RoboDragons 2019 Extended Team Description

Masahide Ito, Reona Suzuki, Shogo Isokawa,
Jiale Du, Ryoto Suzuki, Mayu Nakayama, Yuta Ando,
Yuki Umeda, Yuki Ono, Fusuke Kashiwamori, Fumiya Kishi,
Kazutoshi Ban, Takateru Yamada, Yusuke Adachi, and Tadashi Naruse

School of Information Science and Technology, Aichi Prefectural University
1522-3 Ibaragabasama, Nagakute, Aichi 480-1198, JAPAN
Email: `masa-ito@ist.aichi-pu.ac.jp` and `ssl.robodragons@gmail.com`
URL: `https://www.facebook.com/RoboDragons/`

Abstract. *RoboDragons* is a team of Aichi Prefectural University in the RoboCup Soccer Small Size League. This paper presents a technical overview of our robots and their main changes from 2018 to 2019. We will use the seventh-generation robots in RoboCup 2019 as in last two years. As a change in the hardware part, we replaced small wheels of omni-wheels so as to obtain more smooth mobility and less maintenance; in the software part, we improved a skill used in replacement.

1 Introduction

RoboDragons is a team of Aichi Prefectural University (APU), participating in the Small Size League (SSL) of RoboCup Soccer. This team originated from *Owaribito*—a joint team between APU and Chubu University—which was founded in 1997. In 2002, since two universities have been ready to manage each individual team, APU built a new team, RoboDraongs. After that, RoboDragons has been participating in the SSL, including activities as *CMRoboDragons*—a joint team with Carnegie Mellon University in 2004 and 2005. Our best record was the second place in 2009. We also finished twice in the third place (2007 and 2014) and four times in the fourth place (2004, 2005, 2013, and 2016). In RoboCup 2018, we placed seventh–eighth out of nine teams in Division A.

This paper summarizes the technical information of RoboDragons 2019, which includes the main changes from 2018 to 2019. We will use the seventh-generation

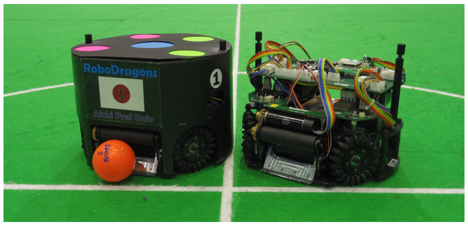
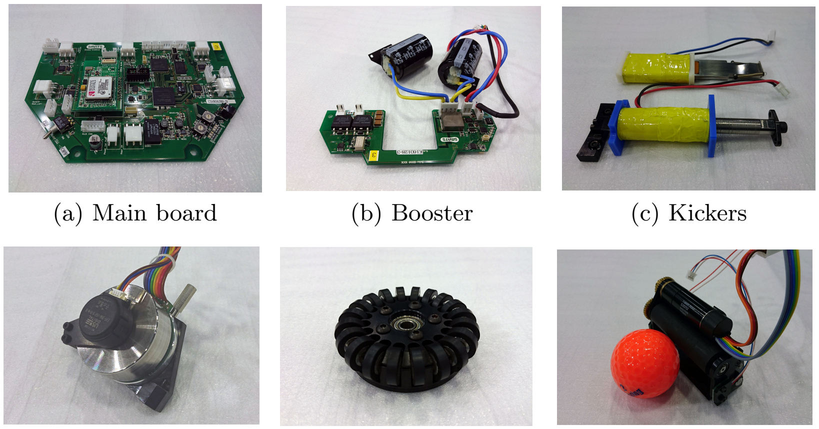


Fig. 1. The seventh-generation RoboDragons robots


(a) Main board (b) Booster (c) Kickers

(d) Wheel motor (e) Omni-wheel (f) Dribbler

Fig. 2. Main hardware components

(7G) robots (Fig. 1)—developed in 2016—in RoboCup 2019 as in last two years. There are a few updates of this year for solving a technical issue and for improving the performance. In the hardware part, we changed the small wheels of the omni-wheels so as to obtain more smooth mobility and less maintenance; in the software part, we improved a skill for ball placement.

2 Overview of RoboDragons System

2.1 Hardware Part

The 7G robots were developed in 2016. The most design is inherited from the 6G robot, but some components—such as the kickers, the dribbler, and the the alignment of four omni-wheels—have been changed. Figure 2 and Table 1 summarize the hardware configuration of the 7G robot. See the details in our previous ETDP [1].

The 7G robots have been used since RoboCup JapanOpen 2017 (Spring 2017). Some technical issues on the hardware emerged through RoboCup 2017. Since then, we prioritized those issues and have tried to solve them one by one. In the last ETDP [2], we reported widening the ball-touchable area of the dribbling roller. This time we focused on friction affected in small wheels of the omni-wheels. See the details in Section 3.

2.2 Software Part

Figure 3 overviews our software system. The three main modules are:

Rserver This module receives the data from *SSL-Vision*, and then the Kalman filter in the *Tracker* submodule estimates the states of a ball and

Table 1. Description of main hardware components.

Device	Description
Main board (Fig. 2 (a))	CPU: SH2A (Renesas Electron. Corp.) operating at 197 MHz. FPGA: Spartan-6 (Xilinx) including peripheral circuits.
Booster (Fig. 2 (b))	Capacitance of capacitor: 4400 μF. Conversion from 15.2 V DC to 150–200 V DC. Electric charge takes about 3 s for 200 V output.
Kickers (Fig. 2 (c))	Material: 7075 alum. alloy. Solenoid: a coil wound by ϕ 0.6 mm enameled wire. Straight kicker can kick a ball at over 8 m/s; chip kicker can kick a ball as far as max. 3 m distance.
Omni-wheel (Figs. 2 (d) & (e))	Four omni-wheels driven by Maxon “EC 45 flat 50 W”. Gear reduction ratio between motor and omni-wheel is 21:64. The core wheel has 20 small wheels in the circumference. Diameter: omni-wheel 55 mm, small wheel 12.4 mm.
Dribbler (Fig. 2 (f))	One roller driven by Maxon “EC 16 30 W”. Roller: alum. shaft with non-repulsive rubber, 16 mm in diameter, and 61 mm in length.
Radio system	IEEE 802.11abgn 2.4/5 GHz wireless LAN.
Ball detector	Infra-red light emission diode and photo diode pair.
Accelerometer	BOSCH BMA250 (3-axis; range: ±2 G to ±16 G)
Gyroscope	InvenSence ITG3400 (pitch, roll & yaw; range: ±250 deg/s)

robots. The estimated states are shared among all modules. Rserver sends a command packet to all robots via the *Radio* submodule; the *SensorWatch* submodule receives the information from the robots.

View This module gives a graphical user interface where a human operator can know and also can command the game situation.

Soccer This module chooses the best strategy for a given situation, assigns each robot a role based on the chosen strategy, and computes velocity commands to perform the role for each robot.

See our ETDP 2017 [1] as for a bi-directional communication and the packets between the host computer and each robot.

In the last ETDP [2], we reported a trajectory tracking controller based on the model predictive controller. This time we focused on improving a skill for ball placement. See the details in Section 4.

3 Friction Reduction between a Small Wheel and Its Shaft by Using a POM Hub

A small wheel adopted for the 7G robot at first was composed of a hub of an A2017, known as duralumin, and a tire of a Nitrile Butadiene Rubber (NBR), as shown in Figs. 4 (a) and (b). The tire was also welded firmly to the circumference of the hub. The small wheel turns passively around a chrome-steel pin as a shaft.

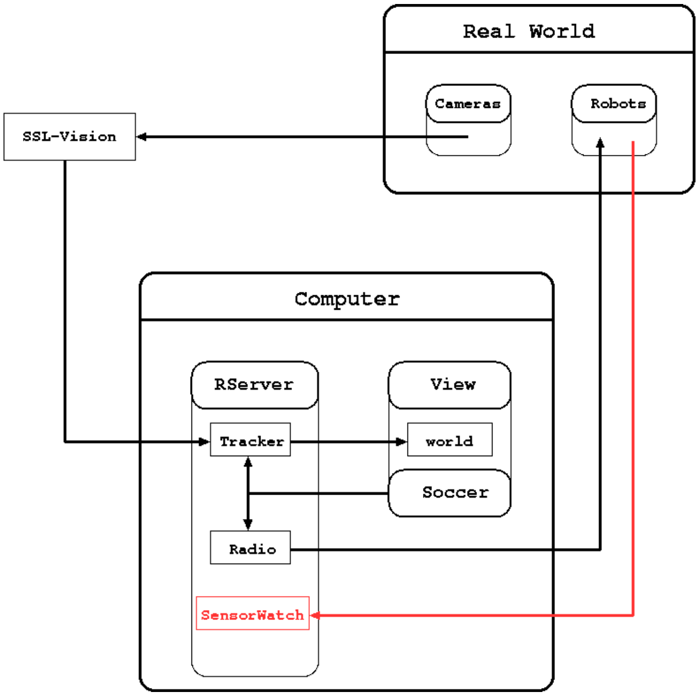
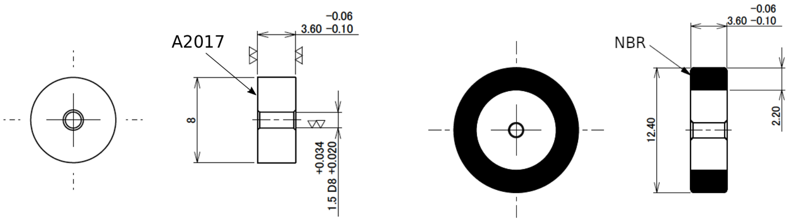


Fig. 3. Overview of the software system

This small wheel needed to be maintained sometimes by a kind of lubricant—usually used a sewing machine oil—because friction between the hub and the shaft affects omni-directional motion of the robot. This kind of maintenance could take a lot of trouble and time.

To solve the issue, we replaced the hub with a double-layered one composed of an inner part of Polyoxymethylene (POM), known as polyacetal, and an outer part of A2017, as shown in Fig. 5. The inner part is fixed to the outer part by press fit. The self-lubricity of POM gives reducing friction between the hub and the shaft. As in the initial version of a small wheel (Fig. 4 (b)), the NBR tire is welded to the (double-layered) hub. To the best of the authors' knowledge, the idea that POM can be used for the hub of the small-wheel in the SSL is first introduced by KIKS. We will see the information about it in their ETDP [3]. The only difference between KIKS' and our small wheels is whether the hub has a metal part or not. The A2017 part should improve decay durability of our small wheel.

We performed a simple experiment to confirm the effectiveness of our new small wheels. As depicted in Fig. 6, a robot was placed at $(-4, -2)$ on the world frame as an initial position. Then, the orientation of the robot is $\pi/2\,\mathrm{rad}$ $(= 90\,\mathrm{deg})$ with respect to the x-axis. We observed how the angle changed when letting the robot move straight parallel to the x-axis while keep the initial ori-



(a) A hub

(b) A hub with a tire

Fig. 4. A small wheel (before replacing)

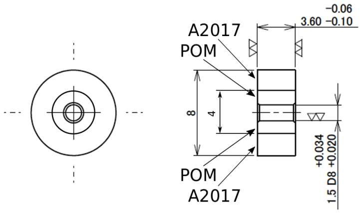


Fig. 5. A hub of a small wheel (after replacing)

entation. Note that we used a trajectory tracking controller for translational motion but feedforward command for angular motion.

Figure 7 shows time responses of x, y, and θ on the world frame. In each graph, the ideal trajectory is given by the velocity command; there are two kinds of actual trajectories: one is for the case of using non-POM hubs (*i.e.*, old small wheels) and the other is for the case of using POM hubs (*i.e.*, new small wheels). It can be seen that the actual trajectories of x and y track their ideal ones. The actual trajectory of θ for the case of using non-POM hubs keeps away from the ideal one; the actual trajectory of θ for the case of using POM hubs keeps around the ideal one without any feedback control. This can be considered that the POM hub resulted reducing friction between the small wheel and its shaft.

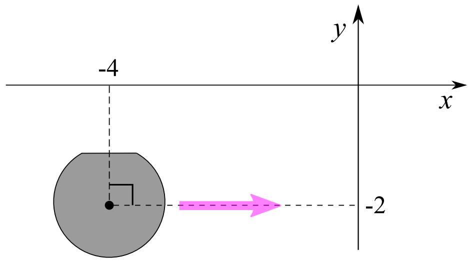


Fig. 6. Experimental Condition

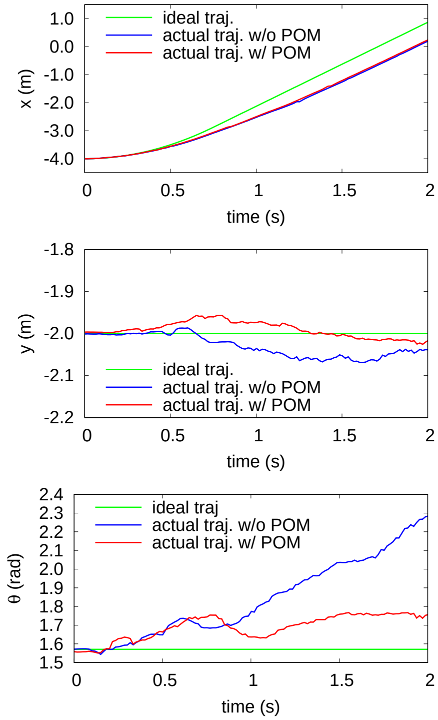


Fig. 7. Time responses

4 Kicking a Ball to the Wall during Ball Placement

Ball Placement has been officially introduced to the SSL rule since RoboCup 2017 through the technical challenge held in RoboCup 2016 [1]. The rule book [4] defines it as follows:

> After the game was stopped, the ball must be placed on the appropriate position, depending on the event that occurred. The automatic ball placement is the preferred way to place the ball at the designated position on the field by the robots of the teams without human interaction.

Ball placement is a mandatory task for all teams in Division A. RoboDragons has tried to improve a skill for ball placement so as not to bring disadvantages into games in Division A. Our main issue in this task is that the dribbler cannot be used for keeping a ball. This is because the dribbler of a non-repulsive rubber is slipper against a ball; due to several reasons, we have not replaced the drib-

---

[1] In RoboCup 2017, ball placement in the game was partially conducted because many teams were not ready to do it. In this context, RoboCup 2018 was the first time to manage this task for all games (of Division A).

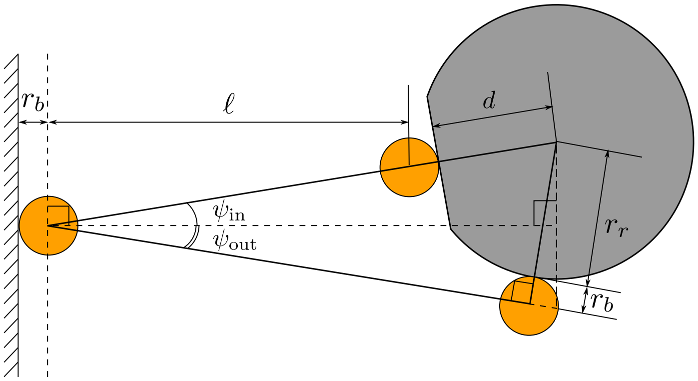


Fig. 8. Geometric Relationship in Diagonally Kicking a Ball to the Wall

bler with an appropriate one yet. Therefore, we have to develop skills for ball placement without using the dribbler.

Our skills for ball placement are mainly developed by pushing and passing. One of them is to kick a ball to the wall. This skill is important to take a ball near the wall without using the dribbler. Last year, we designed it by a quite simple algorithm that the robot kicks a ball to the wall and then backs away. In the skill, the front of the robot supposes to face to the wall. This skill, however, fails sometimes depending on tuning the parameters and the material of the wall. The typical situation is that a ball bouncing from the wall is hit to the robot. Then, the robot tries to kick the ball to the wall again. But, repeating failure induces a situation that the ball is locked between the wall and the robot.

Although one solution would be to estimate the reflection coefficient of the wall in advance, we here introduce another solution that the robot kicks a ball diagonally to the wall. A typical situation that a ball is near the wall can be modeled as in Fig. 8, where ψ_{in} is the angles of incidence to the wall, ψ_{out} is the angle of reflection from the wall, r_b and r_r are radii of the ball and the robot, d is the shortest distance from the robot's center to the front, and ℓ is the distance on the normal line against the wall between an initial ball position and a ball position where the kicked ball is touching the wall, respectively [2]. Let us formulate the design problem of this skill as follows:

Problem 1. Suppose that the position of a ball is given and also the position and orientation of a robot can be controlled. Then, find the minimum angle of ψ_{in} for the robot to kick a ball into the wall such that the bouncing ball from the wall cannot hit the robot.

Note that the minimum angle of ψ_{in} is useful to move on the next action smoothly.

First, to simplify the problem, we assume the following things:

(A1) the ball has a perfectly elastic collision with the wall;

---

[2] The robot and ball positions refer to their center positions.

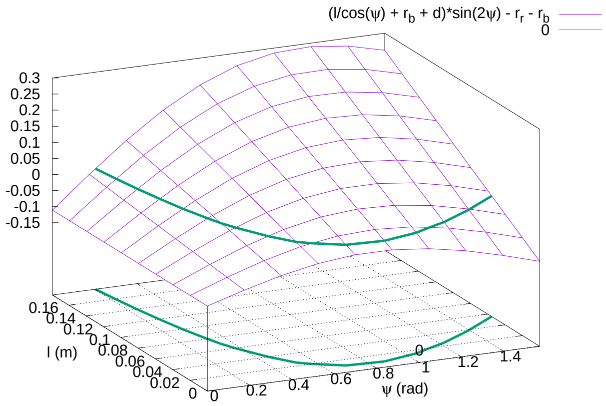


Fig. 9. Solution Curve of Eq.(1) in a Phase Space

(A2) affection of static/viscous friction between the ball and the carpet of the field is enough small.

Under (A1) and (A2), we can consider that ψ_{in} is equivalent to ψ_{out}. By using a constant ψ, the relationship can be represented as $\psi_{\text{in}} = \psi_{\text{out}} = \psi$. Then, focusing on a right triangle drawn by the solid line in Fig. 8, it can be found that the following geometric relationship holds:

$$\sin(2\psi) = \frac{r_r + r_b}{\frac{\ell}{\cos\psi} + r_b + d}. \tag{1}$$

The specific value of ℓ is given depending on a situation. Therefore, *Problem* 1 is reduced to finding the value of ψ that satisfies Eq. (1).

Solving Eq. (1) with respect to ψ is not easy still, but we can use any formula manipulation languages like Maple [5] and Mathematica [6]. Also, we can visualize the solution curve of Eq. (1) as in 9 You can find a demonstration video at `https://youtu.be/BWL8lP_vCgY`.

5 Concluding Remarks

We have presented the system configuration of RoboDragons 2019 robots. The main novelties of this ETDP are to replace small wheels of the omni-wheels with the new one using a POM hub and to present a new skill for ball placement.

Acknowledgement.

This work was supported by The Hibi Science Foundation, JSPS KAKENHI Grant Number JP16K00430, and Aichi Prefectural University.

References

1. Adachi, Y., Kusakabe, H., Suzuki, R., Du, J., Ito, M., and Naruse, T.: "RoboDragons 2017 Extended Team Description," RoboCup Soccer Small Size League, 2017.
2. Ito, M., Kusakabe, H., Adachi, Y., Suzuki, R., Du, J., Ando, Y., Izawa, Y., Isokawa, S., Kato, T., and Naruse, T.: "RoboDragons 2018 Extended Team Description," RoboCup Soccer Small Size League, 2018.
3. Watanabe, M., Sugiura, T., *et al.*: "KIKS 2019 Extended Team Description," RoboCup Soccer Small Size League, 2019 (to submit).
4. RoboCup Soccer Samll Size League: "Rules of the RoboCup Small Size League 2019," 2018-12-14 version (accessed on 29 Jan 2019); `https://robocup-ssl.github.io/ssl-rules/sslrules.html`
5. Maplesoft: "Maple, Version 2018," a division of Waterloo Maple, Inc., Waterloo, Ontario, 2018.
6. Wolfram Reearch, Inc.: "Mathematica, Version 11.1," Champaign, IL, 2018.