King Fahd University of Petroleum and Minerals
Department of Computer Engineering

INTRODUCTION TO ROBOTICS COE 484

Homework No 4 (Due on May 31, 2008)

Questions	Grading
1	
2	
3	
4	
5	
TOTAL	

Student Name and ID: ____

Student Name and ID: ____

Behavior Programming for Soccer Game using SIMROBOT Simulator

Background:

The objective of this homework is to program the behavior of a humanoid robot for soccer game. A subsumptive control architecture consists of a graph of behaviors forming a hierarchical control structure, where the edges and vertices represent the behaviors and inhibiting control, respectively. All behavior takes as input some sensory data, carry out some processing (change from one state to another), and output some data to control actuators or other behaviors. Each behavior is an Augmented Finite State Machine (AFSM), i.e. a set of states where transitions are triggered based on the value of data captured from the environment like the relative position of robot, Ball, Own-Goal, Opponent-Goal, and possible Obstacles. The Behaviors are hierarchical in the sense that one behavior may subsume a lower level behavior by inhibiting its outputs. The German Team 2005 uses the XABSL as a programming tool to optimize the definition and running of the set of behaviors. In this assignment we do not use XABSL for behavior programming or execution. Here our objective is to write a set of behaviors for Soccer Game in C++ as part of the controller structure of SimRobot. At each step of the simulator, the programmed behaviors are to generate motion controls to the Graphic 3D Robot that was previously designed by the COE students. The motion of the Graphical Robot will be seen on the scene (GUI) generated by SimRobot which will allow us to appreciate the logic of the programmed behaviors, assess their adequacy, and modify their definition to meet Soccer Game operations and required intelligence.

Behavior Programming:

Here are the specifics of this assignment:

1. The environment for implementing the Soccer Behaviors consists of a set of behaviors which are Searching_ball, Walking_to_ball, Positioning_behind_ball, and kicking_ball that will be programmed as part of the SimRobot controller. SimRobot is used to simulate the 3D Graphical Robot (humanoid) which will receive motion command from the behaviors. SimRobot will display (show) the sequence of robot motion as part of it Scene displayed on its GUI.

2. To avoid possible data interfacing problems, we will not use the Low-Level Vision nor the Self-Localization and the data percept generated by them. Instead we will define a set of global data values for: (1) the robot absolute location on the soccer field, and (2) a set of relative position to robot which are Robot-to-ball, Robot-to-own-goal, Robot-to-opponent-goal, and possible Robot-to-obstacle. Usually, these parameters and others are provided by the vision and localization system.

3. **Initial state:** The robot has actuators allowing it to walk for some length L in some direction and to turn by some angle theta. Given the direction of its body, we assume the robot can move forward or backward by 0.2m in each walking step. We attach a frame F(X,Y) at the field circle center so that the Y axis is joining the centers of own-goal to opponent-goal and the X axis is the field midline. The robot body direction is specified by an angle Alpha = angle (robot-forward-direction, Y-axis). The width of the field is 4 meters and its length is 6 meters. The goal is 2-meter wide on each side of the field. The robot is initially

position at the field circle center, e.g. at location R/F=(0,0) with respect to F. Therefore, the relative positions of Robot can be set as:

i. Robot-to-ball=(-1,+1),
ii. Robot-to-own-goal=(0,-3),
iii. Robot-to-opponent-goal=(0,3), and
iv. We suppose the robot-to-obstacle=(0,1).

4. **Subsumptive Architecture:** The software architecture for the implementation consists of a main that sequentially invokes the procedures (behaviors) which are Searching_ball, Walking_to_ball, Positioning_behind_ball, and kicking_ball. Each procedure has access to the above global data. Each behavior-procedure is to carry out some state transition based on the value of a sub-set of the global data. Each state is to generate or not to generate some actuator motion parameters. If some motion parameters are generated (state dependent) by some behavior, then the parameters must be left in a global buffer called Behavior-Buffer like Walking_to_ball-Buffer. Also each procedure has a global Boolean called Behavior-Boolean. If a procedure A must inhibit the output of another procedure B, then A must write "False" in B-Boolean to inhibit potential actuator data from being applied during execution of current iteration. The only acceptable motion parameters are to move the robot by some amount (forward or backward), turning by some angle, and kicking with the right or left leg. The controller is be written so that execution will repeatedly carry out the following cycle (This implicit for SimRobot). The controller sequential steps are:

    a. **Initialize:** Initialize to "True" all the Booleans for all the procedure(behavior) and make sure all procedure buffers as cleared, e.g. no queuing from one iteration to another,

    b. **Run:** Execute the procedures in their sequential order as they appear in controller main,

    c. **Update/Output:** Visit each procedure buffer, if it has some actuator data, then apply the data to the previously defined Graphical Robot only if the corresponding Boolean is "True", else, ignore the buffer data.

5. **Interfacing to Graphical Robot:** In the scene of SimRobot GUI, the behavior program produces actuation data applied to the Graphical Robot which will lead to its motion towards the presumed location of the ball, positioning toward the opponent-goal, and kicking with right or left leg. This is accomplished by using some realistic tuning such as speed of motion and general appearance of the motion and kicking.
6. **Optional:** to make the scene realistic one may build a 3D orange ball that will appear at location a location that is defined by using Robot-to-ball=(X,Y) and R/F.
7. **Reporting/Demo:** The student is requested to write a short report including: (1) the AFSM of each behavior and the subsumptive (inhibiting) structure, (2) describe his program and coding of the controller for SimRobot, (3) include snap-shot of robot motion towards the goal, and (4) present a demo using SimRobot to the instructor.