

Contemporary Data Processing Technology (CCOD)

Lab 2 (September 15, 2016)

Shurpo Dmitry, AS-36

Controll two metro cars using Speed, Distance, and Brake

In this practice we made a model of 2 trains on plain loop with length of 1000 points. Every train at the beginning have the speed of 10 points and move in clockwise stream.

On each iteration we change speed adding -2, 0, +2 at random. Speed should be ≥ 0 and < 20 .

Instead of membership value of speed

Brake z is not 0 like last time but $\Rightarrow$ {if $x \geq 14$ then $z=1$, $x=x-1$ } & {if $y \leq 50$ then $z=9$, $x=x-9$ }

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

Every step, Speed x is added +2,0,-2 at random

When the distance between trains is 0 we stop the movement.

The program can be run in every browser(it's an .html page)

Here is the source code:

Style.css

```
body {  
background:url(../img/3.jpg);  
background-size: cover;  
color: white  
}  
  
.subway_loop {  
margin: 10% auto;  
width: 159px;  
height: 159px;  
border: 1px solid white;  
border-radius: 80px;  
}  
  
.train {  
position: relative;  
border: 1px solid;  
}  
  
.firsttrain {  
left: 50%;  
top: 0;  
border-color: red;  
}  
  
.secondtrain {  
left: 50%;  
top: 100%;  
border-color: gray;  
}  
  
.speed {  
position: fixed;  
font-size: 14px;  
left: 5%;  
}  
  
.firsttrainSpeed {  
bottom: 0px;
```

```

}
.secondtrainSpeed {
    bottom: 20px;
}
.distance{
    position: fixed;
    font-size: 14px;
    left: 35%;
}
.firsttrainDistance {
    bottom: 0px;
}
.secondtrainDistance {
    bottom: 20px;
}
.brake {
position: fixed;
font-size: 14px;
left: 75%;
}
.brakefirsttrain {
bottom: 0px;
}
.brakesecondtrain {
bottom: 20px;
}

```

Script.js

```

var secondtrain =
document.querySelector(".secondtrain");
var firsttrain =
document.querySelector(".firsttrain");
var firsttrainSpeedValue =
document.querySelector(".firsttrainSpeedValue")
;
var firsttrainDistanceValue =
document.querySelector(".firsttrainDistanceValue");

```

```

var secondtrainSpeedValue =
document.querySelector(".secondtrainSpeedValue"
);
var secondtrainDistanceValue =
document.querySelector(".secondtrainDistanceValue");
var brakessecondtrainValue =
document.querySelector(".brakefirsttrainValue")
;
var brakefirsttrainValue =
document.querySelector(".brakessecondtrainValue"
);
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 firstSpeed = 0.001,
    secondSpeed = 0.001;
var rand;
var firstway = 0,
    secondway = 0;

var rightFirst = firsttrain.style.left + 20;
var leftFirst = firsttrain.style.left;

var rightSecond = secondtrain.style.left + 20;
var leftSecond = secondtrain.style.left;

var topFirst = firsttrain.style.top;
var bottomFirst = firsttrain.style.top - 10;

var topSecond = secondtrain.style.top - 10;
var bottomSecond = secondtrain.style.top;

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

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

    i = i + firstSpeed;
    j = j + secondSpeed;

```

```

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

    firsttrainSpeedValue.innerHTML =
Math.round(firstSpeed * 10000);

    firsttrainDistanceValue.innerHTML = 500 +
Math.round(Math.abs(firstway - secondway) /
6.0414 * 1000);

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

    secondtrainSpeedValue.innerHTML =
Math.round(secondSpeed * 10000);

    secondtrainDistanceValue.innerHTML = 1000 -
(500 + Math.round(Math.abs(firstway -
secondway) / 6.0414 * 1000));

    if (Math.round(secondSpeed * 10000) >= 14) {
        secondSpeed -= 1;
        brakessecondtrainValue.innerHTML = "1";
        console.log("second 1")
    }
    if (Math.round(firstSpeed * 10000) >= 14) {
        firstSpeed -= 1;
        brakefirsttrainValue.innerHTML = "1";
        console.log("first 1")
    }

    if (1000 - (500 + Math.round(Math.abs(firstway
- secondway) / 6.0414 * 1000)) <= 50) {
        firstSpeed -= 9;
        brakefirsttrainValue.innerHTML = "9";
    }
    if (500 + Math.round(Math.abs(firstway -
secondway) / 6.0414 * 1000) <= 50) {
        secondSpeed -= 9;
        brakessecondtrainValue.innerHTML = "9";
    }

    if (firstSpeed < 0) firstSpeed = 0;
    if (secondSpeed < 0) secondSpeed = 0;

    if (Math.abs(firstway - secondway) > 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 firsttrainSpeed">first speed: <span
class="firsttrainSpeedValue"></span></div>

    <div class="speed secondtrainSpeed">second speed: <span
class="secondtrainSpeedValue"></span></div>


    <div class="distance firsttrainDistance">first to second distance: <span
class="firsttrainDistanceValue"></span></div>

    <div class="distance secondtrainDistance">second to first distance:
<span class="secondtrainDistanceValue"></span></div>


    <div class="brake brakefirsttrain">first train brake: <span
class="brakefirsttrainValue"></span></div>

    <div class="brake brakessecondtrain">second train brake: <span
class="brakessecondtrainValue"></span></div>

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

</body>

</html>

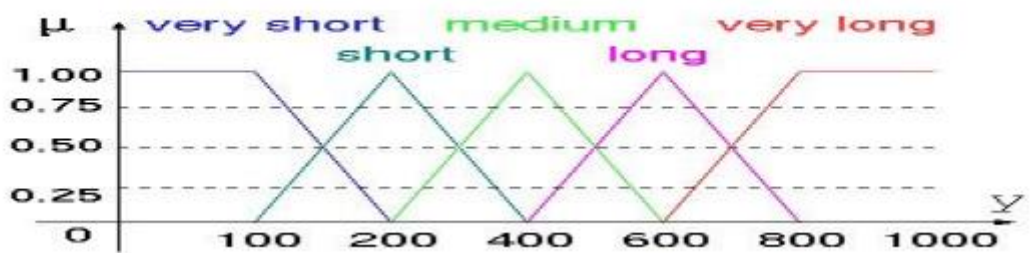
```

Results:

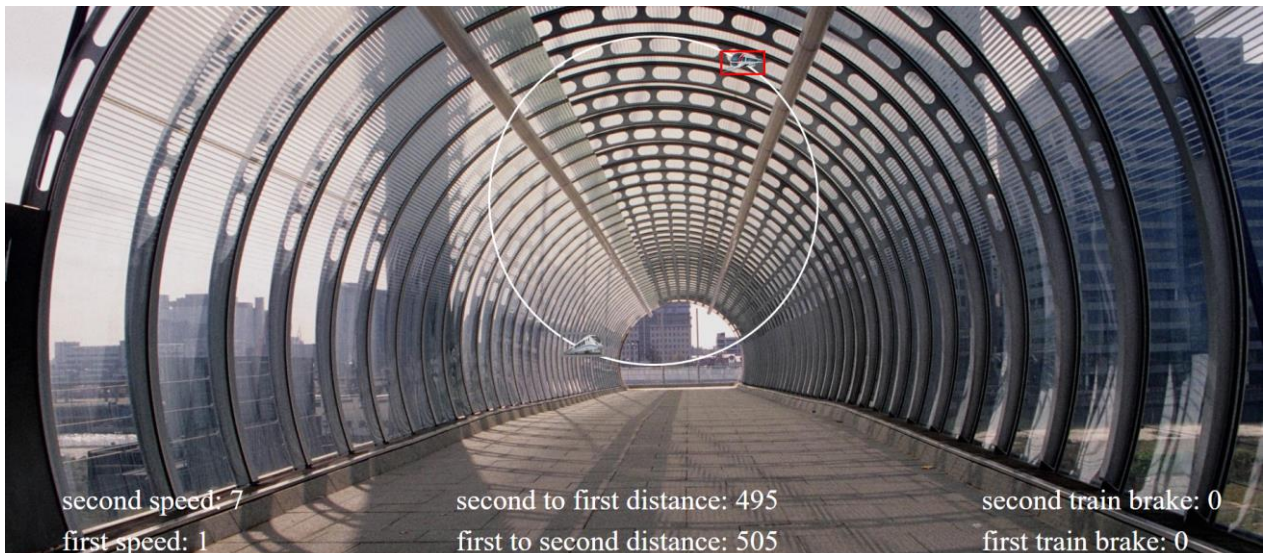
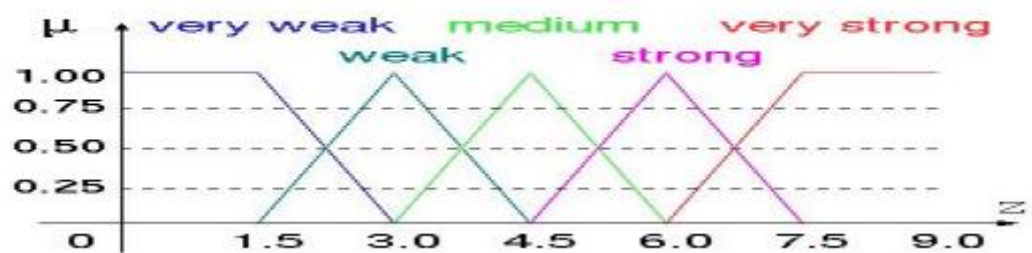
speed



distance y

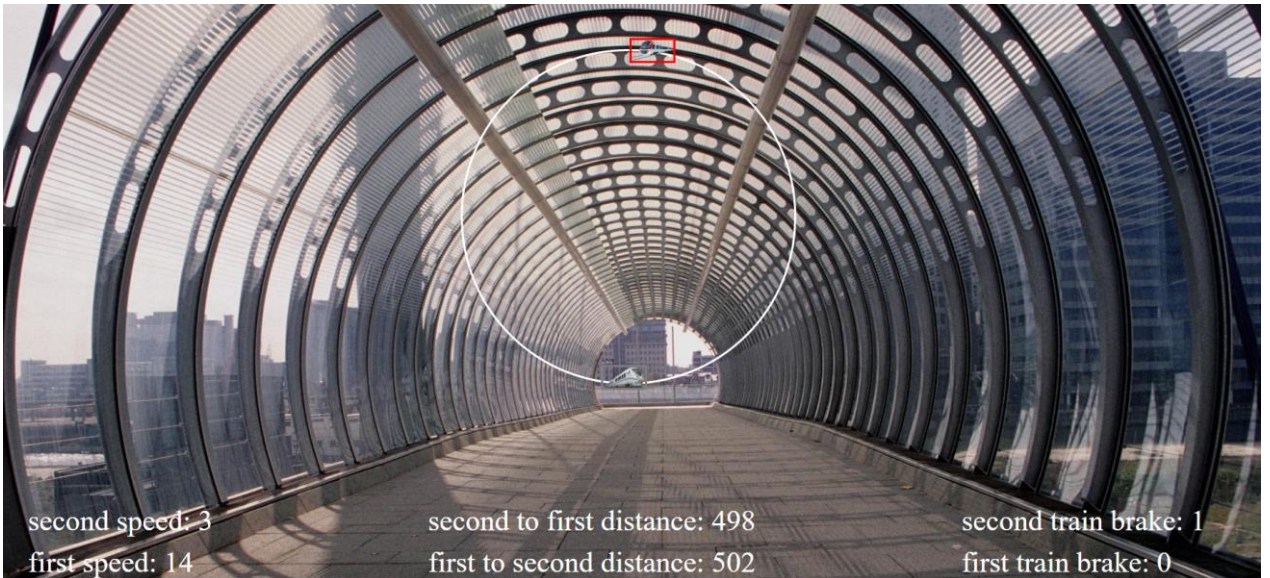


brake



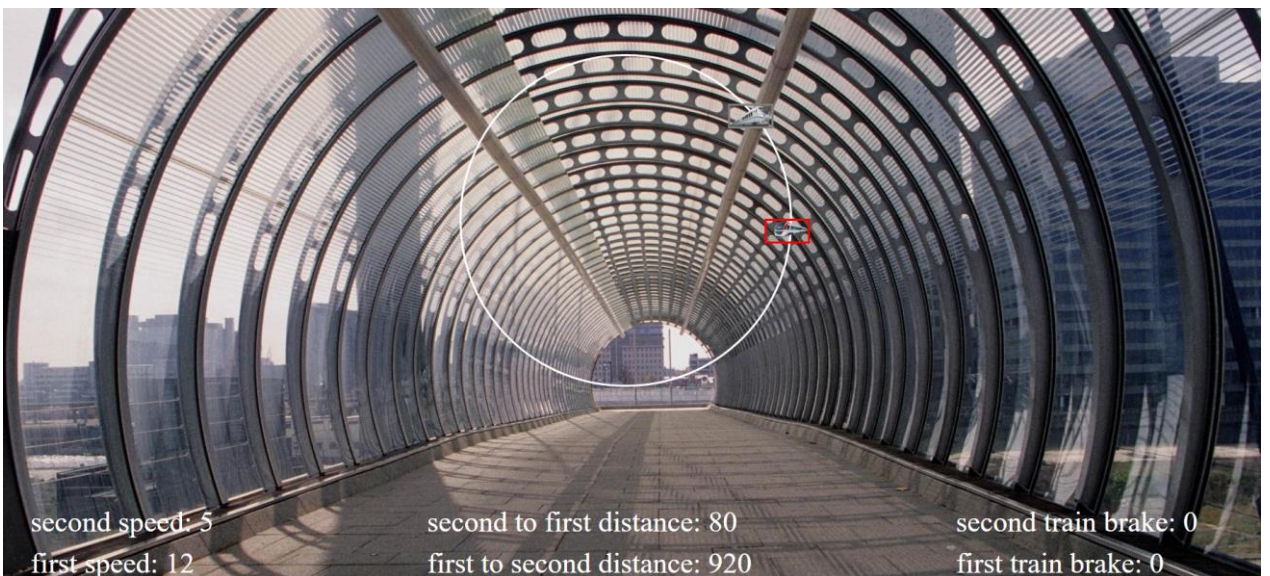
$\text{Mu_Speed} = 7 \Rightarrow \mu_s = 0.33;$
 $\text{Mu_Dist} = 495 \Rightarrow \mu_d = 0.5;$
 $\text{Mu_Brake} = 0 \Rightarrow \mu_b = 0;$
 $M = \min(\mu_s, \mu_d) * \mu_b \Rightarrow M = 0 * 0 = 0$

$\text{Mu_Speed} = 1 \Rightarrow \mu_s = 0;$
 $\text{Mu_Dist} = 505 \Rightarrow \mu_d = 0.5;$
 $\text{Mu_Brake} = 0 \Rightarrow \mu_b = 0;$
 $M = \min(\mu_s, \mu_d) * \mu_b \Rightarrow M = 0 * 0 = 0$



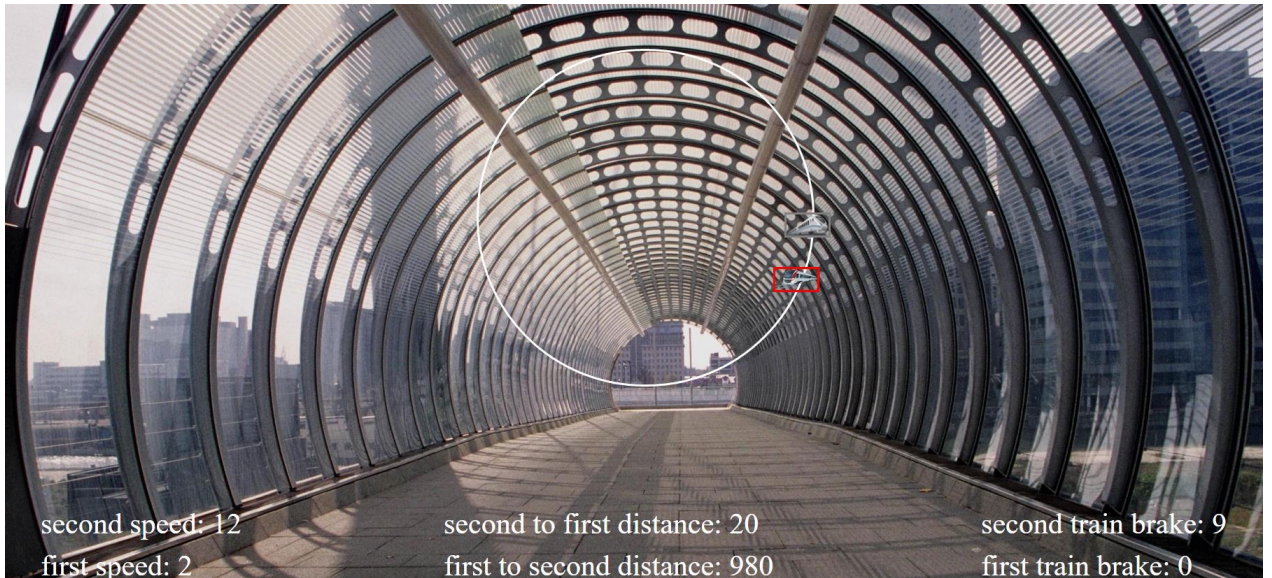
Mu_Speed = 3=>mu_s=0;
Mu_Dist = 498=>mu_d=0.5;
Mu_Brake = 1=>mu_b=0;
M=min(mu_s, mu_d)*mu_b => M=0*0=0

Mu_Speed = 14=>mu_s=0;
Mu_Dist = 502=>mu_d=0.5;
Mu_Brake = 0=>mu_b=0;
M=min(mu_s, mu_d)*mu_b => M=0*0=0



Mu_Speed = 5=>mu_s=0;
Mu_Dist = 80=>mu_d=0;
Mu_Brake = 0=>mu_b=0;
M=min(mu_s, mu_d)*mu_b => M=0*0=0

Mu_Speed = 12=>mu_s=0.33;
Mu_Dist = 920=>mu_d=0;
Mu_Brake = 0=>mu_b=0;
M=min(mu_s, mu_d)*mu_b => M=0*0=0



Mu_Speed = 12=>mu_s=0.5;

Mu_Dist = 20=>mu_d=0;

Mu_Brake = 9=>mu_b=0;

M=min(mu_s, mu_d)*mu_b => M=0*0=0

Mu_Speed = 2=>mu_s=0;

Mu_Dist = 980=>mu_d=0;

Mu_Brake = 0=>mu_b=0;

M=min(mu_s, mu_d)*mu_b => M=0*0=0

Z will always be equal 0 so Mu will always be 0.

It happened because our break don't belong to our interval.