Fangzhaou Xia (xiafz@umich.edu), Zifeng Peng (pzf@umich.edu)

ME 461 Final Project: Virtual Tennis Game

INTRODUCTION

In the Virtual Tennis Game project, we utilized the idea of a Pong Game and placed it on a tennis court with side walls. To play the game, deploy the *CigarBoxSimulink_Tennis_Game_Arduino.slx*, run the *CigarBoxSimulink_Tennis_Game_Host.slx*, and have fun.

GAME FEATURE

The Virtual Tennis Game can be switched between three modes: Testing, 1 Player, and 2 Players. In each mode, three key parameters can be modified to make the game more interesting: gravity acceleration, horizontal velocity, and bouncing factor. Players can choose to have one all or two balls at the same time for playing and control the racket with the sliders. To kick off the ball, player 2 should enlarge gravity acceleration and bounce the ball. The game interface and control method can be found in Fig.1 and Fig. 2.

Fig. 1: Virtual Tennis Game User Interface

Fig. 2: Virtual Tennis Game Control Method

Switch Mode Encoding

Parameter Mode	Encode Value	Player No. SW 1	Ball No. SW 2	Play or Test SW3
1 Ball Welcome	0	0	0	0
1 Player 1 Ball	1	0	0	1
1 Ball AI Game	2	0	1	0
1 Player 2 Balls	3	0	1	1
1 Ball Testing	4	1	0	0
2 Players 1 Ball	5	1	0	1
2 Balls Testing	6	1	1	0
2 Players 2 Balls	7	1	1	1

Roter Parameter Control

	Rot1	Rot 2	Rot 3
Parameter	Bounce Factor	Horizontal Velocity	Gravity

IMPLEMENTATION

Based on the posted sample code, we used the Cigar Box as haptic sensors for controlling game parameters. In addition, we added a spring on to both of the sliders with equilibrium length at middle of the slider and stiffness determined by Rot 1. Ball locations are calculated on the host Simulink file.

Physical System: The dynamic system involved in the game is a bouncing ball moving in 2D. The trajectory of the ball is calculated by double integration of the acceleration in the vertical axis and integration of velocity in the horizontal axis in the discreet time domain at 10 Hz sampling rate.

Virtual Reality: With the help of Simulink interface for Virtual Reality Modeling Language (VRML), we managed to render the Simulink calculation output into a virtual reality tennis court as shown in Fig. 1. We modeled the tennis court basic setups with texture on a block in VRML and used the Simulink output values to control the translation of the balls and the batches.

Game Master: In the game master, we created a 160 by 160 virtual tennis court (unit in VRML) to hold tennis balls of 2.5 in radius. The horizontal velocity is a parameter associated to Rot 2 (from -10 to 10) and the vertical acceleration of the ball is a parameter associated with Rot 3 (from 0 to 20). When the ball goes beyond the border of the virtual tennis court, we check the position of the rackets to determine whether the ball will bounce back or fall out. The coefficient of restitution for the virtual ball and the simulated spring stiffness of the slider are both associated with Rot 1 (from 0 to 2). Once the all falls out of the bound, it will be placed to the center of the virtual tennis court with horizontal velocity unchanged and vertical velocity reversed. The scores for both players are then calculated automatically.

Reset: The scores and location of the balls are set to 0 once a switch has been turned indicating a change in the playing mode. This is done by checking the change in the value of the mode encoding.