The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Navigation in Indoor Environment by an Autonomous Unicycle Robot with Wide-Type Wheel

Junpei Okumura, Toshinobu Takei and Takashi Tsubouchi

***Abstract*— The objective of this research is to perform a long distance navigation in indoor environment, with a unicycle robot that has a wide-type wheel, and has a laser range finder installed on top of the robot. At present, there is no report for long distance navigation by unicycle robot in real indoor environment. We have developed a system that consists of two subsystems; "Balancing and Maneuvering System" for moving robot to desired direction while the posture of the robot is stabilized, and "Navigation System" for calculation of desired direction and recognition of intersections, using external sensor whose scan plain changes when robot steers. We have implemented the system to a single SH-4 CPU microcomputer, and performed a long distance navigation in actual indoor environment. The paper reports the unicycle robot and the system we have developed, and the experiment result of long distance navigation in indoor environment.**

I. INTRODUCTION

The objective of this research is to perform a long distance navigation in indoor environment, with an unicycle robot. In this paper, we define a navigation as "movement of a mobile robot that drives itself autonomously so as to go along the given route, and avoid obstacles in static condition by using external sensors". In order to perform the navigation, acquisition of external environment data and the steering control of the robot are necessary. The robot we are going to present here has no such obvious steering mechanism as a bicycle has. The robot does control its steering by actuating its upper body, which makes it rotate against the lower body in sidewards. The acquired data from the external sensor would be affected by this motion, because it is installed on top of the robot. We must develop a navigation algorithm that takes in account of the effect when the upper body rotates, and also a balancing and maneuvering system for the robot.

Murata Manufacturing Co., Ltd. developed a mobile unicycle robot[6], whose posture stabilized by a flywheel installed inside the body. Honda Motor Company, Ltd. developed a personal mobility device[5] that has multiple small-diameter motor-controlled wheels connected in-line to form one large-diameter wheel, which enables it to move in all directions. Ferreira et al. developed a single-wheel robot that is stabilized and steered by means of an internal mechanical gyroscope[9]. Majima et al. have proposed a control method

J. Okumura is with Graduate School of System and Information Engineering, University of Tsukuba, Tenoudai 1-1-1, Tsukuba, Ibaraki 305-0006, JAPAN kuro.no.ro@gmail.com

T. Takei is with Graduate School of System and Information Engineering, University of Tsukuba, Tenoudai 1-1-1, Tsukuba, Ibaraki 305-0006, JAPAN takei@iit.tsukuba.ac.jp

T. Tsubouchi is with Graduate School of System and Information Engineering, University of Tsukuba, Tenoudai 1-1-1, Tsukuba, Ibaraki 305-0006, JAPAN tbu@esys.tsukuba.ac.jp

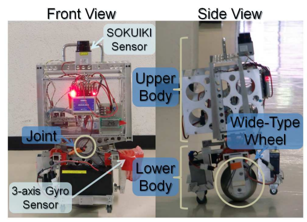

Fig. 1. The Unicycle Robot "Ichiro"

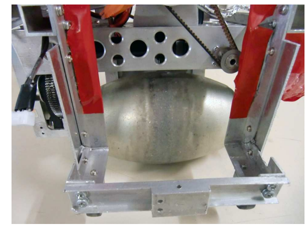

Fig. 2. The Wide-Type Wheel

for changing the yaw direction of a unicycle robot, which could not produce direct force to change the yaw direction, and performed a simulation. Previously mentioned robots do not have external sensors, and also the long distance navigation in real indoor environment is not realized at present. We have realized steering motion for the real unicycle robot with a simple control method, and performed a long distance navigation in indoor environment using a laser range finder.

We have developed a unicycle robot shown in Fig.1 and Fig.2, and installed a SOKUIKI sensor on top of the robot. We have developed a system that consists of two subsystems; "Balancing and Maneuvering System" and "Navigation System". Balancing and Maneuvering System moves the robot so as to move along specified direction and distance, while it keeps its posture stabilized. Navigation System calculates the steering direction of the robot, with the input of acquired data from SOKUIKI sensor. We propose a simple solution for the issue of sensor rotation problem when in steering motion, as shown in Fig.4, Fig.5 and Fig.6. We have performed experiment of long distance navigation in real indoor environment, a total distance of 470m. In this paper, the objective of the work is described in the following section. Environment, motion of the steering, influence to the acquired data from SOKUIKI sensor when in steering motion are explained. Secondly, developed system of the robot is described. Finally, the experiment for long distance navigation performed in Univ. of Tsukuba is described.

II. OBJECTIVE OF THIS STUDY

A. Robot

For the controller, we used a SH-4 CPU microcomputer board HRP-3P-CNA from General Robotix, Inc.[3]. The controller is small and energy eficient, and has a real-time linux OS installed. The posture of the robot, pitch angular velocity of the wheel, and angle between upper body and lower body are necessary information for the control. We installed a 3-axis gyro sensor on lower body for acquiring the posture of

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

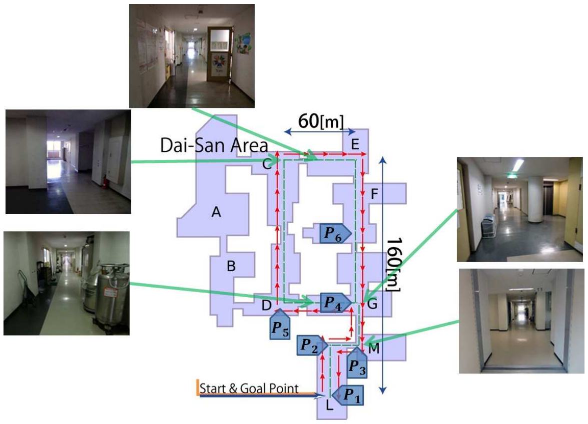

Fig. 3. Route: Dai-san Area in Univ. of Tsukuba (470m).

the robot. We use rotary encoder, that are installed at axle of motors, to acquire pitch angular velocity of the wheel and angle between upper and lower body. Thus we installed SOKUIKI Sensor at top of the robot, for acquiring external environment. Since the data acquisition of SOKUIKI Sensor and motion control of the robot is performed by a single SH-4 CPU microcomputer, the algorithm must be simple. The robot contains batteries which supply 24V, motors, and the controller, so as to move only by itself autonomously.

B. Environment

We set the task as navigation of the route described in Fig.3, a total length of 470m. The route consists of straight paths and three types of intersections. In order to achieve this task, the robot must turn to appropriate direction in each intersections, while avoiding obstacles found throughout the path.

The width of the straight path is approximately 2.5m, and there are places where obstacles such as shelves and gas cylinders are placed at the side of the path. Doors are open by the time, and the robot must avoid these obstacles. We assume that people walks along the side wall. There are L-intersections, T-intersections, and crossings in the route. In Fig.3, there is a L-intersection in building L and E, a crossing in building M and G, a T-intersection in building D and C. Also there are three T-intersections that appears in consequtive in point P_6. Two large spaces can be measured at T-intersections and crossings when scanned with SOKUIKI Sensor.

C. Steering Motion

The robot consists of two parts, upper and lower body. They are connected with a joint whose rotation axle is set perpendicular to the wheel axle ("Joint" in Fig.1). The rotation of the upper body against the lower body is actuated by a DC motor, which is placed inside of the lower body and above the wheel. Torque generated by the motor is transferred by a timing belt, which settles an angle between bodies. A roll angle of the wheel arises when an angle between the bodies is settled so that lateral displacement of the body centroid emerges. The robot rotates in yaw direction, when the wheel rotates in pitch and roll direction at same time.

D. External sensor

We installed SOKUIKI Sensor URG-04LX[1] from Hokuyo Co.[4] as an external sensor, on top of the robot as shown in Fig.1. The sensor scans 2D-dimensional plain, and determines the distance of objects in range. The sensor assumes in use with PCs, however we handle this sensor with SH-4 microcomputer board. Therefore there are some limitation in use of the sensor, as follows. Angular resolution is 3.60deg (68 points per scan), scanning time is 300ms, maximum measuring distance is up to 4095mm, and the measuring area is $\pm$120deg. Angle and distance data is acquired for each points. If the object is not measured, error code is acquired.

Scan plain changes as the robot steers, because the robot must rotate its upper body when in steering motion. Example of scanned data in various states are shown in Fig.4, Fig.5 and Fig.6. A sector in the right figure illustrates the scanned plain. Coordinates in the left figure shows measured distance data of scanned plain. The origin indicates the position of the sensor. Walls and the floor are plotted with a form of lines, however, they move for maximum of 2m in sidewards when the state of the robot changes. Therefore it is difficult to use these lines for driving along the wall.

Our approach for the navigation is to use error codes of the SOKUIKI Sensor. In the left figure, a sector illustrates the section where error codes are acquired in consecutive, and a line is drawn in middle of the sector. We confirmed that the direction of the line would not change drastically, despite changes of states.

III. System configuration

We have developed the navigation system that consists of "Balancing and Maneuvering System" and "Navigation System". System configuration is described in Fig.7.

1) Balancing and Maneuvering System: The objective of Balancing and Maneuvering System is to control rotation of the wheel and the angle between bodies, so as the robot to move itself in steering direction ζ_{ref} and reference velocity v_{ref} given by Navigation System. The system consists of three subsystems; Steering Controller, Controller of pitch angular velocity of the wheel, and Controller of roll angle of the wheel. Steering Controller calculates the reference states of the wheel, pitch angular velocity $\dot{\theta}_{ref}$ and roll angle α_{ref}, when steering direction ζ_{ref} and reference velocity v_{ref} are given. Controller of pitch angular velocity of the wheel drives the wheel so as robot to follow $\dot{\theta}_{ref}$ while it keeps its posture stabilized. Controller of roll angle of the wheel actuates upper body so as the robot to follow α_{ref}. For control of both states of the wheel, state feedbacks are performed. Outputs of the system are calculated in every 2ms.

2) Navigation System: The objective of Navigation System is to calculate steering direction ζ_{ref} and reference velocity v_{ref}. Appropriate steering direction for intersections and for obstacle avoidance is needed in order to achieve the

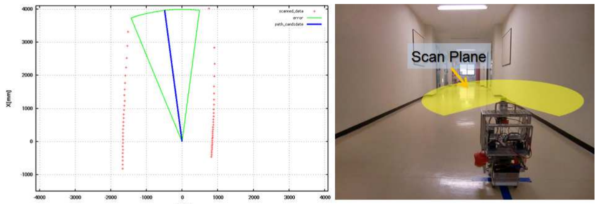

Fig. 4. Scanned plain of the SOKUIKI Sensor (right) and acquired data (left) when the upper body is set straight.

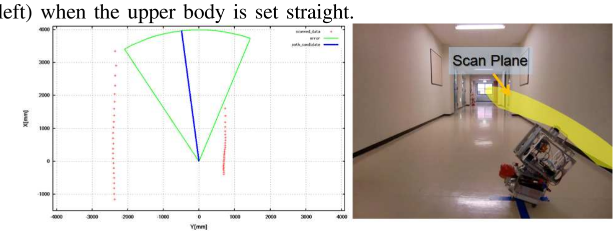

Fig. 5. Scanned plain of the SOKUIKI Sensor (right) and acquired data (left) when the upper body is inclined to the right.

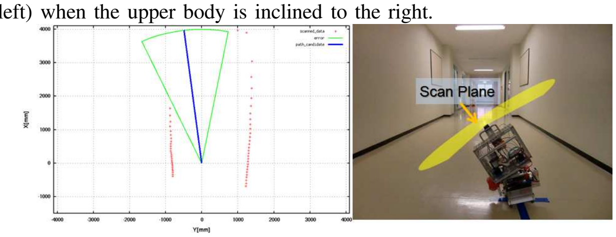

Fig. 6. Scanned plain of the SOKUIKI Sensor (right) and acquired data (left) when the upper body is inclined to the left.

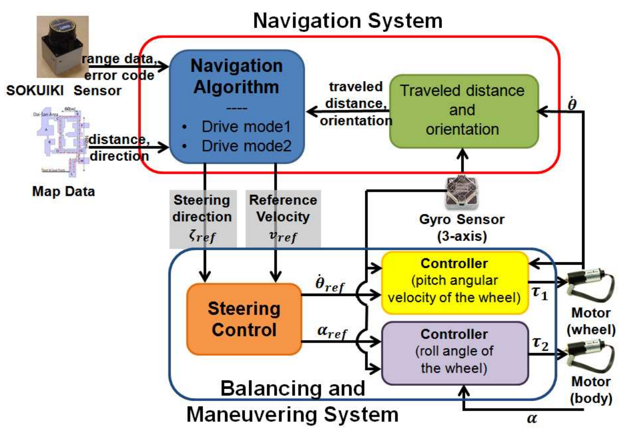

Fig. 7. System Configuration.

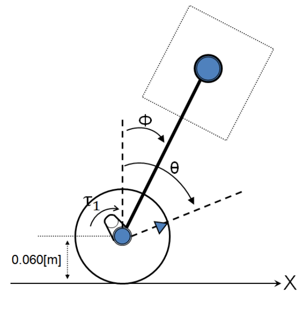

Fig. 8. Dynamics model of the motion (for sagittal direction).

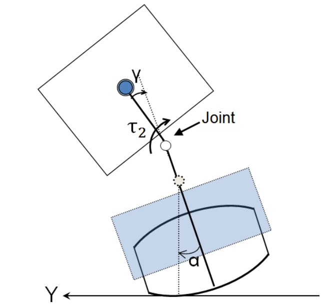

Fig. 9. Dynamics model of the motion (for the lateral direction).

navigation task. The system provides two driving modes, that corresponds to straight paths and intersections. For straight paths, steering direction ζ_{ref} is set to the direction where no obstacles were found by using SOKUIKI Sensor. For intersections, combination of SOKUIKI Sensor and the map data are used to determine ζ_{ref}. Reference velocity v_{ref} is calculated by the correspondence of the distance to the nearest obstacle. ζ_{ref} and v_{ref} are calculated in every 300ms.

IV. Balancing and Maneuvering System

A. Dynamic Model for Balancing and Maneuvering System

There are two torque input; torque τ_1Nm for the revolution of the wheel which is involved in the movement of sagittal direction, and torque τ_2Nm for actuating upper body which is involved in the movement of lateral direction. We have used dynamic models[2] for movements of sagittal direction and lateral direction, under assumption that the robot would not change its posture drastically from the stable state while its balance is kept. We handle these models as a movement of double inverted pendulum in 2 dimensional plain, and as independent movements.

For the sagittal direction, we used the model shown in Fig.8. If we assume that the robot would not change its posture drastically from the stable state($\phi \approx 0$), linearized equations of motion(1)(2) are derived.

$$a_{11}\ddot{\phi} + a_{12}\ddot{\theta} + a_{13}\phi = b_1\tau_1(t) \quad (1)$$

$$a_{21}\ddot{\phi} + a_{22}\ddot{\theta} + a_{23}\phi = 0 \quad (2)$$

For the lateral direction, we used the model shown in Fig.9. If we assume that the robot would not change its posture drastically from the stable state($\alpha \approx 0, \gamma \approx 0$), linearized equations of motion(3)(4) are derived.

$$a_{31}\ddot{\alpha} + a_{32}\ddot{\gamma} + a_{33}\alpha + a_{34}\gamma = b_2\tau_2(t) \quad (3)$$

$$a_{41}\ddot{\alpha} + a_{42}\ddot{\gamma} + a_{43}\alpha + a_{44}\gamma = 0 \quad (4)$$

Coefficients in (1)(2)(3)(4) are described in [7]. Description is omitted in this paper since the space is limited.

B. Balancing and Maneuvering

The objective of Balancing and Maneuvering System is to calculate reference states (pitch angular velocity of the wheel $\dot{\theta}$ and roll angle α_{ref}), and to follow these reference states. We use state feedback control (5)(6) for this issue.

$$\begin{aligned} \tau_1(t) = &-k_{11}\phi(t) \\ &-k_{12}\dot{\phi}(t) \\ &-k_{13}\left(\dot{\theta}_{ref} - \dot{\theta}(t)\right) \\ &-k_{14}\int_0^t \left(\dot{\theta}_{ref} - \dot{\theta}(t)\right) dt \end{aligned} \quad (5)$$

$$\begin{aligned} \tau_2(t) = &+\tau_{2s} \\ &-k_{21}\left(\alpha(t) - \alpha_{ref}\right) \\ &-k_{22}\left(\gamma(t) - \gamma_{2s}\right) \\ &-k_{23}\dot{\alpha}(t) \\ &-k_{24}\dot{\gamma}(t) \end{aligned} \quad (6)$$

where,

$$\begin{bmatrix} \gamma_{1s} \\ \tau_{2s} \end{bmatrix} = -\begin{bmatrix} a_{34} & -b_2 \\ a_{44} & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_{33} \\ a_{43} \end{bmatrix} \alpha_{ref} \quad (7)$$

$$\gamma_{2s} = \gamma_{1s} - k_{GM} v(t)\omega(t) \quad (8)$$

Equation(7) is the calculation of (γ, τ_2) when reference roll angle of the wheel α_{ref} is given, in stationary condition. However, (7) becomes untrue when the robot turns and centrifugal force have worked on the robot. We have added (8) to deal with the force by feedback control (the more yaw angular velocity ω and velocity v increases, the more upper body inclines). In (5)(6), $(k_{11} - k_{24})$ are feedback gains, and k_{GM} is the coefficient. These constant values are calibrated so as the robot to follow reference states with less vibration.

C. Steering Control

Steering Control first calculates the reference roll angle of the wheel α_{ref} and angular velocity $\dot{\theta}_{ref}$ from input (steering direction ζ_{ref} and reference velocity v_{ref}), then perform feedback control (5)(6) by using α_{ref} and $\dot{\theta}_{ref}$. We found out that α_{ref} would not change drastically from stable posture ($\alpha = 6$deg in maximum), while the balance is kept. Therefore, regarding that $\alpha \approx 0$, $\dot{\theta}_{ref}$ is derived as follows.

$$\dot{\theta}_{ref} = v_{ref}/0.060[m] \quad (9)$$

Constant value in the (9) is radius of the wheel.

For the calculation of α_{ref}, following procedure is performed.

1) Reference yaw angular velocity ω_{ref} is calculated by using steering direction ζ_{ref}.
2) α_{ref} is calculated by using ω_{ref} and v_{ref}.

For the calculation of ω_{ref}, state feedback control is used. When the yaw angular velocity of the robot at present state is expressed as $\omega(t)$, moving direction as $\zeta(t)$, ω_{ref} is calculated as follows.

$$\omega_{ref}(t) = -k_{31}(\zeta(t) - \zeta_{ref}) - k_{32}\omega(t) \quad (10)$$

For the calculation of α_{ref}, (11) is used.

$$\alpha_{ref} = \begin{cases} \frac{3}{10}\omega_{ref}(v \leqq 0.3[m/s]) \\ \frac{3}{19}\omega_{ref}(v \geqq 0.3[m/s]) \end{cases} \quad (11)$$

Nakajima et al. [2] proposed a hypothesis model that the circling of the wheel with rolling angle α is equivalent to the rolling of a cone, and used it for calculation of ω. However, this model does not take account of centrifugal force that works on the robot. Therefore we decided to perform experiments to construct a new model of the relationship among v, α, and ω, by using feedback control(5)(6). Following feedback controls were performed in experiments; control of $\dot{\theta}$ so as to follow v_{ref} for the sagittal direction, and control of α to follow α_{ref} for lateral direction. α_{ref} was given from 0 to 6deg, increasing the value by 1deg every 10 seconds. Similar experiments were performed, for $v_{ref} = 0.2, 0.3, 0.4, 0.5, 0.6$. A result of the experiment when $v_{ref} = 0.6$m/s is shown in Fig.10. In Fig.10, although oscillations that is caused by friction effect are found in α, α follows α_{ref} in average. Also the figure indicates that α is propotional to ω. Equations (11) are derived from the result of experiments.

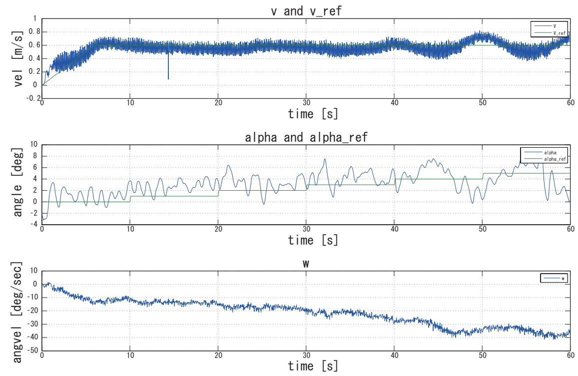

Fig. 10. The relationship among v, α, and ω, when $v_{ref} = 0.6$ m/s

V. Navigation System

A. Driving Modes

The system provides two driving modes, that corresponds to straight paths and intersections.

Mode1 is the driving mode that corresponds to straight paths. Robot seeks for the largest open space by using SOKUIKI Sensor, and sets the direction as steering direction ζ_{ref}. The maximum reference speed is 0.6m/s, and is adjusted by the correspondence of the distance to the nearest obstacle.

Mode2 is the driving mode that corresponds to intersections. ζ_{ref} is determined by the map data and acquired data from SOKUIKI Sensor. The maximum driving speed is 0.25m/s, and is adjusted by the correspondence of the distance to the nearest obstacle.

It is necessary to switch driving modes in appropriate timing, in order to achieve the navigation task. One easy way for the timing is to use traveling distance. However, the traveling distance may change drastically among trials of the navigation, especially when robot had avoided obstacles. In this research, we use acquired data from SOKUIKI Sensor in addition to determine the appropriate timing. Following are conditions for the system to switch the driving mode from mode1 to mode2.

- The robot had moved the specified distance in Map Data (distance between each intersections).
- At least two open space that are larger than the threshold are found by using SOKUIKI Sensor. The threshold is adjusted in actual experiments.

For the switching from mode2 to mode1, robot only uses traveling distance.

B. Mode1: for straight paths

The procedure of how to determine steering direction and reference velocity is illustrated in Fig.11. Acquired data from SOKUIKI Sensor is used for the determination of direction and velocity. Scan range of the sensor is ±120, and maximum measuring distance is up to 4095mm. Angle and distance data is acquired if there is an object within

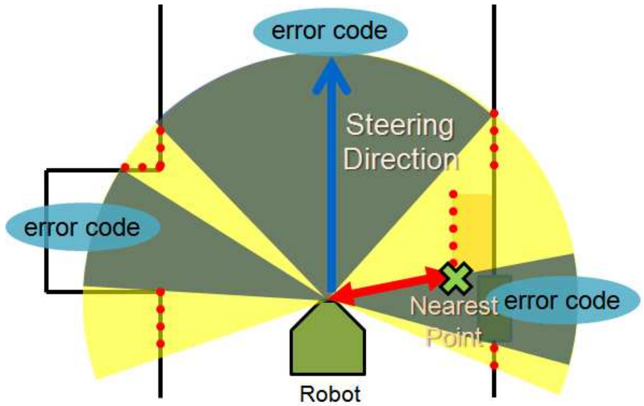

Fig. 11. Mode1: The algorithm to determine steering direction ζ_{ref} and reference velocity v_{ref} when the robot is in straight paths.

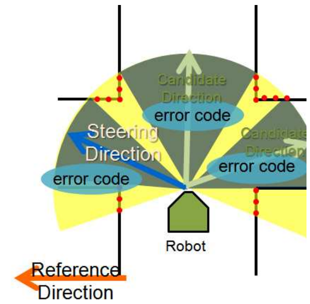

Fig. 12. Mode2: Three candidate directions are produced from three largest consecutive area of error codes, when a intersection is recognized.

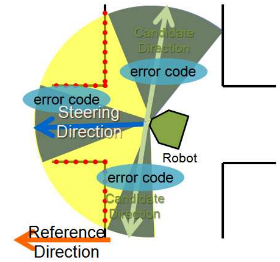

Fig. 13. Mode2: Steering direction is chosen from candidates, the one which is the nearest to the reference direction.

scanned area. Otherwise, an error code is acquired. Fig.11 illustrates an example of the environment, which indicates that the open space in driving direction, hollows on the wall, and glass doors are the area where error code are acquired.

Steering direction ζ_{ref} is set to the vector that is drawn in the middle of the largest consecutive area of error codes. Reference velocity v_{ref} is determined by the correspondence of the distance to the nearest obstacle. v_{ref} is calculated as follows,

$$v_{ref} = v_{max} - k_d * d^{-2} (v_{ref} > 0) \tag{12}$$

where v_{max} is the maximum velocity of 0.6m/s, d is the distance between the robot and the nearest point, and k_d is the gain.

C. Mode2: for intersections

The procedure of how to determine steering direction is illustrated in Fig.12 and Fig.13. Three candidate directions are produced from three largest consecutive area of error codes. The system chooses one of candidates for steering direction ζ_{ref}, so as to turn to the appropriate direction without crashing to the wall or to the obstacle.

Procedure for determination of steering direction is as follows. The robot first recognizes a intersection, and set a ”reference direction” according to the information of Map Data(Fig.12). Once the reference direction is set, it is stored in memory until the mode2 ends. The steering direction ζ_{ref} is chosen from candidate directions that is nearest to the reference direction (Fig.13).

Reference velocity is calculated by (12), with v_{max} set to 0.25m/s.

VI. Experiment

We have implemented the system to the robot, and performed an experiment for navigation. The route is shown in Fig.3. The estimated self-positioning trajectory of the robot during the section $P_1 - P_5$ is shown in Fig.14, the steering direction and orientation of the robot is shown in Fig.15. In Fig.15, the horizontal axis represents time (sec), and vertical axis represents the state. There are changes by 90deg, which means that the robot had turned to the left or right direction when it entered intersections. In Fig.14, lines are plotted with gentle curves, however, the actual route consists of straight lines and crossings, as shown in Fig.3. This is caused by an accumulation of gyro sensor error. In the real environment, robot had driven the given route. The robot uses SOKUIKI Sensor for determination of steering direction, so the accumulation of gyro sensor error would not be a problem for our navigation algorithm.

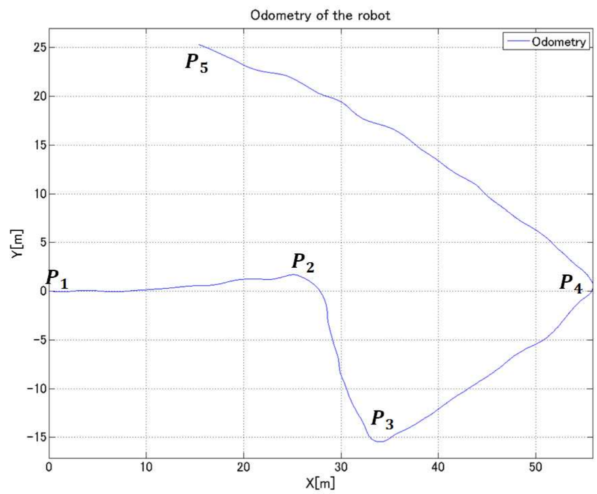

Fig. 14. Estimated self-positioning trajectory of the robot

The actual movement of the robot as it turns left in the crossings (point P_3 in Fig.3) is shown in Fig.16. Acquired data from SOKUIKI Sensor and steering direction is plotted in Fig.17 and Fig.18. In Fig.16, the motion of the robot in every 1 second is shown. The figure shows that the robot had turned crossing to the left, without crashing to the wall. In Fig.17 and Fig.18, plotted dots are scanned points, and the sector with the label (a) represents steering direction. The navigation system was operating in mode1 at the point of Fig.17, where the steering direction is set to go straight. The mode then switched to mode2 at the point of Fig.18, where the steering direction is set to the left.

In the route in Fig.3, P_6 was the only section the robot could not drive autonomously. There are three T-intersections in consecutive, which make it difficult to enter straight in

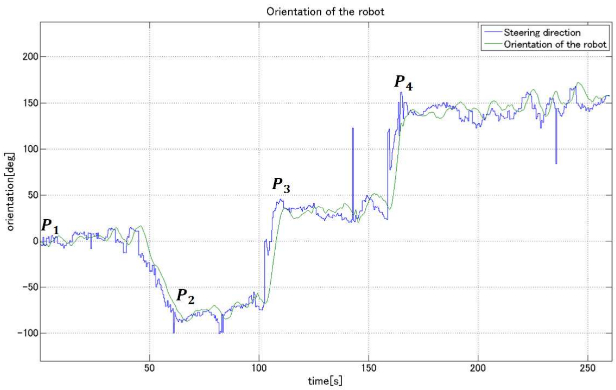

Fig. 15. The steering direction and orientation of the robot

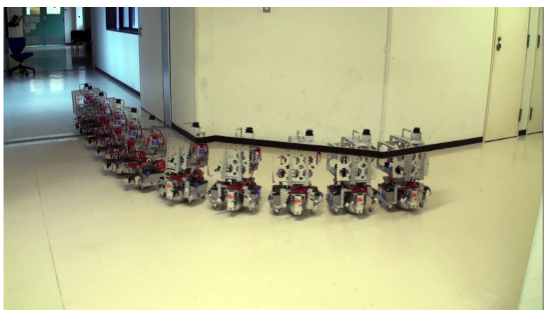

Fig. 16. Movement of the robot in the crossing (Point P_3, drawn every 1 seconds).

each intersection, causing navigation to the wrong direction. The steering direction of the robot indicated the wrong direction for short time, when there is a glass door at side of the path. This is because the SOKUIKI Sensor uses laser for measurement, and the laser would not reflect on glasses.

As a result, the robot had driven all the path in the route, except for P_6. We also did some other experiments by giving the system routes for various goals, such as building-M and building-D. Unless the route goes through P_6, we have succeeded for navigation. The future work for the research is to improve navigation algorithm, in order to navigate in environment where multiple intersections appears in consecutive.

VII. Conclusion

We have developed system to perform a long distance navigation in indoor environment with a unicycle robot. The system had been implemented on SH-4 CPU microcomputer, performing external environment data acquisition from SOKUIKI Sensor, and motion control of unicycle only by a single controller board. We also proposed simple methods for steering control and calculation of steering direction, which solves the issue of sensor rotation problem when in steering motion, and performed long distance navigation of the route shown in Fig.3, except for P_6.

We also performed navigations of various routes, just by changing map data. We think that most of the indoor environment of buildings consists of straight path with some obstacles along side walls, and three major types of intersections, such as L-intersections, T-intersections, and crossings.

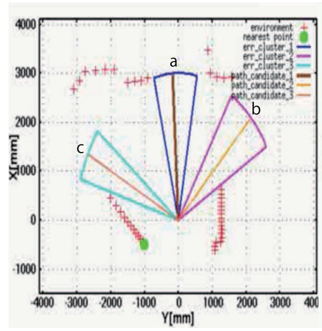

Fig. 17. Acquired data from SOKUIKI Sensor and steering direction (mode1: before entering the crossing).

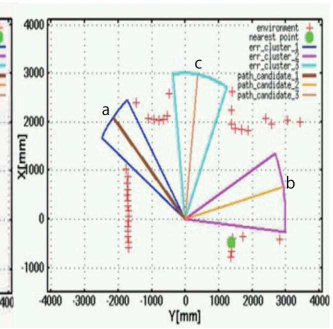

Fig. 18. Acquired data from SOKUIKI Sensor and steering direction (mode1: before entering the crossing).

Therefore, we think that the robot can navigate in various indoor environments.

References

[1] Hirohiko Kawata, Toshihiro Mori, and Shin'ichi Yuta, "Design and Realization of 2-Dimensional Optical Range Sensor for Environment Recognition in Mobile Robots", *Journal of Robotics and Mechatronics*, Vol.17, No.2, pp.116-120, 2005

[2] R.Nakajima, T.Tsubouchi, S.Yuta and E.Koyanagi., "A Development of a New Mechanism of an Autonomous Unicycle", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.906-912, 1997.

[3] General Robotix, Inc. http://www.generalrobotix.com/

[4] HOKUYO SOKUIKI Sensor. http://www.hokuyo-aut.co.jp/02sensor/07scanner/urg_04lx.html/

[5] HONDA U3-X. http://world.honda.com/news/2009/c090924New-Personal-Mobility-Device/

[6] MURATA GIRL. http://www.murata.com/new/news_release/2008/0923.html

[7] J.Okumura, T.Takei, T.Tsubouchi., "Path Tracking of an Unicycle Robot with a Wide-Type Wheel aimed for Navigation", *Advanced Motion Control*, accept.

[8] S.Majima, T.Kasai, T.Kadohara. "A Design of a Control Method for Changing Yaw Direction of an Underacuatted Unicycle Robot", *TENCON IEEE Region 10 Conference*, pp.1-4, 2006.

[9] Enrique D. Ferreira, Shu-Jen Tsai, Christiaan J. J. Paredis, H. Benjamin Brown, "Control of the Gyrover: a single-wheel gyroscopically stabilized robot", *Advanced Robotics*, Volume 14, Number 6, pp.459-475, 2000