

Završni rad

Modeliranje sustava koji
iskače iz vode pod utjecajem
uzgona

Sadržaj

- Uvod
- Problem projekta
- Matematički model
- Rad u ROS-u

Uvod

- Udio vodene površine na Zemlji iznosi oko 72%. Većina tog područja još uvijek je neistražena
- Razvitak upravljanja bespilotnim ronilicama kao i povećanje preciznosti navigacije uvelike doprinosi rješavanju spomenutog problema



Slika 1. Pogled Zemlje s Apolla 17

Uvod -Ronjenje



Problem projekta

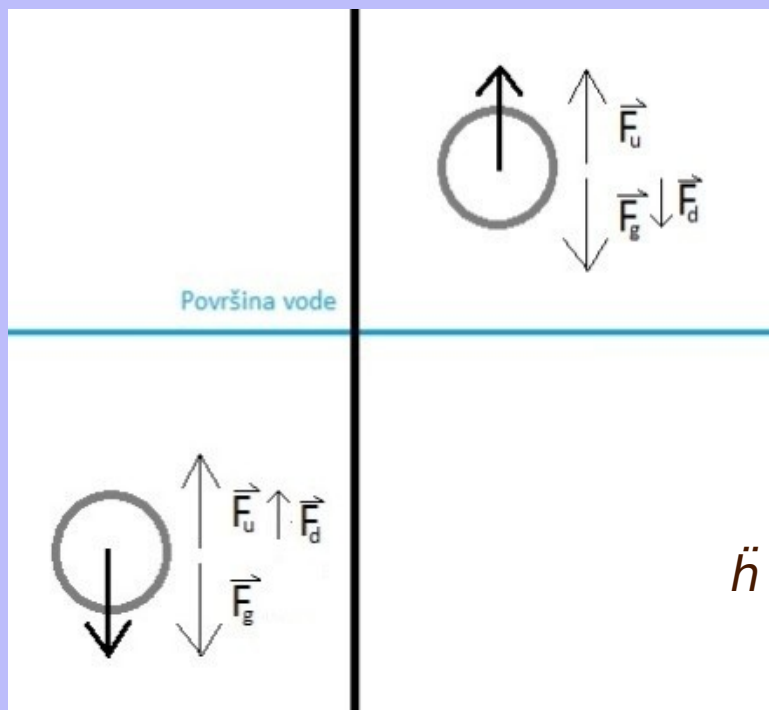
Zadatak završonog rada temelji se na:

1. Definiranju dinamičkog modela kuglice pri kretanju kroz vodu i zrak
2. Modelirati kuglicu u ROS-u
3. Simulirati rad više kuglica

Matematički model

Ideja za sondu:

- izraditi dinamički model kuglice koja „sama” uranja na određenu dubinu te potom iskače iz vode



Sile koje djeluju na kuglicu:

$$F_u = \rho_f * V_u * g$$

$$F_g = m * g$$

$$F_d = 0.5 * \rho_f * v^2 * Cd * A$$

$$F = F_u - F_g - F_d$$

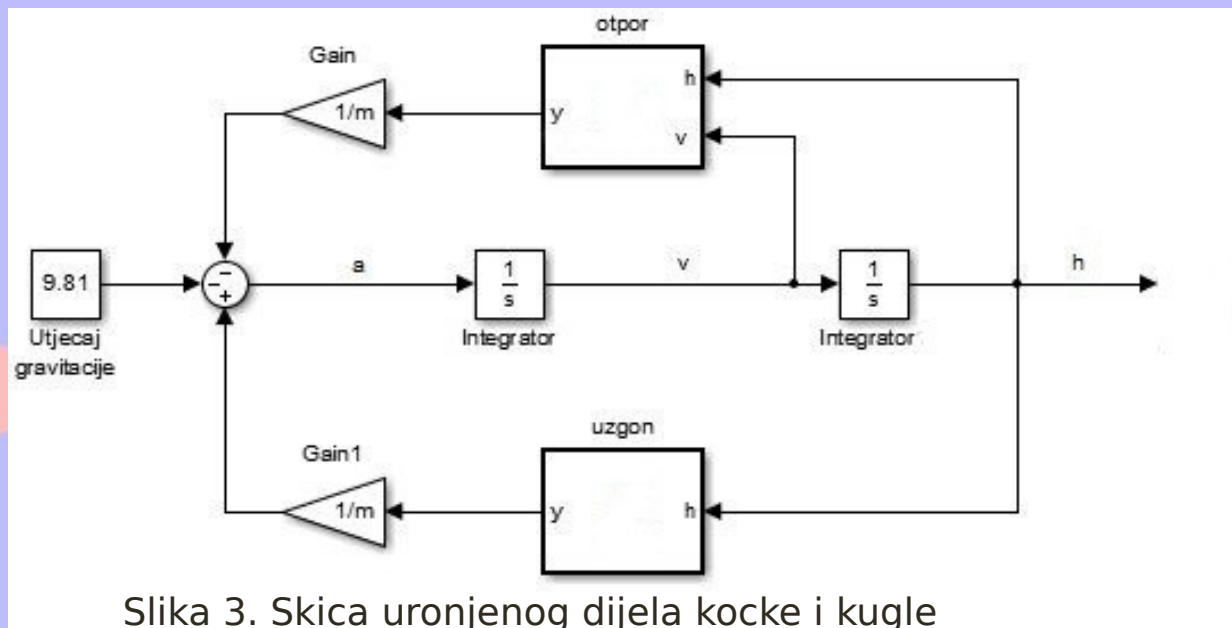
Konačna diferencijalna jednačba:

$$\ddot{h} = g + \frac{1}{m} (0.5 * \rho_f * Cd * A * \dot{h}^2 - \rho_f * V_u * g)$$

Slika 2. Sile pri uranjanju i izranjanju modela kuglice (sonde)

Matematički model

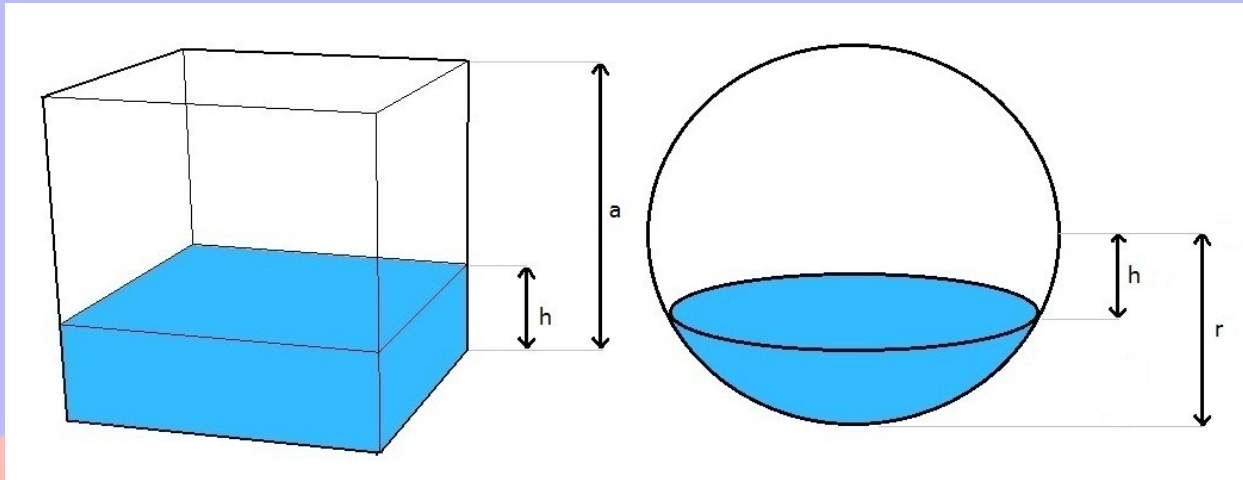
$$\ddot{h} = g + \frac{1}{m}(0.5 * \rho_f * Cd * A * \dot{h}^2 - \rho_f * V_u * g)$$



Slika 3. Skica uronjenog dijela kocke i kugle

Matematički model

- Problem volumena uronjenog tijela tijela

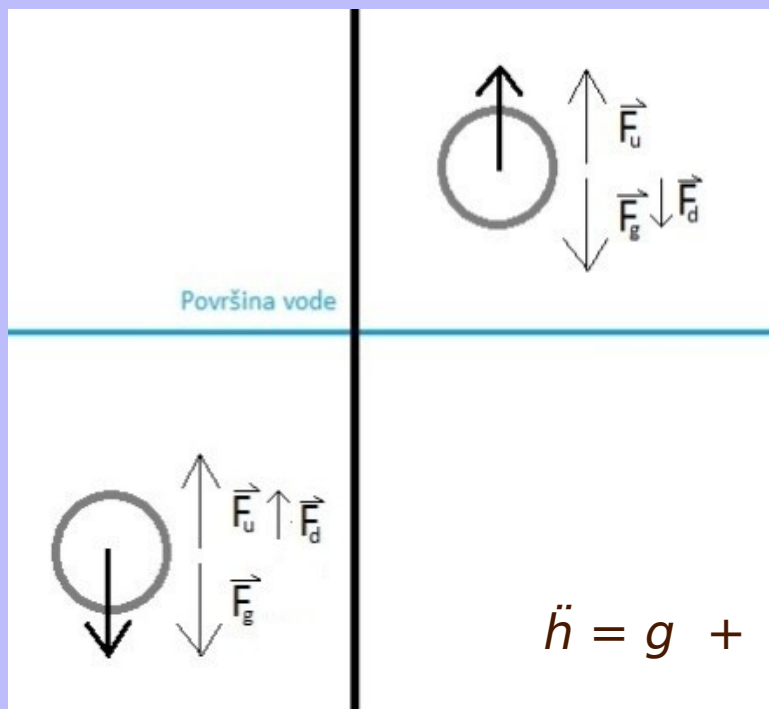


Slika 4. Skica uronjenog dijela kocke i kugle

Matematički model

Ideja za sondu:

- izraditi dinamički model kuglice koja „sama” uranja na određenu dubinu te potom iskače iz vode



Sile koje djeluju na kuglicu:

$$F_u = \rho_f * V_u * g$$

$$F_g = m * g$$

$$F_d = 0.5 * \rho_f * v^2 * Cd * A$$

$$F = F_u - F_g - F_d$$

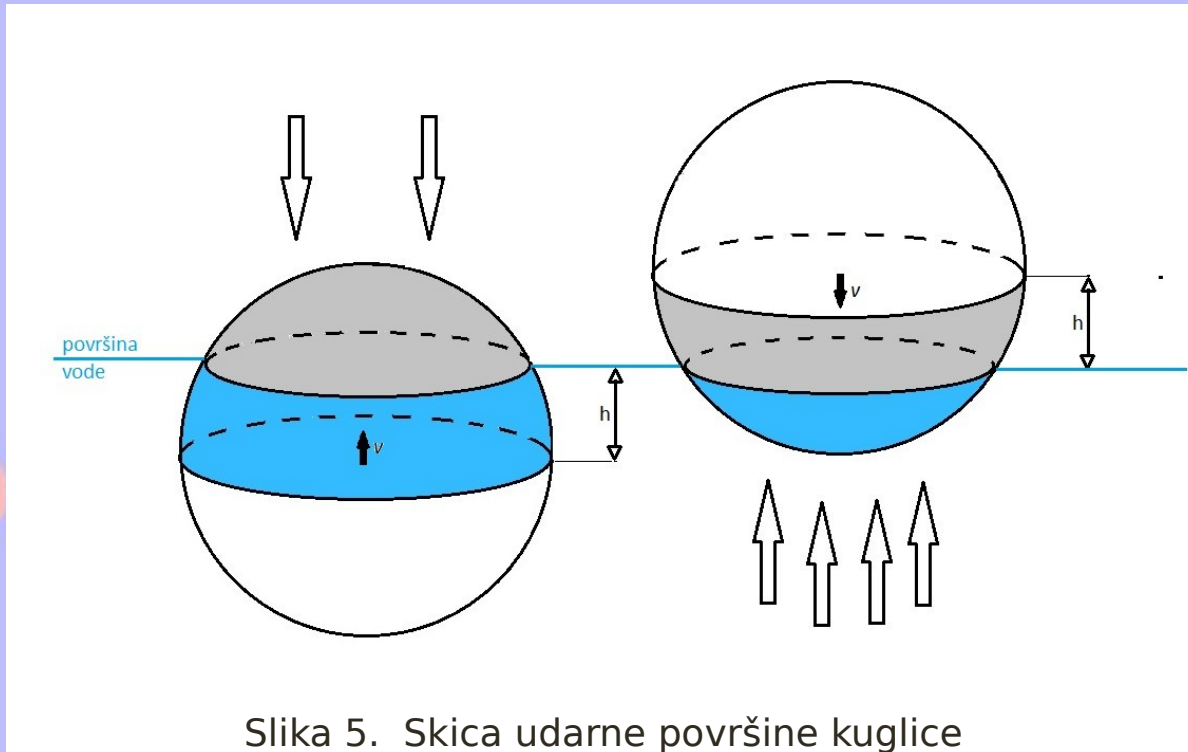
Konačna diferencijalna jednačba:

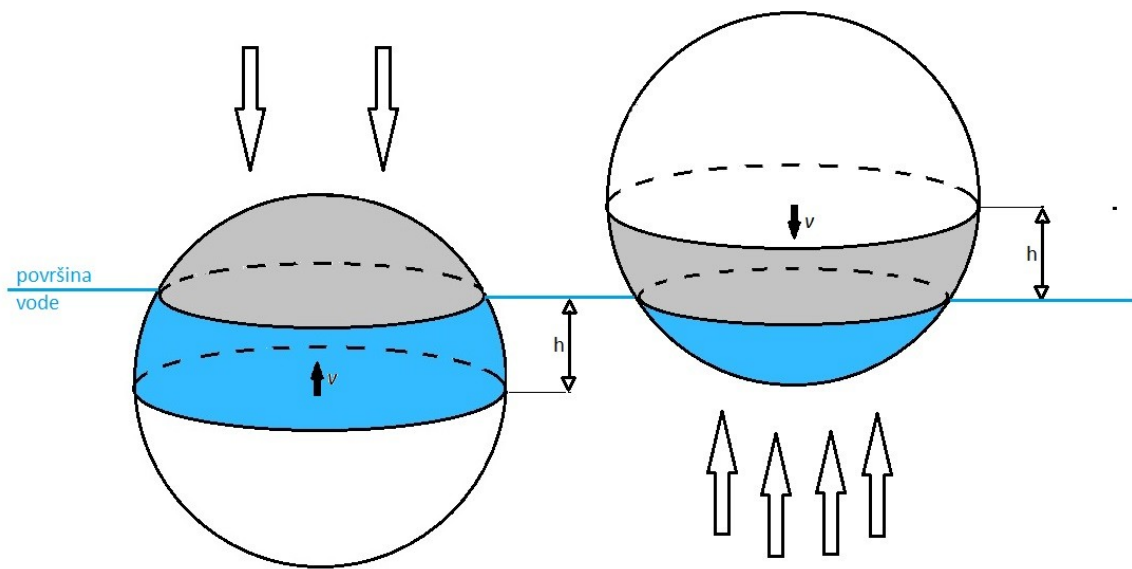
$$\ddot{h} = g + \frac{1}{m} (0.5 * \rho_f * Cd * A * \dot{h}^2 - \rho_f * V_u * g)$$

Slika 2. Sile pri uranjanju i izranjanju modela kuglice (sonde)

Matematički model

- Problem promjene udarnog oplošja tijela





Slika 5. Skica udarne površine kuglice

$v > 0$	$h < -r$	$F_d = 0.5 * \rho_v * v^2 * Cd * A$
	$-r < h < 0$	$F_d = 0.5 * v^2 * Cd * (A * \rho_z + (A - A_k) * \rho_v)$
	$h > 0$	$F_d = 0.5 * \rho_z * v^2 * Cd * A$
$v < 0$	$h < 0$	$F_d = 0.5 * \rho_v * v^2 * Cd * A$
	$0 < h < r$	$F_d = 0.5 * v^2 * Cd * (A * \rho_v + (A - A_k) * \rho_z)$
	$h > r$	$F_d = 0.5 * \rho_z * v^2 * Cd * A$

Tablica 1. Skica udarne površine kuglice

Rad u ROS-u

- Dva noda

Prvi node: Simulator

- Sadrži funkcije za računanje sile uzgona i sile otpora fluida
- Pomoću njih računa poziciju svih kuglica u svakom trenutku
- Objavljuje izračunate dubine na topicu *pozicija*

Drugi node: Display

- Koristi pyGame biblioteku za crtanje
- Dobiva poruke sa trenutnim dubinama
- Preračunava dubine u pozicije na ekranu
- Crta trenutne pozicije



Slika 6. Nodovi i topic

Rad u ROS-u

Oblik poruke za komunikaciju preko topica

- Nepoznat broj kuglica prije simulacije
- String različitih duljina ovisno o predznaku kuglice
- Treba koristiti podatkovno polje Float32MultiArray
- Praznoj varijabli se slijedno dodaju pojedine dubine
- Tako sastavljen niz se objavljuje
- Ciklički se ponavlja s periodom diskretizacije

data: [-6.9892859, -9.5933771, -4.7802720, 0.6517068, -1.51679670]

Slika 7. Primjer poruke Float32MultiArray tipa kao komunikacija između nodova

Rad u ROS-u

Parametri:

- Parametri simulacije postavljaju se korištenjem rospy funkcija `set_param` i `get_param`
- Primjer postavljanja vrijednosti parametra:

```
<rosparam param="zad_dub" subst_value="True">$(arg dubina)</rosparam>
```
- Primjer preuzimanja vrijednosti parametra:

```
masa = rospy.get_param('masa', 0.5)
```
- Korišteni ros parametri su:
 - Masa kuglica
 - Radijus kuglica
 - Broj kuglica
 - Radna dubina

Rad u ROS-u

Pokretanje:

- Mogućnost mijenjanja parametra simulacije
- Očitani parametri pridjeljuje se nodovima kao ros parametri
- Pokreće se svaki nod
- Pokretanje se radi iz linux terminala naredbom roslaunch
- Pri pokretanju može se postaviti vrijednost parametara:
masa kuglica, radius kuglica, broj kuglica i radna dubina

```
donat@bicbozji:~$ roslaunch zavrshi pokret.launch dubina:=-20 masa:=0.6 radius:=0.3
```

Slika 8. Primjer pokretanja simulacije

Rad u ROS-u

Primjena:

- Moguć nastavak rada na projektu – naprednija simulacija, nodovi za upravljanje kuglicama
- Ugradnja kamere i bočnih krilaca za pozicioniranje u x-y ravnini
- Praćenje podvodnih kabela i slikanje
- Komunikacija za daljinsko upravljanje ronilice
- Analiza zagađenja mora I trodimenzionalno mapiranje zagađenja

Hvala na pažnji