

Contemporary Data Processing Technology (CCOD) Lab 2
Aksiutchyk Vitali, AS-36
Controll two metro cars using Speed, Distance, and Brake

Controll 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.

Script.js

```
var blacktrain = document.querySelector(".blacktrain");
var redtrain = document.querySelector(".redtrain");
var redtrainSpeedValue = document.querySelector(".redtrainSpeedValue");
var redtrainDistanceValue = document.querySelector(".redtrainDistanceValue");
var blacktrainSpeedValue = document.querySelector(".blacktrainSpeedValue");
var blacktrainDistanceValue = document.querySelector(".blacktrainDistanceValue");
var brakeblacktrainValue = document.querySelector(".brakeredtrainValue");
var brakeredtrainValue = document.querySelector(".brakeblacktrainValue");
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,
    blackSpeed = 0.001;
var rand;
var redway = 0,
    blackway = 0;

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

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

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

var topGreen = blacktrain.style.top - 10;
var bottomGreen = blacktrain.style.top;
```

```

var timer = setInterval(function () {
    brakeblacktrainValue.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';

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

    i = i + redSpeed;
    j = j + blackSpeed;

    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 - blackway) / 6.0414 * 1000);

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

    blacktrainSpeedValue.innerHTML = Math.round(blackSpeed * 10000);

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

    //start second task code
    //{if x>=14 then z=1, x=x-1}
    if (Math.round(blackSpeed * 10000) >= 14) {
        blackSpeed -= 1;
        brakeblacktrainValue.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 - blackway) / 6.0414 * 1000)) <= 50) {
        blackSpeed -= 9;
        brakeblacktrainValue.innerHTML = "9";
    }
    if (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000) <= 50) {
        redSpeed -= 9;
        brakeredtrainValue.innerHTML = "9";
    }
}

```

```

//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=week}
if (Math.round(blackSpeed * 10000) >= 6 && Math.round(blackSpeed * 10000) <= 14) {
  if (1000 - (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000)) <= 100) {
    blackSpeed -= 8;
  } else if (1000 - (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000)) > 100 &&
    1000 - (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000)) <= 200) {
    blackSpeed -= 5;
  } else if (1000 - (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000)) > 200 &&
    1000 - (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000)) <= 300) {
    blackSpeed -= 2;
  }
}
}

if (Math.round(redSpeed * 10000) >= 6 && Math.round(redSpeed * 10000) <= 14) {
  if (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000) <= 100) {
    redSpeed -= 8;
  } else if ((500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000) > 100 &&
    (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000) <= 200))) {
    redSpeed -= 5;
  } else if ((500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000) > 200 &&
    (500 + Math.round(Math.abs(redway - blackway) / 6.0414 * 1000) <= 300))) {
    redSpeed -= 2;
  }
}
}

if (redSpeed < 0) redSpeed = 0;
if (blackSpeed < 0) blackSpeed = 0;
//end second task code
if (Math.abs(redway - blackway) > 3.02) {
  clearInterval(timer);
}
}, 0.5);

```

```

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

```

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 blackntrainSpeed">black speed: <span class="greentrainSpeedValue"></span></div>

<div class="distance redtrainDistance">red -> distance: <span
class="redtrainDistanceValue"></span></div>
<div class="distance blacktrainDistance">black -> distance: <span
class="greentrainDistanceValue"></span></div>

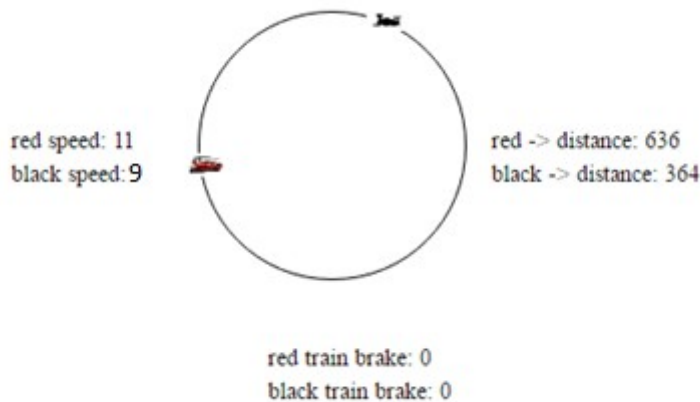
<div class="brake brakeredtrain">red train brake: <span class="brakeredtrainValue"></span></div>
<div class="brake brakeblacktrain">black train brake: <span
class="brakegreentrainValue"></span></div>

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

</html>

```

Results:



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

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

Red	speed	μ	Distance	μ	break	μ	total μ
	11	0.5	636	0	0	0	0
Black	speed	μ	Distance	μ	break	μ	total μ
	9	0.75	364	0	0	0	0

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

A	speed	μ	Distance	μ	break	μ	total μ
	11	0.5	636	0,585	0	0	0
B	speed	μ	Distance	μ	break	μ	total μ
	9	0.75	364	0.415	0	0	0

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

Red	speed	μ	Distance	μ	break	μ	total μ
	11	0.5	636	0.585	0	0	0
Black	speed	μ	Distance	μ	break	μ	total μ
	9	0.75	364	0,415	0	0	0

$\mu_{\text{Speed}} = 11 \Rightarrow \mu_s = 0;$
 $\mu_{\text{Dist}} = 636 \Rightarrow \mu_d = 0.585;$
 $\mu_{\text{Brake}} = 0 \Rightarrow \mu_b = 0;$
 $M = \min(\mu_s, \mu_d) * \mu_b \Rightarrow M = 0 * 0 = 0$

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 $\Rightarrow 0$

