

MODELIRANJE I SIMULIRANJE VIRTUALNOG 3D MODELA NA PRIMJERU ELEKTROMOBILA S TRI KOTAČA U OPEN DYNAMICS ENGINE-U

MODELING AND SIMULATION OF VIRTUAL 3D MODEL ON THE EXAMPLE OF A THREE-WHEEL ELECTRIC VEHICLE DESIGNED IN OPEN DYNAMICS ENGINE

Vrhovski Z.¹, Purković D.¹, Jurković I.¹

¹Visoka tehnička škola u Bjelovaru, Bjelovar, Hrvatska

Sažetak: U ovom radu opisan je virtualni model elektromobila s tri kotača izrađenog u Open Dynamics Engine-u. Model vozila sadrži mnoga fizikalna svojstva stvarnog elektromobila te se s toga ovaj model može primijeniti za analizu i sintezu sustava upravljanja elektromobilom. Virtualno okruženje vozila modelirano je sa različitim podlogama (suhi asfalt, trava, led) na kojima se može pokazati različito vladanje elektromobila u raznim uvjetima. Elektromobil je opremljen s tri istosmjerna motora koji pokreću pojedine kotače. Motori su upravljani PI regulatorom.

Ključne riječi: Open Dynamics Engine, elektromobil, istosmjerni motor, PI regulator

Abstract: This paper describes a virtual model of a three-wheel electric vehicle designed in Open Dynamics Engine. The vehicle model includes many physical properties of a real electric vehicle. Thus, this model may be applied in analyzing and synthesizing the electric vehicle control system. The vehicle's virtual environment was modeled with various surfaces (dry asphalt, grass, ice), on which it may be shown how the electric vehicle behaves under various conditions. The electric vehicle is equipped with three DC motors that drive individual wheels. The motors are controlled by a PI controller.

Key words: Open Dynamics Engine, electric vehicle, DC motor, PI controller

1. UVOD

U današnje vrijeme razvoj i proizvodnja komercijalnih proizvoda mora se odvijati jako brzo. Treba izraditi prototip s obzirom na to da često ne možemo napraviti krajnji proizvod i testirati algoritme upravljanja na samom proizvodu. Prototipovi se mogu naći u raznim izvedbama. U ovom radu zanimljiv je virtualni fizikalni prototip elektromobila. Unutar virtualnog prototipa opisane su važne fizikalne pojave koje su često dovoljne da bi se razvio algoritam upravljanja za takav prototip. Za izradu virtualnog prototipa koristit će se *Open Dynamics Engine*. Virtuani modeli izrađuju se programiranjem u C++ programskom jeziku, a simulacija je izvedena korištenjem Matlab&SIMULINK- a.

2. MODELIRANJE VIRTUALNOG 3D MODELA ELEKTROMOBILA (TRICIKLA) U OPEN DYNAMICS ENGINE-U

Open Dynamics Engine (ODE) je besplatna biblioteka funkcija koja se koristi za modeliranje i simuliranje dinamike zglobnih krutih tijela. Tako je npr. ODE pogodan za simuliranje vozila, hodača te pokretnih objekata u *Virtual Reality* okruženju. Prednosti ODE-a su brzina, fleksibilnost, robusnost te ugrađene funkcije za detekciju sudara između pokretnog objekta i prostora u kojem se objekt kreće. Zbog brzine izvođenja ODE omogućuje simuliranje pokretnih objekata u stvarnom vremenu.

Kreiranje virtualnog modela elektromobila opisan je sljedećim koracima [1], [2]:

- 1) Stvaranje dinamičnog virtualnog svijeta
- 2) Stvaranje fizikalnog modela elektromobila u virtualnom svijetu
- 3) Postaviti položaj pojedinih dijelova tijela elektromobila u virtualnom svijetu
- 4) Stvaranje zglobova u virtualnom svijetu
- 5) Vezati zglobove s tijelom elektromobila
- 6) Postaviti parametre svih zglobova
- 7) Stvaranje virtualnog okruženja fizikalnog modela elektromobila
- 8) Stvaranje grupe zglobova elektromobila za koje se računaju točke kolizije
- 9) Simulacijska petlja
 - a) narinuti moment na kotače elektromobila
 - b) promijeniti/unijeti parametre zglobova ako je potrebno
 - c) pozvati funkciju detekcije sudara
 - d) proračunati točke kolizije pojedinih zglobova
 - e) pročitati vrijeme koraka simulacije i odrediti nova stanja elektromobila
 - f) uništiti trenutne točke kolizije
- 10) Uništiti stvorene objekte u virtualnom svijetu i sami svijet

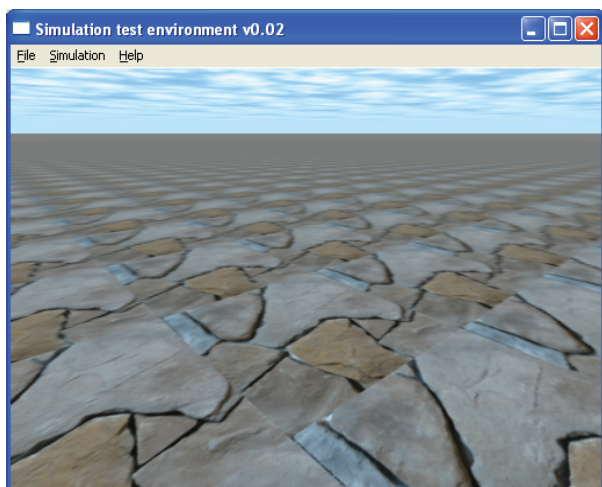
2.1. Stvaranje dinamičkog svijeta

Stvoreni virtualni svijet sadrži kruta tijela i zglobove. Objekti iz različitih virtualnih svjetova ne mogu stupiti u interakciju, odnosno ne kolidiraju. Najčešće aplikacije sadrže samo jedan svijet kao na primjeru svijeta elektromobila koji će biti opisan u nastavku.

Stvaranje svijeta odvija se sljedećim nizom naredbi:

```
static dWorldID world; // deklariranje
world-a
world = dWorldCreate();// stvaranje world-
a
dWorldSetGravity(world, 0,0,-9.81); //
postavljenje gravitacije Zemlje
dWorldSetERP(world,0.5); //error reduction
parametar
dWorldSetCFM(world,0.00001); //constraint
force mixing
ground =
dCreatePlane(space,0,0,1,0);//ravnina u
kojoj leži podloga (x,y)
```

Nakon korištenja prethodnog slijeda naredbi dobiven je virtualni svijet prikazan na slici 1. Gravitacijska konstanta postavljena je u negativnom smjeru z-osi i ima iznos Zemljine gravitacijske konstante. Time vrijede svi fizikalni zakoni za kruta tijela na Zemlji.



Slika 1. Virtualni svijet u ODE-u

2.2. Stvaranje fizikalnog modela elektromobila u virtualnom svijetu

Stvaranje fizikalnog modela u virtualnom svijetu odvija se nizom naredaba:

```
// tijelo automobila; donji dio
body[0] = dBodyCreate (world); //
stvaranje tijela
dQuaternion q;
dQFromAxisAndAngle (q,1,0,0,0); //
orijentacija tijela
dBodySetQuaternion (body[0],q);
dBodySetPosition (body[0],0,0,STARTZ); //
pozicija tijela
dMassSetBox (&m,1,LENGTH,WIDTH,HEIGHT); //
definiranje volumena tijela
```

```
dMassAdjust (&m,CMASS); // definiranje
mase tijela
dBodySetMass (body[0],&m); // postavljanje
mase tijela
```

```
box[0]=dCreateBox(0,LENGTH,WIDTH,HEIGHT);
// geometrijski oblik tijela
dGeomSetBody (box[0],body[0]);
```

```
// tijelo automobila; gornji dio
body[4] = dBodyCreate (world);
dBodySetPosition (body[4],0,0,STARTZ +
3*HEIGHT/4);
dMassSetBox
(&m,1,LENGTH/2,WIDTH/2,HEIGHT/2);
dMassAdjust (&m,CMASS/100);
dBodySetMass (body[4],&m);
ground_box[9] = dCreateBox
(0,LENGTH/2,WIDTH/2,HEIGHT/2);
dGeomSetBody (ground_box[9],body[4]);
// tijelo automobila; kotači
for (i=1; i<=3; i++) {
body[i] = dBodyCreate (world);
dQuaternion q;
dQFromAxisAndAngle (q,1,0,0,M_PI*0.5);
dBodySetQuaternion (body[i],q);
dMassSetSphere (&m,1,RADIUS);
dMassAdjust (&m,WMASS);
dBodySetMass (body[i],&m);
cylinder[i-1] = dCreateCylinder
(0,RADIUS,0.2);
dGeomSetBody (cylinder[i-1],body[i]);
}
```

Prethodnim nizom naredbi definiran je fizikalni model tijela elektromobila. Definirane su dimenzije elektromobila, polumjer kotača, masa pojedinih dijelova elektromobila te orijentacija pojedinih dijelova elektromobila.

2.3. Postavljanje pozicije pojedinih kotača elektromobila

Postavljanje pozicije pojedinih kotača u odnosu na tijelo elektromobila dano je naredbama:

```
dBodySetPosition
(body[1],0.5*LENGTH,0,STARTZ-HEIGHT*0.5);
```

```
dBodySetPosition (body[2],0.5*LENGTH,
WIDTH*0.5,STARTZ-HEIGHT*0.5);
```

```
dBodySetPosition(body[3],0.5*LENGTH,-
WIDTH*0.5,STARTZ-HEIGHT*0.5);
```

Spomenute naredbe postavljaju kotače s obzirom na dimenziju tijela elektromobila. Postavljanje pozicija svih dijelova tijela elektromobila obavlja se u globalnom koordinatnom sustavu virtualnog svijeta. Kasnije, tijekom simulacije, lokalni koordinatni sustav elektromobila giba se zajedno s elektromobilom te se uvijek ostvaruje fiksna pozicija kotača u odnosu na tijelo elektromobila. Pozicija je fiksna. Uvođenjem parametara ERP (*error reduction parametar*) i CFM (*constraint*)

force mixing) tijekom simulacije dolazi do promjene pozicije zglobova kotača u odnosu na tijelo elektromobila, čime je na jednostavan način modeliran ovjes elektromobila. Rezultat stvaranja fizikalnog modela elektromobila prikazan je na slici 2.



Slika 2. Elektromobil u ODE-u

2.4. Stvaranje zglobova elektromobila, povezivanje zglobova i podešavanje postavki zglobova

Stvaranje zglobova elektromobila te postavljanje parametara zglobova i povezivanje s tijelom elektromobila navedeno je sljedećim naredbama:

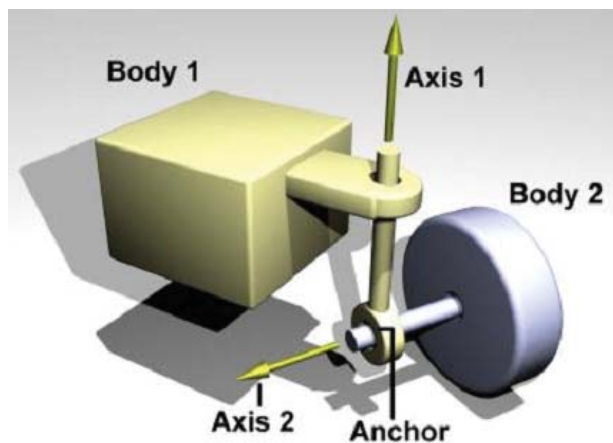
```
// kreiranje zglobova
for (i=0; i<3; i++) {
  joint[i] = dJointCreateHinge2 (world,0);
  dJointAttach (joint[i],body[0],body[i+1]);
  // kreiranje zgloba
  const dReal *a = dBodyGetPosition
  (body[i+1]);
  dJointSetHinge2Anchor
  (joint[i],a[0],a[1],a[2]); // postavljanje
  //pozicije zgloba u odnosu na tijelo
  elektromobila
  dJointSetHinge2Axis1 (joint[i],0,0,1); //
  os rotacije 1. zgloba
  dJointSetHinge2Axis2 (joint[i],0,1,0); //
  os rotacije 2. zgloba
}

// fiksni zglob; gornji dio automobila
joint[3] = dJointCreateFixed(world,0);
dJointAttach (joint[3],body[0],body[4]);
dJointSetFixed(joint[3]);

// postavljanje parametara zgloba
for (i=0; i<3; i++) {
  dJointSetHinge2Param
  (joint[i],dParamSuspensionERP,0.8);
  dJointSetHinge2Param
  (joint[i],dParamSuspensionCFM,0.001);
  dJointSetHinge2Param
  (joint[i],dParamLoStop,0);
```

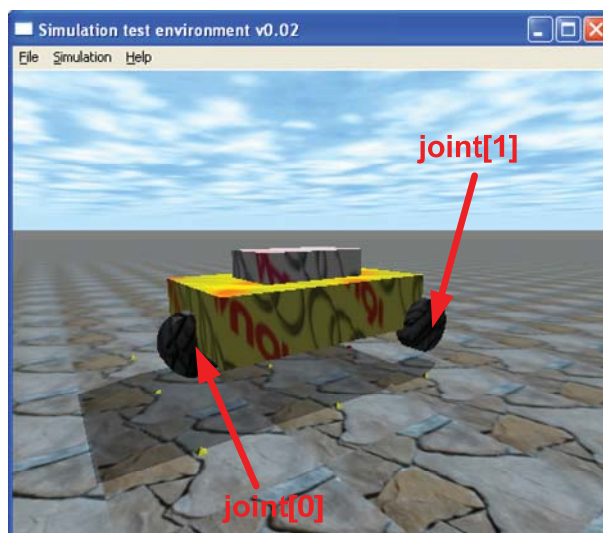
```
dJointSetHinge2Param
(joint[i],dParamHiStop,0);
}
```

Na slici 3. prikazan je zglob tipa *Hinge-2*. Prvi zglob omogućuje skretanje kotača, a drugi zglob okretanje kotača. S obzirom na to da se kut zakreta kotača mora ograničiti, treba podesiti parametre zgloba tako da se definiira najmanji i najveći kut zakreta. Na pojedini zglob moguće je narinuti kutnu brzinu zakreta u zglobu. Pošto se prvim zglobom želi postići određeni zakret, dovoljno je napraviti P regulator zakreta unutar ODE-a te se tako može modelirati skretanje volanom.

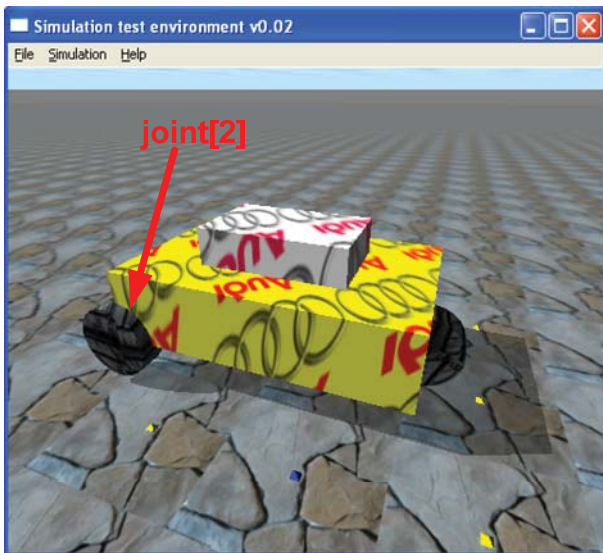


Slika 3. *Hinge-2* zglob

Zglobovi pojedinih kotača prikazani su na slikama 4. i 5. Na drugi zglob moguće je također narinuti brzinu vrtnje, ali s obzirom da su kotači pogonjeni elektromotorima, zanimljivo je modelirati drugi zglob tako da mu se u zglobu narine moment. Na taj način se moment proizveden istosmjernim motorom unutar simulacije može direktno priključiti na drugi zglob virtualnog modela.



Slika 4. Prikaz zglobova na virtualnom modelu elektromobila (1)



Slika 5. Prikaz zglobova na virtualnom modelu elektromobila (2)

2.5. Stvaranje virtualnog okruženja elektromobila

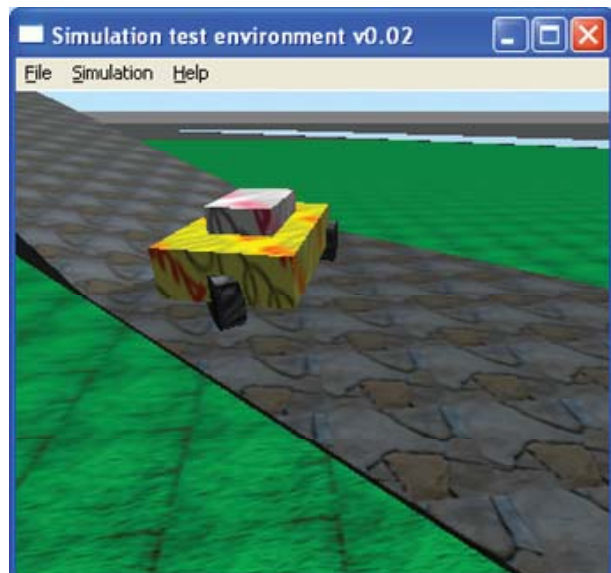
Da bi se moglo ispitati vladanje elektromobila na kosini, ledu i raznim drugim podlogama potrebno je stvoriti željeno virtualno okruženje. Na slikama 6., 7. i 8. prikazano je okruženje automobila koje može biti na cesti, travi, na ledenoj podlozi, na nizbrdici i uzbrdici. Pojedini područjima (podlogama) može se postaviti različiti koeficijent trenja čime je omogućeno modeliranje asfalta, leda i ostalih podloga.



Slika 6. Virtualno okruženje elektromobila (1)



Slika 7. Virtualno okruženje elektromobila (2)



Slika 8. Virtualno okruženje elektromobila (3)

2.6. Izvođenje simulacije

Simulacija u ODE-u izvodi se diskretno. U svrhu simuliranja modela elektromobila koristi se vrijeme diskretizacije od 0.01 s. Unutar jednog ciklusa izvođenja postavljaju se vrijednosti momenata u zglobovima i zakreti kotača, te se proračunava kolizija elektromobila s virtualnim okruženjem. Nakon toga se, koristeći vrijeme diskretizacije i informacije o koliziji, proračunava nova vrijednost brzine vozila u svim smjerovima, kutna brzina kotača, pozicije vozila i orijentacija vozila. Moment kotača zadaje se sljedećim nizom naredbi:

```
dJointAddHinge2Torques ( joint[0] , 0 , -
CarVariable.ReferentTorque[0] );
```

```
dJointAddHinge2Torques ( joint[1] , 0 , -
CarVariable.ReferentTorque[1] );
```

```
dJointAddHinge2Torques ( joint[2] , 0 , -
CarVariable.ReferentTorque[2] );
```

Momente na kotače stvaraju elektromotori koji će biti modelirani u Matlab&SIMULINK-u. Nova stanja vozila računaju se interno u ODE-u u kojem je ostvarena i kinematika i dinamika vozila.

3. MODELIRANJE I SIMULIRANJE SUSTAVA UPRAVLJANJA DC MOTORIMA ELEKTROMOBILA MODELIRANOG U OPEN DYNAMICS ENGINE-U

Virtualni model izveden u ODE-u potrebno je iskoristiti u svrhu simulacije i sinteze sustava upravljanja elektromobilom. Da bi to bilo izvedivo treba uspostaviti vezu između ODE-a i Matlab SIMULINKA (jedno od mogućih rješenja). Ova veza zapravo se svodi na komunikaciju između C++ modela i Matlab SIMULINKA [2]. Simulacijska shema prikazana je na slici 9. U svrhu simulacije, za pogon pojedinog kotača automobila koristi se istosmjerni motor s nezavisnom uzбудom. Diferencijalne jednačbe koje opisuju istosmjerni motor dane su relacijama, [2], [4], [6]:

$$U_a(t) = L_a \frac{di_a(t)}{dt} + i_a(t)R_a + k_m \omega_w(t) \quad (1)$$

$$\tau_e(t) = k_m i_a(t) \quad (2)$$

gdje su:

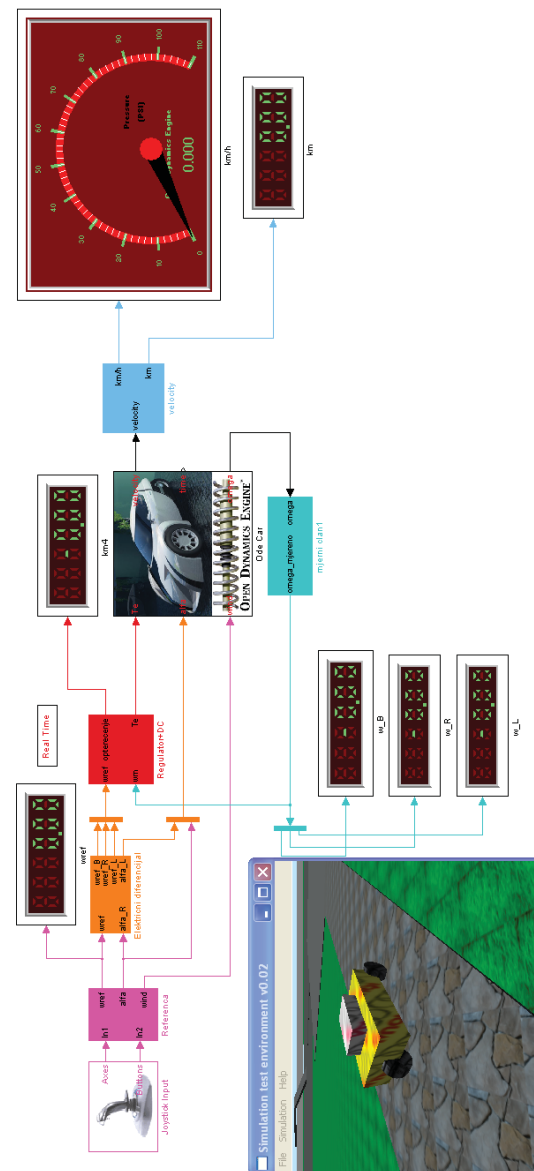
- U_a – napon armature istosmjernog motora, [V]
- I_a – struja armature istosmjernog motora, [A]
- L_a – induktivitet armature istosmjernog motora, [H]
- R_a – otpor armature istosmjernog motora, [Ω]
- k_m – koeficijent elektromagnetskog momenta, [Nm/A]
- ω_w – brzina vrtnje rotora istosmjernog motora (kotača automobila), [rad/s]
- τ_e – elektromagnetski moment, [Nm]

Za upravljanje brzinom vrtnje korišten je PI regulator [7]. Parametri PI regulatora određeni su prema simetričnom optimumu [3], [5]. Upravljački algoritam PI regulatora dan je relacijom:

$$U_a = K_r e(t) + \frac{K_r}{T_I} \int_0^t e(t) dt \quad (3)$$

gdje su:

- K_r – proporcionalno pojačanje PI regulatora, [Vs/rad]
- T_I – integralna vremenska konstanta PI regulatora, [s]
- e – pogreška brzine vrtnje, [rad/s]



Slika 9. Simulacijska shema sustava upravljanja virtualnog modela elektromobila

3.1. Simuliranje virtualnog modela elektromobila u Matlab SIMULINKU

U tablicama 1. i 2. nalazi se popis parametara korištenih u simulaciji.

Tablica 1. Parametri matematičkog modela automobila

$r_w = 0.2593$ m	$M_w = 651$ kg	$L_w = 2.05$ m	$d_w = 1.3$ m
------------------	----------------	----------------	---------------

gdje su :

- r_w – radijus kotača, [m]
- M_w – masa elektromobila, [kg]
- L_w – razmak između prednjih i stražnjih kotača, [m]
- d_w – razmak između prednjih kotača, [m]