RoboDragons 2010 Team Description

Akeru Ishikawa, Takashi Sakai, Jousuke Nagai, Toro Inagaki, Hajime Sawaguchi, Yuji Nunome, Kazuhito Murakami and Tadashi Naruse

Aichi Prefectural University, Nagakute-cho, Aichi, 480-1198 JAPAN

Abstract. This paper describes the system configuration of Aichi Prefectural University's RoboDragons 2010. Main topic in this year is that we developed a new robot hardware. We describe it in detail here. There are a little improvements in soccer software. We describe our soccer software in brief.

1 Introduction

This year's team description paper (TDP) of RoboDragons2010 describes a new robot hardware, an improved software of RoboDragons 2010 system.

First, robot hardware. Our current robots are the fifth generation ones in our Labs. Our team has six robots, five for competition and one for a reserve. Each robot of RoboDragons consists of four omni-directional wheels, a dribbling device, two kicking devices (for chip kick and straight kick) and the embedded computer for controlling the robot. They are controlled by a host computer next to the field. Second, software. The software on the host computer origenates from the one when we made a joint team with CMU in 2004 and 2005. Many improvements have done since 2005, however, this year, noticeable improvement has not done. We overview our software centering on changes from 2009 to 2010.

In the following sections, we describe these topics.

2 Robots

In this section, we discuss our new robots in detail. First, we show the overview of our new robot in Figure 1.

2.1 Dimensions of robot

The robot can be packed in the cylinder with dimensions of 145 *mm* height and 178 *mm* diameter. To protect the internal circuit boards and mechanical devices, the robot is covered by the cardboard (see Fig. 1 left). To strengthen the cardboard, the plastic sheet with 1.5 *mm* thickness is glued.

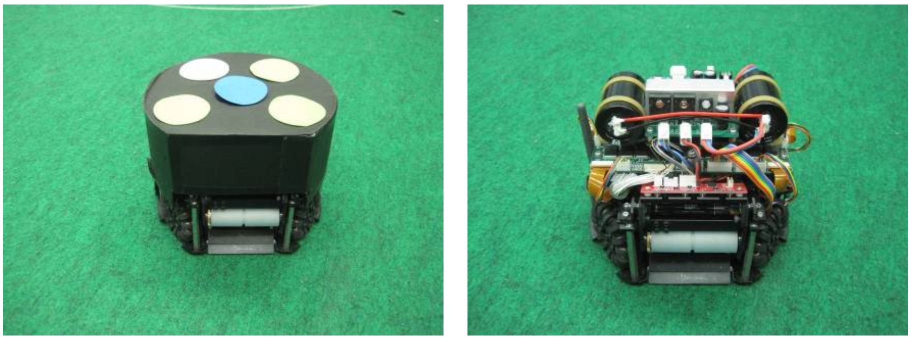

Fig. 1. RoboDragons new robot
(Left: with cover, Right: without cover)

2.2 Drive unit

The robot has 4 wheels each of which is drived by a DC blushless motor. The wheel is so called an omni-wheel. Figure 2 shows the omni-wheels moving the robot.

The DC blushless motor driving the omni-wheel is Maxon's "EC 45 flat 30 W" with encoder unit. The source voltage of the motor is 15 V. The motor also has a pinion gear with 21 teeth and the omni-wheel has a gear with 64 teeth so that the reduction ratio is 1 : 3.047. The diameter of the omni-wheel is 56 mm and the omni-wheel has 15 small tires in circumference. The diameter of the small tire is 13 mm.

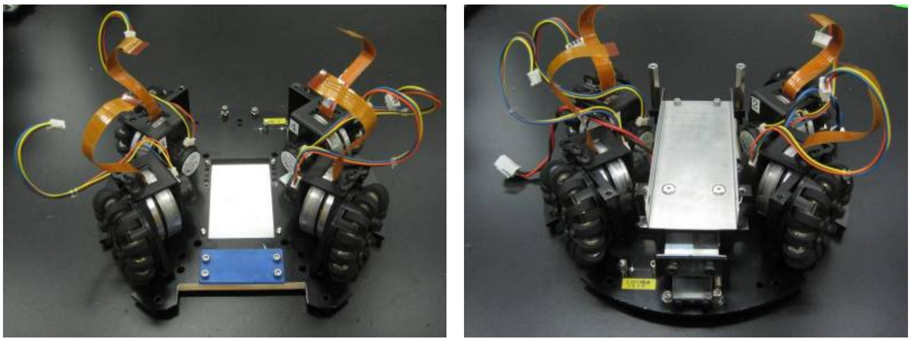

Fig. 2. Omniwheels and DC blushless motors
(Left: without solenoid, Right: with solenoid)

2.3 Kicking device

The kicking device consists of solenoids, kick bars, and a voltage booster. Figure 3 (left) shows the kicking device and figure 3 (right) shows the voltage booster.

Solenoids Three solenoids are built in the robot, one large solenoid is for a main kick device and a small solenoid is for a chip kick device.

The coil of the large solenoid is made winding the 0.6 mm^{ϕ} enamelled wire on a bakelite cylinder in 7 layers. The dimensions of the cylinder are 13 mm in inner diameter, 26 mm in outer diameter and 55 mm in length. The stroke of the solenoid is 30 mm and enables the kicking a ball with speed of 9.5 $msec$ under the ideal conditions.

The coil of the small solenoid is made using the same material with the large one. The dimension of the cylinder is 13 mm, 26 mm and 27 mm, respectively. The stroke is 7 mm and it can kick the ball with flying distance of 2 m and flying height of 1 m.

Each solenoid employs the spring to pull back the plunger.

Kick bar 7075 aluminum alloy which is light and hard is employed for the kick bar. Moreover, V-shaped ABS plastic is attached to the aluminum kick bar as shown in figure 3. This helps to kick the ball to the direction perpendicular to the kick bar within the accuracy of 5 degree.

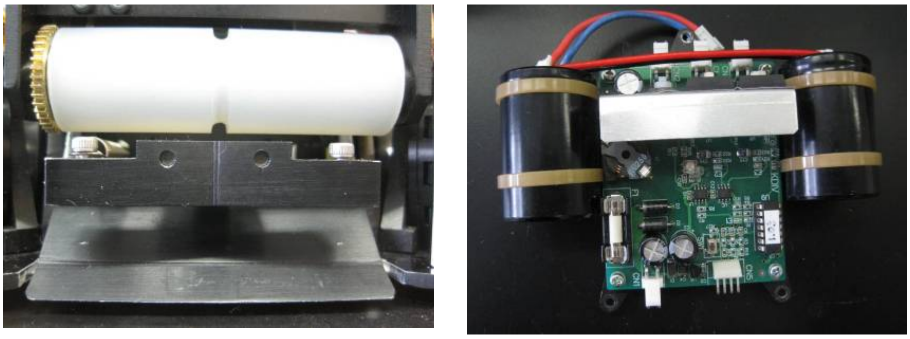

Fig. 3. Kicker and voltage booster
(Left: kicker, Right: voltage booster)

Voltage booster: The voltage booster(Fig.3(right)) is a kind of DC-DC converter. It converts 15 V DC input voltage up to 200 V DC output voltage. Output voltage is adjustable between 150 V and 200 V. The chopper circuit using a choke coil is a heart of the voltage booster. A PIC controls the chopper circuit. Large capacity condensers are used for keeping the high voltage output. Total capacity is 4500 μF. The voltage sensing circuit controls the output voltage. 2 solid state relays are used as the switches to drive the solenoids. These relays are controlled exclusively by the PIC. The charging time of the condensers is about 2 *sec* when the output voltage is 200 V.

2.4 Dribbling device

A dribbling device of the robot has been achieved by combining the dribble roller, the motor and the gear.

The dribbling device uses a Maxon's "EC 16 15W" motor with an encoder and a planetary gear. The reduction ratio of the gear is 1 : 5.4. A pinion gear attached to the motor has 40 teeth and a gear attached to the dribble roller has 36 teeth. Therefore, the net reduction ratio R is given by,

$$R = R_m \times R_g = 5.4 \times \frac{36}{40} = 4.86, \tag{1}$$

where, R_m is the reduction ratio of the planetary gear and R_g is the reduction ratio of the pinion gear and the gear on the dribble roller.

The dimensions of the dribble roller are 20mm in diameter and 73mm in length. The material of the dribble roller is a alminum shaft with silicon rubber of 4mm thickness on the face of the shaft.

2.5 Communication unit

Our wireless communication is a spectrum diffusion communication on the 2.4 GHz band. Futaba's wireless modem "FRH-SD07T" is used for the purpose. The modem has several communication modes. We use a "direct mode" which can send the messages with the least delay between modems. A pair of the FRH-SD03T and the FRH-SD07T realizes the communication between the host computer and the robot(s).

2.6 Proximity sensor unit

The proximity sensor is attached above the dribbling device and it detects the ball just in front of the dribbling device. The heart of the sensor is three infra-red light emission diode (LED) and photo diode pairs. The irradiation angle of the LED is about 15 degree. When one of the three photo diodes gets the reflected infra-red ray more than a preset threshold value, the sensor outputs the signal.

2.7 Control unit

A control unit of robot is a board computer, which is newly developed. It is shown in figure 4 These boards include a The CPU is Hitachi's SH2A processor with

FPGA for peripheral control. SH2A has abundant peripheral circuits in it and makes the compact implementation of the control unit possible. The memories compose of Flash ROM(1MB) and SRAM(1MB). The IO boards have power transistors that can drive the motors and they also have interface circuits with the motors driving wheels and dribble roller, and the proximity sensor.

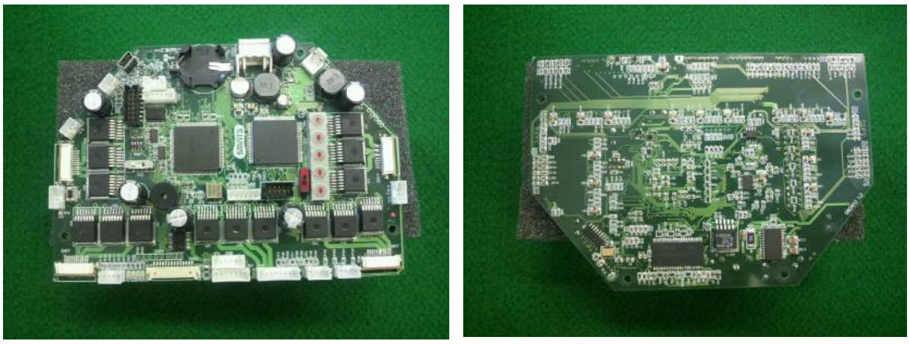

Fig. 4. Processor boards
(Left: face, Right: back)

2.8 Control program on robots

At the time this paper is written, the software shown below is not implemented on the robot yet, however, it will be implemented until RoboCup 2010 competition.

The program is written by the C programming language. The TOPPERS real time OS[1] is used.

Robot control program consists of three modules each of which is invoked as a process. They are communication, command and motor control modules. Figure 5 shows a data/signal flow among modules and peripheral units.

Robot command is sent from the host computer to robots by a packet every 1/60 seconds using asynchronous serial communication. Since the communication speed is 19200 bps in our system, it is possible to send 32 bytes data in 1/60 second if only start bit and one stop bit are used as control bits. Therefore, Our packet is consisted of 32 bytes[1] as shown in figure 6. The packet is broadcasted to all robots. The packet has error correcting code (ECC) to make reliable communication possible. We use the Humming code as the ECC.

The communication module receives the packet and extracts the command of its own. If error is detected, the packet is discarded. As far as the error is not

[1] The command cycle synchronizes with the camera cycle which is equal to 59.94 frames per second. Therefore, the command can send without any problem.

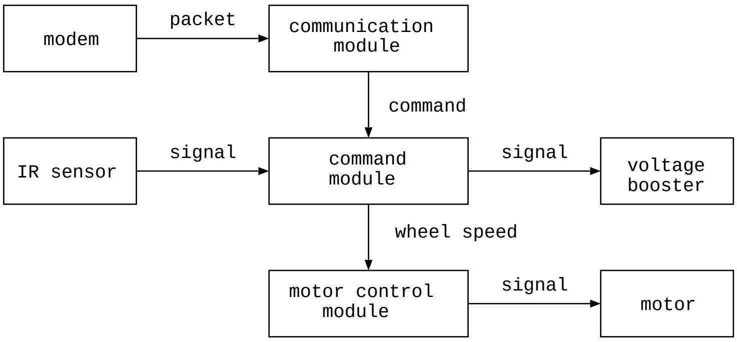

Fig. 5. Data/signal flow

burst error, the discard of packet is a good alternative[2]. Extracted command is sent to the command module.

In the command module, the velocity of each wheel is calculated from the vel, dir and rot value (see Fig. 6) and calculated result is sent to the motor control module. The kick and dribble command is executed, as well.

In the motor control module, the PID control is performed. This is done by using the target velocity given by the command and the current velocity calculated from the encoder pulses. The motor drive control is done by the PWM control.

3 Software system

3.1 Overview

Figure 7 shows the overview of the RoboDragons system. The features of the system are as follows:

1. Host computer is Athlon64 X2 4200+ with 512MB memory and Debian GNU/Linux OS.
2. Each module(shown in box in Fig.7) is implemented as a thread.
3. The *Soccer* module consists of a strategy, a tactics and a path generation submodules, and it produces an action command for each robot.
4. *Radio* module sends the command to each robot through the radio system.

In the next sections, we describe the main improvements of our program.

[2] There are no such errors experienced in recent competitions.

```
class SerialCommand{
  public:
    uint8_t vel;    // Velocity [unit: cm/s]

    uint8_t dir;    // Direction [unit: 2*pi/256 rad]

    uint8_t rot;    // Rotation velocity [unit: 2*pi/60 rad/s]

    uint8_t var;    // kick and dribble command
                // b0-b1: kick condition
                //        0 : No kick
                //        1 : kick when center senser reacts
                //        2 : kick when any sensers react
                //        3 : kick immediately
                // b2-b3・b7: Selection of kicker
                //        0 : No kick
                //        1 - 4 : Main kicker (1: weak ... 4: strong)
                //        5 - 7 : Chip kicker (5: weak ... 7: strong)
                //              : Main and chip kickers are used exclusively
                // b4-b5: Dribble
                //        0 : No dribble
                //        1 : Reverse rotation
                //        2 : Weak normal rotation
                //        3 : Strong normal rotation

    uint8_t ecc01;    // b0-b3: ECC of vel
                      // b4-b7: ECC of dir

    uint8_t ecc23;    // b0-b3: ECC of rot
                      // b4-b7: ECC of var
};

struct SerialPacket{
  uint8_t header0; // 0x80
  uint8_t header1; // 0x0D
  SerialCommand robot[5];
};
```

Fig. 6. Packet configuration

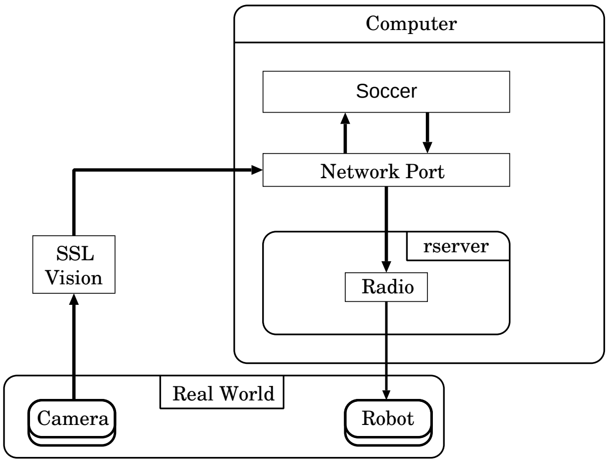

Fig. 7. RoboDragons system: overview

3.2 Determining the number of mark robots

In our old strategy, the role of each teammate robot at the opponent free kick time was determined by the ball's position. For example, if the ball is within 500 mm from teammate goal line at the beginning of the free kick, it is considered to be an opponent corner kick and "3 mark robots" strategy is selected even if an opponent robot is out of field by a penalty. The program is improved to determine the number of mark robots dynamically as follows,

Let $\mathbf{r}_i$ be the position of the opponent robot i. If following conditions are satisfied, the robot i will be marked.

- The distance between $\mathbf{r}_i$ and the center of teammate goal is less than R mm.
- The distance between $\mathbf{r}_i$ and the teammate goal line is less than X mm, and there is a shoot course from $\mathbf{r}_i$ to the goal.

In RoboDragons, $R = 3000$ mm and $X = 4600$ mm. The number of mark robots ranges from 1 to 3.

3.3 Priority of mark robots

In RoboDragons, the number of teammate robots for marking is at most 3. However, there are cases that more than 3 robot should be marked at the opponent free kick. For example, the case four opponent robots are attacking. In this case, we select three opponent robots to mark. For each opponent robot, a priority is given as follows, and higher priority robots are selected.

- the greater the distance between $\mathbf{r}_i$ and the center of teammate goal, the higher the priority.

- the greater the affordance of the shoot course from $\mathbf{r}_i$, the higher the priority.

There are two mark actions, i.e. pass cut mark and shot cut mark. The pass cut mark has two variations, i.e. preventing the pass at around the passing robot and around the receiving robot. The shot cut mark is as well.

3.4 Robots' actions at free kick

In the old strategy, teammate's pass action and shot action at free kick were as follows.

- Pass action (passing robot in case of direct play[2])
  1. At time t_0, the robot which has the greatest affordance of the shoot course is selected as a receiving robot of the pass.
  2. Passing robot moves to the waiting position $\mathbf{T}_r$.
  3. When passing robot arrives at $\mathbf{T}_r$ at t_1, it starts kicking action.
  4. After finishing the pass, passing robot waits there until the strategy change will happen.
- Shot action (shooting robot in case of direct play)
  1. Search the best place to receive the ball and move there until the passing robot will pass.
  2. After passing, the shooting robot moves to the shooting position which is computed from the velosity of the ball and current self position.
  3. Shoot the ball. If shot is failed, then next action is taken.

New actions are,

- Pass action (passing robot in case of direct play[2])
  1. At time t_0, the robot which has the greatest affordance of the shoot course is selected as a receiving robot of the pass.
  2. Passing robot moves to the waiting position $\mathbf{T}_r$.
  3. When passing robot arrives at $\mathbf{T}_r$ at t_1, it starts kicking action.
  4. After finishing the pass, passing robot takes an attacker action.
- Shot action (shooting robot in case of direct play)
  1. Search the best place within the specified region to receive the ball and move there until the passing robot will pass.
  2. After passing, the shooting robot moves to the shooting position which is computed from the velosity of the ball and current self position.
  3. If there are some candidates for shooting, the robot nearest to the teammate goal judges the possibility of shot. If it is not possible, then take a defending action.

In the RoboDragons 1-2-3 shot[2] using three robots, above shot action works well to reduce the time the goal keeper defends the goal without any other defender(s).

References

1. “http://www.toppers.jp/en/index.html”
2. Ryota Nakanishi, James Bruce, Kazuhito Murakami,Tadashi Naruse and Manuela Veloso, “Cooperative 3-Robot Passing and Shooting in the RoboCupSmall Size League”, RoboCup 2006:Robot Soccer World Cup X, LNCS 4434 pp.418-425