PDEng Mechatronic System Design

Mathematics and Computer Science Department

Autonomous Referee System

Feasibility of MSD 2020 Simulation System

Project Team 2021

March 4, 2022

Contents

1 Introduction

MSD 2020 built a match simulation system in Simulink for the purpose of generating simulated game states to be used in verification / validation of their implemented AutoRef code. The system is comprised of the following sub-modules:

- A companion MATLAB script for variable initialization (player X_0, V_0, θ_0; ball X_0, V_0, θ_0; camera X_0, orientation; simulation settings)
- A Simulink model which integrates object positions over time from initial positions and velocities and outputs object trajectories over the entire simulation time.
- A virtual model file .WRL which renders the simulated match based on the computed object trajectories.

The simulation system generates video and / or image files which are then used as inputs for the MSD 2020 AutoRef system software to analyze the game and detect violations.

2 System Interpretation

Simulink Model: The model executes the basic function of simulation sufficiently, but there is room for improvement. In the "Final game simulation" folder's Simulink model, object trajectories are hard-coded into individual scripts which dictate each object's trajectory based on conditional statements over time. In other words, Player 1 starts with an initialized position and velocity generated by the companion MATLAB script, and then Player 1's trajectory script applies linear operations to compute position for each time step. Here's an example for one player's y coordinate computation:

```
function y = fcn(u)
%u = out.simout.time;
if u <= 5
y = -5.5;
else u > 5 && u <= 20
y = 0.2 * u - 6.5;
end
end
```

This means that for each desired gameplay simulation, each object's trajectory computation needs to be hard-coded.

3 Advantages and Disadvantages Analysis

Compared with other approaches, the option to use the MSD 2020 Simulation system has the following advantages:

- No new development would be required to generate additional game scenarios on which to test new AutoRef functions. Only new trajectory hardcoding would need to be created to simulate player and ball positions during the simulation.
- The MSD 2021 team's familiarity with MATLAB is high.

However, using MSD 2020's Simulation system also has the following disadvantages:

- Using the system would not contribute to MSD 2021's learning objectives.

- TechUnited has developed a fully-functional simulation system using their Turtles' autonomous behavior and real game physics which would provide a much richer, more detailed simulation with which to generate artificial game scenarios for testing of the AutoRef system.
- The process to define simulation object trajectories is highly manual and may not reflect real Turtle behavior.

4 Suggested Improvements

If further development and/or use of the MSD 2020 Simulation system is included in the MSD 2021 scope, the following development thrusts are suggested based from the interpretation of the MSD 2020 Simulation system.

1. Implement multiple camera viewpoints in the simulation environment and develop a workflow to perform multi-view image fusion for improved monitoring perspective. This could allow us to predict an optimal camera configuration (number and position) while minimizing the number of cameras required. It would be a good technical challenge and the feasibility seems reasonable.
2. Create a hybrid model-based and real image-based simulation system where real camera images of the field, players, and ball are segmented and applied as objects in the simulation environment. This could allow us to use the flexibility of the simulation environment and apply it to real images for testing. The technical challenge here may be the lowest compared to other options, therefore this option could be implemented in addition to another improvement option without much effort.