

# Pemetaan dan Lokalisasi Secara Simultan Robot iSRo menggunakan Multi-sensor

Achmad Luthfi Rosyadi<sup>1</sup>, Son Kuswadi<sup>2</sup>, Indra Adji Sulistijono<sup>2</sup>

<sup>1</sup>Mahasiswa Jurusan Mekatronika, Politeknik Elektronika Negeri Surabaya

<sup>2</sup>Dosen Jurusan Mekatronika, Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember

<sup>1</sup>ufi\_meka@yahoo.com

## Abstrak

SLAM ( Simultaneous Localization And Mapping) merupakan suatu teknik yang digunakan oleh mobile robot untuk membangun peta dari daerah yang sedang dijelajahi, dimana daerah tersebut tidak diketahui bentuk medannya sebelumnya. Di dalam SLAM, robot akan melakukan dua tugas sekaligus, yaitu membentuk peta dan juga mampu menentukan posisi robot di dalam peta tersebut. Dalam proyek akhir ini, dibuat program visualisasi untuk menggambarkan pergerakan robot, baik arah hadap robot, jarak yang ditempuh robot, bentuk halangan yang telah dideteksi oleh robot. Paket data yang dikirimkan oleh robot berisi data tentang jarak halangan yang berasal dari sensor ultrasonik, arah hadap robot yang berasal dari keluaran kompas digital CMPS03, dan jarak tempuh robot menggunakan data dari rotary encoder. Paket data tersebut akan dikirimkan ke software yang telah dibuat dengan menggunakan komunikasi serial secara terus menerus. Selain itu, di dalam program ini juga, kita juga dapat mengetahui posisi robot yang diupdate secara terus menerus. Dari hasil pengujian proyek akhir ini, didapatkan data bahwa gambar jarak antar proses update posisi ke-x dengan ke-x-1 masih terlalu jauh atau masih ada celah sehingga gambar yang dihasilkan masih kurang halus. Selain itu, selip yang terjadi pada roda juga memberikan tingkat kesalahan pembacaan, karena selip ini tidak akan terbaca oleh sensor rotary. Range pembacaan sensor kompas masih belum detail. Range pengukuran akan berubah tiap 10° derajat saja bukan tiap 1° sehingga mempengaruhi tingkat kehalusan gambar yang dihasilkan ketika proses pemetaan.

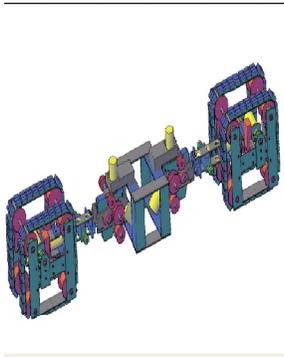
## 1. Pendahuluan

Tuhan telah menciptakan sistem navigasi yang begitu baik di alam ini. Beberapa hewan telah diciptakan jutaan tahun lalu memiliki kemampuan navigasi yang baik, diantaranya adalah tawon, semut, paus punggung bongkok, penyu hijau dll. Hal inilah yang menginspirasi para pakar robot untuk mengaplikasikan system navigasi pada robot terutama mobile robot. Sehingga di zaman sekarang bermunculan bermacam-macam teknik navigasi, salah satunya adalah SLAM (Simultaneous Localization And Mapping). Dengan metode ini, maka kita dapat mengetahui bentuk medan suatu daerah dimana bentuk medan daerah tersebut tidak diketahui sebelumnya. Hasil dari proses SLAM ini akan sangat membantu dalam pergerakan robot. Terutama ketika mobile robot difungsikan menjadi autonomous mobile robot.

## 2. Dasar teori

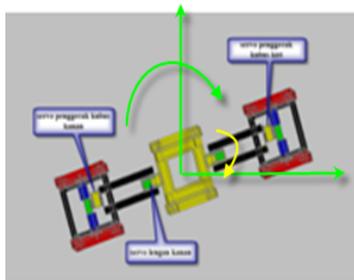
Pemetaan merupakan salah satu kemampuan yang saat ini harus ada pada sebuah mobile robot. Ada beberapa teknik pemetaan yang dikembangkan oleh beberapa peneliti. Salah satunya adalah SLAM ( Simultaneous Localization And Mapping ). Dengan menggunakan metode ini, maka robot akan melakukan dua tugas sekaligus,

yaitu memetakan wilayahnya sekaligus mampu mengetahui posisi dirinya di dalam wilayah yang sedang di jelajahi tersebut. Informasi bentuk peta wilayah dari robot tersebut didapatkan dari beberapa tipe sensor yang mendukung untuk melakukan proses navigasi, dimana sensor tersebut dipasang pada robot. Data-data sensor ini akan dikirimkan secara terus menerus dimana selanjutnya akan diproses agar dapat menghasilkan bentuk peta dan juga posisi terkini dari robot dalam koordinat kartesian. Dalam paper ini akan dibahas tentang pembuatan software SLAM untuk robot iSRo. iSRo merupakan robot rescue yang dibuat dengan keunikan bentuk mekaniknya sehingga mampu melakukan banyak gerakan yang gerakan-gerakan tersebut sangat diperlukan untuk menjelajah medan bencana atau yang lainnya. Robot iSRo merupakan robot yang memiliki 3 kubus, dimana kubus kiri dan kanan dijadikan sebagai penggerak robot tersebut. Dari tiap kubus penggerak ini, terdapat 2 motor penggerak yang bisa dikontrol secara terpisah. Dimana kubus utama dengan kubus kiri dan kanan dihubungkan dengan lengan-lengan yang prinsip kerjanya mirip dengan kerja tangan manusia. Untuk pengontrolannya pun membutuhkan teknik pengontrolan yang khusus. Gambar dibawah ini merupakan gambar design mekanik dari robot iSRo.



**Gambar 2.1** Rancangan robot iSRo

Selain itu teknik perhitungan navigasi yang digunakan adalah bukan menggunakan teknik Differential Drive Mobile Robot, melainkan lebih menggunakan pendekatan Dead Reckoning, walaupun tipe penggerak tiap kubusnya bisa dikatakan menggunakan tipe Differential Mobile Drive. Apabila “lengan depan” digerakkan dengan sudut tertentu, maka robot akan membentuk gerakan sesuai dengan sudut gerakan lengan tersebut. Adapun ilustrasi pergerakan robot seperti tampak pada gambar 2.2



**Gambar 2.2** Ilustrasi pergerakan robot

Untuk mendapatkan posisi robot dalam koordinat kartesian, maka persamaan yang digunakan adalah : (Sumber: Wahyuni, Erni D., 2010, *Rancang Bangun Program Visualisasi Pergerakan Differential Drive Mobile Robot*, Teknik Informatika, PENS-ITS, Surabaya)

$$X = S * \cos \theta \dots\dots\dots(2.1)$$

$$Y = S * \sin \theta \dots\dots\dots(2.2)$$

Dimana :

- X= posisi robot dalam koordinat X
- Y= posisi robot dalam koordinat Y
- S = jarak tempuh robot
- θ = arah hadap robot

Sedangkan untuk mendapatkan posisi terupdate dari robot ketika bergerak dalam koordinat kartesian, maka persamaan yang digunakan adalah:

$$X_x = S_x * \cos \theta_x + X_{x-1} \dots\dots\dots(2.3)$$

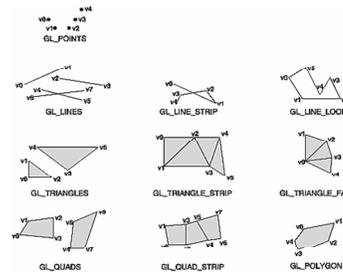
$$Y_x = S_x * \sin \theta_x + Y_{y-1} \dots\dots\dots(2.4)$$

Dimana

- $X_x$  = posisi terkini robot dalam koordinat X
- $Y_x$  = posisi terkini robot dalam koordinat Y
- $S_x$  = jarak tempuh robot
- $\theta_x$  = arah hadap robot
- $X_{x-1}$  = posisi koordinat X robot sebelumnya
- $Y_{x-1}$  = posisi koordinat Y robot sebelumnya

## 2.1 OpenGL

OpenGL merupakan salah satu software interface yang digunakan untuk menampilkan grafik hardware, dimana para pembuat dapat membuat sesuai dengan desain yang mereka harapkan. Untuk membuat aplikasi yang berbasis OpenGL, maka ada beberapa library dan header yang harus ditambahkan, diantaranya adalah glut.h, glut32.lib, opengl32.dll, glut32.dll. OpenGL bersifat open-source, multi platform dan juga multi bahasa. OpenGL bisa bekerja di lingkungan Windows, Unix, SGI, Linux, Free BSD, dll. Dibawah ini merupakan contoh gambar sederhana yang dihasilkan dari pemrograman menggunakan OpenGL. Seperti tampak pada gambar 2.3



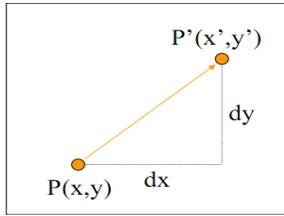
**Gambar 2.3** Gambar sederhana di dalam OpenGL

### 2.1.1 Transformasi dalam OpenGL.

Dalam openGL, terdapat beberapa fungsi transformasi, diantaranya adalah translasi, scaling, dan rotasi. Karena dalam penelitian ini hasil akhirnya dalam bentuk gambar dua dimensi, maka dalam penelitian ini, fungsi transformasi yang digunakan adalah hanya fungsi translasi dan rotasi.

#### 2.1.1.1 Translasi

Translasi adalah proses perpindahan obyek dari titik satu ke titik lainnya secara linier. Proses translasi ini dapat digambarkan dalam bentuk persamaan seperti dibawah ini: (Sumber : Basuki, Achmad & Nana Ramadijanti, *Transformasi 2 Dimensi*(PENS-ITS))



Gambar 2.4 Translasi[2]

$$x' = x + dx \dots\dots\dots(2.5)$$

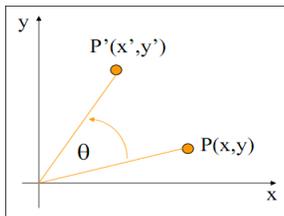
$$y' = y + dy \dots\dots\dots(2.6)$$

sedangkan dalam bentuk matrik, proses translasi dapat dituliskan seperti dibawah ini:

$$= + \dots\dots\dots(2.7)$$

2.1.1.2 Rotasi

Rotasi adalah perpindahan obyek dari titik yang satu ke titik lainnya yang berupa pemindahan berputar sebesar sudut  $\theta$ . Adapun bentuk persamaan untuk proses rotasi ini adalah sebagai berikut: (Sumber : Basuki, Achmad & Nana Ramadijanti, *Transformasi 2 Dimensi*(PENS-ITS))



Gambar 2.5 Rotasi[2]

$$x' = x \cos (\theta) - y \sin (\theta) \dots\dots\dots(2.8)$$

$$y' = x \sin (\theta) + y \cos (\theta) \dots\dots\dots(2.9)$$

2.2 Wsc4c

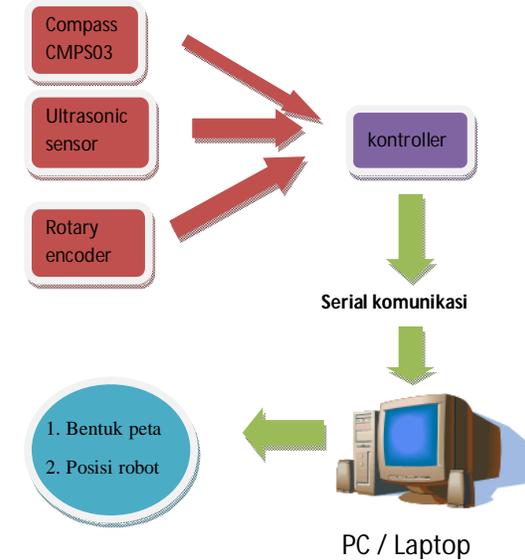
Wsc4c merupakan suatu library yang khusus digunakan untuk pemrograman serial komunikasi pada windows standard. Wsc4c sendiri dapat digunakan atau dikembangkan untuk aplikasi serial pada C/ C++, Visual C++, .NET dan Visual C#. Sama seperti halnya dengan OpenGL agar dapat menggunakan aplikasi ini, maka kita harus menambahkan beberapa header ke dalam program yang kita buat. Beberapa header yang harus ditambahkan adalah wsc.h, sayerror.h, dan keycode.h

2.3 Sensor kompas CMPS03

Sensor ini digunakan untuk mendeteksi arah hadap robot ketika berjalan. Sensor ini bekerja dengan mendeteksi medan magnet bumi.

3 Perancangan system

Secara umum, rancangan system yang digunakan seperti tampak pada gambar 3.1



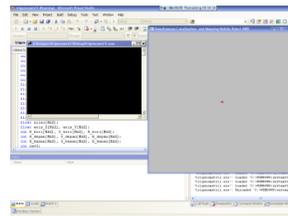
Gambar blok 3.1 Diagram sistem

Dari gambar diatas terdapat beberapa sensor dan juga beberapa fitur, dimana masing-masing memiliki fungsi berbeda-beda, diantaranya:

1. Kompas CMPS03, digunakan untuk menentukan arah hadap robot
2. Sensor ultrasonic, digunakan untuk mengetahui jarak halangan terhadap robot
3. Sensor rotary encoder, digunakan untuk mengetahui jarak yang telah ditempuh robot
4. Komunikasi serial, digunakan untuk mengirimkan data sensor ke dalam computer untuk selanjutnya diolah menjadi bentuk peta dan posisi robot dalam koordinat kartesian

3.1 Simulator SLAM

Adapun rancangan tampilan dari simulator yang dibuat tampak seperti pada gambar 3.2



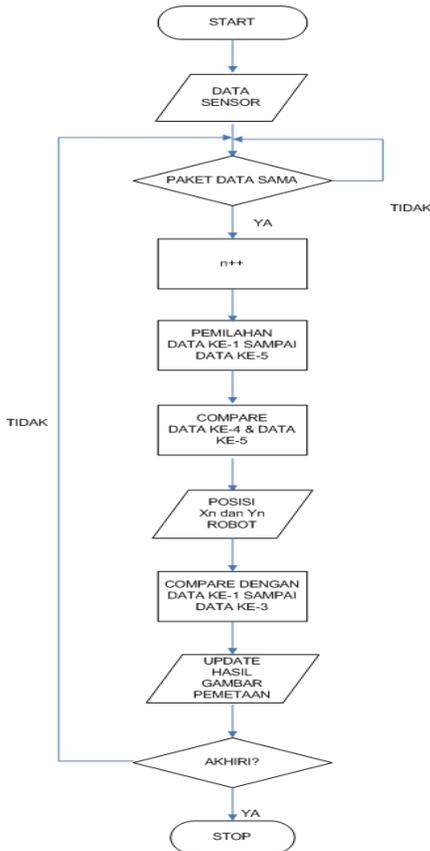
Gambar 3.2 Tampilan Simulator SLAM

Pembuatan simulator SLAM ini digunakan untuk menggambar bentuk peta sesuai dengan data sensor yang dikirimkan ke computer. Simulator ini dibuat menggunakan bahasa pemrograman OpenGL, dengan simulator ini,

memungkinkan kita untuk melihat langsung bentuk peta yang digambar secara real time, dan kita bagaimana pergerakan robot

### 3.2 Perancangan dan pembuatan logika

Adapun flow chart dari sistem secara keseluruhan tampak seperti pada gambar 3.3



Gambar 3.3 Flowchart sistem secara keseluruhan

## 4. Pengujian

### 4.1 Pengujian sensor

#### 4.1.1 Sensor kompas CMPS03

Berikut ini adalah hasil pengujian sensor kompas

Tabel 4.1 Hasil Pengujian Sensor Kompas

Data output	sudut
238 =< x <= 240	0° < utara >
232 =< x <= 235	10°
227 =< x <= 229	20°
225 =< x <= 226	30°

Tabel 4.1 Lanjutan

220 =< x <= 224	40°
216 =< x <= 219	50°
213 =< x <= 215	60°
208 =< x <= 211	70°
204 =< x <= 206	80°
200 =< x <= 202	90° < barat >
195 =< x <= 197	100°
191 =< x <= 193	110°
186 =< x <= 188	120°
181 =< x <= 183	130°
174 =< x <= 178	140°
166 =< x <= 171	150°
158 =< x <= 161	160°
151 =< x <= 152	170°
144 =< x <= 147	180° < selatan >
136 =< x <= 139	190°
128 =< x <= 131	200°
117 =< x <= 123	210°
107 =< x <= 113	220°
97 =< x <= 101	230°
84 =< x <= 90	240°
74 =< x <= 81	250°
60 =< x <= 67	260°
46 =< x <= 51	270° < timur >
35 =< x <= 39	280°
25 =< x <= 30	290°
16 =< x <= 18	300°
8 =< x <= 10	310°
3 =< x <= 5	320°
252 =< x <= 255	330°
246 =< x <= 247	340°
243 =< x <= 244	350°

Dalam pengujian ini range perubahan sudut pada kompas digital hanya dibatasi pada perubahan sudut sebesar 10° dan perubahan lebih kecil dari 10° tidak terdeteksi.

### 4.1.2 Sensor Ultrasonik

Berikut ini adalah hasil pengujian sensor ultrasonik

**Tabel 4.2** Hasil Pengujian Sensor Ultrasonik

Pengukuran real	Pengukuran sensor
1 cm	1 cm
2 cm	2 cm
6 cm	5 cm
10 cm	8 cm
15 cm	11 cm
20 cm	15 cm
25 cm	20 cm
30 cm	22 cm
40 cm	30 cm
50 cm	38 cm
70 cm	53 cm
90 cm	67 cm
100 cm	75 cm

### 4.1.3 Sensor rotary encoder

Adapun hasil pengujian dari rotary encoder adalah sebagai berikut:

Jarak = (keliling lingkaran / jumlah lubang) \* jumlah pulsa yang dibangkitkan sensor rotary

Diketahui :

keliling lingkaran = 13.8 cm

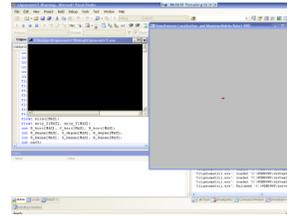
jumlah lubang = 22

dari perhitungan diatas, diketahui bahwa tiap 1 putaran pulsa, maka jarak yang ditempuh oleh robot adalah 0.6 cm.

**Tabel 4.3** Pengujian Sensor Rotary

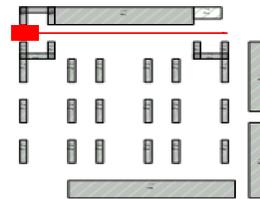
Jumlah pulsa	Jarak tempuh (teori)	Jarak tempuh (praktek)	%error
5	3cm	2cm	33,3%
10	6cm	5cm	16,7%
30	18cm	15cm	16,7%
35	21cm	18cm	14,3%
45	27cm	24cm	11,1%

### 4.2 Pengujian software



**Gambar 4.1** Tampilan simulator SLAM robot iSRo

Adapun denah ruangan yang akan dilakukan sebagai tempat percobaan system dapat dilihat pada gambar dibawah ini:

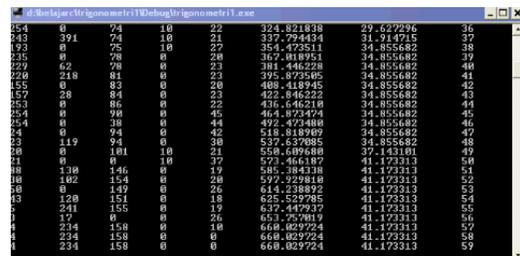


**Gambar 4.2** Denah ruangan dan rute pergerakan robot

Setelah melakukan percobaan system pada ruangan dan rute pergerakan yang sebelumnya telah ditentukan, maka didapatkan hasil seperti tampak pada gambar:



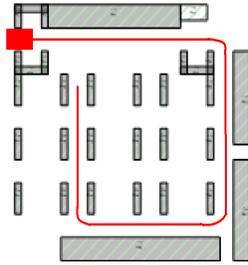
**Gambar 4.3** Hasil pemetaan robot iSRo



**Gambar 4.4** Hasil lokalisasi robot iSRo

Dari percobaan pertama ini, gambar yang dihasilkan sudah hampir memiliki tingkat kesamaan dengan denah ruangan yang sebenarnya. Sedangkan untuk pelokalisasi robot selama percobaan ini diketahui bahwa posisi robot terakhir berada pada koordinat X= 660 cm dan Y= 42cm dari posisi awal robot.

Untuk mengetahui hasil pemetaan pada pergerakan yang lebih kompleks, maka dilakukan percobaan kedua. Adapun rute pergerakan robot dalam percobaan kali ini tampak seperti pada gambar dibawah ini:



**Gambar 4.4** Rute pergerakan robot pada percobaan ke-2

Adapun hasil pemetaan pada percobaan ke- 2 adalah:



**Gambar 4.5** Hasil pemetaan robot iSRo

x	y	z	u	v	w	x	y	z	u	v	w
28	0	45	100	27	93	-181038					-352.054016
292	0	4	100	24	90	566864					-337.228180
156	220	158	100	24	87	952690					-322.402344
136	121	149	100	27	85	811742					-305.723267
173	35	208	100	28	81	961876					-288.426453
89	44	148	100	25	79	238777					-272.982880
83	0	45	100	30	75	971862					-254.456579
32	169	45	100	24	73	356888					-239.624741
31	110	45	100	27	70	415939					-222.945679
29	423	50	100	26	67	583916					-206.884354
27	73	51	100	28	64	534050					-189.587540
26	271	51	100	2	64	316200					-188.352051
25	271	51	100	0	64	316200					-188.352051
26	272	53	100	0	64	316200					-188.352051
25	248	51	100	0	64	316200					-188.352051
25	222	51	100	0	64	316200					-188.352051
25	271	51	100	0	64	316200					-188.352051
25	248	51	100	0	64	316200					-188.352051
25	247	51	100	0	64	316200					-188.352051
26	271	51	100	0	64	316200					-188.352051
25	247	51	100	0	64	316200					-188.352051
26	247	51	100	0	64	316200					-188.352051
26	270	51	100	0	64	316200					-188.352051
25	248	51	100	0	64	316200					-188.352051

**Gambar 4.6** Hasil lokalisasi robot iSRo

Pada percobaan ke-2 ini, hasil yang dihasilkan memiliki tingkat kesamaan seperti percobaan yang pertama. Pada percobaan ini diketahui bahwa posisi terakhir robot berada pada koordinat X= 64cm dan Y= -188cm dari posisi awal robot

## 5. Kesimpulan

1. Slip yang terjadi pada roda mengakibatkan error yang dihasilkan pada proses odometri data menjadi besar
2. Kecepatan robot ketika bergerak sangat mempengaruhi hasil gambar dari proses pemetaan. Semakin cepat pergerakan robot, maka gambar yang dihasilkan sedikit kasar. Berbanding terbalik ketika kecepatan robot lambat, maka gambar yang dihasilkan lebih detail dan halus
3. Hasil pemetaan dari software yang telah dibuat hampir memiliki kesamaan dengan denah ruangan yang sebenarnya baik bentuk halangan maupun letak halangan.
4. Proses pelokalisasi robot dapat berjalan dengan baik di dalam software yang telah dibuat

## 6. Pustaka

- [1] Anonim , 2010, Sistem Navigasi dalam DuniaHewan, <http://img.gd.itb.ac.id/?p=62>. Diakses tanggal 20 Agustus 2010
- [2] Basuki, Achmad & Nana Ramadijanti. *Transformasi 2Dimensi* ( PENS-ITS).
- [3] DatasheetATMEGA16, [www.datasheetcatalog.com](http://www.datasheetcatalog.com)
- [4] Datasheet MAX232, [www.datasheetcatalog.com](http://www.datasheetcatalog.com)
- [5] Datasheet Book SHARP GP1S53, <http://www.Digi-ware.com>
- [6] Datasheet Book Devantech CMPS03, <http://www.Digi-ware.com>
- [7] Datasheet Book PING))) Ultrasonik Range Finder, <http://www.parallax.com>.
- [8] Datasheet Book Omron Rotary Encoder E6A2-C, <http://www.ia.omron.com>
- [9] Datasheet Book Omron Rotary Encoder E6C-N, <http://www.ia.omron.com>
- [10] Datasheet Book RX-64, <http://www.crustrawler.com>
- [11] Datasheet Book DC Gear Motor, <http://www.Digi-ware.com>
- [12] H,Durrant. Fellow. IEEE. T.Bailey. *Simultaneous Localisation and Mapping (SLAM): Part 1 The Essential Algorithms*.
- [13] Jacky,Chang.H,C.S.George Lee, Yung-Hsiang Lu, Y.Charlie Hu, 2006, *Simultaneous Localization and Mapping with Environmental Structure Prediction*.Purdue University.

- [14] Neider.Jackie, Davis.Tom. 1997. *OpenGL Programming Guide*. United State of America : Addison-Wesley Publishing Company.
- [15] Siciliano,Bruno. Oussama Khatib. 2008 *Handbook of Robotics*. Verlag Berlin Heidelberg :Springer
- [16] Siciliano Bruno, Oussama Khatib, Frans Groen. 2008. *Robotics and Cognitive Approaches to Spatial Mapping*. New York : Springer
- [17] Wright Jr .Richard S, Benjamin Lipchak, Nicholas haemel. 2007. *OpenGL Superbible.4<sup>th</sup>.edition*. United State of America : Addison-Wesley Publishing Company