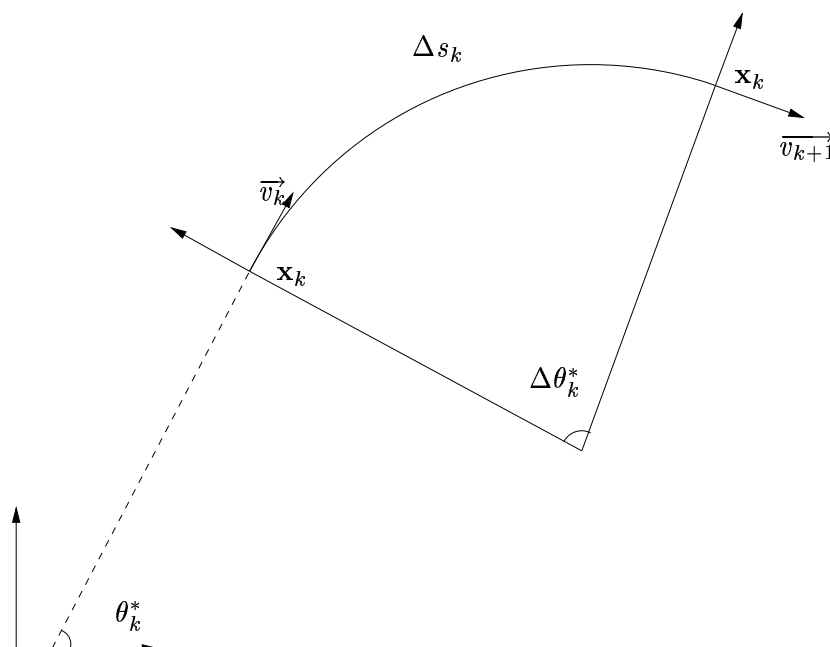


FIG. 11 – *Modele odometrique*

Si on dispose d'encodeurs pour mesurer les positions angulaires des roues arrières, le principe est alors le suivant : à l'instant k , la voiture repérée par le milieu de l'essieu arrière se trouve dans la configuration (x_k, y_k, θ_k^*) dans le repère du monde. Entre l'instant k et l'instant $k+1$, très proche, on assimile la trajectoire de la voiture à un arc de cercle de longueur Δs_k et faisant un angle $\Delta \theta_k^*$ donnés par :

$$\Delta s_k = R \frac{\Delta \theta_{d_k} + \Delta \theta_{g_k}}{2}$$

$$\Delta \theta_k^* = R \frac{\Delta \theta_{d_k} - \Delta \theta_{g_k}}{E}$$

où R est le rayon des roues, E l'empattement (longueur de l'essieu) et $\Delta \theta_{d_k}$, $\Delta \theta_{g_k}$ sont les variations des positions angulaires des roues droite et gauche

entre les instants k et $k+1$. On en déduit alors la configuration à l'instant $k+1$ à l'aide des équations suivantes :

$$x_{k+1} = x_k + \Delta s_k \cos(\theta_k^* + \Delta\theta_k^*/2)$$

$$y_{k+1} = y_k + \Delta s_k \sin(\theta_k^* + \Delta\theta_k^*/2)$$

$$\theta_{k+1}^* = \theta_k^* + \Delta\theta_k^*$$

Ces équations sont obtenues en calculant la position de la voiture à l'instant $k+1$ par rapport au repère de l'instant k puis en la ramenant dans le repère du monde et en linéarisant les équations obtenues pour de petites valeurs de $\Delta\theta_k^*$. Ce modèle convient donc pour un véhicule lent et à roues arrières non-directionnelles ; dans le cas du cycab dont les roues arrières ont un braquage proportionnel au braquage avant on ajoute à l'angle θ du véhicule l'angle $-k\phi$ qui correspond au braquage des roues arrières. D'autre part, le procédé est très sensible aux erreurs de mesure d'une part du fait des intégrations successives et à l'état du sol sur lequel évolue le véhicule d'autre part (glissements, petits obstacles, ...). Les résultats sont donc médiocres surtout pour le calcul de l'orientation du véhicule et sont trop peu précis pour les applications envisagées (navigation autonome). Un moyen d'améliorer le procédé consiste à utiliser des capteurs extéroceptifs et à effectuer un filtrage de Kalman. En effet, le filtre de Kalman permet de prendre en compte les incertitudes et les erreurs introduites par les imprécisions du modèle et les phénomènes aléatoires (glissements, ...) en modélisant l'ensemble par des bruits blancs ; d'autre part il permet de fusionner les données odométriques avec d'autres données mesurables (gyroscope, ...). Dans la section qui suit, nous allons détaillé le procédé mis en oeuvre pour tenter d'améliorer l'odométrie à l'aide du gyroscope.

5.2 La localisation par filtrage de Kalman

5.2.1 Le principe

Le filtre de Kalman est un algorithme incrémental permettant une estimation optimale des systèmes linéaires en tenant compte des imprécisions de mesures successives (considérées comme un bruit aléatoire). Le filtre de Kalman étendu permet d'estimer des systèmes dont les équations sont non-linéaires. Soit un système dont les équations d'état et de mesure sont les suivants :

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{w}_k$$

où :

- \mathbf{x}_k représente l'état du système à l'instant k
- \mathbf{u}_k représente un vecteur de commande
- \mathbf{z}_k est le vecteur d'observation

- \mathbf{v}_k est le bruit du modèle
- \mathbf{w}_k est le bruit de mesure

Le principe est le suivant : à l'instant k , on connaît une estimation de l'état du système $\hat{\mathbf{x}}_{k/k-1}$ et de la matrice de variance-covariance de l'erreur d'estimation $\mathbf{P}_{k/k-1}$, on calcule alors la matrice de gain de Kalman \mathbf{K}_k suivant l'équation :

$$\mathbf{K}_k = \mathbf{P}_{k/k-1} \mathbf{H}_k^T \mathbf{S}_{k/k-1}^{-1}$$

où $\mathbf{S}_{k/k-1} = \mathbf{H}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^T + \mathbf{R}_k$ et \mathbf{R}_k est la matrice de variance-covariance du bruit $\hat{\mathbf{w}}_{k+1}$ et où \mathbf{H}_k est le jacobien de h selon \mathbf{x}_k en $\hat{\mathbf{x}}_{k/k}$. on effectue alors une mise à jour de l'estimation :

$$\hat{\mathbf{x}}_{k/k} = \hat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

où \mathbf{z}_k est la mesure effective et $\hat{\mathbf{z}}_k = h(\hat{\mathbf{x}}_{k/k-1})$ la prédiction de la mesure. Puis on effectue une mise à jour de la matrice de variance-covariance de l'erreur d'estimation :

$$\mathbf{P}_{k/k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k/k-1}$$

On passe alors à la deuxième étape qui consiste à prédire une estimation de l'état du système et de la matrice de variance-covariance de l'erreur d'estimation pour l'étape suivante :

$$\hat{\mathbf{x}}_{k+1/k} = f(\hat{\mathbf{x}}_{k/k}, \mathbf{u}_k)$$

La matrice de variance-covariance de l'erreur d'estimation $\mathbf{P}_{k+1/k}$ associée à la prédiction est donnée par :

$$\mathbf{P}_{k+1/k} = \mathbf{F}_k \mathbf{P}_{k/k} \mathbf{F}_k^T + \mathbf{Q}_k$$

où \mathbf{F}_k est le jacobien de f selon \mathbf{x}_k en $\hat{\mathbf{x}}_{k/k}$, \mathbf{Q}_k est la matrice de variance-covariance du bruit \mathbf{v}_k .

5.2.2 Application

L'utilisation du filtre de Kalman que nous avons faite vise non seulement à essayer d'améliorer la localisation par odométrie en filtrant proprement dit les données odométriques mais aussi à fusionner celles-ci avec les données fournies par le gyroscope. Ainsi dans le cas qui nous intéresse, le vecteur d'état est $(x \ y \ \theta \ \dot{\theta})^T$. Les équations du modèle sont celles de l'odométrie auxquelles on rajoute celle concernant $\dot{\theta}$ donné par le gyroscope :

$$\dot{\theta}_{k+1} = \dot{\theta}_k + \int_{t_k}^{t_{k+1}} \ddot{\theta} dt$$

On a donc une matrice \mathbf{F}_k :

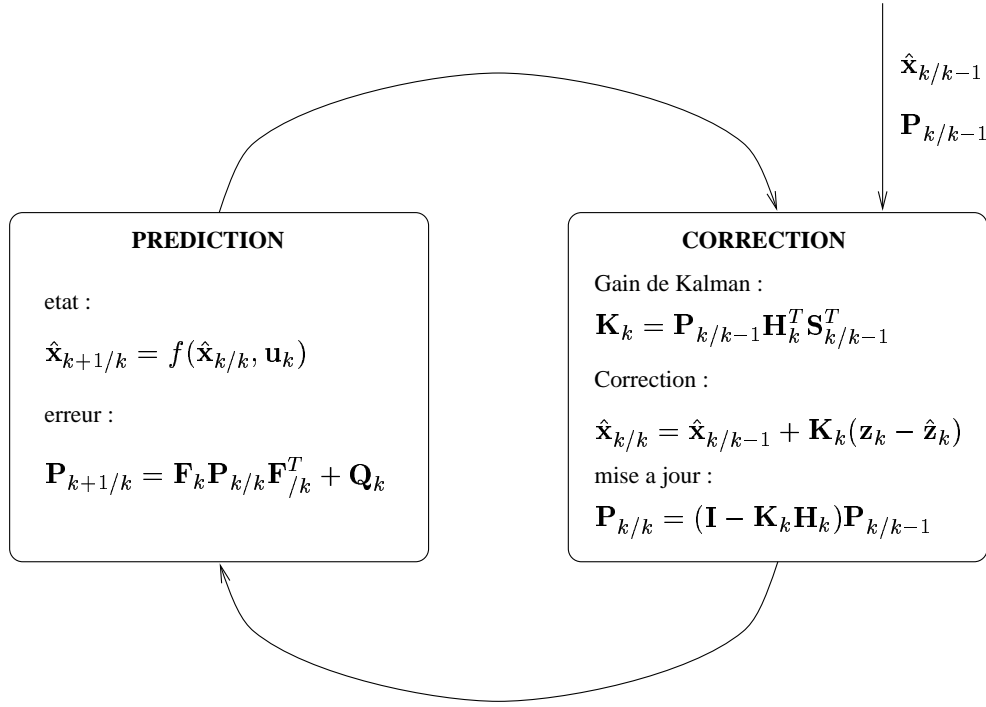


FIG. 12 – *Algorithme du filtre de Kalman*

$$\mathbf{F}_k = \begin{pmatrix} 1 & 0 & 0 & -\Delta s_k \sin(\theta_k + \Delta\theta_k/2) \\ 0 & 1 & 0 & \Delta s_k \cos(\theta_k + \Delta\theta_k/2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

D'autre part, la seule mesure est celle donnée par le gyroscope, donc $\mathbf{H}_k = (0\ 0\ 1\ 0)$ La matrice de variance-covariance du bruit du modèle est calculée à chaque étape en considérant que les différentes composantes sont indépendantes: la matrice est donc diagonale. Les termes diagonaux correspondent donc à la variance de chaque composante. On a :

$$\begin{aligned} \sigma_x^2 &= k_{ss} |\text{delta}S * \cos(\theta_k + \Delta\theta_k/2)| \\ \sigma_y^2 &= k_{ss} |\text{delta}S * \sin(\theta_k + \Delta\theta_k/2)| \\ \sigma_\theta^2 &= k_{\theta s} |\text{delta}S| + k_{\theta\theta} |\text{delta}Theta| \end{aligned}$$

ou les constantes k_{ss} , $k_{\theta s}$, $k_{\theta\theta}$ sont exprimées respectivement en m/m, rad/m, rad/rad :

- k_{ss} correspond à la distance entre la position réelle et la position théorique d'arrivée après une distance l parcourue en ligne droite (théoriquement) ;
- $k_{\theta s}$ correspond à l'angle du véhicule après une distance l parcourue en ligne droite (théoriquement) ;

- $k_{\theta\theta}$ correspond a l'angle du vehicule apres un virage theorique de 360 degres.

Le bruit en $\dot{\theta}$ est fixé empiriquement et correspond au bruit de mesure du gyroscope.

5.2.3 Résultats

Cette implémentation du filtre de Kalman n'apporte pas de une nette amélioration de la localisation par rapport à la simple odométrie: en effet on obtient une précision supérieure au décimètre pour des déplacements faibles (environ 50m). D'autre part les résultats sont nettement meilleure en ligne droite que lorsqu'on décrit des cercles: en ligne droite la précision est identique à l'odométrie tandis qu'en décrivant des cercles la précision chute fortement comme le montrent les resultats suivants :

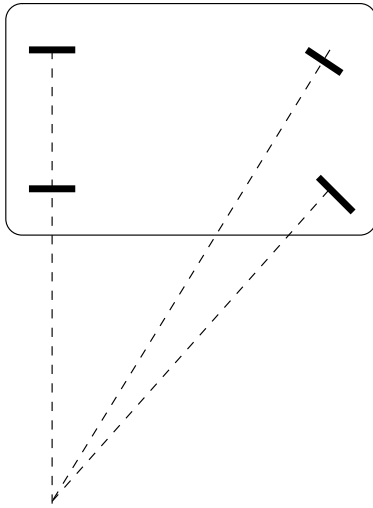
Ligne droite 50m		
	Odometrie	Ecart
X	0.08	<0.05
Y	-0.29	>0.20
θ	3	<5

3 cercles 50m		
	Odometrie	Ecart
X	1.46	env 1.50
Y	0.45	env 0.3
θ	30	<30

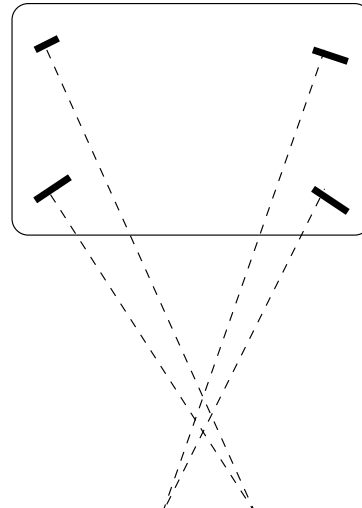
Les résultats sont donnés en mètre et en degré.

La principale source d'erreurs reste les imprécisions du modèle du cycab; en effet le mécanisme de direction du cycab ne respecte pas le critère de Janteau (cf figure) ce qui introduit inévitablement des erreurs systématiques qui, pour l'instant ne sont pas prises en compte (malgré leur origine qui les rends sans doute modélisables). Il serait sans doute intéressant d'essayer de quantifier ces glissements dans un modèle qui tient compte de la conception imparfaite de la direction du cycab et de l'existence des 2 centres de rotation et d'intégrer ces résultats dans le filtre; une autre méthode pourrait être d'intégrer des accéléromètres dans notre implémentation du filtre de Kalman afin de pouvoir quantifier en permanence ces glissements.

D'autre part le filtre est relativement sensible aux variations des paramètres tels que les différentes variances des bruits modélisant les imperfections du modèle et les phénomènes aléatoires. Une autre implémentation dans laquelle la prédiction est faite en utilisant les équations du modèle du cycab et la mesure en utilisant uniquement celles de l'odométrie est en cours de réalisation.



critere de Janteau respecte :
un seul centre de rotation



cas du cycab :
le critere n'est pas respecte, il y a 2
centres de rotation d'ou glissement

FIG. 13 – *Critère de Janteau*

5.3 Localisation par filtrage de Kalman (suite)

Une autre implémentation dans laquelle la prédiction est faite en utilisant les équations du modèle du cycab et la mesure en utilisant celles de l'odométrie est en cours de réalisation. Nous avons essayé d'implémenter une autre forme du filtre de Kalman ne faisant pas intervenir le gyroscope afin de pouvoir "mesurer" l'intérêt de ce dernier. En gardant le même principe général : prédiction à l'aide d'équations du modèle cinématique et mesure par odométrie, nous avons essayé différents "jeux" d'équations pour la phase

6 L'interface utilisateur

Cette interface a pour but de réaliser un tableau de bord pour le cycab qui doit être implanter sur l'écran tactile des futures versions du cycab. L'interface que réalisée affiche 4 témoins : direction (consigne de direction du joystick), cap donné par une boussole, tension de la batterie et vitesse du véhicule (consigne joystick) et permet aussi de piloter le cycab à l'aide de la souris sur un poste distant. A terme, l'interface doit être le centre de contrôle du cycab et doit permettre de lancer interactivement différentes applications (conduite automatique ou manuelle ou autres). Le programme a été écrit en Java et utilise les composants graphiques Swing.

6.1 Structure du programme

La version actuelle du programme tourne sur une station distante et communique avec le cycab par socket. La connexion se fait par des fonctions natives écrites en C. L'acquisition des données des capteurs se fait par timer tous les 200 ms ; de même le contrôle du cycab par la souris utilise un timer qui envoie les consignes tous les 200 ms. Les classes définies sont :

- CycabGui2.java qui définit le panneau principal et son menu ; c'est dans ce panneau que s'affiche tous les panneaux secondaires (les capteurs, le "controlpad",...);
- Capteur.java qui définit l'aspect visuel des panneaux des capteurs ;
- myTimer.java qui est une sous-classe de la classe Timer du jdk ;
- DataTimer.java, une sous-classe de myTimer.java qui permet de lire les données des capteurs sur le cycab ;
- CoordinateArea.java qui définit l'aspect visuel du controlpad et permet de transformer la position de la souris en commande pour le cycab ;
- CmdTimer.java qui est une sous-classe de myTimer.java qui permet d'envoyer les commandes lues sur la souris au cycab.

L'utilisation d'un timer pour l'acquisition des données des capteurs et la commande au lieu de threads s'explique d'une part par la simplicité d'utilisation et d'autre part par le principe de gestion des événements swing qui exige que toutes modifications d'un élément d'une interface soient codées dans la "boucle de gestion des événements". Sun conseille donc d'utiliser les timers pour les actions répétées à intervalle réguliers.

6.2 Résultat et problèmes connus

La version actuelle du programme permet de connaître à tous moments l'état du véhicule ou de le piloter avec la souris à partir d'une station distante mais le programme "plante" si on essaye de réaliser les 2 tâches à la fois. D'autre part la structure du programme (références croisées entre composants graphiques) rends difficile l'intégration de nouvelles fonctionnalités.

Troisième partie

Impressions personnelles

7 Intérêts pratiques

Sur le plan pratique, ce stage a été très bénéfique. En premier lieu j'ai pu acquérir une plus grande maîtrise des outils couramment utilisés sur Unix (Xemacs, xfig, ...) et du système Unix lui-même par une utilisation quotidienne et d'autre part la variété des travaux auxquels j'ai participé m'a permis de mieux connaître les possibilités de l'informatique dans le domaine de l'automatique et de la robotique et d'avoir un aperçu de l'état de l'art dans ce domaine. Enfin durant ces deux mois j'ai eu l'opportunité de travailler avec du matériel peu commun et de voir fonctionner des machines qui ne sont pour l'instant que des prototypes de laboratoire que peu de gens peuvent voir (le cycab, un robot bipède, et d'autres choses encore). Par ailleurs il est apparu clairement qu'une méthodologie rigoureuse est impérative aussi bien dans la phase d'étude que dans la phase de test.

8 Intérêts personnels

Après un stage dans une entreprise en première année, je souhaitais connaître un peu mieux le milieu de la recherche (l'ambiance, le mode de travail et les domaines actuelles de recherche) afin de décider de l'opportunité de poursuivre mes études en DEA. Sur ce point, la question n'est pas résolue et je n'ai pu trancher. Cependant je sais maintenant que je souhaite travailler dans un service de recherche et développement ou l'équivalent ; en effet le travail de recherche n'est pas répétitif et laisse une large part à l'initiative personnelle, ce que j'apprécie beaucoup (les chercheurs sont en effet relativement libres dans leurs choix et beaucoup de voies peuvent être étudiées en parallèle) et malgré les résultats un peu décevants obtenus l'expérience est enrichissante.