

Foul Model for RoboCup 3D Soccer Simulation League

RoboCup 2016 3D Soccer Simulation Technical Committee

¹RoboCup Federation

1. Proposed Model

The model proposed will use the **useCharging** variable, found in the file **naosoccer-sim.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.

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

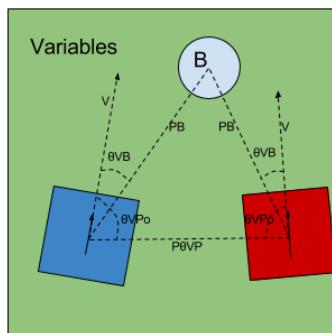


Figure 1. Graphical Representation of variables used by the model

- 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.
- $\theta V_i B$: Resulting angle between V_i and $\overline{P_iB}$.
- $\theta V_i P_o$: 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 = 0.3m/s.
- $Min\theta V_i B$: Minimum value for the $\theta V_i B$ angle = 30°.
- $Min\Delta D$: Minimum distance difference between players = 0.2m.
- $Min\Delta\theta V_i P_o$: Minimum difference between the angle $\theta V_i P_o$ for each player = 15°.
- $ChargingImmunityTime$: Time an agent becomes immune from foul verifications = 1s.

- *ChargingMaxBallDist*: Maximum distance for which the ball is taken into consideration for the foul = 1m.
- *ChargingMinBallDist*: Minimum distance from the ball for a foul to be checked = 0.1s.
- *ChargingMinCollisioSpeed*: Minimum speed for the collision as a whole, considering the sum of speed vectors = 0.2m/s.

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.

$$\theta V_i B \geq \theta V_i P_o \quad (1)$$

$$\theta V_i B > Min\theta V_i B \quad (2)$$

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.

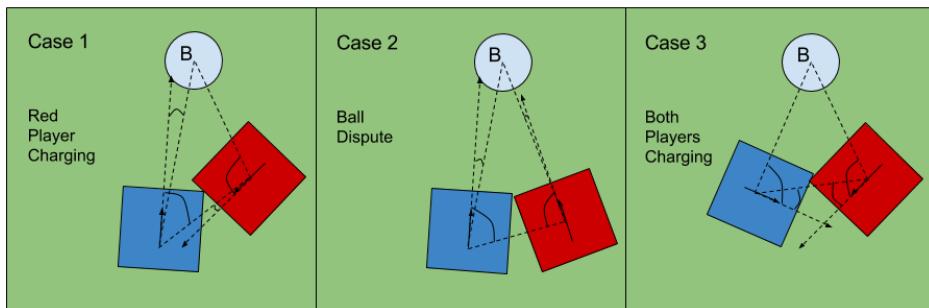


Figure 2. Possible situations resulting from Equation (1) and (2)

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:

$$|D_1 - D_2| > Min\Delta D \quad (3)$$

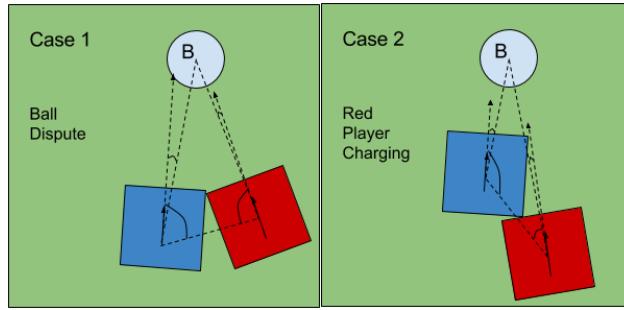


Figure 3. Difference between a ball dispute and a foul in the second case

$$|\theta V_1 P_2 - \theta V_2 P_1| > \text{Min}\theta V_i P_o \quad (4)$$

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.

These verifications are only taken in consideration if both players are within *ChargingMaxBallDist(1m)*. For the remaining cases, and as a extra check for these cases, we calculate the collision point.

In order to do that, *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 its *CollisionPos*, defined as *PC_i*, which is normalized, this value is calculated for both agents and then added up, as a vector sum, to obtain their relative speed, which is compared to *ChargingMinCollisionSpeed*. Its usage is shown both in the figure and in the inequality below.

$$V_i \cdot PC_i < \text{ChargingMinCollisionSpeed} \quad (5)$$

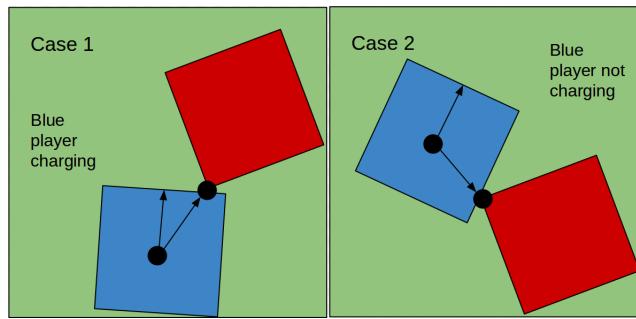


Figure 4. Charging point of collision verification

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:

$$S_i \geq \text{Min}S_i \quad (6)$$

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. Special conditions

Some conditions during play have made clear to us that we needed some extra conditions during the game for the foul model to work, these were:

- If a player touches the ball it is then immune to committing a charging foul for a period of time (ChargingImmunityTime currently set to 1 second).
- If an agent is fouled by an agent that is currently immune to committing charging fouls then the agent which is fouled is then immune to committing charging fouls for the remainder of the time that the agent who committed the foul remains immune to committing charging fouls.
- Goalies in their own penalty area can't be called for charging (same as the current foul model).
- If an agent is fouled by a goalie in the goalie's penalty area then the agent who was fouled is immune to committing charging fouls for ChargingImmunityTime.
- After the player is beamed for committing a foul, it's imune for 1s, before being able to foul again.

Acknowledgements

RoboCup Technical Committee thanks to teams FC Portugal, BahiaRT, UT Austin Vila and magmaOffenburg for their efforts on validation and implementation of this model.