CDA6530: Performance Models of Computers and Networks (Fall 2015)

Project 5: Wolf Chasing a rabbit (discrete-time simulation)

(Assigned 11/26; due: 12/10 midnight)

This project is modified based on a project in a course taught by Dr. Mark Claypool:
http://web.cs.wpi.edu/~claypool/courses/533-S04/projects/proj1/index.html

You are to simulate a wolf chasing a rabbit in discrete-time simulation approach in a flat surface area. Basically, the wolf and the rabbit see each other initially and begin running at the same time. The wolf tries to catch the rabbit and the rabbit tries to get away.

Because the simulation time step will affect the output, you must simulate with each time tick representing 0.2 seconds.

Details

Let (x(t), y(t)) be the position of the wolf at any time t (where distance is in feet and ***t* is in terms of seconds**). The rabbit is located at the position (p(t), 0) and is moving straight line in the positive x direction at any time t, i.e., the rabbit is running away following the X-axis to the right. In every time tick (0.2 seconds), the wolf always adjusts its running direction and runs towards the rabbit with a speed of s(t) beginning at t=0 in that direction during the entire time tick. Assume that (x(0), y(0)) = (0, 150), and the wolf was lying down at beginning so s(0) = 0. The initial position of the rabbit is (80,0).

In the discrete-time simulation, the phrase “wolf always runs towards the rabbit” means that at time tick k, the wolf determines its running direction towards the rabbit’s current position at time tick k, and runs in straight line in that direction until this time tick finishes. If the wolf’s straight-line trace during a time tick meets and crosses the X-axis, the wolf’s motion stops at the crossing location and continue from the beginning of the next time tick (i.e., do not considering running over the X-axis).

The wolf has the following running dynamics: it runs faster than the rabbit, but occasionally slips and falls and thus interrupts its running. The speed of the wolf is:

$s(t_{run}) = 3t_{run}$ for $0 < t_{run} < 5$

$s(t_{run}) = 15$ for $5 <= t_{run}$

where t_{run} is the time since the wolf last slipped and fell. To determine if the wolf slips and falls, call the function slip(t_{run}) (shown at the end of this document) at each time tick of your simulation. When slip happens, the wolf will begin to run again at the next time step and gradually speed up according to the above formulas.

The rabbit has the simple running dynamics with a constant speed of 5. Your simulation ends when the wolf catches the rabbit (i.e., wolf’s Y-axis is 0 and X-axis position is equal or bigger than rabbit’s X-axis position).

Simulation Results:

1. Simulate the wolf catching rabbit for 100 simulation runs. Compute the average catching time and its variance, and the average catching position and its variance.
2. Draw the wolf’s running trace for the first 3 simulation runs on one figure. That is to say, you need to draw the wolf’s running curve of the first 3 simulation runs in one figure. You only need

to show the part of trace where the wolf reaches the X-axis (i.e., don't need to show the trace when the wolf runs along the X-axis).

Submission:

1. A report containing A short description of your simulation code, and answer and figure to the above two questions in "simulation results" section.
2. The source code of any scripts you used as a separate code file (if you use Matlab for simulation, you can change the following slip(t) function into matlab code).

Appendix: the slip(t) function:

```
#include <stdlib.h>    /* for srandom() */
#include <time.h>      /* for time() */
#include <math.h>         /* for floor() */
/* Return 1 if slip, 0 if not.  */
int slip(double t) {
  double x;
  if (floor(t)==t){   /* slip could only happen when the time
                         is integer (in terms of seconds) */
     x = random() % 20;
     if (t > x)     return(1);
     else     return(0);
  }
  return(0);
}
```