ODENS 2013 Team Description

Yoshihiro Fukumoto, Yasuhiro Masutani

Osaka Electro-Communication University
1130-70, Kiyotaki, Shijonawate, Osaka 575-0063, Japan

Abstract. In this paper, the system of team ODENS, Osaka Electro-Communication University, Japan is introduced. Although the hardware of robot is designed and manufactured by a company, the software of controlling 4-wheeled omnidirectional mechanism is original. The system outside of the field consists of a server PC and some client PCs. The server converts information from SSL-Vision to the clients and manages wireless communication to the robots. The clients decide action of corresponding robots one by one based on the information received from the server. In ODENS 2013, an onboard program for controlling robot motion is improved.

1 Introduction

Team ODENS consist of members of Masutani Laboratory in Department of Computer Science, Faculty of Information Science and Arts, Osaka Electro-Communication University, Japan. ODENS have participated in RoboCup competition since RoboCup Japan Open 2007. The results in the Japanese competitions of ODENS were the 3rd place in 2009, the 2nd place in 2010, and the 4th place in 2011 and 2012. Moreover, they were the 4th place in RoboCup 2009 Graz and in the eight best teams in RoboCup 2011 Istanbul.

In the Department of Computer Science, Exercise in Robot Programming is organized for the second grade students, which uses the actual robot system for RoboCup competition. Students learn basic programming of deciding action of soccer robots in the exercise. In the room for the exercise, full-size SSL field and two ceiling cameras are readied. Since ODENS develop robots and programs there, it can always do exercises and experiments on the assumption of regular game.

In this department, students belong to professors' laboratories from the second semester of the second grade. Students who are interested in soccer robot through "Exercise in Robot Programming" wish to enter Masutani Laboratory. In Masutani Laboratory, projects for RoboCup are themes for pre-seminar before regular graduation thesis. Moreover some students study RoboCup as also theme of graduation thesis.

Since the department is in the field of computer science and technology, the second grade students focus on development of software for deciding action. The hardware of robot is designed and manufactured by a company. The software of controlling omnidirectional mechanism is improved by graduate students and higher grade students.

In the following sections, the hardware and software of robot is introduced first. After that the computer system outside of field, especially a method of deciding action is described.

2 Overview of the system

The system of ODENS is a distributed and autonomous system, which consists of one server and some clients as shown in Fig.1. One client program corresponds to one robot. All client is independent of each other. The server receives coordinate information of objects in the field provided by SSL-Vision. They are transformed in appropriate format and sent to the clients. Signals from the referee box are also sent to them. Each client program decides the next action for the corresponding robot based on the information received from the server and sends a command back to the server. The server collects the commands from all clients and sends them to the robots in the field by wireless. The above cycle is executed 60 times in one second.

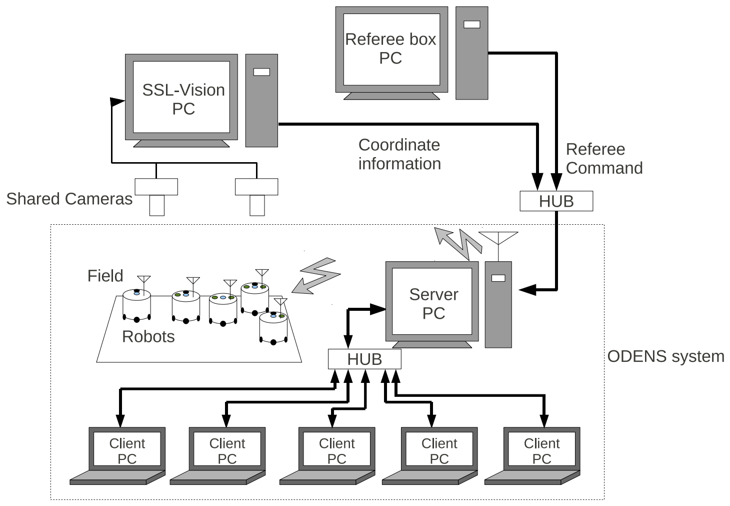

Fig. 1. System of ODENS

3 Robots

The mechanical and electrical hardware of robots is designed and manufactured by SMATS Corp., which is very similar to RoboDragons 2009's robot[1]. However, the onboard software for control is original.

3.1 Hardware

An appearance of the robot is shown in Fig.2. The main specification is shown in Table 1.

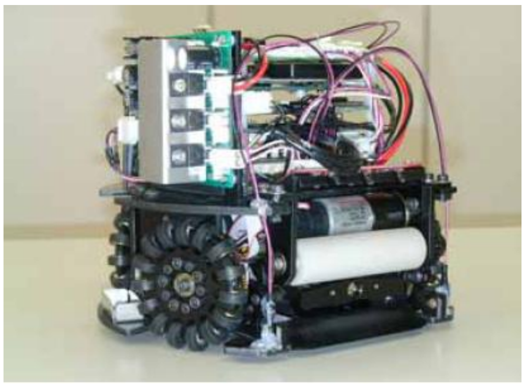

Fig. 2. ODENS robot

Table 1. Specification of the robot

Dimension	length 179[mm], width 179[mm], height 140[mm]
Mass	2.2[kg]
Wheel	number 4, radius 30.5[mm], sub wheels 15
Motor	maxon RE-max23 222050, 11[W], reduction ratio 7.916
Kicking device	3 solenoids, 240[V]
Dribbling device	radius 10[mm], length 72[mm]
Ball sensor	LED and Phototransistor, number 4
Gyro sensor	ADXRS610
Wireless modem	Futaba FRH-SD07T, 2.4[GHz]]
CPU	Renesas SH7045F(SH-2)
Battery	Li-Polymer 14.8[V]$\times$1, 7.4[V]$\times$1

3.2 Software for Robot Control

The program of CPU on the robot is developed with GNU C compiler. Three tasks are concurrently processed on the CPU. Task for feedback control is executed every 2[ms]. Task for communication receives a command from the outside via wireless modem every 16.7[ms]. The command consists of magnitude and direction of linear velocity, angular velocity, and on/off of dribbling device and kicking device.

ODENS used a control law based on dynamics model by 2011. It assumes that the four wheels are evenly grounded. Therefore, good performance can not be obtained in case that the grounding condition is not good due to worn tire or poor maintenance. To overcome this problem, another control law was introduced in 2012, which is based on kinematics and resolves velocity command of the body into the four wheels' as shown in Fig.3.

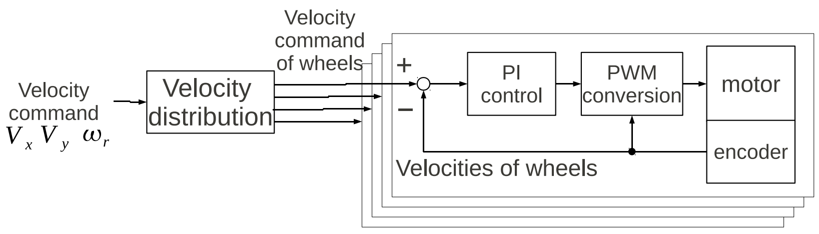

Fig. 3. Diagram of control law

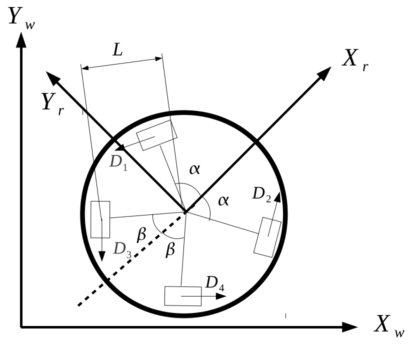

Fig. 4. Kinematic model of omnidirectional robot

Velocity Resolving As shown in Fig.4, a coordinate system is attached on the robot. Then numbers are assigned for wheels. Let α be angle of axes of front wheels from the front direction $(+X_r)$, β be angle of axes of rear wheels from the rear direction $(-X_r)$, and L be distance between the center and the wheel.

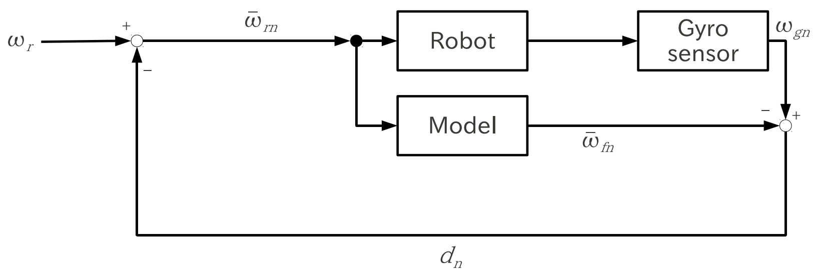

Fig. 5. Compensation using a gyro sensor

Letting V_x and V_y[mm/s] be translational velocity commands and ω_r[deg/s] be angular velocity command in the robot coordinate system that is received form the outside, desired velocity of each wheel $v_i(i = 1, 2, 3, 4)$[mm/s] can be represented as their linear combination Eq.1-4 based on kinematic relation..

$$v_1 = -V_x \sin\beta - V_y \cos\beta + L\frac{\pi}{180}\omega_r \tag{1}$$

$$v_2 = V_x \sin\alpha + V_y \cos\alpha + L\frac{\pi}{180}\omega_r \tag{2}$$

$$v_3 = -V_x \sin\alpha + V_y \cos\alpha + L\frac{\pi}{180}\omega_r \tag{3}$$

$$v_4 = V_x \sin\beta - V_y \cos\beta + L\frac{\pi}{180}\omega_r \tag{4}$$

Control Law Letting N be reduction ratio and r be radius of the wheel, desired speed of each motor $\omega_{ti}(i = 1, 2, 3, 4)$[rpm] is defined in Eq.5.

$$\omega_{ti} = \frac{60Nv_i}{2\pi r} \tag{5}$$

Letting B be viscous coefficient, K_t be torque constant of the motor, β be static friction, $\omega_i(i = 1, 2, 3, 4)$ be the actual rotational speed, and sign(*) be function that returns -1, 0, or $+1$ corresponding to the sign of the argument, estimated value of the torque by viscous friction and Coulomb friction $g'(\omega_i)$ can obtained in Eq.6.

$$g'(\omega_i) = B\omega_i + \text{sign}(\omega_i)\beta \tag{6}$$

Letting K_p be proportional gain, K_i be integral gain, and, Δt be control period, current to drive the i-th motor i_i is given in Eq.7.

$$i_i = K_p(\omega_{ti} - \omega_i) + \frac{g'(\omega_i)}{K_t} + K_i \Delta t \sum(\omega_{ti} - \omega_i) \tag{7}$$

For PWM driven motor, current i can be represented as a function of the duty ratio p and the angular velocity ω, $i = g(p, \omega)$, based on model of electrical circuit. The program has numerical table of inverse of this function. For given current and angular velocity, duty ratio is decided by looking up the table.

$$p_j = g^{-1}(i_i, \omega_i) \tag{8}$$

Compensation Since the omni-wheels are slippery, the robot often rotates during translation and turns gradually. To overcome this problem, the angular velocity command ω_r given from the outside is compensated by subtracting a disturbance as shown in Fig.7. The compensated command $\bar{\omega}_{rn}$ is represented in Eq.9.

$$\bar{\omega}_{rn} = \omega_{rn} - d_n \tag{9}$$

where n is a sampling number. The disturbance d_n is estimated as difference between output from the gyro sensor ω_{gn} and output from 1st order delay model of the robot ω_{fn} in Eq.10.

$$d_n = \omega_{gn} - \omega_{fn} \tag{10}$$

The 1st order delay model can be represented in Eq.11.

$$\omega_{fn} = \alpha \bar{\omega}_{rn} + (1 - \alpha)\omega_{fn-1} \tag{11}$$

where α is a constant.

Experiments In order to evaluate the effectiveness of the above-mentioned compensation, some experimental results are shown. The condition is as follows;

- Translational velocity is 500 and 1000[mm/s]
- Translational direction is 0 and -90[deg]
- Angular velocity is 0[deg/s]
- Two conditions:
  - None: no compensation
  - Compensated: by using Eq.9

Table.2 shows comparison of the average errors in the angular velocity under the respective conditions. Fig.6 and Fig.7 show temporal change in the direction of the robot without the compensation and with compensation respectively in case that the command is in the direction of -90[deg] at 1000[mm/s]. It is found that the compensation is effective in reducing the rotation during translation.

Table 2. Comparison of averages of angular velocity errors

Command velocity[mm/s]	500		1000	
Translational direction[deg]	0	−90	0	−90
None[deg/s]	−6.89	2.23	−12.94	23.45
Compensation[deg/s]	1.46	1.36	1.80	3.33

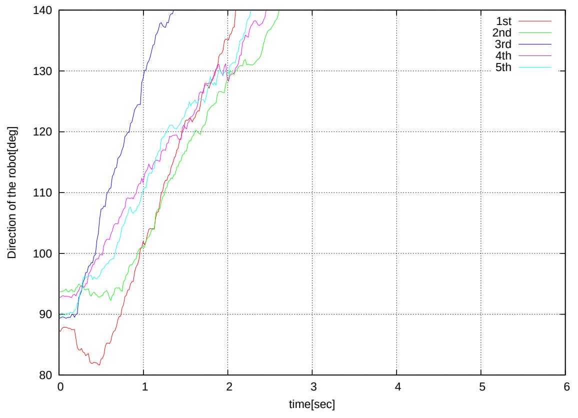

Fig. 6. Temporal change in the direction of the robot when the translation to −90[deg] at 1000[mm/s] without compensation

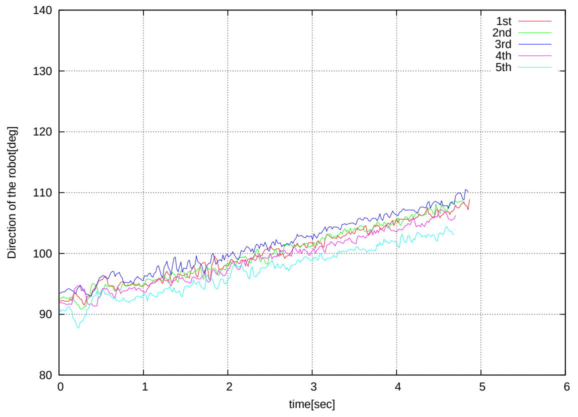

Fig. 7. Temporal change in the direction of the robot when the translation to −90[deg] at 1000[mm/s] with compensation

4 Server

4.1 Hardware

PC The server PC is equipped with Intel Core2 Duo P8700 2.53[GHz] and 2[GB] RAM. It is connected to the SSL-Vision PCs, referee box PCs, and some client PCs through LAN. It also uses a USB-serial converter to connect the wireless modem.

Wireless communication Wireless modem Futaba Corp. FRH-SD3T is used to communicate with the robots in the field, whose communication rate is 38.4[kbps] and whose frequency is 2.4[GHz].

4.2 Software

OS for server PC is Microsoft Windows XP Professional SP3. Visual Studio 2008 C++ is used to develop the program. The server program has GUI and consists of multiple threads. These threads have functions respectively as follows.

S-V Watcher SSL-Vision Watcher is a thread to merge and manage the field information divided into two that is given from SSL-Vision. It gives the merged field information to the "Position estimation" thread mentioned later.

Threads

Kalman Filter thread The positions and velocities of objects are estimated by Kalman filter, whose observed values are the position of the objects given from S-V Watcher and whose model is a constant velocity motion. Furthermore, this function tackles exceptional situations by evaluating the difference between the observed value and the estimated value. It gives the estimated information such as robot number and coordinate to the network thread.

Network thread This thread sends the information received from the Kalman filter thread and the signals from the referee box to the clients. Moreover this thread manages the connection request from the clients.

Referee thread This thread transforms the signals received from the referee box to the network thread.

Receiver thread This thread communicates with the clients. However, this thread does not transmit the message to the clients. It receives the command to the robot from the clients. Since this thread corresponds to a client, the number of the threads increases and decreases depending on the number of the connected clients. It gives data which received form the clients to the radio thread.

Radio thread This thread collects commands received from all clients and send them to the robots in the field.

GUI thread This thread accepts the input from the user and displays various status on the screen. The processing priority of this thread is lower than other threads not to obstruct the more important threads.

5 Client System

5.1 Hardware

PC executing the client program does not need special specification and performance, whose OS is not specified(Both Windows and Linux are used). Only function of network communication is needed to connect to the server.

5.2 Software

Structure of action decision One client program corresponds to the one robot. Therefore, all clients program are executed independently of each other. The structure of client program is separated into four layers in Game layer, Role layer,Task layer, and Communication layer.

Game layer This layer is processed based on a state transition, whose state is switched mainly by command received from the referee box. There are five states, “Out of play”, “Pre set play”, “Set play”, “In play”, and “Halt”.

Role layer This layer decides role of each robot at each state. The role is assigned based on a role table mentioned below when the game state is “In play”. Otherwise, The role is assigned based on only distances between robots and ball in the team.

The role table is used to decide the role from the following four conditions.

- Rank: Rank of distance from the ball in the team (1...6)
- Relation: which team is near the ball? (not both, own team, opponent. team, or both)
- Distance: distance from the ball (possessed, near, middle, or, far)
- Position: in which area is the ball as shown in Fig.8? (0...5)

An example of the table shown in Table 3. Symbols in the table means roles as follows.

- Keeper: to defend goal in penalty area
- GoalDefender: to defend goal outside of penalty area
- PassCutter: to cut passing as for opponent team’s ball
- BallGetter: to get opponent team’s ball
- Attacker1: to shot at the opponent team’s goal
- Attacker2: to assist Attacker1 or Attacker3
- Attacker3: to attack by passing
- PassWaiter: to wait for ball at specific position

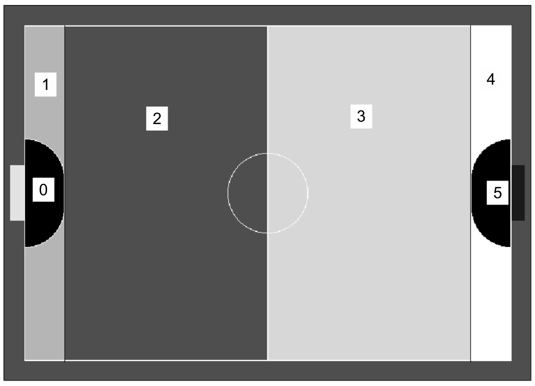

Fig. 8. Six areas in the field

Table 3. Role table

	Relation	both teams are far						my team is near						opponent team is near						both teams are near					
Rank	Position / Distance	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	possessed	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1	AT1
	near	BG	AT1	AT1	AT1	AT1	AT1	BG	AT1	AT1	AT1	AT1	AT1	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG
	middle	BG	BG	BG	BG	BG	AT1	BG	BG	BG	BG	BG	AT1	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG
	far	BG	BG	BG	BG	BG	AT1	BG	BG	BG	BG	BG	AT1	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG	BG
2	possessed	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	GD	GD	AT2	AT2	AT2	AT2
	near	PC	PC	PC	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	GD	PC	AT2	AT2	AT2	AT2	GD	GD	AT2	AT2	AT2	AT2
	middle	GD	GD	PC	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	GD	GD	GD	AT2	AT2	AT2	GD	GD	GD	AT2	AT2	AT2
	far	GD	GD	GD	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	AT2	GD	GD	GD	AT2	AT2	AT2	GD	GD	GD	AT2	AT2	AT2
3	possessed	GD	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PC	PC	GD	GD	GD	GD	PW2	PC
	near	GD	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PC	PC	GD	GD	GD	GD	PW2	PC
	middle	GD	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PC	PC	GD	GD	GD	GD	PW2	PC
	far	GD	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PC	PC	GD	GD	GD	GD	PW2	PC
4	possessed	PW2	PW1	GD	GD	GD	GD	PW2	PW1	PW2	GD	GD	GD	PW2	GD	GD	GD	GD	GD	PW2	GD	GD	GD	GD	GD
	near	PW2	PW1	PW2	GD	GD	GD	PW2	PW1	PW2	GD	GD	GD	PW2	GD	GD	GD	GD	GD	PW2	GD	GD	GD	GD	GD
	middle	PW2	PW1	PW2	GD	GD	GD	PW2	PW1	PW2	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PW2	PW1	PW2	GD	GD	GD
	far	PW2	PW1	PW2	GD	GD	GD	PW2	PW1	PW2	GD	GD	GD	PW2	PC	GD	GD	GD	GD	PW2	PW1	PW2	GD	GD	GD
5	possessed	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD	PW2	GD	GD	GD	GD	GD
	near	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD	PW2	GD	GD	GD	GD	GD
	middle	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD
	far	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD	PW2	PW2	PW2	GD	GD	GD

AT1:	Attacker1	PC:	PassCutter	PW1:	PassWaiter1
AT2:	Attacker2	GD:	GoalDefender	PW2:	PassWaiter2
BG:	BallGetter				

Task layer This layer execute concrete actions of the robot. For example, "Turn to the ball", "Move to specified coordinates while avoiding obstacles " and so on. It does not have state transition model internally. Therefore, when a destination to move and information of the field are given, the output action is uniquely decided.

Communication layer This layer has functions for communication.

6 Conclusion

As mentioned above, the part of deciding action is perfectly separated from image processing and robot control in the system. Owing to this feature, this system became useful platforms for both education and research.

Although under the SSL regulation it is possible to make centralized system, in the system of ODENS, the program for each robot is independent of others. In the future, following this policy we will study robot system and participate in competitions.

References

[1] J. Maeno et al., “RoboDragons 2009 Team Description”, RoboCup2009 (2009).
[2] K. Kanaya et al., “ODENS 2009 Team Description”, RoboCup2009 (2009).
[3] M. Nakajima et al., “ODENS 2010 Team Description”, RoboCup2010 (2010).
[4] Y. Fukumoto et al., “ODENS 2011 Team Description”, RoboCup2011 (2011).