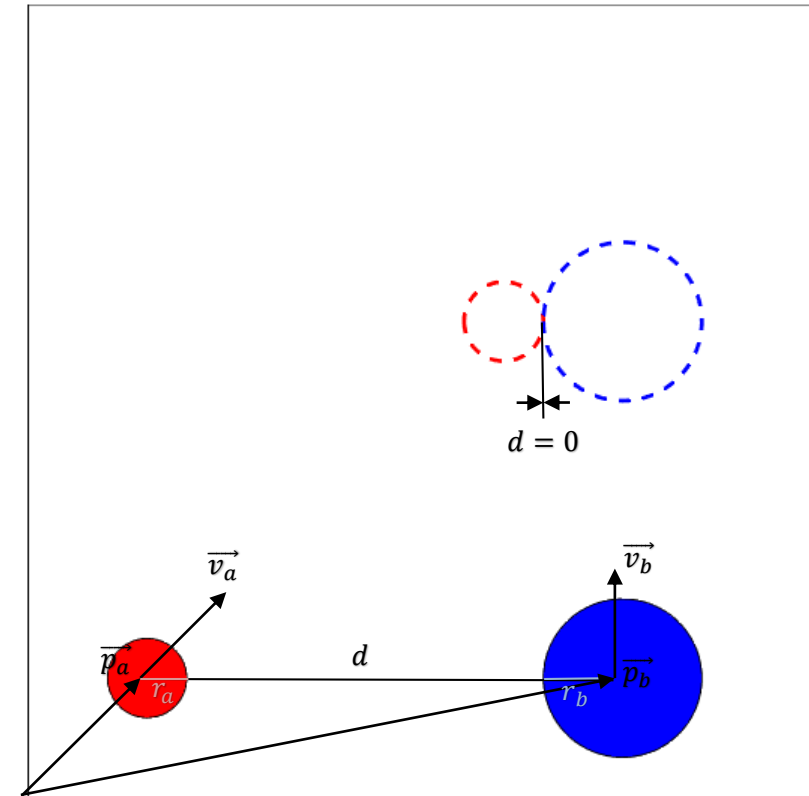


4. Rešavanje nelinearnih jednačina

Predviđanje sudara



Slika 1. Predviđanje sudara

Poznavajući fizičke karakteristike objekata i početne uslove kretanja, potrebno je odrediti trenutak sudara 2 objekta (slika 1). Radi jednostavnosti primera odabrani su krugovi i konstantne brzine kretanja.

Funkcija položaja takvog kretanja je:

$$\vec{p}(t) = \vec{p}_0 + \vec{v}t \quad (1)$$

, gde su \vec{p} trenutni položaj tela, \vec{p}_0 početni položaj tela, \vec{v} konstantna brzina tela, a t proteklo vreme.

Funkcija rastojanja između 2 kruga je:

$$d(t) = \sqrt{(\vec{p}_b(t) - \vec{p}_a(t))^2} - (r_a + r_b)$$

, gde su d trenutno rastojanje između 2 kruga, \vec{p}_a položaj kruga A, r_a poluprečnik kruga A, \vec{p}_b položaj kruga B, r_b poluprečnik kruga B, a t proteklo vreme.

Cilj je naći t , kada se krugovi dodiruju, tj. kada je rastojanje $d = 0$:

$$\sqrt{(\vec{p}_b(t) - \vec{p}_a(t))^2} - (r_a + r_b) = 0 \quad (2)$$

Zamenom jednačine (1) u (2) dobija se:

$$\begin{aligned}
 at^2 + bt + c &= 0 & (3) \\
 a &= \|\vec{v}_{ab}\|^2 \\
 b &= 2(\vec{p}_{ab} \cdot \vec{v}_{ab}) \\
 c &= \|\vec{p}_{ab}\|^2 - r_{ab}^2 \\
 \vec{v}_{ab} &= \vec{v}_b - \vec{v}_a \\
 \vec{p}_{ab} &= \vec{p}_{0b} - \vec{p}_{0a} \\
 r_{ab} &= r_a + r_b
 \end{aligned}$$

, gde su \vec{p}_{0a} početni položaj kruga A, \vec{v}_a brzina kruga A, r_a poluprečnik kruga a, \vec{p}_{0b} početni položaj kruga B, \vec{v}_b brzina kruga B, r_b poluprečnik kruga B, a $\vec{p}_{ab} \cdot \vec{v}_{ab}$ je skalarni proizvod vektora.

Traženjem nule funkcije $at^2 + bt + c = 0$ (pri čemu su a , b i c unapred izračunate konstante) se dobija trenutak sudara t dva kruga.

Zadatak 1

Naći trenutak sudara između 2 kruga:

krug	A	B
r	$0.05m$	$0.1m$
\vec{v}	$(x, y) = (0.1, 0.1) \frac{m}{s}$	$(x, y) = (0.0, 0.1) \frac{m}{s}$
\vec{p}_0	$(x, y) = (0.0, 0.0)m$	$(x, y) = (0.75, 0.0)m$

a) Definisati fizičke konstante:

```

worldSize = [1.0 1.0]; % [m]; dimenzije prostora

% krugovi
circleCount = 2;
r = [
    0.05
    0.10]; %[m]; dimenzije krugova
v = [
    0.1 0.1
    0.0 0.1]; %[m/s]; brzine krugova
p0 = [
    0.0 0.0
    0.75 0.0]; %[m]; početni položaji krugova
colors = [
    1.0 0.0 0.0
    0.0 0.0 1.0]; % (R,G,B); boje krugova

fMotion = @(p0, v, t) p0 + v*t; % funkcija kretanja

```

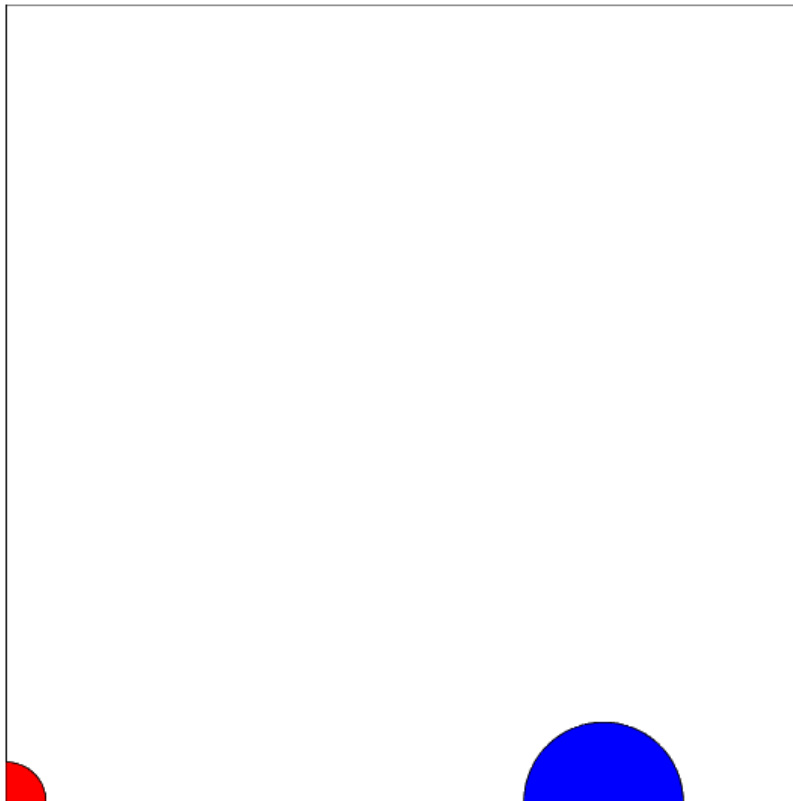
b) Inicijalizovati grafički interfejs:

```
% GUI
fig = figure('Name', 'Collision', 'Units', 'normalized', 'Position', [0.2 0.2 0.6 0.6]); % prozor
axis([0 worldSize(1) 0 worldSize(2)]) % ograničavanje prikaza u okviru dimenzija prostora
axis square % sprečavanje reskaliranja prikaza
axis off % sakrivanje osa

% inicijalizacija grafičkih objekata
graphics = gobjects(4 + circleCount);
% ivice
graphics(1) = line([0 worldSize(1)], [0 0], 'color', 'black');
graphics(2) = line([0 worldSize(1)], [worldSize(2) worldSize(2)], 'color', 'black');
graphics(3) = line([0 0], [0 worldSize(2)], 'color', 'black');
graphics(4) = line([worldSize(1) worldSize(1)], [0 worldSize(2)], 'color', 'black');
% krugovi
for circle = 1:circleCount % za svaki krug (circle)
    location = p0(circle, :);
    radius = r(circle);
    diameter = 2*radius;
    x = location(1) - radius;
    y = location(2) - radius;

    position = [x y diameter diameter];
    color = colors(circle, :);
    % grafički objekti od 1 do 4 su ivice
    graphics(4 + circle) = rectangle('Position', position, 'Curvature', [1 1], 'EdgeColor', 'black',
'FaceColor', color);
end
```

Rezultat:



Slika 2. Početak simulacije

c) Simulirati kretanje krugova po funkciji kretanja (jednačini (1) iz uvoda):

```
fps = 60; % broj osvežavanja prikaza u sekundi
timeScale = 2.0; % brzina simulacije

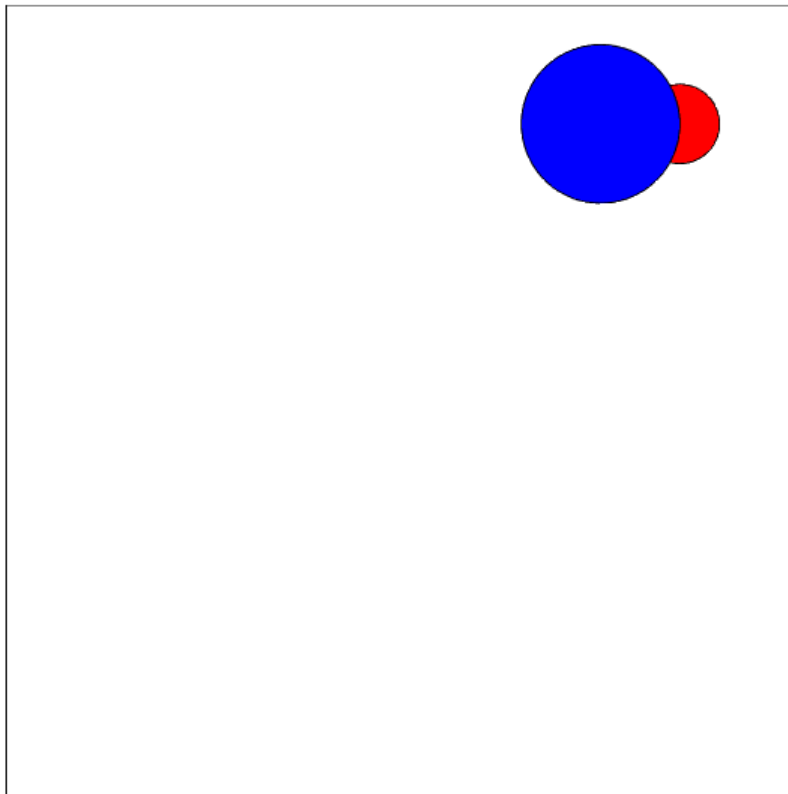
t = 0; % [s]; početni vremenski trenutak
dt = 1/fps; % [s]; vremenska razlika između koraka
while ishandle(fig) % dokle god je prozor otvoren
    t = t + dt*timeScale; % naredni vremenski trenutak

    % ažuriranje položaja i iscrtavanje
    for circle = 1:circleCount
        location = fMotion(p0(circle, :), v(circle, :), t); % trenutni položaj
        radius = r(circle);
        diameter = 2*radius;
        x = location(1) - radius;
        y = location(2) - radius;

        position = [x y diameter diameter];
        set(graphics(4 + circle), 'Position', position); % ažuriranje položaja grafičkih objekata
    end

    pause(dt);
end
```

Rezultat:



Slika 3. Simulacija

d) Pre procedure iz koraka c), pročitati vrednosti \vec{p}_{0a} , \vec{v}_a , r_a , \vec{p}_{0b} , \vec{v}_b i r_b :

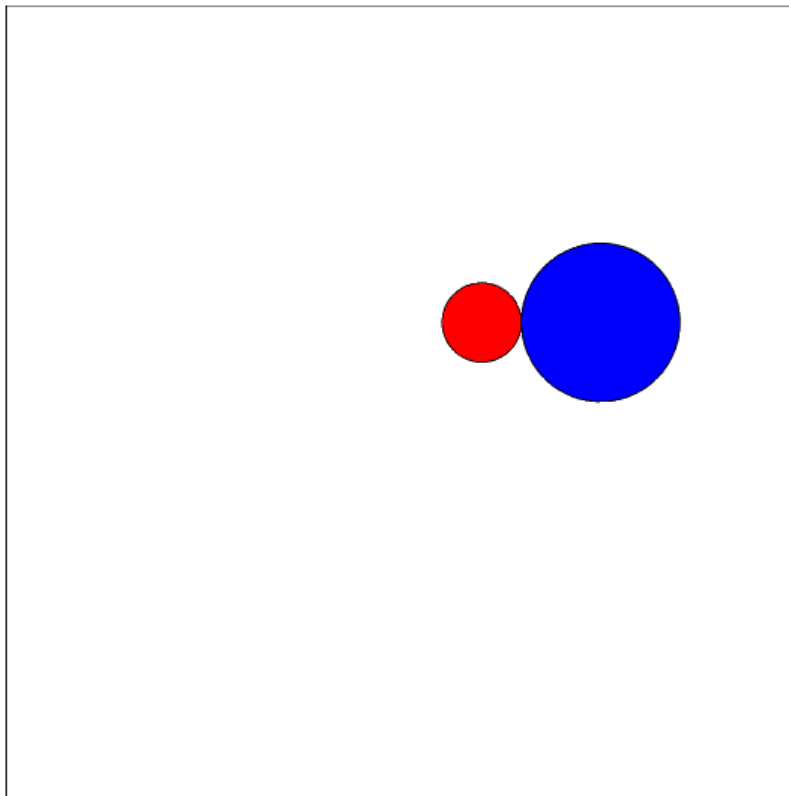
```
rA = r(1); % poluprečnik 1. kruga
vA = v(1, :); % brzina 1. kruga
p0A = p0(1, :); % početni položaj 1. kruga

rB = r(2); % poluprečnik 2. kruga
vB = v(2, :); % brzina 2. kruga
p0B = p0(2, :); % početni položaj 2. kruga
```

- e) Pre procedure iz koraka c), naći nulu funkcije rastojanja (definisane jednačinom (3) iz uvoda) i rezultat sačuvati u promenljivu `tCollision`. Ona predstavlja trenutak sudara. Za računanje skalarnog proizvoda dva vektora A i B koristiti MATLAB funkciju `dot(A,B)`.
- f) Sprečiti da se simulacija odvija nakon izračunatog trenutka sudara:

```
.  
.   
.   
  
while ishandle(fig) % dokle god je prozor otvoren  
    t = t + dt*timeScale; % naredni vremenski trenutak  
  
    if t >= tCollision % ograničiti vreme na trenutak sudara  
        t = tCollision;  
    end  
  
    .  
    .  
    .
```

Rezultat:



Slika 4. Sudar