

# Numeričko modeliranje kontaktne dinamike sustava velikog broja krutih čestica na primjeru rovera za Mars

---

Pipunić, Fran

Undergraduate thesis / Završni rad

2023

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:072667>

*Rights / Prava:* [Attribution-ShareAlike 4.0 International / Imenovanje-Dijeli pod istim uvjetima 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2025-02-03**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



Sveučilište u Zagrebu  
Fakultet strojarstva i brodogradnje

# **ZAVRŠNI RAD**

**Fran Pipunić**

Zagreb 2023.

Sveučilište u Zagrebu  
Fakultet strojarstva i brodogradnje

# ZAVRŠNI RAD

*Mentor:*

Doc. Dr. sc. Dario Zlatar

*Autor:*

Fran Pipunić

Zagreb 2023.



Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 23 - 6 / 1	
Ur.broj: 15 - 1703 - 23 -	

## ZAVRŠNI ZADATAK

Student: **Fran Pipunić**

JMBAG: **0035225375**

Naslov rada na hrvatskom jeziku: **Numeričko modeliranje kontaktne dinamike sustava velikog broja krutih čestica na primjeru rovera za Mars**

Naslov rada na engleskom jeziku: **Numerical modeling of rigid-particles-system contact dynamics by using Mars rover test case**

Opis zadatka:

Numeričko modeliranje sustava koji sadrže velik broj krutih čestica u međusobnom kontaktu zahtjevna je zadaća i kao takva predmet je velikog interesa istraživača posljednjih godina. Reprezentativan primjer takvog sustava je sustav suspenzije koji omogućuje kretanje po različitim konfiguracijama terena što je apsolutno neophodan atribut svakog robota namijenjenog za istraživanje Marsa. Princip rada takvog sustava suspenzije zasniva se na održavanju jednakog tereta na sve kotače, iako je moguće da se neki od kotača nalazi na višoj ili nižoj razini od ostalih.

U tom smislu, u radu je potrebno:

- Opisati princip rada sustava suspenzije Marsova rovera te objasniti kako je taj sustav instaliran na rover.
- Opisati model kontaktne dinamike koji će se koristiti u numeričkoj simulaciji.
- Izraditi računalni model rovera koji uključuje opisani kontaktni model u programskom paketu Project Chrono ili drugom prikladnom programskom okruženju otvorenog koda.
- Usporediti simulaciju i eksperimentalne rezultate dobivene mjerenjem na stvarnom roveru, a preuzetih iz članka.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2022.

Zadatak zadao:

Doc. dr. sc. Dario Zlatar

Datum predaje rada:

1. rok: 20. 2. 2023.  
2. rok (izvanredni): 10. 7. 2023.  
3. rok: 18. 9. 2023.

Predviđeni datumi obrane:

1. rok: 27. 2. – 3. 3. 2023.  
2. rok (izvanredni): 14. 7. 2023.  
3. rok: 25. 9. – 29. 9. 2023.

Predsjednik Povjerenstva:

Prof. dr. sc. Milan Vrdoljak

Izjavljujem da sam završni rad izradio samostalno koristeći navedenu literaturu i znanja stečena tijekom studija.

Zahvaljujem se mentoru, doc. dr. sc. Dariu Zlataru na prihvaćanju mentorstva te vodstvu i pomoći tijekom izrade završnog rada.

Zahvaljujem se i svojoj obitelji i kolegama na potpori tijekom čitavog mog školovanja.

Fran Pipunić

## Sažetak

Sustavi s velikim brojem krutih tijela predstavljaju vrlo složen zadatak i predmet su proučavanja brojnih znanstvenika diljem svijeta. Njihova analiza je veoma zahtjevna zbog velikog broja stupnjeva slobode gibanja pa se numeričke simulacije nameću kao jedino rješenje za njihovo proučavanje. Međutim, čak i takve simulacije su veoma zahtjevne za provedbu, prvenstveno sa stajališta računalnog vremena i složenosti proračuna. U ovom radu je provedena numerička simulacija jednog takvog sustava na primjeru kretanja rovera namijenjenog istraživanju Marsa. Rad je podijeljen na četiri cjeline. U prvoj cjelini se opisuje i analizira konstrukcija sustava suspenzije promatranog rovera. Nakon toga su opisane jednačbe koje upravljaju simulacijom, a posebna pozornost je dana modelu kontaktne dinamike temeljenom na diferencijalnim varijacijskim nejednačbama. U trećoj cjelini je detaljno opisan postupak pripreme numeričke simulacije u programskom paketu *Project Chrono* s primjerima koda korištenog za izradu simulacije. Na kraju su grafički prikazani rezultati dobiveni simulacijom te su uspoređeni s rezultatima ekvivalentnog eksperimenta.

Ključne riječi: dinamika sustava više tijela, kontaktna dinamika, DVI, *Project Chrono*, numerička simulacija

# Sadržaj

1. Uvod .....	1
2. Sustav suspenzije.....	2
2.1. Princip rada i osnovni elementi .....	2
2.2. Spajanje suspenzije i tijela rovera .....	4
3. Matematički model .....	6
3.1. Dinamika čestice.....	6
3.2. Dinamika krutog tijela .....	7
3.3. Model kontaktne dinamike .....	10
3.3.1. Metoda diskretnih elemenata (DEM).....	10
3.3.2. Diferencijalne varijacijske nejednakosti (DVI) .....	12
4. Priprema simulacije .....	16
4.1. CAD model.....	16
4.2. Inicijalizacija simulacije.....	18
4.3. Uvoz geometrije .....	19
4.4. Kinematička ograničenja .....	20
4.5. Podloga.....	20
4.6. Pokretanje simulacije i izlazni podaci .....	23
5. Rezultati simulacije.....	25
5.1. Obrada rezultata.....	25
5.2. Prikaz i usporedba rezultata.....	25
6. Zaključak .....	28
Literatura .....	29

## Popis slika

Slika 1. Scarecrow tijekom prelaska preko polusfere [4] .....	1
Slika 2. Slijetanje rovera na Mars pomoću sustava Skycrane [7]. Kada bi se koristili opružni elementi u sustavu suspenzije postojala bi opasnost od udara tijela rovera u tlo zbog sabijanja opruga. ....	2
Slika 3. Osnovni dijelovi rocker-bogie mehanizma [8] .....	3
Slika 4. Shematski prikaz principa rada rocker-bogie mehanizma .....	4
Slika 5. Dodatni elementi za spajanje sustava suspenzije s tijelom [9] .....	5
Slika 6. Gibanje čestice [10] .....	6
Slika 7. Kruto tijelo u gibanju [10] .....	7
Slika 8. Dijagram toka metode diskretnih elemenata .....	11
Slika 9. Dvije čestice u kontaktu [12] .....	12
Slika 10. Dva tijela tijekom kontakta [14] .....	14
Slika 11. Hijerarhijski prikaz programskog paketa Project Chrono [18] .....	16
Slika 12. 3D model rovera izrađen u SolidWorksu .....	17
Slika 13. Primjer markera na spoju desnog rockera i tijela rovera .....	18
Slika 14. Rover i podloga u početnom trenutku simulacije .....	23
Slika 15. Rezultatne sile na kotače dobivene simulacijom .....	26
Slika 16. Rezultatne sile dobivene eksperimentom .....	26
Slika 17. Komponenta brzine u smjeru osi x prednjeg desnog kotača .....	27



## Popis oznaka

$\mathbf{a}$	vektor akceleracija
$d$	promjer čestice
$\mathbf{E}$	jedinična matrica
$\mathbf{f}$	vektor sila
$\mathbf{f}(t, \mathbf{q}, \mathbf{v})$	vektor poopćenih sila
$F_1, F_2, F_3$	reakcijske sile na kotačima
$\mathbf{F}_{k,N}$	normalna sila $k$ -tog sudara
$\mathbf{F}_{k,T}$	tangencijalna sila $k$ -tog sudara
$\mathbf{h}_O$	moment količine gibanja oko ishodišta inercijskog koordinatnog sustava
$\mathbf{H}_R$	matrica koja povezuje derivacije Eulerovih kuteva i kutne brzine
$\mathbf{I}_A$	tenzor tromosti s obzirom na referentnu točku A
$\mathbf{I}_C$	tenzor tromosti s obzirom na središte mase
$k_1, k_2$	krakovi <i>rockera</i>
$k_3, k_4$	krakovi <i>bogiea</i>
$\mathbf{I}_O$	aktivni momenti oko ishodišta inercijalnog koordinatnog sustava
$m$	masa tijela/čestice
$\mathbf{M}$	matrica inercije
$N_C$	ukupni broj sudara
$\mathbf{n}_k$	jedinični vektor normale $k$ -tog sudara
$\mathbf{p}$	količina gibanja
$\mathbf{q}$	vektor poopćenih koordinata
$\mathbf{r}$	vektor položaja u inercijalnom koordinatnom sustavu
$\mathbf{r}'$	vektor položaja u koordinatnom sustavu vezanom za tijelo
$\mathbf{r}_A$	vektor položaja referentne točke na tijelu u inercijalnom koordinatnom sustavu
$\mathbf{r}_C$	vektor položaja središta mase u inercijalnom koordinatnom sustavu
$\mathbf{r}_C'$	vektor položaja središta mase u koordinatnom sustavu vezanom za tijelo
$\mathbf{r}_i$	vektor položaja $i$ -te čestice
$\mathbf{r}_{ij}$	vektor relativnog položaja $i$ -te i $j$ -te čestice

---

$T(\mathbf{q})$	matrica linearnog preslikavanja
$\mathbf{u}_k, \mathbf{w}_k$	jedinični tangencijalni vektori $k$ -tog sudara
$\mathbf{v}$	vektor brzina
$\mathbf{x}_R$	vektor Eulerovih kuteva
$\gamma_{k,n}$	iznos normalne sile $k$ -tog sudara
$\gamma_{k,u}, \gamma_{k,w}$	iznosi tangencijalnih sila $k$ -tog sudara
$\delta_{ij}$	preklapanje $i$ -te i $j$ -te čestice
$\Delta t$	korak integracije
$\varepsilon_j$	kvaternion rotacije $j$ -tog tijela
$\mu_k$	faktor trenja
$\mu_{k,d}$	dinamički faktor trenja
$\mu_{k,s}$	statički faktor trenja
$\varphi, \vartheta, \psi$	Eulerovi kutevi
$\Phi$	funkcija razmaka
$\Psi_i$	funkcija $i$ -te kinematičke veze
$\boldsymbol{\omega}$	kutna brzina
$\tilde{\boldsymbol{\omega}}$	koso simetrična matrica kojom se zamjenjuje vektorsko množenje kutnom brzinom

# 1. Uvod

6. kolovoza 2012. godine, NASA-in *Mars Science Laboratory* (MSL) rover, poznatiji kao *Curiosity*, sletio je na površinu Marsa. Nekoliko godina kasnije, *Curiosity* je pronašao čvrste dokaze o postojanju tekuće vode na Marsu [1], ispunivši tako svoju osnovnu misiju. Osim, ovih neprocjenjivih rezultata, *Curiosity* je demonstrirao i brojne tehnološke inovacije od kojih su brojne korištene i na dvojcu najnovijih marsovskih rovera, američkom *Perseveranceu* i kineskom *Zhurongu*. Sve ovo čini *Curiosity* najpoznatijim robotom koji se kretao površinom Marsa.

S druge strane, *Curiosityjev* uspjeh ne bi bio moguć bez njegovog daleko manje poznatog brata blizanca *Scarecrowa*. *Scarecrow* je NASA-in rover napravljen prvenstveno za testiranje sustava suspenzije korištenog na *Curiosityju*. Naime, *Scarecrow* posjeduje sustav suspenzije identičan onome korištenom na *Curiosityju*, ali ne i svu naprednu elektroniku i znanstvenu opremu ugrađenu u *Curiosity*, zbog čega je i nazvan *Scarecrow* (eng. Strašilo)[2]. To je učinjeno iz dva razloga: prvi je očita financijska isplativost, a drugi je smanjenje mase *Scarecrowa* kako bi se kompenziralo smanjeno gravitacijsko ubrzanje na Marsu u odnosu na Zemlji i na taj način težina *Scarecrowa* na Zemlji bila jednaka težini *Curiosityja* na Marsu [3].

Na *Scarecrowu* su provedeni brojni eksperimenti kako bi se što bolje razumjelo kako će se *Curiosity* kretati po Marsu. Jedan od testova je prikazana na slici 1., a sastoji se od vožnje po ravnoj pjeskovitoj podlozi i prelaska preko čvrsto postavljene aluminijske polusfere radijusa 15 cm.



Slika 1. *Scarecrow* tijekom prelaska preko polusfere [4]

Cilj ovog rada je izraditi numeričku simulaciju prikazanog eksperimenta korištenjem programskog paketa *Project Chrono* te tako dobivene rezultate usporediti s rezultatima provedenog eksperimenta. Pri tome će biti opisan i princip rada sustava suspenzije rovera te matematički model kontaktne dinamike korištene tijekom simulacije.

## 2. Sustav suspenzije

Osnovna zadaća sustava suspenzije je osigurati što je ravnomjerniju raspodjelu sila na svim kotačima tome ne ograničavajući manevarske sposobnosti rovera. Zbog toga se na gotovo svim vozilima namijenjenim kretanju po površini Marsa koristi tzv. *rocker-bogie* sustav. U ovom poglavlju će biti detaljnije opisan princip rada i navedeni osnovni dijelovi ovakvog sustava, te objašnjena njegova ugradnja na tijelo rovera.

### 2.1. Princip rada i osnovni elementi

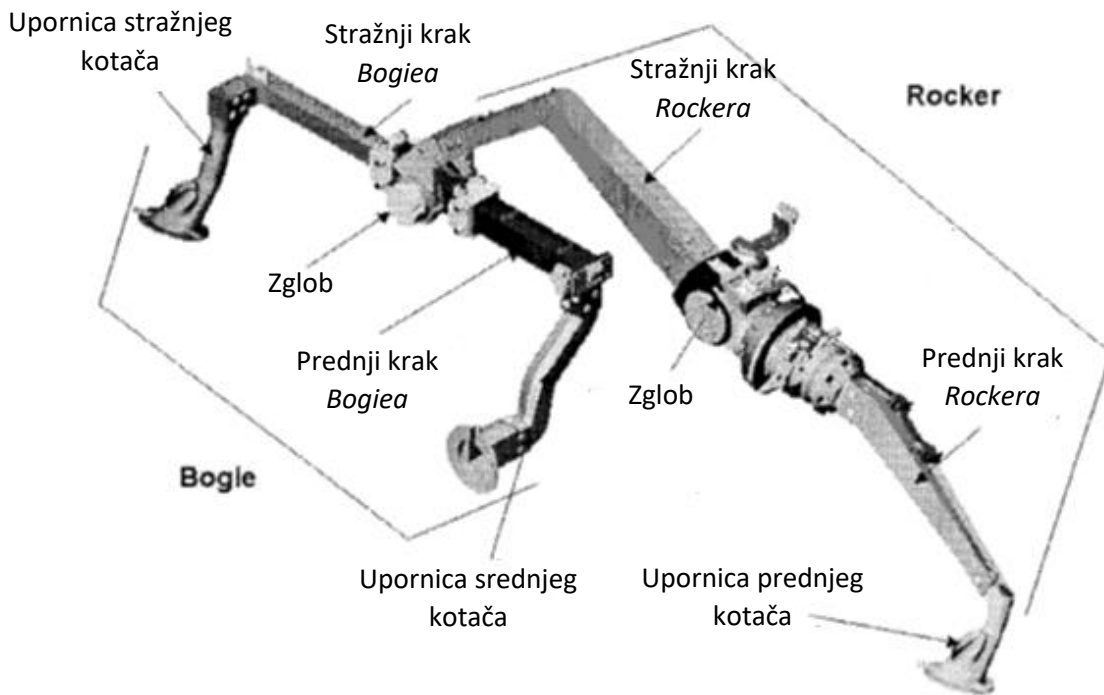
Održavanje ravnomjerne raspodjele sila na kotačima pri različitim nagibima je iznimno bitno za rovere namijenjene istraživanju Marsa jer se na taj način sprječava potonuće pojedinih kotača u pijesak, ali se smanjuje i proklizavanje pojedinih kotača. Ovo je moguće postići, aktivnim sustavom suspenzije [5], korištenjem opružnih elemenata ili *rocker-bogie* mehanizmom. Kako aktivni sustavi suspenzije nisu potpuno pouzdani u svim uvjetima rada, a opruge značajno smanjuju krutost čitavog sustava [6] (što bi stvaralo probleme i mogućnost oštećenja tijela rovera prilikom slijetanja korištenjem sustava *Skycrane*, prikazanog na slici 2.), korištenje *Rocker-Bogie* sustava suspenzije nameće se kao logično rješenje za rovere poput *Curiosityja* i *Scarecrowa*.



Slika 2. Slijetanje rovera na Mars pomoću sustava *Skycrane* [7]. Kada bi se koristili opružni elementi u sustavu suspenzije postojala bi opasnost od udara tijela rovera u tlo zbog sabijanja opruga.

Dva osnovna dijela, po kojima je mehanizam i dobio ime, *rocker* i *bogie* nisu ništa drugo nego dvije klackalice. *Rocker* je glavna klackalica koja se zgloбно veže za tijelo rovera tako da mu je omogućena rotacija oko poprečne osi rovera. Na njegov stražnji kraj je također zgloбно vezan *bogie*, dok se na

prednjem kraju nalazi upornica prednjeg kotača. Na oba kraja *bogiea* se nalaze upornice kotača. Na prednju i stražnju upornicu su preko elektromotora postavljene vilice prednjeg i stražnjeg kotača te je na taj način omogućena bolja upravljivost, dok je vilica srednjeg kotača kruto vezana vilica srednjeg kotača. Na svaku vilicu je vezan kotač sa zasebnim elektromotorom što omogućuje bolju upravljivost. Osnovni dijelovi *rocker-bogie* mehanizma su prikazani i označeni na slici 3.



Slika 3. Osnovni dijelovi *rocker-bogie* mehanizma [8]

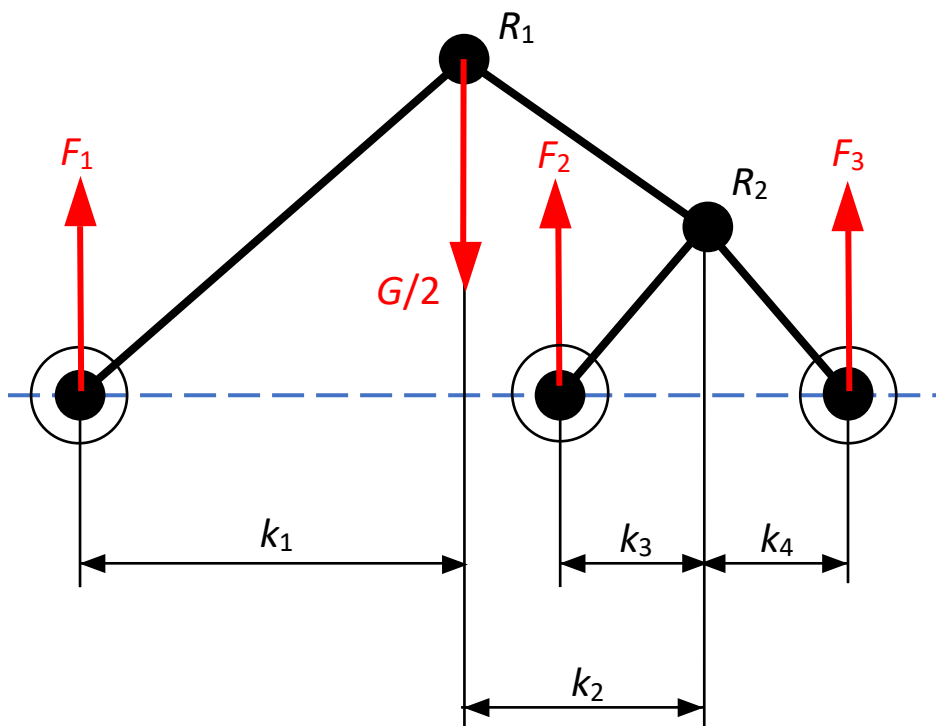
Kako zglobovi prenose sile, ali ne i moment oko osi rotacije, na zglob kojim je *rocker* vezan na tijelo rovera djeluje samo sila uslijed težine tijela rovera,  $G/2$ , kako je shematski prikazano na slici 4. S druge strane, kotači preko vilica na upornice prenose sile reakcije podloge. Pri tome prednji kotač prenosi silu  $F_1$ , srednji silu  $F_2$ , a stražnji silu  $F_3$ , kako je prikazano na slici 4. Da bi *bogie* mirovao, odnosno, da ne bi došlo do njegove rotacije oko zgloba označenog točkom  $R_2$ , sile  $F_2$  i  $F_3$  moraju stvarati jednak moment oko točke  $R_2$  pa mora vrijediti:

$$k_3 F_2 = k_4 F_3. \quad (2.1)$$

S druge strane, kako se zahtjeva da sile na svim kotačima budu jednake, sile  $F_2$  i  $F_3$  također moraju biti jednake pa se može zaključiti da je:

$$k_3 = k_4. \quad (2.2)$$

Nadalje, zbog djelovanja sila  $F_2$  i  $F_3$ , u točki  $R_2$  na *rocker* djeluje rezultantna sila  $F_2+F_3$ , dok na prednjem kraju djeluje sila  $F_1$ .



Slika 4. Shematski prikaz principa rada *rocker-bogie* mehanizma

Ponovno, kako ne bi došlo do rotacije, momenti koje stvaraju ove dvije sile moraju biti jednaki, odnosno:

$$k_1 F_1 = k_2 (F_2 + F_3). \quad (2.3)$$

Pošto se traži da sile  $F_1$ ,  $F_2$  i  $F_3$  budu jednake, mora vrijediti:

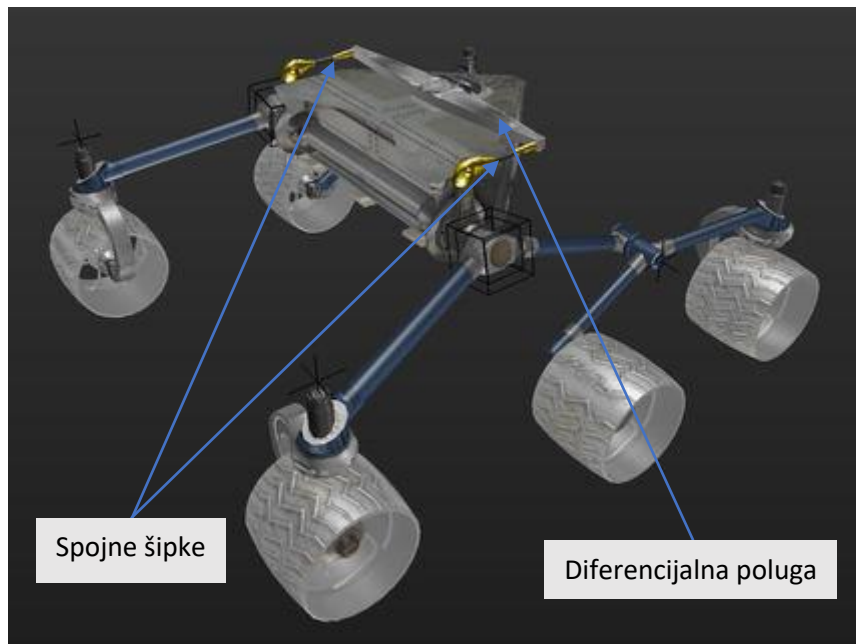
$$k_1 = 2k_2. \quad (2.4)$$

Dakle pri projektiranju rovera, moguće je odrediti dimenzije krakova *rockera* i *bogiewa* za koje će biti ispunjene jednačbe (2.2) i (2.3), odnosno za koje će sile koje djeluju na kotače rovera biti jednake.

## 2.2. Spajanje suspenzije i tijela rovera

Kao što je već rečeno, *rockeri* se na tijelo rovera spajaju zglobo, tako da im je omogućena rotacija oko bočne osi. Međutim, kada bi ovo bile jedine veze između tijela i sustava suspenzije, tijelo rovera bi se moglo pod utjecajem vlastite težine nekontrolirano rotirati oko bočne osi. Kako bi se to

izbjeglo, nužno je dodati još jednu vezu između tijela i sustava suspenzije. To je učinjeno pomoću diferencijalne poluge i šipki koje povezuju polugu s *rockerom*, kao što je prikazano na slici 5.



Slika 5. Dodatni elementi za spajanje sustava suspenzije s tijelom [9]

Diferencijalna poluga je spojena s tijelom pomoću osovinice, te joj je na taj način omogućena rotacija oko vertikalne osi, dok su šipke s odgovarajućim *rockerom* i polugom vezane kuglastim spojevima. Na taj način je definirana još jedna točka kontakta između sustava suspenzije i tijela rovera, a nije spriječena rotacija *rockera*.

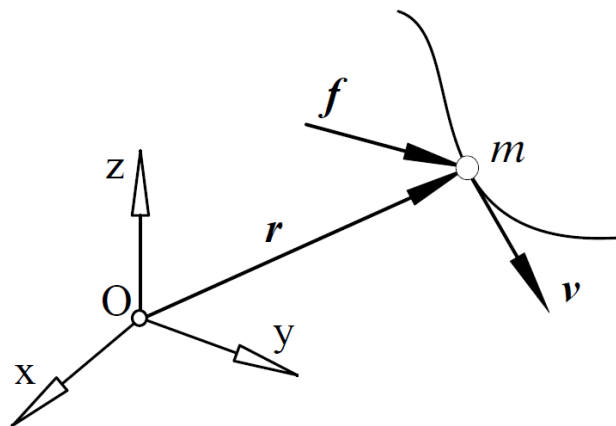
### 3. Matematički model

Za potpuno razumijevanje simulacije važno je najprije upoznati se s matematičkim modelom koji upravlja samom simulacijom. Stoga će u ovom poglavlju biti prikazane jednačbe koje upravljaju kontaktnom dinamikom sustava više tijela. Najprije će se izvesti jednačbe gibanja čestice, a nakon toga će se te jednačbe proširiti na jednačbe gibanja krutog tijela. Na kraju će biti opisan model kontaktne dinamike korišten tijekom simulacije.

#### 3.1. Dinamika čestice

Prije samog izvoda valja razlikovati čestice i tijela. Tijela su objekti koji uz masu posjeduju i momente inercije, odnosno tijela mogu biti podvrgnuta i translaciji i rotaciji, dok čestice mogu biti podvrgnute samo translatornom gibanju. Dakle, razlika između ova dva pojma nije u veličini kao što je to česta zabluda pa se tako pri određenim problemima orbitalne mehanike, zvijezde i planeti mogu promatrati kao čestice, a pri drugim problemima se i mikroskopski objekti mogu promatrati kao tijela.

Dakle, pošto čestica ne posjeduje rotaciju, za potpuno poznavanje njenog gibanja dovoljno je poznavati njen vektor položaja  $\mathbf{r}$  u svakom trenutku, kao što je prikazano na slici 6, pri čemu vektor  $\mathbf{f}$  predstavlja sile koje djeluju na česticu, a vektor  $\mathbf{v}$  brzinu čestice.



Slika 6. Gibanje čestice [10]

Pri tome za brzinu  $\mathbf{v}$  vrijedi:

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}}. \quad (3.1)$$



Sada je direktnom primjenom drugog Newtonovog zakona gibanja moguće doći do vektorske jednadžbe koja u potpunosti opisuje gibanje čestice:

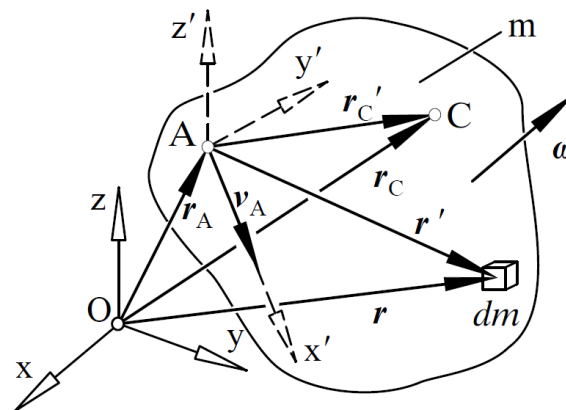
$$\mathbf{f} = \frac{d}{dt}(m\mathbf{v}) = m\dot{\mathbf{v}} = m\ddot{\mathbf{r}}, \quad (3.2)$$

pri čemu  $\ddot{\mathbf{r}}$  predstavlja akceleraciju čestice koja se često označava i s  $\mathbf{a}$ .

### 3.2. Dinamika krutog tijela

Jednadžbe koje opisuju dinamiku krutog tijela često se nazivaju Newton-Eulrove jednadžbe, a njihov izvod prikazan je u nastavku ovog poglavlja [10].

Slika 7. prikazuje kruto tijelo u gibanju pri čemu  $\boldsymbol{\omega}$  predstavlja kutnu brzinu tijela, točka A je proizvoljno odabrana referentna točka tijela s vektorom položaja  $\mathbf{r}_A$  dok je orijentacije tijela definirana pomoću Eulerovih kuteva,  $\mathbf{x}_R = [\varphi \ \vartheta \ \psi]^T$ , između inercijalnog koordinatnog sustava s ishodištem u O i koordinatnog sustava vezanog za tijelo, s ishodištem u točki A.



Slika 7. Kruto tijelo u gibanju [10]

Obzirom da se razmatra kruto tijelo, duljina vektora položaja svake točke tijela u koordinatnom sustavu vezanom za tijelo,  $\mathbf{r}'$  se ne može mijenjati, odnosno vektor  $\mathbf{r}'$  može samo mijenjati svoju orijentaciju uslijed rotacije tijela, pa vrijedi

$$\dot{\mathbf{r}}' = \boldsymbol{\omega} \times \mathbf{r}'. \quad (3.3)$$

Sa slike je očito da vrijedi

$$\mathbf{r} = \mathbf{r}_A + \mathbf{r}' \quad (3.4)$$

pa je moguće dobiti brzinu svake točke krutog tijela kao

$$\mathbf{v} = \mathbf{v}_A + \dot{\mathbf{r}}' = \mathbf{v}_A + \boldsymbol{\omega} \times \mathbf{r}'. \quad (3.5)$$

Također je moguće uspostaviti vezu između kutne brzine i derivacije Eulerovih kuteva

$$\boldsymbol{\omega} = \mathbf{H}_R \dot{\chi}_R. \quad (3.6)$$

Po definiciji, položaj težišta C u inercijalnom koordinatnom sustavu se računa prema jednadžbi

$$\mathbf{r}_C = \frac{1}{m} \int \mathbf{r} dm, \quad (3.7)$$

gdje je  $m$  masa tijela. Analogno tome se računa i položaj težišta u neinercijalnom koordinatnom sustavu vezanom za točku A

$$\mathbf{r}'_C = \frac{1}{m} \int \mathbf{r}' dm. \quad (3.8)$$

Količinu gibanja tijela je sada moguće, korištenjem jednadžbe (3.5) zapisati na sljedeći način

$$\begin{aligned} \mathbf{p} &= \int \mathbf{v} dm = \int (\mathbf{v}_A + \boldsymbol{\omega} \times \mathbf{r}') dm \\ &= \mathbf{v}_A \int dm + \boldsymbol{\omega} \times \int \mathbf{r}' dm. \end{aligned} \quad (3.9)$$

Korištenjem jednadžbe (3.8) dobiveni izraz je moguće napisati kao

$$\mathbf{p} = \mathbf{v}_A m + \boldsymbol{\omega} \times m \mathbf{r}'_C \quad (3.10)$$

Izlučivanjem mase te ponovnom upotrebom jednadžbe (3.5) dobiva se konačni izraz za količinu gibanja krutog tijela

$$\mathbf{p} = m\mathbf{v}_C. \quad (3.11)$$

S druge strane moment količine gibanja oko ishodišta inercijalnog koordinatnog sustava O iznosi

$$\mathbf{h}_O = \int \mathbf{r} \times \dot{\mathbf{r}} \, dm, \quad (3.12)$$

odnosno, nakon uvrštavanja izraza (3.4) i (3.5),

$$\begin{aligned} \mathbf{h}_O &= \int (\mathbf{r}_A + \mathbf{r}') \times (\mathbf{v}_A + \boldsymbol{\omega} \times \mathbf{r}') \, dm \\ &= \mathbf{r}_A \times (\mathbf{v}_A + \boldsymbol{\omega} \times \mathbf{r}'_C)m + \mathbf{r}'_C \times \mathbf{v}_A m + \int \mathbf{r}' \times (\boldsymbol{\omega} \times \mathbf{r}') \, dm. \end{aligned} \quad (3.13)$$

Posljednji član u izrazu (3.13) se može zamijeniti izrazom

$$\int \mathbf{r}' \times (\boldsymbol{\omega} \times \mathbf{r}') \, dm = \int ((\mathbf{r}' \cdot \mathbf{r}')\mathbf{E} - \mathbf{r}'\mathbf{r}') \, dm \cdot \boldsymbol{\omega} = \mathbf{I}_A \cdot \boldsymbol{\omega}, \quad (3.14)$$

gdje je  $\mathbf{E}$  jedinična matrica, a  $\mathbf{I}_A$  tenzor inercije tijela s obzirom na točku A. Ako se kao referentnu točku A uzme težište tijela C, jednadžba (3.13) može se zapisati u sljedećem obliku, pogodnom za računanje momenta količine gibanja

$$\mathbf{h}_O = \mathbf{r}_C \times \mathbf{v}_C m + \mathbf{I}_C \cdot \boldsymbol{\omega}. \quad (3.15)$$

Poznati izrazi za računanje količine gibanja i momenta količine gibanja sada se mogu uvrstiti u Newtonovu odnosno Eulerovu jednadžbu. Newtonova jednadžba nalaže da je promjena količine gibanja jednaka rezultatnoj sili koja djeluje na tijelo

$$\frac{d\mathbf{p}}{dt} = \mathbf{f}, \quad (3.16)$$

dok Eulerova jednadžba nalaže da je promjena momenta količine gibanja jednaka rezultatnom momentu koji djeluje oko promatrane točke (u ovom slučaju ishodišta inercijalnog koordinatnog sustava O)

$$\dot{\mathbf{h}}_O = \mathbf{l}_O. \quad (3.17)$$

Uvrštavanjem izraza (3.11) i (3.15) u izraze (3.16) i (3.17) te njihovim sređivanjem, dobiva se konačni oblik Newton-Eulerovih jednadžbi korišten pri analizi dinamike krutog tijela

$$m\mathbf{a}_C = \mathbf{f}, \quad (3.18)$$

$$I_C\boldsymbol{\alpha} + \tilde{\boldsymbol{\omega}}I_C\boldsymbol{\omega} = \mathbf{l}_C, \quad (3.19)$$

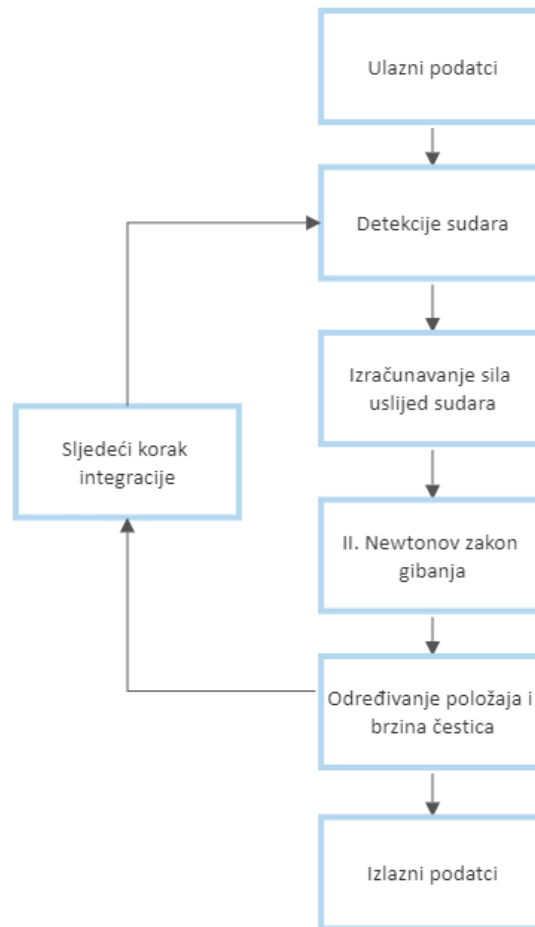
gdje  $\boldsymbol{\alpha}$  kutno ubrzanje tijela,  $\tilde{\boldsymbol{\omega}}$  koso simetrična matrica koja zamjenjuje vektorsko množenje vektorom  $\boldsymbol{\omega}$ , a  $\mathbf{l}_C$  moment koji djeluje na tijelo oko središta mase C.

### 3.3. Model kontaktne dinamike

Pri analizi kontaktne dinamike najznačajniju ulogu imaju dvije metode: metoda diskretnih elemenata (DEM, od eng. *Discrete Element Method*) i diferencijalne varijacijske nejednakosti (DVI, od eng. *Differential Variational Inequalities*). Obje metode imaju svoje prednosti i mane te je programski paket u kojem je izrađena simulacija sposoban koristiti obje metode, ali zbog mogućnosti korištenja većeg koraka integracije [11] (što dovodi do kraćeg vremena potrebnog za provedbu simulacije) u simulaciji je korištena DVI metoda.

#### 3.3.1. Metoda diskretnih elemenata (DEM)

Premda u simulaciji nije korištena metoda diskretnih elemenata, radi boljeg razumijevanja metode korištene tijekom simulacije i njihovih razlika, ukratko će biti prikazan i način rada DEM-a na primjeru analize kontaktne dinamike sustava sfernih čestica [12]. Modeliranje kontakata ovom metodom moguće je prikazati dijagramom toka sa slike 8.



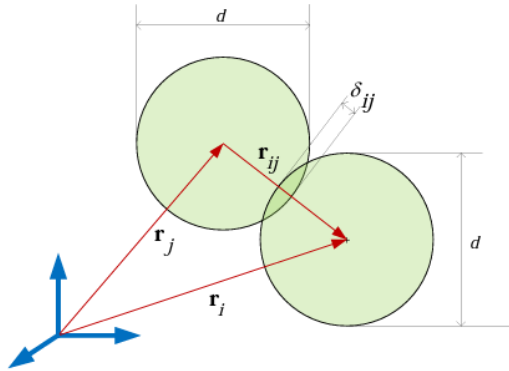
Slika 8. Dijagram toka metode diskretnih elemenata

Za početno zadane podatke položaja i brzina čestica ili podatke dobivene prethodnim korakom integracije najprije se provodi detekcija sudara. Za sferne čestice poznatog polumjera, ovo se provodi vrlo jednostavno, uspoređujući udaljenost njihovih središta s dvostrukom duljinom radijusa. Ako se promatraju  $i$ -ta i  $j$ -ta čestica sustava, tada se položaji njihovih središta dani vektorima  $\mathbf{r}_i$  i  $\mathbf{r}_j$ , a za vektor njihovog relativnog položaja  $\mathbf{r}_{ij}$  vrijedi

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j. \quad (3.20)$$

Udaljenost središta dvaju čestica sada se može izračunati kao norma vektora  $\mathbf{r}_{ij}$ ,  $r_{ij}$ . Ukoliko je udaljenost dvaju čestica manja od promjera čestice  $d$ , te dvije čestice se nalaze u kontaktu kao što je prikazano na slici 9. Također, na slici je označeno i preklapanje čestica  $\delta_{ij}$  za koje vrijedi

$$\delta_{ij} = d - r_{ij}. \quad (3.21)$$



Slika 9. Dvije čestice u kontaktu [12]

Između čestica koje su u kontaktu djeluje normalna i tangencijalna komponenta sile. Ovdje se neće ulaziti u detalje proračuna tih sila, ali će se reći da je normalna komponenta sile proporcionalna preklapanju  $\delta_{ij}$ , a da se tangencijalna komponenta sile može računati prema Coulombovom modelu trenja koji će biti detaljno predstavljen u sljedećem poglavlju.

### 3.3.2. Diferencijalne varijacijske nejednakosti (DVI)

Formulacija diferencijalnih varijacijskih nejednakosti koja će biti iznesena u nastavku prati formulaciju iznesenu u [13] i [14]. U formulaciji se razmatra sustav  $n$  krutih tijela koja se mogu tijekom razmatranja naći u međusobnom kontaktu. Za poznavanje stanja sustava u nekom trenutku potrebno je poznavati vektor poopćenih koordinata

$$\mathbf{q} = [\mathbf{r}_1^T, \varepsilon_1^T, \dots, \mathbf{r}_n^T, \varepsilon_n^T]^T \quad (3.22)$$

te vektor njihovih derivacija

$$\dot{\mathbf{q}} = [\dot{\mathbf{r}}_1^T, \dot{\varepsilon}_1^T, \dots, \dot{\mathbf{r}}_n^T, \dot{\varepsilon}_n^T]^T. \quad (3.23)$$

Pri tome vektor  $\mathbf{r}_j$  predstavlja vektor položaja težišta  $j$ -tog tijela (odnosno odgovara vektoru  $\mathbf{r}_c$  iz prethodnog poglavlja), dok je  $\varepsilon_j$  označen kvaternion koji opisuje rotaciju  $j$ -tog tijela, umjesto prethodno korištenih Eulerovih kuteva. Kvaternioni su matematički objekti koji sadrže četiri skalarna elementa, a ako se koriste za opisivanje rotacije, prvi element predstavlja kut za koji se tijelo rotira oko vektora čije komponente definiraju [15].

Ako se želi izbjeći baratanje s derivacijama kvaterniona, može se definirati vektor poopćenih brzina u kojemu se pojavljuju kutne brzine pojedinih tijela u njihovim lokalnim koordinatnim sustavima

$$\mathbf{v} = [\dot{\mathbf{r}}_1^T, \dot{\boldsymbol{\omega}}_1^T, \dots, \dot{\mathbf{r}}_n^T, \dot{\boldsymbol{\omega}}_n^T]^T. \quad (3.24)$$

Tada je moguće dobiti vezu između vektora  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  i  $\mathbf{v}$  linearnim preslikavanjem

$$\dot{\mathbf{q}} = \mathbf{T}(\mathbf{q})\mathbf{v} \quad (3.25)$$

pokazanim u [16].

Za promatrani sustav krutih tijela moguće je dobiti matricu inercije  $\mathbf{M}$ , dimenzija  $6n \times 6n$ , a za sustav dvaju tijela bi poprimila oblik

$$\mathbf{M} = \begin{bmatrix} m_1 \mathbf{E} & 0 & 0 & 0 \\ 0 & \mathbf{I}_1 & 0 & 0 \\ 0 & 0 & m_2 \mathbf{E} & 0 \\ 0 & 0 & 0 & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{E} & 0 & 0 & 0 \\ 0 & \mathbf{H}_1 & 0 & 0 \\ 0 & 0 & \mathbf{E} & 0 \\ 0 & 0 & 0 & \mathbf{H}_2 \end{bmatrix}, \quad (3.26)$$

gdje je  $\mathbf{E}$  jedinična matrica dimenzija  $3 \times 3$ ,  $\mathbf{I}$  tenzor tromosti odgovarajućeg tijela, a matrica  $\mathbf{H}$  matrica koja daje vezu između derivacija Eulerovih kuteva i kutnih brzina [9].

Dodatno, moguće je definirati i vektor poopcenih sila koje djeluju na tijela u promatranom sustavu, u ovisnosti o vremenu, položaju i brzini pojedinih tijela  $\mathbf{f}(t, \mathbf{q}, \mathbf{v})$  te na taj način postaviti Newton-Eulerovu jednadžbu za promatrani sustav u matričnom obliku

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{f}(t, \mathbf{q}, \mathbf{v}). \quad (3.27)$$

Ovo je osnovna jednadžba koja upravlja sustavom krutih tijela bez kinematičkih veza. Ukoliko postoji  $n$  kinematičke veze u promatranom sustavu, one se očitavaju kroz algebarske jednadžbe oblika

$$\Psi_i(\mathbf{q}, t) = 0, \quad (3.28)$$

za  $i$ -tu kinematičku vezu [17].

Nadalje, pošto se radi o sustavu krutih tijela, prilikom kontakta između dva tijela ne smije doći do prodiranja jednog tijela u drugo. Kako bi se to osigurao uvodi se funkcija razmaka  $\Phi(\mathbf{q}, t)$  koja računa razmak između svih tijela sustava koja su dovoljno blizu da bi između njih moglo doći do sudara te u svakom trenutku između bilo koja dva tijela mora vrijediti

$$\Phi(\mathbf{q}, t) \geq 0. \quad (3.29)$$

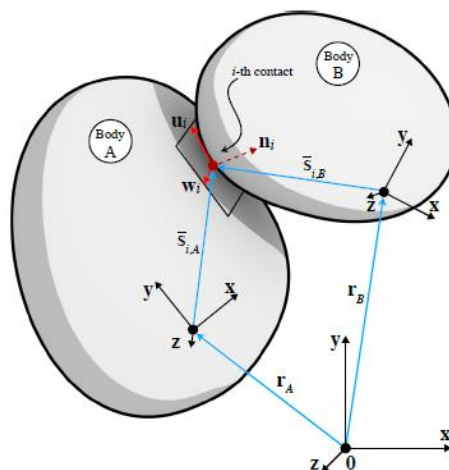
Pri tome, ukoliko tijela nisu u kontaktu, odnosno ako je  $\Phi(\mathbf{q}, t) > 0$ , između ta dva tijela je normalna sila jednaka nuli, a ako su tijela u kontaktu onda je  $\Phi(\mathbf{q}, t) = 0$  te postoji normalna sila između ta dva tijela. Ukoliko se normalna sila koja je vezana za  $k$ -ti sudar definira kao

$$\mathbf{F}_{k,N} = \gamma_{k,n} \mathbf{n}_k, \quad (3.30)$$

gdje je  $\mathbf{n}_k$  jedinični vektor normale na mjestu kontakta dvaju tijela (kao što je prikazano na slici 10), a  $\gamma_{k,n}$  iznos normalne sile  $k$ -tog sudara, prethodno izrečeni slučajevi mogu se matematički zapisati na sljedeći način

$$0 \leq \Phi(\mathbf{q}, t) \perp \gamma_{k,n} \geq 0, \quad k = 1, 2, \dots, N_C, \quad (3.31)$$

gdje je  $N_C$  ukupan broj sudara u sustavu.



Slika 10. Dva tijela tijekom kontakta [14]

Tijekom sudara, osim normalne sile postoji i tangencijalna sila koja se javlja uslijed trenja između dvaju tijela. Za opis trenja koristi se Coulombov model trenja.

Tangencijalna sila prilikom  $k$ -tog kontakta iznosi

$$\mathbf{F}_{k,T} = \gamma_{k,u} \mathbf{u}_k + \gamma_{k,w} \mathbf{w}_k \quad (3.32)$$

pri čemu su vektori  $\mathbf{u}_k$  i  $\mathbf{w}_k$  dva međusobno okomita jedinična vektora u kontaktnoj ravnini, a  $\gamma_{k,u}$  i  $\gamma_{k,w}$  iznosi tangencijalnih sila u smjeru vektora  $\mathbf{u}_k$  odnosno  $\mathbf{w}_k$ . Također je, prema Coulombovom



modelu trenja poznato da ukupna tangencijalna sila djeluje u smjeru relativne tangencijalne brzine između dva tijela te da iznosi

$$\|\mathbf{F}_{k,T}\| = \mu_{k,d} \|\mathbf{F}_{k,N}\|, \quad (3.33)$$

odnosno

$$\sqrt{\gamma_{k,u}^2 + \gamma_{k,w}^2} = \mu_{k,d} \gamma_{k,n}, \quad (3.34)$$

gdje je  $\mu_{k,d}$  dinamički faktor trenja između tijela koja sudjeluju u  $k$ -tom sudaru. Međutim, kada je relativna tangencijalna brzina jednaka nuli, tada smjer djelovanja tangencijalne sile nije tako jednostavno određen nego može djelovati u bilo kojem od beskonačno mnogo tangencijalnih smjerova [16], ali je njen iznos ograničen izrazom

$$\|\mathbf{F}_{k,T}\| \leq \mu_{k,s} \|\mathbf{F}_{k,N}\|, \quad (3.35)$$

gdje je  $\mu_{k,s}$  statički faktor trenja za koji će se radi jednostavnosti pretpostaviti da je jednak dinamičkom faktoru trenja pa će se u buduće ispuštati indeksi  $s$  i  $d$ .

Ova dva slučaja zajednički se mogu opisati sljedećim matematičkim izrazima:

$$\begin{aligned} \mu_k \gamma_{k,n} &\geq \sqrt{\gamma_{k,u}^2 + \gamma_{k,w}^2} \text{ i} \\ \|\mathbf{v}_{k,T}\| (\mu_k \gamma_{k,n} - \sqrt{\gamma_{k,u}^2 + \gamma_{k,w}^2}) &= 0. \end{aligned} \quad (3.36)$$

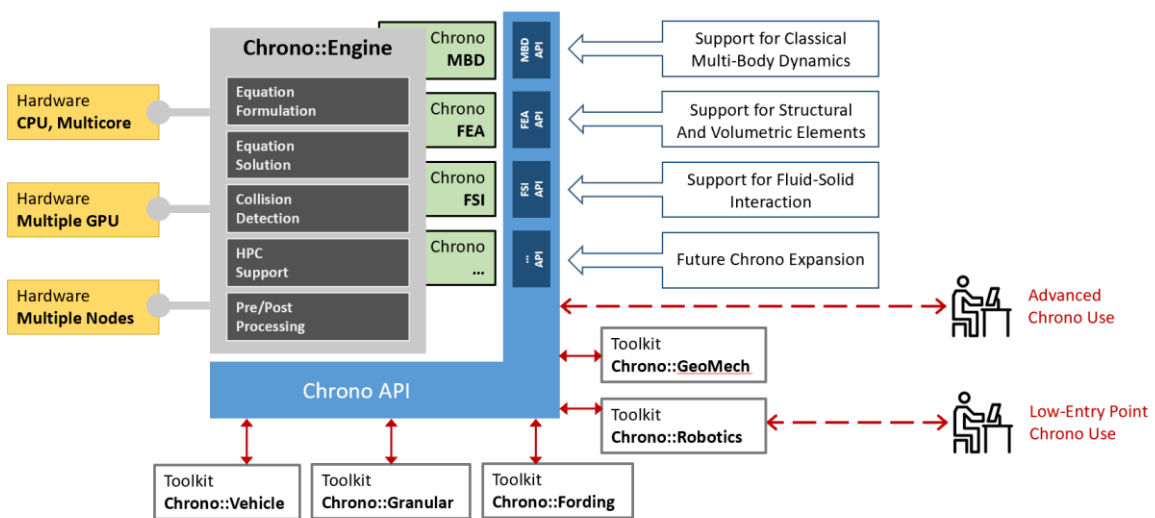
Prvi od ova dva izraza osigurava da se sila trenja nalazi unutar konusa trenja, dok drugi osigurava da ukoliko je relativna tangencijalna brzina dvaju tijela različita od nule sila trenja bude jednaka umnošku normalne sile i faktora trenja.

Izrazi (3.25), (3.27), (3.28), (3.31) i (3.36) u potpunosti definiraju problem diferencijalnih varijacijskih jednakosti potreban za promatranje kontaktne dinamike sustava krutih tijela.

Za kraj valja napomenuti da je prikazani problem pogodan za rješavanje numeričkim putem te se može pokazati da takvo rješenje konvergira rješenju početnog problema kada korak integracije  $\Delta t$  teži nuli [16].

## 4. Priprema simulacije

Simulacija sustava je provedena u programskom paketu *Project Chrono*. *Project Chrono* je programsko okruženje otvorenog koda, što znači da svatko ima pristup izvornom kodu te nije potrebno kupovati licencu. Sastavljen je od skupa programskih biblioteka, a može se koristiti rješavanje problema dinamike sustava čestica, nelinearne metode konačnih elemenata te interakcije između fluida i krutih objekata. Hijerarhijska struktura programskog paketa je prikazan na slici 11. Važno je naglasiti da nije nužno instalirati čitav *Project Chrono*, nego se, zahvaljujući njegovoj modularnoj arhitekturi, mogu koristiti i samo pojedini njegovi dijelovi (moduli), ovisno o potrebama.



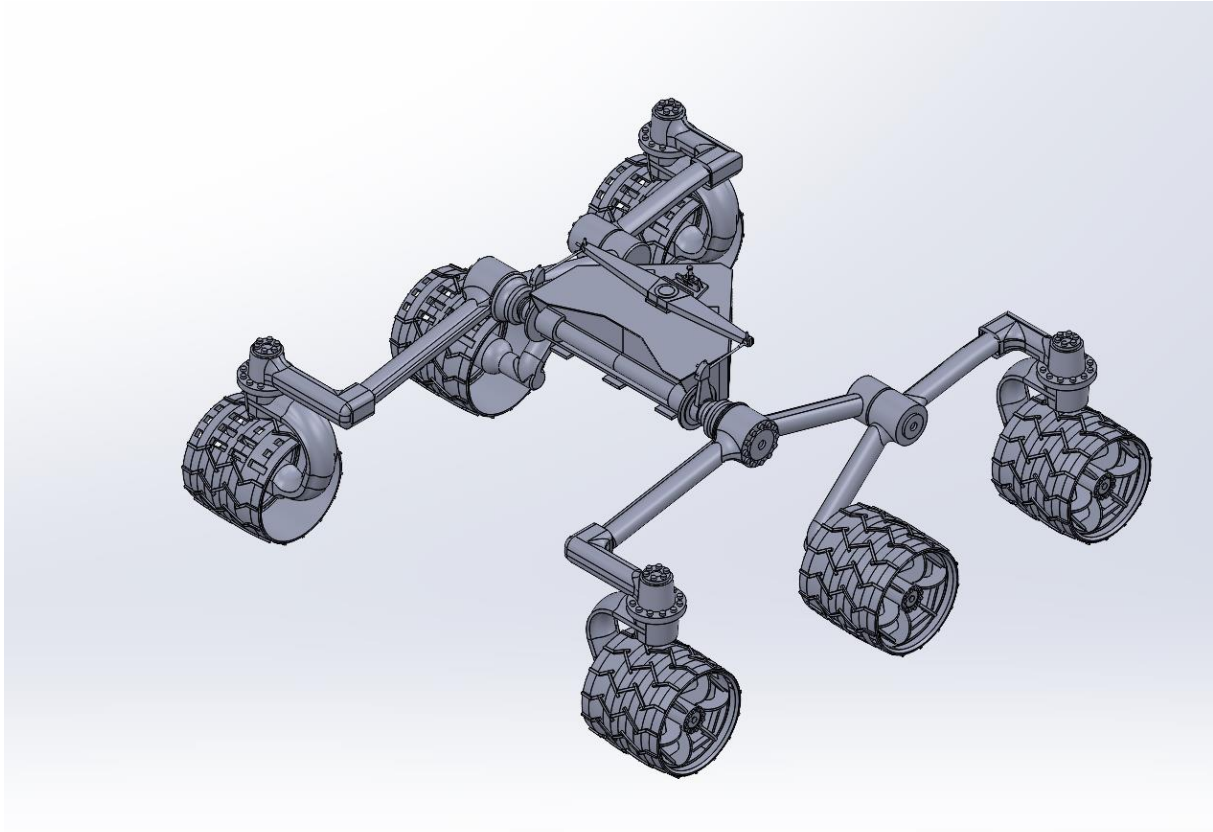
Slika 11. Hijerarhijski prikaz programskog paketa *Project Chrono* [18]

Obzirom da *Project Chrono* ne posjeduje grafičko sučelje, svi parametri simulacije moraju biti postavljeni unošenjem koda u programskom jeziku C++. U ovom poglavlju će biti opisan proces izrade simulacije uz primjere korištenog koda i pripadajuća objašnjenja.

### 4.1. CAD model

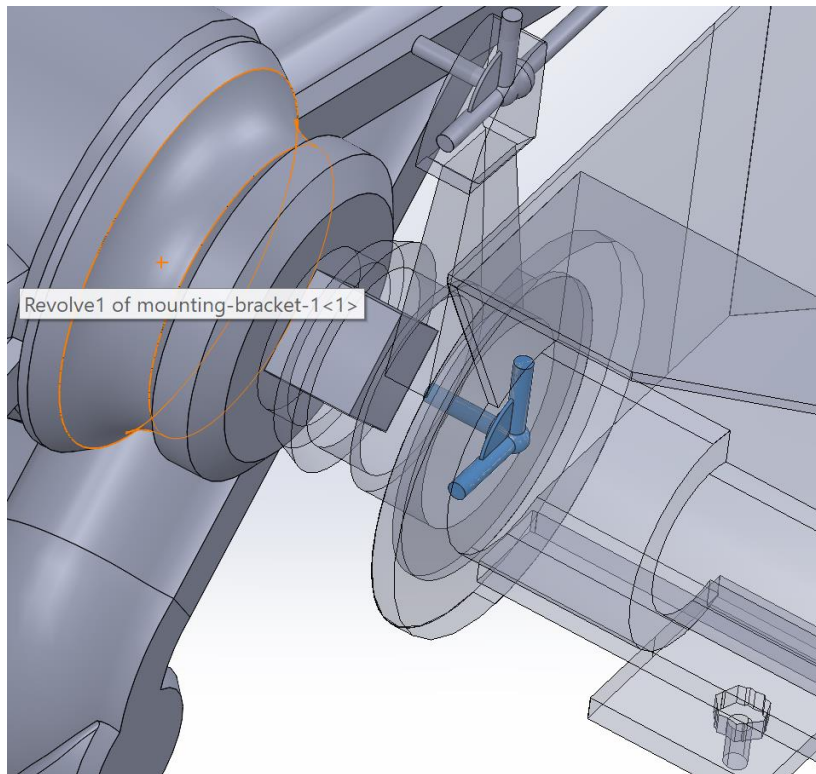
Premda *Project Chrono* sam po sebi ima mogućnost stvaranja geometrije, ona je ograničena na vrlo oblike poput sfera, cilindara, kvadara i sličnih osnovnih geometrijskih oblika. Stoga je, prije izrade simulacije, potrebno izraditi 3D model rovera koji će se koristiti tijekom simulacije. Kako je za simulaciju potreban 3D model u .STEP formatu, za ovu namjenu se može koristiti bilo koji od popularnih CAD alata poput *SolidWorksa*, *Autodesk Inventora*, *CATIA-e*, itd. Za izradu 3D modela korištenog u simulaciji korišten je *SolidWorks*, a model je napravljen prema 3D modelima dostupnim na [8] i [10]. Ovi 3D modeli su korišteni samo kao predlošci na temelju kojih je napravljen konačni model. Pošto je model rovera *Scarecrow*, dostupan na [8], izrađen u .blend formatu, njegovo

korištenje u *SolidWorksu*, čak i kao predložka, je iznimno otežano. Stoga je samo tijelo rovera izrađeno prema ovom modelu, dok je sustav suspenzije izrađeno prema modelu rovera *Curiosity* [19], dostupnom u .STL formatu (originalno namijenjenom za korištenje prilikom proizvodnje pomoću aditivnih tehnologija). Činjenica da je za izradu modela sustava suspenzije korišten model *Curiosityja*, ne bi trebala izazvati nikakve problema jer je suspenzija tih dvaju rovera identična[3].



Slika 12. 3D model rovera izrađen u *SolidWorksu*

Nakon što je izrađen konačan model rovera, a prije prelaska na izradu simulacije, potrebno je na odgovarajućim mjestima postaviti pomoćne dijelove, tzv. *markere*, koji neće biti direktno korišteni u simulaciji, ali će se koristiti njihove pozicije za postavljanje kinematičkih ograničenja. Primjer jednog *markera*, na spoju desnog *rockera* i tijela rovera dan je na slici 13. Nakon što je 3D model gotov, vrlo ga je jednostavno izvesti u .STEP format, pomoću *SolidWorksove* ugrađene naredbe za izvoz u .STEP format.



Slika 13. Primjer *markera* na spoju desnog *rockera* i tijela rovera

## 4.2. Inicijalizacija simulacije

Prilikom pisanja koda, najprije je potrebno deklarirati i uključiti sve biblioteke koje će biti korištene tijekom pisanja koda. Kod kojim su uključene biblioteke dan je u nastavku:

```
#include "chrono/core/ChRealtimeStep.h"  
#include "chrono/physics/ChLinkMotorRotationSpeed.h"  
#include "chrono/physics/ChSystemNSC.h"  
#include "chrono/physics/ChBodyEasy.h"  
#include "chrono_cascade/ChCascadeBodyEasy.h"  
#include "chrono_cascade/ChCascadeDoc.h"  
#include "chrono_cascade/ChCascadeVisualShape.h"  
#include "cmath"
```

Nakon toga je nužno inicijalizirati sustav koji će baratati svim tijelima i kinematičkim ograničenjima unutar simulacije. Za to je korištena naredba `ChSystemNSC` kojom se odmah na početku određuje i model kontaktne dinamike korišten u simulaciji.

### 4.3. Uvoz geometrije

*CASCADE* modul omogućuje jednostavan uvoz geometrije iz *.STEP* formata u simulaciju. Pomoću *CASCADE* modula je moguće uvesti pojedine dijelove, podsklopove ili čitave sklopove iz *.STEP* datoteke, ali svaki sklop koji se direktno uveze će biti tretiran kao jedno kruto tijelo u simulaciji. Primjer ovakvog uvoza geometrije prikazan je u nastavku.

```
bool load_ok = mydoc.Load_STEP(GetChronoDataFile("cascade/rover.STEP").c_str());
mydoc.Dump(GetLog());

std::shared_ptr<ChCascadeBodyEasy> body;

if (load_ok) {
    TopoDS_Shape shape1;
    if (mydoc.GetNamedShape(shape1, "rover/body"))
    {
        body = chrono_types::make_shared<ChCascadeBodyEasy>(shape1, 807, true,
true);
        sys.Add(body);

        GetLog() << "body succesfully imported! \n";
    }
    else
        GetLog() << "Warning. Desired object not found in document \n";
}
else
    GetLog() << "Warning: rover.STEP file could not be opened/parsed \n";
```

Prvim retkom ovog primjera se učitava *.STEP* datoteka i istovremeno stvara varijabla *load\_ok* koja poprima vrijednost 1 ukoliko je datoteka učitana, odnosno 0 ako datoteka nije učitana. Drugim retkom se ispisuju imena svih podsklopova i dijelova koji se nalaze u učitanj datoteci, dok se u trećem retku stvara pokazivač koji će se koristiti svaki put kada se želi provesti neka operacija s određenim dijelom. Nakon toga se ulazi u *if-else* petlju u kojoj će uvjet biti zadovoljen ako je datoteka uspješno učitana (ako varijabla *load\_ok* ima vrijednost 1), a u suprotnome će se ispisati poruka da tražena datoteka nije pronađena. Potom se na vrlo sličan način formira još jedna *if-else* petlja koja najprije kreira oblik iz traženog dijela, a potom naredbom *ChCascadeBodyEasy* i tijelo koje se naredbom *sys.Add* dodaje u ranije stvoreni sustav. Argumenti naredbe *ChCascadeBodyEasy* su redom: oblik iz kojeg se stvara tijelo, gustoća tijela, sudjelovanje u vizualizaciji te sudjelovanje u kontaktima s drugim tijelima. Ovdje je pretpostavljena uniformna raspodjela gustoće od 807 kg/m<sup>3</sup> u cijelom roveru, a dobivena je na temelju poznate mase rovera od 318 kg [3] i volumena rovera izračunatog u *SolidWorksu*.

## 4.4. Kinematička ograničenja

Nakon što je uvezena geometrija rovera potrebno je propisati kinematička ograničenja između pojedinih krutih tijela od kojih se rover sastoji. Za to će poslužiti ranije opisani *markeri* pomoću kojih možemo dobiti koordinate na kojima je potrebno propisati kinematička ograničenja. Ovaj postupak će biti prikazan na primjeru zglobnog oslonca između desnog *rockera* i tijela rovera, na mjestu *markera* prikazanog na slici 8.

```
TopoDS_Shape shapeMarker3;  
ChFrame<> referenceMarker3;  
if (mydoc.GetNamedShape(shapeMarker3, "rover/markerCoordSys#3"))  
{  
    ChCascadeDoc::FromCascadeToChrono(shapeMarker3.Location(), referenceMarker3);  
    GetLog() << "referenceMarker3 succesfully imported! \n";  
}  
else  
    GetLog() << "warning: desired marker not found in document - Marker3 \n";  
auto revolute1 = chrono_types::make_shared<ChLinkLockRevolute>();  
revolute1->Initialize(body, rightRocker, referenceMarker3.GetCoord());  
sys.AddLink(revolute1);
```

Slično kao i kod uvoza geometrije, najprije se stvara oblik traženog *markera*, ali se u ovom slučaju iz njega ne stvara tijelo nego se uzimaju njegove koordinate i na njihovo mjesto se postavlja koordinatni sustav imena *referenceMarker3* na čijem se mjestu naredbom *ChLinkLockRevolute* stvara zglobni oslonac.

Na ovaj način je propisano ukupno 19 kinematičkih ograničenja od čega je 5 zglobnih oslonaca, 4 kuglasta spoja, 4 kruta spoja i 6 motora s propisanom brzinom vrtnje. Zglobni oslonci su korišteni na spojevima *rockera* i tijela, *rockera* i *bogiea* te na spoju diferencijalne poluge i tijela rovera. Kuglasti spojevi su korišteni na spojevima diferencijalne poluge i spojnih šipki te na spojevima spojnih šipki i *rockera*, dok su kruti spojevi korišteni na spojevima vilica prednjih i stražnjih kotača s odgovarajućim upornicama. To je učinjeno tako jer se u konkretnoj simulaciji rover treba kretati samo ravno, ali je moguća promjena u budućim simulacijama.

## 4.5. Podloga

Sljedeći element simulacije koji je potrebno napraviti je podloga. U sklopu nje je potrebno izraditi krutu površinu na kojoj će sva ostala tijela stajati, polusferu koja predstavlja prepreku u provedenom eksperimentu te čestice tla.

Najprije je pomoću naredbe *ChBodyEasyBox* stvoren kvadar koji predstavlja krutu podlogu na kojoj će se nalaziti čestice tla i rover. Naredbom *SetBodyFixed* je osigurano da kvadar bude fiksiran na istom mjestu tijekom cijele simulacije.

Nakon toga je potrebno stvoriti prepreku. Prepreka u eksperimentu je polusfernog oblika, ali *Project Chrono* ne nudi mogućnost stvaranja polusfera, nego samo sfera. Međutim, ovo ne predstavlja

problem, pošto će se sfera pozicionirati tako da donja polovica bude zatrpána česticama pa će iz perspektive rovera prepreka uistinu biti polusfera. Sfera je stvorena pomoću naredbe *ChBodyEasySphere*, a njeno pozicioniranje u prostoru je riješeno naredbama *SetPos* i *SetBodyFixed*. Cijeli proces stvaranja sfere dan je u kodu ispod.

```
std::shared_ptr<ChBodyEasySphere> obstacle(new ChBodyEasySphere(0.15, 1000, true, true));
obstacle->SetBodyFixed(true);
obstacle->SetPos(ChVector<>(0.3, -0.25, 0.15));
sys.Add(obstacle);
```

Argumenti u naredbi *ChBodyEasySphere* redom predstavljaju radijus, gustoću, sudjelovanje u vizualizaciji i sudjelovanje u sudarima s drugim tijelima. Pošto je sfera fiksirana u prostoru, njena je masa nebitna za rezultate simulacije pa je za gustoću izabrana proizvoljna vrijednost od  $1000 \text{ kg/m}^3$ .

Treći i najsloženiji dio stvaranja podloge je stvaranje čestice tla. Obzirom da složenost i vrijeme potrebno za provedbu simulacije drastično rastu s povećanjem broja čestica cilj je bi smanjiti broj čestica na način da se smanji površina koja se mora prekriti njima. Zbog toga se česticama prekrivaju samo dvije trake po kojima će se kretati kotači rovera tijekom simulacije, pri čemu je širina svake trake jednaka dvostrukoj širini kotača, kako bi se ipak mogao vidjeti utjecaj i čestica koje nisu ravno ispod kotača. Da bi se spriječilo raspršivanje čestica po čitavoj površini podloge, oko traka ispunjenih česticama su napravljeni zidovi pomoću naredbi *ChBodyEasyBox*.

Čestice u svakoj traci se generiraju pomoću trostrukih ugniježđenih *for* petlji, tako se pomičući po *x*, *y* odnosno *z* osi. Također, prije samih petlji je definiran materijal čestica pri čemu je zadan koeficijent trenja 0.4, koeficijent restitucije 0.9 te gustoća  $2000 \text{ kg/m}^3$ . Slijedi kod kojim se generiraju čestice unutar lijeve trake.

```

double pRad = 0.03; //particle radius
double pRho = 2000; //particle density
double delta = 0.001; //initial distance between two particles (to avoid collision
at moment zero)
double numx = (6 - delta) / (2 * pRad + delta); //number of particles in each axis
direction
double numz = (0.9 - delta) / (2 * pRad + delta);
double numy = (0.2 - delta) / (2 * pRad + delta);
int n = 0; //counter for total number of particles
//nested for loops for creating spherical ground particles, 1st for left track
ChVector<> refPosL = ChVector<>(-3.0 + pRad + delta, -0.45 + pRad + delta, -2.8 + pRad
+ delta); //position of 1st particle

for (int iy = 0; iy < numy - 1; ++iy) {
    for (int ix = 0; ix < numx-1; ++ix) {
        for (int iz = 0; iz < numz-1; ++iz) {
            ChVector<> relPos = ChVector<>(ix * (2 * pRad + delta), iy * (2 * pRad +
delta), iz * (2 * pRad + delta)); //position relative to 1st particle
            ChVector<> absPos = refPosL + relPos; //absolute position of each
particle
            n++; //particle number counter

            std::shared_ptr<ChBodyEasySphere> particle(new
ChBodyEasySphere(pRad, pRho, true, true, p_mat));
            particle->SetPos(absPos);
            sys.Add(particle);
        }
    }
}

```

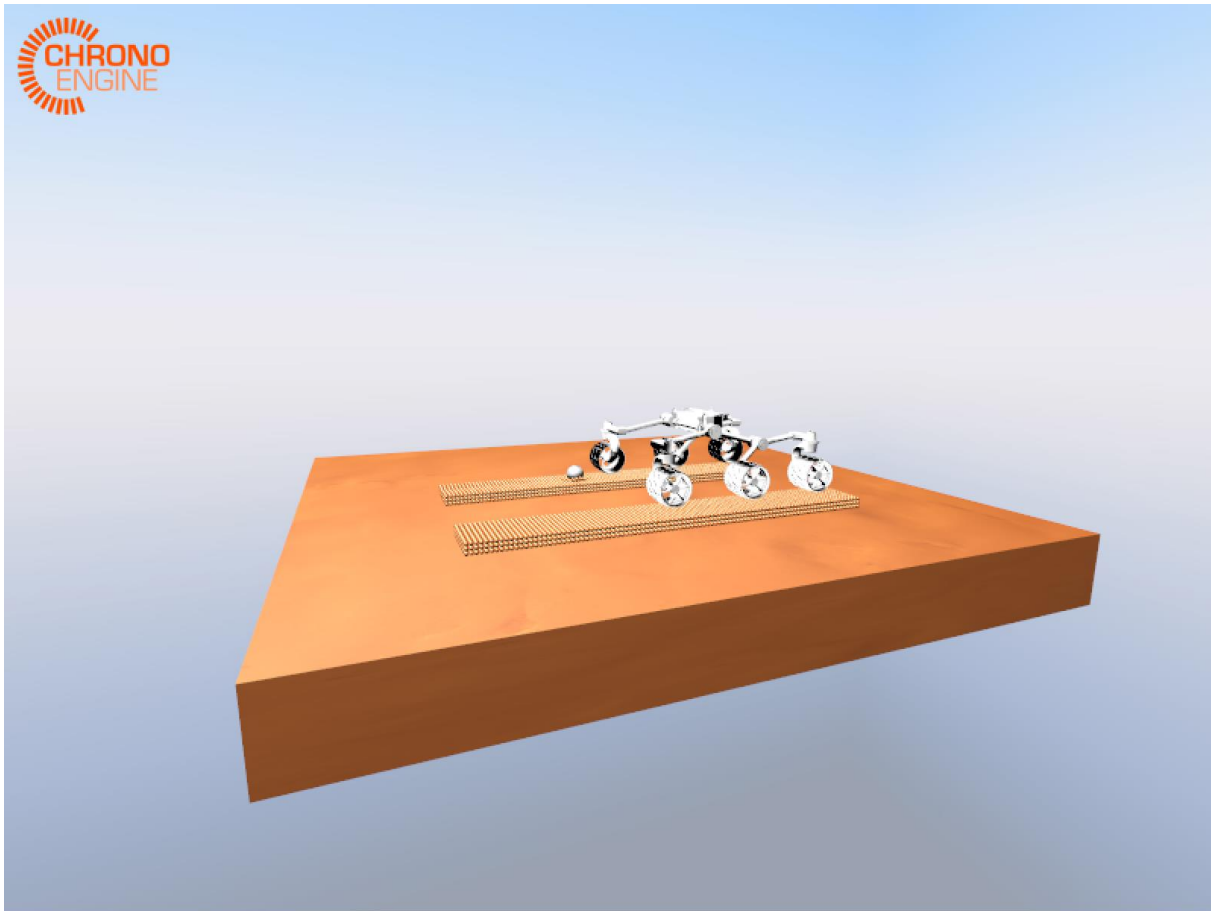
Kao što je vidljivo čestice su kuglice radijusa 3 cm i u početnom trenutku je ostavljen razmak od 1 mm između svake čestice kako bi se izbjeglo sudaranje čestica na samom početku simulacije. Varijabla *refPosL* definira vektor položaja prve čestice u lijevoj traci, a *relPos* položaj trenutne čestice u odnosu na početnu česticu. Tada je vektor položaja trenutne čestice u iteraciji *absPos* jednak zbroju varijabli *refPosL* i *relPos* pa je na već pokazani način, korištenjem naredbe *ChBodyEasySphere* generirati česticu i zadati joj početnu poziciju.

Radijus čestica od 3 cm je veoma velik za čestice pijeska, ali zbog ograničenosti računalnih resursa i činjenice da se već i s tolikim radijusom ukupan broj čestica popeo na 8232, nije bilo moguće provesti simulaciju s većim brojem čestica manjeg radijusa.

Kako bi se izbjegla mogućnost bilo kakve zabune, čestice o kojima se priča u ovom poglavlju nisu iste čestice o kojima se pričalo u poglavlju 3.1. Dinamika čestice. Čestice koje se spominju u ovom poglavlju su sa stajališta dinamike kruta tijela i *Project Chrono* ih promatra kao takve.

Slika 14. prikazuje uvezeni model rovera i kreiranu podlogu u početnom trenutku simulacije. Na slici je moguće uočiti veliki kvadar koji služi kao osnovna podloga i sferu koja predstavlja prepreku preko koje rover mora prijeći te dvije trake ispunjene česticama. Zidovi koji omeđuju trake s česticama su ovdje namjerno učinjeni nevidljivima kako bi se bolje vidjele čestice, ali oni i dalje postoje i utječu na ponašanje čestica





Slika 14. Rover i podloga u početnom trenutku simulacije

## 4.6. Pokretanje simulacije i izlazni podaci

Sada, kada su svi elementi simulacije spremni, preostaje samo definirati izlazne podatke i vremenske parametre simulacije. Pošto su iz eksperimenta poznate samo rezultantne sile na desne kotače (kotače koji prelaze preko prepreke), izlazni podaci sadrže upravo te sile za svaki trenutak u kojemu se provodi simulacija. Za dobivanje tih podataka najprije je potrebno stvoriti datoteke u .txt formatu u koje će se upisivati dobiveni rezultati. To je učinjeno sljedećim kodom:

```
ChStreamOutAsciiFile result_FRwheel("output_forces_on_front_right_wheel.txt");  
ChStreamOutAsciiFile result_MRwheel("output_forces_on_middle_right_wheel.txt");  
ChStreamOutAsciiFile result_RRwheel("output_forces_on_rear_right_wheel.txt");
```

Nakon toga su podešeni vremenski parametri simulacija, tako da je ukupno trajanje simulacije 30 sekundi, a korak 0.001 s. Nakon toga se pristupa jednostavnoj *while* petlji koja za vrijeme trajanja simulacije provodi simulaciju, u zadane datoteke upisuje sile na kotače te ispisuje trenutno vrijeme integracije.

```
while (time <= timeEnd) {  
    sys.DoStepDynamics(timeStep);  
  
    double FRx = engine1->Get_react_force().x();  
    double FRy = engine1->Get_react_force().y();  
    double FRz = engine1->Get_react_force().z();  
    double FRforce = sqrt(pow(FRx, 2) + pow(FRy, 2) + pow(FRz, 2));  
  
    double MRx = engine2->Get_react_force().x();  
    double MRy = engine2->Get_react_force().y();  
    double MRz = engine2->Get_react_force().z();  
    double MRforce = sqrt(pow(MRx, 2) + pow(MRy, 2) + pow(MRz, 2));  
  
    double RRx = engine3->Get_react_force().x();  
    double RRy = engine3->Get_react_force().y();  
    double RRz = engine3->Get_react_force().z();  
    double RRforce = sqrt(pow(RRx, 2) + pow(RRy, 2) + pow(RRz, 2));  
  
    result_FRwheel << FRforce << "; ";  
    result_MRwheel << MRforce << "; ";  
    result_RRwheel << RRforce << "; ";  
  
    GetLog() << time << "\n";  
  
    time += timeStep;}
```

## 5. Rezultati simulacije

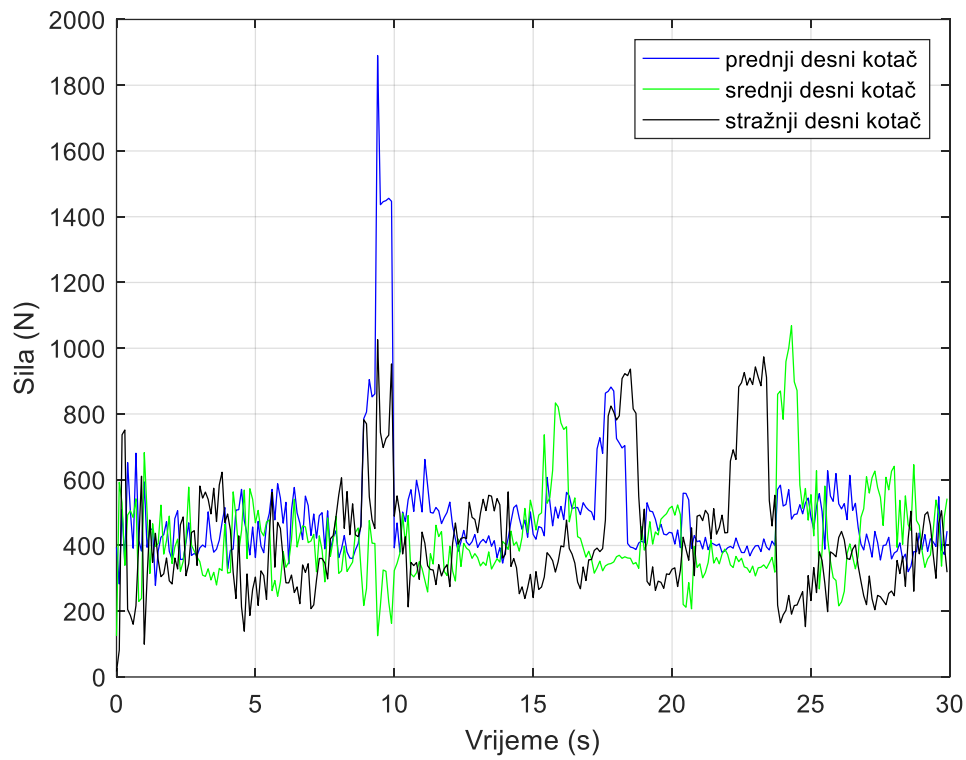
Kada je simulacija završena u .txt datotekama koje su zadane kao izlazne datoteke u koje se upisuju rezultati nalazi se po 30 000 rezultantnih sila na svaki kotač (za svaki korak integracije po jedna). Kada bi se crtali dijagrami na temelju tih sirovih podataka, dijagrami bi bili zagušeni zbog čestih oscilacija rezultata. Stoga su prije crtanja dijagrama rezultati obrađeni u programskom paketu *Matlab* u kojem su onda nacrtani i dijagrami. U ovom poglavlju će najprije biti opisano kako su rezultati obrađeni, koje su potencijalne opasnosti takve obrade podataka te će na kraju biti prikazani dijagrami u kojima se uspoređuju rezultati dobiveni simulacijom s rezultatima dobivenim eksperimentom.

### 5.1. Obrada rezultata

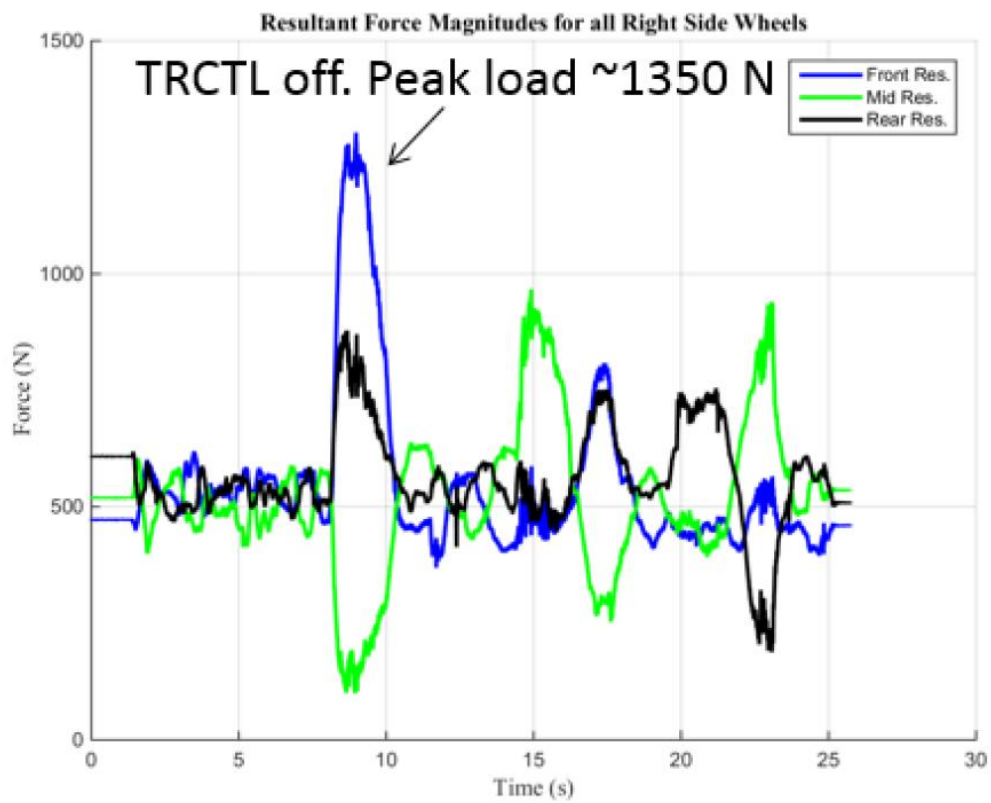
Kako bi se smanjila zagušenost dijagrama s rezultatima bilo je potrebno smanjiti broj točaka koje će se prikazati na dijagramu. Odlučeno je da se broj točaka smanji 300 puta, odnosno da se na dijagram ucrtava jedna točka za svaku desetinku sekunde simuliranog vremena. Međutim, kada bi se samo uzeli podatci dobiveni u točno desetinki sekunde postojala bi velika opasnost da zbog već spomenutih oscilacija prikazani rezultati ne budu reprezentativni. Stoga je vrijednost u svakoj točki prikazanoj na dijagramu dobivena računanjem aritmetičke sredine 50 rezultata prije i 50 rezultata nakon trenutka u kojem će biti ucrtana točka na dijagramu. Na ovaj način se postiže traženo smanjenje broja točaka na dijagramu, ali se i osigurava da svaki dobiveni rezultat jednako utječe na konačan izgled dijagrama. Naravno, opasnost ovakvog pristupa je da se pojedini izolirani ekstremi mogu vrlo lako „zagubiti“ prilikom uprosječavanja, ali ako se radi o izoliranim ekstremima tada je puno vjerojatnije da se jednostavno radi o grešci tijekom simulacije, nego o fizikalnoj pojavi. S druge strane, ako se ne radi izoliranom slučaju, tada će već i svega nekoliko ekstrema imati dovoljan utjecaj da pojedine točke dijagrama izgledaju sumnjivo.

### 5.2. Prikaz i usporedba rezultata

Na slici 12 su prikazani rezultati dobiveni simulacijom, dok slika 13 prikazuje rezultate dobivene eksperimentom, dostupne u literaturi [3].



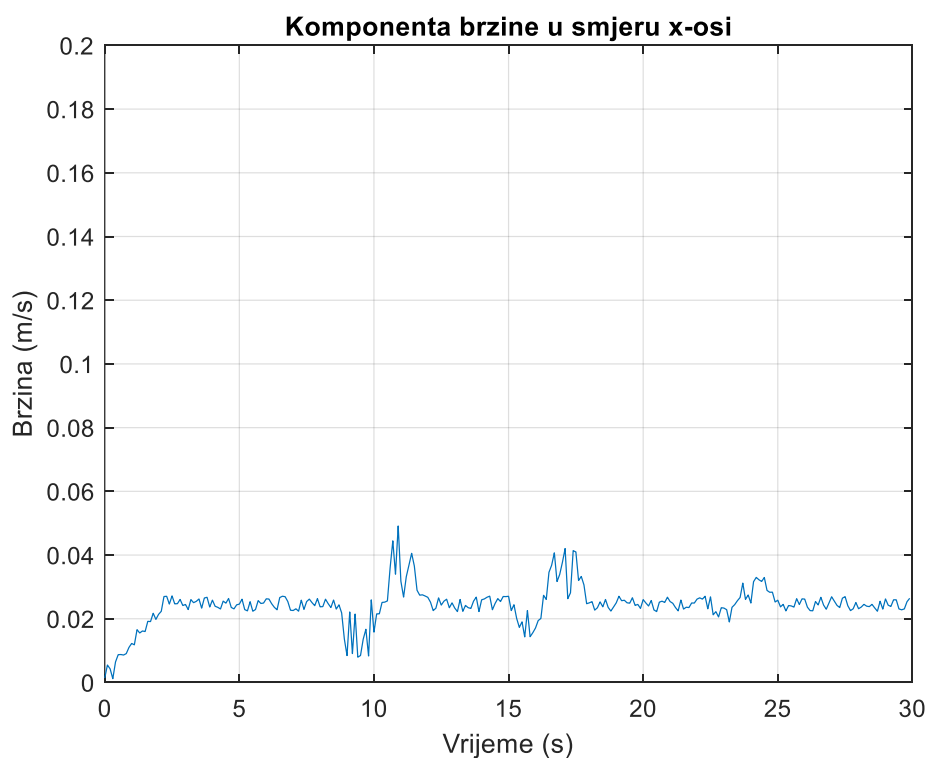
Slika 15. Rezultatntne sile na kotače dobivene simulacijom



Slika 16. Rezultantne sile dobivene eksperimentom

Usporedbom tih dviju slika, odmah je jasno da simulacija ne daje identične rezultate kao eksperiment, ali su karakteristični ekstremi koji se javljaju u trenucima kada pojedini kotač rovera prelazi preko prepreke prisutni. Prvo značajno odstupanje simulacije od eksperimenta javlja se na samome početku simulacije gdje su prisutne ogromne oscilacije. Te oscilacije su rezultat činjenice da na samome početku simulacije, za razliku od eksperimenta, rover ne može biti u kontaktu s tlom, a oscilacije nastaju prilikom udarca rovera o tlo. Još jedna posljedica tog udarca je ta da simulacija vremenski blago zaostaje za eksperimentom, što je najbolje vidljivo usporedbom trenutaka u kojima se događaju karakteristični skokovi na dijagramima. Nadalje, tijekom simulacije su vidljive i značajne oscilacije između ekstrema, značajno veće nego u eksperimentu što je najvjerojatnije posljedica grubih čestica korištenih u simulaciji. Za kraj, vidljivo je i da je prvi karakteristični skok znatno veći u rezultatima dobivenim simulacijom nego kod rezultata eksperimenta, ali su drugi skokovi značajno bliži po svojim iznosima. Ovo je vjerojatno rezultat brojnih faktora, ali jedan od njih, a vrlo lako moguće i najznačajniji, je pretpostavka o uniformnoj raspodjeli gustoće po čitavom roveru koja u stvarnosti nije ni približno zadovoljena.

Uzrok ranije spomenutom zaostajanju simulacije za eksperimentom najbolje je vidljiv na slici 17. gdje je prikazana ovisnost komponente brzine prednjeg desnog kotača u smjeru osi  $x$  o vremenu. Pri tome je os  $x$  usmjerena prema naprijed, odnosno, pozitivna komponenta brzine u smjeru osi  $x$  znači da se rover kreće prema naprijed. Na slici je vidljivo da je u početnom trenutku brzina jednaka nuli te je potrebno nekoliko sekundi da se rover ubrza na propisanu brzinu. Nadalje, vidljivi su i skokovi u brzini prilikom prelaska pojedinih kotača preko prepreke, pri čemu je najizraženiji skok vidljiv pri prelasku prvog kotača preko prepreke, što je i očekivano pošto dijagram prikazuje brzinu upravo tog kotača. Ponovno su prisutne i oscilacije uzrokovane relativno grubim česticama, ali je njihova amplituda manje izražena.



Slika 17. Komponenta brzine u smjeru osi  $x$  prednjeg desnog kotača

## 6. Zaključak

Ubrzanim razvojem računala tijekom proteklih desetljeća, sve vrste računalnih simulacija, pa tako i simulacije kontaktne dinamike sustava s velikim brojem krutih tijela, su doživjele procvat. Međutim, čak i za današnja računala, simulacije s velikim brojem tijela predstavljaju veoma složen zadatak te je broj tijela, a samim time i točnost dobivenih rezultata, ograničena. No, nastavi li se trend razvoja računala, vrlo brzo će se pojaviti računala koja će omogućiti još bolje i preciznije simulacije.

Na primjeru stvarne konstrukcije testnog rovera pokazan je postupak izrade simulacije, a usporedbom s dostupnim rezultatima eksperimenata može se zaključiti da i uz grube čestice tla korištene u simulaciji, rezultati simulacije kvalitativno vrlo dobro odgovaraju stvarnim rezultatima. U svrhu postizanja još boljih rezultata, moglo bi se koristiti snažnije računalo koje bi omogućilo korištenje većeg broja finijih čestica. Dodatno poboljšanje koje se može napraviti je odbacivanje pretpostavke o uniformnoj raspodjeli gustoće i definiranje gustoće pojedinih dijelova na temelju gustoće materijala od kojeg su napravljeni, ali to bi zahtijevalo detaljan uvid u tehničku dokumentaciju rovera.

## Literatura

- [1] NASA, <https://www.nasa.gov/press-release/nasa-confirms-evidence-that-liquid-water-flows-on-today-s-mars> (pristupljeno 21. 2. 2023.)
- [2] NASA, <https://nasa3d.arc.nasa.gov/detail/scarecrow> (pristupljeno 20. 2. 2023.)
- [3] O. Toupet *et al.*, (2018.) "Traction control design and integration onboard the Mars science laboratory curiosity rover," *2018 IEEE Aerospace Conference*, Big Sky, MT, USA, 2018, str. 1.-20.
- [4] NASA, <https://mars.nasa.gov/resources/20325/traction-control-testing/?site=msl> (pristupljeno 21. 2. 2023.)
- [5] I. Fialho and G. J. Balas, (2002.) "Road adaptive active suspension design using linear parameter-varying gain-scheduling," in *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, str. 43.-54.
- [6] D. B. Tickler, US Pat. Appl. 184,236, 21. 4. 1988.
- [7] National Geographic, <https://www.nationalgeographic.com/science/article/120804-mars-science-laboratory-nasa-space-rover-landing> (pristupljeno 21. 2. 2023.)
- [8] Harrington, B. D., Voorhees, C. (2004.) The Challenges of Designing the Rocker-Bogie Suspension for the Mars Exploration Rover, Proceedings of the 3<sup>rd</sup> Aerospace Mechanisms Symposium, Johnson Space Center, May 19-21, 2004
- [9] NASA, <https://nasa3d.arc.nasa.gov/detail/scarecrow> (pristupljeno 22. 2. 2023.)
- [10] Eiber, A., Terze, Z., (2017.) Dynamics of Multibody Systems. Fakultet strojarstva i brodogradnje
- [11] Modeling Friction and Contact in Chrono, [https://webcache.googleusercontent.com/search?q=cache:xwluLb3L7eMJ:https://www.projectchrono.org/assets/slides\\_3\\_0\\_0/3\\_Contact/1\\_Chrono\\_TheoryContact.pdf&cd=1&hl=hr&ct=clnk&gl=hr](https://webcache.googleusercontent.com/search?q=cache:xwluLb3L7eMJ:https://www.projectchrono.org/assets/slides_3_0_0/3_Contact/1_Chrono_TheoryContact.pdf&cd=1&hl=hr&ct=clnk&gl=hr), (pristupljeno 22. 2. 2023.)
- [12] Tupy, M., (2010.) A Study on the Dynamics of Granular Material with a Comparison of DVI and DEM Approaches, diplomski rad, Univeristy of Wisconsin-Madison
- [13] Melanz, D., Jayakumar, P., Negrut, D., (2016.) Experimental validation of a differential variational inequality-based approach for handling friction and contact in vehicle/granular-terrain interaction, *Journal of terramechanics*, volume 65, str. 1.-13.
- [14] Melanz, D. et al., (2017.) A comparison of numerical methods for solving multibody dynamics problems with frictional contact modeled via differential variational inequalities, *Computer Methods in Applied Mechanics and Engineering*, 9. 3. 2017.
- [15] Großkatthöfer, K., Yoon, Z., (2012.) Introduction into quaternions for spacecraft attitude representation, Technical University of Berlin
- [16] E. J. Haug, *Computer-Aided Kinematics and Dynamics of Mechanical Systems Volume-I*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[17] Tasora, A., Negrut, D., Anitescu, M., (2017.) Gpu-based parallel computing for the simulation of complex multibody systems with unilateral and bilateral constraints: an overview. In *Multibody Dynamics*, pages 283–307. Springer

[18] Tasora, A. et al., (2016.). Chrono: An Open Source Multi-physics Dynamics Engine. 19-49. 10.1007/978-3-319-40361-8\_2.

[19] NASA, <https://solarsystem.nasa.gov/resources/2398/curiosity-3d-model/> (posjećeno 23. 2. 2023.)