Foul Model

Bahia Robotics Team[1] **, FCPortugal**[2]

[1]Bahia State University, Brazil

[2]University of Aveiro, Portugal
University of Minho, Portugal
University of Porto, Portugal

1. Proposed Model

The model proposed will use the **useCharging** variable, found in the file **naosoccersim.rb**, in order to activate the method implemented in the **SoccerRuleAspect** class. The method will make use of *TouchGroups*, a vector of agents currently touching each other, in order to select the agents to be used in the calculations. This implementation will make use of some variables used in the model proposed last year by Patrick MacAlpine, but it uses different criteria to identify a foul.

Once the agents are selected, meaning we have a total of two agents of different teams and neither of them is on the ground, the model then will collect some variables, in order to proceed with the algorithm. These variables, for each player, are graphically represented in figure 1 and listed below.

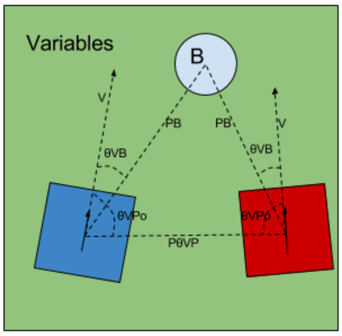

Figure 1. Graphical Representation of variables used by the model

- P_i: Position of the agent.
- V_i: Speed vector of the agent.
- S_i: Magnitude of the speed vector.
- D_i: Distance between the agent and the ball.
- $\overline{P_iB}$: Vector between the agent and the ball.
- $\overline{P_iP_o}$: Vector between both agents.
- θV_iB: Resulting angle between V_i and $\overline{P_iB}$.
- θV_iP_o: Resulting angle between V_i and $\overline{P_iP_o}$.

These values, throughout the model, will be compared with some thresholds, which are:

- $MinS_i$: Minimum speed value = 0.2m/s.
- $Min\theta V_iB$: Minimum value for the θV_iB angle = 30º.
- $Min\Delta D$: Minimum distance difference between players = 0.2m.
- $Min\Delta\theta V_iP_o$: Minimum difference between the angle θV_iP_o for each player = 15º.

The values proposed to these thresholds can be modified at any time, by changing the variables found in the **naosoccersim.rb** script.

The next step of the charging model is to evaluate which of the players committed a foul. To do that, it compares the agents intention to reach the ball, that is, the angle between their move direction and the direction to the ball, with their approach to the opponent, represented by the angle of their move direction with the position of the opponent. We also verify its angle in relation to the ball, to check if it passes a threshold. Both these verifications can be seen in the equations bellow.

$$\theta V_iB \geq \theta V_iP_o \tag{1}$$

$$\theta V_iB > Min\theta V_iB \tag{2}$$

Once these calculations are completed, we will have a boolean variable for each player, with the results. Leaving us with three different possibilities regarding how the collision occurred. These possibilities can be seen in figure 2.

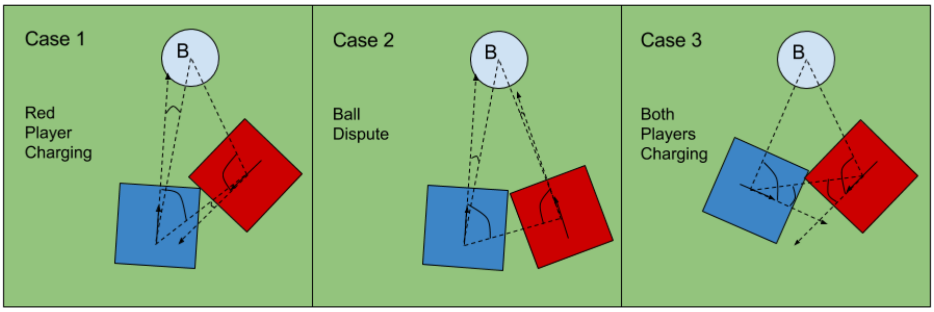

Figure 2. Possible situations resulting from Equation (1) and (2)

For the first case, when an agent is visibly colliding with its opponent without moving towards the ball, this player will be tagged as charging.

The second case requires some extra calculations, as it initially seems as both players are moving towards the ball. However, if one of the players is approaching the ball from behind, resulting in a collision, this player should be tagged as charging. The difference between both approaches can be seen in figure 3.

In order to verify if such a thing happened, we use the thresholds defined earlier, in order to verify if the following conditions are true:

$$|D_1 - D_2| > Min\Delta D \tag{3}$$

$$|\theta V_1P_2 - \theta V_2P_1| > Min\theta V_iP_o \tag{4}$$

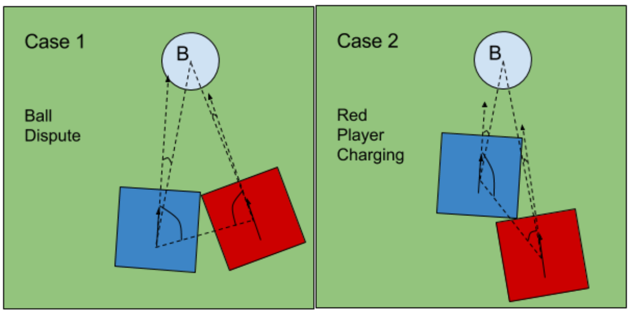

Figure 3. Difference between a ball dispute and a foul in the second case

If both statements are valid, the agent farthest from the ball will be tagged as charging.

For the third case, in which neither of the players seem to be moving towards the ball, the first proposition is to ignore the foul. If, however, this proves to be prejudicial to the league, it can be easily modified.

Once the cases are solved, and at least one player is thought to have committed a foul, we test one more value for that player, as shown in the inequality below:

$$S_i \geq MinS_i \tag{5}$$

This player will only be punished by charging if this condition is true. This guarantees that the player who committed the foul was actually moving, and not just standing still in the field.

As the charging model will be something entirely new to the teams, it's proposed that the initial punishment for these cases to be a beam outside of the field. Once the teams start committing less fouls throughout the game, it can be altered to change the playmode to a free kick for the opposite team.

2. Model Modifications

Once enough tests were made, we noticed some of the fouls were subjective, leading to some confusion regarding the reason the model charged those falls. One of the easiest situations to solve were collisions that lead to a stopped robot to fall, in such cases the fall speed could go over the previously established threshold and that agent could be charged with a foul penalty. To solve this issue, all that needed to be done was to increase the minimum speed.

The most complex problem noted was that the collision handle capture collisions for each body part of the agent, for each server cycle(0.02s). This resulted in small grazes counting towards a foul check, along with each collision between two robots being represented as multiple collisions for as long as they are still touching. In Fouls of the second case, when both agents seemed to be moving towards the ball, this would result in an initial ball dispute becoming a foul by the result of the robots collisions pushing them out of their initial path. To solve that, we decided to only verify the first actual contact between the robots, giving them an immunity time after the first collision, this

$ChargingImmunityTime$ is parametrized in the same file as the previous thresholds, being possible to remove it by simply altering it to zero.

Other than these two problems, two other adjustments were made to the model. The first one being that while inside the small area, the goalie(player with the 1 shirt) will not be charged with fouls, while making this change, we noticed fouls were being charged outside the field, so we corrected that as well. Secondly, we decided that collisions that happened too far from the ball wouldn't affect the game in any relevant way, therefore we added a maximum distance to ball for a foul to be charged. This value can also be changed in the same file.

Finally, it was proposed that the code should verify which part of the agent's body collided with the opponent, in order to check if the point of collision is coherent with the foul charged. This variable, containing the point of the collision, $CollisionPos$ was added to the $AgentState$, and by calculating the dot product between the robot's speed vector and the vector between its position and is $CollisionPos$, defined as PC_i, we can verify if it surpasses a given threshold, defined as $ChargingMinCollisionPoint$, also added to **naosoccersim.rb**. Its usage is shown both in the figure and in the inequality below. This verification is only applied once a player is already thought to be committing a charging foul.

$$V_i \cdot PC_i < ChargingMinCollisionPoint \tag{6}$$

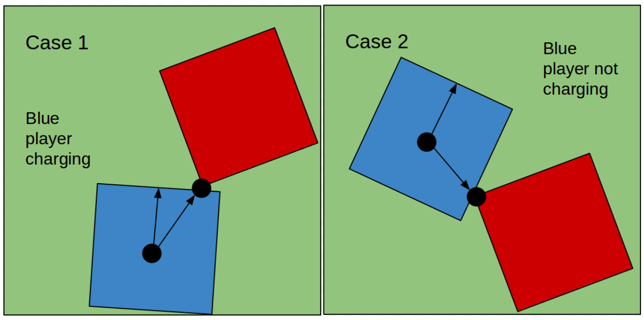

Figure 4. Charging point of collision verification

With the new modifications and adjustments, the new thresholds are:

- $MinS_i$= 0.4m/s.
- $Min\theta V_iB$ = 30º.
- $Min\Delta D$ = 0.2m.
- $Min\Delta\theta V_iP_o$: = 15º.
- $ChargingImmunityTime$ = 1s.
- $MaxBallDist$ = 10m.
- $ChargingMinCollisionPoint$ = 0.0.