COMP4807 Midterm Exam - Marking Scheme

/37

1. [1 mark] data is often invalid, or unpredictable situations

2. [2 marks] any 2 of quicker response, more robust code, simpler to code, easy to handle unforeseen situations

3. [1 mark] any 2 of noisy data, range limitations, resolution problems)

4. [1 mark] Both will display 23 properly ... there is no difference

5. Omitted

6. [1 mark] ServoControl
   [1 mark] Waits until the system counter reaches the specified parmeter value (provides delay = ½ mark)
   [1 mark] Needs to keep moving the wheels at the desired speed

7. [1 mark] Code that determines which behavior should have control over the robot. It perfoms a kind of behavior-conflict-resolution.
   [1 mark] behaviors compete and may conflict, arbitrator decides which one to use

8. [1 mark] One will rotate the servo clockwise ... the other counter clockwise.

9. [1 mark] Simple IR cannot detect if too close to wall ... forced to make loops

10. [1 mark] estimates become increasingly inaccurate over time, making them useless quite quickly

11. a) [1 mark] Distance between the two wheels ((diameter of robot = ½) 10.8 = ½)
    b) [1 mark] angular velocity or radians per second.
    c) [1 mark] No. [1 mark] When $v^r_t = v^l_t$ or $v^r_t = -v^l_t$ then **r** is undefined or unusable.

11 d) $\theta_{t+\delta} = \theta_t + \omega\delta \rightarrow \delta = (\theta_{t+\delta} - \theta_t) / \omega$

$$\begin{aligned}\delta &= (\theta_{t+\delta} - \theta_t) / ((v^r_t - v^l_t) / L) \\ &= (2.618) / ((1 - -1) / 10) \\ &= (2.618) / (2 / 10) = 2.618 / 0.2 = 13.09 \text{ sec}\end{aligned}$$

1 mark = right equation
1 mark = converting angles to radians
1 mark = correct answer (may be off a little)

11 e)

$r = L/2 * (v^l_t + v^r_t) / (v^r_t – v^l_t) = 5(9)/1 = 45\text{cm}$

$\omega = (v^r_t – v^l_t) / L = 1/10 = 0.1$

$x_{t+\delta} = r\cdot\mathbf{cos}\,\omega\delta\,\mathbf{sin}\,\theta_t + r\cdot\mathbf{cos}\,\theta_t\,\mathbf{sin}\,\omega\delta + x_t – r\,\mathbf{sin}\,\theta_t$

$= 45(0.54)(0.866) + 45(-0.5)(0.841) + 0 – 45(0.866) = 3.15$

$y_{t+\delta} = r\cdot\mathbf{sin}\,\omega\delta\,\mathbf{sin}\,\theta_t - r\cdot\mathbf{cos}\,\theta_t\,\mathbf{cos}\,\omega\delta + y_t + r\,\mathbf{cos}\,\theta_t$

$= 45(0.841)(0.866) - 45(-0.5)(0.54) + 40 + 45(-0.5) = 62.45$

$\theta_{t+\delta} = \theta_t + \omega\delta = 2.375 \text{ radians} = 177°$

Note that the answers will vary according to calculators…accept any reasonable close answer

1 mark = correct radius
1 mark = correct omega
1 mark = correct x
1 mark = correct y
1 mark = correct theta

12.

```
if (turnCount > 0) turnCount--;           [1 mark] something similar is ok
else {
      int avoidLeft = 0;
      int avoidRight = 0;
      Random ranGen = new Random();
      if (leftProxSensor.getValue() > 0)
            if (rightProxSensor.getValue() > 0)
                  if (ranGen.next() % 2 == 0)
                        avoidLeft = 1;
                  else
                        avoidRight = 1;   after this add turnCount = 12; [1 mark]
            else
                  avoidRight = 1;
      else
            if (rightProxSensor.getValue() > 0)
                  avoidLeft = 1;
}
```

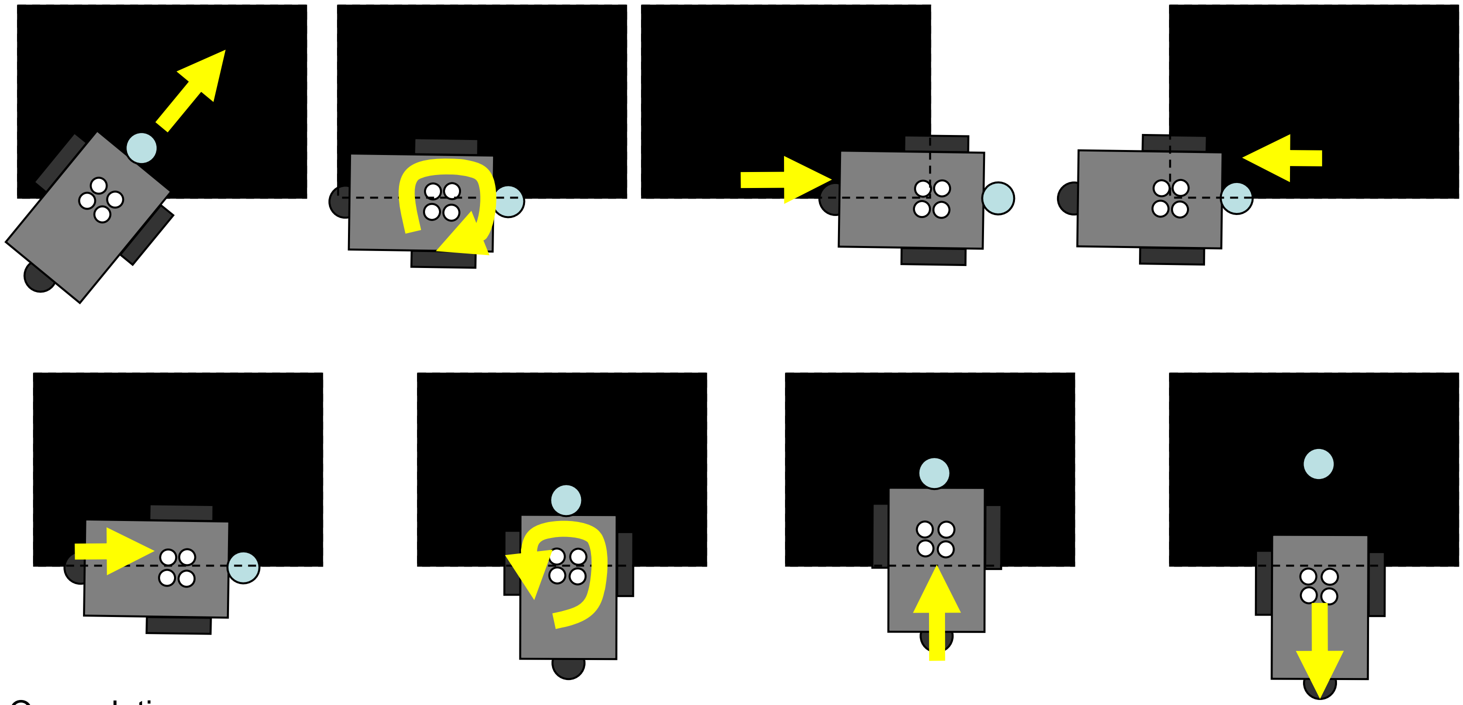
When reaching a corner, the robot may oscilate back and forth [1 mark]

13. (a) [1 mark] All of them
(b) [1 mark] BDEF
(c) [1 mark] E, because it usese grey code so only one bit changes at a time between consecutive ticks

(d) 40 pulses per rotation.
100 pulses = 2.5 wheel rotations = 2.5 circumference = 2.5 (πD) = 125π = 393mm
[1 mark – correct formula of 2.5 (πD) or 5πR]
[1 mark for correct answer]

14.

1 mark if they considered encountering black paper at an angle
1 mark if they have the notion of traveling half way across the black paper using encoder counts
1 mark if they have a strategy that somehow aligns the robot to the paper's edge
1 mark if they attempt to determine a reference point … perhaps look for corner of black
1 mark if their idea of heading towards the center seems valid



One solution:

1. Use 4 sensors as shown.
2. Move until black encountered.
3. Spin to align to edge (left 2 sensors detect on, right 2 off).
4. Move (and maintain alignment) along edge until corner detected.
5. Back up (stay aligned) until other corner to find out if it is the 8.5" or 11" edge
6. Move forward ½ way along edge
7. Spin until perpendicular (+- 90 degrees)
8. Move straight ahead either 4.25" or 5.5" (minus offset for block) depending on which edge was detected before.
9. Stop and back up, leaving block behind.