ABIA : Environnement de Simulation pour les robots mobiles

A.Moussaoui, C.Kara Terki, B.Cherki

Laboratoire d'automatique, Université Abou Bekr Belkaid, Tlemcen,
B.P. 119, 13000 Tlemcen, Algérie.
Fax (213) 43 272 435
E-Mail : abd_moussaoui@mail.univ-tlemcen.dz

Résumé : *Dans cet article, un environnement de simulation (***ABIA***) pour les robots mobiles est proposé. Cet environnement est composé d'un ensemble d'utilitaires, pour la modélisation, la simulation et l'animation en 3D des robots mobiles (Type véhicule et tricycle).* **ABIA** *permet de commander et de contrôler le comportement d'un Robot mobile. L'utilisateur peut modéliser et configurer le robot et son environnement (obstacles), et visualiser leur comportement suivant certaines critères de commandes.*

1.INTRODUCTION :

L'implantation de robots dans les ateliers a mis en évidence différents problèmes liés à l'utilisation et à la gestion des sites robotisés dont le manipulateur est l'un des composants.

Afin de contribuer à la maîtrise de ces problèmes, différents outils ont été développés. L'objectif de ces outils est d'apporter une aide à l'utilisateur ou au concepteur pour résoudre les différents problèmes qui se posent lors des différentes phases de son travail. Ces problèmes concernent par exemple, le choix du robot en fonction des tâches visées, l'implantation de celui-ci dans son site, les méthodes de programmation de tâches.

Ces différents problèmes peuvent être résolus ou potentiellement traités par l'utilisation de systèmes de CAO robotique.

En effet, ces systèmes offrent de puissants outils graphiques qui permettent de traiter facilement certains des problèmes cités. Ils permettent en outre, au moyen de simulateurs graphiques, de programmer et de simuler les tâches.

Ainsi, les problèmes d'accessibilité de la tâche et d'évitement de collisions peuvent être vérifiés lors de la simulation ce qui permet de réduire la phase de vérification sur le site réel **[1]**.

2. Description générale :

ABIA est un système ouvert et indépendant.

Les principales caractéristiques de conception du simulateur sont:

1) La portabilité: l'environnement de simulation peut être compilé, et exécuté sous différents systèmes d'exploitation, comme Linux, Ms-Windows9X, Ms-WindowsNT.

2) La présentation graphique tridimensionnelle,

3) Calcul Cinématique en temps réel.

4) Interface utilisateur graphique avec OpenGl.

Pour configurer un environnement de simulation, l'utilisateur dispose d'un ensemble d'utilitaires de configuration indépendants, chacun d'eux peut être exécuté séparément.

2.1.Configuration de la plateforme mobile :

Cet utilitaire propose deux types de robots mobiles, véhicule ou tricycle. Dans les deux cas, l'utilisateur peut configurer tous les paramètres du robot mobile.

Robot mobile de type véhicule :
Pour le modèle géométrique et cinématique voir [6].

Robot mobile de type tricycle :
Le modèle cinématique voir [7].

2.2Configuration de l'environnement :

A l'aide d'un modeleur, l'utilisateur peut modéliser n'importe quel

environnement statique. Il dispose d'un ensemble de primitives standards comme les cubes, parallélépipèdes, sphères, cylindres, cônes, tores. Le choix des sources de lumière ainsi que leurs configuration (type, position, direction, couleur …) est possible.

2.3.Paramètres de simulation :

L'ensemble des tâches dans l'environnement de simulation, calcul de la cinématique et de la commande, rendu graphique, détection des collisions, doit être exécuté simultanément avec une fréquence d'exécution constante. L'utilisateur peut choisir la fréquence d'exécution de chaque tâche.

3.Modélisation Géométrique:

La méthode de modélisation polyédrique est la méthode la plus utilisée pour décrire les solides dans les systèmes de CAO robotique **[1]**.

Notre système contient une bibliothèque de modélisation d'environnements polyédriques et de systèmes mécaniques évoluant dans l'espace. Les fonctions de cette bibliothèque permettent de simplifier la représentation des scènes complexes composées d'objets fixes (les obstacles) et mobiles (les robots).

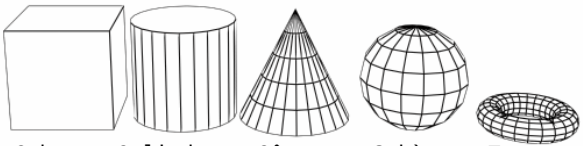

Figure 1. Exemple de Primitives Standards

Les différents objets de la scène sont modélisés sous la forme d'ensembles de polyèdres définis à partir de primitives standards simples (boites, sphères, cylindres, tores...) et de polyèdres quelconques (convexes ou concaves).

4.Détection de Collision :

Etant donnée une configuration de l'objet (ou des objets) mobile, on veut savoir si il est en collision avec son environnement, ou si l'un de ses corps est en collision avec un autre (auto collision).

Un Algorithme fondé sur le calcul de distance est utilisé, il ne traite que les objets convexes, mais présente l'avantage de donner la distance entre les objets rendant possible de prévenir les collisions.

5.Simulation de Capteurs ultrasonores:

Le fonctionnement de ce type de capteurs (ultrasonore) dépend d'un nombre important de paramètres comme la lumière, la vitesse, et la taille, la forme, et la matière des obstacles.

Dans notre travail, nous cherchons à obtenir un modèle permettant de valider une approche ou une autre, pour cette raison nous n'avons pris en compte que la distance entre l'obstacle et le capteur.

6. Architecture de l'environnement :

L'architecture logicielle du système de simulation s'appuie sur *OpenGL* pour effectuer ses opérations de rendu graphique 3D et *SDL* (*Simple DirectMedia Layer*) pour communiquer avec le système d'exploitation

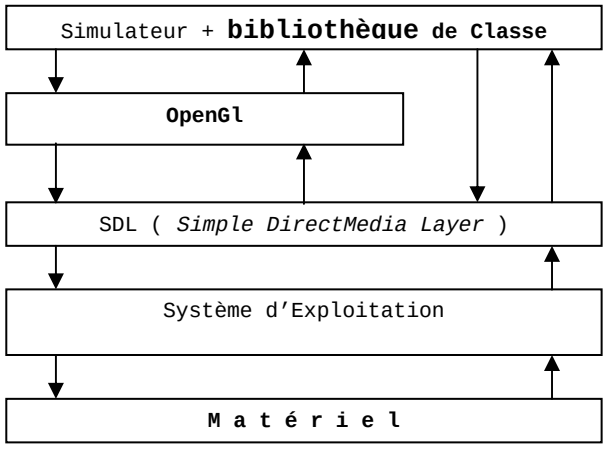

Figure 2. Architecture Logiciel du Système CAO.

(clavier, souris, joystick, Timer, Thread ...).

6.1.OpenGL :

Est une interface de programmation vers le matériel graphique. La programmation avec l'API *OpenGl* est indépendante du matériel graphique utilisé, et son domaine d'utilisation

n'est pas confiné aux PC/Windows ou Mac **[4]**.

A l'origine développé par *Silicon Graphics*, *OpenGL* est aujourd'hui disponible sur les plates-formes x86 et 68000 entre autres, que ce soit sous Windows, Linux ou Mac.

OpenGL se base toujours sur d'autres bibliothèques graphiques qui sont en général spécifiques au système d'exploitation ou qui apportent des fonctions complexes par dessus la bibliothèque *OpenGL*, par exemples *OpenGL Utility Library (GLU)* ainsi que l'*OpenGL Utility Toolkit (GLUT)*. **[4]**

6.2.SDL :

Est une API multimédia multi plateforme, permet de rendre notre application indépendante du système d'exploitation. Elle est composée de 6 API principales, conçues pour la gestion des événements, *threads*, Timers, Vidéo, Audio, CD-ROM.

7.Description des classes de la Bibliothèque :

La bibliothèque est écrite en langage de programmation C++. Elle est composée d'un ensemble de classes assurant le Chargement, la

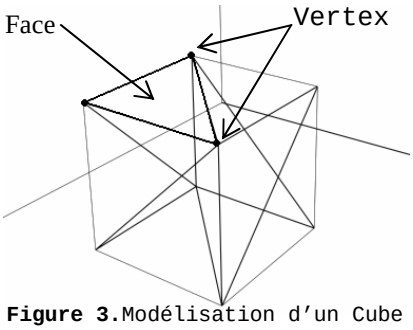

Figure 3. Modélisation d'un Cube

transformation et le rendu de la scène 3D.

Les classes principales du Kit proposé sont :

Vertex : Pour la représentation des vertex (point en 3D), défini par leur cordonnées cartésiennes.

Facette : Surface triangulaire élémentaire, composée de 3 vertex représentant les sommets du triangle, et sa normale.

Polyèdre : il est défini par une liste de faces composant son enveloppe.

Objet3D : C'est un ensemble de polyèdres liés entre eux par des liaisons de type rigide ou articulation (pivot, prismatique). Il peut être un objet simple (un seul polyèdre) ou complexe, statique ou dynamique.

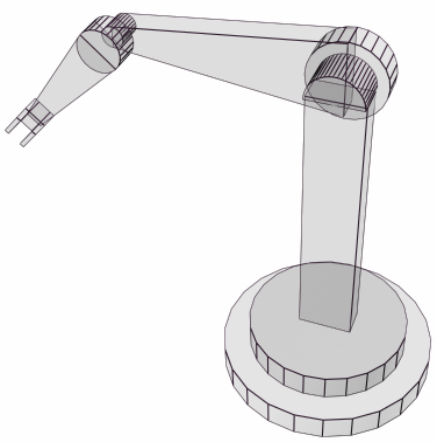

Figure 4. Robot manipulateur - Exemple d'un objet complexe

Camera : Cette classe représente une camera, définissant une position d'observation d'un lieu quelconque de

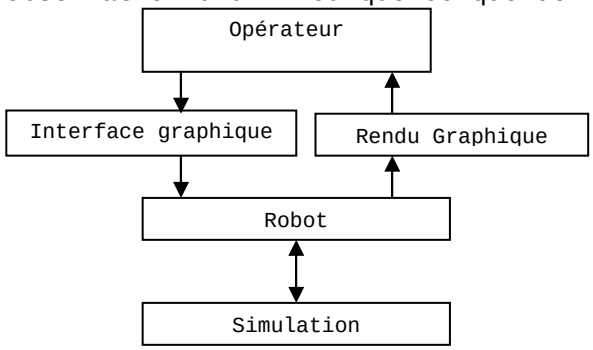

Figure 7.Schéma de principe

l'univers dans une direction donnée.

Lumière Omnidirectionnel : Gestion des lumières définies par *OpenGL*.

8.Exemple de simulation :

Déplacement d'un Robot Mobile de type véhicule Télécommandé.

Notre but dans cet exemple est la conception d'un programme de simulation en temps réel de la commande à distance d'un robot mobile de type véhicule.

Le véhicule choisi est muni :

- d'une camera embarquée.

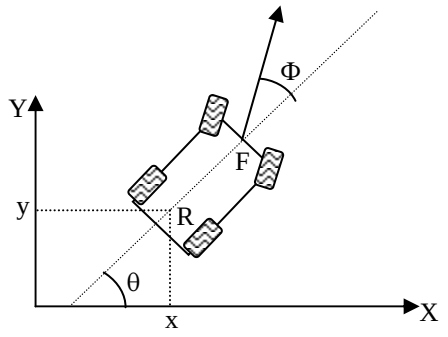

Figure5 . Modèle géométrique.

- 8 capteurs ultrasonores (porté 100 cm).

Le modèle cinématique du véhicule est donné par les équations suivantes :

$$\dot{X}_R = V_R \cos(\theta).$$

$$\dot{Y}_R = V_R \sin(\theta).$$

$$\dot{\theta} = V_R \frac{\tan(\Phi)}{RF}$$

L'architecture proposée pour le programme de simulation est représentée par le schéma suivant :

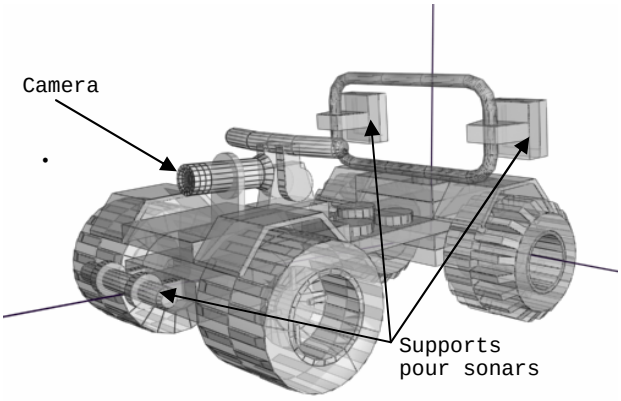

Figure 6. Robot Mobile de type véhicule (Objet3D de 7375 facettes)

La fréquence du rendu graphique est de 30 images par seconde (c.a.d 30. Hz).

Le module Robot est une classe Objet3D.

Le module simulation utilise le modèle cinématique du véhicule pour calculer le mouvement du véhicule. Il assure aussi lecture des données des capteurs ultrasonore. La fréquence d'exécution est de 100 Hz.

Chaque module est représenté dans l'application par une classe, ou un processus informatique.

Le modèle 3D du véhicule est constitué de 7375 facettes.

Conclusion :

L'objectif essentiel de ce travail, a consisté à mettre au point un Kit de développement d'application pour la simulation 3D des plateformes robotiques mobiles.

Les travaux futurs envisagent l'élargissement du Kit vers des classes d'animation d'objets déformantes, et de simulation dynamique.

La coopération avec d'autres laboratoires de mécanique, génie civil, physique, ou structure de la matière, peut donner à notre travail plusieurs extensions et d'applications.

Référence :

[1] J.-P.Lallemand et S. Zeghloul. *Robotique, Aspects fondamentaux, Modélisation mécanique, CAO robotique – Commande.* MASSON, Paris 1994.

[2] P.Coiffet. *La Robotique, Principe et application.* HERMES, Paris 1986.

[3] P. Taylor. *3D Graphics Programming in Windows.* Addison-Wesley Publishing Company 1994.

[4] J. Fuller. *OpenGL Programming Guide, Second Edition.* Addison-Wesley Publishing Company 1997.

[5] J.M.Degeeter.*Approche du temps réel industriel.* ELLIPSES, Paris 1999.

[6] E.GAUTHIER .*Utilisation des réseaux de neurones artificiels pour la commande d'un véhicule autonome.* Thèse de doctorat. L'Institut National Polytechnique de GRENOBLE. Jan 1992.

[7] J.P.Laumond. *Robot Motion Planning and Control*. SPRINGER 1998.

[8] P.REIGNIER. *Pilotage réactif d'un robot mobile*. L'Institut National Polytechnique de GRENOBLE. Déc 1994.