RoboDragons 2009 Team Description

Junya Maeno, Hiroki Achiwa, Junya Tamaki, Tatsuya Morihayashi, Yuki Arai, Kazuhito Murakami and Tadashi Naruse
Aichi Prefectural University, Nagakute-cho, Aichi, 480-1198 JAPAN

Abstract. After the two years joint project (2004, 2005) with CMDragons, RoboDragons developed some unique techniques that make the RoboDragons system flexible and robust. This paper gives a brief overview of the RoboDragons system and our unique techniques. These include abnormal behavior detection of robot, strategies based on safety region, kicking action analysis of robot.

1 Introduction

RoboDragons 2009 system builds on the RoboDragons 2008 system and the ongoing research. Each RoboDragons robot consists of four omni-directional wheels, a dribbling device, two kicking devices (for chip kick and straight kick) and computer boards for controlling the robot. Our team consists of five robots and they are controlled by a host computer beside the field. Over-field cameras grab the field images and send them to the host computer.

In the RoboDragons system, we have following three distinctive techniques.

1. Abnormal behavior detection of the robot: Our system detects the abnormal behavior of our robots with high accuracy using support vector machine (SVM). The abnormal behavior is discriminated by using commands sent to the robot by the system and the information from the vision system.
2. Strategies based on the safety region: We will develop strategies based on evaluation index. The safety region is defined as a region that the opponent’s shoot from the inside of the safety region can be blocked by the teammate robot(s). The safety region can be used as an index to evaluate the strategy. We propose approximate calculation of safety region and abstract strategies using safety region.
3. Kicking action analysis of robot: We propose the technique to detect kicking actions of robots. Proposed technique can detect kicking actions with high accuracy. This analysis can be useful to develop strategies.

In this paper, we give a brief overview of the RoboDragons system and then three distinctive techniques.
2 Team Member

Team Leader:

– Junya Maeno(cooperative control)

Team Members:

– Hiroki Achiwa(mechanics and vision)
– Junya Tamaki(system design)
– Tatsuya Moribayashi(vision)
– Yuki Arai(strategy and tactics)
– Kazuhito Murakami(supervisor)
– Tadashi Naruse(supervisor)

3 Overview of the RoboDragons system

In this section, we describe the overview of our robots and software system.

3.1 Robots

Fig.1 shows the RoboDragons robot.

![Fig. 1. RoboDragons robot](image)

The features of the robots are shown below.

– Each robot has four omni-wheels that allow it to move in any direction.
– Each robot has a main kick device and a chip kick device, and a dribble device. The main kick device is driven by a solenoid and the chip kick device is driven by two solenoids.
For each robot, one Hitachi SH2 processor is employed as a control processor which runs at 24MHz.

- Lithium-Polymer batteries are used.
- The number of small tires of the omni wheel is 15, while it was 12 for our old robot. It makes the movement of the robot smooth.

We have improved the kick device from 2008 to 2009. We have improved solenoids of the main kick device and the chip kick device. The main kick device can kick the ball of 9.5m/sec or less, and the chip kick device can kick the ball of the height 1m or less and the distance of 2m or less. The main change point is shown below.

- The copper wire was changed from 0.4mm to 0.7mm.
- The stroke was extended from 22mm to 30mm.
- The capacity has been increased from 1500µF to 4500µF.

In addition, we have improved the straight advancement of the kick. It can kick the ball straight by error margin within five degrees.

### 3.2 Software system

Fig. 2 shows the overview of the RoboDragons system. The features of the system are as follows:

![Fig. 2. RoboDragons system: overview](image)

1. Host computer is Athlon64 X2 4200+ with 512MB memory and Debian GNU/Linux OS.
2. Two or more Panasonic DVR-D310 cameras with RAYNOX HD-5050PRO lens and capture cards with Philips SAA7133 are used. Each camera grabs an image of size 640 × 240 pixels in every 1 / 60 seconds.
3. Each module (shown in box in Fig.2) is implemented as a thread.
4. Two or more image processing modules process the grabbed images from multiple cameras to produce the locations of the robots and a ball. They are integrated by the Cords Integrator module.
5. The Soccer module consists of a strategy, a tactics and a path generation submodules, and it produces an action command for each robot.
6. Finally, Radio module sends the command to each robot through the radio system.

4 The abnormal behavior detection of our robot

The vision system identifies the abnormal behavior of our robot. This system detects abnormality with support vector machine (SVM).

The method is shown follows.

1. The position and the direction of the robot are obtained from the vision system.
2. The movement parameter of the robot is obtained from the difference between the position and the direction of a previous frame and a current frame from the vision system.
3. The movement parameter is also obtained by the robot commands from the system.
4. The movement parameter which obtained from the system and the vision system are learned to SVM.
5. The learning data is used to discriminate abnormal from normal.

The movement parameter in this section includes a speed, an acceleration, an angular velocity, and an angular acceleration.

Fig.3 and Fig.4 show one example of the movement parameter that is obtained ideally. Fig.5 and Fig.6 show one example of the movement parameter that is obtained with the problem in the robot (ex. The control line is cut.) “Navigation” in figure is movement parameter that was given to the robot by the system. “Log” in figure is movement parameter that was made from the vision system.

![Fig. 3. Speed](image1)

![Fig. 4. Angular Velocity](image2)
The graph of “Navigation” and “Log” is corresponding when robot behaves normally. The graph of “Navigation” and “Log” is not corresponding in a specific frame when robot behaves abnormally.

When the speed, the acceleration, the angular velocity, and the angular acceleration were given as learning data, we can obtain the identification rate of about 86%.

5 Strategies based on the safety region

5.1 Safety region

Safety region is defined as a region that the opponent’s shoot from the inside of the safety region can be blocked by the teammate robot(s). Remaining region of the field given by removing the safety region is called unsafety region. The safety region can be used as an index to evaluate the strategy. We will develop the defending strategies by using this.

5.2 Approximate calculation of safety region

Since it is difficult to strictly get the safety region, we propose algorithms that compute the approximated safety region.

Let $b$ and $e$ be the positions of the ball and the shooting robot at time $t$, respectively, and $r_i$ be the position of each defending robot $i$ at time $t$. Let $L_r$ and $L_l$ be the lines connecting $e$ and the right goalpost and $e$ and the left goalpost, respectively. We assume the goalkeeper stands in the defence area and other defending robots stand outside defence area. Then, let $p_{r,i}$ and $p_{l,i}$ be the cross-points of the moving line of the robot $i$ and the lines $L_r$ and $L_l$, respectively. Assuming that the passing robot holds the ball at time $t$ and makes the cooperative play[1], following Eq. (1) is obtained for computing the safety region.

$$\frac{||b - e||}{v_p} + \frac{||e - p_{j,i}||}{v_s} > \sqrt{\frac{2(||r_i - p_{j,i}|| - R)}{a_i}}, \quad (j = r, l)$$

(1)
where, $v_p$, $v_s$, $a_i$ and $R$ are the speed of the ball at passing, the speed of the ball at shooting, the acceleration of the defending robot $i$ and the sum of radii of the defending robot and the ball, respectively.

When Eq. (1) holds for $j = r$ and $l$ or there is no pass line between $b$ and $e$, $e$ is a point in the safety region. In case there are more than one defending robot, if at least one of them satisfies Eq. (1), $e$ is defined as a point in the safety region. When $||r_i - p_{j,i}|| - R < 0$, $b$ is defined as a point in the safety region.

Eq. (1) compares the time which defending robot needs to block the most difficult shoot (toward the goal post) with the time which ball takes to move. Fig. 7 shows the result of the approximate safety region by Eq. (1). The ball is located 1500mm away from center of the field toward defending robot’s goal, and number of defending robots is three. Table 1 shows other parameters to compute. The white region in Fig. 7 is the safety region and the black region is unsafety region.

$$\text{Table 1. parameter}$$

<table>
<thead>
<tr>
<th>acceleration</th>
<th>maximum velocity</th>
<th>pass velocity</th>
<th>shoot velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0m/sec$^2$</td>
<td>2.0m/sec</td>
<td>5.0m/sec</td>
<td>10.0m/sec</td>
</tr>
</tbody>
</table>

5.3 Strategies using safety region concept

We propose strategies using safety region concept. As an example, we propose a strategy which maximize safety region on the field. For another instance, if opponent robot(s) is in unsafety region(s), teammate robots need some responses to opponent robot(s). So teammate robots mark the opponent robot preferentially. We will translate these into reality.
6 Kicking action detection of opponent robot

In this section, we propose the kicking detection algorithms using logged data.

6.1 Classification of kicking actions

The basic kicking actions of robot in SSL are classified into following detailed actions,

- “Kicking” actions:
  - “Shot”
  - “Pass”
  - “Clear”

We detect each action classified above in the logged data. Our logged data includes following data as the time series data. In the following, an object means a ball, a teammate robot, or an opponent robot.

- Time stamp,
- Referee signal,
- Position, direction, velocity and angle velocity of each robot,
- Position and velocity of ball,
- Reliability of each recognized object by image processing,
- Camera number that each object is captured.

These data are recorded every 1/60 seconds.

6.2 Detection and classification of kicking actions

The detection of the kicking actions here is extraction from the segment that the ball starts liner motion and ends it. First, pull out the linear motion segment from the trajectory of the ball, then check the segment whether it is caused by the kicking of a robot or not. Second, classify the action of the detected segment into detailed kicking action by an intention analysis. Third, classify the effect of the kicking action.

Algorithm for detecting kicking actions

The detection of the kicking segment consists of an extraction of maximal linear segment in the ball trajectory and a judgement what is caused the move of the ball. The algorithms are shown below.

In the algorithms, we use the following notation.

$(X_i,Y_i)$ is a position of the ball at frame $i$,

$\vec{a}_i$ is a vector from $(X_i,Y_i)$ to $(X_{i-1},Y_{i-1})$,

$\vec{b}_i$ is a vector from $(X_i,Y_i)$ to $(X_{i+1},Y_{i+1})$,
θ_i is an angle which is given by the following equation,

\[ \theta_i = \left\lVert 180 - \frac{180}{\pi} \cdot \cos^{-1} \left( \frac{\vec{a}_i \cdot \vec{b}_i}{\|\vec{a}_i\| \|\vec{b}_i\|} \right) \right\rVert \]  

(2)

\( \overline{\theta}_i \) is an average of \( \theta_i \) between frame \( i \) and \( i + n - 1 \), which is given by,

\[ \overline{\theta}_i = \frac{1}{n} \sum_{m=i}^{i+n-1} \theta_m \]  

(3)

\( V_i \) is an average velocity of a ball between frame \( i \) and \( i + n - 1 \), which is given by,

\[ V_i = \frac{1}{n} \sum_{m=i}^{i+n-1} \| \vec{b}_m \| \]  

(4)

\( f \) is a function that returns a value in proportion to the inverse of input value, i.e. if \( B = f(A) \) then \( B \propto A^{-1} \).

\( T_\alpha, T_\beta \) and \( T_\omega \) are threshold values, which are given in advance.

In the above, \( n \) is a given number for reducing noise by smoothing data.

Using above notation, first, we describe an algorithm to detect the linear motion segment.

**Algorithm 1** detection of linear motion segment

Extract a linear motion segment in the trajectory of the ball,

**Step 1** Set \( i \leftarrow 1 \)

**Step 2** Compute \( \{ \overline{\theta}_i < f(V_i) \} \wedge \{ V_i > T_\alpha \} \)  

(5)

If Eq. (5) satisfies, then set frame \( i \) as the starting frame \( s \) of the linear motion segment, set \( j \leftarrow i + 1 \) and go to Step 3. If not, set \( i \leftarrow i + 1 \) and repeat Step 2.

**Step 3** Compute \( \overline{\theta}_j \) and \( V_j \), and compute \( \{ \overline{\theta}_j < f(V_j) \} \wedge \{ \theta_{j+n} < T_\beta \} \)  

(6)

If Eq. (6) satisfies, then set \( j \leftarrow j + 1 \) and repeat Step 3. Otherwise, set frame \( j + n \) as the ending frame \( e \) of the linear motion. If the end of trajectory is reached, then exit, otherwise, set \( i \leftarrow j + n + 1 \) and go to Step 2.

When the segment is detected, a classification of the kicking action is done by the following algorithm.

**Algorithm 2** Classification of kicking action

For the detected segment, do the following.
Step 1 Let $s$ and $R_s$ be the starting frame of the linear motion segment and
the robot which is closest to the ball at frame $s$, respectively. Let $D_k$ be the
distance between the robot $R_s$ and the ball at frame $k$, where $s \leq k \leq s + n$.
Compute the following equations.

$$\left\{ D_s < T_\gamma \right\} \land \left\{ (D_s < D_{s+1}) \land \cdots \land (D_{s+n-1} < D_{s+n}) \right\} \quad (7)$$

$$\max_{0\leq k \leq n-2} \left\{ (D_{s+k+2} - D_{s+k+1}) - (D_{s+k+1} - D_{s+k}) \right\} > T_\omega \quad (8)$$

Eq. (7) shows that the ball goes away from the kicked robot $R_s$ and Eq. (8)
shows that the maximal acceleration in the segment should be greater than
the threshold $T_\omega$ if the ball is kicked by the robot $R_s$.

If Eq. (7) is true and Eq. (8) holds, the linear motion segment is caused by
the kicking action. Then, search the past frame. If the previous linear motion
segment ends at some frame $l$ ($s - m \leq l \leq s$), the kicking is a direct play\(^1\).

This leads that the value of $m$ is about 5.

Step 2 If Eq. (7) is true but Eq. (8) does not hold, the linear motion segment
is caused by the ball bouncing off the robot. If Eq. (7) is false, it is caused
by the unknown reason.

Intention of kicking actions

For the kicking actions satisfying both Eqs. (7) and (8), the intention of the
kicking actions is judged by the following algorithm.

Algorithm 3 Intention of kicking action

Step 1 Find the team (teammate/opponent) the robot $R_s$ belongs. It is easily
found in the logged data.

Step 2 Calculate the half line $L$ that begins at starting point $(X_s, Y_s)$ of
the linear motion and goes through ending point $(X_e, Y_e)$.

Step 3 If the half line goes across one of the objects, where objects are end-line
(EL), goal-line (GL), side-line (SL) and teammate robots of $R_s$ at frame $s$,
then find the closest object to the robot $R_s$ and its intersection $c$.

Step 4 From the intersection $c$, the intention of the kicking action is classified
as shown in Table 2.

Effect of kicking actions

The effect of the kicking action is often different from its intention. Therefore,
we need an algorithm that classifies the effect of the kicking action.

Algorithm 4 Effect of kicking action

Step 1 Find the team (teammate/opponent) the robot $R_s$ belongs. It is easily
found in the logged data.

Step 2 Calculate the line segment $L$ that begins at starting point $(X_s, Y_s)$ of
the linear motion and ends in ending point $(X_e, Y_e)$. Extend the segment for
the length of $D_L$ over the end point.

\(^1\) The direct play is an action that kicks the ball immediately after receiving it[1].
Table 2. Classification of kicking action

<table>
<thead>
<tr>
<th>$R_s \cap c$</th>
<th>TR’s EL</th>
<th>OR’s EL</th>
<th>TR’s GL</th>
<th>OR’s GL</th>
<th>SL</th>
<th>TR</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR</td>
<td>clear</td>
<td>shot</td>
<td>unknown</td>
<td>shot</td>
<td>clear</td>
<td>pass</td>
<td></td>
</tr>
<tr>
<td>OR</td>
<td>shot</td>
<td>clear</td>
<td>shot</td>
<td>unknown</td>
<td>clear</td>
<td>pass</td>
<td></td>
</tr>
</tbody>
</table>

TR denote teammate robot.
OR denote opponent robot.

Step 3 If the extended line segment goes across one of the objects, where objects are end-line (EL), goal-line (GL), side-line (SL) and teammate and opponent robots of $R_s$ at frame $e$, then find the closest object to the robot $R_s$ and its intersection $c$.

Step 4 From the intersection $c$, the effect of the kicking action is classified as shown in Table 3.

Table 3. Effect of kicking actions

<table>
<thead>
<tr>
<th>$R_s \cap c$</th>
<th>TR’sEL</th>
<th>OR’sEL</th>
<th>TR’sGL</th>
<th>OR’sGL</th>
<th>SL</th>
<th>TR</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR</td>
<td>clear</td>
<td>shot</td>
<td>miss</td>
<td>own goal</td>
<td>clear</td>
<td>pass</td>
<td>interception</td>
</tr>
<tr>
<td>OR</td>
<td>shot</td>
<td>clear</td>
<td>goal</td>
<td>own goal</td>
<td>clear</td>
<td>interception</td>
<td>pass</td>
</tr>
</tbody>
</table>

TR denote teammate robot.
OR denote opponent robot.

6.3 Experimental result

We got the 11 minutes logged data in RoboCup 2007 competitions. From them, one of the authors got extracted the kick intervals. We make them the supervisory data. Applying the algorithms described in the previous sections to the 11 minutes logged data, computer extracted those actions automatically. Then, we compared the supervisory data and the computer extracted data. Table 4 shows the results.

7 Conclusion

This team description paper gave a brief overview of robot hardware and software on the host computer of the RoboDragons 2009 system and then gave the distinctive techniques that we have developed. Continuing improvement of the software modules makes the RoboDragons system flexible and robust.
In SSL, the strategy of each team has been complicated every year. We can analyze the strategy quickly by using robot’s action analysis system, and that gives us a great advantage. Our robots keep the best conditions during games with proposed technique that detect the abnormal behavior. We propose the safety region concept and approximate calculation of it. We will develop flexible strategies by using safety region concept.

References

1. Ryota Nakanishi, James Bruce, Kazuhiro Murakami, Tadashi Naruse and Manuela Veloso, “Cooperative 3-Robot Passing and Shooting in the RoboCup Small Size League”, RoboCup 2006: Robot Soccer World Cup X, LNCS 4434 pp. 418-425