

Contemporary Data Processing Technology (CCOD)

Lab 2 (September 15, 2016)

Potapiuk Aleksei, AS-36

Control two metro cars using Speed, Distance, and Brake. In this work we are design a virtual loop with 1000 pixels on which two metro cars(A – first train car and B – second train car). We don't assume any stations. That is, both train always run. The speed of train is denoted as X_a and X_b . Speed ranges from 0 to 20 pixels per unit of time. The distance from train A to B is denoted as Y_a and from train B to train A is Y_b , so $Y_a + Y_b = 1000$. Both train move in a clockwise direction. At the beginning speed of trains is 10 pixels per unit of time ($X_a = 10$, $X_b = 10$) and $Y_a = 500$, $Y_b = 500$; Step by step the speed changes by -2, 0, +2 randomly. Speed will be controlled by the distance to the train in front via its brake. The break of train A and B is denoted as Z_a and Z_b . And we change break every step with the following algorithm:

```
{if  $x \geq 14$  then  $z=1$ ,  $x=x-1$ } & {if  $y \leq 50$  then  $z=9$ ,  $x=x-9$ }
```

We also need to show the membership value of 3 rules:

```
{IF  $x=medium$  AND  $y=small$  THEN  $z=strong$ } OR {IF  $x=medium$  AND  $y=medium$  THEN  $z=medium$ } OR {IF  $x=medium$  AND  $y=large$  THEN  $z=weak$ }
```

I wrote a program that simulates this virtual metro. The result of the program shown at the bottom.

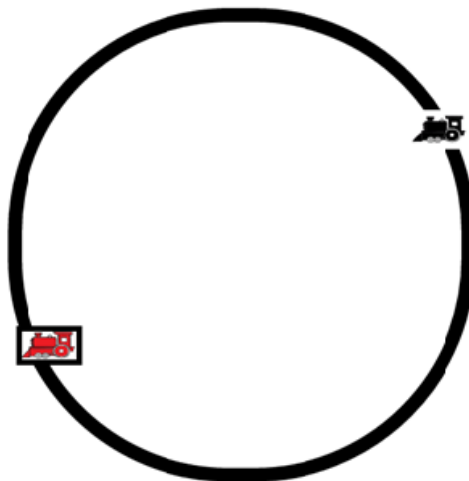
1st
speed: 12

red to black distance: 522
black to red distance: 478

2nd
speed: 9

break: 0

break: 0



Calculation of membership value of 3 rule according to the screenshot above.

IF x=medium AND y=small THEN z=strong

A	speed	μ	Distance	μ	break	μ	total μ
	12	0.5	522	0	0	0	0
B	speed	μ	Distance	μ	break	μ	total μ
	9	0.75	488	0	0	0	0

IF x=medium AND y=medium THEN z=medium

A	speed	μ	Distance	μ	break	μ	total μ
	12	0.5	522	0,585	0	0	0
B	speed	μ	Distance	μ	break	μ	total μ
	9	0.75	488	0.415	0	0	0

IF x=medium AND y=large THEN z=week

A	speed	μ	Distance	μ	break	μ	total μ
	12	0.5	522	0.585	0	0	0
B	speed	μ	Distance	μ	break	μ	total μ
	9	0.75	488	0,415	0	0	0

$\mu_{\text{Speed}} = 12 \Rightarrow \mu_s = 0;$
 $\mu_{\text{Dist}} = 522 \Rightarrow \mu_d = 0.585;$
 $\mu_{\text{Brake}} = 0 \Rightarrow \mu_b = 0;$
 $M = \min(\mu_s, \mu_d) * \mu_b \Rightarrow M = 0 * 0 = 0$

Membership value:

As shown above, in every situation the membership value is equal to 0. Because of Z only takes the following values: 0, 1, 9. Membership function strong, medium and weak is always equal to zero when break takes these values.

A: Rule1 or Rule2 or Rule3 $\Rightarrow 0$

B: Rule1 or Rule2 or Rule3 $\Rightarrow 0$

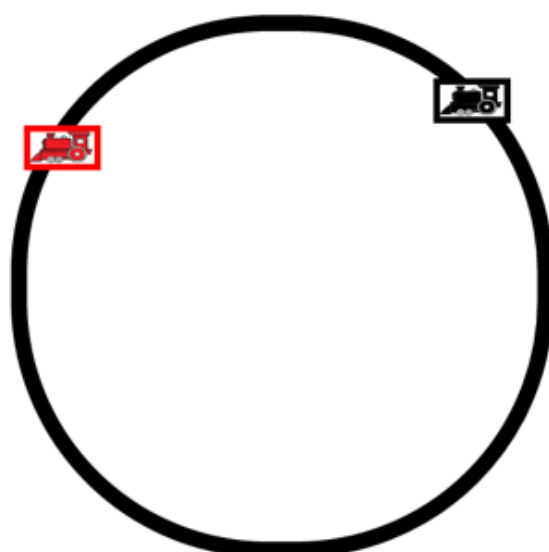
1st
speed: 13

red to black distance: 285
black to red distance: 715

2nd
speed: 15

break: 0

break: 0



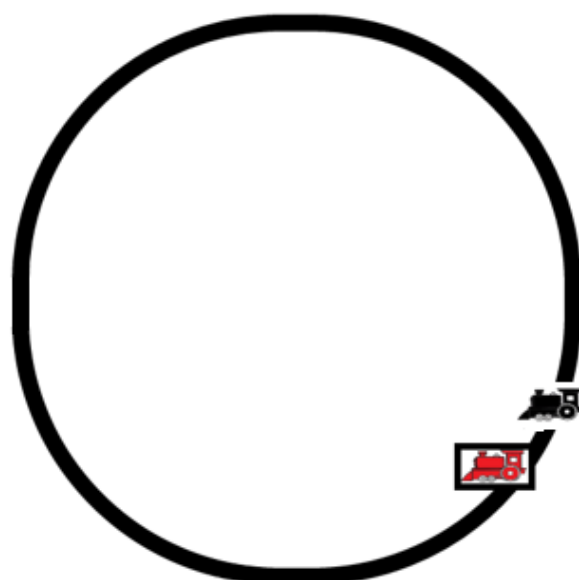
1st
speed: 4

red to black distance: 950
black to red distance: 50

2nd
speed: 1

break: 0

break: 9



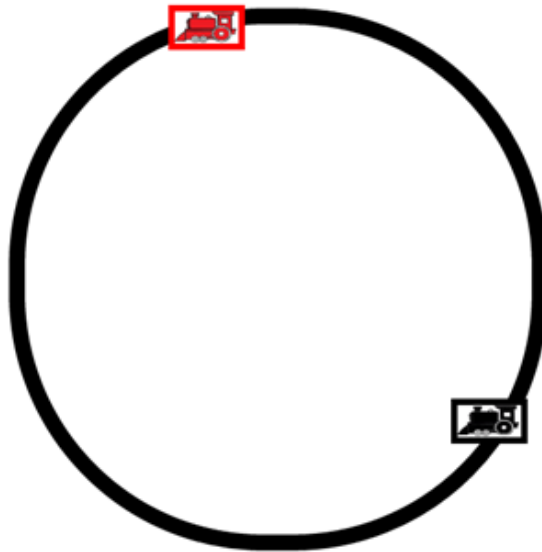
1st
speed: 18

red to black distance: 368
black to red distance: 639

2nd
speed: 15

break: 1

break: 1



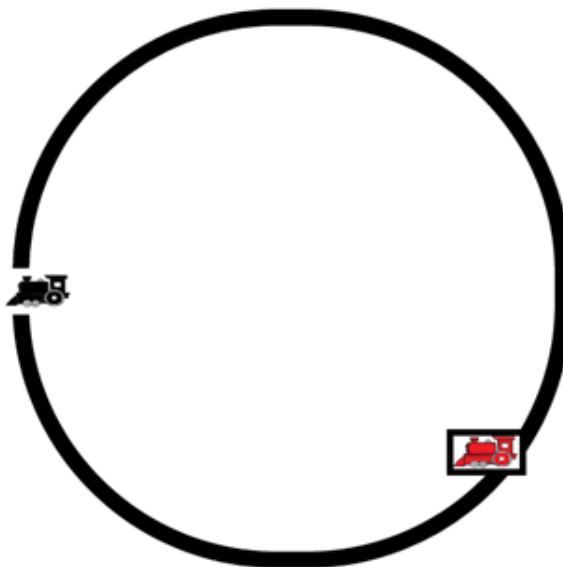
1st
speed: 13

red to black distance: 350
black to red distance: 650

2nd
speed: 15

break: 0

break: 0



Source code:

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title></title>
  <link rel="stylesheet" href="css/style.css">
</head>

<body>

  <div class="subway_loop">
    
    
  </div>

  <div class="speed redtrainSpeed">red speed: <span
class="redtrainSpeedValue"></span></div>
  <div class="speed greentrainSpeed">green speed: <span
class="greentrainSpeedValue"></span></div>

  <div class="distance redtrainDistance">red to green distance: <span
class="redtrainDistanceValue"></span></div>
  <div class="distance greentrainDistance">green to red distance: <span
class="greentrainDistanceValue"></span></div>

  <div class="brake brakeredtrain">red train brake: <span
class="brakeredtrainValue"></span></div>
  <div class="brake brakegreentrain">green train brake: <span
class="brakegreentrainValue"></span></div>

  <script src="js/script.js"></script>
</body>

</html>
```

Script.js

```
var greentrain = document.querySelector(".greentrain");
var redtrain = document.querySelector(".redtrain");
var redtrainSpeedValue = document.querySelector(".redtrainSpeedValue");
var redtrainDistanceValue = document.querySelector(".redtrainDistanceValue");
var greentrainSpeedValue = document.querySelector(".greentrainSpeedValue");
var greentrainDistanceValue =
document.querySelector(".greentrainDistanceValue");
var brakegreentrainValue = document.querySelector(".brakeredtrainValue");
var brakeredtrainValue = document.querySelector(".brakegreentrainValue");
var startSpeed = 0.001;
var x0 = 77;
var y0 = 74;
var r = 79;
var i = -1.5;
var j = 1.5;
var start = Date.now();
var redSpeed = 0.001,
    greenSpeed = 0.001;
var rand;
var redway = 0,
```

```

    greenway = 0;

var rightRed = redtrain.style.left + 20;
var leftRed = redtrain.style.left;

var rightGreen = greentrain.style.left + 20;
var leftGreen = greentrain.style.left;

var topRed = redtrain.style.top;
var bottomRed = redtrain.style.top - 10;

var topGreen = greentrain.style.top - 10;
var bottomGreen = greentrain.style.top;

var timer = setInterval(function () {
    brakegreentrainValue.innerHTML = 0;
    brakeredtrainValue.innerHTML = 0;
    var timePassed = Date.now() - start;
    redtrain.style.left = x0 + r * Math.cos(i) - 35 + 'px';
    redtrain.style.top = y0 + r * Math.sin(i) - 5 + 'px';

    greentrain.style.left = x0 + r * Math.cos(j) - 10 + 'px';
    greentrain.style.top = y0 + r * Math.sin(j) - 5 + 'px';

    i = i + redSpeed;
    j = j + greenSpeed;

    rand = randomValue(-2, 2) / 10000;
    if (redSpeed > 0 || rand > 0) {
        redSpeed += rand;
    }
    if (redSpeed > 0.002) redSpeed = 0.002;
    redway += redSpeed;

    redtrainSpeedValue.innerHTML = Math.round(redSpeed * 10000);

    redtrainDistanceValue.innerHTML = 500 + Math.round(Math.abs(redway -
greenway) / 6.0414 * 1000);

    rand = randomValue(-2, 2) / 10000;
    if (greenSpeed > 0 || rand > 0) {
        greenSpeed += rand;
    }
    if (greenSpeed > 0.002) greenSpeed = 0.002;
    greenway += greenSpeed;

    greentrainSpeedValue.innerHTML = Math.round(greenSpeed * 10000);

    greentrainDistanceValue.innerHTML = 1000 - (500 +
Math.round(Math.abs(redway - greenway) / 6.0414 * 1000));
    //{if x>=14 then z=1, x=x-1}
    if (Math.round(greenSpeed * 10000) >= 14) {
        greenSpeed -= 1;
        brakegreentrainValue.innerHTML = "1";
        console.log("green 1")
    }
    if (Math.round(redSpeed * 10000) >= 14) {
        redSpeed -= 1;
        brakeredtrainValue.innerHTML = "1";
        console.log("red 1")
    }
}

```

```

    //{if y<=50 then z=9, x=x-9}
    if (1000 - (500 + Math.round(Math.abs(redway - greenway) / 6.0414 * 1000))
    <= 50) {
        greenSpeed -= 9;
        brakegreentrainValue.innerHTML = "9";
    }
    if (500 + Math.round(Math.abs(redway - greenway) / 6.0414 * 1000) <= 50) {
        redSpeed -= 9;
        brakeredtrainValue.innerHTML = "9";
    }

    if (redSpeed < 0) redSpeed = 0;
    if (greenSpeed < 0) greenSpeed = 0;
    if (Math.abs(redway - greenway) > 3.02) {
        clearInterval(timer);
    }

}, 0.5);

function randomValue(min, max) {
    return Math.round(min - 0.5 + Math.random() * (max - min + 1));
}

```