Numeričko modeliranje dinamike sustava velikog broja čestica u međusobnom kontaktu na primjeru svemirskog rovera

Horvat, Margareta

Master's thesis / Diplomski rad

2024

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:235:231147

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-27**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Margareta Horvat

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentori:

Izv. prof. dr. sc. Dario Zlatar

Student:

Margareta Horvat

Zagreb, 2024.

Izjavljujem da sam ovaj rad izradila samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, izv. prof. dr. sc. Dariu Zlataru, na stručnoj pomoći i strpljenju pri izradi diplomskog rada.

Zahvaljujem se svojim roditeljima i sestri na podršci u životu i tijekom studiranja te prijateljima koji su bili uvijek uz mene.

Margareta Horvat

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment, mehatronika i robotika, autonomni sustavi i računalna inteligencija

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum 21 -03- 2024	Prilog
Klasa: 602 - 04 / 24 - 06 / 1	
Ur.broj: 15 - 24 - 165	

DIPLOMSKI ZADATAK

Student: **Margareta Horvat** JMBAG: 0035210365

Naslov rada na hrvatskom jeziku: **Numeričko modeliranje dinamike sustava velikog broja čestica u međusobnom kontaktu na primjeru svemirskog rovera**

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

Opis zadatka:

Numeričko modeliranje sustava koji sadrže velik broj čestica u međusobnom kontaktu zahtjevna je zadaća koja tijekom posljednjih godina privlači velik interes istraživača. Tipičan primjer takvog sustava je sustav aktivnog podvozja koji omogućava gibanje po različitim konfiguracijama terena što je važna značajka mobilnih robota namijenjenih istraživanjima površina planeta i drugih nebeskih tijela. Način rada takvog sustava podvozja zasniva se na održavanju jednakih iznosa tereta na svim kotačima vozila, premda je moguće da se neki od kotača nalazi na višoj ili nižoj razini od preostalih.

U tom smislu, u radu je potrebno:

- Istražiti mehanizme podvozja postojećih svemirskih rovera, opisati princip rada odabranog sustava podvozja i objasniti kako je taj sustav instaliran na rover.
- Opisati modele kontaktne dinamike i odabrati najprikladniji model za numeričku simulaciju odabrane konfiguracije podvozja između formulacija metode diskretnih elemenata (engl. discrete elements method, DEM) temeljenim na glatkom kontaktu i diferencijalnoj varijacijskoj nejednakosti (engl. differential variational inequalities, DVI) (neglatka formulacija dinamičkog kontakta).
- Izraditi računalni model odabranog rovera koji uključuje opisani kontaktni model u programskom paketu Project Chrono.
- Usporediti rezultate numeričkih simulacija s eksperimentalnim rezultatima dobivenim mjerenjem na stvarnom roveru, a preuzetih iz znanstvenih radova.

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

Zadatak zadan:	Datum predaje rada:	Predvideni datumi obrane:
18. siječnja 2024.	21. ožujka 2024.	25. – 29. ožujka 2024.

Zadatak zadao:

Izv. prof. dr. sc. Dario Zlatar

Predsjednik Povjerenstva:

Prof. dr. sc. Ivica Garašić

SADRŽAJ

POPIS SLIKA

POPIS OZNAKA

SAŽETAK

SUMMARY

1. Uvod 1

2. Project Chrono 3

3. Curiosity 5

3.1. Scarecrow 7

4. Podvozja i sustavi ovjesa svemirskih rovera 11

4.1. Rocker-bogie sustav 12

4.1.1. Statička analiza 13

4.1.2. Prednosti i mane 15

5. Matematički model 16

5.1. Dinamika čestica 16

5.2. Dinamika krutih tijela 18

5.3. Metoda diskretnih elemenata (DEM) 20

5.4. Diferencijalna varijacijska nejednakost (DVI) 25

5.5. Kontaktni model tla (SCM) 28

6. Numeričko modeliranje 31

6.1. Izrada CAD modela rovera 31

6.2. Izrada simulacije temeljenoj na diferencijalnoj varijacijskoj nejednakosti 32

6.2.1. Obrada rezultata 37

6.3. Izrada simulacije temeljena na kontaktnom modelu tla 40

6.3.1. Rezultati i obrada rezultata 47

6.4. Usporedba rezultata simulacija i postojećeg eksperimenta 47

6.4.1. Usporedba rezultata simulacije Scarecrow-a i eksperimenta 47

6.4.2. Usporedba rezultata simulacije Scarecrow-a i Curiosity rovera 51

7. Zaključak 56

Literatura 58

POPIS SLIKA

Slika 1. Spuštanje Perseverance rovera na Mars 23 lipnja 2021. godine [1] 1
Slika 2. Primjer simulacije u programskom paketu Project Chrono [2] 3
Slika 3. Struktura [2] 4
Slika 4. Curiosity [5] 5
Slika 5. Kotač [5] 6
Slika 6. Prikaz tijela, sustava ovjesa te kotača [4] 7
Slika 7. Scarecrow [5] 8
Slika 8. Oštećenja kotača [7] 9
Slika 9. Prelazak kotača preko aluminijske prepreke [8] 10
Slika 10. Primjer aktivnog sustava ovjesa – Sherpa [18] 11
Slika 11. Spuštanje rovera na Mars [5] 12
Slika 12. Rocker-Bogie sustav ovjesa [9] 13
Slika 13. Pojednostavljeni mehanizam [16] 14
Slika 14. Gibanje čestica [14] 16
Slika 15. Sustav čestica [14] 17
Slika 16. Kruto tijelo [14] 18
Slika 17. Koraci diskretne metode elemenata 21
Slika 18. Pozicija čestica[15] 21
Slika 19. Kontaktne sile između dva tijela [15] 22
Slika 20. Dva tijela u kontaktu [16] 27
Slika 21. Kretanje cilindričnog tijela po površini 29
Slika 22. CAD model 31
Slika 23. Primjer markera 32
Slika 24. Vizualni prikaz simulacije 37
Slika 25. Dijagram rezultantnih sila (DVI) 39
Slika 26. Vizualni prikaz simulacije 47
Slika 27. Dijagram rezultantnih sila dobivenih simulacijom Scarecrow-a 48
Slika 28. Rezultantne sile dobivene eksperimentom [7] 48
Slika 29. Simulacija kretanja Scarecrow-a na ravnom terenu 50
Slika 30. Prijelaz srednjeg kotača preko prepreke 51
Slika 31. Curiosity rover u simulaciji 52
Slika 32. Prelazak Curiosity rovera preko prepreke 52

Slika 33. Dijagram rezultantnih sila dobivenih simulacijom Curiosity rovera 53

Slika 34. Dijagram brzina Scarecrow-a 54

Slika 35. Dijagram brzina za Curiosity rover 54

POPIS OZNAKA

Oznaka	Jedinica	Opis
N_1	N	Reakcijska sila između kotača i tla
N_2	N	Reakcijska sila između kotača i tla
N_3	N	Reakcijska sila između kotača i tla
θ_1	°	Kut omeđen zadanom površinom i mehaničkom spojnicom mehanizma
θ_2	°	Kut omeđen zadanom površinom i mehaničkom spojnicom mehanizma
θ_3	°	Kut omeđen mehaničkom spojnicom između dva dijela mehanizma i zadane površine
θ_4	°	Kut omeđen zadanom površinom i mehaničkom spojnicom mehanizma
φ_1	°	Kut omeđeni mehaničkim spojnicama dva dijela mehanizma
φ_2	°	Kut omeđeni mehaničkim spojnicama dva dijela mehanizma
r_1	m	Radijus stražnjeg kotača
r_2	m	Radijus srednjeg kotača
r_3	m	Radijus prednjeg kotača
h_1	m	Visina kotača u odnosu na zadani zglob
h_2	m	Visina kotača u odnosu na zadani zglob
h_3	m	Visina kotača u odnosu na zadani zglob
l_1	m	Duljina mehaničke spojnice zgloba i stražnjeg kotača
l_2	m	Duljina mehaničke spojnice zgloba i srednjeg kotača
l_3	m	Duljina mehaničke spojnice između dva zgloba
l_4	m	Duljina mehaničke spojnice zgloba i prednjeg kotača
m	kg	Masa cijelog tijela
mv	$\frac{\text{kgm}}{\text{s}}$	Količina gibanja čestica
r		Vektor položaja čestice u inercijskom koordinatnom sustavu
v	$\frac{\text{m}}{\text{s}}$	Brzina čestica
a	$\frac{\text{m}}{\text{s}^2}$	Ubrzanje čestica

h_O	$\frac{\mathrm{Nm}^2}{\mathrm{s}}$	Kinetički moment čestice u odnosu na zadano ishodište O
f_i	N	Rezultantna sila i-te čestice
f_i^e	N	Rezultantna sila svih vanjskih sila koje djeluje na i-tu česticu
f_i^i	N	Rezultantna sila svih unutarnjih sila koje djeluje na i-tu česticu
m_i	kg	Masa i-te čestice
a_i	$\frac{\mathrm{m}}{\mathrm{s}^2}$	Ubrzanje i-te čestice
r_c		Vektor položaja mase sustava čestica
v_c	$\frac{\mathrm{m}}{\mathrm{s}}$	Brzina mase sustava čestica
a_c	$\frac{\mathrm{m}}{\mathrm{s}^2}$	Ubrzanje sustava čestica
r'		Vektor položaja tijela
ω	s^{-1}	Kutna brzina tijela
$\dot{r}'$		Vremenska derivacija vektora položaja tijela
r_A		Vektor položaja tijela A
v_A		Brzina tijela A
ϕ, ϑ, ψ	°	Eulerovi kutovi
$\dot{x}_R$		Derivacija Eulerovi kutova
r_{AC}		Vektor položaja točke C u odnosu na točku A
r_C'		Vektor položaja mase u koordinatnom sustavu vezanom za točku A
I_A		Tenzor inercije tijela u odnosu na točku A
E		Jedinični vektor
I_C		Tenzor inercije tijela u odnosu na točku C
l_O		Rezultantni moment sile u odnosu na točku O
r_{ij}		Vektor relativnog položaj između dvije čestice
r_i		Vektor položaja i-te čestice
r_j		Vektor položaja j-te čestice
d	m	Promjer kugle
$F_{n_{ij}}$	N	Ukupna sila u sustavu
F_{el}	N	Elastična sila

F_{diss}	N	Sila u prigušivaču
k_n		Koeficijent krutosti opruge
γ_n		Koeficijent prigušenja
k_{load}		Koeficijent krutosti punjenja
k_{unload}		Koeficijent krutosti pražnjenja
δ_{ij}	m	Preklapanje i-te i j-te čestice
$v_{n_{ij}}$	$\frac{\text{m}}{\text{s}}$	Relativna normalna brzina sustava
m_i	kg	Masa i-te čestice
m_j	kg	Masa j-te čestice
m_{eff}	kg	Efektivna masa
$v_{t_{ij}}$	$\frac{\text{m}}{\text{s}}$	Relativna tangencijalna brzina sustava
ω_i	°	Rotacija i-te čestice
ω_j	°	Rotacija j-te čestice
$u_{t_{ij}}$		Tangencijalni pomak
$F_{t_{ij}}$	N	Tangencijalna sila sustava
F_i^{tot}	N	Ukupna sila koja djeluje na i-tu česticu
M_i^{tot}	Nm	Moment oko središta mase i-te čestice
Δt	s	Korak integracije
n_b		Broj krutih tijela
ϵ_j		Kvaternion rotacije j-tog tijela
$L(q)v$		Linearno preslikavanje
$\bar{\omega}_j$	$\frac{\text{rad}}{\text{s}}$	Kutna brzina j-tog tijela
$\Psi_i(q,t)$		Funkcija skalarnih jednadžbi ograničenja
$\Phi_i(q)$		Funkcija razmaka tijela u sustavu
$f(t,q,v)$		Funkcija generalizirane sile u sustavu
$\hat{\gamma}_{i,b}$		Lagrangeov multiplikator (množitelj)
$\bar{s}_{i,A}$		Kontaktna točka u odnosu na referentni okvir tijela A
$\bar{s}_{i,B}$		Kontaktna točka u odnosu na referentni okvir tijela B
n_i		Globalni jedinični vektor

u_i, w_i		Jedinični vektori koji premošćuju kontaktnu ravninu u točki dodira
$F_{i,N}$	N	Normalna sila u dodiru dva tijela
$F_{i,t}$	N	Tangencijalna sila u dodiru dva tijela
$\hat{\gamma}_{i,n}, \hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}$		Iznos sila u svakom smjeru
$\mu_{i,d}$		Dinamički faktor trenja
$\mu_{i,s}$		Statički faktor trenja
σ	Pa	Tlak kontaktne površine
y		Potonuće kotača
k_c	$\frac{\text{kN}}{\text{m}^{n+1}}$	Empirijski koeficijent kohezivnog učinka tla
k_ϕ	$\frac{\text{kN}}{\text{m}^{n+2}}$	Empirijski koeficijent krutosti tla
n		Eksponent potonuća
b	m	Duljina kraće stranice pravokutne kontaktne površine
y_h	m	Udaljenost oblika
y_r	m	Početna visina tla
s_y	m	Deformacija na tlo
σ_{try}	Pa	Testni tlak
s_{sink}	m	Ukupna deformacija
s_p	m	Plastična deformacija
s_e	m	Elastična deformacija
σ_{tot}	Pa	Viskozni učinak prigušenja

SAŽETAK

U diplomskom radu opisan je programski paket Project Chrono koji se koristio pri izradi računalnog modela odabranog rovera te je opisan Curiosity rover na temelju kojega je u zadnjem poglavlju izrađena numerička simulacija. Rezultati simulacije uspoređeni su s eksperimentalnim rezultatima dobivenim mjerenjem na stvarnom roveru. U sljedećem poglavlju opisane su osnovne značajke sustave ovjesa koje današnji svemirski roveri koriste te je detaljno opisan najzastupljeniji mehanizam Rocker-bogie. Provedena je statička analiza kako bi bolje shvatili što je sve potrebno za izradu jednog takvog mehanizma te su opisane prednosti i mane njegova korištenja. Također, ovaj rad obuhvaća analizu dinamike čestica i dinamiku krutih tijela te su opisani modeli kontaktne dinamike, odnosno formulacije metode diskretnih elemenata temeljenim na glatkom kontaktu, diferencijalna varijacijska nejednakost te kontaktni model tla.

Ključne riječi: Project Chrono, Curiosity, Rocker-bogie mehanizam, dinamika čestica, dinamika krutih tijela, modeli kontaktne dinamike, metoda diskretnih elemenata, diferencijalna varijacijska nejednakost, kontaktni model tla, numerička simulacija

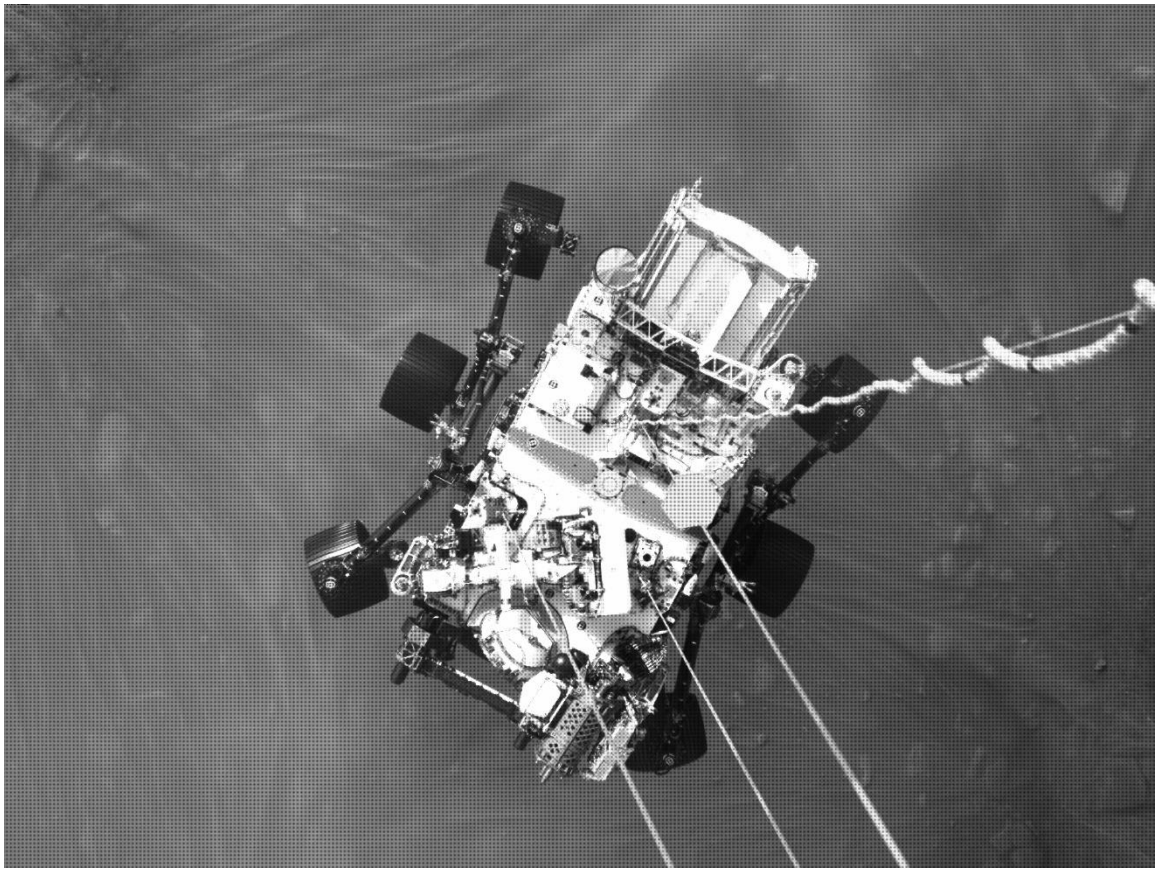
SUMMARY

The thesis describes the Project Chrono software package used in the development of a computer model of a selected rover. Additionally, it provides an overview of the Curiosity rover, based on which a numerical simulation was created in the final chapter. The simulation results were compared with experimental data obtained from measurements on the actual rover. The following chapter describes the basic features of suspension systems used in today's space rovers, with detailed insights into the most prevalent mechanism, the Rocker-bogie. A static analysis was conducted to better understand the requirements for constructing such a mechanism, including its advantages and disadvantages. Furthermore, the thesis encompasses an analysis of particle dynamics and rigid body dynamics, with descriptions of contact dynamics models, formulations of the discrete element method based on smooth contact, differential variational inequalities and soil contact model.

Keywords: Project Chrono, Curiosity, Rocker-bogie mechanism, particle dynamics, rigid body dynamics, contact dynamics models, discrete element method, differential variational inequalities, soil contact model numerical simulation

1. Uvod

U posljednjim desetljećima svemirska istraživanja postigla su nevjerojatne uspjehe, a jedna od najambicioznijih misija fokusirana je na crveni planet - Mars. Svojom fascinantnom atmosferom i potencijalom za otkrivanje tragova života, Mars je postao centar istraživačkih misija svemirskih agencija diljem svijeta. Misije na Marsu imaju duboko ukorijenjene znanstvene, tehnološke i istraživačke ciljeve, usmjerene na pronalaženje odgovora na ključna pitanja o podrijetlu i mogućem postojanju života izvan Zemlje. Opremljeni najnovijom tehnologijom, robotski roveri poput Curiosity i Perseverance (slika 1) pružaju nam jedinstvenu priliku za proučavanje površine Marsa i potragu za znakovima biološke aktivnosti. Svaka misija predstavlja korak naprijed prema eventualnom ljudskom naseljavanju ovog planeta, otvarajući vrata ne samo znanstvenim otkrićima, već i potencijalnim budućim avanturama čovječanstva izvan granica Zemlje. Ovo istraživanje često predstavlja suradnju međunarodnih svemirskih agencija, a svaka uspješna misija pridonosi dubljoj slici o Marsu kao ključnom elementu u razumijevanju evolucije Sunčevog sustava i samog života.



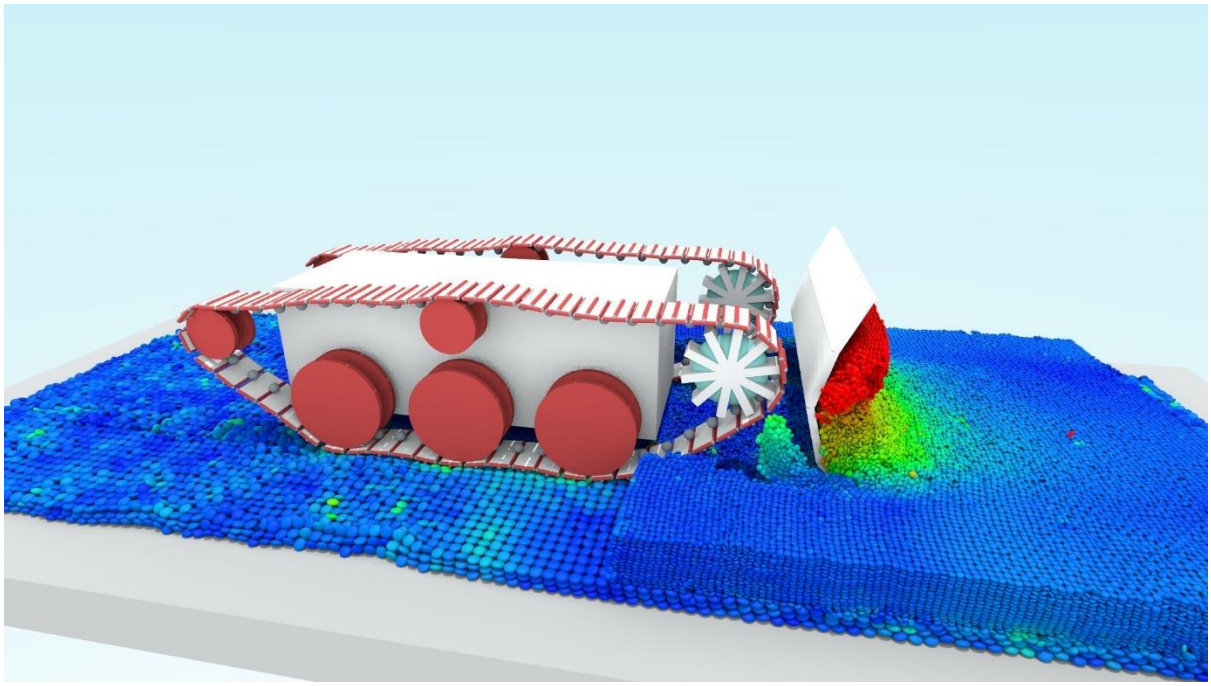
Slika 1. Spuštanje Perseverance rovera na Mars 23 lipnja 2021. godine [1]

Rocker-bogie mehanizam predstavlja ključnu komponentu u dizajnu ovjesa svemirskih rovera. Ova inovativna tehnologija omogućuje vozilu da premosti prepreke i nepravilnosti terena na nepristupačnim površinama, osiguravajući stabilnost i pokretljivost. Sustavi ovjesa za svemirske rovere su od vitalnog značaja za uspješnu misiju istraživanja, pružajući potrebnu podršku i zaštitu tijekom kretanja po nepoznatim terenima. Važnost eksperimentalnih istraživanja pomoću numeričkog modeliranja leži u njihovoj sposobnosti da simuliraju različite scenarije i uvjete s visokom preciznošću, pružajući uvid u performanse i ponašanje rovera u različitim okruženjima, te omogućujući identifikaciju potencijalnih izazova i rješenja prije stvarne primjene u svemirskim misijama.

Cilj ovog rada je detaljno opisati različite sustave ovjesa i posebno se usredotočiti na sustav koji se primjenjuje na odabranom modelu rovera. Nadalje, bit će pružen opsežan pregled u model kontaktne dinamike, odnosno dinamiku čestica i dinamiku krutih tijela te pristupima pod nazivom metoda diskretnih elemenata, diferencijalna varijacijska nejednakost te kontaktni model tla. Na završetku, izradit će se numerička simulacija eksperimenta korištenjem programskog paketa otvorenog koda Project Chrono, čiji će rezultati biti uspoređeni s eksperimentalnim rezultatima kako bi se analizirala njihova točnost i primjenjivost.

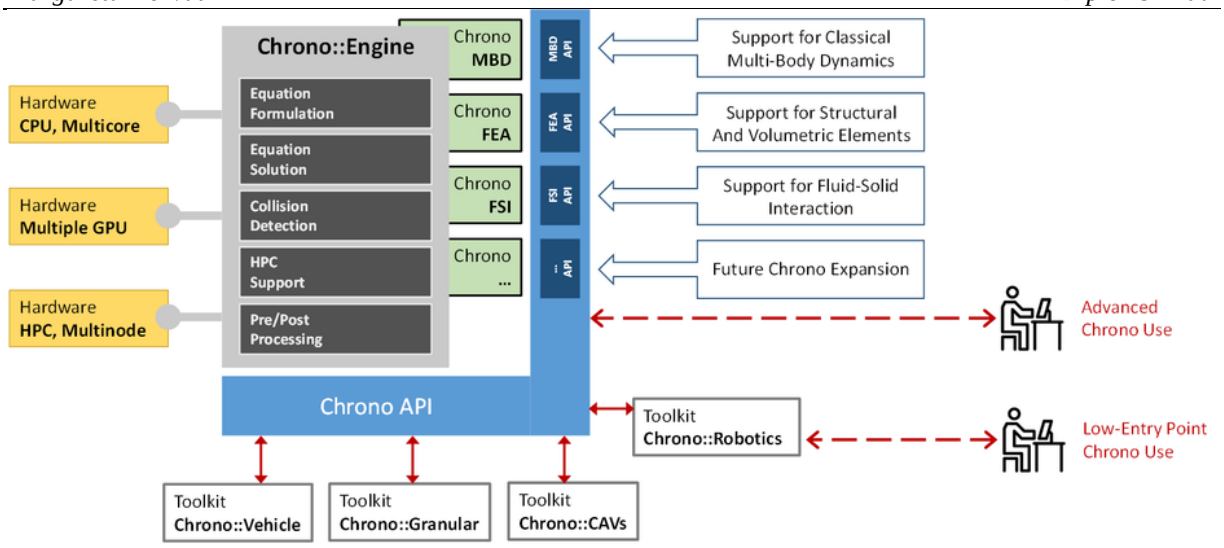
2. Project Chrono

Project Chrono je infrastruktura za modeliranje i simulaciju koja se temelji na fizici i temelji se na dizajnu otvorenog koda (engl. open-source) neovisnog o platformi implementiranom u C++. Programska biblioteka PROJECTCHRONO može se ugraditi u softverski projekt za simulaciju, na primjer, vozila s kotačima i gusjenicama koja rade na deformabilnim terenima, robota, mehatroničkih sustava i interakcije fluida i strukture (FSI). Sustavi mogu biti izrađeni od krutih i fleksibilnih/podložnih dijelova s ograničenjima, motorima i kontaktima; dijelovi mogu imati trodimenzionalne oblike za detekciju sudara. Postoji i Python verzija Chrona, PyChrono. Project Chrono može se koristi na sljedećim područjima: robotika, dinamika vozila na kotačima, dinamika gusjeničnog vozila, nelinearna analiza konačnih elemenata, mehatronika, mobilnost terenskih vozila, teramehanika, virtualna stvarnost, tok zrnatih materijala, otkrivanje sudara, autonomna vozila, seizmičko inženjerstvo te proširena stvarnost. Jedna od primjena prikazana je na slici 2.



Slika 2. Primjer simulacije u programskom paketu Project Chrono [2]

Project Chrono može se koristiti na raznim operativnim sustavima te se sastoji od glavnog i dodatnih modula. Neki od dodatnih modula su: FEA modul, FSI modul, IRRLICHT modul, MATLAB modul, VEHICLE modul itd. Primjer strukture može se vidjeti na slici 3.

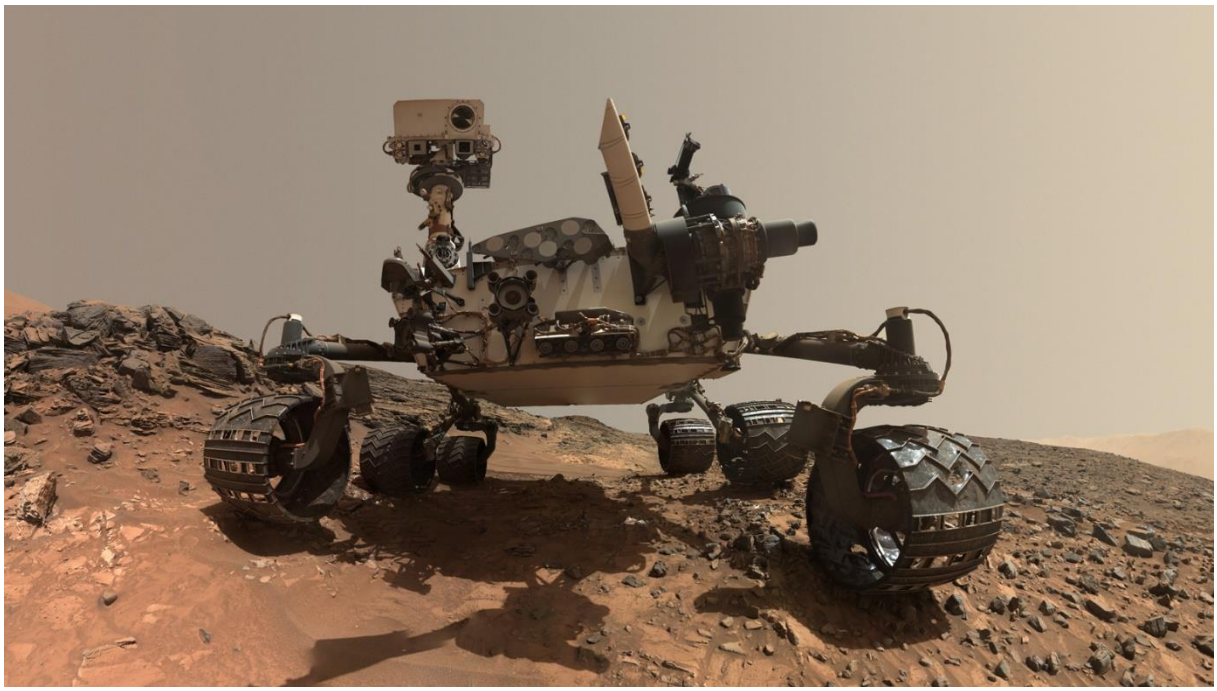


Slika 3. Struktura [2]

Prvu verziju pod nazivom Chrono::Engine razvio je 1998. godine prof. Alessandro Tasora dok je bio student na Politehničkom sveučilištu u Milanu (Politecnico di Milano). Bio je to rezultat diplomskog rada iz strojarstva. Izvorno, Chrono::Engine je trebao biti alat za simulaciju više tijela za aplikacije robotike i biomehanike. Do 2002. Chrono::Engine bio je usko povezan s Realsoft3D, programom za modeliranje. Od 2002. do 2005. softver je postupno reorganiziran u obliku samostalne programske knjižnice, baš kao što je to danas. Alessandro je 2005. godine počeo raditi s prof. Mihaijem Anitescuom sa Sveučilišta u Chicagu i Nacionalnim laboratorijem Argonne (Argonne National Laboratories). Prof. Dan Negrut pridružio se Project Chrono 2007. godine. Članovi Laboratorija za inženjerske simulacije (SBEL) na Sveučilištu Wisconsin-Madison od tada su aktivni u njegovom razvoju. Dr. Radu Serban počeo je raditi na Project Chrono 2013., u vrijeme kada se pridružio Sveučilištu Wisconsin-Madisonu i Laboratoriju za inženjerske simulacije. Project Chrono 2013. godine postaje softver otvorenog koda, kada se počelo koristiti naziv PROJECTCHRONO kako bi se prepoznala činjenica da je ova softverska infrastruktura postala multi-fizički simulacijski motor. Trenutna verzija programske biblioteke Project Chrono je 8.0.0 i objavljena je u studenom 2021. [2].

3. Curiosity

Curiosity je najveći i najsposobniji rover ikada poslan na Mars (slika 4). Lansiran je 26. studenog 2011. i sletio je na Mars 6. kolovoza 2012. Cilj misije je određivanje je li Mars ikada mogao podržavati život, kao i određivanje uloge vode, te proučavanje klime i geologije Marsa. Rezultati misije također će pomoći u pripremi za ljudsko istraživanje. Lansiran je na raketi Atlas V-541, mase oko 531 t i visine 58 m s Cape Canaveral Space Force Station-a (CCSFS) Svemirskih snaga Sjedinjenih Država Space Launch Delta 45, u okrugu Brevard, Florida. Curiosity je 2,9 m dugačak, 2,7 m širok, 2,2 m visok te mu je masa 899 kg.



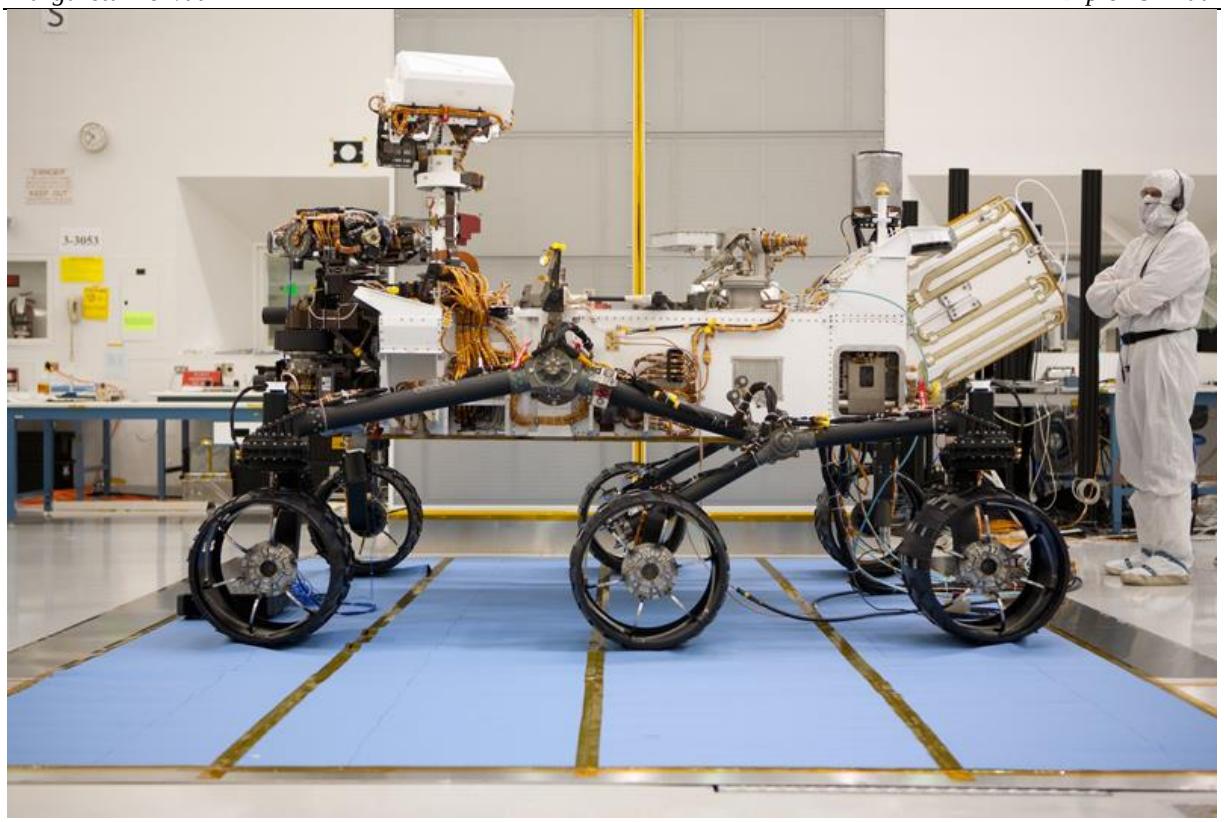
Slika 4. Curiosity [5]

Tijelo rovera naziva se toplom elektroničkom kutijom (engl. Warm Electronics Box - WEB). Poput karoserije automobila, karoserija rovera je jak, vanjski sloj koji štiti roverovo računalo i elektroniku (koji su u osnovi ekvivalent roverovog mozga i srca). Tijelo rovera tako čuva vitalne organe rovera zaštićenima i kontrolira temperaturu. Dno i stranice su okvir šasije, a vrh je paluba opreme rovera (njegova "leđa"). Kutija s toplom elektronikom zatvorena je s gornje strane dijelom koji se zove platforma opreme rovera (engl. Rover Equipment Deck - RED). Platforma opreme rovera čini rover poput kabrioleta, dopuštajući mjesto za jarbol rovera i kamere da sjede u marsovskom zraku, snimaju slike i jasno promatraju marsovski teren dok putuje.

Curiosity ima šest kotača, svaki sa svojim motorom. Dva prednja i dva stražnja kotača također imaju zasebne motore za upravljanje (svaki po jedan). Ova mogućnost upravljanja omogućuje okretanje vozila na mjestu, punih 360°. Upravljanje na četiri kotača također omogućuje roveru da zaokreće i zakrivljuje, praveći lučne zavoje. Kotači su izrađeni od aluminija, s klinovima za vuču i zakrivljenim oprugama od titana za opružnu potporu te im je promjer 0,5 m. Kotač je prikazan na slici 5. Klinovi omogućuju prianjanje za penjanje po mekom pijesku i penjanje po stijenama. Jedan puni okret kotača bez proklizavanja je oko 157 cm vožnje. Također, imaju mogućnost kotrljanja po kamenju veličine jednog od svojih kotača, što mu omogućuje sustav ovjesa.

Slika 5. Kotač [5]

Dizajn sustava ovjesa za kotače temelji se na nasljeđu sustava "rocker-bogie" na misijama Pathfinder i Mars Exploration Rover. Sustav ovjesa je način na koji su kotači povezani i međusobno djeluju s tijelom rovera. Rocker-bogie sustav ovjesa biti će detaljnije opisan u sljedećem poglavlju. Rover je dizajniran da izdrži nagib od 45° u bilo kojem smjeru bez prevrtanja. Međutim, rover je programiran kroz svoja "ograničenja zaštite od grešaka" u svom softveru za izbjegavanje opasnosti kako bi izbjegao prekoračenje nagiba od 30° tijekom svojih kretanja.



Slika 6. Prikaz tijela, sustava ovjesa te kotača [4]

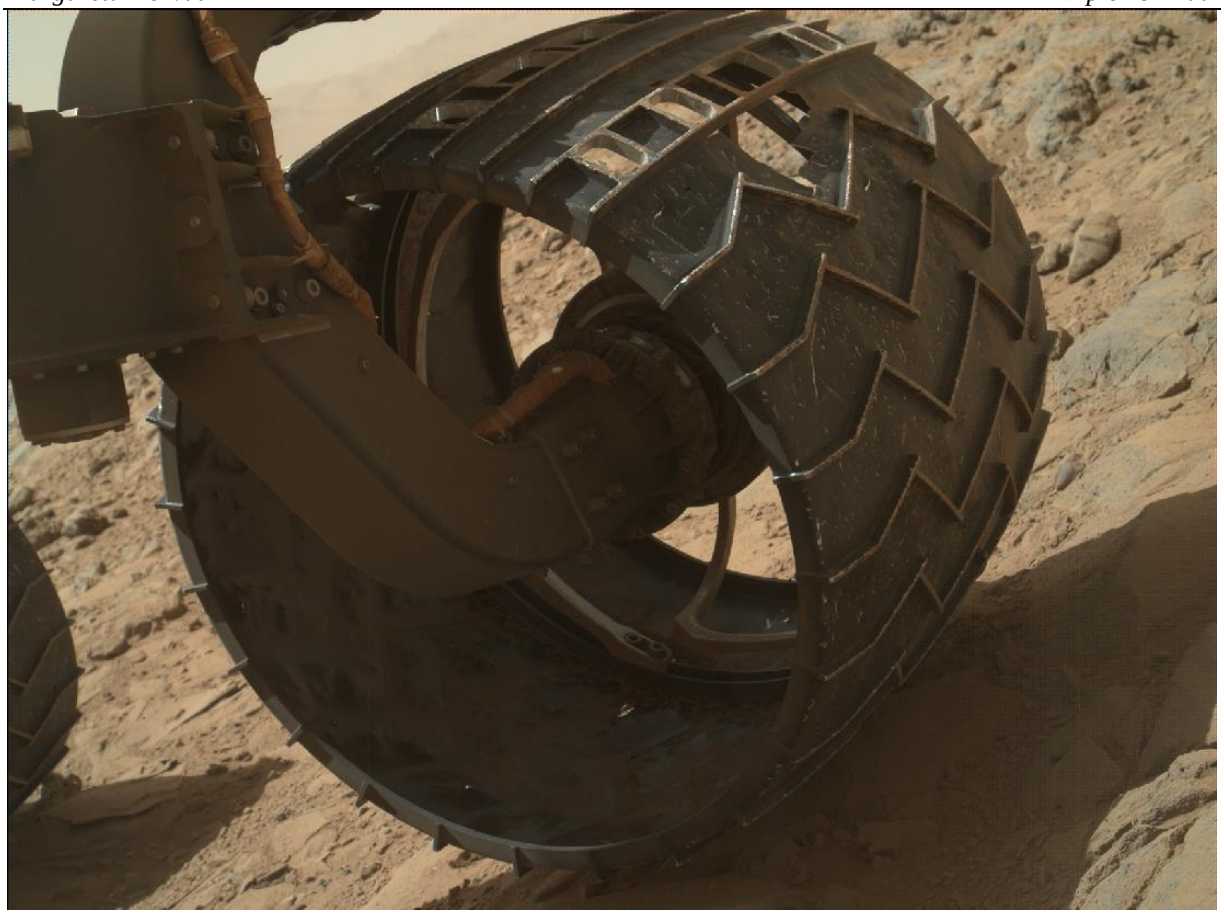
Curiosity ima naprednu znanstvenu opremu na Marsu. To je četvrti NASA-in robotski rover poslan na Mars od 1996. Prethodni uspješni marsovski roveri su Sojourner iz misije Mars Pathfinder (1997.), te roveri Spirit (2004. – 2010.) i Opportunity (2004. – 2018.) iz misije Mars Exploration Rover. U trenutku pisanja ovoga rada, Curiosity je prešao preko 31 km te je na Marsu 4125 solarnih dana [3,4].

3.1. Scarecrow

Testni rover (slika 7), nazvan Scarecrow jer nema "mozak" ugrađenog računala kao Curiosity, ima punu verziju Curiosityjevog sustava mobilnosti, ali je ogoljen tako da na Zemlji teži otprilike isto koliko će težiti Curiosity u slabijoj gravitaciji Marsa. To je omogućilo NASA-i da radi stvari poput testiranja sposobnosti rovera da prijeđe preko pješčane dine bez ugrožavanja skupog računala u tom procesu. Također, Scarecrow može informirati kako se Perseverance kreće na Marsu, budući da je Perseveranceovo tijelo bilo temeljeno na Curiosityjevom.

Slika 7. Scarecrow [5]

U listopadu 2013. slike snimljene Mars Hand Lens Imager (MAHLI) Mars Science Laboratory (MSL) Curiosity Rovera otkrile su da su oštećenja na kotačima rovera napredovala neočekivano velikom brzinom. Oštećenja mogu se vidjeti na slici 8. Iako se očekivalo oštećenje kotača tijekom misije rovera, probušeni kotači nisu bili nalik na pukotine koncentracije naprezanja viđene nakon opsežne vožnje s velikim opterećenjem tijekom testiranja na zemlji.



Slika 8. Oštećenja kotača [7]

Nakon detaljnog istraživanja uzroka oštećenja kotača započet je projekt s nastojanjem da se smanji daljnje oštećenje kotača mijenjanjem načina na koji vozilo prelazi preko prepreka. Jedan od provedenih testiranja bio je prelazak preko aluminijske polusfere radijusa 15 cm (slika 9). Rezultati ovog testiranja poslužiti će kao podloga za usporedbu s dobivenim rezultatima numeričke simulacije.

Slika 9. Prelazak kotača preko aluminijske prepreke [8]

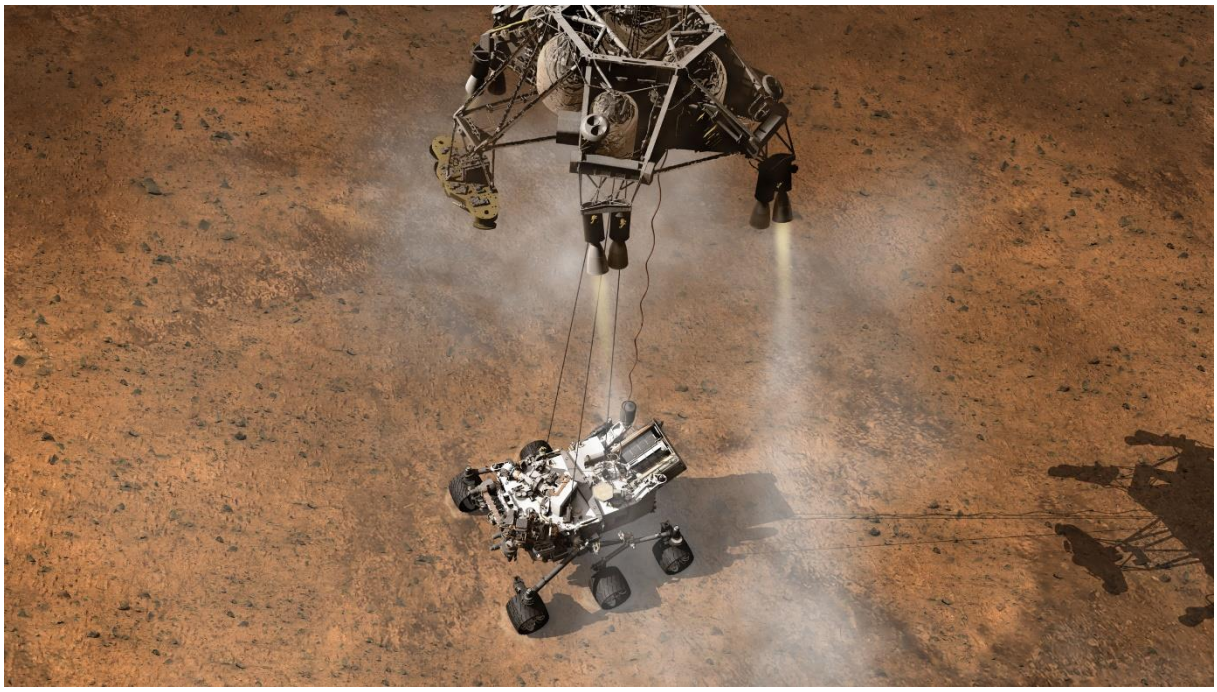
4. Podvozja i sustavi ovjesa svemirskih rovera

Autonomni roboti za istraživanje izvan Zemlje zahtijevaju učinkovite sustave kretanja. Aktivni sustavi suspenzije, poput Rocker-bogie sustava koji se koristi na Mars roverima, pružaju visoku pokretljivost uz nisku složenost upravljanja i rekonfiguracijske mogućnosti koje nisu dostupne pasivnim sustavima. Aktuatori sustava mogu aktivno sudjelovati u pokretima, povećavajući vuču na tlu ili omogućujući undulirajuće sposobnosti kretanja. Ovi sustavi nisu korisni samo za neravan teren, već pružaju stabilnost u unutarnjim okolinama, posebno kod manipulacije teškim teretima ili prolaska kroz uske prolaze. Primjeri poput Sample Return Rovera (SRR), Scarab-a i Sherpa rovera (slika 10) pokazuju kako aktivno upravljanje sustavom suspenzije povećava stabilnost na padinama i omogućuje različite načine kretanja te sudjeluje u stabilizaciji tijela za specifične zadatke poput bušenja podzemnih slojeva [9].

Slika 10. Primjer aktivnog sustava ovjesa – Sherpa [18]

Korištenje pasivnih sustava i opružnih elemenata nije preporučljivo zbog nepovoljne okoline, ograničene izdržljivosti i kompleksnosti u dizajnu. Nedostatak pasivnih sustava postaje očit u situacijama poput zapinjanja u mekim tlima, gdje je izvlačenje vozila teško bez vanjske intervencije. Opruge se mogu brzo oštetiti na neravnom terenu, što može rezultirati čestim kvarovima. Također, postizanje pravilne ravnoteže između čvrstoće, izdržljivosti i težine s

oprugama može biti izazovno. Opruge pružaju pasivno oslanjanje i ne pružaju dovoljno kontrole nad kretanjem i stabilnošću rovera, posebno na nepristupačnom terenu. Naposljetku, kada bi se koristili opružni elementi u sustavu, prilikom spuštanja rovera postojala bi mogućnost sabijanja opruge što bi rezultiralo udarcem tijela rovera u tlo. Jedan od primjera pasivnog sustava je sustav u kojemu su kotači fiksirani na tijelo rovera, što znači da ne posjeduju fleksibilnost i prilagođavanje na neravne terene.



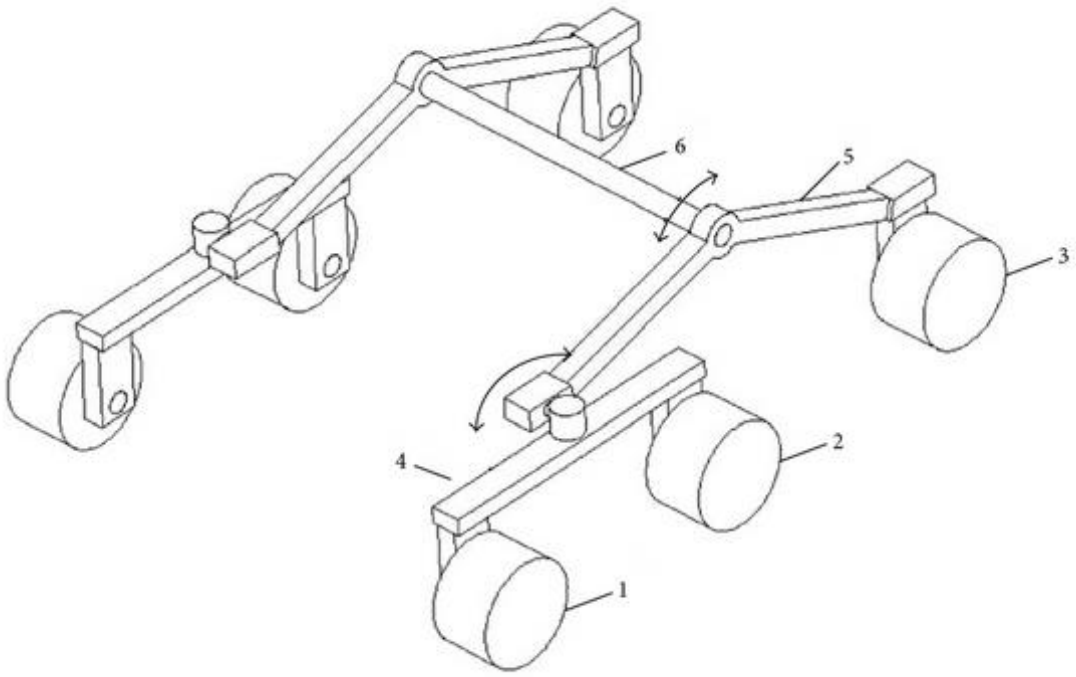
Slika 11. Spuštanje rovera na Mars [5]

4.1. Rocker-bogie sustav

Tijekom proteklog desetljeća, Rocker-bogie sustav ovjesa postao je dokazana primjena za mobilnost te je poznat po svojoj izvrsnoj stabilnosti vozila i sposobnosti penjanja preko prepreka. Prvi put ga je predstavio Donald B. Bickler 1989. te ga zajedno s NASA-om patentirao. Nakon nekoliko implementacija tehnologije istraživačkih rovera, sustav je uspješno korišten kao dio rovera Sojourner u sklopu misije Mars Pathfinder. Rocker-bogie ovjes je mehanizam koji, zajedno s diferencijalom, omogućuje vozilu sa šest kotača da pasivno održava svih šest kotača u kontaktu s površinom čak i pri vožnji po vrlo neravnom terenu. Izraz "rocker" opisuje aspekt ljuljanja većih karika prisutnih sa svake strane sustava ovjesa. Ove klackalice su povezane jedna s drugom i sa šasijom vozila preko diferencijala. U odnosu na šasiju, kada se jedna klackalica podigne, druga se spusti. Šasija održava prosječni kut nagiba obje klackalice. Jedan kraj klackalice opremljen je pogonskim kotačem, a drugi kraj zakrenut je na okretnom

postolju. U sustavu, "bogie" odnosi se na spojne karike koje imaju pogonski kotač pričvršćen na svakom kraju.

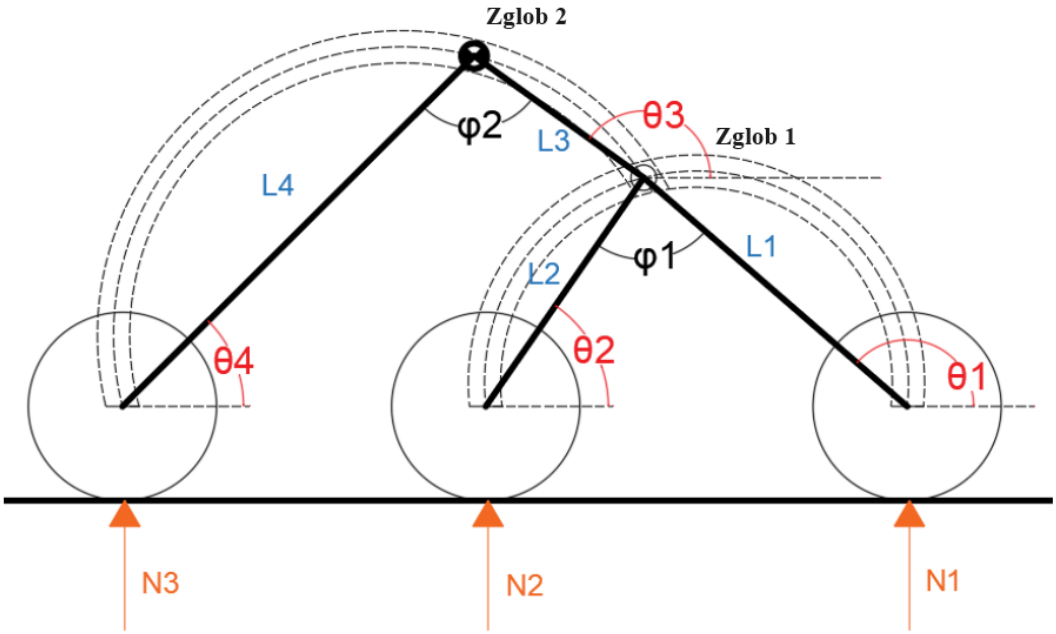
Roboti koji koriste mehanizam klackalice koriste mehanizam ovjesa koji se sastoji od nekoliko krutih elemenata povezanih zglobovima određenog broja stupnjeva slobode. Ovjes ima 6 kotača sa simetričnom strukturom s obje strane. Svaka strana ima 3 kotača koji su međusobno povezani s dvije karike. Glavni spoj koji se naziva rocker, ima 2 zgloba. dok je prvi zglob povezan s prednjim kotačem, drugi spoj je spojen na drugu vezu zvanu bogie, koje je slično ovjesnom članu vagona vlaka. Na slici 12 brojem 1, 2 i 3 označeni su kotači, brojem 4 bogie, brojem 5 rocker te brojem 6 diferencijal [10,11].



Slika 12. Rocker-Bogie sustav ovjesa [9]

4.1.1. Statička analiza

Za rover Rocker-bogie mehanizam je neophodan kako bi imao visoku mobilnost na neravnom terenu. Stoga je potrebno podijeliti težinu robota jednako za svaki motor. Kotač koji nosi najveći dio težine trebao bi više snage od ostalih za kretanje, što znači da bi svaki motor morao biti veći i teži. Stoga je ravnomjerna raspodjela mase rovera između kotača ključna za konstrukciju. Slika 13 prikazuje jednostavno modeliranje Rocker-bogie mehanizma.



Slika 13. Pojednostavljeni mehanizam [16]

Svaki kotač preuzima reakcijsku silu od tla te je zbog pojednostavljenja uzete su samo normalne sile N_1, N_2 i N_3. Osim toga, pretpostavlja se da je površina glatka kako bi sila trenja bila jednaka nuli. Na temelju slike 10 može se zaključiti:

$$\theta_1 = \varphi_1 + \theta_2 \tag{4.1}$$

$$\theta_3 = \varphi_2 + \theta_4 \tag{4.2}$$

Primjernom trigonometrijskih pravila dobiju se sljedeći izrazi:

$$\theta_1 = \sin^{-1}\frac{r_2 - r_1 + h_2 - h_1}{\sqrt{(l_1 - l_2 cos\varphi_1)^2 + (l_2 sin\varphi_1)^2}} - \mathrm{tg}^{-1}\frac{l_2 sin\varphi_1}{l_1 - l_2 cos\varphi_1} \tag{4.3}$$

$$\theta_3 = \sin^{-1}\frac{r_3 - r_1 + h_3 - h_1 - l_1 sin\theta_1}{\sqrt{(l_3 - l_4 cos\varphi_2)^2 + (l_4 sin\varphi_2)^2}} - \mathrm{tg}^{-1}\frac{l_4 sin\varphi_2}{l_3 - l_4 cos\varphi_2} \tag{4.4}$$

gdje su r_1, r_2 i r_3 radijusi pojedinih kotača, h_1, h_2 i h_3 visine u odnosu na zglobove, l_1, l_2, l_3 i l_4 duljine nogu između kotača i zglobova te dva zgloba, odnosno između rocker i bogie dijela mehanizma, θ_1, θ_2 i θ_4 kutovi omeđeni zadanom površinom i mehaničkom spojnicom mehanizma, θ_3 kut između mehaničke spojnice koja spaja rocker i bogie dio te zadane površine te φ_1 i φ_2 su kutovi omeđeni mehaničkim spojnicama rocker i bogie dijela mehanizma.

Reakcijske sile u kotačima moguće je izračunati prema sljedećim izrazima:

$$N_1 = \frac{mgl_2l_4cos\theta_2cos\theta_4}{2(l_1cos\theta_1 - l_2cos\theta_2)(l_3cos\theta_3 - l_4cos\theta_4)} \tag{4.5}$$

$$N_2 = \frac{-l_1cos\theta_1}{l_2cos\theta_2}N_1 \tag{4.6}$$

$$N_3 = \frac{mg}{2} + \frac{l_1cos\theta_1 - l_2cos\theta_2}{l_2cos\theta_2}N_1\,. \tag{4.7}$$

Kako bi rover što uspješnije prošao neravan teren, centar mase cijelog tijela ne smije se nalazi prenisko ili biti postavljeno daleko iza. Iz toga proizlazi da duljina l_3 ne smije biti predugačka [16].

4.1.2. Prednosti i mane

Prva prednost je da će pritisak kotača na tlu biti izjednačen. To je izuzetno važno na mekom terenu gdje pretjerani pritisak može rezultirati uranjanjem vozila u površinu. Druga prednost je da će, dok se penje po tvrdom i neravnom terenu, sva šest kotača nominalno ostati u kontaktu s površinom i pod opterećenjem, pomažući guranju vozila preko terena.

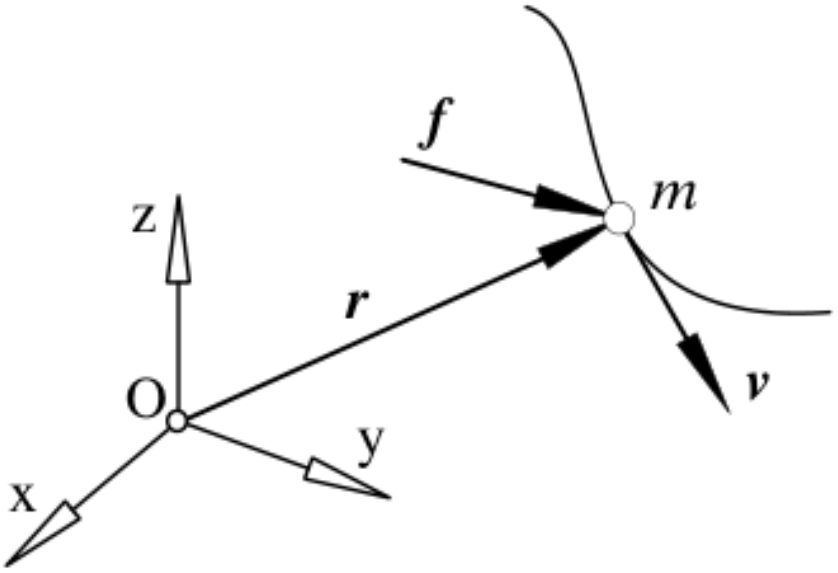
Istraživački roveri iskorištavaju ovu konfiguraciju integracijom svakog kotača s pogonskim aktuatorom, čime se maksimizira sposobnost pokretačke sile vozila. Jedan od glavnih nedostataka sadašnjih rocker-bogie rovera je to što su spori. Maksimalna brzina Curiosity rovera iznosi 0,14 km/h. Kako bi mogli svladati izrazito neravan teren (tj. prepreke veće od nekoliko postotaka radijusa kotača) bez značajnog rizika od prevrtanja vozila ili oštećenja ovjesa, ovi roboti kreću se polako i penju se preko prepreka tako što kotači podižu svaki dio ovjes preko prepreke dio po dio. Dok je izvedba na preprekama na neravnom terenu važna, treba uzeti u obzir i situacije gdje je površina ravna ili ima gotovo neprimjetne prepreke, gdje bi rover trebao povećati svoju brzinu kako bi brže stigao od točke A do točke B [12,13].

5. Matematički model

U ovom poglavlju, istražit će se dinamika čestica i krutih tijela temeljem zakona klasične mehanike kako bi se bolje razumio princip rada simulacije. Nadalje, bit će opisan model i sve potrebne jednadžbe kontaktne mehanike te kontaktni model tla.

5.1. Dinamika čestica

Dinamika čestica biti će opisana na temelju slike 14.



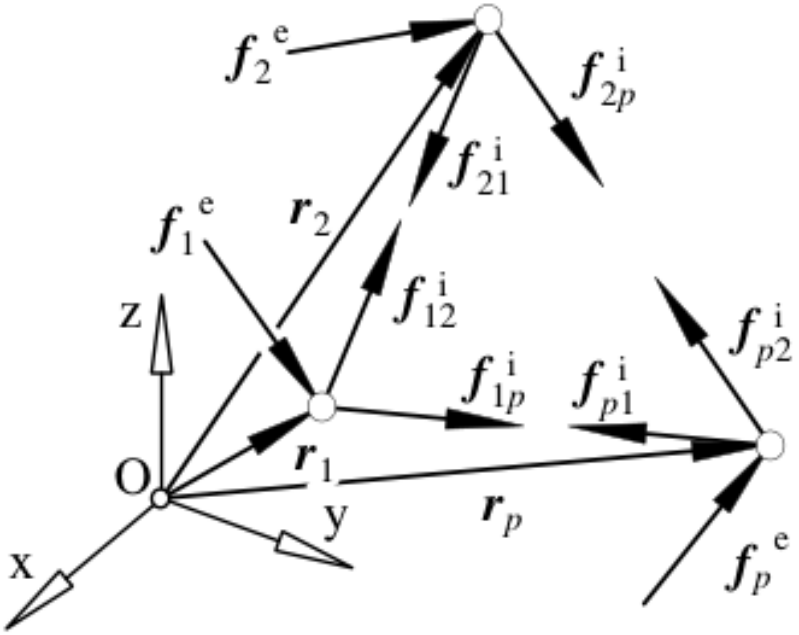
Slika 14. Gibanje čestica [14]

Primjenom drugog Newtonovog zakona, jednadžba gibanja čestice glasi:

$$f = m\ddot{r} = m\dot{v} = ma = \frac{d}{dt}(mv) \tag{5.1}$$

gdje je mv količina gibanja čestica. Vektor r označava vektor položaja čestice u inercijskom koordinatnom sustavu. Brzina čestice dana je s $v = \dot{r}$ dok je ubrzanje određeno s $a = \ddot{r}$. Da bi se ispravno primijenio drugi Newtonov zakon, derivacije u (5.1) moraju se izvesti u odnosu na inercijski koordinatni sustav. Kinetički moment čestice u odnosu na O dan je izrazom:

$$h_O = r \times mv\,. \tag{5.2}$$



Slika 15. Sustav čestica [14]

Ako gledamo sustav čestica, kao što je to prikazano na slici 14, Newtonov zakon glasi:

$$f_i = f_i^e + f_i^i = m_i a_i \tag{5.3}$$

gdje f_i^e predstavlja rezultantnu silu vanjskih sila. Rezultantna sila unutarnjih sila u sustavu dana je jednadžbom:

$$\sum_{j=1}^{p} f_{ij} = f_i^i \,. \tag{5.4}$$

Suma svih sila u sustavu čestica:

$$\sum_{i=1}^{p} f_i^e + \sum_{i=1}^{p} f_i^i = \sum_{i=1}^{p} m_i a_i \;, \sum_{i=1}^{p} f_i^i = 0 \tag{5.5}$$

$$\sum_{i=1}^{p} f_i^e = \sum_{i=1}^{p} m_i a_i \;. \tag{5.6}$$

Jednadžbu (5.6) moguće je detaljnije objasniti koristeći sljedeće jednakosti:

$$m r_c = \sum_{i=1}^{p} m_i r_i \tag{5.7}$$

gdje je r_c pozicija mase sustava u centru C. Prvom derivacijom jednadžbe (5.7) dobivamo količinu gibanja sustava čestica:

$$mv_c = \sum_{i=1}^{p} m_i v_i \ . \tag{5.8}$$

Druga derivacija jednadžbe (5.7) glasi:

$$ma_c = \sum_{i=1}^{p} m_i a_i \ . \tag{5.9}$$

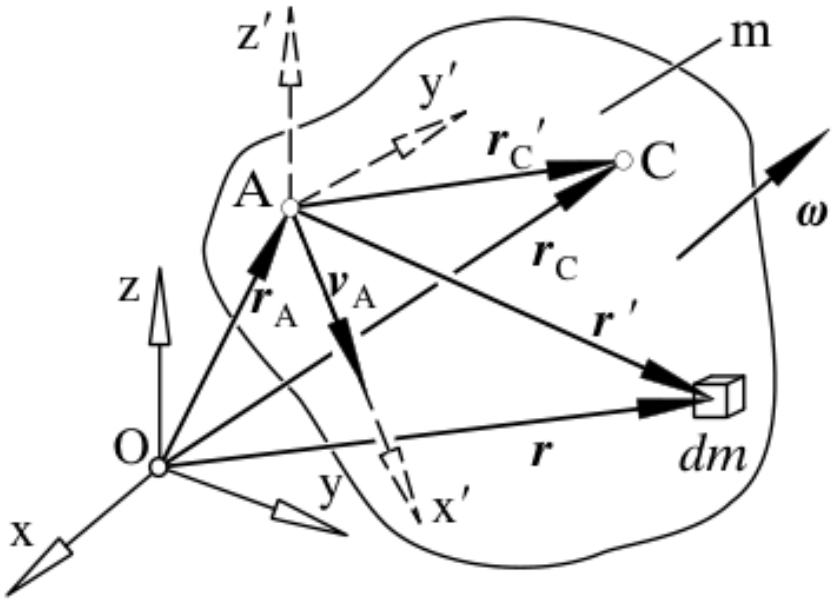
Uvrštavanjem jednadžbe (5.9) u (5.8) dobivamo:

$$\sum_{i=1}^{p} f_i^e = ma_c \tag{5.10}$$

što znači da se središte mase sustava čestica pomiče kao da je cijela masa sustava koncentrirana u toj točki i u njoj djeluju sve vanjske sile [14].

5.2. Dinamika krutih tijela

Kao i dinamiku čestica, dinamiku krutih tijela prikazati će se pomoću relacija koje se nalaze na slici 16.



Slika 16. Kruto tijelo [14]

Obzirom da se radi o krutom tijelu, r' je vektor položaja tijela koji ne mijenja svoju veličinu, već samo orijentaciju zbog rotacije tijela. Njegova vremenska derivacija u odnosu na inercijski sustav k može se izraziti kao

$$\dot{r}' = \omega \times r'. \tag{5.12}$$

Izraz (5.12) može se zapisati u obliku

$$r = r_A + r' \tag{5.13}$$

iz čega dobivamo izraz za brzinu

$$v = v_A + \dot{r}' = v_A + \omega \times r' \,. \tag{5.14}$$

Orijentacija tijela u inercijskom koordinatnom sustavu može se odrediti preko Eulerovih kutova ϕ, ϑ, ψ koji određuju orijentaciju nepomičnog sustava tijela k' u odnosu na inercijski sustav k. Odnos između ω i derivacija Eulerovih kutova $\dot{x}_R = \left[\dot{\phi}\ \dot{\vartheta}\ \dot{\psi}\right]^T$ može se izraziti u matričnom obliku kao

$$\omega = H_R \dot{x}_R \,. \tag{5.15}$$

Masa tijela zadana je

$$m = \int_m dm \ , \tag{5.16}$$

a vektor položaja centra mase tijela C u koordinatnom sustavu k

$$r_c = \frac{1}{m}\int r \, dm \ . \tag{5.17}$$

Pozicija točke C obzirom na fiksiranu točku A zadana je izrazom:

$$r_{AC} = r'_C = \frac{1}{m}\int r' dm \ . \tag{5.18}$$

Uvrštavanjem izraza (5.14) u (5.18) dobit će se količina gibanja:

$$\int (v_A + \omega \times r') dm = v_A m + \omega \times m r'_C = m(v_A + \omega \times r'_C) = m v_c = m \dot{r}_c \ . \tag{5.19}$$

Kinetički moment tijela obzirom na točku O može se prikazati pomoću:

$$h_O = \int_m r \times \dot{r} dm \tag{5.2}$$

ili uvrštavanjem izraza (5.13) i (5.14)

$$h_O = \int_m (r_A + r') \times (v_A + \omega \times r') dm$$

$$= r_A \times (v_A + \omega \times r'_C) m + r'_C \times v_A m + \int_m r' \times (\omega \times r') dm \ . \tag{5.21}$$

Izraz $\int_m r' \times (\omega \times r') dm$ moguće je zapisati u drugom obliku:

$$\int_m r' \times (\omega \times r')dm = \int_m (r'^2 E - r'r')dm * \omega = I_A * \omega \tag{5.22}$$

gdje je I_A tenzor inercije tijela u odnosu na A, a E je jedinični vektor, a $r'r'$ označava dijadni produkt. Konačno, uvrštavanjem izraza (5.22) u (5.21) te ako je središte mase tijela C odabrano kao referentna točka A dobije se:

$$h_O = r_C \times v_C m + I_C * \omega \,. \tag{5.23}$$

Jednadžbu dinamike translacijskog gibanja krutih tijela moguće je prikazati pomoću Newtonove jednadžbe:

$$\frac{d}{dt}(mv_C) = f \,. \tag{5.24}$$

Ako je masa tijela konstantna, izraz (5.24) postaje:

$$ma_C = f \,. \tag{5.25}$$

Eulerove jednadžbe određuju dinamiku rotacijskog gibanja tijela:

$$h_O = l_O \tag{5.26}$$

gdje je l_O rezultantni moment sile u odnosu na O. Rezultat derivacije jednadžbe (5.23) po vremenu je

$$\dot{h_O} = v_C \times v_C m + r_C \times ma_C + I_C * \dot{\omega} + \omega \times I_C * \omega \,, \tag{5.27}$$

a obzirom da je $v_C \times v_C m = 0$ vrijedi

$$I_C * \dot{\omega} + \omega \times I_C * \omega = l_O - r_C \times f = l_C \tag{5.28}$$

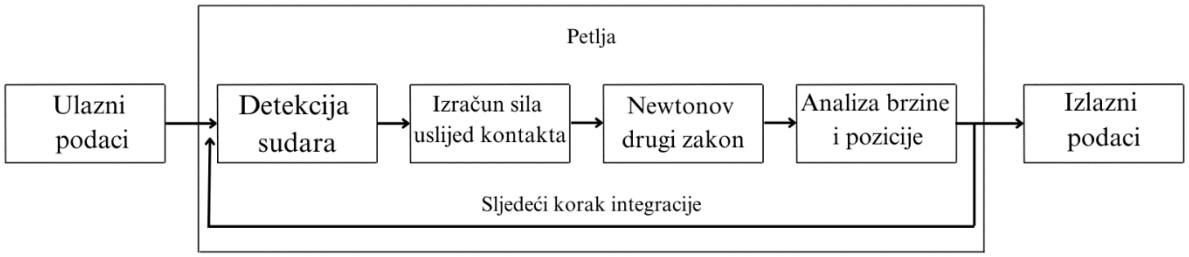
ili jednostavnije

$$I_C * \dot{\omega} + \omega \times I_C * \omega = l_C \,. \tag{5.29}$$

5.3. Metoda diskretnih elemenata (DEM)

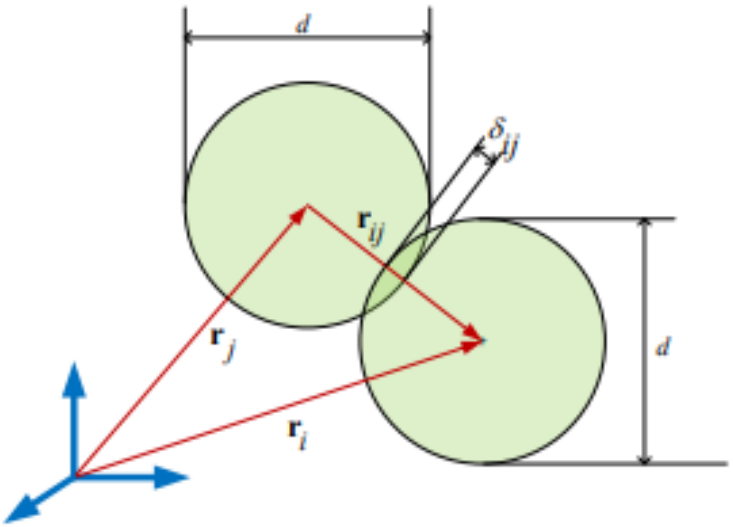
Metoda diskretnih elemenata (engl. discrete elements method - DEM) je računalna tehnika koja se koristi za simuliranje ponašanja zrnatih materijala, poput pijeska, šljunka, praškastih tvari i čestica. Umjesto da tretira materijal kao kontinuirani medij, DEM modelira pojedinačne čestice i njihove interakcije. Svaka čestica je predstavljena diskretnim elementom s definiranim svojstvima poput mase, veličine, oblika i kontaktnih sila. Simulacija prati kretanje i interakcije tih čestica tijekom vremena koristeći numeričke metode, obično zasnovane na Newtonovim zakonima gibanja i principima mehanike kontakta. DEM ima primjene u različitim područjima, uključujući geotehničko inženjerstvo, rudarstvo, farmaceutiku i biomehaniku.

Struktura DEM simulacije prikazana je na slici 17 te će se u nastavku detaljnije opisati svaki njezin dio.



Slika 17. Koraci diskretne metode elemenata

Detekcija sudara je neophodna u svakom vremenskom koraku kako bi se izračunale sile reakcije za svaki pronađeni kontakt. Broj elemenata može varirati od stotina do milijuna, što ovaj dio simulacije može učiniti dugotrajnim. Najjednostavniji algoritam za otkrivanje sudara provjerava svaku sferu u odnosu na sve ostale, što dovodi do kvadratnog skaliranja s brojem elemenata u sustavu. Umjesto provjeravanja svake kugle jedna s drugom, učinkovitiji način je provjeriti samo je li kugla u kontaktu sa svojim susjedima. U tu svrhu promatrano okruženje je podijeljeno u male spremnike u kojima se svako tijelo provjerava na koliziju s drugim tijelima koja su prisutna u istom spremniku. Ovaj pristup dovodi do linearnog skaliranja.



Slika 18. Pozicija čestica[15]

Slika 18 prikazuje dvije čestice na pozicijama, u kontaktu. Relativni položaj između obiju čestica označen je s r_{ij}, te vrijedi:

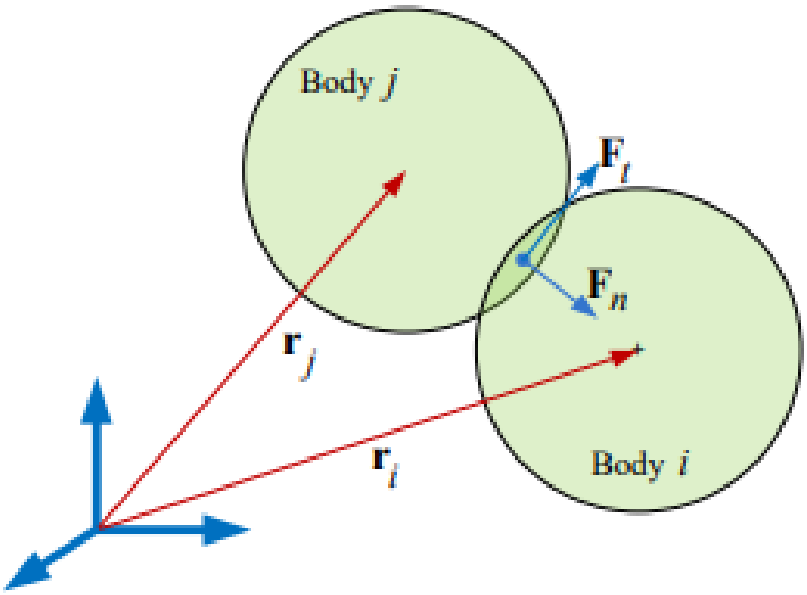
$$r_{ij} = r_i - r_j \,. \tag{5.3}$$

Dvije kugle su u kontaktu ako je udaljenost između njihovih središta masa manja od zbroja njihovih radijusa. Za dvije kugle jednakog radijusa uvjet izgleda ovako:

$$r_{ij} \leq d \tag{5.31}$$

gdje d predstavlja promjer kugle.

Izračun kontaktne sile između dva tijela može se podijeliti na normalnu i tangencijalnu komponentu, kao što se vidi na slici 19.



Slika 19. Kontaktne sile između dva tijela [15]

Modeli normalne sile mogu se organizirati u četiri različite kategorije, naime, kontinuirani potencijalni modeli, linearni viskoelastični modeli, nelinearni viskoelastični modeli i histerezni modeli. U linearnim viskoelastičnim modelima normalna sila je funkcija elastične i relativne normalne brzine između čestica koje se sudaraju. Model normalne sile, elastično odbijanje, normalna brzina i viskozna disipacija prikazani su u (5.32). Ovaj model sile može se shvatiti kao sustav prigušivača proporcionalan opruzi i brzini s određenom krutošću opruge i koeficijentom prigušenja, koji ovise o materijalu čestica.

$$F_{n_{ij}} = F_{el} + F_{diss} = -k_n d_{ij} + \gamma_n \dot{d}_{ij} \, . \tag{5.32}$$

Budući da su oba koeficijenta postavljena na konstantne vrijednosti, podrazumijeva se konstantan koeficijent restitucije. Koeficijent restitucije zapravo nije konstanta, već ovisi o veličini brzina koje čestice posjeduju prije sudara. Stoga su uvedeni nelinearni viskoelastični modeli kako bi se prevladao nedostatak konstantnog koeficijenta restitucije.

Normalna sila u tom slučaju glasi:

$$F_{n_{ij}} = -k_n d_{ij}^{\frac{3}{2}} + \gamma_n \dot{d}_{ij} \,. \tag{5.33}$$

Ovdje se koeficijenti restitucije izračunavaju kao funkcija relativnih brzina udara između čestica.

Histerezni modeli uključuju učinak plastičnosti ostavljajući čestice u kontaktu trajno deformiranima. Mogu biti linearni ili nelinearni i tipično dijele sudar na slučaj punjenja i pražnjenja s različitim koeficijentima krutosti.

$$F_{n_{ij}} = \begin{Bmatrix} -k_{load} d_{ij}, \dot{d}_{ij} \geq 0 \\ -k_{unload}\left(d_{ij} - d_0\right), \dot{d}_{ij} < 0 \end{Bmatrix} \tag{5.34}$$

Model se koristi za konstantni i varijabilni koeficijent restitucije.

Prema slici 16., vrijedi:

$$\delta_{ij} = d - r_{ij} \tag{5.35}$$

$$v_{ij} = v_i - v_j \,. \tag{5.36}$$

Da bi se izračunala normalna sila, relativna normalna brzina se određuje na sljedeći način:

$$v_{n_{ij}} = \left(v_{ij} * n_{ij}\right) n_{ij} \,. \tag{5.37}$$

Iz izraza (5.33) dobivamo:

$$F_{n_{ij}} = f\left(\frac{\delta_{ij}}{d}\right)\left(k_n \delta_{ij} n_{ij} - \gamma_n m_{eff} v_{n_{ij}}\right) \tag{5.38}$$

gdje je

$$m_{eff} = \frac{m_i m_j}{\left(m_i + m_j\right)} \tag{5.39}$$

za $f(x) = 1$ za model linearnog opružnog prigušivača ili $f(x) = \sqrt{x}$ za Hertzijeve kontakte.

Uz brojne modele normalne sile postoji niz različitih modela tangencijalnih sila koji na različite načine opisuju ponašanje pomaka sile. Relativna tangencijalna brzina zbroj je relativne brzine središta masa i brzine uslijed rotacije tijela koja se sudaraju

$$v_{t_{ij}} = v_{ij} - v_{n_{ij}} - \frac{1}{2}\left(\omega_i - \omega_j\right) \times r_{ij}. \tag{5.4}$$

Tangencijalni pomak je ekvivalent normalnoj kompresiji, ali je složenije izračunati ovo svojstvo. Na početku kontakta tangencijalni pomak je postavljen na nulu, ali treba ga nositi kroz cijeli interval dok čestice ostaju u kontaktu.

$$u_{t_{ij}} = \int_{(coll)} v_{t_{ij}}(\tau)\, d\tau \,. \tag{5.41}$$

Ako se u (5.41) uključi učinak rotacije tijela oko kontaktne točke što osigurava da uvijek leži u tangentnoj ravnini kontakta

$$\frac{du_{t_{ij}}}{dt} = v_{t_{ij}} - \frac{\left(u_{t_{ij}} * v_{ij}\right) r_{ij}}{r_{ij}^2} \ . \tag{5.42}$$

Tangencijalna sila se tada formulira na isti način kao normalna sila:

$$F_{t_{ij}} = f\left(\frac{\delta_{ij}}{d}\right)\left(k_t u_{t_{ij}} - \gamma_t m_{eff} v_{t_{ij}}\right) \ . \tag{5.43}$$

Ukupna sila koja djeluje na jednu česticu je zbroj svih kontaktnih sila i gravitacije.

$$F_i^{tot} = m_i g + \sum_j F_{n_{ij}} + F_{t_{ij}} \ . \tag{5.44}$$

Tangencijalne sile stvaraju moment oko središta mase svake nastale čestice što uzrokuje kutno ubrzanje.

$$M_i^{tot} = -\frac{1}{2}\sum_j r_{ij} \times F_{t_{ij}} \ . \tag{5.45}$$

Prema drugom Newtonovom zakonu dobivamo

$$a_i = \frac{F_i^{tot}}{m_i} \tag{5.46}$$

$$\alpha_i = \frac{M_i^{tot}}{I} \tag{5.47}$$

Metode eksplicitne numeričke integracije koriste se za izračunavanje položaja i brzine čestica u svakom vremenskom koraku, kao što su Eulerova metoda i Verletova integracija brzina. Razlika između obje numeričke metode je u tome što se prva derivacija ažurira u dvije faze. Sljedeće jednadžbe pokazuju korake potrebne za Verletova integracija brzina. Za tijelo i njegov novi položaj ovisi o prethodnom položaju, brzini i ubrzanju:

$$r_i(t + \Delta t) = r_i(t) + v_i(t)\Delta t + \frac{1}{2} a_i(t)\Delta t^2 \tag{5.48}$$

Brzina se izračunava u dva koraka ovisno o prethodnom ubrzanju i trenutnom ubrzanju određenom 2. Newtonovim zakonom. Iako su potrebne vrijednosti u dva različita vremena, one se ne moraju pohranjivati jedna pored druge, dijeljenjem ažuriranja brzine u dvije odvojene jednadžbe.

$$v_i\left(t + \frac{\Delta t}{2}\right) = v_i(t) + \frac{1}{2} a_i(t)\Delta t \tag{5.49}$$

$$v_i(t + \Delta t) = v_i\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2} a_i(t + \Delta t)\Delta t \tag{5.5}$$

Velika prednost ove metode je to što je potpomognuta s velikim brojem literature i brojem validacijskih studija. Može se prilagoditi širenju udarnog vala što nije slučaj s metodom koja će se detaljno opisati u sljedećem dijelu. Jednostavna je za implementaciju. Lako se poveća na velike probleme te je pogodna za paralelno računanje. Numerička stabilnost zahtijeva male vremenske korake integracije što može znatno utjecati na rezultate te ju karakterizira dugo vrijeme simulacije [15,20].

5.4. Diferencijalna varijacijska nejednakost (DVI)

Diferencijalna varijacijska nejednakost (engl. differential variational inequalities - DVI) je matematički koncept koji se pojavljuje u proučavanju optimizacijskih i ravnotežnih problema koji uključuju diferencijalne jednadžbe i nejednakosti. Bavi se pronalaženjem rješenja koje istovremeno zadovoljava sustav diferencijalnih jednadžbi i skup nejednakih ograničenja. Cilj je pronaći stanje ili putanju koja minimizira ili maksimizira određenu ciljnu funkciju pod tim ograničenjima. Diferencijalne varijacijske nejednakosti susreću se u mnogim područjima poput mehanike, ekonomije, teorije upravljanja i matematičke fizike. Pružaju okvir za modeliranje i rješavanje složenih optimizacijskih i ravnotežnih problema s kontinuiranim i diskretnim varijablama.

Stanje mehaničkog sustava s n_b krutim tijelima u trodimenzionalnom prostoru može se prikazati generaliziranim koordinatama

$$q = \left[r_1^T, \epsilon_1^T, \dots, r_{n_b}^T, \epsilon_{n_b}^T\right]^T \in \mathbb{R}^{7n_b} \tag{5.6}$$

te njihove derivacije po vremenu

$$\dot{q} = \left[\dot{r}_1^T, \dot{\epsilon}_1^T, \dots, \dot{r}_{n_b}^T, \dot{\epsilon}_{n_b}^T\right]^T \in \mathbb{R}^{7n_b} \tag{5.61}$$

gdje r_j je apsolutni položaj centra mase j-tog tijela i kvaterniona ϵ_j, što predstavlja njegovu rotaciju. Kvaternioni su algebarsko proširenje kompleksnih brojeva te za razliku od kompleksnih brojeva, imaju tri imaginarne jedinice Također, moguće je uvesti generalizirane brzine

$$v = \left[\dot{r}_1^T, \bar{\omega}_1^T, \dots, \dot{r}_{n_b}^T, \bar{\omega}_{n_b}^T\right]^T \in \mathbb{R}^{6n_b} \tag{5.62}$$

što je direktno povezano s $\dot{q}$ pomoću linearnog preslikavanja $\dot{q} = L(q)v$ koji transformira svaku kutnu brzinu $\bar{\omega}_j$ (izraženo u lokalnom koordinatnom sustavu tijela) u odgovarajući kvaternion $\dot{\epsilon}_j$ linearnom algebarskom formulom $\dot{\epsilon}_j = \frac{1}{2}G^T(\epsilon_j)\bar{\omega}_j$ gdje je

$$G(\epsilon_j) = \begin{bmatrix} +\epsilon_1 & +\epsilon_0 & -\epsilon_3 & +\epsilon_2 \\ +\epsilon_2 & +\epsilon_3 & +\epsilon_0 & -\epsilon_1 \\ +\epsilon_3 & -\epsilon_2 & +\epsilon_1 & +\epsilon_0 \end{bmatrix}. \tag{5.63}$$

Između dijelova mogu postojati mehanička ograničenja, poput rotacijskih ili prizmatičnih zglobova. Ova bilateralna ograničenja prevode se u algebarske jednadžbe koje ograničavaju relativni položaj i/ili orijentaciju parova tijela. Pod pretpostavkom da je skup B ograničenja prisutan u sustavu, ona vode do skupa skalarnih jednadžbi

$$\Psi_i(q,t) = 0, \quad i \in B\,. \tag{5.64}$$

Kako bi se osiguralo da se ograničenja ne krše u smislu brzina, mora se zadovoljiti i prva derivacija jednadžbi ograničenja, to jest

$$\nabla\Psi_i^T v + \frac{\partial \Psi_i}{\partial t} = 0, \quad i \in B \tag{5.65}$$

s Jacobijevom matricom $\nabla_q \Psi_i = \left[\frac{\partial \Psi_i}{\partial q}\right]^T$ i $\nabla\Psi_i^T = \nabla_q \Psi_i^T L(q)$.

Treba imati na umu da je izraz $\frac{\partial \Psi_i}{\partial t}$ jednak nuli za sva skleronomna ograničenja, ali može biti različit od nule za ograničenja koja nameću neki zakon putanje ili gibanja, kao što je slučaj s motorima i aktuatorima. Ograničenje je skleronomno ako jednadžba ograničenja ne uključuje vrijeme kao eksplicitnu varijablu.

Ako se uzimaju u obzir dodiri između krutih tijela, za svako tijelo moraju biti definirani oblici sudara. Algoritam za otkrivanje sudara, kao što je prethodno opisano, mora se koristiti za pružanje skupa ograničenja kontakta. Svako ograničenje kontakta predstavlja par tijela čiji su oblici dovoljno blizu, tako da se nejednakost može koristiti za koncizno izražavanje uvjeta ne prodiranja između volumena oblika:

$$\Phi_i(q) \geq 0, \;\; i \in A \tag{5.66}$$

Za zakrivljene konveksne oblike, kao što su kugle i elipsoidi, postoji jedinstveni par kontaktnih točaka (par najbližih točaka na njihovim površinama), ali u slučaju nekonveksnih oblika može postojati više parova kontaktnih točaka , čija definicija nije uvijek trivijalna.

Zbog pretpostavke krutog tijela i izbora središnjih referentnih okvira, generalizirana matrica mase M je konstantna i dijagonalna, te jednadžba gibanja glasi:

$$M\dot{v} = f(t,q,v) \tag{5.67}$$

gdje $f(t,q,v)$ predstavlja generaliziranu silu primijenjenu na sustav.

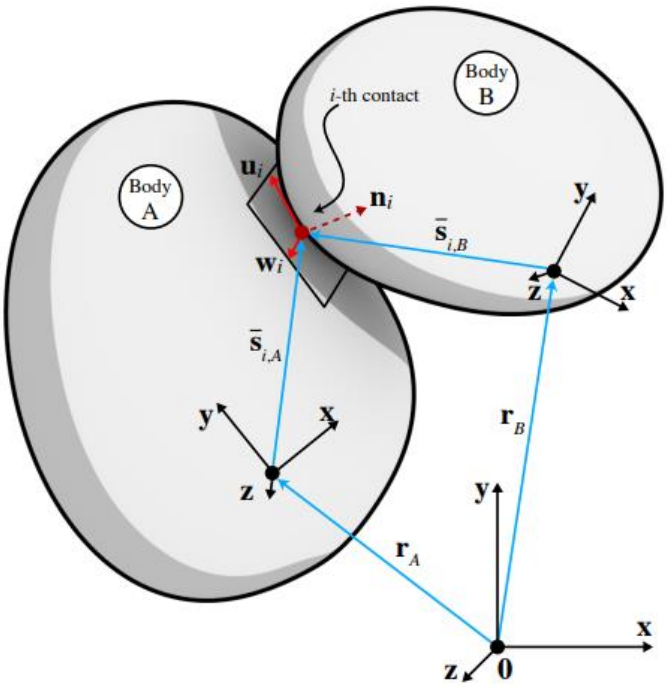
Također, prisutnost bilateralnih ograničenja kao što su sferni zglobovi, rotacijski zglobovi, translacijski zglobovi, itd., ograničavaju relativno gibanje dvaju krutih tijela. Skup holonomnih algebarskih jednadžbi induciran je prisutnošću svakog od ovih fizičkih spojeva

$$g_i(q,t) = 0, i \in \mathcal{B} \tag{5.68}$$

gdje je $\mathcal{B}$ skup svih bilateralnih ograničenja. Prisutnost fizičkih zglobova također izaziva kinematička ograničenja na razini brzine dobivena uzimanjem vremenske derivacije kinematičkih ograničenja položaja iz jednadžbe (5.68):

$$\dot{g}_i(q,t) = \vec{G}_i(q)v + \frac{\partial g_i}{\partial t} = 0\,. \tag{5.69}$$

Nametanje ovih ograničenja zahtijevaju prisutnost sile reakcije ograničenja $\vec{G}_i^T(q)\hat{\gamma}_{i,b}$, koja se dodaje desnoj strani jednadžbe gibanja (5.67) i ovisi o nepoznatom Lagrangeovom multiplikatoru (množitelju) $\hat{\gamma}_{i,b}$.



Slika 20. Dva tijela u kontaktu [16]

Na slici 20, dva tijela su u dodiru,

$$1 \leq A < B \leq n_b \tag{5.7}$$

gdje su $\bar{s}_{i,A}$ i $\bar{s}_{i,B}$ kontaktne točke u odnosu na referentni okvir tijela A i tijela B, n_i globalni jedinični vektor koji označava normalni smjer u kontaktnim točkama, a u_i i w_i dva jedinična vektora koji premošćuju kontaktnu ravninu u točki dodira.

Prema (5.7) dobije se:

$$F_{i,N} = \hat{\gamma}_{i,n} n_i \tag{5.71}$$

$$F_{i,T} = \hat{\gamma}_{i,u} u_i + \hat{\gamma}_{i,w} w_i \tag{5.72}$$

gdje su $\hat{\gamma}_{i,n}, \hat{\gamma}_{i,u}$ i $\hat{\gamma}_{i,w}$ iznosi sila u svakom smjeru.

Obzirom na tijelo B, $\hat{\gamma}_{i,n} = 0$, odnosno ako tijelo B dođe u kontakt s drugim tijelom normalna sila djelovati će na tijelo B da spriječi prodor.

Prema Coulombovom modelu trenja, u slučaju relativne tangencijalne brzine različite od nule smjer tangencijalne kontaktne sile poravnat je s vektorom tangencijalne brzine i njezina veličina proporcionalna je veličini normalne sile:

$$\|F_{i,T}\| = \mu_{i,d}\|F_{i,N}\| \tag{5.73}$$

gdje je $\mu_{i,d} \in \mathbb{R}^+$ dinamički faktor trenja. Kada je tangencijalna brzina jednaka nuli, tangencijalna sila ograničena je izrazom

$$\|F_{i,T}\| \leq \mu_{i,s}\|F_{i,N}\| \tag{5.74}$$

gdje je $\mu_{i,s} \in \mathbb{R}^+$ statički faktor trenja. Zbog jednostavnosti, moguće je pretpostaviti da su statički i dinamički faktor trenja jednaki

$$\mu_{i,d} = \mu_{i,s} = \mu_i \tag{5.75}$$

iz čega proizlazi:

$$\hat{\gamma}_{i,n} \geq 0, \qquad \Phi_i(q)\,\hat{\gamma}_{i,n} = 0, \tag{5.76}$$

$$\mu_i\hat{\gamma}_{i,n} \geq \sqrt{\hat{\gamma}_{i,u}^2 + \hat{\gamma}_{i,w}^2} \tag{5.77}$$

$$\langle F_{i,T}, v_{i,T}\rangle = -\|F_{i,T}\|\|v_{i,T}\| \tag{5.78}$$

$$\|v_{i,T}\|\mu_i\hat{\gamma}_{i,n} - \sqrt{\hat{\gamma}_{i,u}^2} + \hat{\gamma}_{i,w} = 0\,. \tag{5.79}$$

Diferencijalna varijacijska nejednakost omogućuje velike korake integracije budući da se ne mora baviti krutošću kontakta, ima smanjen broj parametara modela s kojima se može baratati te sagledava cijeli problem, ne odvaja ga na umjetan način. No, postoje i mane ovog pristupa. Zahtijeva globalno rješenje, što znači da veliki sustavi dovode do velikih povezanih problema, postoje izazovi vezani za modeliranje koeficijenta restitucije. Za razliku od nekih drugih optimizacijskih pristupa koji imaju dobro utvrđene formulacije i metodologije rješenja, DVI metodama može nedostajati standardizacija u smislu formulacije problema, dizajna algoritma i analize konvergencije. To može dovesti do nedosljednosti u terminologiji, notaciji i praksi implementacije u različitim istraživačkim studijama i primjenama [15, 16, 19, 20].

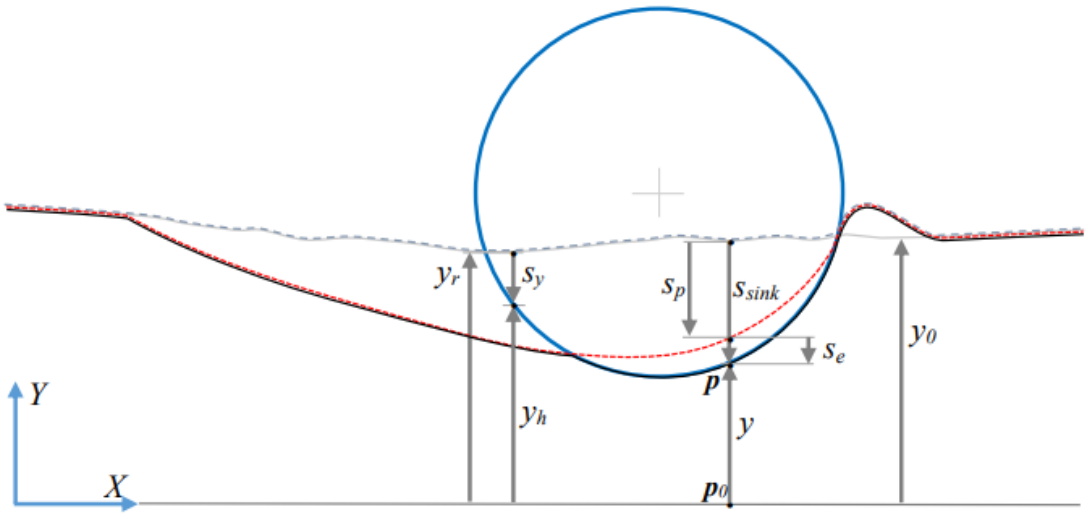
5.5. Kontaktni model tla (SCM)

Osim korištenja metode diskretnih elemenata i diferencijalne varijabilne nejednakosti, moguće je koristiti kontaktni model tla (engl. Soil Contact Model - SCM). Model se temelji na

poluempirijskom modelu s nekoliko parametara, stoga ga je lako prilagoditi. Nudi veliku brzinu simulacije i dovoljno je točan za mnoge scenarije. Može se smatrati generalizacijom Bekker-Wong modela na slučaj kotača proizvoljnih trodimenzionalnih oblika. Bekkerova formula za kotač koji se kreće po deformabilnom tlu koja nam govori o odnos tlaka i potonuća dan je izrazom:

$$\sigma = \left(\frac{k_c}{b} + k_\phi\right) y^n \tag{5.8}$$

gdje je σ tlak kontaktne površine, y je potonuće kotača, k_c je empirijski koeficijent koji predstavlja kohezivni učinak tla, k_ϕ je empirijski koeficijent koji predstavlja krutost tla, n je eksponent potonuća, izražavajući učinak stvrdnjavanja u zbijanje tla. Konačno, prema Bekkeru, b je duljina kraće stranice pravokutne kontaktne površine. Izvorni Bekkerov model pretpostavlja savršeno cilindričnu gumu u kontaktu s ravnom podlogom. Slika 21 prikazuje utjecaj kretanja tijela cilindričnog oblika po površini.



Slika 21. Kretanje cilindričnog tijela po površini

Pretpostavljamo da je za bilo koju točku $p_0(x, y)$ na globalnoj horizontalnoj ravnini moguće izračunati visinu nadolazećeg pomičnog oblika izvođenjem testa pogotka zrake od $p_0(x, y)$ duž y-osi do oblika . Udaljenost oblika duž y-osi je označena y_h; ako nije pronađen nijedan oblik, koristi se pomoćna velika vrijednost $y_h = \epsilon_\infty$. Utisnuta deformacija na tlo (kao "poniranje", pretpostavljamo pozitivno za deformaciju prema dolje), može se izračunati pomoću sljedećeg izraza:

$$s_y = y_r - y_h \tag{5.81}$$

gdje je y_r početna visina tla.

Problem kod ovog modela se javlja kod simulacija sustava više tijela zato što deformacija na tlo nije monotona, stoga se uvodi koeficijent k_e koji govori o elastičnoj krutosti tla. Nadalje, pretpostavlja se elastično-plastično ponašanje tla te se uvodi Reuss-Prandtl pretpostavka:

$$\dot{s}_y = \dot{s}_p - \dot{s}_e \tag{5.82}$$

gdje je $\dot{s}_p$ plastično strujanje za plastično tonjenje s_p te s_e elastična deformacija.

Testni tlak se računa

$$\sigma_{try} = k_e(s_y - s_p)\,. \tag{5.83}$$

Ako je $\sigma_{try} \leq 0$ prekršeni su uvjeti pa proizlazi da je $\sigma = 0, \dot{s}_p = 0$ i $y = y_p$.

Ako je $\sigma_{try} > 0$ računa se:

$$\sigma^* = \sigma_{try} \tag{5.84}$$

$$s_{sink} = s_y \tag{5.85}$$

$$y = y_h\,. \tag{5.86}$$

Također, pozitivni σ^* je ograničen na granicu tečenja zadanu s plastičnim pravilom. Ako je $\sigma^* < \sigma_{yield}$ onda je $\sigma^* = \sigma$, ali ako je $\sigma^* > \sigma_{yeld}$ onda se računa za vremenski korak h:

$$\sigma = k_\phi s_y^n \tag{5.87}$$

$$\sigma_{yield} = \sigma \tag{5.88}$$

$$s_p^o = s_p \tag{5.89}$$

$$s_p = s_{sink} - \frac{\sigma}{k_e} \tag{5.9}$$

$$\dot{s}_p = \left(s_p - s_p^o\right)\frac{1}{h} \tag{5.91}$$

$$s_e = s_{sink} - s_p\,. \tag{5.92}$$

Opcionalno može se uvesti viskozni učinak prigušenja:

$$\sigma_{tot} = \sigma + v_n r \tag{5.93}$$

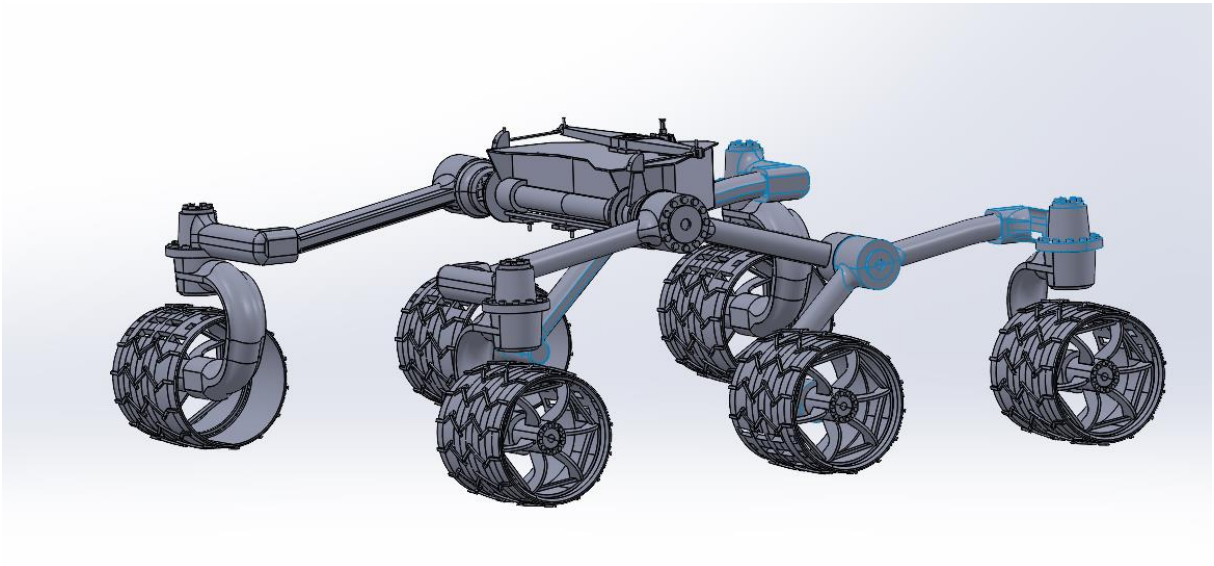
gdje je r parametar prigušenja, a v_n modul brzine sabijanja tla [21].

6. Numeričko modeliranje

Kao što je već navedeno u uvodu, simulacije će se provesti u programskom paketu otvorenog koda Project Chrono. Za potrebe simulacije, osim osnovnog modula, korišteni su Vehicle i Irrlicht moduli. U ovom poglavlju biti će opisani dijelovi koda, odnosno izrada simulacije te obrada dobivenih podataka i usporedba s već postojećim eksperimentima na temelju primjera testnog rovera Scarecrow. Za izradu simulacije korišten je Microsoft Visual Studio 2022 te je simulacija pokrenuta na osobnom računali s procesorom 12th Gen Intel® Core™ i7-12700K te grafičkom karticom Intel® UHD Graphics 770 i NVIDIA GeForce RTX 3070.

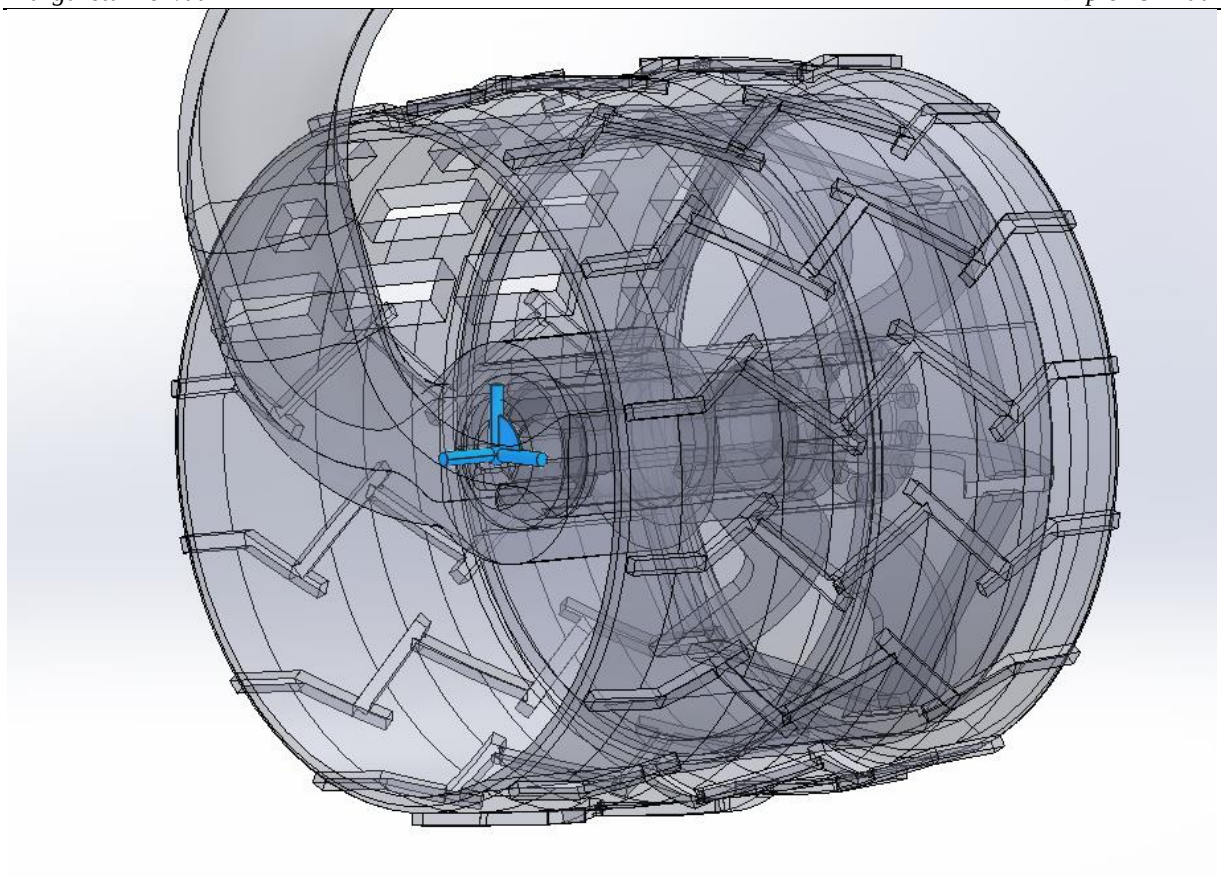
6.1. Izrada CAD modela rovera

CAD (engl. Computer Aided Design – CAD) model rovera moguće je uvesti pomoću CASCADE modula u obliku STEP datoteke, koji svaki uvezeni dio ili sklop tretira kao jedno kruto tijelo. Iako Project Chrono sam po sebi ima mogućnost stvaranja geometrijskog tijela, ona su vrlo ograničena i jednostavnog oblika. CAD model izrađen je u programu SolidWorks prema stvarnim dimenzijama Curiosity rovera (slika 22).



Slika 22. CAD model

Obzirom da modul prikazuje svaki uvezeni dio kao jedno kruto tijelo, potrebno je između njih staviti kinematička ograničenja. Kinematička ograničenja označiti ćemo pomoću markera koji su napravljeni unutar CAD modela te se nalaze na svim mjestima gdje se dijelovi rovera spajaju. Jedan takav marker prikazan je na slici 23. Markeri se neće prikazati kao geometrijsko tijelo, već se samo uzimaju njihove koordinate.



Slika 23. Primjer markera

U Curiosity roveru potrebno je odrediti 5 zglobnih oslonaca na spojevima rockera i tijela rovera, 4 kruta oslonca na spojevima vilica prednjih i stražnjih kotača i 4 kuglasta oslonca koji su postavljeni između diferencijalne poluge i spojnih šipki te na spojevima rockera i spojnih šipki. Osim zglobova, potrebno je zadati i 6 motora s propisanom brzinom na svaki kotač.

6.2. Izrada simulacije temeljenoj na diferencijalnoj varijacijskoj nejednakosti

Prvo je potrebno deklarirati i uključiti sve biblioteke te navesti određeni namespace kako bi se izbjeglo sukobljavanje imena između različitih dijelova programa. On sadrži funkcije, klase, varijable i druge identifikatore. U programskom paketu Project Chrono postoje više modula za vizualni prikaz, a to su modul Irrlicht, modul VSG te modul OpenGL te je moguće simulaciju prikazati u svima od njih. U ovoj simulaciji odabran je modul Irrlicht. Također, parametri vizualnog prikaza definirati će se kasnije u kodu.

```
#include "chrono_models/robot/curiosity/Curiosity.h"
#include "chrono/physics/ChBodyEasy.h"
#include "chrono/assets/ChTexture.h"
#include "chrono/utils/ChUtilsCreators.h"
#include "chrono/utils/ChUtilsGeometry.h"
```

```
#include "chrono/utils/ChUtilsInputOutput.h"
#include "chrono/assets/ChVisualShapeBox.h"
#include <cstdio>
#include <cmath>
#include <iostream>
#include <fstream>
#include "chrono_thirdparty/filesystem/path.h"
#include "chrono/utils/ChUtilsCreators.h"
#include "chrono/physics/ChSystemNSC.h"

#include "chrono/assets/ChVisualSystem.h"
#ifdef CHRONO_IRRLICHT
#include "chrono_irrlicht/ChVisualSystemIrrlicht.h"
using namespace chrono::irrlicht;
#endif

using namespace chrono;
using namespace chrono::curiosity;
```

U sljedećem koraku, odabran je sustav ChSystemNSC. To je sustav u kojem se koristi neglatki kontakt. Također, određena je gravitacija te vrsta sudara.

```
ChSystemNSC sys;
sys.Set_G_acc(ChVector<>(0, 0, -9.81));
sys.SetCollisionSystemType(ChCollisionSystem::Type::BULLET);
```

Nadalje, potrebno je napraviti nevidljivu kutiju kako bi čestice pijeska zadržali na okupu. Prikazan je dio koda u kojem se stvara tlo te jedan od četiri zida.

```
// Create the ground and obstacles

auto ground_mat = chrono_types::make_shared<ChMaterialSurfaceNSC>();
auto ground = chrono_types::make_shared<ChBodyEasyBox>(6, 3.8, 0.1, 1000, true, true,
ground_mat);
ground->SetPos(ChVector<>(1, 0, -0.5));
ground->SetBodyFixed(true);
ground->GetVisualShape(0)->SetTexture(GetChronoDataFile("textures/concrete.jpg"), 60,
45);
sys.Add(ground);

// Create the first wall
auto mbox_1 = chrono_types::make_shared<ChBodyEasyBox>(6, 0.1, 1, 1000, false, true,
ground_mat);
mbox_1->SetPos(ChVector<>(1, -2, 0));
mbox_1->SetBodyFixed(true);
mbox_1->SetCollide(true);
sys.Add(mbox_1);
```

Osim tla i zidova, potrebno je stvoriti prepreku. Project Chrono nema mogućnost stvaranja polusfere, no taj problem je riješen na način da je stvorena sfera te je pomaknuta u tlo.

```
//Creating sphere as obstacle
auto obstacle = chrono_types::make_shared<ChBodyEasySphere>(0.15, 1000, true, true, ground_mat);
obstacle->SetBodyFixed(true);
obstacle->SetPos(ChVector<>(1.6, -1, -0.4));
obstacle->GetVisualShape(0)->SetTexture(GetChronoDataFile("textures/concrete.jpg"));
sys.Add(obstacle);
```

Čestice pijeska stvoriti ćemo na sličan način. Prvo ćemo odabrati koeficijent restitucije i koeficijent trenja, zatim radijus, gustoće i početnu udaljenost između svake čestice. Pomoću petlje stvoriti će se čestice u smjeru osi x, y i z. Nakon završetka petlje, ispisati će se broj stvorenih čestica.

```
//creating material properties for particles

std::shared_ptr<ChMaterialSurface> particle_mat;
auto p_mat = chrono_types::make_shared<ChMaterialSurfaceNSC>();
p_mat->SetRestitution(0.9f);
p_mat->SetFriction(0.4f);
particle_mat = p_mat;

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 numy = (3.8 - delta) / (2 * pRad + delta);
double numz = (0.2 - delta) / (2 * pRad + delta);
int n = 0;      //counter for total number of particles

//loops for creating spherical ground particles
ChVector<> refPosL = ChVector<>(-2 + pRad + delta, -1.9 + pRad + delta, -0.5 + 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);
```

```
            particle->GetVisualShape(0)-
>SetTexture(GetChronoDataFile("textures/concrete.jpg"));
            sys.Add(particle);
        }
    }
}
GetLog() << "Created " << n << "particles in total \n";
```

U sljedećem koraku, uvezen je odabrani rover te je određena brzina motora svakog kotača.

```
// Create a Curiosity rover and the asociated driver
auto driver = chrono_types::make_shared<CuriositySpeedDriver>(0, 0.08889);
Curiosity rover(&sys, chassis_type, wheel_type);
rover.SetDriver(driver);
rover.Initialize(ChFrame<>(ChVector<double>(0, 0, -0.2), QUNIT));
```

Kako je već spomenuto, potrebno je odrediti parametre vizualnog prikaza.

```
      std::shared_ptr<ChVisualSystem> vis;
    switch (vis_type) {
    case ChVisualSystem::Type::IRRLICHT: {
#ifdef CHRONO_IRRLICHT
        auto vis_irr = chrono_types::make_shared<ChVisualSystemIrrlicht>();
        vis_irr->AttachSystem(&sys);
        vis_irr->SetCameraVertical(CameraVerticalDir::Z);
        vis_irr->SetWindowSize(800, 600);
        vis_irr->SetWindowTitle("Curiosity Rover");
        vis_irr->Initialize();
        vis_irr->AddLogo();
        vis_irr->AddSkyBox();
        vis_irr->AddCamera(ChVector<>(0, 3, 0));
        vis_irr->AddTypicalLights();
        vis_irr->AddLightWithShadow(ChVector<>(2.5, 7.0, 0.0), ChVector<>(0, 0, 0), 50, 4,
25, 130, 512,
            ChColor(0.8f, 0.8f, 0.8f));
        vis_irr->EnableContactDrawing(ContactsDrawMode::CONTACT_NONE);
        vis_irr->EnableShadows();

        vis = vis_irr;
#endif
        break;
    }
    }
```

Željene rezultate zapisati će se u tekstualnu datoteku koja će se kreirati na početku simulacije i zatvoriti na kraju. Obzirom da su potrebne sile tri kotača, za svaki kotač je potrebno napraviti tekstualnu datoteku. Prikazan je dio koda za samo jedan od tri kotača, ali za svaki kotač je postupak isti.

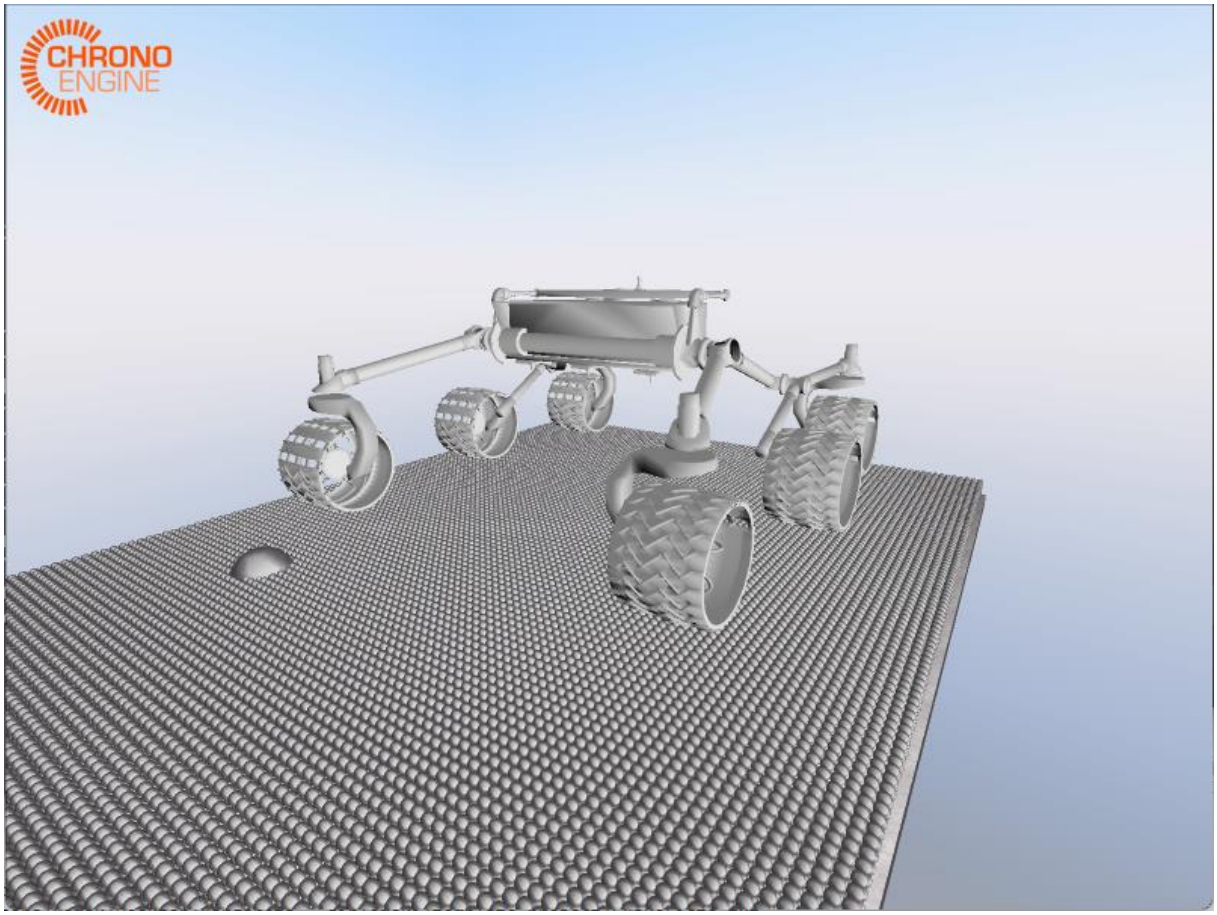
```
const bool output = true;  // Set to true if you want to output data
const std::string out_dir = "C:\\Users\\Desktop\\Margareta -
diplomski\\template_project\\Build\\Debug";  // Change this to your desired output directory

if (output) {
    if (!filesystem::create_directory(filesystem::path(out_dir))) {
        std::cout << "Error creating directory " << out_dir << std::endl;
        return 1;
    }
}

// Open a text file for writing
std::ofstream txt_file_front_R(out_dir + "/Force_FR.txt");

if (!txt_file_front_R.is_open()) {
    std::cerr << "Error opening Force_FR.txt" << std::endl;
    return 1;
}
```

Važno je odrediti parametre simulacije, odnosno vrijeme trajanja simulacije te korak. U ovoj simulaciji odabran je korak 0,001 s te ukupno trajanje simulacije 30 sekundi. Također, prikazan je dio u kojem se tijekom simulacijske petlje zapisuje sila desnog prednjeg kotača. Time izrada koda završava.

```
    //Simulation settings
    double time = 0.0;
    double timeStep = 0.001;
    double timeEnd = 30.0;

    // Simulation loop
    while (time <= timeEnd) {
#if defined(CHRONO_IRRLICHT) || defined(CHRONO_VSG)
        vis->BeginScene();
        vis->Render();
        vis->EndScene();
#endif
        if (output) {
            // Write force
            txt_file_front_R << rover.GetWheelAppliedForce(CuriosityWheelID::C_FR) << "; ";
        }
    // Update Curiosity controls
    rover.Update();
    sys.DoStepDynamics(timeStep);
    GetLog() << time << "\n";
    time += timeStep;
}
// Close the file
txt_file_front_L.close();
```

```
    return 0;
}
```

Vizualni prikaz simulacije nalazi se na slici 24.



Slika 24. Vizualni prikaz simulacije

6.2.1. Obrada rezultata

Simulacija je trajala 30 sekundi, a obzirom da je definiran korak 0,001 s, dobiveno je 30 tisuća rezultata samo za jednu komponentu svake sile, što znači da je stvarni broj podataka 90 tisuća zato što su iznosi svake sile zapisani su u tri komponente, za sve tri osi. Broj dobivenih čestica u ovoj simulaciji iznosi 53728. Obrada rezultata odrađena je u Pythonu. Prvo su učitane sve potrebne pomoću naredbe import.

```
import pandas as pd
import numpy as np
from scipy.interpolate import make_interp_spline
import matplotlib.pyplot as plt
```

Pandas se koristi za manipulaciju podacima, posebno za rad s DataFrame-ima. Numpy pruža podršku za matematičke operacije na nizovima. Scipy.interpolate.make_interp_spline koristi se za stvaranje glatkih krivulja koje prolaze kroz točke podataka. Matplotlib.pyplot omogućava stvaranje grafova i vizualizacija.

Nakon učitavanja biblioteka, program učitava podatke iz CSV datoteke rezultati.csv i pohranjuje ih u DataFrame pod imenom rezultati. Zatim se iz DataFrame-a odabiru x i y vrijednosti za grafikon, koje se nalaze u posljednje dvije kolone.

```
# Load the dataset
rezultati = pd.read_csv('rezultati.csv')

# Select the last two columns for plotting
x_values = proba.iloc[:, -1].values
y_values = proba.iloc[:, -2].values
```

Da bi se osiguralo ispravno povezivanje točaka na grafu, x vrijednosti se sortiraju, a zatim se sortirane x i odgovarajuće y vrijednosti pohranjuju u x_values_sorted i y_values_sorted.

```
# Sort x_values and corresponding y_values together
sorted_indices = np.argsort(x_values)
x_values_sorted = x_values[sorted_indices]
y_values_sorted = y_values[sorted_indices]
```

Nakon toga, koristi se funkcija make_interp_spline iz scipy biblioteke kako bi se stvorila glatka krivulja koja prolazi kroz sortirane točke. Xnew varijabla predstavlja novi set x vrijednosti na kojima će se izračunati odgovarajuće y vrijednosti za glatku krivulju.

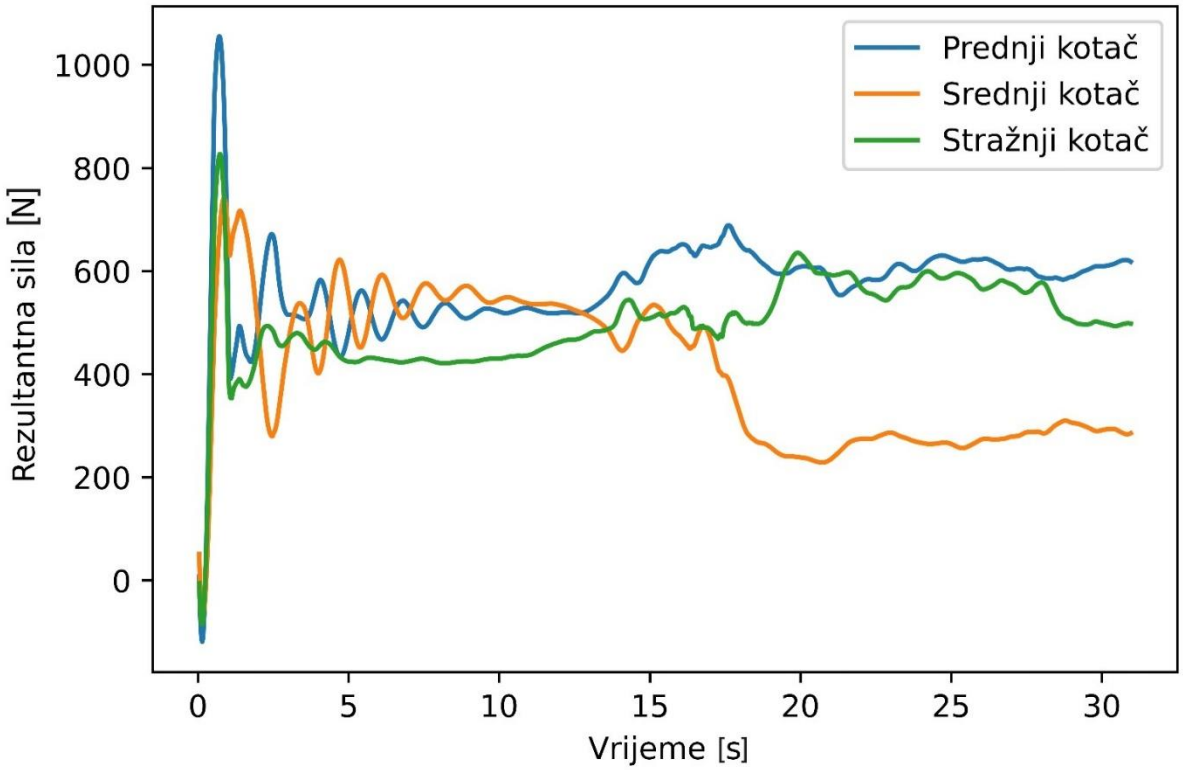
```
# Create a smooth curve
xnew = np.linspace(x_values_sorted.min(), x_values_sorted.max(), 300)
spl = make_interp_spline(x_values_sorted, y_values_sorted, k=3)
y_smooth = spl(xnew)
```

Konačno, koristi se matplotlib za crtanje grafikona. Funkcija plt.plot() crta krivulju, a zatim se postavljaju oznake osi i naslov grafa. Nakon toga, pozivom plt.show() prikazuje se grafikon.

```
# Plotting
plt.figure(facecolor='white')
plt.plot(xnew, y_smooth, label='')
plt.xlabel('Vrijeme,s')
plt.ylabel('Rezultantna sila, N')
plt.title('')
plt.legend().remove()
plt.show()
```

U obradi se koristi interpolacija te se time smanjuje broj prikazanih točaka na grafu te smanjuje njegova zagušenost. Mana ove metode je to što su se određeni rezultati eliminirali te su se pojedini ekstremi izbrisali. Obzirom da nije moguće utvrditi što točno utječe na te ekstreme, jesu li oni stvarni dobiveni rezultat simulacije ili jednostavno greška, ovaj pristup se prikazao dovoljno dobar za prikaz rezultata.

Cilj ove simulacije bio je prikazati sile u svakom kotaču koji prelazi preko prepreke. Dijagram sila prikazan je na slici 25.



Slika 25. Dijagram rezultantnih sila (DVI)

U početku simulacije vidi se veliki skok u silama kod sva tri kotača, što je rezultat uvoza CAD modela. CAD model se uvozi tako da se spusti na tlo, u ovom slučaju čestice. Sljedećih par sekundi model oscilira, odnosno počinje se kretati. Prednji kotač prvi prelazi preko prepreke te bi u trenutku prelaska preko nje dijagram trebao pokazati značajan skok, no to se na ovom dijagramu ne vidi. Isto se dogodi i s preostala dva kotača. Osim što se vidi da ne postoji skok u trenutku prelaska guma preko prepreka, vidi se da nakon određenog vremena sile sve manje i manje osciliraju, iz čega se da zaključiti da se rover ne kreće po zadanim česticama nego se vrti u mjestu i propada sve do tla te se po njemu kreće kretati. Razlog tome mogu biti razni faktor, no najvjerojatnije je ograničenosti računala za ovakvu vrstu simulacije. Postavljeni korak iznosi 0,001 s, a obzirom da se u svakom koraku zapisuje što se točno u tom trenutku dogodi, kako ne

bi došlo po ovakvih problema potrebno je postaviti veći korak. Simulacija je jako zahtjeva obzirom na veliki broj čestica te je zbog ograničenosti računala nemoguće s preciznošću prikazati što se točno dogodi. Na temelju ovoga moguće je zaključiti da su rezultati ove simulacije nereprezentativni.

6.3. Izrada simulacije temeljena na kontaktnom modelu tla

Kako bi prikazali što se točno dogodi sa silama kada rover dolazi na prepreku, ponovljena je simulacija u kojoj se koristi kontaktni model tla. Kao i u izradi simulacije temeljene na diferencijalnoj varijacijskoj nejednakosti, prvo je potrebno navesti sve potrebne biblioteke. Kao što je već napomenuto, postoji više različitih modula za vizualni prikaz. Iz toga razloga, u kodu su definirani modul Irrlicht i modul VSG te će se kasnije u kodu odabrati i pozivati jedan od njih.

```
#include <chrono>
#include "chrono_models/robot/curiosity/Curiosity.h"
#include "chrono/core/ChRealtimeStep.h"
#include "chrono/physics/ChSystemSMC.h"
#include "chrono/physics/ChBodyEasy.h"
#include "chrono/physics/ChInertiaUtils.h"
#include "chrono/assets/ChTexture.h"
#include "chrono/assets/ChVisualShapeTriangleMesh.h"
#include "chrono/utils/ChUtilsInputOutput.h"
#include "chrono/utils/ChUtilsCreators.h"
#include "chrono/core/ChStream.h"
#include <iostream>
#include <fstream>
#include "chrono_vehicle/ChVehicleModelData.h"
#include "chrono_vehicle/terrain/SCMTerrain.h"
#include "chrono_thirdparty/filesystem/path.h"

#include "chrono/assets/ChVisualSystem.h"

#ifdef CHRONO_IRRLICHT
#include "chrono_irrlicht/ChVisualSystemIrrlicht.h"
using namespace chrono::irrlicht;
#endif

#ifdef CHRONO_VSG
#include "chrono_vsg/ChVisualSystemVSG.h"
using namespace chrono::vsg3d;
```

```
#endif

using namespace chrono;
using namespace chrono::curiosity;
```

U sljedećem koraku, potrebno je odrediti koji modul za vizualni prikaz želimo koristiti. Odabran je modul Irrlicht. Također, obzirom da će se teren prikazivati kao mreža, potrebno je odrediti hoće li se prikazivati bulldozing efekt i moving patch značajku te rezolucija mreže. Također, definiran je korak simulacije te je postavljena na 0,001 s.

```
// Run-time visualization system (IRRLICHT or VSG)
ChVisualSystem::Type vis_type = ChVisualSystem::Type::IRRLICHT;

// SCM grid spacing
double mesh_resolution = 0.02;

// Enable/disable bulldozing effects
bool enable_bulldozing = true;

// Enable/disable moving patch feature
bool enable_moving_patch = true;

// Specify rover chassis type (Scarecrow or FullRover)
CuriosityChassisType chassis_type = CuriosityChassisType::Scarecrow;

// Specify rover wheel type (RealWheel, SimpleWheel, or CylWheel)
CuriosityWheelType wheel_type = CuriosityWheelType::RealWheel;

// Simulation time step
double time_step = 0.001;
```

Nadalje, potrebno je definirati inicijalizirati sustav simulacije. Kreiran je sustav ChSystemSMC. ChSystemSMC je klasa za fizički sustav u kojem se kontakt modelira upotrebom glatke metode. Također, inicijaliziran je Curiosity rover, odnosno kutna brzina svakog kotača te pozicija u simulaciji.

```
// Create the Chrono system and the associated collision system
ChSystemSMC sys;
sys.SetCollisionSystemType(ChCollisionSystem::Type::BULLET);

// Create the Curiosity rover
```

```
Curiosity rover(&sys, chassis_type, wheel_type);
rover.SetDriver(chrono_types::make_shared<CuriositySpeedDriver>(1, 0.08889));
rover.Initialize(ChFrame<>(ChVector<>(-5, -0.52, 0), Q_from_AngX(-CH_C_PI / 2)));
```

Obzirom da želimo da rover u simulaciji prelazi preko prepreke potrebno ju je definirati njezine parametre te dodati u simulaciju u obliku obj datoteke. Nadalje, potrebno je definirati njezinu poziciju unutar simulacije, odnosno postaviti ju tako da rover prelazi preko nje tijekom simulacije. Pomoću rock_scale možemo ju skalirati da bude željene veličine. Obzirom da je u eksperimentu s kojim ćemo usporediti dobivene rezultate prepreka polusfera radijusa 15 centimetara, odabrali smo rock_scale = 0,08.

```
// Create obstacles
std::vector<std::shared_ptr<ChBodyAuxRef>> rock;
std::vector<std::string> rock_meshfile = {
   "robot/curiosity/rocks/Prepreka.obj" //
};
std::vector<ChVector<>> rock_pos = {
   ChVector<>(-3.5, -0.75, -1.2) //
};
std::vector<double> rock_scale = {
   0.08   //

};
double rock_density = 8000;
std::shared_ptr<ChMaterialSurface> rock_mat =
ChMaterialSurface::DefaultMaterial(sys.GetContactMethod());

for (int i = 0; i < 1; i++) {
   auto mesh =
ChTriangleMeshConnected::CreateFromWavefrontFile(GetChronoDataFile(rock_meshfile[i])
, false, true);
   mesh->Transform(ChVector<>(0, 0, 0), ChMatrix33<>(rock_scale[i]));

   double mass;
   ChVector<> cog;
   ChMatrix33<> inertia;
   mesh->ComputeMassProperties(true, mass, cog, inertia);
   ChMatrix33<> principal_inertia_rot;
   ChVector<> principal_I;
   ChInertiaUtils::PrincipalInertia(inertia, principal_I, principal_inertia_rot);
```

```
    auto body = chrono_types::make_shared<ChBodyAuxRef>();
    sys.Add(body);
    body->SetBodyFixed(true);
    body->SetFrame_REF_to_abs(ChFrame<>(ChVector<>(rock_pos[i]), QUNIT));
    body->SetFrame_COG_to_REF(ChFrame<>(cog, principal_inertia_rot));
    body->SetMass(mass * rock_density);
    body->SetInertiaXX(rock_density * principal_I);

    auto ct_shape = chrono_types::make_shared<ChCollisionShapeTriangleMesh>(rock_mat,
mesh, false, false, 0.005);
    body->AddCollisionShape(ct_shape);
    body->SetCollide(true);

    auto vis_shape = chrono_types::make_shared<ChVisualShapeTriangleMesh>();
    vis_shape->SetMesh(mesh);
    vis_shape->SetBackfaceCull(true);
    body->AddVisualShape(vis_shape);

    rock.push_back(body);
}
```

Unutar programskog paketa Project Chrono postoje više vrsta terena koji se mogu koristiti, a odabran je kontaktni model tla (engl. Soil Contact Model - SCM) te njegovi parametri. Vrlo važno je prilagoditi koordinatni sustav terena i koordinatni sustav rovera. U koordinatnom sustavu terena z-os postavljena je prema gore, odnosno vertikalno, dok je u koordinatnom sustavu rovera y-os postavljena gore. Stoga je koordinatni sustav terena rotiran za 90 stupnjeva oko x-osi. Također, postavljena je dužina i širina terena te teramehanička svojstva tla.

```
// Create the SCM terrain
vehicle::SCMTerrain terrain(&sys);

// Displace/rotate the terrain reference plane.
// Note that SCMTerrain uses a default ISO reference frame (Z up). Since the mechanism is
modeled here in a Y-up global frame, we rotate the terrain plane by -90 degrees about the X
axis.
terrain.SetPlane(ChCoordsys<>(ChVector<>(0, -0.5, 0), Q_from_AngX(-CH_C_PI_2)));

// Use a regular grid
double length = 14;
double width = 4;
terrain.Initialize(length, width, mesh_resolution);
```

```
// Set the soil terramechanical parameters
terrain.SetSoilParameters(0.82e6,   // Bekker Kphi
    0.14e6,   // Bekker Kc
    1.0,      // Bekker n exponent
    0,  // Mohr cohesive limit (Pa)
    30,       // Mohr friction limit (degrees)
    0.01,  // Janosi shear coefficient (m)
    4e7,      // Elastic stiffness (Pa/m), before plastic yield, must be > Kphi
    3e4       // Damping (Pa s/m), proportional to negative vertical speed (optional)
);
```

Kao što je ranije napomenuto, simulaciju je moguće prikazati pomoću modula VSG i modula Irrlicht. Na samome početku odabran je modul Irrlicht te je potrebno izmijeniti željene parametre.

```
// Create the run-time visualization interface
#ifndef CHRONO_IRRLICHT
    if (vis_type == ChVisualSystem::Type::IRRLICHT)
        vis_type = ChVisualSystem::Type::VSG;
#endif
#ifndef CHRONO_VSG
    if (vis_type == ChVisualSystem::Type::VSG)
        vis_type = ChVisualSystem::Type::IRRLICHT;
#endif

    std::shared_ptr<ChVisualSystem> vis;
    switch (vis_type) {
    case ChVisualSystem::Type::IRRLICHT: {
#ifdef CHRONO_IRRLICHT
        auto vis_irr = chrono_types::make_shared<ChVisualSystemIrrlicht>();
        vis_irr->AttachSystem(&sys);
        vis_irr->SetCameraVertical(CameraVerticalDir::Y);
        vis_irr->SetWindowSize(800, 600);
        vis_irr->SetWindowTitle("Curiosity Obstacle Crossing on SCM");
        vis_irr->Initialize();
        vis_irr->AddLogo();
        vis_irr->AddSkyBox();
        vis_irr->AddCamera(ChVector<>(-1, 1, 3), ChVector<>(-5, 0.0, 0.0));
        vis_irr->AddTypicalLights();
        vis_irr->AddLightWithShadow(ChVector<>(-5.0, 8.0, -0.5), ChVector<>(-1, 0, 0), 100,
1, 35, 85, 512,
```

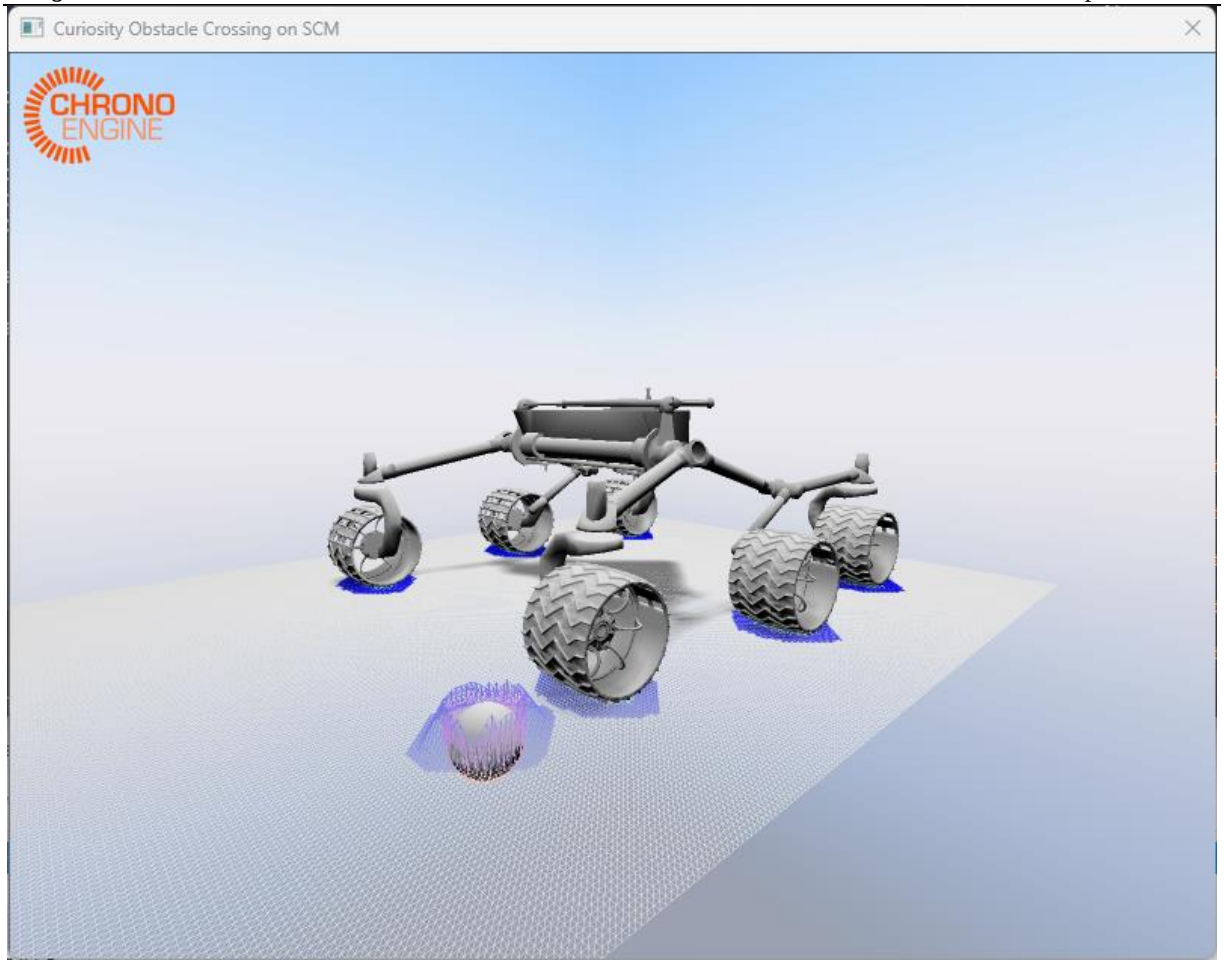
```
            ChColor(0.8f, 0.8f, 0.8f));
        vis_irr->EnableShadows();

        vis = vis_irr;
#endif
        break;
    }
    default:
    case ChVisualSystem::Type::VSG: {
#ifdef CHRONO_VSG
        auto vis_vsg = chrono_types::make_shared<ChVisualSystemVSG>();
        vis_vsg->AttachSystem(&sys);
        vis_vsg->SetCameraVertical(CameraVerticalDir::Y);
        vis_vsg->SetWindowSize(800, 600);
        vis_vsg->SetWindowTitle("Curiosity Obstacle Crossing on SCM");
        vis_vsg->AddCamera(ChVector<>(-1.0, 1.0, 3.0), ChVector<>(-5.0, 0.0, 0.0));
        vis_vsg->AddGuiColorbar("Pressure yield [kPa]", 0.0, 20.0);
        vis_vsg->Initialize();

        vis = vis_vsg;
#endif
        break;
    }
    }
```

Cilj simulacije je dobiti željene izlazne podatke, zbog čega je potrebno stvoriti tekstualnu datoteku u koju će se nakon simulacije ti podaci zapisati. Unutar simulacijske petlje definiramo koje točno podatke želimo zapisati u tekstualnu datoteku, a to su sile u kotačima koji prelaze preko prepreke te brzinu rovera. Zatvaranjem tekstualne datoteke završava kod simulacije.

```
const bool output = true;  // Set to true if you want to output data
const std::string out_dir = "C:\\Users\\Desktop\\Margareta -
diplomski\\Moj_projekt\\Build\\Debug";  // Change this to your desired output directory

if (output) {
    if (!filesystem::create_directory(filesystem::path(out_dir))) {
        std::cout << "Error creating directory " << out_dir << std::endl;
        return 1;
    }
}

// Open a text file for writing
std::ofstream txt_file(out_dir + "/output.txt");
```

```
if (!txt_file.is_open()) {
    std::cerr << "Error opening output.txt" << std::endl;
    return 1;
}
    // Simulation loop
    while (vis->Run()) {
#if defined(CHRONO_IRRLICHT) || defined(CHRONO_VSG)
        vis->BeginScene();
        vis->Render();
        vis->EndScene();
#endif

        if (output) {
            // Write drive torques of all six wheels into the file
            txt_file
                << rover.GetChassisVel() << " "
                << rover.GetWheelAppliedForce(CuriosityWheelID::C_LF) << " "
                << rover.GetWheelAppliedForce(CuriosityWheelID::C_LM) << " "
                << rover.GetWheelAppliedForce(CuriosityWheelID::C_LB) << std::endl;
        }

        rover.Update();
        sys.DoStepDynamics(time_step);
    }

    // Close the file
    txt_file.close();

    return 0;
}
```

Vizualni prikaz simulacije prikazan je na slici 26.



Slika 26. Vizualni prikaz simulacije

6.3.1. Rezultati i obrada rezultata

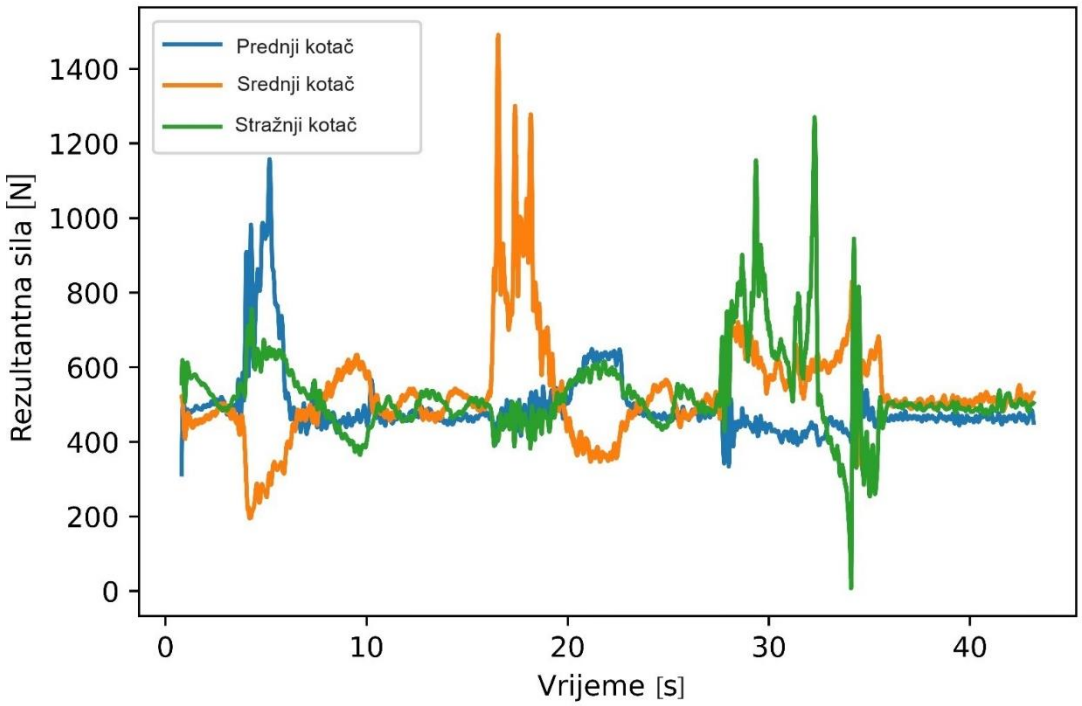
Simulacija je trajala 40 sekundi te je definiran korak 0,001 s. Dobiveno je 40 tisuća rezultata samo za jednu komponentu svake sile, što znači da je stvarni broj podataka 120 tisuća. Obrada rezultata odrađena je na isti način kao i u prethodnoj simulaciji.

6.4. Usporedba rezultata simulacija i postojećeg eksperimenta

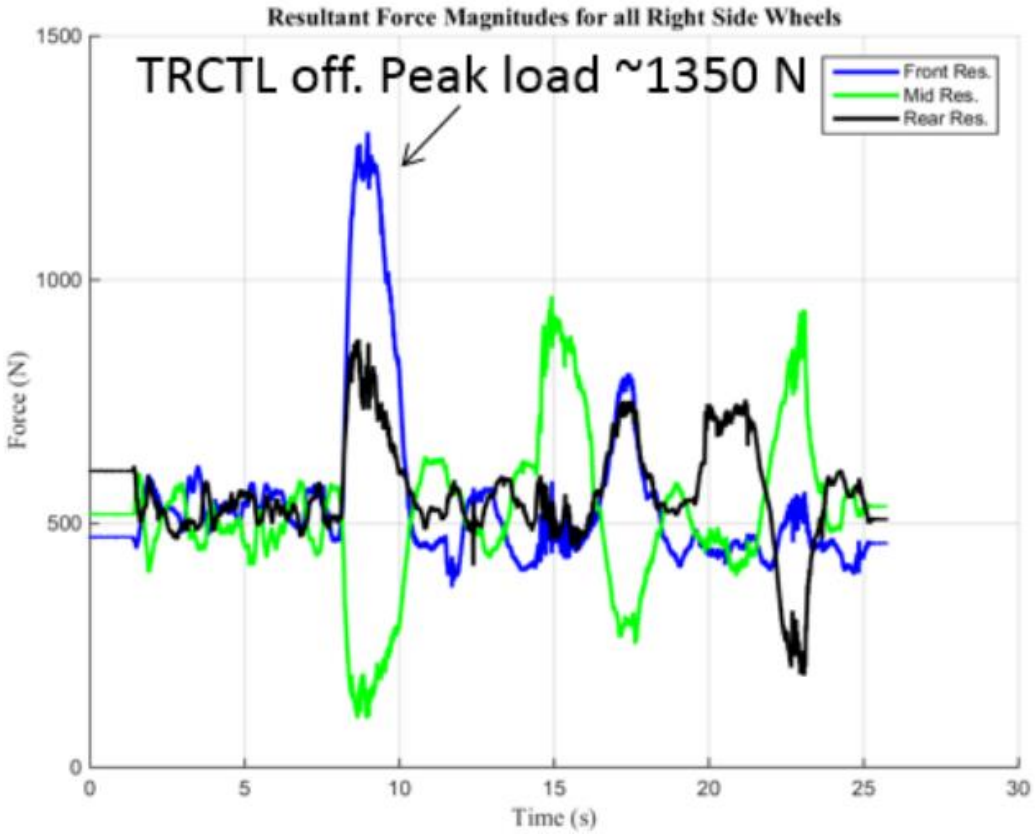
Kako je već navedeno, rezultati simulacije biti će uspoređeni s postojećim eksperimentom prikazanim u literaturi [7]. Obzirom na prednosti i mane korištenja ove vrste terena, napraviti će se usporedba rezultantnih sila kotača Curiosity rovera i Scarecrow-a.

6.4.1. Usporedba rezultata simulacije Scarecrow-a i eksperimenta

Dijagram rezultantnih sila u svim kotačima prikazan je na slici 27, a dijagram rezultantnih sila prikazan je na slici 28.

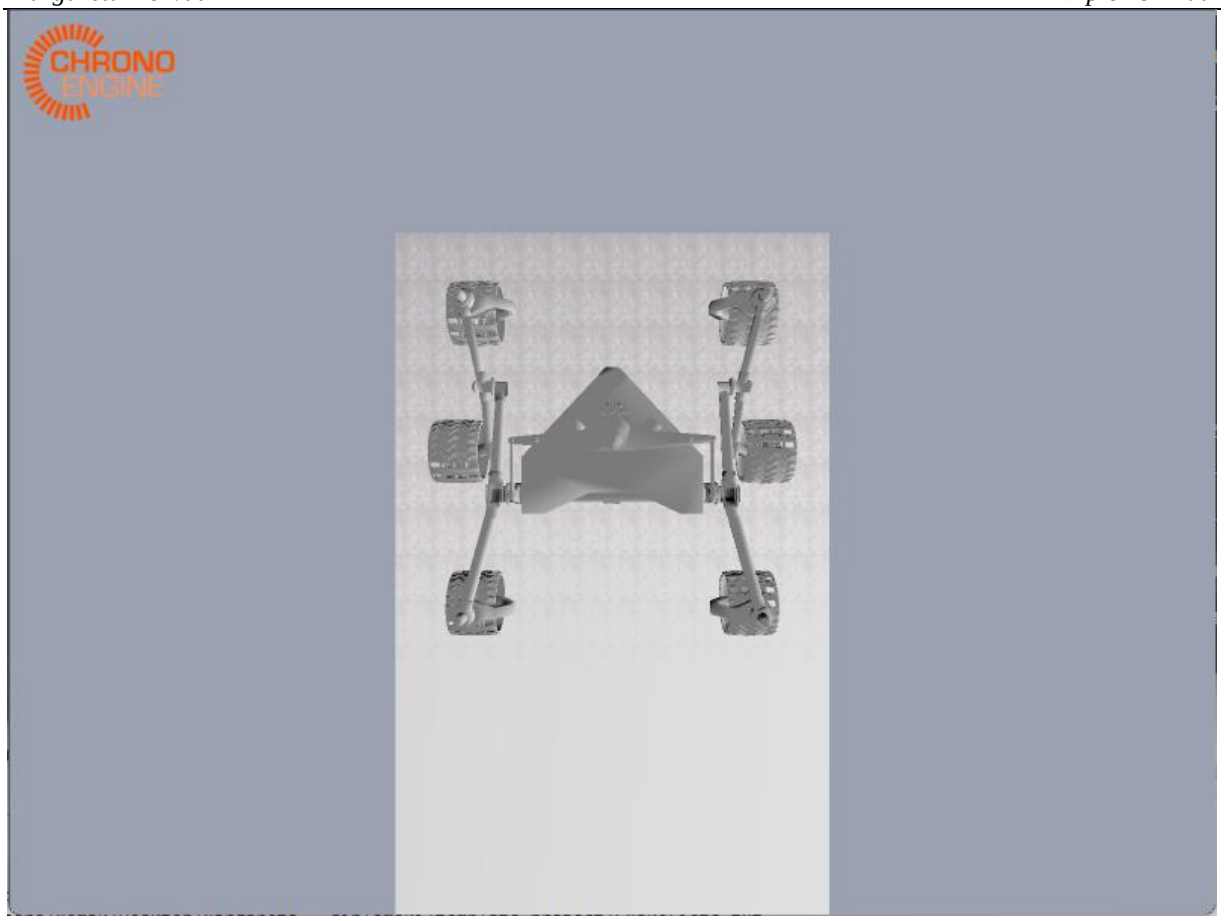


Slika 27. Dijagram rezultantnih sila dobivenih simulacijom Scarecrow-a



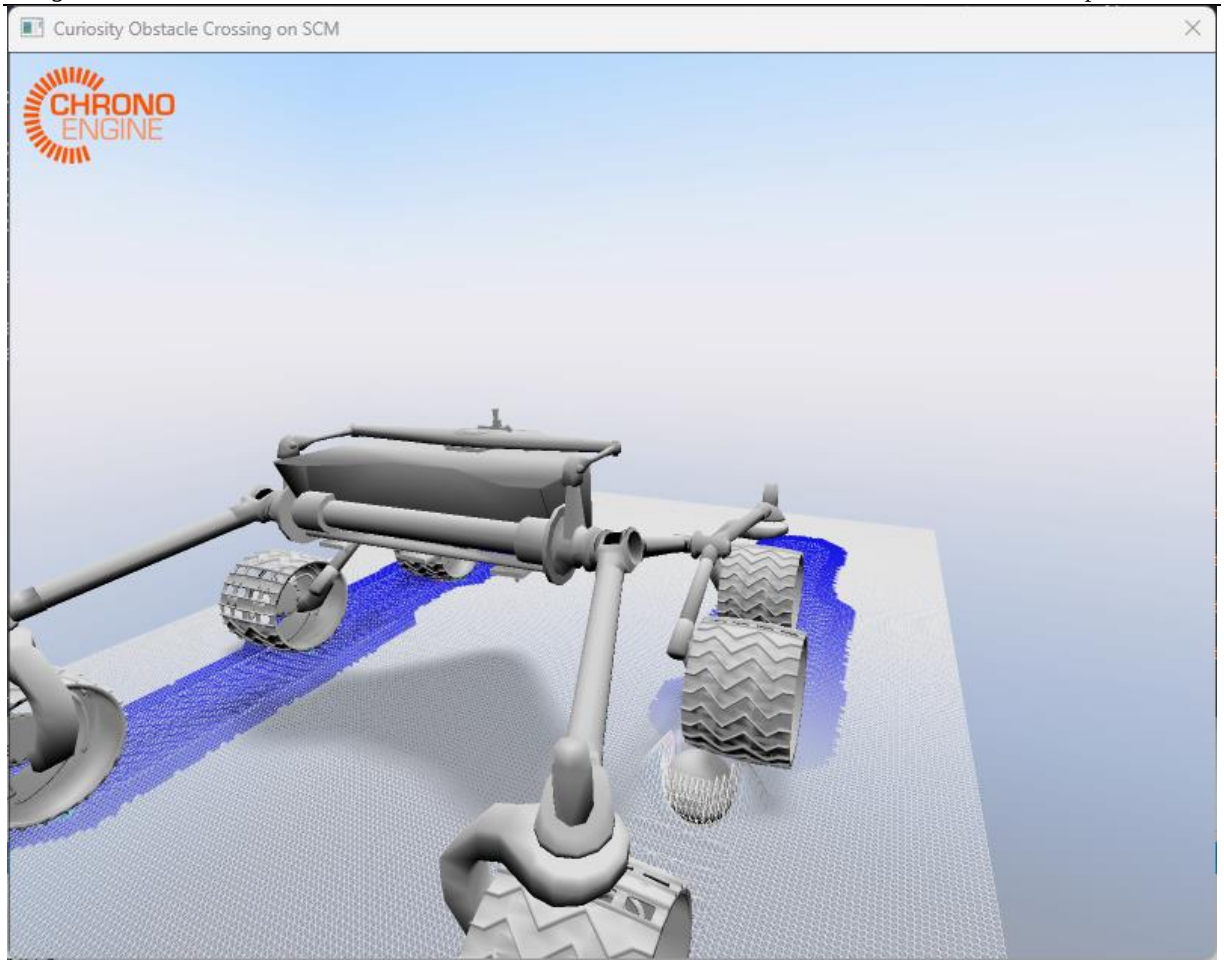
Slika 28. Rezultantne sile dobivene eksperimentom [7]

Usporedbom ova dva dijagrama vidimo mnoge različitosti, ali i neke sličnost, za što postoji mnogo razloga. Zadana masa CAD modela iznosi 280 kilograma, dok je masa u eksperimentu 318 kilograma zbog čega se razlikuju sile u smjeru y-osi. Jedna od sličnosti je to što se vidi značajan skok, odnosno pad, u iznosima sila kada pojedini kotač prelazi preko prepreke. Prvi karakteristični skok odnosi se na prednji kotač te se vidi da u tom trenutku sile srednjeg kotača padaju. Drugi karakterističan skok odnosi se na srednji kotač kada dođe na prepreku, zbog čega su sile prednjeg i stražnjeg približnog iznosa. Isti obrazac vidi se na dijagramu eksperimenta. Treći karakteristični skok odnosi se na sile stražnjeg kotača. Nakon što stražnji kotač prijeđe prepreku, sila naglo pada te se povećava sila srednjeg kotača. Na temelju slike 28, na kojoj se nalazi vizualni prikaz simulacije može se uočiti da su kotači na različitim udaljenostima od tijela rovera, što znači da će kotači pod različitim kutovima doći na prepreku obzirom da je prepreka postavljena da se nalazi na sredini prednjeg kotača. Slika 29 preuzeta je iz simulacije kada se Scarecrow kreće po ravnom terenu. Srednji kotač je najudaljeniji od tijela što znači da će na prepreku doći sa rubom kotača, zbog čega je vidljiv veći skok iznosa sile. Također, prednji i stražnji kotač nisu postavljeni na jednakim udaljenostima pa će sile stražnjeg kotača biti veće nego sile prednjeg kotača.



Slika 29. Simulacija kretanja Scarecrow-a na ravnom terenu

Obzirom da je srednji kotač došao na prepreku pod drukčijim kutom nego prednji kotač, događa se pomak rovera po z-osi što uvjetuje pomicanje stražnjeg kotača u odnosu na početnu konfiguraciju. Dolazak srednjeg kotača na prepreku prikazan je na slici 30, gdje se može vidjeti kako se prepreka ne nalazi na sredini kotača kako je to napravljeno za prednji kotač.



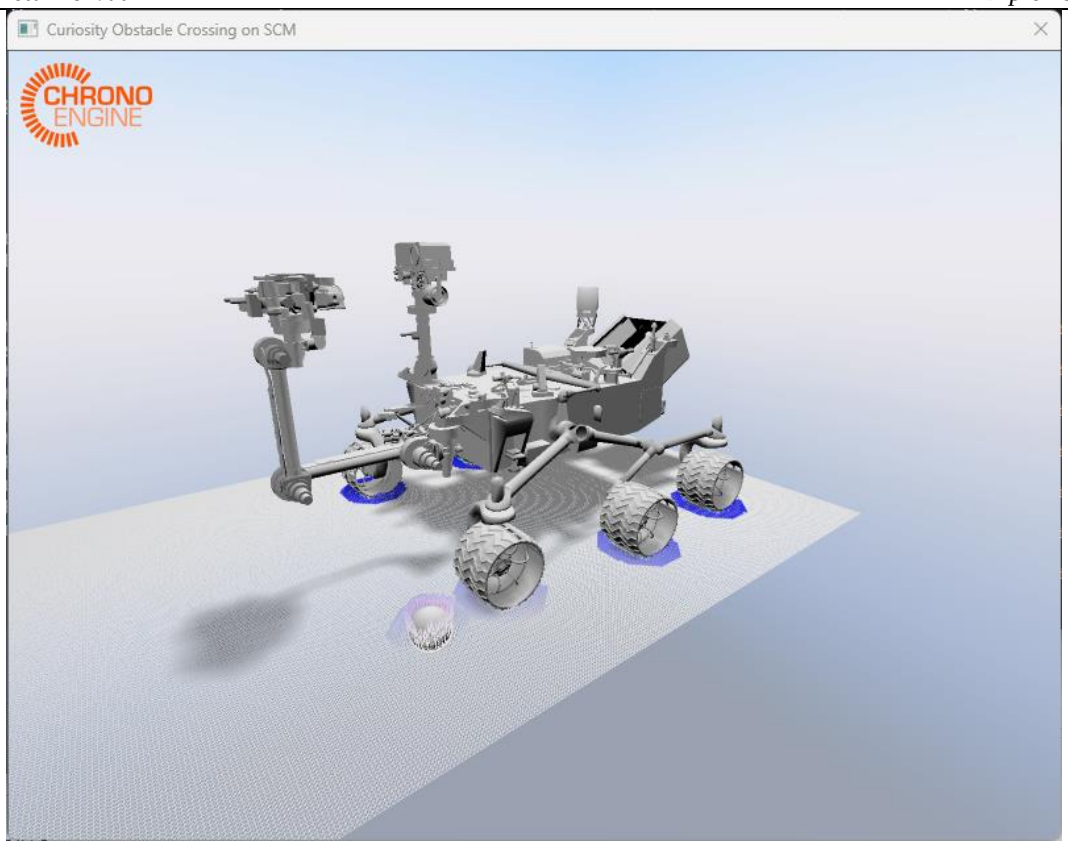
Slika 30. Prijelaz srednjeg kotača preko prepreke

Tijekom cijele simulacije vidljive su oscilacije, a na samom kraju, odnosno nakon što su svi kotači prešli prepreku, sile u svim kotačima se stabiliziraju.

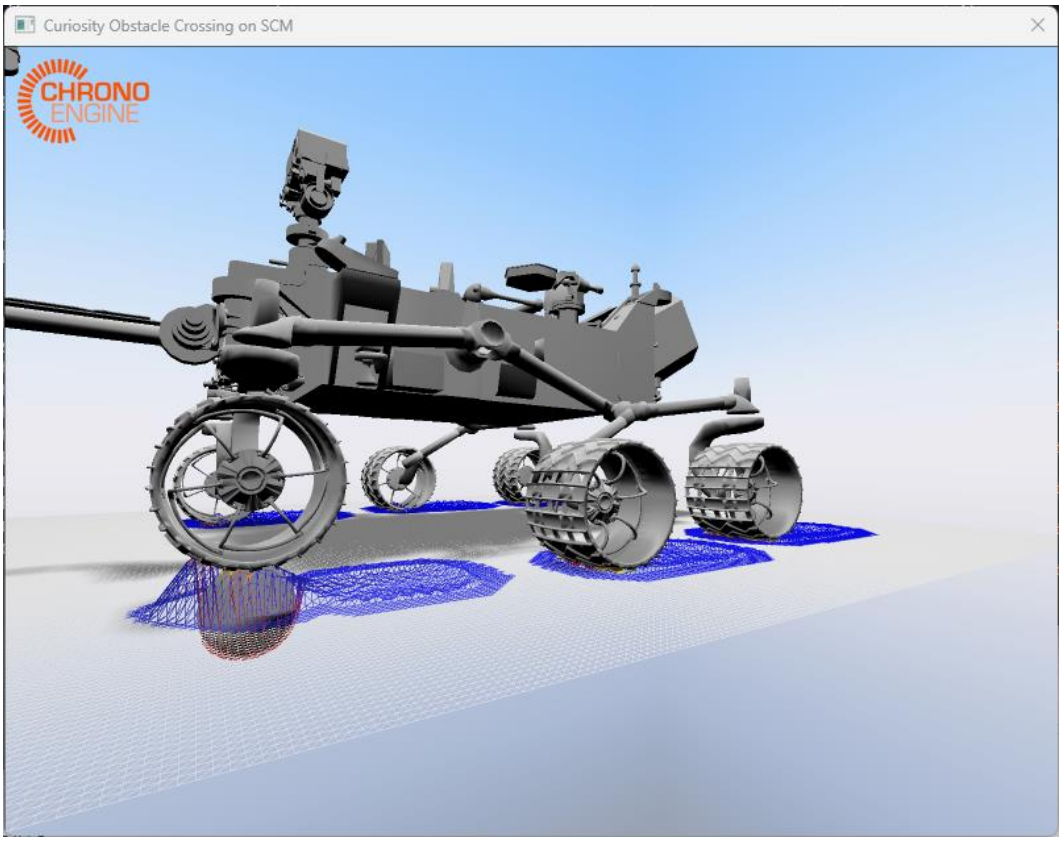
6.4.2. Usporedba rezultata simulacije Scarecrow-a i Curiosity rovera

Kako bi dobili nove rezultate, potrebno je izmijeniti dio koda tako da umjesto Scacrecrow-a koristi Curiosity rover.

Slika 31 prikazuje početak simulacija, a slika 32 prelazak prednjeg kotača preko prepreke.

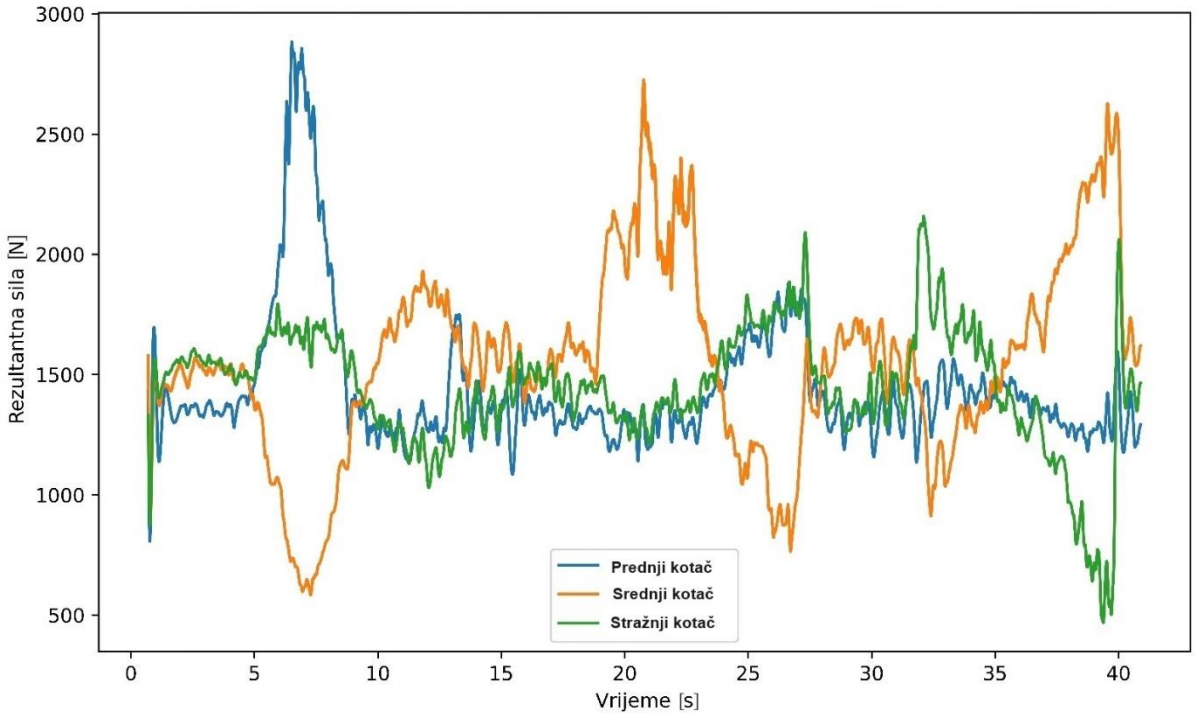


Slika 31. Curiosity rover u simulaciji



Slika 32. Prelazak Curiosity rovera preko prepreke

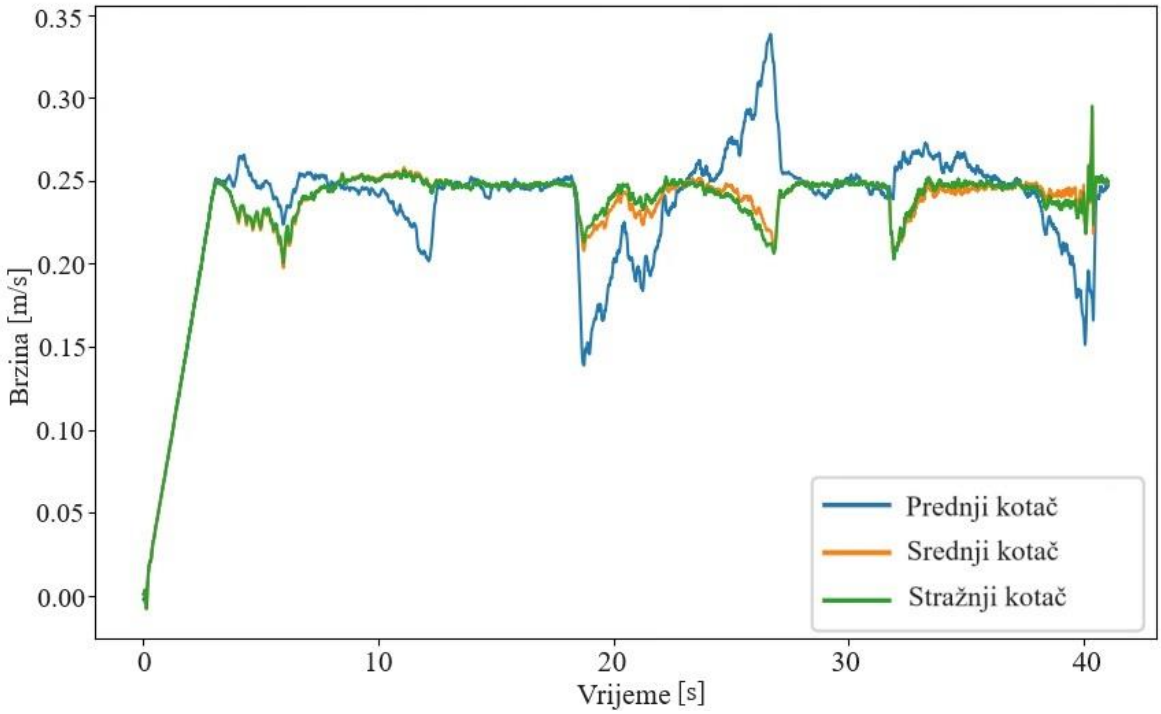
Dijagram rezultantnih sila Curiosity rovera prikazan je na slici 33, a usporediti ćemo ga s dijagramima na slici 27 i 28.



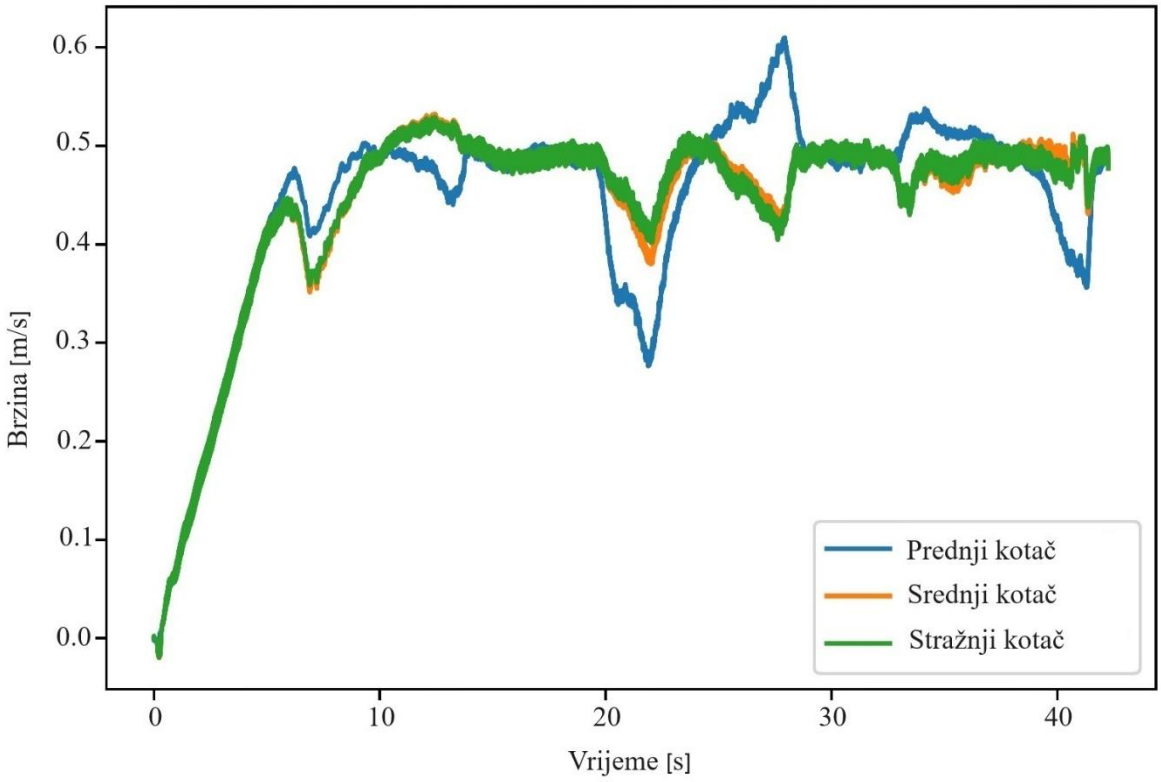
Slika 33. Dijagram rezultantnih sila dobivenih simulacijom Curiosity rovera

Na slici 33 vidljive su puno veće oscilacije u samom početku simulacije, a razlog tome je veća masa rover. Iako su iznosi sila neusporedivi, tijekom simulacije vide se karakteristični skokovi i padovi kao i kod dijagrama simulacije Scarecrow-a (slika 28). Karakteristični skokovi i padovi sada su još izraženiji te više sliče onima na dijagramu sa slike . Prvi karakteristični skok dogodi se kada prvi kotač naiđe na prepreku, kada sila srednjeg kotača naglo padne. Drugi karakteristični skok dogodi se kada srednji kotač udari u prepreku, gdje sile prednjeg i stražnjeg kotača podjednako iznose. U trenutku kada srednji kotač prijeđe preko prepreke, iznos sile naglo pada, a sile prednjeg i stražnjeg kotača se povećavaju. Treći karakteristični skok dogodi se kada stražnji kotač naiđe na prepreku. U tom trenutku sila u srednjem kotaču pada. Nadalje, nakon što stražnji kotač prijeđe prepreku, sila stražnjeg kotača naglo pada, a sila srednjeg kotača naglo raste. Nakon što su svi kotači prešli preko prepreka, sile se stabiliziraju te nema više velikih oscilacija. Obzirom na obradu podataka u kojoj je bilo potrebno smanjiti broj rezultata, moguće je da su se neki podaci jednostavno izgubili te da je to razlog zašto neki ekstremi nisu prikazani na sličan način. Obje simulacije potrebno je provesti više puta kako bih vidjeli jesu li to stvarno podaci ili samo iznimke.

Također, usporediti ćemo dijagrame brzina u kotačima koji prelaze na prepreku. Dijagram brzina za Scarecrow nalazi se na slici 34, a dijagram brzina za Curiosity rover na slici 35.



Slika 34. Dijagram brzina Scarecrow-a



Slika 35. Dijagram brzina za Curiosity rover

Na oba dijagrama vidimo sličan obrazac ponašanja, odnosno što se događa s brzinama u trenucima kada kotači dolaze na prepreku. Prvo se ubrzavaju od 0 do određene brzine te nakon toga u sva tri kotača brzina pada zato što prednji kotač nailazi na prepreku te raste kada prelazi preko nje. Isto se dogodi u trenucima kada preostala dva kotača dolaze na prepreku. Ravni dijelovi na dijagramima nam prikazuju brzine kada se roveri kreću samo po podlozi.

7. Zaključak

U radu opisan je programski paket Project Chrono koji je korišten za izradu računalnog modela odabranog rovera. Nadalje, rad se fokusira na Curiosity rover kao osnovu za izradu numeričke simulacije u posljednjem poglavlju. U tom poglavlju, opisan je napisani kod simulacije s opisom pojedinih linija te su rezultati simulacije uspoređeni s eksperimentalnim mjerenjima provedenim na stvarnom roveru. Sljedeće poglavlje analizira osnovne značajke sustava ovjesa koji se koriste u današnjim svemirskim roverima, s posebnim naglaskom na detaljnoj analizi najzastupljenijeg mehanizma, Rocker-bogie. U sklopu rada, provedena je statička analiza kako bi se bolje razumjelo što je potrebno za izradu jednog takvog mehanizma, uz opisivanje prednosti i mana njegove primjene. Nadalje, rad obuhvaća analizu dinamike čestica i dinamiku krutih tijela, pri čemu su opisani modeli kontaktne dinamike, formulacije metode diskretnih elemenata temeljene na glatkom kontaktu te diferencijalna varijacijska nejednakost. Ovaj sveobuhvatni pristup omogućuje cjelovitu analizu sustava s krutim tijelima te njihovu dinamiku, pružajući dublje razumijevanje mehanizama i procesa koji se koriste u svemirskim istraživanjima.

Numeričko modeliranje je temelj niza disciplina, uključujući inženjerstvo, fiziku, građevinu, znanost o okolišu i slično. Njegov značaj leži u njegovoj sposobnosti da simulira zamršene fenomene koji bi mogli biti nepraktični ili neizvedivi za istraživanje samo tradicionalnim eksperimentalnim metodama. Kroz numeričke modele znanstvenici i inženjeri mogu predvidjeti ponašanje sustava u različitim uvjetima, nudeći neprocjenjive uvide ključne za projektiranje i optimiziranje procesa i tehnologija. Točnim predviđanjem performansi konstrukcija i sustava prije izgradnje, numeričko modeliranje štedi vrijeme i resurse u razvojnim fazama. Nadalje, pruža siguran i troškovno učinkovit način za istraživanje scenarija koji bi inače mogli biti opasni ili financijski opterećujući za proučavanje u fizičkim eksperimentima. Preciznost koju nudi numeričko modeliranje pomaže u razvoju inovativnih rješenja za izazove iz stvarnog svijeta, dok također služi kao platforma za testiranje hipoteza, potvrđivanje znanstvenih teorija i usavršavanje postojećeg znanja. U sektorima od klimatskih znanosti do vremenske prognoze, numeričko modeliranje služi kao nezamjenjiv alat za razumijevanje složenih sustava i projiciranje budućih trendova i rezultata. U konačnici, njegova svestranost i učinkovitost čine numeričko modeliranje neprocjenjivom vrijednošću u unapređenju znanstvenog razumijevanja, poticanju tehnoloških inovacija i usmjeravanju informiranog donošenja odluka u raznim područjima i industrijama.

Na primjeru svemirskog rovera, numerička simulacija od velikog je značaja obzirom na kompleksnost svemirskih misija, neistraženosti drugih planeta i velike financijske troškove. Iako je s ubrzanim razvojem računala i računalnih komponenti protekla dva desetljeća izrada numeričke simulacije znatno ubrzana, i dalje nailazimo na mnoge zapreke. Numerička simulacija koja može reprezentativno prikazati dinamiku sustava velikog broja čestica predstavlja veliki zadatak te je njezina izrada jako ograničena zbog slabosti današnjih računala. Iz toga razloga, nakon što su rezultati prve simulacije prikazali nereprezentativne rezultate, u ovom radu korišten je teren u obliku mreže, odnosno korišten je kontaktni model tla. Kada bi bilo moguće tlo prikazati s česticama stvarnih veličina zrna pijeska, dobili bi preciznije podatke. Iznosi dobivenih rezultata numeričke simulacije ne poklapaju se s onim iznosima dobivenim mjerenjem stvarnog rovera u eksperimentu, no uočili smo sličnosti na temelju kojega možemo odrediti obrazac. Paralelnim prikazivanjem sila za sve kotače možemo točno odrediti kako se koji kotač ponaša u trenutku dolaska na prepreku, odnosno što se u tom trenutku dogodi s preostala dva kotača.

Literatura

[1] NASA, https://mars.nasa.gov/mars2020/multimedia/raw-images/ (pristupljeno 10.3.2024.)

[2] Project Chrono, https://projectchrono.org/ (pristupljeno 5.1.2024.)

[3] NASA, https://mars.nasa.gov/msl/spacecraft/rover/summary/ (pristupljeno 10.3.2024.)

[4] Wikipedia, https://en.wikipedia.org/wiki/Curiosity_(rover) (pristupljeno 10.3.2024.)

[5] NASA, https://mars.nasa.gov/msl/home/ (pristupljeno 11.3.2024.)

[6] NASA, https://mars.nasa.gov/resources/3456/scarecrow-climbs-rocks/?site=msl (pristupljeno 11.3.2024.)

[7] Toupet, O., Biesiadecki, J., Rankin, A., Steffy, A., Meirion-Griffith, G., Levine, D., Schadegg, M., Maimone, M., (2018.) *Traction control design and integration onboard the Mars science laboratory curiosity rover*, 2018 IEEE Aerospace Conference, Big Sky, MT, USA, str. 1-20.

[8] NASA, https://mars.nasa.gov/news/an-algorithm-helps-protect-mars-curiositys-wheels/ (pristupljeno 11.3.2024.)

[9] Hindawai, https://www.hindawi.com/journals/ijae/2016/5181097/ (pristupljeno 11.3.2024.)

[10] Nayar, H., Kim, J., Chamberlain-Simon, B., Carpenter, K., Hans, M., Boettcher, A., Meirion-Griffith G., Wilcox B., Bittner B., (2019.) *Design optimization of a lightweight rocker–bogie rover for ocean worlds applications*, International Journal of Advanced Robotic Systems, str. 1.-20.

[11] Jotheess, S., (2017.) *Design and Optimization of a Mars Rover's Rocker-Bogie Mechanism*, IOSR Journal of Mechanical and Civil Engineering, vol. 14, no. 5, str. 74-79.

[12] Cordes, F., Kuehn, D., Oekermann, C., Babu, A., Stark, T., Kirchner, F., (2014.), *An Active Suspension System for a Planetary Rover*, i-SAIRAS.

[13] Wikipedia, https://en.wikipedia.org/wiki/Rocker-bogie (pristupljeno 13.3.2024.)

[14] Eiber, A., Terze, Z., (2017.) *Dynamics of Multibody Systems*, Fakultet strojarstva i brodogradnje.

[15] 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.

[16] Melanz, D., Fang, L., Jayakumar, P., Negrut, D., (2017.), *A comparison of numerical methods for solving multibody dynamics problems with frictional contact modeled via differential variational inequalities*, Elsevier, vol. 320, str. 668-693.

[17] Nuñez-Quispe, J., Lleren-Sernaque, J., Lara-Chavez, E., (2021.) *Mechanical Design of a ROVER prototype for Exploration tasks on Mars: Structural and Transient Dynamics simulation analysis*, 2021 IEEE MIT Undergraduate Research Technology Conference (URTC).

[18] German Research Center for Artificial Intelligence GmbH, Robotics Innovation Center https://robotik.dfki-bremen.de/en/research/robot-systems/sherpa (pristupljeno 13.3.2024.)

[19] 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*, Elsevier, vol. 65, str. 1.-13.

[20] Project Chrono, https://www.projectchrono.org/assets/slides_3_0_0/3_Contact/1_Chrono_TheoryContact.pdf (pristupljeno 13.3.2024.)

[21] Tasora, A., Mangoni, D., Negrut, D., (2017.) *An Overview of the Chrono Soil Contact Model (SCM) Implementation*.