GS2 Intelligent Robots and Mobile Robots (II)

One-wheel Personal Transportation Vehicle : Gyrocycle

June Hyung Park and Seul Jung
Intelligent Systems and Emotional Engineering Lab
Department of Mechatronics
Chungnam National University, Dajeon, Korea
http://isee.cnu.ac.kr, jungs@cnu.ac.kr

***Abstract*—This paper presents design and implementation of one-wheel mobile robot called Gyrocycle. Gyrocycle is designed to carry a passenger as a personal transportation vehicle aimed for commuting in urban environment. Gyrocycle uses fast rotating flywheels to generate gyro effects so as to prevent Gyrocycle from falling in the roll angle direction. A prototype design of Gyrocycle is implemented and controlled. Experimental studies of balancing the Gyrocycle system are presented.**

Keywords; Gyro effect, single-wheel robot, balancing control, personal transportation vehicle

I. INTRODUCTION

A popular personal transportation vehicle is an automobile that has a combustion engine using Gasoline as fuel. Due to the environmental acts, air pollution by Carbon Oxide emission is regulated by law in many cities. Therefore, current paradigm of automobile design is rapidly shifting from combustion engine-based vehicles to hybrid-typed vehicles that use both electric power and a combustion engine for driving although the cost problem is still present.

Furthermore, paradigm of automobiles is gradually shifting from hybrid to electrical vehicles. Specially, a variety of the small-sized electrical vehicles are promoted in auto showcase as a future concept car. This enables automobile technology and mobile robot technology to merge together into one. The size of mobile robots is enlarged to carry human passengers and on the contrary automobiles are downsized to carry a small number of passengers to meet the current requirement of a battery charging capacity.

One popular personal transportation robotic vehicle is Segway [1]. Segway is a typical mobile robot based on an inverted pendulum concept driven by two wheels. It has been commercialized to carry a human passenger in the urban area. TransBot has been designed and implemented for a human driver as a personal transportation vehicle [2]. TransBot has two wheels to drive like Segway, but it is designed for a human driver to sit and drive the vehicle like an automobile [3]. PUMA with two wheels is developed to carry two passengers based on two-wheel mechanism [4].

More challengingly, single-wheel typed mobile robots are introduced in the literature. Gyrover has been developed for a long time and has shown the possibility as a single-wheel mobile robot system [5]. Control and implementation of Gyrover have been demonstrated by experimental studies.

In the similar framework of design, in our previous research, GYROBO has been developed [6, 7]. A series of single-wheel robots has been implemented and controlled. GYROBO I has made successful balancing and navigation under the intelligent control method, but notable oscillation in the balancing performance was observed. The later version of GYROBO was rebuilt to suppress oscillatory behaviors caused by vibration. A single-wheel GYROBO II has been empirically controlled for successful balancing and navigation.

In this paper, as an extension of a single-wheel mobile robot, a larger sized single-wheel mobile robot, Gyrocycle is presented. Gyrocycle is designed and developed to carry a human driver as a personal transportation vehicle. In the structure of Gyrocycle, dual flywheels are employed to generate a larger gyro effect force to sustain balancing a human driver. The balancing angle is detected by a gyro sensor and controlled by linear control method. Experimental studies of balancing control of Gyrocycle are performed.

II. DESIGN OF GYROCYCLE

A. Whole Body Design

Gyrocycle is aimed to carry a human driver so that the wheel size becomes larger. Overall schematic design is shown in Fig.1.

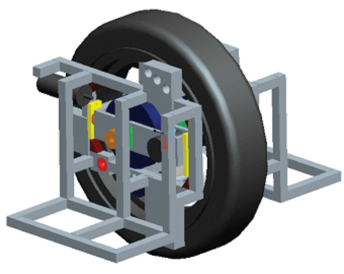


Fig. 1 Gyrocycle

An outer wheel is made of rubber used for tires of an automobile. All necessary hardware is located inside the wheel. Footsteps for a human driver's standing are designed in the lateral direction. Two flywheels are located in the opposite direction away from the center for generating gyro effect.

B. Flywheels

There are two flywheels inside the outer wheel. Two flywheels are located at 0.075*m* away from the center. To suppress vibration caused from high speed rotation of flywheels, a flywheel is designed as one body. The housing of the flywheel is also solidly designed to reduce the vibration. The schematic design of a flywheel is shown in Fig. 2. There are two motors to drive the flywheel, one is for rotating the flywheel at high speed and another is a tilt motor to rotate the housing of the flywheel.

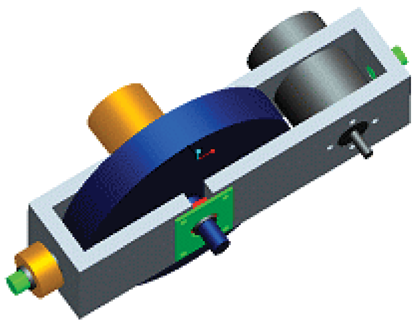


Fig. 2 Flywheel & housing

III. CONTROL STRUCTURE

A. Control Schemes

Control variables are roll and yaw angles of Gyrocycle. Two flywheels are controlled by the same controller of tilt motors to enlarge gyro effect forces. There are 5 actuators to be controlled for the Gyrocycle system. Spinning control for flywheels is done in open loop control fashion to generate $u_{\gamma R}, u_{\gamma L}$ for rotating right and left flywheels at 8000 *rpm*, respectively.

Roll angle, β_w and yaw angle, α_w are measured by a gyro sensor and fed back to form PD controllers as shown in Fig. 3.

$$u_\beta = k_{d\theta}(\dot{\beta}_{dw} - \dot{\beta}_w) + k_{p\theta}(\beta_{dw} - \beta_w) \qquad (1)$$
$$u_\alpha = k_{d\alpha}(\dot{\alpha}_{dw} - \dot{\alpha}_w) + k_{p\alpha}(\alpha_{dw} - \alpha_w)$$

where α_{dw}, β_{dw} are desired angles, and $k_{d\theta}, k_{p\theta}$ and $k_{d\alpha}, k_{p\alpha}$ are controller gains.

Two control inputs are summed together to control tilt motors for dual flywheels.

$$u_t = u_\beta + u_\alpha \qquad (2)$$

where is u_t the major control input to the system. Two flywheels take the same control input, u_t to increase the power.

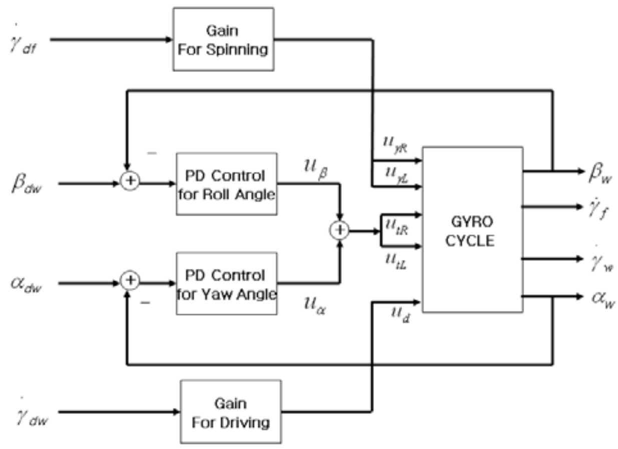


Fig. 3 Overall control structure

B. Hardware Structure

An overall control hardware structure is shown in Fig. 4. A DSP and an AVR processor are main hardware to control associated motors. An AVR processor controls the rotation of flywheels and a DSP controls tilt motors and a drive motor. A 3-axis gyro sensor is used to detect a balancing angle and a yaw angle.

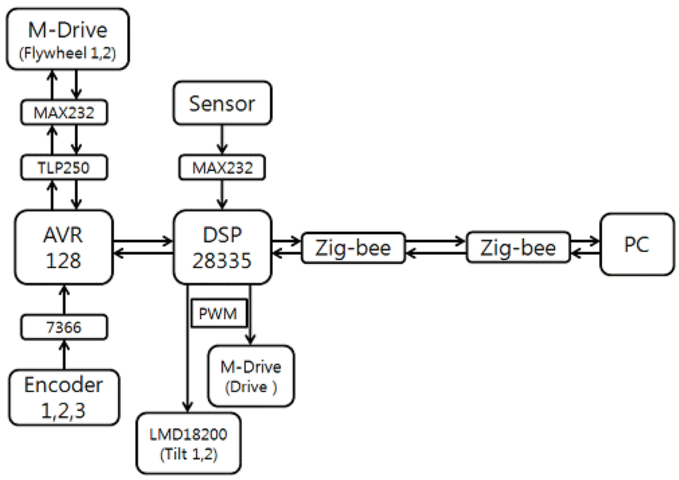


Fig. 4 Overall hardware structure

IV. Experimental Studies

A. Experimental Setup

The real Gyrocycle is developed and shown in Fig. 5 and its description is listed in Table 1. There are two tilt motors, two flywheel motors and a driving motor to actuate the system. The outer wheel mass is 12Kg and the inner wheel mass is 3.5 kg. Including other necessary hardware leads to the total mass of 29.5 Kg. The flywheel speed is set to rotate at 8000 *rpm*. Control sampling time is set to 50 *Hz*.

(a)Top view

(b) Front view

Fig. 5 Real Gyrocycle

Table 1. System Parameters

Wheel Specifications	Value	Flywheel Specifications	Value
Mass(*Kg*)	12	Mass(*Kg*)	3.5
Diameter (*m*)	0.505	Diameter (*m*)	0.15
Width (m)	0.12	Width (m)	0.025

B. Experimental Result

Experimental study of balancing control for Gyrocycle is conducted. Gyrocycle is required to maintain balancing at one point without moving. The current experiment is performed when a human driver is not aboard, but Gyrocycle itself.

Although linear control methods are used to control the balance, Gyrocycle well maintains balancing. The result of balancing performance is plotted in Fig. 6. Gyrocycle balances well and its balancing angle error is less than 1 degree.

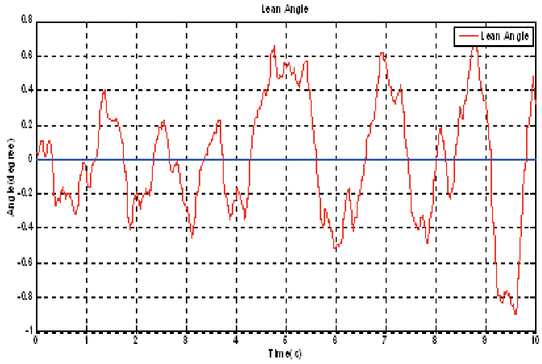


Fig. 6 Balancing control result of Gyrocycle

V. Conclusion

A single-wheel mobile robot to carry a human driver is designed, developed and controlled. To induce larger gyroscopic effect for balancing the body, dual flywheels are utilized. Simple linear control methods are used to control the balancing angle. Experimental studies confirmed successful balancing performance of Gyrocycle.

However, balancing control experiments were conducted for Gyrocycle itself without a human driver on board. To function as a personal transportation system for Gyrocycle, experimental studies are required when a human driver is on board.

Acknowledgment

This research has been partially supported by Korea Research Fund (KRF 2011-0027055) and the center for autonomous intelligent manipulation (AIM) for service robots of the MKE (The Ministry of Knowledge Economy), Korea, under the Human Resources Development Program for Convergence Robot Specialists support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C7000-1001-0003).

References

[1] SEGWAY, http://segway.com

[2] H. J. Lee, H. W. Kim, and S. Jung, "Development of a mobile inverted pendulum robot system as a personal transportation vehicle with two driving mode : TransBOT", ISORA, 2010

[3] H. W. Kim and S Jung, "Design and Analysis of a Robotic Vehicle for Entertainment Using Balancing Mechanism", *URAI 2011*, pp. 855-857, 2011.

[4] PUMA, http://segway.com

[5] Y. S. Xu and Y. S. Oh, "Control of single wheel robots", *Springer*, 2005

[6] P. K. Kim, J. H. Park, and S. Jung, "Experimental Studies of Balancing Control for a Disc-Typed Mobile Robot Using a Neural Controller: GYROBO", *ISIC2010, Multi-Conference on Systems and Control*, September 2010, pp. 1499

[7] P. K. Kim, J. H. Park, and S. Jung, "Experimental Studies of Neural Network Control for One-Wheel Mobile Robot", *Journal of Control Science and Engineering, Article ID 194397, 2011*

Design of Robotic Handling Gripper for Pre-finished Metal Sheet

Kyoung Taik Park
Robotics and Mechatronics Lab
Research Division of Advanced Manufacturing System
Korea Institute of Machinery & Materials
156 Gajeongbuk-ro,Yuseong-gu, Daejeon-city, Korea
ktpark@kimm.re.kr

Doo Hyung Kim
Robotics and Mechatronics Lab
Research Division of Advanced Manufacturing System
Korea Institute of Machinery & Materials
156 Gajeongbuk-ro,Yuseong-gu, Daejeon-city, Korea
Kdh649@kimm.re.kr

***Abstract*—This paper presents the robotic handling gripper using three finger gripper without marking the trace of pad or holder on pre-finished surface of color metal sheet. The trace to be caused in process to grip and transfer the pre-finished color metal sheet is to degrade the quality of products. To cope to these problems the pre-finished color metal sheet is exactly gripped by tool and then transferred to exact position. To exactly and easily grip the pre-finished color metal sheet the robotic handling gripper system is designed.**

Keywords-Multi link; handling gripper; driving mechanism; stepwise driving control; pre-finished color metal sheet

I. INTRODUCTION

The flaw to be caused in process to grip and transfer the pre-finished color metal sheet is the cause to degrade it. To cope to these problems the pre-finished color metal sheet is exactly gripped by tool and then transferred to exact position. To exactly grip the pre-finished color metal sheet the gripping tools are shown in Figure 1. The pre-finished color metal sheets in Figure 1 are classified as steel or non-steel material. To protect the luster and to prevent the scratch the pre-finished color metal sheet is classified in two metal sheets to have protecting film and non-protecting film. The gripping tools are classified into vacuum pad, magnetic holder and hybrid pad to be composed of magnetic holder and vacuum pad[1,2,3].

In the case of pre-finished color metal sheet of non-protecting film when the gripping tools are contacted with the surface of metal sheet the trace of holder or pad is remained on surface of metal sheet. To prevent this trace the vacuum pad is used. At this time, the pads to be used are halogenated NBR pad, stuck fluoro-resin pad and cyclone pad. The hybrid pad is used to grip solidly. In the case of pre-finished color metal sheet of protecting film the various tools are used because the worry of trace on surface of metal sheet is not needed. The magnetic holder is much better than vacuum pad because there is the possibility that the protecting film is separated from the metal sheet. The economical burden is considered in selection of gripping tools. In the consideration of economical burden, magnetic holder is much better than others. But in this case the influence of magnetic field is a cause of several problems and it requires the control technology of holding force. In the case of non-steel sheet the vacuum pad is only used with regardless of the used protecting film. To develop the gripping tools the vacuum pad is considered to select the capacity of pad in design process. The design processes are required to select the vacuum pad to grip the automobile parts. The holding forces of vacuum pad to grip and handle the automobile parts are classified into horizontal grip and vertical motion, horizontal grip and horizontal motion, vertical grip and vertical motion[4,5,6].

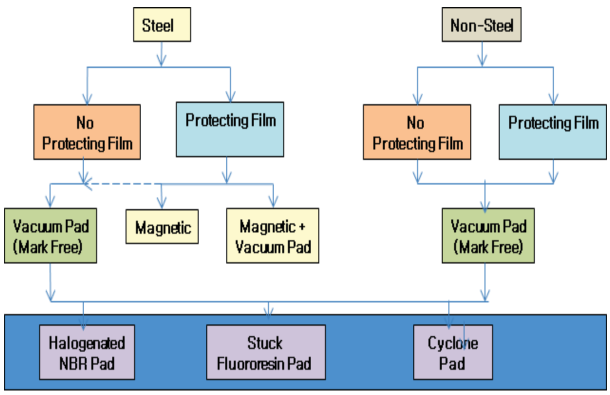


Figure 1. Classification of gripping tools for part handling

II. STEPWISE DRIVING ACTUATOR OF MULTI LINKS

A. Robotic Gripper Using Vacuum Pad

To grip and transfer the color metal sheet the magnetic holder and hybrid holder are used. The gripped metal sheet is transferred by using robot from one position to another position. Therefore, the vacuum and tooling gripper system to be attached on the end-effector is required. In the automobile company the manufactured body, chassis frame and pressed metal sheet are classified into fender, t/gate and hood as moving part, BACK, PAD SUPL COMPL and RR FLR as floor part and RR C/Lamp L/R, Side OTR L/R and OTR

ONR COMPL L/R as side part. To handle these parts by using robot the vacuum gripper system is used as device to be attached on end-effector in most case of working process.

The different vacuum pad gripper is used for each shape of pressed metal sheet. Therefore, there is an economical loss because the shape of vacuum gripper is exchanged and modified when the usage of gripper to be attached on end-effector is changed. Figure 2 and 3 shows some of major gripper systems to be represented in various working usages. Therefore, the fixed gripper system is needed to be changed into the type of flexible gripper system that the shape and mechanism of gripper can be automatically changed as the required usage. The flexible gripper system is required to improve the quality of product by gripping and handling of pre-finished color metal sheet.

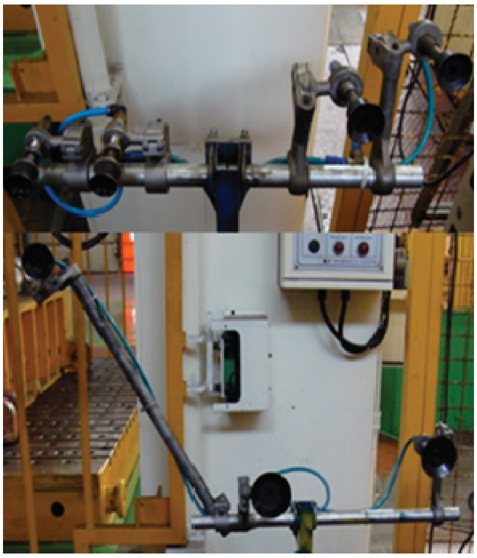
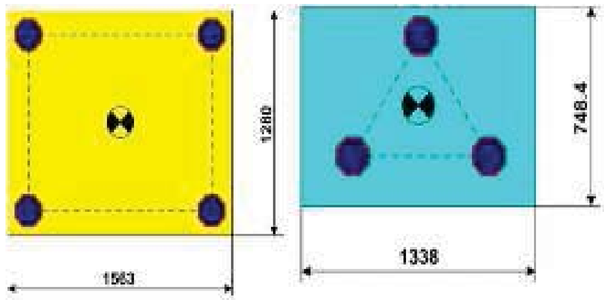


Figure 2. Conventional gripper using vacuum pad

Figure 3. Conventional gripper using vacuum pad

B. *Flexible Gripper Using Three Finger*

To grip and handle a various shape of automobile parts the flexible gripper system is developed. The developed flexible gripper system is designed to grip the various shape of automobile parts exactly by using the attached vacuum pad and the structure of gripping mechanism is flexibly changed by electrical actuator. First, it is needed to define the method of contact between gripping tool and automobile parts. This has the close relationship between complexity and economical cost of flexible gripper system. Figure 4 shows the method of three contact points and four contact points between gripping tool and automobile parts. The method of three contact points is more stable than four contact points in order to grip the various shape of automobile parts. The method of four contact points has the disadvantage of complexity of driving mechanism structurally. The method of three contact points is less complex than four contact points and the similar result to be compared with four contact points is obtained by extending contact points structurally.



(a) four contact points (b) three contact points

Figure 4. Contact points between gripping tool and metal sheet

The gripping mechanism structure is designed to be attached on the end effector. To grip the various shape of automobile parts the flexible mobility of gripping mechanism is required. At this time, to make the flexible mobility of gripper the electrical actuator is used. The electrical actuator is classified as two kinds of rotation and linear type in view point of driving motion. In the case of rotation type of actuator the gripper system requires the big capacity of electrical actuator because the inertia is increased by length and mass of composed links. This caused to increase the size of gripper system. The linear motion actuator has the advantage to be compacted in small size and to reduce economical cost.

C. *Kinematical Analysis and Detail Design*

To design the gripper using linear motion actuator, the weight of color metal sheet to be used as automobile parts is computed as, $W_{kg} \approx 20$ kgf, approximately. It is assumed as, $W_{kg} \approx 30$ kgf, by considering safety factor. When the method of three contact points is used to design gripping mechanism the gripping force of one finger is used as, $W \approx 10$ kgf, approximately. Therefore, the linear motion actuator 1 has the gripping force and the linear motion actuator 2 has the gripping force, W ≈ 10kgf, approximately.

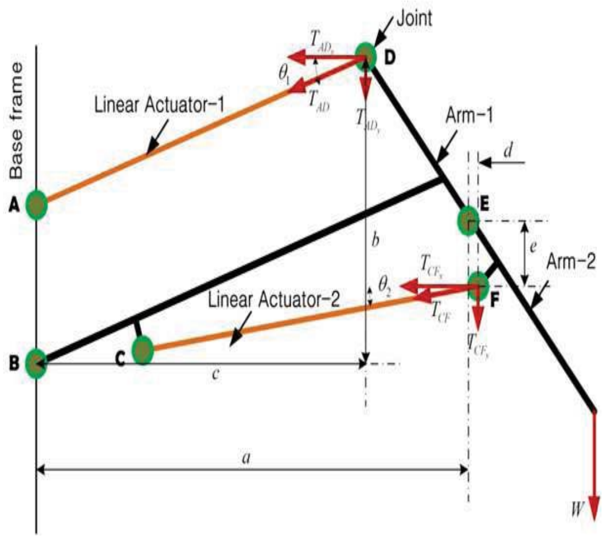


Figure 5. Free body diagram of gripper

D. *Design and Analysis Using CAE*

The previous constraint condition and motion analyses are accomplished by using CAE tool to be based on fundamental design. Figure 6 shows the results of CAE design of single finger gripper and Figure 7 shows the various projection view of it.

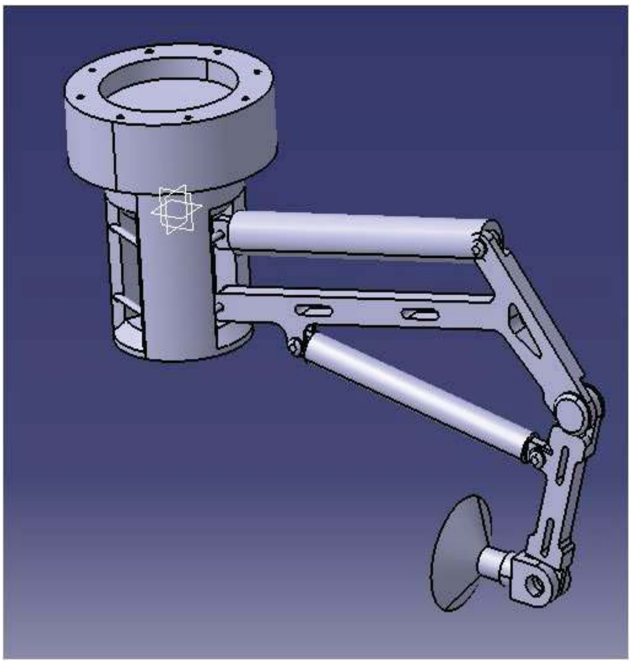


Figure 6. Single finger gripper

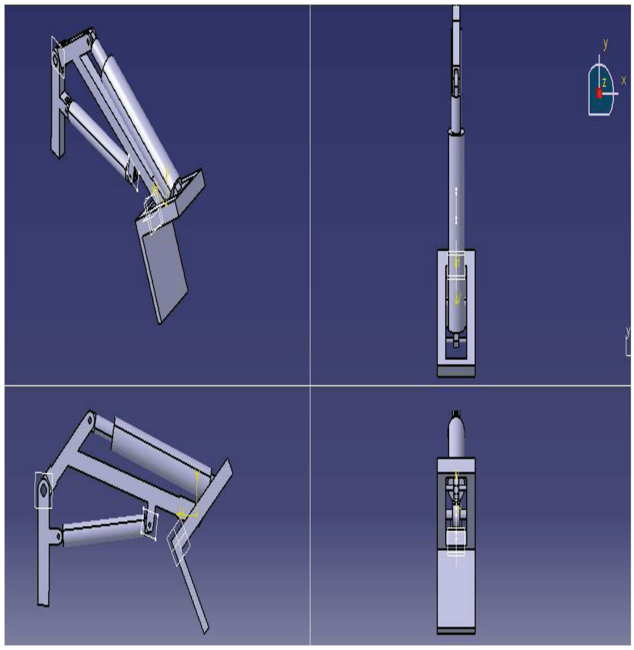


Figure 7. Projection view of single finger gripper

Figure 8 shows the kinematic constrain and motion space of single finger gripper. When the stroke range of actuator is changed in 0~50mm the raising angle of link1 is changed in 0~46° and the raising angle of link2 is changed in 0~36° . The change of raising angle has the gripping distance of automobile parts, 115~740 mm.

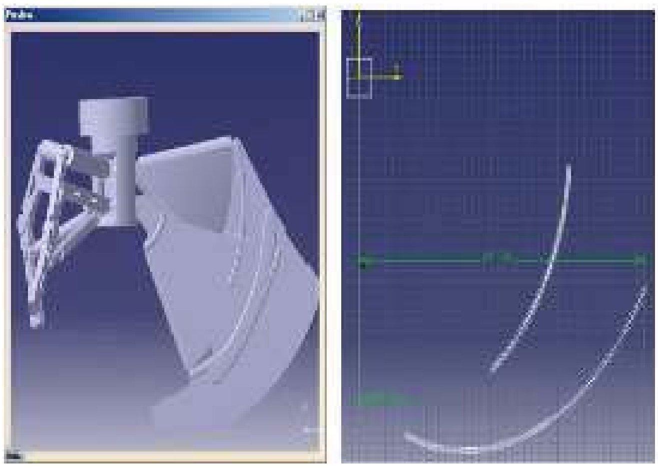


Figure 8. Analysis of kinematic constraint and motion space

Figure 9 shows three fingers gripper that has gripping tool, links, vacuum pad and bracket to be attached on end-effector of robot.

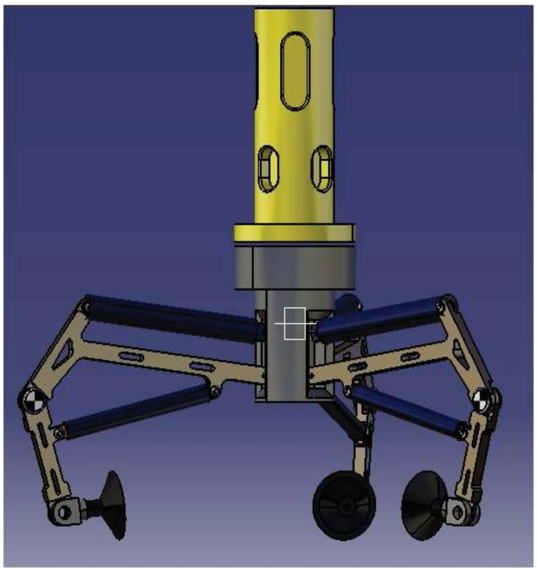


Figure 9. 3D modeling of three finger gripper

Figure 10 shows the electrical/electronic interface of flexible gripper system. To measure rotating angle at joint of single finger to be composed of 2 DOF mechanism the rotating potentiometer is attached on it. To grip automobile parts at end effector the vacuum pad, magnetic holder and hybrid pad are attached on it. These electrical/electronic systems are interfaced and controlled with embedded computer.

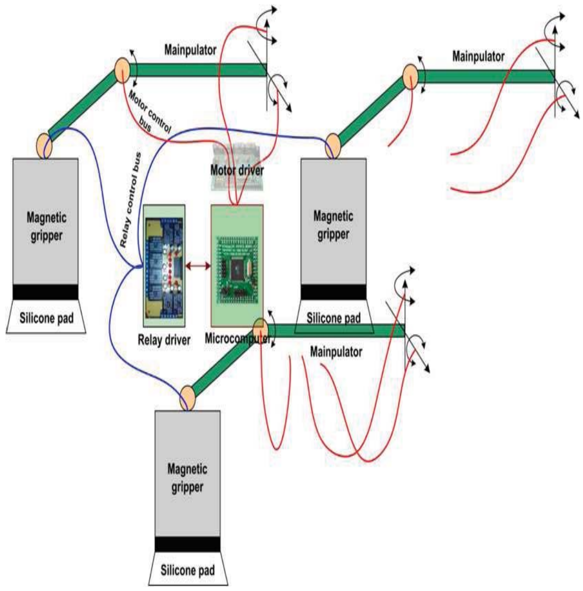


Figure 10. Electrical/electronic interface of gripper

III. Conclusion

In the pre-finished color metal sheet of non-protecting film when the gripping tools are contacted with the surface of metal sheet the trace of holder or pad is remained on surface of metal sheet. To prevent this trace the non-contact cyclone vacuum pad is used. The method of three contact points is used to design gripping mechanism. To measure rotating angle at joint of single finger to be composed of 2 DOF mechanism the rotating potentiometer is attached on it. To grip and handle a various shape of automobile parts the flexible gripper system is developed. The developed flexible gripper system is designed to grip the various shape of automobile parts exactly by using the attached vacuum pad and the structure of gripping mechanism is flexibly changed by electrical actuator.

ACKNOWLEDGEMENT

This work is supported by the Technology Innovation Program funded by Ministry of Knowledge Economy (MKE, Korea).

REFERENCES

[1] A. Schiele, "Undesired Constraint Forces in non-ergonomic wearable Exoskeletons," Extended Abstract for IROS'07 Workshop on Assistive Technologies: Rehabilitation and Assistive Robotics.

[2] Joel C. Perry, Jacob Rosen, "Design of a 7 Degree-of-Freedom Upper-Limb Powered Exoskeleton," BioRobo 2006-The first IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics Pisa, Tuscany, Italy, February 20-22, 2006.

[3] N. Hogan, "Impedance control, an approach to manipulation: Part I-III", ASME Journal of Dynamics Systems, Measurement, and Control, Vol. 107, 1985, pp. 1-24.

[4] C. H. Park and K. T. Park, "Design and Kinematics Analysis of Dual Arm Robot Manipulator for Precision Assembly," INDIN2008 IEEE, Daejeon, Korea, 2008.

[5] K. T. Park and C. T. Park, "Performance Evaluation of Industrial Dual-Arm Robot," ICSMA2008 ICROS, Seoul, Korea, 2008.

[6] K. T. Park, D. H. Kim, S. J. Kim, H. M. Kim and C. D. Lee, "Tool Handling System Using Hydraulic Actuator," URAI2011 KROS, Songdo, Korea, Nov. 23-26, 2011, pp. 579-581.

A Simulation Platform Used for Cooperative Control of Multi-Robot System

Tao Zhang[1], Xiaqin Li[1], Weining Lu[1], Yi Zhu[1], Masatoshi Nakamura[2]
[1]Department of Automation, School of Information Science and Technology, Tsinghua University, Beijing 100084, China
[2]Institute of Systems Control, Saga University, Saga 840-0047, Japan

Abstract- **The purpose of this paper is to introduce a new simulation platform. By working with this platform, we can realize the simulation of cooperative control algorithms designed for multi-robot system more conveniently. Meanwhile, the success compiler of algorithms in this platform means that we can also implement these algorithms in real multi-robot system, such as Pioneer-3AT robots. To reach these goals, a double-layer structure simulation platform is proposed in this paper. The high layer is designed for achieve the convenience of multi-robot system simulation, which based on Python and MySQL, which is named as monitoring layer. The low layer, called executing layer, whose main feature is to realize the algorithms in virtual and real multi-robot systems and ensure the effectiveness of simulation.**

Key words: **simulation platform; multi-robot system; cooperative control**

I. Introduction

Most researches on multi-robot system just focus on theoretical study and computer simulation realization at present[1]. There are however some problems existing in simulation nowadays. Some algorithms which can be verified validity in simulation can not be applied successfully in the real multi-robot system. What's more, it is demanded to do simulation several times in order to analyze the effectiveness of some algorithms. It is a boring and time-wasting work to input initial working data into simulation systems again and again. Considering the issues mentioned above, we explore a new simulation platform for cooperative control of multi-robot system to overcome these problems.

By working with this platform, we can realize the simulation of cooperative control algorithms designed for multi-robot system more conveniently. Meanwhile, the success compiler of algorithms in this platform means that we can also implement these algorithms in real multi-robot system, such as Pioneer-3AT robots [2]. To reach these goals, a double-layer structure simulation platform is proposed in this paper. The high layer is designed for achieve the convenience of multi-robot system simulation, which based on Python and MySQL, which is named as monitoring layer. The low layer, called executing layer, whose main feature is to realize the algorithms in virtual and real multi-robot systems and ensure the effectiveness of simulation.

We choose the MobileSim as our basic simulation software. MobileSim is designed for simulating mobile robots and their environments. Besides, Pioneer-3AT mobile robots are our real multi-robot system.

II. System Architecture

We build a double-layer structure simulation platform to fulfill our goals. The top layer is monitoring layer and the bottom layer is executing layer.

A. Monitoring layer

This layer can assist us to get rid of repeated and tedious work when doing multi-repeated simulation. In this layer, we implement the convenience improvement of simulation, including inputting the initial data and recording last output data automatically.

B. Executing layer

The bottom layer focuses on ensuring the effectiveness of simulation, no matter in virtual environment or in real multi-robot system.

III. Monitoring Layer Implementation

We realize this layer with Python programming Language. Python [3] is high-level programming language whose standard library is large and comprehensive. Python can be a powerful glue language between languages and tools because of the wide variety of tools provided by its large standard library. By means of Python, we can assemble robot's control program, the communication of system process together to extend the performance of our simulation platform.

A. *Single-robot simulation*

For single-robot multi-repeated simulation, we can describe the procedures as following:

- Step 1: Put forward the algorithm and fulfill the programming work
- Step 2: Input initial data and start the simulation
- Step3: Recording the simulation's output
- Step4: If the program satisfies the terminating condition, end the simulation. If not, go to the step 2.

Our target is to execute step 2 and step 3 automatically during the process of simulation. At first, we store the initial data in a data document so as to we can read it convenient whenever we need. For example, considering the case shown in Fig.1, 100 pairs of initial conditions are as Table 1. The data are stored into the file mazeinput.

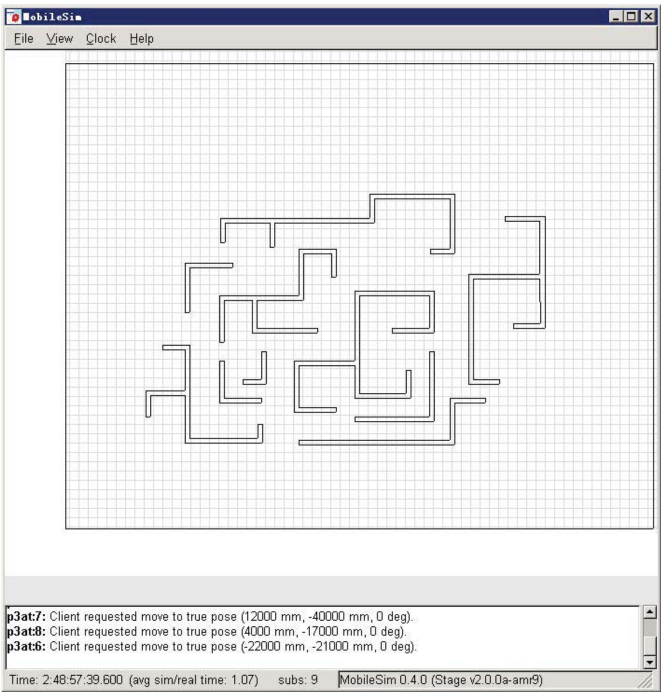


Fig.1 Maze map based simulation

Table 1 Initial conditions of Maze map.

编号	起点 X 坐标	起点 Y 坐标	初始朝向 Φ	目标点 X 坐标	目标点 Y 坐标
1	-14000	-23000	0	15000	-4000
2	-14000	-23000	0	22000	-7000
3	-14000	-23000	0	15000	-10000
4	1000	-27000	0	-15000	4000
5	8000	-30000	0	-22000	7000
6	1000	-33000	0	-15000	10000
7	-10000	-21000	0	11000	-6000
8	-10000	-21000	0	18000	-9000
⋮	⋮	⋮	⋮	⋮	⋮
100	4000	-41000	0	-8000	23000

This file may consist of virtual map information, start point coordinates and so on. We can do initial data input following the code show below:

```
``````
for x in mazeinput
alreadyDone = file("MAZEDONE").readlines()
if x not in alreadyDone
        open("TENP", "w").write(x)
        os.system("robotControl.exe < TEMP")
        sleep(3)
        open("MAZEDONE", "a").write(x)
`````
```

At last, we record the output data of simulation in a text file. In this way we can fulfill the repeated single-robot simulation in a convenient way.

However, this method can not apply into multi-robot system simulation successfully, Since this method do not ensure every robot to read the same initial information from the initial file or write the output data in the appropriate location in the recording file. What's worse, this method does not have the function which can active different robots. Therefore, we do some improvement to handle these issues.

## B. *Multi-robot simulation*

Multi-robot system's simulation is more complex than that of single-robot. For a multi-robot system which has *N* robots, we should bear inputting initial data for *N* times and some other repeated works. Although Mobilesim can support the work mode that one robot send orders to other robots, we abandon this mode because pioneer-3AT robots all equip with independent computer which means improper to that mode. So we construct a mechanism which allows to active multiple robots, input initial data and record output data automatically.

In order to resolve the data reading and storing problem, we introduce the MySQL database to record the data, the advantage of this database for our simulation is:

- Improve the ability of data sharing
- Ensure the consistence and integrity of data
- Enhance the independence of data

The first step is to input initial information to MySQL database, which include map information and coordinates of free points (not in the obstacle), then we program the code PyRandom.py to generate the initial data, which consist of start points' and end points' coordinates. The key code is:

```
`````
allResults = set()
while len(allResults)<100:   //100 is repeated times
    result = tuple(random.sample(allPoints,8)
    if resultIsOk(result):
        allResults.add(result)
    else
        print("NOT OK:{0}". Format(result));
``````
resultFileSQL = open("SimPointsSQL.txt", "w")
for result in allResults
      resultFileSQL.write("INSERT ONTO Maze ……
      (robot0StartX,resultFile. Close()
…….
```

In this simulation platform, we adopt the Client-Server mode which based on Python programming language to realize the communication and simulation automatically. So our two core codes in this platform are PyServerSim and PyClientSim.

The PyServerSim procedures can be abstracted into several steps:

- Step1:
  Connect to the MySQL database, meanwhile, make a judgment whether there is left task or not, if all tasks are completed, send an end command and quit. In our experiment, task means simulation.
- Step 2:
  Monitor the pyServerPort port, wait for PyClientSim connection. Once detect the signal 'Ready' comes from one of the PyClientSim, send a "wait" command back to it.
- Step 3:
  After receiving all 'Ready' signals from PyClientSim, send command 'Begin' to every robot and run the simulation.
- Step 4:
  Making sure that receive the 'Finished' signal transmit from PyClienSim, go back to the MySQL database and mark that we finish the simulation for one time. Then go to the Step1.

Similarly, we can list the work procedures of ClientSim,

- Step 1:
  Send 'Ready' signal to PyServerSim
- Step 2:
  Receive the command comes from the PyServerSim, if command is 'wait', the program will execute code 'sleep(0.1)' and go back to step 1. If command is 'begin', then go to the step 3.
- Step 3:
  Send a request to PyServerSim for the initial information of this simulation and acquire the initial data by connecting MySQL database.
- Step 4:
  Active the simulation program with the initial information given by MySQL.
- Step 5:
  After the program finishing, send signal 'Finished' to PyServerSim and store the data into the MySQL database.

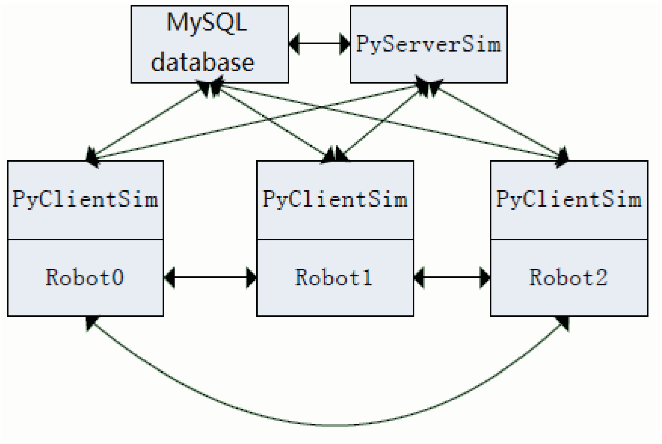



Fig.2 Client-Server Mode

The way how PyServerSim and PyClientSim to communicate with have been illustrated in the Fig.2.

So far, we have implemented our first goal mentioned in the head of this paper, which is realize the simulation of cooperative control algorithms designed for multi-robot system much conveniently. It also means that the construction of monitoring layer has been finished.

## IV. Executing Layer Implementation

As introduced at the beginning of this paper, we choose the MobileSim as basic simulation software and use Pioneer-3AT robots as real multi-robot system. MobileSim and Pioneer-3AT's development platform are both based on ARIA. So we can guarantee the consistency of simulation and experiment in real environment. They are binary compatibility. Firstly, we'll introduce the ARIA.

ARIA is developed for Mobile Robots, which can be used on Linux or Win32. ARIA is supported by C++ programming language, Java as well as Python. ARIA will provide rich library functions when we code the control program for Pioneer-3AT robots. Fig.3 is the structure of ARIA.

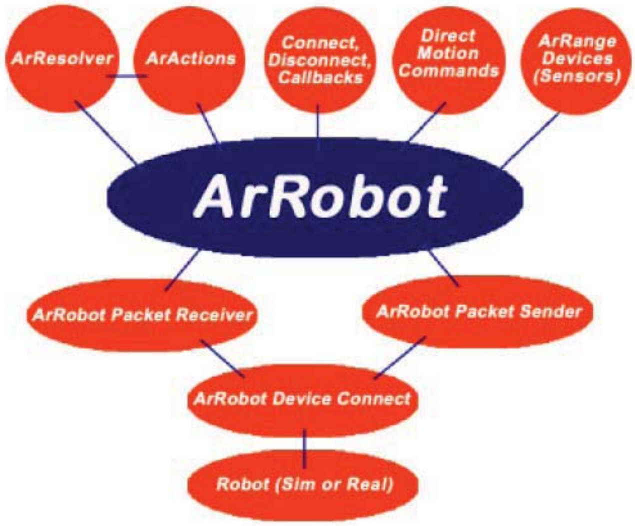



Fig.3 Structure of ARIA

After finishing the control program by using ARIA, we should realize the simulation in MobileSim, as illustrated by Fig.4. If it does work well, we may have a high possibility to gain the success in real multi-robot system.

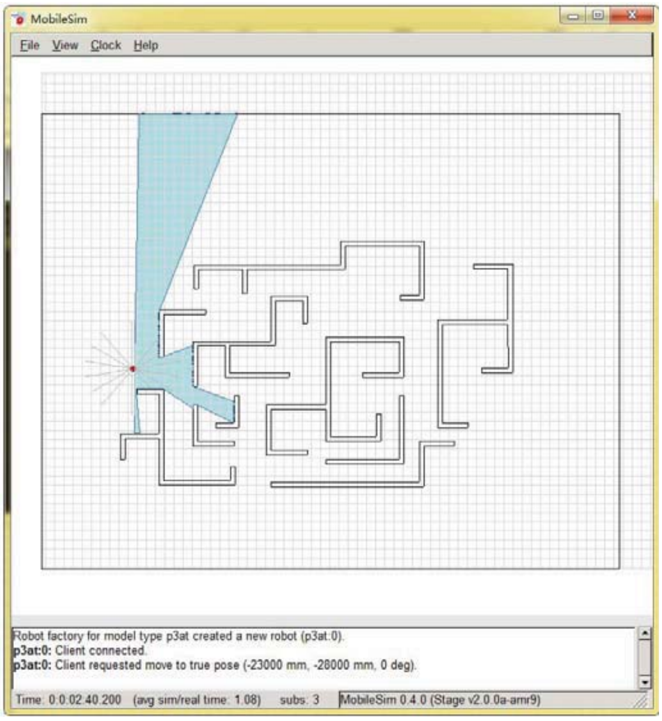



Fig.4 Interface of MobileSim

When get this point, we can say we have solved the second problem which puts forward at the beginning by combining MobileSim simulation software and Pioneer-3AT real robot system.

## V. CONCULSIONS

In this paper, we propose a new simulation platform which has a double-layer structure. The new simulation platform is mainly used for realize the simulation of cooperative control algorithms designed for multi-robot system conveniently and improve the consistency of simulation and experiment in real environment.

## REFERENCES

[1] Zhu A, Yang S, A survey on intelligent interaction and cooperative control of multi-robot systems, *Proceedings of IEEE International Conference on Control and Automation*, pp.1812-1817, 2010.
[2]Zhu Y, Zhang T, Song L, et al., A new bug-type navigation algorithm for mobile robots in unknown environments containing moving obstacles, *Industrial Robot – An International Journal*, vol.39, no.1, pp.27-39, 2012.
[3]Summerfield M., Programming in Python 3: a complete introduction to the Python language ($2^{nd}$ Edition), USA: Addison-Wesley Professional, 2009.

# Multi-layered Global Path Planning to Solve Path Mismatch Problem of Integrating Global Path Planning and Local Path Planning

Joon-Hong Seok, Joon-Woo Lee, Jeong-Hyun Wang, Ju-Jang Lee
Department of Electrical Engineering and Robotics Program
Korea Advanced Institute of Science and Technology
Daejeon, Korea
Email: seokjh@kaist.ac.kr, jwl@kaist.ac.kr, jh_wang@kaist.ac.kr, jjlee@ee.kaist.ac.kr

Ho Joo Lee
Agency for Defense Development
Daejeon, Korea
Email: hojoolee@yahoo.co.kr

***Abstract*—This paper proposes multi-layered global path planning for solving path mismatch problem in an integrated path planning. Local path planner can avoid obstacles within the range of built-in sensors, but it is impossible to generate efficient path out of the sensor range. Global path planner can generate efficient global path with built-in global map in a target region, but fidelity is very low because of the absence of local sensors. Simple combination of the global path planner and local path planner causes path mismatch problem due to the difference of information acquired from local sensor and information of preliminary global map. When some wrong global waypoints which are caused by low fidelity information or the change of terrain are given to robots, the proposed method can escapes a standstill of mobile robots using multi-layered map and automatic layer transition. When there is no problem to follow global waypoints, the proposed method tries to search global waypoints in a detailed layer map for acquiring more optimal paths as much as possible. The results of simulations show the better performance of multi-layered global path planner by comparing with other algorithms.**

## I. Introduction

The path planning has been one of the important research issues in the autonomous navigation. Path planner can be classified into global path planner (GPP) and local path planner (LPP). GPPs plan an optimal path using preliminary information, i.e., given terrain information, start point, goal point and obstacle information. A cost of each path is evaluated with terrain information and distance as objective functions. Cost commonly depends on whether global map is composed as binary or real values. In contrast, LPPs plan a reactive path using sensory information from range sensors attached to mobile robots or cars to avoid collision with obstacles.

GPPs can provide efficient paths using preliminary information, but global map may have low fidelity. A high-resolution global map generates more optimal paths, but higher computational cost is needed to process excessive map data to find optimal paths. In addition, map building also suffers from resolution specification. Last problem is that there are often environment change after map building procedure. In the worst case, GPPs lead mobile robots to non-traversable region and navigation objectives may be failed [1].

LPPs can provide feasible path from start point to goal point without any collision with obstacles [2]. The range sensor attached to mobile robot can limited information within a sensing range, but fidelity is higher than preliminary information used in GPPs. Most LPPs use the goal point to lead and proceed the direction and location of mobile robots. If there are no obstacles blocking the mobile robots, the final path is given as a straight line from the start point to the goal point. In fact, the region out of the sensing range of range sensors attached to robots is unknown, thus LPPs can find feasible paths without collision, but it is unavoidable that inefficient routes are generated.

A simple and intuitive approach to solve above problems, we let both path planners combine each other [3], [4]. GPPs generate intermediate subgoals or waypoints from the global map and LPPs generate inter-paths between intermediate waypoints. Waypoints generated from GPPs provide efficiency and shorter feasible paths generated from LPPs lose optimality at least. But, there are trade-off between large but inaccurate preliminary information and accurate but small sensing area of sensory information. Sometimes, these two different information cause inconsistent with each other. When the next global waypoint is blocked from found unrecorded obstacles by the range sensor, the refinement is not easy decision, because we cannot carry conviction with both global map and sensory information and whether which information is correct or not. Generally, ignoring a GPP and following a LPP are commonly is used but the efficiency is lost.

In this paper, A multi-layered integrated path planner (MLIPP) is proposed. Each layer map composition is explained from a base-level map to a highest-level map. In order to perform map level transition, adaptive level transition is introduced to increase the quality of generated paths automatically. An upward transition avoids getting stuck in local cell due to path mismatch problem. In contrast, a downward transition tries to find more optimal path by providing detail information from the global map. The resolution of an obstacle map used in LPP is used as a base level map. A higher-level map composition is based on a lower-level map. In a highest-level map, multi-layered integrated path planner is equivalent as LPP. When the fidelity of global map is very poor, finding any feasible path should be considered, thus the found highest-level map through upward transition is needed.

The remainder of this paper is organized as follows. Section II presents the definition of the path mismatch problem between GPP and LPP. Section III explains the proposed multi-layered global map, integrated path planning with adaptive level transition. Section IV shows the simulation results in various maps. Section V concludes this paper.

## II. Path Mismatch Problem

Path mismatch problem is caused by the information mismatch between GPP and LPP.

### A. Global Path Planning

GPPs are used to calculate optimal path according to defined objective functions. GPPs plan paths based on a global map. Global maps have own unique costs related with distance, terrain information, risk factor, etc. Any wide region also can include to calculate optimal paths using preliminary information. However, the fidelity of the information is not much because there are no sensors or observers whether the preliminary information is right or not in the mobile robots far from the interesting region.

Fig. 1(a) shows the satellite picture around KAIST and Fig. 1(b) shows the raster map around KAIST to represent roads, building, mountains, rivers and various terrains [5]. Fig. 1(c) and 1(d) show the real-valued cost map around KAIST with unit cells by 10m and 50m respectively to evaluate the risk factor about various terrains based on the feature database map [6]. Fig. 1(c) shows the detail information about the terrain information, but fidelity is restricted due to many changes according to the time and heavy computational time is required. In contrast, Fig. 1(d) shows the approximate information using light computational time, but the mobile robot cannot depend on this map totally for performing missions.

In summary, the global map provides approximate information regardless of the limit of the sensing range of the physical sensors to help optimal navigation of mobile robots. But, the fidelity is not much and it is not enough to perform missions without the collision with obstacles out of the global maps due to the small size or time change.

### B. Local Path Planning

LPPs provides the detail information to the mobile robots to avoid collision with obstacles. Although any optimal paths are given, when the mobile robots collide with other objects and the operation is halted or the robots are broken, the mission perform is finally failed. Undoubtedly, LPPs with various sensors include in the entire path planning structure, but the weaknesses of LPPs also exist. In the narrow indoor region, we can perform the map building using range sensors and this map information is used to path planning directly. However, in the outdoor environment, the target region is very wide compared with the sensing range attached to the mobile robots. The detail

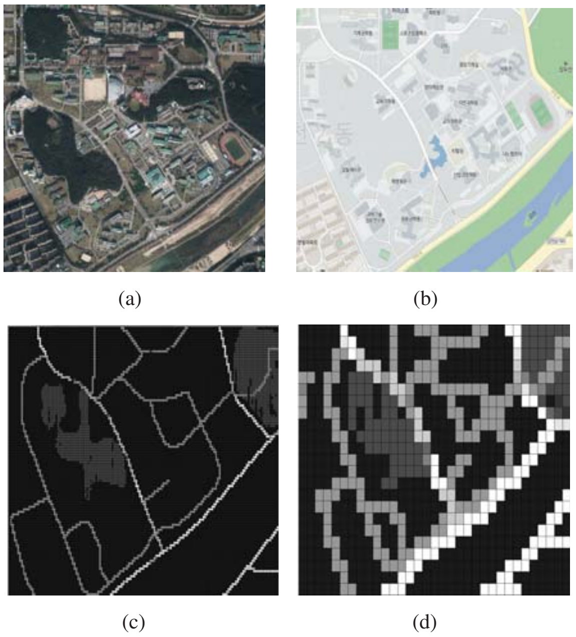



Fig. 1. (a) The satellite picture around KAIST. (b) The raster map around KAIST. (c) The real-valued cost map around KAIST with unit cells by 10m. (d) The real-valued cost map around KAIST with unit cells by 50m.

information of LPPs cannot be represented and stored in the memory due to huge size of data. In addition, the map building can avoid getting stuck in local region, but it is still so far from the optimal path. When we assume that the map building is not performed by the mobile robots due to heavy computational cost, the oscillation of the mobile robots can be happened.

Fig. 2(a) shows a robot with predefined sensing range, obstacles and a goal point. In the view of GPP's, the entire region can be prospected. However, in the view of the mobile robot, the region out of the sensing range is unknown. When considered with the given goal point, the robot oscillates around the U-shape obstacle in Fig. 2(c) and 2(d), global view and local view of the robot respectively.

### C. Integrated Path Planning and Path Mismatch Problem

In the previous section, the need of integrated path planning (IPP) is obvious. One of the most simple strategies is to combine both GPPs and LPPs. Optimal waypoints are extracted from the result of the GPPs and they are provided into the subgoals of the LPPs. These approaches are often adopted to compose IPP. However, there is a problem when both planners are combined. The sensory information and preliminary global information are not consistent with each other. For instance, in the global map, there is a traversable route between two obstacles. However, when the mobile robot approaches near the route, the sensor detects some obstacle blocking the route. In the case, the mobile robot cannot proceed to next waypoints generated from the GPPs. Instead, LPPs try to find a detour to approach the next waypoints. This is a path mismatch problem due to information inconsistency.

To solve the path mismatch problem, many path planners change the cost of blocked cell or route as non-traversable

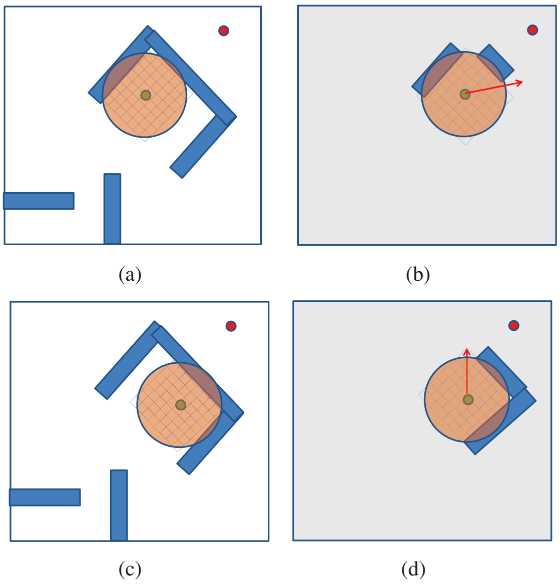



Fig. 2. (a) The global view point of the GPP in the first step. (b) The local view point of the robot in the first step. (c) The global view point of the GPP in the second step. (d) The local view point of the robot in the second step.

cost in the global map, i.e., D* Lite or any other planning method to revise premature global paths and depend on the LPP's ability to find feasible paths to the goal point.

However, two perspectives should be considered. First, when the mobile robots moves the different cell, not the cell which includes the next waypoint, LPPs cannot decide whether there is no feasible path from the current cell to the cell which includes the next waypoint. In addition, even though the mobile robot cannot reach the planned waypoint, the path planners cannot change the cost of the cell in the global cost map. Because the sensing range of LPPs is limited, the cost change decision can be premature. Second, when there are environmental changes, the previously calculated global cost is not valid to provide the optimal path. As previously mentioned, the LPPs can find feasible paths to the goal point, but the optimality is not preserved except for an aim for the shortest path. Therefore, the global map should not be modified carelessly, but the global waypoints should be calculated according to the current movement of the mobile robots according to the operation of the LPP.

## III. Proposed Multi-layered Integrated Path Planner

This section introduces MLIPP to solve the path mismatch problem when the information inconsistency is appeared between GPPs and LPPs. There are two properties to represent efficient IPP. First is the adaptive waypoint check distance. Second is the multi-layered IPP using various global maps.

The adaptive waypoint check distance scheme leads to avoid oscillation due to disruptive global waypoint change in the narrow region. For instance, when next global waypoints are located behind the wall, the mobile robot should detour the wall to get the next global waypoint and final goal point. Otherwise, the later global waypoints obstruct the robot's movement into wrong way. In contrast, in the wide region, finding a feasible path between the global waypoints is much easier than the narrow region. In summary, the adaptive waypoint check distance help to find the next waypoint carefully in the narrow region and to find the more optimal and shortest path to the goal point in the wide region by checking intermediate waypoints. Fig. 3 shows the relation of the adaptive waypoint check distance and nearby obstacles.

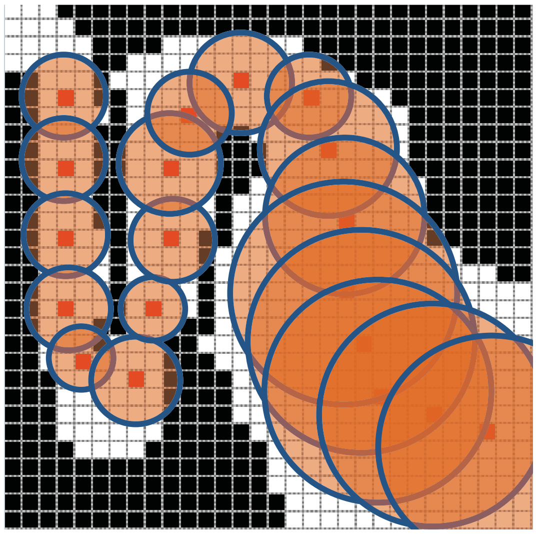


Fig. 3. Illustration of adaptive waypoint check distance scheme

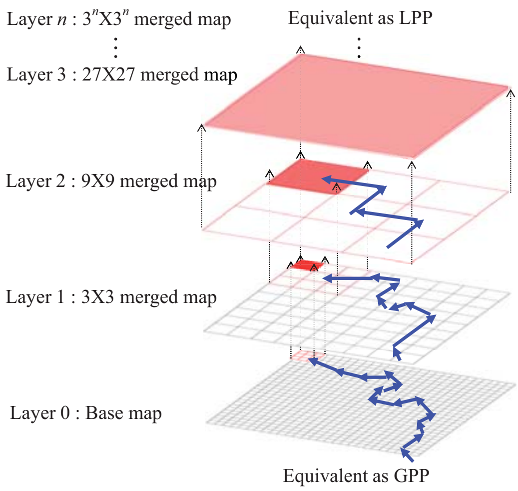



Fig. 4. Illustration of multi-layered integrated path planner

In spite of adaptive waypoint check distance scheme, there remains critical problem: the environment change without the global map revision. When the global map is revised, D* Lite can find the new optimal paths by adjusting inconsistency in the cost map. However, when the global map is not revised, D* Lite cannot perform replan procedure. In order to find better paths without changing the global map, the multi-layered IPP is proposed. Generally, the computational cost of GPP is less than that of LPP. Therefore, global path planning is performed

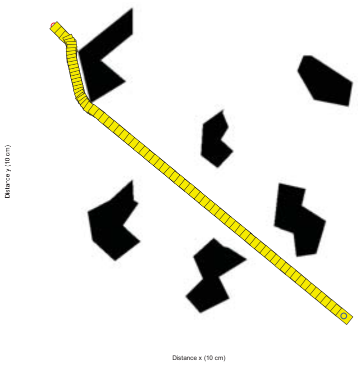



Fig. 5. The result path of LPP in the Map1.

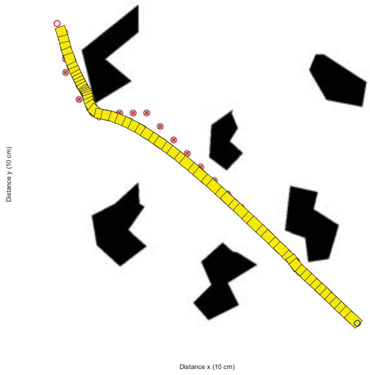



Fig. 6. The result path of IPP in the Map1.

on the all layers of the global map. In the base map, the size of each cell is same as that of LPP. Therefore, in the base map, the path planning result is equivalent as GPP totally. If the global map is fixed and there are no sudden eventuality, the result path in the base map is always optimal. In contrast, in the highest-level map, there are no global waypoint at all. It is equivalent as the result path of LPP. If there is no guarantee about the effectiveness of the preliminary global information at all, the mobile robot only depends on the given goal point and obstacles information within the sensing range of the sensor. As the level of the global map is increased, the number of global waypoints is reduced and the coarse information is produced through the GPP. When the fidelity is not much in the global information, the waypoints of higher level map are adopted. In contrast, as the level of the global map is increased, the number of global waypoints is increased and the fine information is produce through the GPP. When the fidelity is enough to plan the optimal path and there are risky terrains about LPP, the waypoints of lower level map are adopted. Fig. 4 shows the structure of the MLIPP.

## IV. Simulation Results

In this section, LPP, IPP and MLIPP are compared in various maps. Vector Field Histogram (VFH) [7] is used as LPP and D* Lite [8] is used as GPP. IPP combines both path planners with a single layer. MLIPP uses VTV and D* lite with the multi-layered global maps.

When the shortest path is similar with the optimal path, a LPP and an IPP show the similar results in Fig. 5 and 6. Because it is possible to find an optimal path in lower layer global map, IPP and MLIPP show identical results.

In the case of Map2, there is an U-shape obstacle. LPP tries to find a feasible path using sensory information and the given goal point. Because there is no map building procedure and no ability to plan out of the sensing range, the path oscillation can be shown in the U-shape obstacle in Fig. 7. In contrast, IPP can detect optimal waypoints using preliminary global map. The mobile robot detour an U-shape obstacle and successfully reach the goal point in Fig. 8. The result of MLIPP is also identical with the simple IPP. This example shows the superiority of IPP compared as LPP.

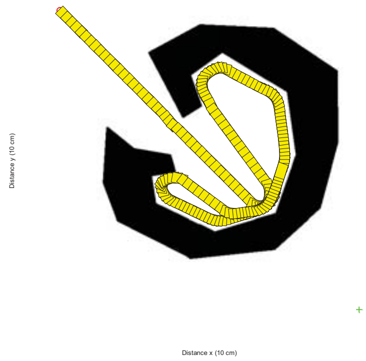



Fig. 7. The result path of LPP in the Map2.

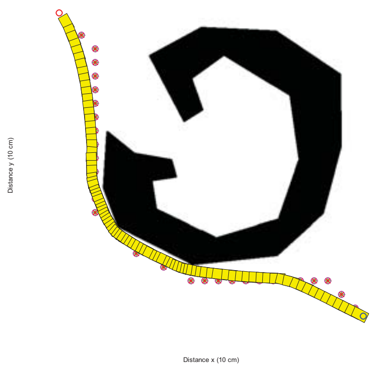



Fig. 8. The result path of IPP in the Map2.

In the Map3, the complex obstacle map is given. The result provides different characteristics compared with Map2. In Fig. 9, LPP finds a feasible path in spite of a lot of curves and detour paths. In contrast, Fig. 10 shows the detour path, because too strict waypoints may lead to wrong operation of LPP. Although the mobile robots moves to wrong direction, the global waypoint results is still valid, the robot can reach the goal point finally. Fig. 11 shows the different result. Because a lot of obstacles are detected within the sensing range, the waypoint reaching condition becomes more strict adaptively. So, without changing the layer of global map, the better path can be obtained.


Fig. 9. The result path of LPP in the Map3.

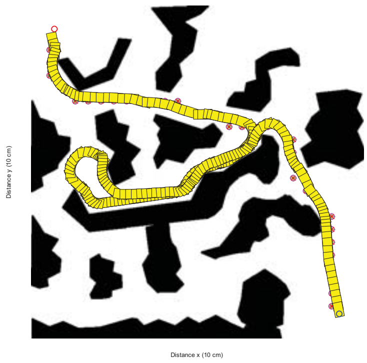



Fig. 10. The result path of IPP in the Map3.

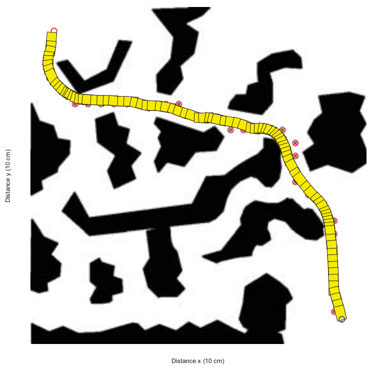



Fig. 11. The result path of MLIPP in the Map3.


Fig. 12. The result path of LPP in the Map4.

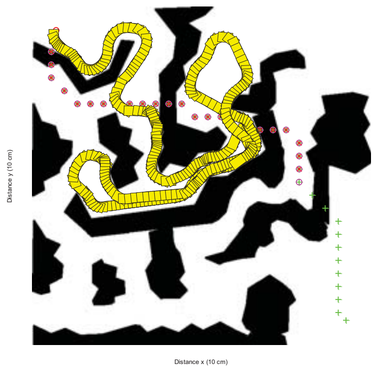



Fig. 13. The result path of IPP in the Map4.

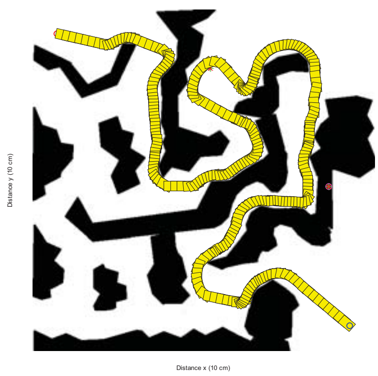



Fig. 14. The result path of MILPP in the Map4.

In the Map4, assume that environmental changes are appeared in the Map3. When the preliminary global map is composed, there are no blockade between obstacles. When the robot approaches the unknown blockade, the sensor attached on the robot can detect the route is non-traversable. Because the global map cannot be updated, the result of the LPP and simple IPP show the path oscillation within the map in Fig. 12 and 13. In that case, MLIPP realizes that the mobile robot moves by different direction compared with the next global waypoint. Therefore, the global waypoints from the higher level of the global map which is more coarse than lower level is called. Finally, the feasible path without any oscillation can be found in the only MLIPP result as shown in Fig. 14.

## V. Conclusion

This paper proposes multi-layered global path planning for solving path mismatch problem in an integrated path planning. Local path planner can avoid obstacles within the range of built-in sensors, but it is impossible to generate efficient path out of the sensor range. Global path planner can generate efficient global path with built-in global map in a target region, but fidelity is very low because of the absence of local sensors. Simple combination of the global path planner and local path planner causes path mismatch problem due to the difference of information acquired from local sensor and information of preliminary global map.

When some wrong global waypoints which are caused by low fidelity information or the change of terrain are given to robots, the proposed method can escapes a standstill of mobile robots using multi-layered map and automatic layer transition. When there is no problem to follow global waypoints, the proposed method tries to search global waypoints in a detailed layer map for acquiring more optimal paths as much as possible. The results of simulations show the better performance of multi-layered global path planner by comparing with other algorithms.

## VI. ACKNOWLEDGMENTS

The Authors gratefully acknowledge the support from UTRC(Unmanned technology Research Center) at KAIST(Korea Advanced Institute of Science and Technology),originally funded by DAPA, ADD.

## REFERENCES

[1] S. Simhon and G. Dudek, “A global topological map formed by local metric maps,” in *International Conference on Intelligent Robots and Systems*, vol. 3, 1998, pp. 1708–1714.

[2] A. Yahja, S. Singh, and A. Stentz, “An efficient on-line path planner for outdoor mobile robots,” *Robotics and Autonomous systems*, vol. 32, no. 2, pp. 129–144, 2000.

[3] C. Castejón, D. Blanco, and L. Moreno, “Compact modeling technique for outdoor navigation,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 38, no. 1, pp. 9–24, 2008.

[4] A. Howard, H. Seraji, and B. Werger, “Global and regional path planners for integrated planning and navigation,” *Journal of Robotic Systems*, vol. 22, no. 12, pp. 767–778, 2005.

[5] (2012) Daum map at kaist. [Online]. Available: http://local.daum.net/map/index.jsp

[6] C. Oh, B. Choi, J. Kim, J. Lee, and H. Lee, “Modified a* algorithm for outdoor environments with risk and velocity map,” in *International Symposium on Industrial Electronics*, 2009, pp. 1587–1591.

[7] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobile robots,” *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 278–288, 1991.

[8] S. Koenig and M. Likhachev, “Dˆ* lite,” in *Proceedings of the National Conference on Artificial Intelligence*, 2002, pp. 476–483.

# *Beacon Selection Algorithm for Localization of a Mobile Robot*

Jaehyun Park
Dept. of Electrical Engineering
Pusan National University
Busan, Republic of Korea
jae-hyun@pusan.ac.kr

Jangmyung Lee
Dept. of Electrical Engineering
Pusan National University
Busan, Republic of Korea
jmlee@pusan.ac.kr

***Abstract*—This paper proposes a localization scheme using ultrasonic beacons in an unstructured multi-block workspace. Indoor localization schemes using ultrasonic sensors have been widely studied due to their low price and high accuracy. However, ultrasonic sensors are susceptible to environmental noise due to the propagation characteristics of ultrasonic waves. On account of the decay phenomena of ultrasonic signals when they are transmitted over a long distance, ultrasonic sensors are not suitable for use in large indoor environments. To overcome these shortcomings of ultrasonic sensors while retaining their advantages, a multi-block approach was proposed by dividing the indoor space into several blocks with multiple beacons in each block. There should be a dynamic algorithm to divide into multiple blocks and to select suitable beacons. A beacon selection algorithm is developed to select the optimal beacons according to the robot's position and beacon arrangement for the mobile robot navigation. The performance of the proposed localization system is verified through real experiments.**

***mobile robot; localization;beacon selection algorithm; beacon;***

## I. INTRODUCTION

With the development of robot-related technologies such as microprocessors, sensors and computers, the practical uses of mobile robots have greatly increased. For service applications in various environments, position recognition, environmental perception using sensors, obstacle avoidance and path planning for autonomous navigation is vitally necessary. Especially, the precise localization of the mobile robot is one of the most important issues in the robotics field.

Localization schemes of mobile robots are classified into two categories, viz. relative and absolute position estimation. Relative position estimation is a dead reckoning method using odometry, gyroscopes and accelerometer sensors [1]. This method is easy and inexpensive to implement. However, with increase of driving distance, the accumulative error becomes larger, thus making precise localization more difficult [2]. The absolute position estimation obtained using GPS, lasers, vision, RFID, and ultrasonic sensors may resolve the problems of the relative position estimation [3]. However, the problems due to the uncertainty of the sensor information and the difficulty of installing the sensor are still remained.

This paper uses ultrasonic sensors for the indoor localization of a mobile robot. Ultrasonic sensors have been widely used, since they are inexpensive and easy to control. However, ultrasonic sensors are susceptible to environmental noise resulting from the propagation characteristics of the ultrasonic signals. On account of the decay phenomena of ultrasonic signals when they are transmitted over a long distance [4], ultrasonic sensors are not suitable for use in large indoor environments. To overcome these shortcomings of ultrasonic sensors while retaining their advantages, a new approach was proposed by dividing the indoor space into several blocks with multiple beacons in each block [5]. As a representative scheme, the block ID recognition scheme was developed by Ninety System to widen the operating area of the mobile robot [6].

The mobile robot is able to estimate its own position using beacon selection algorithm independent of the placement of the beacons. This paper is organized as follows. In section II, the indoor localization system, iGS, is introduced in detail, and the beacon selection algorithm in the multi-block environment is described in section III. In section IV, the effectiveness and usefulness of the beacon selection algorithm is verified by real experiments. Finally, section V concludes this research work.

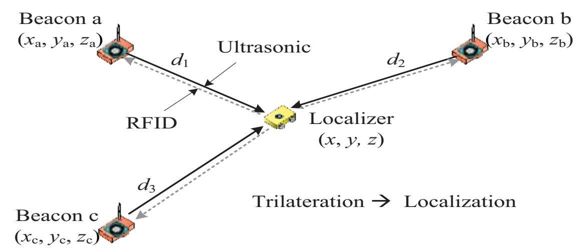



Figure 1. Beacons in the iGS space.

## II. IGS(INDOOR GPS SYSTEM)

The indoor localization system developed by Ninety System and the Intelligent Robot Laboratory of PNU are utilized as a platform for the mobile robot navigation. Its localization accuracy is ±5 cm and the localization period is 200 msec. The basic operating space is defined as 5 m (width) x 5 m (depth) x 2.5 m (height). Fig.1 illustrates the navigational environment of the mobile robot in the iGS space. There are at least three active beacon sensors in a room and one localizer on the mobile robot. The localizer calls a specific beacon to send ultrasonic signals which are received by the localizer to measure the distance to the beacon. When such distance data from three known beacon locations are available, the unknown location of the mobile robot can be estimated by the trilateration method [7].

## III. BEACON SELECTION ALGORITHM

The beacon selection algorithm proposed in this paper is based on the distance from the mobile robot to the beacon and geometrical arrangement of the beacons. In the case of iGS which uses a threshold value, all beacons have the same detectable range. Thus, it is possible to schedule the beacons according to the position of the mobile robot, if the mobile robot knows its own position, the position of the beacons and their detectable range. Since three distance data are enough to obtain the location by trilateration, the mobile robot needs to select three of the five beacons. To select the best three beacons, beacon selection algorithm composed geometric dilution of precision (GDOP) analysis and the distance sum (DS) analysis is proposed to be selected for the localization.

GDOP is commonly used in GPS and describes the effect of geometry on the relationship between measurement and position determination error [8]. The performance of the trilateration algorithm is affected by the ranging errors, by the geometrical arrangement of the beacons, and by the location of the mobile robot. Therefore, it is useful to know what is the best(lowest) attainable GDOP. A formal derivation of the GDOP relation begins with the linearization of the range measurement equation.

The Euclidean distance is usually used to calculate the distance between two points. Since the strength of ultrasonic signals decays according to the inverse square of the distance from the source, the error of measuring distance based on threshold value increases according to the distance. The error in a short distance is small and the error in a long distance is large. Therefore, the sum value of three Euclidean distance values from three positions is adopted to determine which beacons set has the smallest error.

In three dimensional spaces, the distance from the robot to the beacon is calculated as

$$d_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2} \tag{1}$$

where $(x_i, y_i, z_i)$ is position of the beacon bi and $(x_r, y_r, z_r)$ is position of the robot. The DS value of three distance values is obtained as

$$DS = \sum_{i=1}^{n} d_i \tag{2}$$

which can be used for selecting the beacons. A low DS value may have small position error of the robot, but a high DS value may have a large position error. Therefore, it is useful to distinguish what has small position error.

The GDOP values and DS values in the beacon selection algorithm are calculated from the positions of the mobile robot and the usable set of beacons. The most reliable subset of beacons has a low GDOP value, and this subset is selected using the beacon selection algorithm. Based upon these observations, we propose a beacon selection algorithm that uses the level of trust, which incorporates GDOP and DS values. The beacon-selecting algorithm uses the level of the trust, $f_i$, for each subset beacon useable in the robot coverage area. The value of trust indicator $f_i$, is defined as

$$f_i(B\_list, p_r) = k_1 \cdot GDOP_i + k_2 \cdot DS_i \tag{3}$$

where *B_list* are the useable beacons in the robot's coverage area and $p_r$ is the position of the robot. *GDOP* and *DS* represent the GDOP and DS values of the *ith* beacon subset, respectively. The *k* is a weighting, which is used to balance the level of importance desired for the DS term. The value of trust indicator $f_i$ depends on the values of GDOP and DS in each beacon subset.

A low value of $f_i$ means it has good geometry and small position error. On the contrary, a high value of fi means it has poor geometry and large position error. Fig. 2 summarizes the developed beacon selection algorithm.

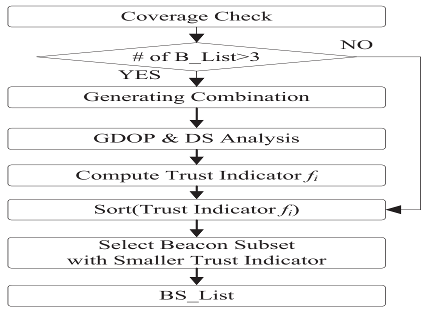



Figure 2. Beacon selection algorithm.

A coverage check is used to make the beacon list, B_List, which allows the mobile robot to utilize the beacons which are obtained by using the position of the mobile robot and coverage of the beacons. If there are more than three beacons, the possible combinations of beacons are generated to select the best subset of beacons, which has the smallest trust indicator value. For the calculation of the trust indicator fi, a GDOP Analysis is performed to calculate the GDOP values and a DS Analysis is calculating for DS values for the given the beacon subset. Then, B_List is sorted in ascending order of the trusts indicator $f_i$ and selection of beacon subset with the smaller trust indicator value is performed. Finally, the beacon selection list, BS_List, which has the optimal beacons geometrically, is returned for the precise localization.

## IV. EXPERIMENTS

The experiments were performed in an indoor environment, where eight beacons are deployed at the vertices (x, y)[units : mm] of the workspace containing the mobile robot, to prove the effectiveness of the algorithms proposed in this paper. Fig. 3 shows the deployment of the beacons and mobile robot. Seven beacons were deployed in the workspace. Before the experiment began, *Range error* and *k* in eq. 3 were determined. *Range error* was set at 22.5mm, and then *k* was calculated. *Range error*·GDOP means positional error, and therefore, *k* is obtained by the relationship between the positional error and the DS value. *k* calculated using

$$k = p_1 + \frac{p_2}{DS} \tag{4}$$

where $p_1 = 0.0045$ and $p_2 = 2.6636$.

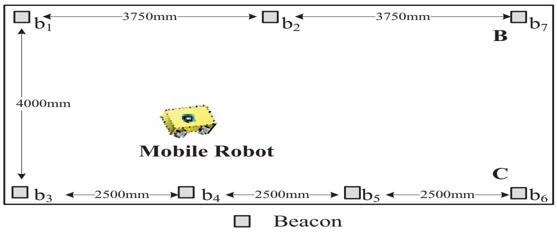


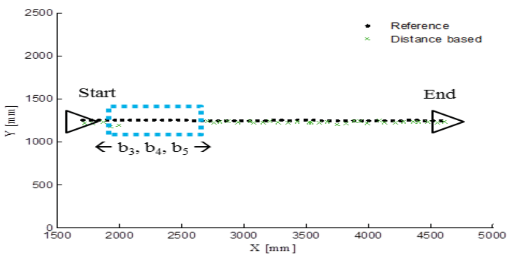

Figure 3. The deployment of the beacons and the route of the mobile robot.

To compare the performance of position estimation, distance and GDOP based beacon selection algorithms are used. In order to compare the effectiveness of the three-beacon selection algorithm, straight was used.

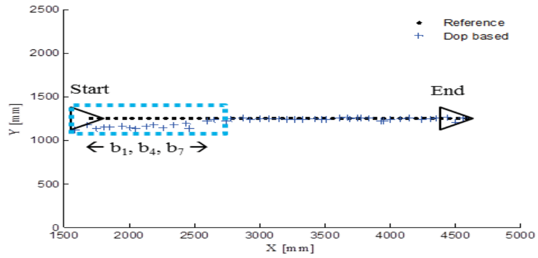
Fig. 4(a), (b) and (c) show the trajectory obtained using the three-beacon selection algorithm. In Fig. 4(a), trajectory loss occurred near the start point, since beacons $b_3$, $b_4$, and $b_5$ were used. These beacons cause a singularity problem because they are arranged along with X axis. The GDOP based beacon selection algorithm in Fig. 4(b) showed a large error at the start point, where the $b_1$, $b_4$, and $b_7$ beacons had the lowest GDOP values, on the other hand, the $b_7$ beacon had large distance error since it was located further away than the other beacons. On the other hand, the beacon trust indicator based selection algorithm produced positioning results with a lower position than the other algorithms because it considers both distance and GDOP.

## V. CONCLUSIONS

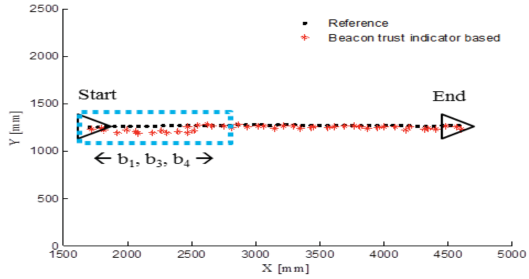
Ultrasonic sensor based localization systems have been widely used, since they are inexpensive, accurate and easy to control for indoor localization. However, on account of the decay phenomena of ultrasonic signals when they are transmitted over a long distance, ultrasonic sensors are not suitable for use in large indoor environments. For the real-time localization of a mobile robot, a beacon selection algorithm, which uses the trust indicator of beacon subset based on GDOP analysis and DS analysis, was developed. The effectiveness of the localization system is verified through real experiments of mobile robot navigation. This localization system can be applied to location based services in various fields, as well as to robot positioning systems.




(a) Distance based trajectory




(b) DOP based trajectory




(c) Beacon trust indicator based trajectory

Figure 4 The straight trajectory comparison

## ACKNOWLEDGMENT

"This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the Human Resources Development Program for Specialized Navigation/Localization Technology Research Center support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2012-H1502-12-1002)

## REFERENCES

[1] H. Myung, H. K. Lee, K. choi and S. W. Bang, "Mobile robot localization with gyroscope and constrained Kalman filter," International Journal of Control, Automation, and Systems, vol 8, pp. 667-676, 2010.

[2] C. C. Tsai, "A Localization System of a Mobile Robot by Fusing Dead-reckoning and Ultrasonic Measurements," Industrial Electronics, IEEE Transactions on, vol. 47, pp. 1399-1404, 1998.

[3] T. S. Abuhashim, M. F. Adbedl-Hagez and M. A. Al-Jarrah, "Building a Robust Integrity Monitoring Algorithm for a Low Cost GPS-aided-INS System," International Journal of Control, Automation, and Systems, vol. 85, pp. 1108-1122, 2010.

[4] Ching Chih Tsai, "A Localization System of a Mobile Robot by Fusing Dead-reckoning and Ultrasonic," IEEE Trans. on Instrumentation and Measurement, vol. 47, pp. 1399-1404, 1998.

[5] W. Qinhe and H. Hashimoto, "Fast Localization of Multi-targets in the Intelligent Space," Annual Conf. SICE. 2007, pp. 264-269, 2007.

[6] D. G. Seo and J. M. Lee, "Localization Algorithm for a Mobile Robot Using iGS," The 17th International Federation of Automatic Control World Congress, pp. 742-747, 2008.

[7] F. Thomas and L. Ros, "Revisiting Trilateration for Robot Localization," IEEE Trans. on Robotics, vol. 21, pp. 93-101, 2005.

[8] D. E. Manolakis, "Efficient Solution and Performance Analysis of 3-D Position Estimation by Trilateration," IEEE Transactions on Aerospace and Electronic Systems, vol. 32, pp. 1239–1248, 1996.