

Dizajn podatljivih mehanizama pomoću numeričke optimizacije

Katalenić, Mihael

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:442587>

Rights / Prava: [Attribution-NonCommercial 4.0 International/Imenovanje-Nekomercijalno 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-01-05**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski sveučilišni studij strojarstva

Diplomski rad

**DIZAJN PODATLJIVIH
MEHANIZAMA POMOĆU
NUMERICKE OPTIMIZACIJE**

Rijeka, svibanj 2023.

Mihael Katalenić
0069075480

Posveta

Ludbreškim bregima, zeleni Bednji, dlakavim anđelima i malome Mišu velkoga srca.

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski sveučilišni studij strojarstva

Diplomski rad

**DIZAJN PODATLJIVIH
MEHANIZAMA POMOĆU
NUMERICKE OPTIMIZACIJE**

Mentor: prof. dr. sc. Stefan Ivić

Komentor: prof. dr. sc. Marko Čanađija

Rijeka, svibanj 2023.

Mihael Katalenić
0069075480

Umjesto ove stranice umetnuti zadatak
za završni ili diplomski rad

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad. Koristeći stečena znanja tijekom studija te koristeći se navedenom literaturom. Rad je izrađen prema zadatku Povjerenstva za diplomske radove diplomskog sveučilišnog studija strojarstva i Uputama za pisanje diplomskih radova, pod vodstvom mentora prof. dr. sc. Stefana Ivića i komentora prof. dr. sc. Marka Čanađije.

Rijeka, svibanj 2023.

Mihael Katalenić

Zahvala

Zahvaljujem se mentoru prof. dr. sc. Stefanu Iviću i komentoru prof. dr. sc. Marku Čanadiji za vodstvo, strpljenje i potporu tijekom pisanja diplomskog rada.

Posebno se zahvaljujem roditeljima mami Steli, tati Draženu i potatku Draženu kao i cijeloj svojoj obitelji na nesebičnoj potpori, strpljenju i ukazanom povjerenju tijekom studija. Posebno hvala teti Jeleni na inspiraciji za napredovanje u akademskom životu i na poticanju znanstvene radoznalosti.

Hvala prijateljima Marku, Ivani, Eli i Niku na korisnim savjetima tijekom izrade rada, kao i ostalim prijateljima na besprijekornom i ponekad apsurdnom optimizmu i predivnim nezaboravnim studentskim danima.

Veliko hvala djevojci Anji bez čije ljubavi, nesebičnog odricanja i pomoći moji studentski dani i život ne bi bili potpuni.

Sadržaj

Popis slika	x
Popis tablica	xviii
1 Uvod	1
1.1 Metodologija i korišteni alati	2
1.1.1 Korišteni alati	2
2 Podatljivi mehanizmi	4
2.1 Prednosti podatljivih mehanizama	5
2.2 Mane podatljivih mehanizama	5
2.3 Primjena podatljivih mehanizama	7
3 Modeliranje i simuliranje podatljivih mehanizama	9
3.1 Metoda konačnih elemenata	9
3.1.1 Matrica polja i matrica interpolacijskih funkcija	10
3.1.2 Jednodimenzijski linearni elementi	11
3.1.3 Gredni konačni element	17
3.2 Postavljanje modela	20
3.2.1 Bazna rešetka, oslonci i opterećenja	21
3.3 Programski paket <i>Calculix</i>	23

4 Dizajn podatljivih mehanizama pomoću optimizacije	28
4.1 Optimizacijske varijable	28
4.2 Ciljevi	30
4.3 Ograničenja	31
4.4 Multi-Scale Grid Search algoritam	32
4.4.1 Početni odabir smjera kretanja lokalne optimizacije	32
4.4.2 Nastavak kretanja optimizacije	33
4.4.3 Ugušćivanje mreže	33
4.5 Višefazna optimizacija i redukcija broja optimizacijskih varijabli	37
5 Primjeri	39
5.1 Primjer 1. - podatljivi invertor pomaka	39
5.1.1 Mogući oblik podatljivog mehanizma	40
5.2 Primjer 2. - Podatljiva hvataljka	43
5.2.1 Postavljanje slučaja	43
6 Rezultati	45
6.1 Rezultati optimizacije prvog primjera	45
6.1.1 Iznos funkcije cilja kod lokalnih optimizacija svakog stupnja optimizacije	45
6.1.2 Iznos mase kod lokalnih optimizacija svakog stupnja optimizacije	49
6.1.3 Rješenja optimizacija unutar svakog stupnja algoritma	52
6.2 Rezultati drugog primjera	59
6.2.1 Iznos funkcije cilja kod lokalnih optimizacija svakog stupnja optimizacije	59
6.2.2 Rješenja optimizacija unutar svakog stupnja algoritma	65
6.3 Odabir dizajna mehanizma i izrada prototipa mehanizama	68

Sadržaj

6.3.1	3D ispis podatljivih mehanizama	68
6.3.2	Primjer 1. - Invertor pomaka	69
6.3.3	Primjer 2. - Hvataljka	73
7	Zaključak	77
	Bibliografija	79
	Sažetak	81

Popis slika

2.1	Primjer podatljivih plastičnih klješta za odstranjivanje udica izrađenih lijevanjem plastike. [1]	7
2.2	Primjer mikroskopskih metalnih podatljivih mehanizama hvataljke koja aksijalni pomak izazvan djelovanjem elektromagneta na dio mehanizma pretvara u zatvaranje “krakova” mehanizma [2].	8
3.1	Način podjele jednodimenzijuskog područja na podintervale.	11
3.2	Prikaz jednodimenzijuskog elementa.	12
3.3	Primjer potpornih greda. Polje a) označava polje bez dijagonalnih potpora, omeđeno s četiri grede. Polje s križnom potporom b) i polja s dijagonalnim potporama c) i d)	22
3.4	Primjer bazne rešetke s prednje naginjućim dijagonalama te s apstraktno definiranim silama F , traženim pomacima D1 , D2 , D3 , nepomičnim osloncom C1 , pomičnim osloncom C2 i uklještenjem C3 . S ovako definiranom baznom rešetkom možemo pokrenuti optimizacijski proces koji će promjenom debljina greda postići tražene pomake mehanizma.	22
3.5	Ekspanzija $B32$ 1D elementa u 3D element $C3D20$. Točke označene brojevima pripadaju ekspanziranom obliku $C3D20$ dok točke označene slovom G i brojkom pripadaju ne ekspanziranom obliku $B32$. [3]	24
3.6	3x3x3 integracijska shema točaka kod heksaedarskih elemenata. [3] .	25

3.7	Primjer prikaza rezultata korištenjem <i>GraphiX</i> programa.[3] Prikazuje se iznos pomaka čvorova u smjeru Y osi jednostavnog podatljivog mehanizma.	27
4.1	Prikaz mreže diskretiziranih vrijednosti optimizacijskih varijabli X_1 i X_2 . Gornje i donje granice mreže označene su UB kao <i>upper bound</i> i LB kao <i>lower bound</i> . Crvene linije predstavljaju početnu šablonu tj. točke na temelju kojih će se računati početni vektor kretanja optimizacije. Točka koja je označena crvenom bojom predstavlja početnu točku optimizacije ($T_{i,j}$), dok ostale točke ($T_{i\pm 1,j\pm 1}$) predstavljaju moguću slijedeću početnu poziciju unutar iteracije. ΔX predstavlja razmak između diskretiziranih vrijednosti optimizacijskih varijabli koji unutar optimizacijskog stupnja ostaje konstantan. ΔY prikazuje smjer kretanja u slučaju kad su rezultati funkcije cilja na pozicijama $T_{i+1,j}$ i $T_{i,j+1}$ bolji od svih ostalih rezultata te od iznosa funkcije cilja u početnoj točki $T_{i,j}$	35
4.2	Nastavak kretanja optimizacije dvostrukim korakom u smjeru prvog vektora smjera.	36
4.3	Šablona koja se sastoji od m evaluacija.	36
4.4	Prikazano je ugušćivanje mreže u slučaju postizanja lokalnog minimuma kod grube mreže. Također prikazan je i način na koji se nanovo diskretizirano područje pomiče kod evaluacije točke na rubu diskretizirane domene.	36
5.1	Početna mreža mehanizma klješta	39
5.2	Jednostavni dizajn potrebnog podatljivog mehanizma. Zbog geometrijske simetrije mehanizma, grede označene istim brojevima istih su debljina.	41

Popis slika

5.3	Pomaci točaka krutog mehanizma koji se sastoji od 6 greda međusobno spojenim zglobovima. Postavljanjem opruge krutosti k postavlja se uvijet vraćanja mehanizma u početni položaj. Same debljine greda nemaju nikakav utjecaj na djelovanje mehanizma (uzimajući u obzir da se grede ne saviju pod izazvanim opterećenjem.)	42
5.4	Pretpostavljeni oblik deformiranog podatljivog mehanizma. Opruga postaje nepotrebna, kako umjesto da se energija pohrani u obliku unutarnje energije deformacije opruge, sada zapravo cijeli mehanizam sudjeluje u pohrani energije u obliku unutarnje energije deformacije podatljivog mehanizma.	42
5.5	Prikaz zamišljenih granica mehanizma, te početni i krajnji uvijeti. .	43
5.6	Prikaz zamišljenih granica mehanizma, te početni i krajnji uvijeti. .	44
6.1	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija prvog stupnja algoritma. Vrijednosti svih lokalnih optimizacija su konvergirale. Ovaj stupanj lokalnih optimizacija optimizira vektor od 28 varijabli.	46
6.2	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija drugog stupnja optimizacije. Ovaj stupanj optimizira vrijednosti 19 optimizacijskih varijabli. Za razliku od prvog stupnja, više rezultata počinje težiti istoj vrijednosti.	46
6.3	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija trećeg stupnja algoritma. Ovaj stupanj optimizira vrijednosti 18 optimizacijskih varijabli. Za razliku od prva dva stupnja, rezultati vrlo brzo konvergiraju, te samo dvije lokalne optimizacije nisu konvergirale.	47
6.4	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija četvrtog stupnja algoritma. Stupanj optimizira vrijednosti 16 optimizacijskih varijabli. Za razliku od prethodnog stupnja, vidljivo je kako veliki broj lokalnih optimizacija nije konvergiralo u blizinu minimalne vrijednosti funkcije cilja.	47

Popis slika

6.5	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija petog stupnja algoritma. Algoritam je zadržao sve optimizacijske varijable, te nastavlja optimizaciju s 16 optimizacijskih varijabli.	48
6.6	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija šestog stupnja algoritma. Kao i kod prethodnog stupnja, optimizira se 16 optimizacijskih varijabli. Na temelju zadržavanja svih 16 optimizacijskih varijabli kroz protekla 3 stupnja algoritma, može se pretpostaviti kako daljnjeg smanjenja optimizacijskog vektora neće biti, te se algoritam zadržava.	48
6.7	Graf prikazuje iznos mase mehanizma kod prvog stupnja algoritma.	49
6.8	Graf prikazuje iznos mase mehanizma kod drugog stupnja algoritma.	49
6.9	Graf prikazuje iznos mase mehanizma kod trećeg stupnja algoritma.	50
6.10	Graf prikazuje iznos mase mehanizma kod četvrtog stupnja algoritma.	50
6.11	Graf prikazuje iznos mase mehanizma kod petog stupnja algoritma. .	51
6.12	Graf prikazuje iznos mase mehanizma kod šestog stupnja algoritma.	51
6.13	Prikaz rezultata prvog stupnja optimizacije. Od 10 MSGS lokalnih optimizacija, svaka je optimizacija pronašla lokalni minimum. Pokazujući kako je problem koji riješavamo svakako multimodalan s većim brojem lokalnih optimuma. Dva rješenja (lokalna optimizacija 2 i optimizacija 8) su <i>nemoguće</i> rješenja kod kojeg je oslonac koji se ne smije maknuti iz proračuna odvojen od ostatka mehanizma. Također, kod tri rješenja (5, 6, 7) grede ulaze u definirano kritično područje naprezanja.	53
6.14	Prikaz rezultata drugog stupnja optimizacije. Vidljivo je kako već sada par lokalnih optimizacija kreće u relativno istu minimalnu vrijednost. Ponovo tri lokalne optimizacije dobivaju “nemoguće” rješenje, dok dvije valjane optimizacije ulaze u kritično područje. . .	54
6.15	Prikaz rezultata trećeg stupnja optimizacije. Kod ovog stupnja vidimo kako više nema “nemogućih” rješenja, već su sva rješenja valjana. Također sve manje greda ulazi u kritično područje.	55

Popis slika

6.16	Prikaz rezultata četvrtog stupnja optimizacije. Lokalne optimizacije sve se više kreću prema jednakom rješenju (zbog manjeg broja optimizacijskih varijabli).	56
6.17	Slike prikazuju peti stupanj algoritma. Jedna od lokalnih optimizacija daje “nemoguće” rješenje	57
6.18	Posljednji stupanj prikazuje definitivan odabir krajnjeg dizajna podatljivog mehanizma. Kako se primjetilo kod rezultata funkcije cilja, u posljednjem stupnju algoritma 6 lokalnih optimizacija postiče vrlo slično rješenje. Kako se proračun zaustavlja, odabire se upravo taj dizajn mehanizma.	58
6.19	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija prvog stupnja optimizacije. Dvije lokalne optimizacije su konvergirale, ali s vrlo visokim iznosom funkcije cilja. Problem se sastoji od 47 optimizacijskih varijabli.	59
6.20	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija drugog stupnja algoritma. Kako se u drugom stupnju broj optimizacijskih varijabli smanjio samo za jednu vrijednost (46 optimizacijskih varijabli) kao i kod prvog stupnja vidimo da niti jedna optimizacija nije konvergirala.	60
6.21	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija trećeg stupnja algoritma. U ovome stupnju koristi se 16 optimizacijskih varijabli. Nagli pad broja optimizacijskih varijabli razlog je nagle konvergencije rezultata za razliku od prijašnjih optimizacija.	60
6.22	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija četvrtog stupnja optimizacije.	61
6.23	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija petog stupnja optimizacije.	61
6.24	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija šestog stupnja optimizacije.	62

Popis slika

6.25	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija sedmog stupnja optimizacije.	62
6.26	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija osmog stupnja optimizacije.	63
6.27	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija devetog stupnja optimizacije.	63
6.28	Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija desetog stupnja optimizacije. Uvjerljivo su sve lokalne optimizacije ovog stupnja konvergirale. Sve lokalne optimizacije osim jedne postigle su jednaki rezultat. Algoritam se zaustavlja.	64
6.29	Slike prikazuju rješenja prvog stupnja algoritma. Kako se moglo pretpostaviti na temelju grafova iznosa funkcije cilja, samo je jedna lokalna optimizacija uspjela postići zadane pomake iako uz moguće prekoračenje debljina greda u kritično područje.	65
6.30	Kao i kod prvog stupnja algoritma, lokalne optimizacije sve konvergiraju u drugačija rješenja. Većina lokalnih optimizacija nije postigla optimalno traženo rješenje.	66
6.31	Prikazan je treći stupanj algoritma. Nakon velikog smanjenja broja optimizacijskih varijabli većina rješenja prikazuje zadovoljene uvjete optimizacije. Kod većine rješenja doduše grede ulaze u kritično područje.	66
6.32	Prikazan je četvrti stupanj algoritma. U ovome stupnju nije moguće maknuti više optimizacijskih varijabli (greda). Micanjem bilo koje grede mičemo poveznicu između točaka s zadanim rubnim uvjetima potrebnim za rješavanje optimizacijskog problema. Svaki idući stupanj algoritma koristi jednaki broj optimizacijskih varijabli, tako da nema potrebe prikazivati ostale rezultate. Zaključuje se da je optimizator postigao zadovoljavajuće rješenje.	67
6.33	Prikazuje se dobiveni 3D računalni model jednog od zadanih mehanizama korištenjem <i>OpenSCAD</i> ovog <i>post-processinga</i>	68

Popis slika

- 6.34 Graf prikazuje odabrani dizajn mehanizma na temelju iznosa funkcije cilja. Prikazuje se rezultat 4. lokalne optimizacije 5. stupnja algoritma. Može se uočiti kako niti jedna greda ne ulazi u kritično područje naprezanja. 70
- 6.35 Slika prikazuje nedeformirani prototip odabranog mehanizma izrađen na 3D pisaču. Zbog lakšeg nanošenja opterećenja (sile) dodaje se prsten na hvatištu sile. Također, problem ispisa oslonaca riješen je postavljanjem šupljih valjaka na čvorove mehanizma na kojima se nalaze oslonci. Cijeli mehanizam se pomoću čavlića zabija u drvo, što omogućuje postavljanje oslonca na tražene čvorove. 71
- 6.36 Slika prikazuje deformirani prototip odabranog mehanizma izrađen na 3D pisaču. Uspoređujući dobiveni rezultat dizajna podatljivog mehanizma i isprintani mehanizam vidljivo je kako su fizikalni aproksimirani model i strukturalni proračun pravilno predvidjeli ponašanje fizikalnog modela. 72
- 6.37 Graf prikazuje odabrani dizajn mehanizma na temelju iznosa funkcije cilja. Prikazuje se rezultat 0. lokalne optimizacije 11. stupnja algoritma. 73
- 6.38 Nedeformirani oblik prototipa mehanizma hvataljke izrađenog na 3D pisaču. Zbog lakšeg nanošenja sile, dodan je prsten povezan s hvatištem sile. Također postavlja se i kruti okvir na koji se kruto spajaju čvorovi modela kod kojih je definirano uklještenje. 74
- 6.39 Slika prikazuje deformirani prototip mehanizma opterećenog manjom silom nego što je zadana u modelu. Vidljivo je kako dobivene deformacije mehanizma prate trend deformiranja dobiven strukturalnom analizom mehanizma. 75

- 6.40 Slika prikazuje deformirani prototip mehanizma opterećen punim iznosom sile. Zbog geometrijskih nepravilnosti uvedenih 3D ispisom deformacije polovica nisu jednake, te deformacija mehanizma nije u potpunosti osno simetrična kako je bilo predviđeno kod postavljanja primjera. Također točke na desnom kraju hvataljke nisu postigle zadani pomak. Pretpostavlja se da zbog uvjeta minimizacije mase grede na krajevima hvatalje postaju vrlo tanke (minimalna moguća veličina grede) što u matematičkom modelu ne igra nikakvu ulogu, ali kod prototipa mehanizma prouzrokuje pogrešno deformirani mehanizam. Podebljanjem opisanih greda neće se promijeniti deformacija mehanizma, ali će se osigurati konstantna paralelnost greda tijekom deformacije mehanizma (kako je predviđeno matematičkim modelom). 76

Popis tablica

3.1	Lokalne i globalne čvorne varijable	15
3.2	Iznosi funkcija oblika i njihovih derivacija	19
5.1	Konstantni parametri korišteni kod lokalnih optimizacija.	40
5.2	Konstantni parametri korišteni kod lokalnih optimizacija drugog slučaja.	44

1 Uvod

Podatljivi mehanizmi su mehanizmi kod kojih jedan od članova ili više njih koristi svojstvo velikih elastičnih deformacija kod postizanja traženih kinematskih i dinamičkih svojstava.

Ovaj rad bavi se dizajnom podatljivog mehanizma pomoću numeričke optimizacije. Kompletan proces dizajna mehanizma automatizira se korištenjem računala, uključujući: parametrizaciju geometrijskog dizajna, analizu deformiranja mehanizma te optimizaciju geometrijskih karakteristika mehanizma s ciljem postizanja zadanih pomaka dijelova mehanizma tijekom primjene opterećenja. Kako se ne zadaje jedan određeni mehanizam kod kojeg se traži optimalna struktura, rad se bavi generalizacijom optimizacijskog postupka za potrebe analize većeg broja dvodimenzionalnih podatljivih mehanizama.

Cilj rada je pronalazak relativno jednostavnog načina postavljanja podatljivog (savitljivog) mehanizma koji sa svojim zadanim početnim uvjetima (opterećenjima i osloncima) ispunjava zadane završne uvjete (pomake dijelova mehanizma na točno određene pozicije, zadržavanje udaljenosti između zadanih točaka, rotaciju dijelova mehanizma oko stacionarne točke, itd.). Treba uzeti u obzir da numerički dizajn mehanizma pomoću računala nije savršen i svako pojednostavljenje mehanizma za potrebe jednostavnijeg i bržeg proračuna omogućuje generiranje grešaka na stvarnom mehanizmu.

U potpunosti definirani mehanizam moguće je isprintati na 3D printeru, a eksperimentalnim ispitivanjem prototipa mehanizma možemo utvrditi nastale greške na mehanizmu te ih pravovremeno ispraviti. Ispis mehanizma na 3D printeru, te prethodno postavljanje ograničenja mehanizma kako bi svojim materijalnim i geometrijskim karakteristikama omogućili ispis na 3D printeru jednostavan je način proizvodnje prototipa i analize kinematskih karakteristika proizvedenog mehanizma.

1.1 Metodologija i korišteni alati

Postavlja se inženjerski problem automatiziranog dizajna podatljivog mehanizma koji na temelju zadanih karakteristika geometrije i zadanih ograničenja (oslonci, opterećenja, minimizacija mase, itd.) dizajniranim oblikom ispunjava tražene finalne pomake čvorova mehanizma. Kako bi se provela strukturalna analiza podatljivog mehanizma, odabire se numerički pristup simulacije podatljivog mehanizma pomoću *Metode konačnih elemenata*. Definirana struktura mehanizma (gredna rešetka) ostvaruje traženi oblik na temelju velikih pomaka, deformacija i rotacija rešetke dobivene mijenjanjem debljine pojedinih greda ili micanjem pojedinih greda. Određivanje oblika mehanizma može se svesti na strukturalnu optimizaciju čije su optimizacijske varijable debljine pojedinih greda rešetke mehanizma. Dobiveni rezultati strukturalne analize mehanizma koriste se kod proračuna funkcije cilja unutar optimizacijskog procesa. Zbog prirode problema, strukturalna optimizacija svodi se na numeričku optimizaciju, a odabire se pristup određivanja lokalnih minimuma korištenjem *Multi-Scale Grid Search* algoritma na temelju nasumičnog odabira iznosa početnih optimizacijskih varijabli. Zbog velikog broja lokalnih optimuma ovako opisanog problema te velikog broja optimizacijskih varijabli, potrebno je iterativnim putem micati optimizacijske varijable na temelju iznosa funkcije cilja optimuma lokalnih optimizacija. Prototip dizajniranog mehanizma izrađuje se ispisom na 3D pisaču u svrhu analize numerički optimiziranog modela mehanizma na stvarnom primjeru.

1.1.1 Korišteni alati

Parametrizacija cijelog procesa radi se pomoću *Python* programskog jezika. Kreacija geometrije mehanizma, kao i strukturalna analiza mehanizma obrađuje se pomoću *Calculix* [3] programskog paketa za strukturalnu analizu konstrukcija. Numerička optimizacija dizajna koristi *Python* biblioteku *Indago* [4] te njen lokalni optimizacijski algoritam *Multi-Scale Grid Search*. Vizualizacija rezultata, kao i vizualizacija deformacije mehanizma postavlja se korištenjem *Python* biblioteke *Matplotlib*. Kako je proces optimizacije računalno zahtjevan, koriste se računalni resursi superračunala *Bura* [5] Sveučilišta u Rijeci. Za potrebe izrade 3D modela za ispis koristi se računalni

Poglavlje 1. Uvod

program *OpenSCAD* [6] automatiziran putem *Python* biblioteke *OpenPySCAD* [7]. Dodatne preinake 3D modela postavljaju se korištenjem programskog paketa *FreeCAD* [8]. Brzi prototip mehanizma izrađuje se 3D ispisom na pisaču *Flashforge Creator PRO* u prostorijama *Zajednice tehničke kulture* [9] grada Ludbrega dok se priprema za ispis postavlja u programu *FlashPrint* [10].

2 Podatljivi mehanizmi

Mehanizam je složeno sredstvo sastavljeno od pokretnih članova koje samostalno ili uz ljudsko vođenje obavlja kakav rad ili kakvu zadaću. Koristi se kod prenosa pokreta, sile ili energije [1]. Sastoji se od pokretnih elemenata (članova) koji su međusobno povezani na način da pokretanje jednoga uzrokuje gibanje ostalih. Takvom međusobnom interakcijom, mehanizmi pretvaraju jedan oblik kretanja u drugi.

Prema principu očuvanja energije, generalni cilj mehanizma je prijenos energije, a najčešće bez gubitaka energije unutar članova mehanizma. Takve mehanizme zovemo krutim mehanizmima jer se sastoje od krutih dijelova međusobno povezanih pokretnim vezama (zglobovima).

Ako zahtijevamo automatizirano vraćanje mehanizma u početni položaj, tj. pohranjivanje energije unutar samog mehanizma, potrebno je: dodatno postaviti podatljivi (elastični) član koji svojim deformiranjem pohranjuje energiju u obliku energije deformacije, modificirati jedan član mehanizma ili jedan od već postojećih članova zamijeniti za podatljivi član.

Za razliku od krutih mehanizama podatljivi mehanizmi svoja kinematička svojstva dobivaju velikim elastičnim deformacijama podatljivih dijelova. Energija je pohranjena u obliku unutarne energije deformiranja omogućavajući kasnije preusmjerenje ili korištenje. Zbog mogućnosti korištenja podatljivih svojstva materijala od kojeg je mehanizam izrađen, gubi se potreba povezivanja dva člana mehanizma pomičnom vezom (zglobom). Sama veza je podatljiva, omogućavajući međusobno kretanje dvaju susjednih članova. Ovakav pristup dizajnu mehanizama ima svoje dobre i loše osobine objašnjene u nastavku. Mora se pritom uzeti u obzir kako postoje kruti mehanizmi koje ne možemo lako zamijeniti podatljivim mehanizmima.

2.1 Prednosti podatljivih mehanizama

Podatljivi mehanizmi, za razliku od krutih mehanizama, mogu se izraditi od jedne cjeline homogenog materijala. Smanjenu cijenu krajnjeg proizvoda osigurava manje članova mehanizma (manje djelova od kojih je mehanizam sastavljen) te pojednostavljena proizvodnja mehanizma. Podatljiv mehanizam primjerice može biti izrađen od plastike ljevanjem plastike ili 3D printanjem cijelog mehanizma. Također postoji mogućnost izrade mikroskopskih mehanizama iz metala raznim tehnikama: *Wire electrical discharge machining*, mikroskopsko lijevanje, mikro 3D printanje itd. Više o postupcima izrade, ograničenjima i primjeni mikroskopskih podatljivih mehanizama može se pronaći u radu [2].

Pojednostavljeno je održavanje mehanizma kojem više ne treba podmazivanje zglobova za potrebe smanjenja tarnog trošenja. Također, manjak zglobova omogućuje rad mehanizma u uvjetima koji loše utječu na zglobove, kao što su rad u prašnjavom okruženju ili rad u vlažnom okruženju [1]. Zglobovi podatljivih mehanizama imaju dvojnju funkciju fleksijske opruge i krutog zgloba, te temeljem geometrijskih karakteristika zgloba (spoja članaka) možemo mijenjati karakteristike takve opruge, usput mijenjajući karakteristike kinematike i dinamike mehanizma.

Velika prednost podatljivih mehanizama je mogućnost postizanja visoke preciznosti pomaka pojedinih dijelova mehanizma. Ova karakteristika uz sposobnost proizvodnje mikrometarskih podatljivih mehanizama dopušta izradu visokopreciznih mikroskopskih mehanizama izrađenih od jednog homogenog neprekinutog materijala.

2.2 Mane podatljivih mehanizama

Kako kretanje mehanizma više ne možemo analizirati isključivo kinematski kao spoj krutih greda zglobovima, već je potrebno proračunati velike deformacije savijanja njegovih članova te međusobnu interakciju članova, dinamički proračun mehanizma je poprilično otežan. Sam dizajn podatljivih mehanizama postaje kombinacija više različitih grana analize konstrukcija, gdje uz same metoda sinteze moramo pratiti nelinearnu analizu čvrstoće konstrukcije, zbog geometrijske nelinearnosti i proračuna

Poglavlje 2. Podatljivi mehanizmi

velikih deformacija.

Problem se javlja i kod samog prenošenja ulazne energije u izlaznu, gdje se dio energije svakako gubi na unutarnju energiju deformiranja mehanizma. Za razliku od krutih mehanizama, gdje se manji postotak energije gubi na trenje unutar zglobova, mnogo veći postotak energije potrošen je na deformaciju samog podatljivog mehanizma. Svakako se mora uzeti u obzir da ovo nije mana svakog podatljivog mehanizma, jer kao posljedicu elastičnosti mehanizma često želimo da se energija predana mehanizmu pohrani u obliku unutarnje energije deformacije te se kasnije koristi. Na taj način, primjerice kod najjednostavnijeg podatljivog mehanizma luka (luk i strijela) energija korištena za savijanje luka pohranjuje se u obliku unutarnje energije deformiranja, te kod otpuštanja strune sva ta unutarnja energija deformiranja drva prenosi se na povećanje kinetičke energije strijele.

Ako zahtijevamo ciklično opterećenje mehanizma, dolazimo do zahtjeva za dodatnom analizom zamora materijala podatljivog mehanizma. Zbog toga, ako planiramo proizvesti mehanizam koje će se ciklički opterećivati, moramo uzeti u obzir i proračun vijeka trajanja mehanizma

Problem se javlja i kod podatljivih mehanizama koji su kontinuirano opterećeni. Kontinuirano opterećenje ili rad na povišenim temperaturama može prouzročiti tečenje ili relaksaciju materijala od kojeg je mehanizam izrađen. Kao na primjer lopatice koje osiguravaju kontakt u elektromotoru (unutarnja energija deformiranja lopatice osigurava kontinuiranu silu i kontakt s statorom motora), zbog kontinuiranog opterećenja lopatica postoji mogućnost slabljenja sile pritiska lopatica, onemogućujući normalno funkcioniranje motora.

Podatljivi mehanizmi nisu zamjena za krute mehanizme, jer ne mogu izvršiti sva gibanja omogućena od strane zglobova u krutim mehanizmima [1]. Primjerice ne mogu proizvesti kontinuiranu rotaciju oko zgloba što nam recimo omogućuje kuglični zglob ili ležaj.

Također iako su dizajnirani kako bi bili trajniji od krutih mehanizama, u slučaju havarije na mehanizmu, nije moguće promijeniti par dijelova mehanizma, kako se podatljivi mehanizam sastoji od jednog homogenog materijala.

2.3 Primjena podatljivih mehanizama

Najstariji dokazi korištenja podatljivih mehanizama luka i strijele datirani su na 60 000 godina starosti [11]. Oružje *luk i strijela* radi na mehanizmu otpuštanja pohranjene energije u obliku energije deformacije luka izazivajući pritom naglo zatezanje pričvršćenog konopa koji energiju deformacije (uz naravno neke gubitke) pretvara u kinetičku energiju strijele. Ovakvi i slični povijesni primjeri podatljivih mehanizama koriste drvo ili metalnu lamelu kao dio mehanizma koji pohranjuje predanu energiju u unutarnju energiju deformiranja. Primjerice zamke, katapult, samostreli, suspenzije kočija itd. koriste neku vrstu podatljivog mehanizma ili podatljivog člana.

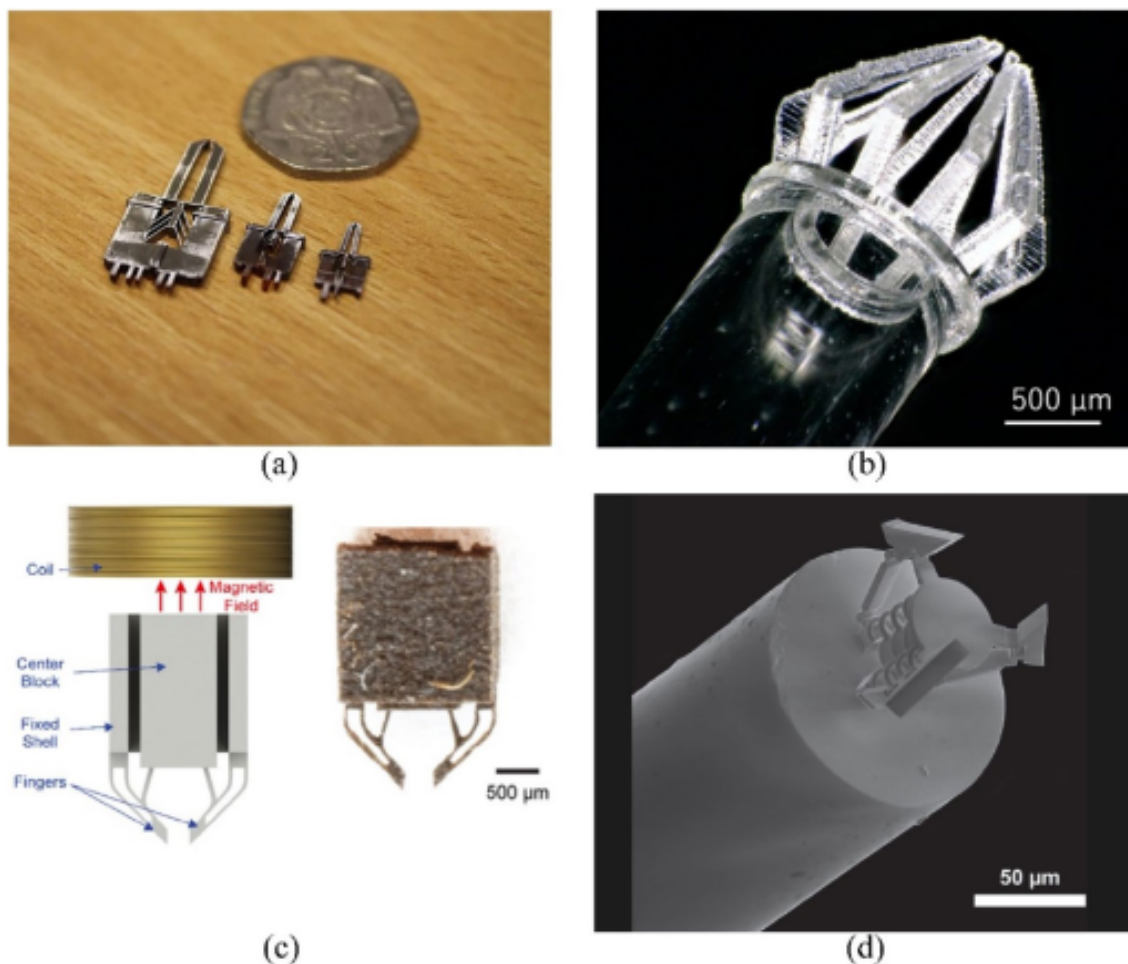
Danas podatljive mehanizme možemo pronaći na svakom uglu, od podatljivih plastičnih kopči, bistabilnih podatljivih prekidača, plastičnih podatljivih ribičkih klješta (prikazanih na slici 2.1), itd.



Slika 2.1 Primjer podatljivih plastičnih klješta za odstranjivanje udica izrađenih lijevanjem plastike. [1]

Također se podatljivi mehanizmi koriste kod mikroskopskih sklopova, kako se mikroskopske opruge, grede i međusobne veze vrlo teško izrađuju/postižu u obliku krutog mehanizma. Jednostavnije rješenje je kreiranje i postavljanje mikroskopskog podatljivog mehanizma koji može postizati tražene kinematičke uvjete te pritom imati mogućnost pohrane predane energije u obliku unutarnje energije deformacije podatljivog mehanizma.

Poglavlje 2. Podatljivi mehanizmi



Slika 2.2 Primjer mikroskopskih metalnih podatljivih mehanizama hvataljke koja aksijalni pomak izazvan djelovanjem elektromagneta na dio mehanizma pretvara u zatvaranje “krakova” mehanizma [2].

Vidljivo je kako ljudska potreba i intuicija vezana uz podatljive mehanizme seže u daleku prošlost, te se kasnije praćena novopronađenim materijalima usavršava i otvara mogućnosti kreiranja kompleksnih podatljivih mehanizama.

3 Modeliranje i simuliranje podatljivih mehanizama

Kako se ovaj rad bavi dizajnom mehanizama, umjesto odabira jednog određenog mehanizma uzima se skup parametara koje mehanizam mora zadovoljavati (geometrijska ograničenja, opterećenja, rubni uvjeti, tid.). Generalizacijom postupka optimizacije otvara se mogućnost primjene metodologije na dizajn različitih mehanizama.

Mehanizam će se sastojati od međusobno kruto spojenih grednih elemenata što nam omogućuje praćenje savijanja pojedinih elemenata te cijelog mehanizma. Pomaci i naprezanja nakon elastične deformacije podatljivog mehanizma, zbog jednostavnosti proračuna, računat će se korištenjem metode konačnih elemenata. Postoji mogućnost provođenja računalno jeftinijeg proračuna metodom “Pseudo rigidnog mehanizma” gdje spojeve krutih članaka mehanizma (zglobove mehanizma) tretiramo kao fleksijske opruge. Ova metoda je pobliže opisana u [1].

Za sam proces analize mehanizma koristi se nelinearni proračun metodom konačnih elemenata korištenjem grednih konačnih elemenata pomoću računalnog programa *Calculix* [3]. Proračun mora biti nelinearan zbog geometrijske nelinearnosti izazvane velikim deformacijama te velikim pomacima i rotacijama podatljivog mehanizma.

3.1 Metoda konačnih elemenata

Metoda konačnih elemenata (MKE) je numerička metoda za rješavanje rubnog problema koja se temelji na diskretizaciji kontinuuma. Uporabom MKE rješavanje problema elastičnosti konstrukcija svodi se na rješavanje sustava diferencijalnih jednadžbi. Prva primjena MKE najčešće se veže za matematičara R. Couranta [12], koji je u časopisu *Bulletin of the American Mathematical Society* (1943) dao rješenje torzijskog problema primjenom polinomne aproksimacije i diskretizirane domene. Pedesetih godina prošlog stoljeća dolazi do praktične primjene Courta-

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama

nalove metode u avionskoj industriji. Nakon početne primjene MKE-a u analizi konstrukcija, upotreba se širi na termodinamiku, dinamiku fluida itd. Tijekom sedamdesetih godina počinju se razvijati komercijalni računalni programi temeljeni na MKE, a krajem osamdesetih godina programi imaju mogućnost *post-processinga* i *pre-processinga* kakav nam je poznat danas u većini komercijalnih programa.

MKE se temelji na diskretizaciji kontinuuma, odnosno konstrukcije, na odgovarajući broj podkontinuumata ili konačnih elemenata. Elementi su međusobno povezani u točkama koje se nazivaju čvorovi. Za svaki element pretpostavlja se rješenje zadane diferencijalne jednačbe u obliku interpolacijskih funkcija koje povezuju zavisne varijable s njihovim vrijednostima u čvorovima. Prvo se konačni elementi povezuju u mrežu konačnih elemenata povezanih čvorovima, a zatim se pomoću odgovarajućih postupaka formira globalni sustav jednačbi za cijeli diskretizirani model. Konačni elementi dijele se na jednodimenzijske, dvodimenzijske i trodimenzijske konačne elemente, te elemente za rješavanje problema posebnih geometrijskih oblika, npr. pločasti i ljuskasti konačni elementi [13].

Jednačbe konačnih elemenata izvode se pomoću diferencijalne i varijacijske formulacije, odnosno moguće je primijeniti Galerkinovu i Rayleigh-Ritzovu metodu. Navedeni izvodi temelje se na [13].

3.1.1 Matrica polja i matrica interpolacijskih funkcija

Svaki konačni element ima čvorove i polje. Na primjer, pomak u u polju elementa mijenja se po određenom zakonu u funkciji koordinata, a analogno tome mijenjaju se i ostali pomaci u polju elementa, te je vektor pomaka u polju konačnog elementa:

$$\mathbf{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{Bmatrix} \quad (3.1)$$

Pomaci se u polju konačnih elemenata najčešće aproksimiraju punim i reduciranim algebarskim polinomima, dok se kod dvodimenzijskih i trodimenzijskih problema za odabir polinoma rabe Pascalov trokut i Pascalov tetraedar.

3.1.2 Jednodimenzijski linearni elementi

Kako se u ovom radu koriste gredni elementi, potrebno je prvo definirati jednodimenzijski linearni element. Jednodimenzijski konačni elementi najjednostavniji su konačni elementi te se primjenjuju za jednodimenzijsku analizu kontinuuma pri čemu su sve varijable funkcije samo jedne koordinate [14].

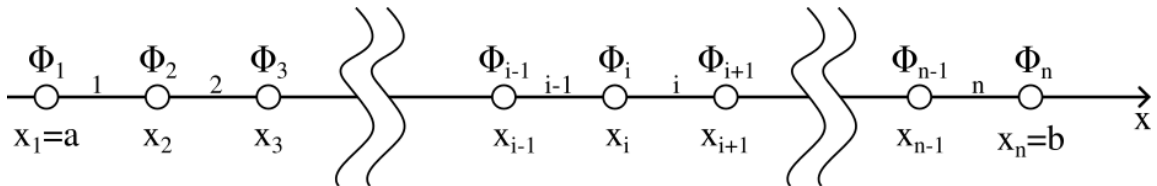
Jednadžba konačnog elementa

Razmatra se problem opisan jednodimenzijskom Poissonovom diferencijalnom jednadžbom (3.2) u intervalu $[a, b]$ s Dirichletovim i Neumanovim rubnim uvjetima (3.3) [13].

$$-\alpha = \frac{d^2\phi}{dx^2} = f(x) \quad (3.2)$$

$$\phi(a) = \phi_a, \quad \alpha \frac{d\phi}{dx} \Big|_b = h_b \quad (3.3)$$

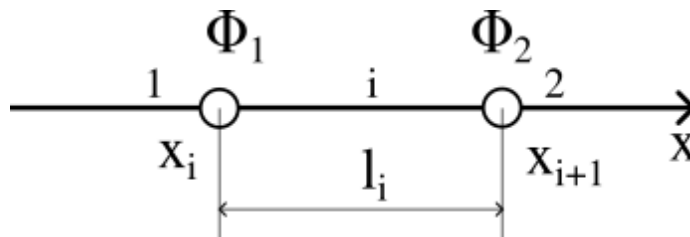
Jednadžba (3.2) prikazuje kako je druga derivacija pomaka u smjeru pravca na kojem konačni element leži jednaka koeficijentu $\alpha = \frac{F}{AE}$ gdje su F zadano aksijalno opterećenje, A površina poprečnog presjeka štapa i E modul elastičnosti štapnog elementa. Pri čemu se pretpostavlja da je $\alpha = konst.$ Pomoću jednadžbe (3.2) moguće je opisati problem osnog opterećenja štapa te jednodimenzijski problem toplinskog provođenja. Zadani interval $[a, b]$ podijelit će se na manje podintervale $[x_i, x_{i+1}]$, gdje je $i = 1, \dots, n - 1$.



Slika 3.1 Način podjele jednodimenzijskog područja na podintervale.

Svaki podinterval prikazan na slici (3.1.2) prikazuje konačni element s nepoznatim vrijednostima funkcije ϕ .

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama



Slika 3.2 Prikaz jednodimenzijskog elementa.

Kako je formulacija svih elemenata jednaka, prikazat će se samo formulacija samo jednog i – tog elementa na slici 3.2 Zbog jednostavnosti, čvorovi će biti označeni brojevima 1 i 2, a čvorne varijable ϕ_1 i ϕ_2 .

Duž elementa pretpostavlja se linearna raspodjela funkcije ϕ

$$\phi = a_1 + a_2x. \quad (3.4)$$

Da bi se funkcija ϕ povezala s njezinim vrijednostima u čvorovima ϕ_1 i ϕ_2 polazi se od rubnih uvjeta:

$$\begin{aligned} x = x_i &\longrightarrow \phi = \phi_1, \\ x = x_{i+1} &\longrightarrow \phi = \phi_2 \end{aligned} \quad (3.5)$$

na temelju kojih se dobiva:

$$\begin{aligned} \phi_1 &= a_1 + a_2x_i, \\ \phi_2 &= a_1 + a_2x_{i+1} \end{aligned} \quad (3.6)$$

Iz jednadžbe 3.6 dobivaju se izrazi za a_1 i a_2 :

$$\begin{aligned} a_1 &= \frac{\phi_1x_{i+1} - \phi_2x_i}{x_{i+1} - x_i}, \\ a_2 &= \frac{\phi_2 - \phi_1}{x_{i+1} - x_i} \end{aligned} \quad (3.7)$$

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama

Nakon uvrštavanja jednadžbe 3.7 u jednadžbu 3.4

$$\phi = \frac{x_{i+1} - x}{x_{i+1} - x_i} \phi_1 + \frac{x - x_i}{x_{i+1} - x_i} \phi_2 \quad (3.8)$$

gdje su :

$$\begin{aligned} N_1(x) &= \frac{x_{i+1} - x}{x_{i+1} - x_i}, \\ N_2(x) &= \frac{x - x_i}{x_{i+1} - x_i} \end{aligned} \quad (3.9)$$

Lako se zaključi iz jednadžbe (3.9) da za $x = x_i$ vrijedi $N_1 = 1$, $N_2 = 0$, a za $x = x_{i+1}$ vrijedi $N_1 = 0$, a $N_2 = 1$. Što pokazuje svojstvo funkcije oblika: funkcija je jednaka jedinici u čvoru na koji se odnosi, a u svim ostalim čvorovima jednaka je nuli. Izraz jednadžbe (3.8) može se napisati u obliku:

$$\phi(x) = N_1(x)\phi_1 + N_2(x)\phi_2 \quad (3.10)$$

Sljedeći korak je pomoću Galerkinove metode težinskih reziduala izvesti jednadžbu konačnog elementa. Za područje konačnog elementa pretpostavit će se rješenje (3.2) u obliku funkcije (3.10). U skladu s Galerkinovom metodom težinske su funkcije $W_1 = N_1(x)$ i $W_2 = N_2(x)$, a težinska prosječna vrijednost ostatka za područje konačnog elementa jednaka je:

$$\int_{x_i}^{x_{i+1}} N_j \left(\alpha \frac{d^2 \phi}{dx^2} + f \right) dx = 0 \quad j = 1, 2 \quad (3.11)$$

Primjenom parcijalne integracije

$$\int_{x_i}^{x_{i+1}} N_j \frac{d^2 \phi}{dx^2} dx = N_j \frac{d\phi}{dx} \Big|_{x_i}^{x_{i+1}} - \int_{x_i}^{x_{i+1}} \frac{dN_j}{dx} \frac{d\phi}{dx} dx \quad (3.12)$$

te se (3.11) može transformirati u oblik:

$$\alpha \int_{x_i}^{x_{i+1}} \frac{dN_j}{dx} \frac{d\phi}{dx} dx = \int_{x_i}^{x_{i+1}} N_j f dx + N_j \alpha \frac{d\phi}{dx} \Big|_{x_i}^{x_{i+1}} \quad (3.13)$$

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama

Izraz (3.10) zajedno s njegovim integralnim oblicima moguće je prikazati u indeksnom zapisu

$$\phi = N_k \phi_k, \quad \frac{d\phi}{dx} = \frac{dN_k}{dx} \phi_k, \quad k = 1, 2 \quad (3.14)$$

Uvrštavanjem (3.14) relacija (3.13) postaje:

$$\left(\alpha \int_{x_i}^{x_{i+1}} \frac{dN_j}{dx} \frac{dN_k}{dx} \right) \phi_k dx = \int_{x_i}^{x_{i+1}} N_j f dx + N_j \alpha \frac{d\phi}{dx} \Big|_{x_i}^{x_{i+1}} \quad j = k = 1, 2 \quad (3.15)$$

Ovdje se provodi zbrajanje po ponovljenom indeksu k . Drugi član s desne strane jednadžbe (3.15) označuje Neumannove rubne uvjete. Sada se jednadžba (3.15) može prikazati u matričnom obliku:

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} F_{S1} \\ F_{S2} \end{bmatrix} + \begin{bmatrix} -\alpha \frac{d\phi}{dx}(x_i) \\ -\alpha \frac{d\phi}{dx}(x_{i+1}) \end{bmatrix} \quad (3.16)$$

Izraz (3.16) je jednadžba konačnog elementa koja je u simboličkom matričnom zapisu za element i s koordinatama čvorova x_i i x_{i+1} jednaka:

$$\mathbf{k}^i \boldsymbol{\psi}^i = \mathbf{F}_S^i + \begin{bmatrix} -\alpha \frac{d\phi}{dx}(x_i) \\ -\alpha \frac{d\phi}{dx}(x_{i+1}) \end{bmatrix} \quad (3.17)$$

gdje je

- $\mathbf{k}^i = [k_{jk}]$ matrica krutosti elementa,
- $\boldsymbol{\psi}^i = [\phi_k]$ vektor nepoznatih veličina u čvorovima,
- $\mathbf{F}_S^i = [F_{Sj}]$ vektor opterećenja

Globalna jednadžba konačnih elemenata

Pri formulaciji globalne jednadžbe konačnih elemenata prvo će se izvesti globalna matrica krutosti. Pritom je potrebno obratiti pažnju na globalne i lokalne čvorne varijable. Čvorne varijable vezane za element nazivaju se lokalne čvorne varijable,

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama

dok se varijable vezane za proračunski model nazivaju globalne čvorne varijable. Pri formiranju globalne matrice krutosti počinje se s nul-matricom čiji je red jednak redu globalne matrice krutosti. Nakon toga počinje proces popunjavanja s elementima matrica krutosti pojedinih konačnih elemenata. Postupak za opći slučaj proizvoljnih konačnih elemenata:

Ako globalnoj čvornoj varijabli ϕ_1 odgovaraju lokalne čvorne varijable npr. elemenata $1, 2, \dots, \phi_j^1, \phi_k^2, \dots$, a čvornoj varijabli ϕ_i odgovaraju druge čvorne varijable istih elemenata $\phi_m^1, \phi_n^2, \dots$, elementi globalne matrice krutosti izračunavaju se superpozicijom na ovaj način:

$$K_{il} = k_{jm}^1 + k_{kn}^2 + \dots \quad (3.18)$$

K_{il} je element globalne matrice krutosti koji je u i -tom retku i l -tom stupcu, k_{jm}^1 je element matrice krutosti elementa 1 analogno vrijedi za k_{kn}^2 elementa 2. Za proračunski model prikazan na slici podudaranje lokalnih i globalnih čvornih varijabli prikazano je u tablici.

Tablica 3.1 Lokalne i globalne čvorne varijable

globalne čvorne varijable	ϕ_1	ϕ_2	ϕ_3	ϕ_{i-1}	ϕ_i	ϕ_{i+1}	ϕ_{n-1}	ϕ_n
1	1	2						
2		1	2					
elementi	$i-1$			1	2			
i čvorne	i				1	2		
varijable	n						1	2

Zbog jednostavnosti varijable su označene s 1 i 2, tako je npr. čvorna varijabla elementa 1, ϕ_1^1 označena brojem 1, a ϕ_2^1 označena brojem 2. Sada se na temelju tablice može popunjavati matrica krutosti gdje su elementi globalne matrice krutosti jednaki:

$$K_{11} = k_{11}^1, \quad K_{12} = k_{12}^1, \quad K_{21} = k_{21}^1, \quad K_{22} = k_{22}^1 + k_{11}^1 \quad (3.19)$$

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama

$$\mathbf{K} = \begin{bmatrix} k_{11}^1 & k_{12}^1 & & & & \\ k_{21}^1 & k_{22}^1 + k_{11}^1 & k_{12}^1 & & & \\ & k_{21}^2 & k_{22}^2 + k_{11}^3 & & 0 & \\ & & & \ddots & & \\ & & 0 & & \ddots & \\ & & & & & \ddots & \\ & & & & & & k_{22}^{n-1} + k_{11}^n & k_{12}^n \\ & & & & & & k_{21}^n & k_{22}^n \end{bmatrix} \quad (3.20)$$

pri čemu je globalni vektor čvornih varijabli

$$\Phi^T = [\phi_1 \ \phi_2 \ \phi_3 \ \cdot \ \cdot \ \cdot \ \cdot \ \phi_{n-1} \ \phi_n] \quad (3.21)$$

Globalni vektor opterećenja određuje se također na temelju podudaranja lokalnih i globalnih čvornih varijabli. Globalni vektor \mathbf{R}_S jednak je

$$\mathbf{R}_S = \begin{bmatrix} F_{S1}^1 \\ F_{S2}^1 + F_{S1}^2 \\ F_{S2}^2 + F_{S1}^3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ F_{S2}^{n-1} + F_{S1}^n \\ F_{S2}^n \end{bmatrix} \quad (3.22)$$

Sada se globalni sustav jednadžbi može kraće zapisati kao

$$\mathbf{K} \Phi = \mathbf{R}_S \quad (3.23)$$

Da bi se ovaj sustav jednadžbi mogao riješiti potrebno je uzeti u obzir rubne uvjete.

3.1.3 Gredni konačni element

Kod opisanog primjera konačnog elementa, štapni konačni element može prenositi samo aksijalnu silu. Potrebno je definirati dodatne stupnjeve slobode te moguća opterećenja grednog konačnog elementa kako si se izveo sustav jednažbi potreban za rješavanje strukturalnog problema.

Stupnjevi slobode grednog konačnog elementa su poprečni pomaci čvorova, te kutevi zakreta.

$$\phi^T = \{w_1, \varphi_{1,z}, w_2, \varphi_{2,z}\} \quad (3.24)$$

Raspodjela poprečnog pomaka opisana je jednodimenzijским polinomom trećeg stupnja [14]:

$$w = a_1 + a_2x + a_3x^2 + a_4x^3 \quad (3.25)$$

Što je u matičnom obliku:

$$\mathbf{u} = \alpha \mathbf{a} \quad (3.26)$$

gdje je:

$$\mathbf{u} = \{w\} \quad (3.27)$$

$$\alpha = \{1 \ x \ x^2 \ x^3\} \quad (3.28)$$

$$\mathbf{a}^T = \{a_1 \ a_2 \ a_3 \ a_4\} \quad (3.29)$$

$$(3.30)$$

Kut zakreta definiran je izrazom: $\varphi_z = -\frac{dw}{dx}$ Potrebno je zadovoljiti rubne uvjete u čvorovima na početku i kraju elementa:

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama

$$\begin{aligned} \text{za } x = 0 \text{ (čvor 1) , } \quad w = w_1, \quad \varphi = -\frac{dw}{dx} = \varphi_{1z} \\ \text{za } x = l \text{ (čvor 2) , } \quad w = w_2, \quad \varphi = -\frac{dw}{dx} = \varphi_{2z} \end{aligned}$$

Uvrštavanjem jednadžbi raspodjele poprečnog pomaka u rubne uvjete, dobivamo sljedeći oblik vektora pomaka:

$$\mathbf{v} = \mathbf{C}\mathbf{a} \quad (3.31)$$

gdje je matrica \mathbf{C} definirana na sljedeći način:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & l & l^2 & l^3 \\ 0 & -1 & -2l & -3l^2 \end{bmatrix} \quad (3.32)$$

Matrica funkcija oblika \mathbf{N} potrebna za definiranje matrice krutosti dobiva se preko sljedeće relacije [14]:

$$\mathbf{u} = \alpha\mathbf{a} \quad (3.33)$$

$$\mathbf{a} = \mathbf{C}^{-1}\mathbf{v} \quad (3.34)$$

$$\mathbf{u} = \alpha\mathbf{C}^{-1}\mathbf{v} = \mathbf{N}\mathbf{v} \rightarrow \mathbf{N} = \alpha\mathbf{C}^{-1} \quad (3.35)$$

Dobiva se sljedeća matrica funkcije oblika:

$$\mathbf{N}^T = \begin{bmatrix} N_1 = 1 - \frac{3x^2}{l^2} + \frac{2x^3}{l^3} \\ N_2 = -x + \frac{2x^2}{l} - \frac{x^3}{l^2} \\ N_3 = \frac{3x^2}{l^2} - \frac{2x^3}{l^3} \\ N_4 = \frac{x^2}{l} - \frac{x^3}{l^2} \end{bmatrix} \quad (3.36)$$

Tablica 3.2 Iznosi funkcija oblika i njihovih derivacija

Promatrani čvor	Funkcija oblika	Iznos	Derivacija funkcije oblika	Iznos
Čvor 1	N_1	1	dN_1/dx	0
	N_2	0	dN_2/dx	1
	N_3	1	dN_3/dx	0
	N_4	0	dN_4/dx	1
Čvor 2	N_1	0	dN_1/dx	0
	N_2	0	dN_2/dx	0
	N_3	0	dN_3/dx	0
	N_4	0	dN_4/dx	0

Funkcije oblika N_1 i N_3 pridružene su pomacima w_1 i w_2 , a N_2 i N_4 kutovima zakreta φ_1z i φ_2z . Uvrštavanjem $x_0 = 0$ i $x_1 = l$ dobivaju se vrijednosti funkcija oblika:

Iznosi prikazani u tablici 3.2 pokazuju kako su prethodno definirani rubni uvjeti zadovoljeni.

Kako bi dobili matricu krutosti, potrebno je odrediti matricu \mathbf{B} koja se dobiva relacijom $\mathbf{B} = \mathbf{D}_k \mathbf{N}$. Diferencijalni operator \mathbf{D}_k za rješavanje problema savijanja grede jednak je:

$$\mathbf{D}_k = \left[-\frac{d^2}{dx^2} \right] \quad (3.37)$$

te je tada matrica \mathbf{B} jednaka:

$$\mathbf{B} = \left[-\frac{d^2}{dx^2} \right] \left\{ N_1 \quad N_2 \quad N_3 \quad N_4 \right\} \quad (3.38)$$

Deriviranjem funkcija oblika dobiva se:

$$\mathbf{B}^T = \left\{ \begin{array}{l} \frac{6}{l^2} - \frac{12x}{l^3} \\ -\frac{4}{l} + \frac{6x}{l^2} \\ -\frac{6}{l^2} + \frac{12x}{l^3} \\ -\frac{2}{l} + \frac{6x}{l^2} \end{array} \right\} \quad (3.39)$$

Raspodjelu momenata savijanja duž konačnog elementa moguće je odrediti:

$$\sigma = \mathbf{D}\epsilon \quad (3.40)$$

$$\epsilon = \mathbf{B}\mathbf{v} \quad (3.41)$$

$$\sigma = \mathbf{D}\mathbf{B}\mathbf{v} = \mathbf{S}\mathbf{v} \rightarrow \mathbf{S} = \mathbf{D}\mathbf{B} \quad (3.42)$$

Raspodjela naprezanja tj. momenata savijanja određena je matricom \mathbf{S} , budući da je $\mathbf{D} = [EI_z]$, matrica \mathbf{S} jednaka je:

$$\mathbf{S}^T = [EI_z] \begin{Bmatrix} \frac{6}{l^2} - \frac{12x}{l^3} \\ -\frac{4}{l} + \frac{6x}{l^2} \\ -\frac{6}{l^2} + \frac{12x}{l^3} \\ -\frac{2}{l} + \frac{6x}{l^2} \end{Bmatrix} \quad (3.43)$$

Kako je matrica krutosti $k = \int_0^l \mathbf{B}^T \mathbf{D} \mathbf{B} dx$ nakon množenja i integriranja dobivamo:

$$\mathbf{k} = EI_z \begin{bmatrix} \frac{12}{l^3} & -\frac{6}{l^2} & -\frac{12}{l^3} & -\frac{6}{l^2} \\ -\frac{6}{l^2} & \frac{4}{l} & \frac{6}{l^2} & \frac{2}{l} \\ -\frac{12}{l^3} & \frac{6}{l^2} & \frac{12}{l^3} & \frac{6}{l^2} \\ -\frac{6}{l^2} & \frac{2}{l} & \frac{6}{l^2} & \frac{4}{l} \end{bmatrix} \quad (3.44)$$

Sile i momenti u čvorovima pridruženi stupnjevima slobode elementa članovi su vektora koncentriranih sila \mathbf{F} :

$$\mathbf{F}^T = \left\{ F_{1y} \quad M_{1z} \quad F_{2y} \quad M_{2z} \right\} \quad (3.45)$$

3.2 Postavljanje modela

U ovom poglavlju opisuje se parametrizacija problema gredne rešetke koji se optimizira. Postavljanjem modela aproksimira se stvarno fizikalno djelovanje mehanizma pomoću matematičkih i fizikalnih aproksimacija [15]. Modeli svakako zanemaruju određene karakteristike stvarnih sustava, ali su matematički dovoljno

robusni kako bi pomoću njih postavljali predviđanja ponašanja stvarnog sustava. Kompletan kod korišten kod izrade ovog rada može se naći na *GitHub* repozitoriju [16].

3.2.1 Bazna rešetka, oslonci i opterećenja

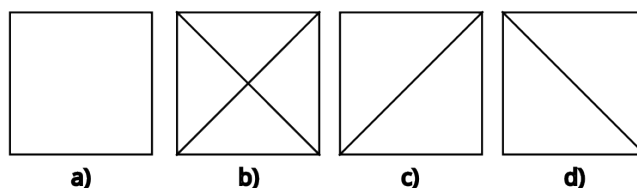
Mehanizme opisane ovim radom moguće je izraditi 3D printanjem, tako da je veličina mehanizma u ovom radu ograničena mogućom veličinom printanja 3D printera. Zbog jednostavnije analize mehanizma i izrade, zadaje se ograničenje kretanja dizajniranog mehanizma isključivo unutar jedne ravnine. Ovo ograničenje postiže se korištenjem grednih elemenata s pravokutnim poprečnim presjekom čija visina (dimenzija okomita na ravninu rada mehanizma) ne prelazi veličinu debljine pojedine grede. Kod tako postavljenih elemenata, svim grednim elementima konstrukcije centralna ravnina inercije postaje jednaka te je možemo nazvati centralnom ravninom inercije mehanizma.

Pokušaj generalizacije ovakvog mehanizma je ograničenje početnog oblika na pravokutnu rešetku grednih elemenata s proizvoljnim brojem horizontalnih i vertikalnih podjela. Jedan od doprinosa ovog rada je parametrizacija geometrijskih karakteristika i automatizacija izrade geometrijskog modela odabirom nekog od tipičnih oblika rešetkaste strukture. Dodaje se mogućnost postavljanja dijagonalnih potpornih greda unutar polja rešetke prikazane na slici 3.3 (praznog pravokutnog prostora omeđenog gredama) koja dijagonalno spaja međusobno suprotne čvorove polja rešetke ili križnom potporom sastavljenom od četiri grede koje spajaju sve čvorove polja rešetke s njegovim središtem.

Također integrirana je mogućnost micanja većeg broja greda unutar zadanog mnogokuta ako je poznato da određeni dio gredne rešetke sigurno neće biti potreban u proračunu.

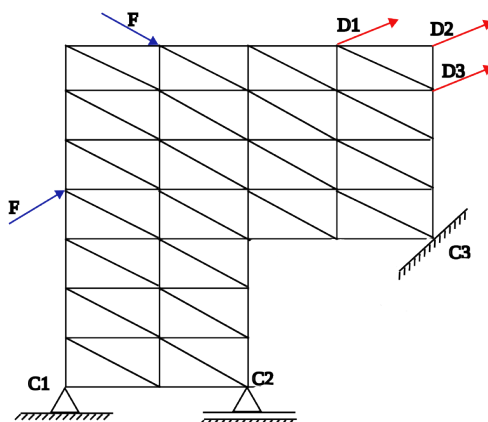
Na slici 3.4 prikazuje se primjer mogućeg modela. Ovaj model primjer je rešetke s 4 podjele u horizontalnom smjeru i 7 podjela u vertikalnom smjeru, te se koriste lijevo dijagonalne potpore. Recimo da je poznato kako donji desni dio rešetke nije potreban kod krajnjeg oblika mehanizma, koristi se integrirana mogućnost micanja greda u pojedinom području mehanizma kako bi te grede unutar pravokutne

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama



Slika 3.3 Primjer potpornih greda. Polje **a)** označava polje bez dijagonalnih potpora, omeđeno s četiri grede. Polje s križnom potporom **b)** i polja s dijagonalnim potporama **c)** i **d)**.

rešetkaste konstrukcije odmah u početku optimizacije maknuli iz optimizacijskog procesa. Postavljaju se početni uvjeti dvije sile (lijevi i gornji lijevi dio modela) te oslonac **C1**, pomični oslonac **C2** te uklještenje **C3**. Traži se optimizirano rješenje mehanizma takvo da postoje pomaci **D1**, **D2** i **D3**.



Slika 3.4 Primjer bazne rešetke s prednje naginjućim dijagonalama te s apstraktno definiranim silama **F**, traženim pomacima **D1**, **D2**, **D3**, nepomičnim osloncom **C1**, pomičnim osloncom **C2** i uklještenjem **C3**. S ovako definiranom baznom rešetkom možemo pokrenuti optimizacijski proces koji će promjenom debljina greda postići tražene pomake mehanizma.

Ovako definirani model spreman je za postavljanje mreže konačnih elemenata.

Kako je opisano u poglavlju 3.1 koriste se gredni konačni elementi. Podjela

rešetkaste gredne konstrukcije na gredne konačne elemente radi se na temelju maksimalne veličine konačnog elementa. U slučaju da se neka od greda ne može točno podijeliti na elemente zadane maksimalne veličine, svi elementi te grede se skaliraju kako bi bili relativno nezamjetno manji od maksimalne veličine.

Prihvaćena je ova podjela geometrije na konačne elemente umjesto podjele svih greda na određeni broj elemenata zbog mogućnosti velikih razlika u dužini pojedinih greda. Što bi rezultiralo velikom razlikom između diskretizacija susjednih greda, postizujući različitu točnost rješenja proračuna deformacije pojedinih greda.

3.3 Programski paket *Calculix*

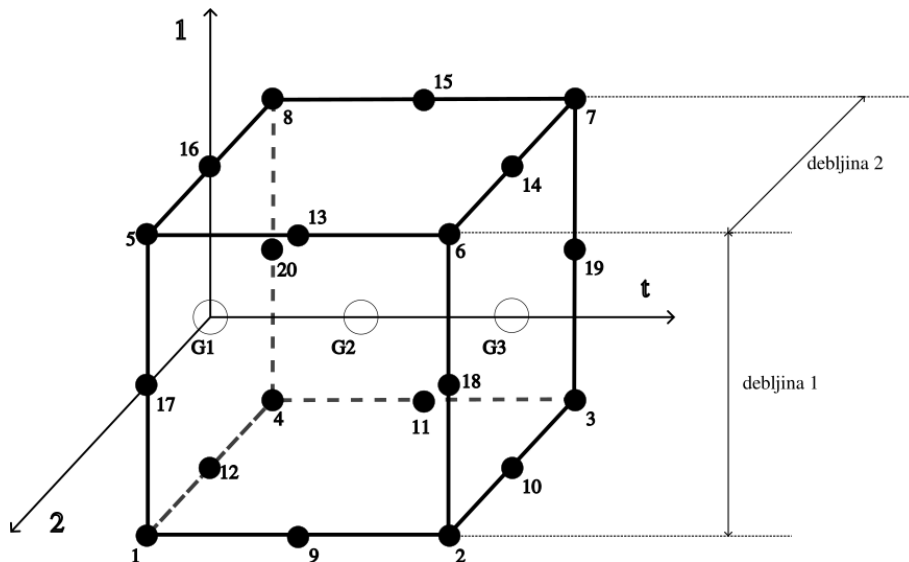
Calculix [3] je program otvorenog koda pod *GNU GPL* licencom. Računalni program koristi metodu konačnih elemenata kako bi riješio problem definiran unutar skupa vektorskih i skalarnih polja. Program nudi mogućnost izrade mreže konačnih elemenata, proračun modela, *pre-processing* i *post-processing* mogućnosti. Paket *Calculix* sastavio je tim zaposlenika *MTU Aero Engines* iz Münchena (Njemačka).

Kako je prethodno spomenuto, za potrebe analize potrebno je odabrati gredne konačne elemente. *Calculix* nudi 4 različita gredna konačna elementa: *B31*, *B31R*, *B32* i *B32R*. Elementi *B31R* ne mogu simulirati savijanje greda, a kako je to pretežiti oblik deformacije u ovom slučaju, ne možemo ih koristiti. Za razliku od njih, *B32* i *B32R* elementi mogu simulirati savijanje greda, te se odabire *B32* element. Za razliku od *B31* elemenata koji se definiraju preko početnog i krajnjeg čvora, svaki *B32* element definira se početnim, krajnjim i središnjim čvorom kako bi se analiziralo izvijanje elementa i omogućilo postavljanje zakrivljenih elemenata. U primjerima zapisanim u ovom radu, koriste se isključivo ravni gredni elementi tj. *B32* elementi čija se središnja točka nalazi na geometrijskom središtu između početne i krajnje točke.

Koristeći parametrizirani model automatizirani proces pretvara jednostavnu definiciju modela u oblik podoban za korištenje kod proračuna *Calculixom* tj. pretvara zadanu mrežu uz njena opterećenja, oslonce, materijalne i geometrijske karakteristike u .inp datoteku. Postavlja se mreža čvorova koji će se koristiti kao

krajnje točke grednih elemenata u proračunu te kao hvatišta sila i pozicije oslonaca. Nakon postavljenih čvorova, postavljaju se grede koje ih povezuju. Grede čiji se vanjski čvorovi (prvi i posljednji) poklapaju, povezuju se krutom vezom. Svaka se greda, nakon postavljanja geometrijskih karakteristika modela dijeli na određeni broj konačnih elemenata (prethodno definiranih veličina) te se zatim svaki element dodatno dijeli na 2 dijela (sveukupno 3 čvora) te se tako zapisuje u ulaznu datoteku *Calculixa*.

B32 elementi se interno, tijekom proračuna, pretvaraju u *C3D20* elemente na temelju slike 3.5. Za svaki čvor *B32* elementa postavlja se 8 novih čvorova. Pozicije novih čvorova te tako i sam oblik *C3D20* elementa definiran je pomoću lokalnih osi elementa, te zadanih debljina elementa u smjeru lokalnih osi. Kada promatramo isključivo 2D primjer, smjer normala 1 svih korištenih elemenata istog je smjera i okomit je na centralnu ravninu inercije mehanizma.



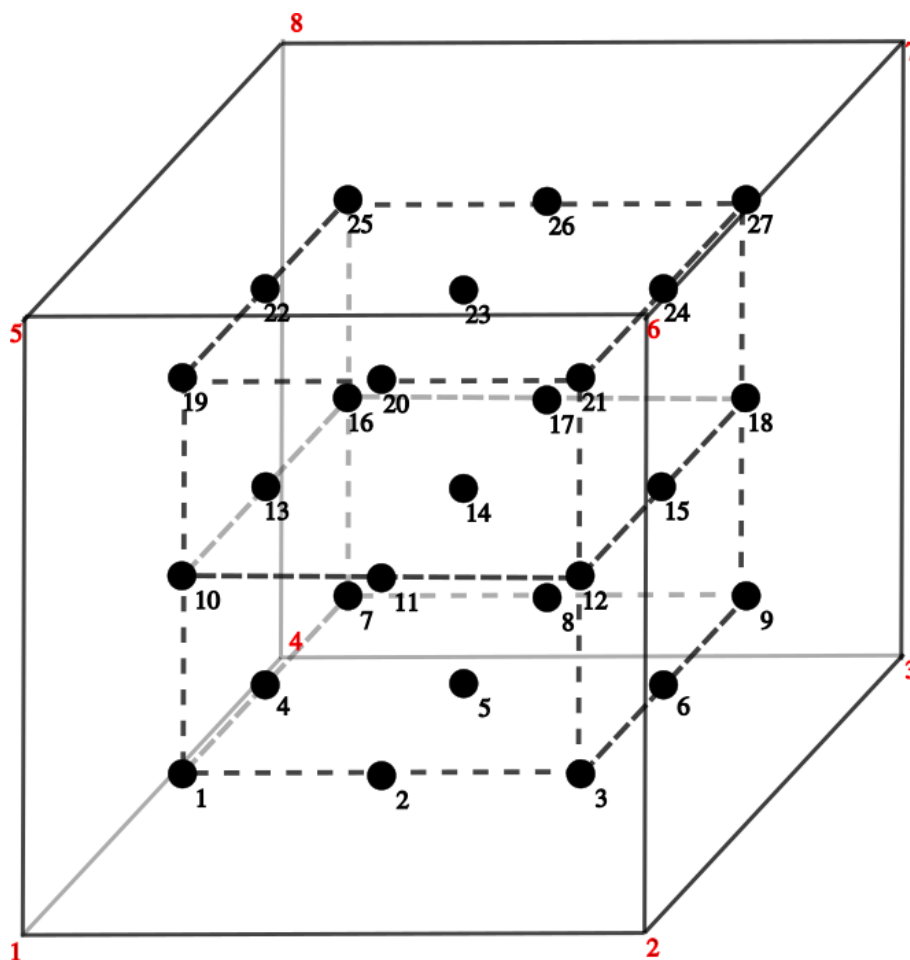
Slika 3.5 Ekspanzija *B32* 1D elementa u 3D element *C3D20*. Točke označene brojevima pripadaju ekspanziranom obliku *C3D20* dok točke označene slovom G i brojkom pripadaju ne ekspanziranom obliku *B32*. [3]

Definira se poprečni presjek pojedinih grednih elemenata gdje visina grede označava visinu poprečnog presjeka grednog elementa u smjeru okomitom na centralnu ravninu inercije mehanizma (os 1 na slici 3.5) dok širina grede označava

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama

širinu poprečnog presjeka grednog elementa.

Calculix vrši proračun korištenjem konačnih elemenata *C3D20* koji se sastoji od 20 čvorova. Ovaj element dobar je za proračun linearno elastičnih slučajeva. Također zbog pozicija integracijskih točaka prikazanih na slici 3.6 element vrlo dobro prikazuje površinske koncentratore naprezanja.



Slika 3.6 3x3x3 integracijska shema točaka kod heksaedarskih elemenata.[3]

Iako je dobar za proračun linearno elastičnih slučajeva, postoje određene mane kod proračuna nelinearnih slučajeva:

- zbog potpune integracije element će se loše ponašati kod proračuna korištenjem izohornih materijala npr. kod visokih vrijednosti Poissonovog koeficijenta ili

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama

plastične deformacije.

- Element je često previsoke krutosti za analizu savijanja tankih greda ili tankih ploča koje se uvijaju.

Uz naravno postavke geometrijskih i materijalnih karakteristika mehanizma (Poissonovog broja, modul elastičnosti), opterećenja i oslonaca, potrebno je zadati nelinearnu strukturalnu analizu konstrukcije te pokrenuti proračun pokretanjem programa *Crunchx*.

Kod korištenja geometrijski nelinearne strukturalne analize koristi se nelinearni tenzor naprezanja (Lagrangeovo naprezanje kod hiperelastičnih materijala, Eulerovo naprezanje kod plastičnih deformacija te devijatorski dio lijevog Cauchy-Greenovog tenzora kod inkrementalne plastičnosti). U proračunu se inkrementalno uvode opterećena, te se Newtonovim iteracijama rješava svaki inkrement. Iako se tijekom proračuna koriste Piola naprezanja drugog reda, naprezanja se pretvaraju u Cauchijeva naprezanja prije ispisa u datoteku rezultata.

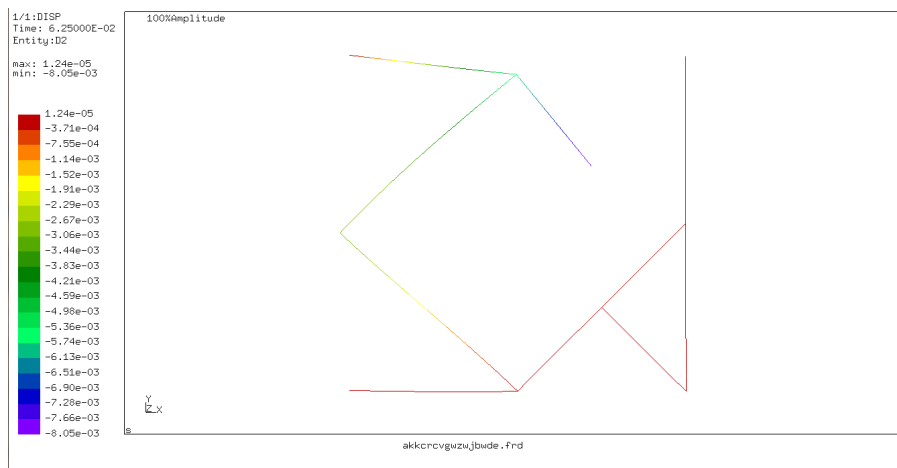
Dio *Calculix* računalnog paketa je rješavač *Crunchx* koji čita ulaznu .inp datoteku te sve potrebne ulazne datoteke vezane uz proračun, numerički riješi problem, te vrati izlazne datoteke koje se koriste kod *post-processinga*. *Crunchx* se može pokretati i paralelno, ali kako su proračuni ovakvih konstrukcija relativno računalno jeftini (ne zahtijevaju mnogo računalnih resursa) nije bilo potrebe za paralelizacijom procesa samog proračuna deformacija podatljivog mehanizma.

Rezultati proračuna nalaze se u .frd datoteci, te se ponovo pomoću automatiziranog postupka prevode u potreban format kompatibilan s ostatkom automatiziranog procesa. Najvažniji podaci proračuna kod analiziranih problema su dobiveni pomaci čvorova (svakog konačnog elementa) te dobivena naprezanja konačnih elemenata. Ovi podaci koriste se kao ulazni parametri funkcija cilja i tijekom ispitivanja postavljenih ograničenja.

Valja napomenuti da *Calculix* sadrži i *post-processor Graphix* kojim možemo tijekom procesa optimizacije jednostavno analizirati dobivene mehanizme. Primjer prikaza mehanizma pomoću *Graphixa* vidljiv je na slici 3.7.

Iako je *Graphix* koristan program kojim relativno brzo možemo analizirati deformaciju zadane konstrukcije, za prikaz konstrukcije koristi se *Python* te biblioteke

Poglavlje 3. Modeliranje i simuliranje podatljivih mehanizama



Slika 3.7 Primjer prikaza rezultata korištenjem *GraphiX* programa.[3] Prikazuje se iznos pomaka čvorova u smjeru *Y* osi jednostavnog podatljivog mehanizma.

matplotlib [17] zbog veće mogućnosti podešavanja prikaza i kompatibilnosti s ostatkom programa.

4 Dizajn podatljivih mehanizama pomoću optimizacije

Kod dizajna podatljivog mehanizma možemo imati različite pristupe. Neki od pristupa dizajnu mehanizama prikazani su u literaturi [1]. Primjerice, jedan od pristupa dizajnu mehanizma je topološka optimizacija koja mijenja postavljenu topologiju mehanizma s ciljem postizanja traženih kinematskih karakteristika mehanizma. Naravno ovisno o početnoj definiciji modela, potrebno je odabrati pravilnu metodologiju dizajna mehanizma. Zbog oblika postavljenog mehanizma, odabire se oblik strukturalne optimizacije gdje se na temelju optimizacijskih varijabli mijenjaju debljine greda u rešetkastoj mreži grednih elemenata.

4.1 Optimizacijske varijable

Kako je cilj ove optimizacije dobivanje elastičnih i kinematskih svojstava mehanizma, potrebno je odgovoriti na pitanje: "Promjenom kojih parametara utječemo na kinematska svojstva konstrukcije?" Glavni mehanizam dobivanja traženih pomaka je regulacija savijanja pojedinih greda unutar mehanizma.

Korištenjem *Bernoulli-Euler* jednadžbe:

$$\frac{d\vartheta}{ds} = \frac{M_0}{EI} \quad (4.1)$$

Lijeva strana jednakosti označava promjenu nagiba grede (ϑ) u ovisnosti o udaljenosti od početne točke grede (s). Vidimo kako promjena nagiba grede u ovisnosti o udaljenosti od početne točke grede ovisi o opterećenju (u ovo slučaju momentu M_0), materijalnim karakteristikama (Youngov modul E) te o geometrijskim karakteristikama (moment inercije površine I). Kako bi regulirali količinu savijanja grede, možemo mijenjati četiri parametra: opterećenje, duljinu grede, materijal grede ili vlastiti moment inercije grede. Opterećenja i duljine greda zadani su početnom definicijom parametara mehanizma, te njihovom promjenom mijenjamo namjenu i početni oblik mehanizma. Također, homogeni materijal od kojeg je cijeli mehanizam

Poglavlje 4. Dizajn podatljivih mehanizama pomoću optimizacije

izrađen ne može se mijenjati od grede do grede jer tako gubimo jednostavnost proizvodnje samog mehanizma i homogenost materijala kroz mehanizam. Preostaje jedino mijenjati geometrijsku karakteristiku pojedinih greda, a to je moment inercije pravokutnog presjeka grede.

Momenti inercije pravokutnog poprečnog presjeka glase:

$$I_x = \frac{hw^3}{12}, \quad I_y = \frac{h^3w}{12} \quad (4.2)$$

Momenti inercije površina(I_x, I_y) ovise o visini poprečnog presjeka grede (h) te o širini poprečnog presjeka grede (w). Uzimajući u obzir da svodimo mehanizam na gibanje unutar jedne ravnine, bitan nam je samo moment inercije površine oko osi x . Sam moment površine oko osi x ovisi o trećoj potenciji širine poprečnog presjeka h te o visini poprečnog presjeka w . Postavljanjem jednake visine svih greda mehanizma (veličine okomite na ravninu djelovanja mehanizma), ostaje nam jedino promjena širine pojedinih greda kao metoda regulacije savijanja mehanizma.

Kako je cilj optimizacijskog postupka dobivanje željenih elastičnih i kinematskih karakteristika mehanizma za optimizacijske varijable odabiru se vrijednosti debljina greda koje se lako mogu prikazati vektorom:

$$\mathbf{X} = \{X_1, X_2, \dots, X_m\} \quad (4.3)$$

Tijekom optimizacijskog procesa (promjene optimizacijskih varijabli) gredama se može dodjeliti debljina manja od minimalne moguće debljine greda w_{\min} . U tom slučaju, debljina grede se postavlja na 0 te se greda privremeno miče iz rešetke mehanizma i ne uzima se u obzir kod proračuna, s mogućnošću ponovne "aktivacije" grede u slučaju da optimizacijski algoritam gredi dodijeli debljinu veću od w_{\min} . Svakako, mora se uzeti u obzir da su grede na kojima se nalaze oslonci, opterećenja ili kod čijih se točaka traži određeni pomak kod deformacije mehanizma izuzete iz ovog pravila u slučaju da kod njihovog micanja nestaje i čvor s definiranim ograničenjima. Njihova debljina ne može biti manja od zadane minimalne debljine grede.

Moguće je također definirati grede čija se debljina neće mijenjati tijekom proračuna, a grede se neće moći maknuti.

Maksimalne i minimalne debljine greda w_{\max} i w_{\min} različite su od slučaja do slučaja, te će se definirati naknadno uz definiciju ostalih parametara promatranog

mehanizma. Generalno, maksimalna debljina greda jednaka je visini mehanizma (h), dok je minimalna debljina greda ograničena mogućnostima izrade mehanizma, što je u ovom slučaju preciznost i finoća ispisa na 3D printeru.

4.2 Ciljevi

Kako je cilj optimizacije postizanje svojstva mehanizma, funkcija cilja mora sadržavati način provjere odstupanja dobivenih vrijednosti pomaka čvorova od traženih vrijednosti pomaka čvorova. Također, potrebno je minimizirati ukupnu masu mehanizma, kako bi povećali broj maknutih greda te osigurali fleksibilnost cijelog mehanizma. Minimizacija mase može se lako svesti na minimizaciju volumena mehanizma, kako se mehanizam sastoji od homogenog materijala. Minimizacija mase također rješava sporedni problem *bесmисlenih greda* koje su jednim krajem vezane za strukturu mehanizma, dok je drugi kraj grede neopterečen i nevezan za strukturu.

Funkcija cilja kombinacija je triju vrijednosti odstupanja pozicije promatranih točaka nakon analize mehanizma od zadanog položaja, promjena volumena mehanizma i dodatne uvjete ovisne o samom mehanizmu. Javlja se problem drastične razlike u redu veličine pojedinih ciljeva. Ovaj problem izbjegava se svođenjem traženih vrijednosti na bezdimenzijski oblik. Zadaje se sljedeća funkcija cilja:

$$F_V = \frac{V}{V_{\max}} \quad (4.4)$$

$$F_u = \sum_{i=1}^n \left(\frac{\mathbf{u}_i - \bar{\mathbf{u}}_i}{\epsilon} \right)^2 \quad (4.5)$$

$$F_{\text{goal}} = \omega_V \cdot F_V + \omega_u \cdot F_u + \omega_\alpha \cdot F_\alpha \quad (4.6)$$

Potpuna funkcija cilja (F_{goal}) sastoji se od zbroja umnožaka pojedinih funkcija cilja (F_i) te njihovih težinskih faktora (ω_i). Funkcija cilja minimizacije mase mehanizma (4.4) zadaje se kao omjer trenutnog iznosa volumena mehanizma (V) i maksimalnog mogućeg volumena mehanizma (V_{\max}) dok je funkcija minimizacije odstupanja pomaka (4.5) definirana kvadratom skalirane razlike vektora traženih

pomaka (\mathbf{u}_i) i vektora zadanih krajnjih pomaka ($\bar{\mathbf{u}}_i$). Također se nadodaje mogućnost postavljanja dodatnih funkcija cilja (F_α). Kod vektora pomaka točaka, uzima se kvadrat razlike izračunatih od traženih pomaka podijeljenih koeficijentom ϵ kako bi se postigao rezultat kuncije cilja istog reda veličine.

4.3 Ograničenja

Kako bi se održala elastičnost mehanizma i kako bi se mehanizam očuvao od puknuća kojeg člana, postavlja se ograničenje mogućeg postignutog naprezanja nešto manje od granice elastičnosti materijala. Mora se uzeti u obzir da printani mehanizam zbog same prirode 3D ispisa neće biti savršeno homogen i izotropan za razliku od ljevanja plastike, te dodatno smanjiti granicu elastičnosti.

Kako bi zadovoljili ovo ograničenje, za proračun ekvivalentnog naprezanja koristi se teorija najveće distorzijske energije (teorija Von Misesa). Ekvivalentno naprezanje računa se temeljem:

$$\sigma_{\text{ekv}} = \sqrt{\frac{1}{2} [(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)]} \quad (4.7)$$

U ovoj jednadžbi σ_i predstavlja normalno naprezanje u smjeru lokalne osi i i dok τ_{ij} predstavlja tangencijalna naprezanja. Kod i i j vrijedi $i, j \in \{x, y, z\}$.

Prati se maksimalno ekvivalentno naprezanje greda unutar konstrukcije gdje nam ekvivalentno naprezanje σ_{ekv} bilo koje grede konstrukcije ne smije biti veće od maksimalnog naprezanja σ_{max} .

$$\max \{ \sigma_{\text{ekv}} \} < \sigma_{\text{max}} \quad (4.8)$$

Također postoji mogućnost da micanjem greda u potpunosti maknemo vezu između čvorova koji imaju definirane početne uvijete i čvorova kod kojih tražimo definirani pomak. Ovaj problem eliminiramo provjerom krajnjih pomaka definirane konstrukcije unutar neke evaluacije tijekom iteracija optimizacije. Ako su dobiveni pomaci traženih čvorova jednaki nuli, a uvijek očekujemo nekakav rezultat, znamo da je evaluacijski korak pogrešan, te se njegovi rezultati ne uvode u daljnji proračun lokalnog optimuma.

4.4 Multi-Scale Grid Search algoritam

Algoritam *Multi-Scale Grid Search* (MSGs) jedan je od algoritama lokalne optimizacije uključenih u *Indago* biblioteci [4]. Algoritam je varijacija Pattern Search algoritma.

Algoritmu *MSGs* mogu se dodijeliti 2 parametra: parametar n koji regulira veličinu mreže u svim dimenzijama te $xtol$ koji daje dodatan kriterij zaustavljanja mreže. Algoritam podijeli mrežu u smjeru svih dimenzija (m dimenzionalni slučaj podrazumjeva m optimizacijskih varijabli) na $2n + 1$ podjela. Kao i kod svakog algoritma moguće je postaviti početni vektor optimizacijskih varijabli $X0$ od kojeg optimizacija kreće, u protivnom kreće iz neke nasumične točke, tj. nasumične vrijednosti početnog vektora u zadanim optimizacijskim granicama.

4.4.1 Početni odabir smjera kretanja lokalne optimizacije

Nakon proračuna vrijednosti funkcije cilja za početne vrijednosti optimizacijskih varijabli, računa se funkcija cilja u okolnim diskretiziranim vrijednostima na mreži. Dvodimenzionalan primjer odabira početnog smjera prikazan je na slici 4.1.

Na temelju usporedbe iznosa funkcije cilja okolnih točaka i početne točke, odabire se smjer kretanja optimizatora predstavljen vektorom ΔY .

$$\Delta Y = \begin{Bmatrix} \Delta i_1 \\ \Delta i_2 \\ \vdots \\ \Delta i_m \end{Bmatrix} \quad (4.9)$$

U 2D slučaju prikazanom na slici 4.1 dan je primjer vektora ΔY koji iznosi:

$$\Delta Y = \begin{Bmatrix} (i + 1) - (i) \\ (j + 1) - (j) \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \quad (4.10)$$

Kod prvog proračuna smjera kretanja lokalne optimizacije u m -dimenzionalnoj mreži, evaluira se $2m + 1$ slučajeva: $2m$ proračuna kod kojih variramo vrijednosti

optimizacijskih varijabli za iznos ΔX te naravno vrijednosti početnog optimizacijskog vektora.

4.4.2 Nastavak kretanja optimizacije

Nakon dobivenog smjera kretanja lokalne optimizacije, sljedeća iteracija optimizacije kreće s evaluacijom funkcije cilja za vrijednosti optimizacijskih varijabli jednake zbroju dvostrukog vektora smjera i trenutnih vrijednosti optimizacijskih varijabli. Vektor smjera se udvostručuje kako bi ubrzali "koračanje" u smjeru dobrog rješenja (Slika 4.2).

$$\mathbf{X0}_1 = \mathbf{X0}_0 + 2 \cdot \Delta \mathbf{Y}_0 \quad (4.11)$$

U slučaju da je vrijednost funkcije cilja kod trenutne evaluacije manja od prethodnog minimalnog rješenja (najboljeg) zadržava se jednaki vektor smjera. U protivnom se postavlja šablona koja se sada sastoji od m točaka tj. m evaluacija funkcije cilja prikazana na slici 4.3.

Pronalazi se novi vektor smjera kretanja optimizacije te se iteracija ponavlja. U slučaju dobivanja lokalnog optimuma, mreža se ugušćuje te se nastavlja potraga za optimalnom vrijednosti.

4.4.3 Ugušćivanje mreže

Mreža se ugušćuje 8 puta, tj. $\Delta X_{\text{new}} = \Delta X_{\text{old}}$ te se postavlja tako da se njeno središte nalazi na poziciji lokalnog optimuma grube mreže tj. trenutne točke evaluacije, broj diskretnih vrijednosti se ne mijenja. Na novodobivenoj domeni ponovo pokrećemo iteracijski postupak kretanja po domeni i traženja lokalnog optimuma. U slučaju evaluacije ruba novodobivene domene, domena jednake veličine i diskretizacije postavlja se sa svojim središtem u točki trenutne evaluacije.

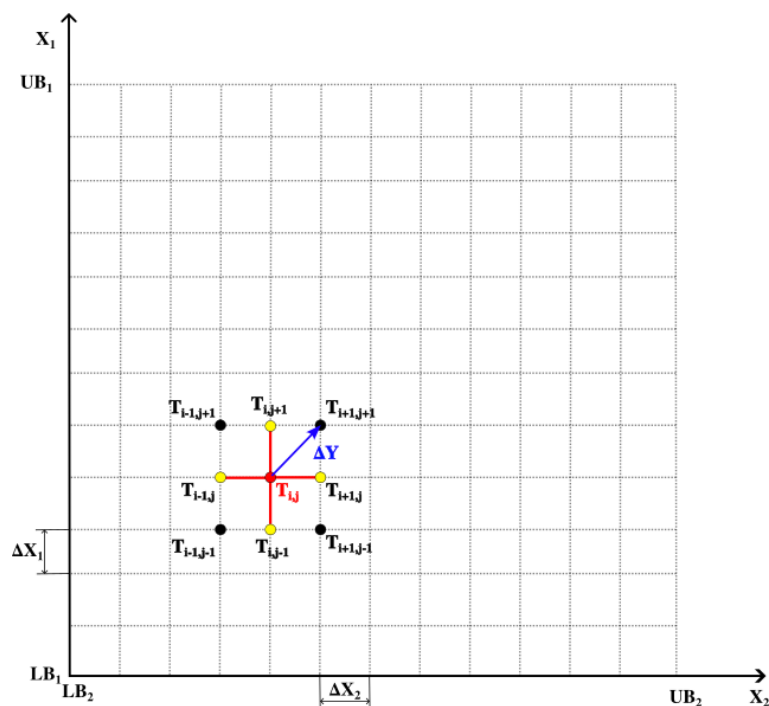
Gušća mreža se tako može pomaknuti 2 puta, te se zatim veličina domene, a s njom i korak udvostručuje (zbog bržeg kretanja ka optimalnom rješenju). U slučaju

Poglavlje 4. Dizajn podatljivih mehanizama pomoću optimizacije

postizanja lokalnog optimum gušće mreže, mreža se dodatno ugušćuje te se postupak ponavlja.

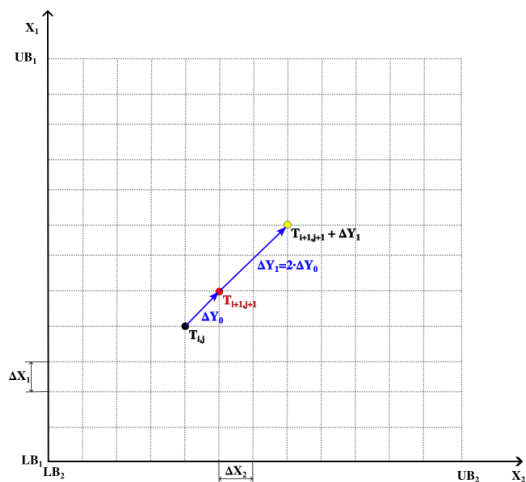
Kompletan opisani postupak se ponavlja sve dok vrijednost ΔX ne padne ispod zadanog *x_{tol}* parametra ili broj evaluacija/iteracija ne postigne svoj maksimum.

Poglavlje 4. Dizajn podatljivih mehanizama pomoću optimizacije

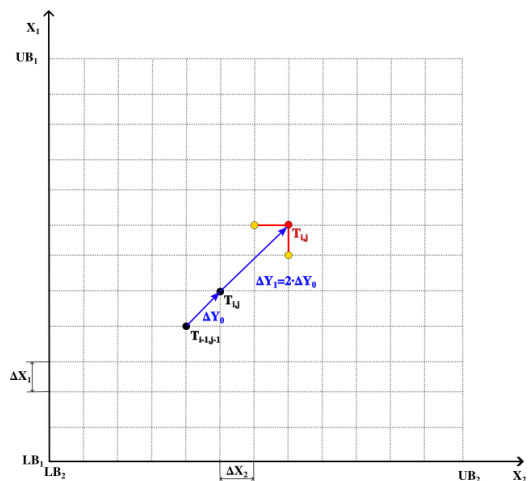


Slika 4.1 Prikaz mreže diskretiziranih vrijednosti optimizacijskih varijabli X_1 i X_2 . Gornje i donje granice mreže označene su UB kao *upper bound* i LB kao *lower bound*. Crvene linije predstavljaju početnu šablonu tj. točke na temelju kojih će se računati početni vektor kretanja optimizacije. Točka koja je označena crvenom bojom predstavlja početnu točku optimizacije ($T_{i,j}$), dok ostale točke ($T_{i\pm 1,j\pm 1}$) predstavljaju moguću slijedeću početnu poziciju unutar iteracije. ΔX predstavlja razmak između diskretiziranih vrijednosti optimizacijskih varijabli koji unutar optimizacijskog stupnja ostaje konstantan. ΔY prikazuje smjer kretanja u slučaju kad su rezultati funkcije cilja na pozicijama $T_{i+1,j}$ i $T_{i,j+1}$ bolji od svih ostalih rezultata te od iznosa funkcije cilja u početnoj točki $T_{i,j}$.

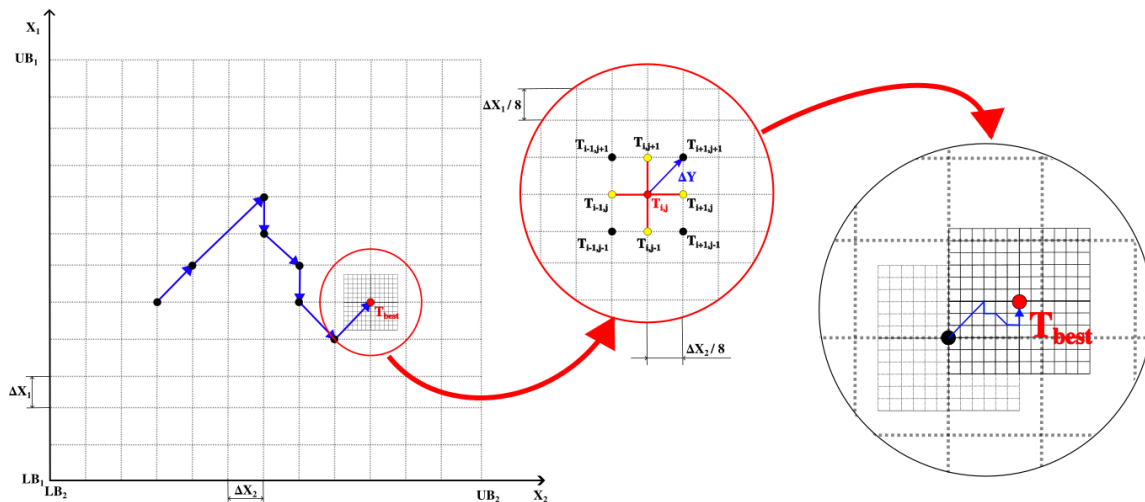
Poglavlje 4. Dizajn podatljivih mehanizama pomoću optimizacije



Slika 4.2 Nastavak kretanja optimizacije dvostrukim korakom u smjeru prvog vektora smjera.



Slika 4.3 Šablona koja se sastoji od m evaluacija.



Slika 4.4 Prikazano je ugušćivanje mreže u slučaju postizanja lokalnog minimuma kod grube mreže. Također prikazan je i način na koji se nanovo diskretizirano područje pomiče kod evaluacije točke na rubu diskretizirane domene.

4.5 Višefazna optimizacija i redukcija broja optimizacijskih varijabli

Kako su optimizacijski problemi za koje se žele postići optimizirano rješenje podosta veliki i sadrže veliki broj optimizacijskih varijabli, istražuje se mogućnost micanja nekog broja optimizacijskih varijabli iteracijskim postupkom.

Postavljen je sljedeći postupak: U prvom stupnju pokreće N lokalnih optimizacija kod kojih svaki početni optimizacijski vektor \mathbf{X}_z , $z \in [0, N]$ sadrži m optimizacijskih varijabli nasumičnih vrijednosti. Dobiva se N rješenja lokalnih *MSG* optimizacija zadanog mehanizma. Uz optimizirane vrijednosti optimizacijskih varijabli $\bar{\mathbf{X}}_z$ dobivamo i iznose funkcije cilja optimiziranog mehanizma f_z .

Kako bi se smanjio broj optimizacijskih varijabli u sljedećem stupnju optimizacije, postavlja se podešeni postupak težinske sume optimiziranih vrijednosti optimizacijskih varijabli. Najbolje ocijenjeno rješenje lokalnih optimizacija sadrži neki broj varijabli koje su manje od minimalne moguće vrijednosti. Kako se u sljedećem stupnju algoritma miče određeni broj varijabli koje su manje od minimalne moguće vrijednosti, postoji mogućnost da trenutno optimalno rješenje makne pre veliki broj optimizacijskih varijabli. U tom slučaju optimizacija se svodi na daljnje rafiniranje već postojećeg lokalnog optimuma. Kako bi se taj slučaj izbjegao vrijednosti optimiziranih varijabli (isključujući najbolje rješenje) modificiraju se množenjem s težinskim faktorom w_z . Svakom od vektora optimiziranih vrijednosti optimizacijskih varijabli dodjeljuje se težina na temelju odstupanja od najbolje vrijednosti funkcije cilja

$$w_z = \frac{\min(\mathbf{f})}{f_z} \quad (4.12)$$

kod koje $\min(\mathbf{f})$ predstavlja minimalni iznos vektora vrijednosti funkcije cilja za svih N optimizacija tj. najbolji iznos funkcije cilja za svih N optimizacija unutar stupnja algoritma.

$$\mathbf{X}_\epsilon = \sum_{z \in [0, N] \setminus b} \bar{\mathbf{X}}_z \cdot w_z, \quad \bar{\mathbf{X}}_b = \bar{\mathbf{X}}_{\text{best}} \quad (4.13)$$

Zbrajanjem težinski podešenih optimiziranih varijabli (isključujući naravno

Poglavlje 4. Dizajn podatljivih mehanizama pomoću optimizacije

najbolje rješenje) dobiva se vektor \mathbf{X}_ϵ . Vrijednosti vektora najboljeg rješenja $\bar{\mathbf{X}}_{\text{best}}$ koje su manje od minimalne dopuštene vrijednosti x_{\min} mijenjaju se za vrijednosti vektora \mathbf{X}_ϵ . Ovako izmjenjeno rješenje zadržava karakteristike najboljeg rješenja, uz postavljanje mogućnosti vraćanja optimizacijskih varijabli koje je glavno rješenje označio kao "nepotrebnim" tj. optimizacijske varijable čija je vrijednost manja od minimalne dopuštene vrijednosti.

$$X_{1_i} = \begin{cases} \bar{X}_{\text{best},i} & \text{ako je } x_{\min} \leq \bar{X}_{\text{best},i} \leq x_{\max} \\ X_{\epsilon,i} & \text{ako je } \bar{X}_{\text{best},i} \leq x_{\min}, \text{ te ako je } x_{\min} \leq X_{\epsilon,i} \\ x_{\max} & \text{ako je } \bar{X}_{\text{best},i} \leq x_{\min}, \text{ te ako je } X_{\epsilon,i} \geq x_{\max} \\ \text{ne postoji,} & \text{inače} \end{cases} \quad (4.14)$$

Indeks i označava redni broj varijable unutar vektora vrijednosti. U slučaju da je vrijednost $X_{\epsilon,i}$ veća od w_{\max} varijabla poprima vrijednost w_{\max} . Ovako definirana težinska suma pokazatelj je "potrebnih" i "nepotrebnih" varijabli tj. u ovom slučaju greda.

Novodobiveni vektor ima manje dimenzija od vektora korištenog u prvom stupnju optimizacije. Sljedeći stupanj optimizacije ponovo pokreće N lokalnih optimizacija, ali ovaj put s manjim optimizacijskim vektorom, koji se sastoji od m_2 optimizacijskih varijabli. Dobiveni vektor $\mathbf{X1}$ koristi se kod prve lokalne optimizacije kao početni vektor, dok ostale $N - 1$ lokalne optimizacije kreću s nasumičnim vrijednostima vektora optimizacijskih varijabli jednakih dimenzija kao i $\mathbf{X1}$.

Dobivajući rezultate lokalnih optimizacija 2. stupnja optimizacije, ponavljamo postupak micanja optimizacijskih varijabli. Iteriranje postupka traje sve dok rezultati svih lokalnih optimizacija najnovijeg stupnja optimizacije upućuju na jedno jedinstveno rješenje.

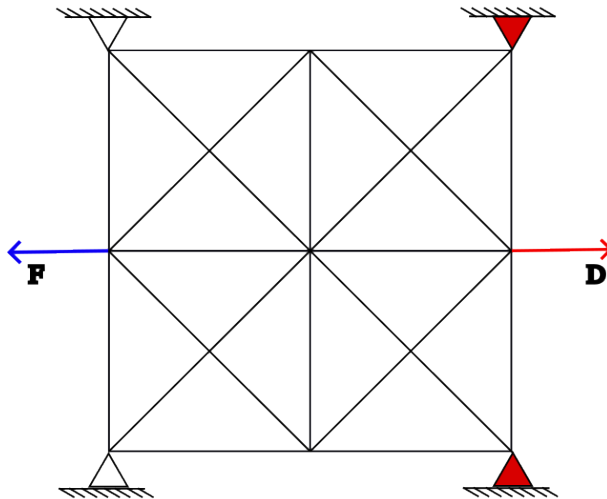
5 Primjeri

Sljedeći primjeri služe kao testni primjeri opisanog optimizacijskog procesa koristeći parametriziranu definiciju gredne rešetke. Oba primjera rađena su s mogućnošću lake izrade prototipa 3D ispisom. Zbog toga se koriste materijalne karakteristike PLA plastike kod koje su:

Modul elastičnosti	$E = 2.89 \cdot 10^9 \text{Pa}$
Poissonov koeficijent	$\mu = 0.2$

5.1 Primjer 1. - podatljivi invertor pomaka

Ovim primjerom analizira se osnovna primjena opisanih optimizacijskih algoritama. Bazna rešetka ovog slučaja prikazana je na slici 5.1.



Slika 5.1 Početna mreža mehanizma klješta

Cilj prikazanog mehanizma je postizanje pomaka D kod opterećivanja mehanizma silom F . Dakle traži se pomak čvora nasuprot čvoru u kojem je zadana sila. Traženi pomak je u smjeru suprotnom od smjera sile.

Poglavlje 5. Primjeri

Postavljaju se 4 nepomična oslonca od kojih se desni oslonci (prikazani crvenom bojom) ne smiju micati tokom optimizacijskih procesa.

Kod lokalnih optimizacija unutar stupnjeva algoritma smanjenja broja optimizacijskih varijabli većina parametara se ne mijenja. Konstantni parametri prikazani su u sljedećoj tablici:

Tablica 5.1 Konstantni parametri korišteni kod lokalnih optimizacija.

Naziv	Vrijednost
Maksimalni broj iteracija	300
Maksimalni broj evaluacija	3000
Maksimalni zastalih evaluacija	300
Broj dretvi kod paralelnog proračuna	48
Donja granica optimizacijskih varijabli	0 mm
Gornja granica optimizacijskih varijabli	5 mm
MSGs parametar n	11
MSGs parametar $xtol$	1e-4

Iz stupnja u stupanj broj greda se smanjuje na temelju prethodno definirane težinske sume vrijednosti optimizacijskih varijabli 9 lokalnih optimizacija i najbolje vrijednosti optimizacijskih varijabli. Zbog toga, potrebno je definirati manji broj dimenzija optimizacijskog vektora kao i manji optimizacijski vektor X_0 lokalne optimizacije u narednim stupnjevima algoritma.

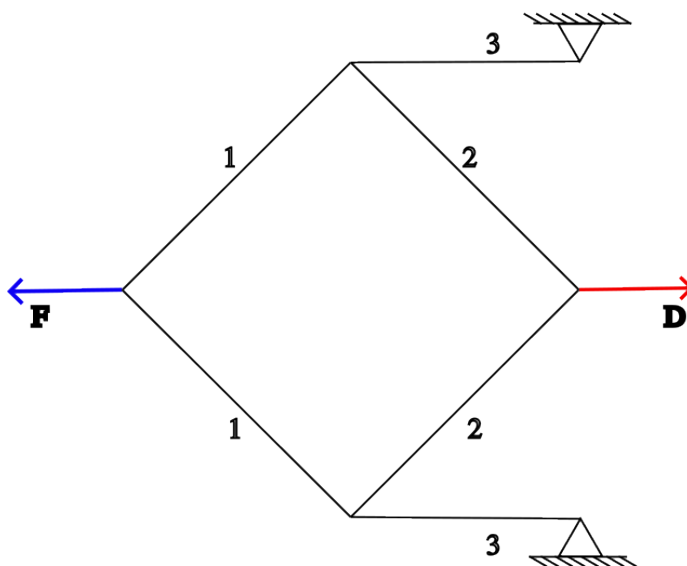
Mehanizam se u početku sastoji od 28 greda, a cilj opisanog algoritma je smanjiti količinu "potrebnih" greda na minimum. Postavlja se optimizacija u 6 stupnjeva s 10 lokalnih optimizacija unutar jednog stupnja. Kako bi provjerili mogućnost postojanja rješenja slučaja, postavlja se jednostavan mogući oblik mehanizma za koji provjeravamo maksimalni mogući pomak traženog čvora za zadanu silu.

5.1.1 Mogući oblik podatljivog mehanizma

Potrebno je postaviti nekakav jednostavan podatljivi mehanizam na temelju prijašnjeg znanja kako bi odredili krajnje granice deformacija postavljenog slučaja. Na temelju dobivenih vrijednosti moguće je odrediti funkcije cilja kod optimizacije podatljivog mehanizma.

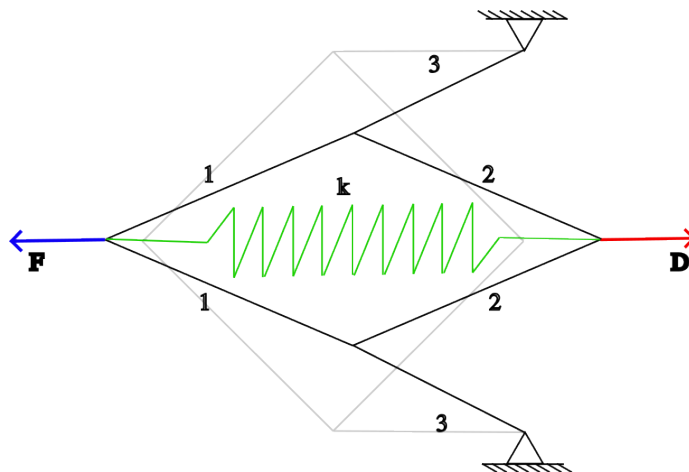
Poglavlje 5. Primjeri

Za razliku od podatljivih mehanizama kod kojih se pomak čvorova dobiva temeljem deformacija mehanizma (u našem slučaju velikih deformacija koje rezultiraju velikim pomacima), izračun pomaka čvorova krutog mehanizma je relativno jednostavan. Kruti mehanizam sastoji se od krutih članova međusobno spojenih zglobovima. Kako bi se jednostavnije prikazalo samo ponašanje mehanizma, možemo ga (uz par preinaka) prikazati krutim mehanizmom. Zamišljeni dizajn mehanizma prikazan je na slici 5.2.

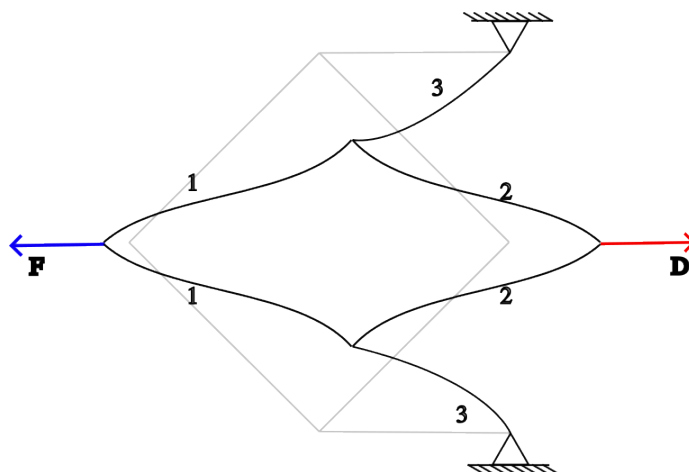


Slika 5.2 Jednostavni dizajn potrebnog podatljivog mehanizma. Zbog geometrijske simetrije mehanizma, grede označene istim brojevima istih su debljina.

Ovako definiranom mehanizmu zadajemo debljine greda 1, 2 i 3 u rasponu od 0.5mm do 5mm diskretiziranom na korake od 0.5mm. Kombiniranjem različitih konfiguracija greda pokreće se 1000 nelinearnih proračuna deformacije konstrukcije. Pokretanjem proračuna maksimalni otklon tražene točke dobiva se za minimalni iznos debljina svih greda tj. 0.5 mm. Kod tako postavljene mreže proračun deformacije daje pomak od 1.4 cm što nam daje omjer zadanog pomaka (pomak čvora na kojem je postavljeno opterećenje) i dobivenog pomaka izlaznog čvora 160%. Ova vrijednost traženog pomaka zadaje se kao referentna vrijednost praćenog pomaka tijekom procesa optimizacije.



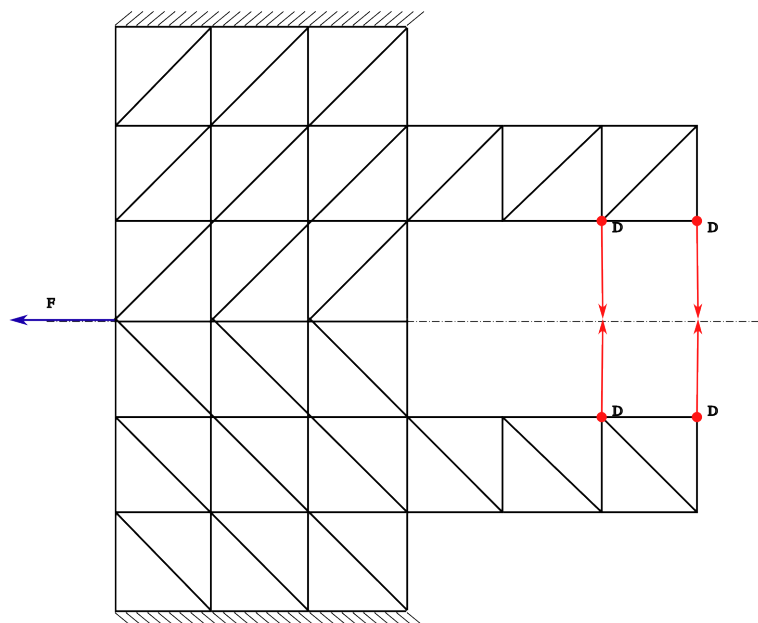
Slika 5.3 Pomaci točaka krutog mehanizma koji se sastoji od 6 gređa međusobno spojenim zglobovima. Postavljanjem opruge krutosti k postavlja se uvijet vraćanja mehanizma u početni položaj. Same debljine gređa nemaju nikakav utjecaj na djelovanje mehanizma (uzimajući u obzir da se gređe ne saviju pod izazvanim opterećenjem.)



Slika 5.4 Pretpostavljeni oblik deformiranog podatljivog mehanizma. Opruga postaje nepotrebna, kako umjesto da se energija pohrani u obliku unutarnje energije deformacije opruge, sada zapravo cijeli mehanizam sudjeluje u pohrani energije u obliku unutarnje energije deformacije podatljivog mehanizma.

5.2 Primjer 2. - Podatljiva hvataljka

Kako i samo ime govori, ovaj primjer je postizanje mehanizma koji će primjenom sile na jednom kraju prihvatiti neki objekt na drugome. Osnovna geometrija mehanizma prikazana je na sljedećoj slici.



Slika 5.5 Prikaz zamišljenih granica mehanizma, te početni i krajnji uvijeti.

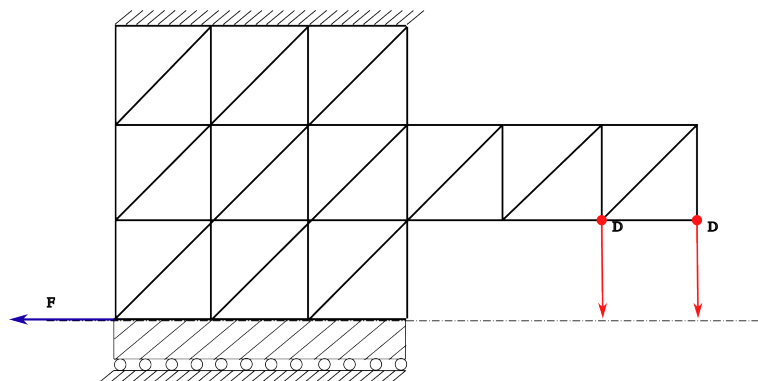
Primjenom sile F u horizontalnom smjeru, postizemo pomak točaka D u vertikalnom smjeru prihvaćajući tako neki predmet.

5.2.1 Postavljanje slučaja

Kako je slučaj koji se rješava simetričan u odnosu na središnju os, moguće je riješavati problem koristeći pola opisanog mehanizma te zadajući pravilne uvjete simetrije. Ovakav pristup ubrazava proračun se već u početku smanjuje broj optimizacijskih varijabli na pola.

Postavljaju se nepomični oslonci na gornji rub mehanizma te klizni oslonci (uvijet simetrije) na donji rub mehanizma. Klizni oslonci predstavljaju uvijet

Poglavlje 5. Primjeri



Slika 5.6 Prikaz zamišljenih granica mehanizma, te početni i krajnji uvijeti.

simetrije. Pomično uklještenje (klizni oslonac) dopušta pomak čvora u smjeru globalne horizontalne osi, te sprječava pomak u smjeru globalne vertikalne osi i rotaciju oko osi okomite na ravninu mehanizma.

Ponovo se kod lokalnih optimizacija koriste jednaki paramteri neovisno o stupnju iteracije prikazani u tablici 5.2

Tablica 5.2 Konstantni parametri korišteni kod lokalnih optimizacija drugog slučaja.

Naziv	Vrijednost
Maksimalni broj iteracija	300
Maksimalni broj evaluacija	3000
Maksimalni zastalih evaluacija	300
Broj dretvi kod paralelnog proračuna	48
Donja granica optimizacijskih varijabli	0 mm
Gornja granica optimizacijskih varijabli	5 mm
MSGGS parametar n	11
MSGGS parametar $xtol$	1e-4

Za razliku od prethodnog primjera, gdje se korištenjem više lokalnih optimizacija unutar jednog stupnja algoritma, u ovom primjeru koristi se 11 stupnjeva s sveukupno 5 lokalnih optimizacija po stupnju algoritma.

6 Rezultati

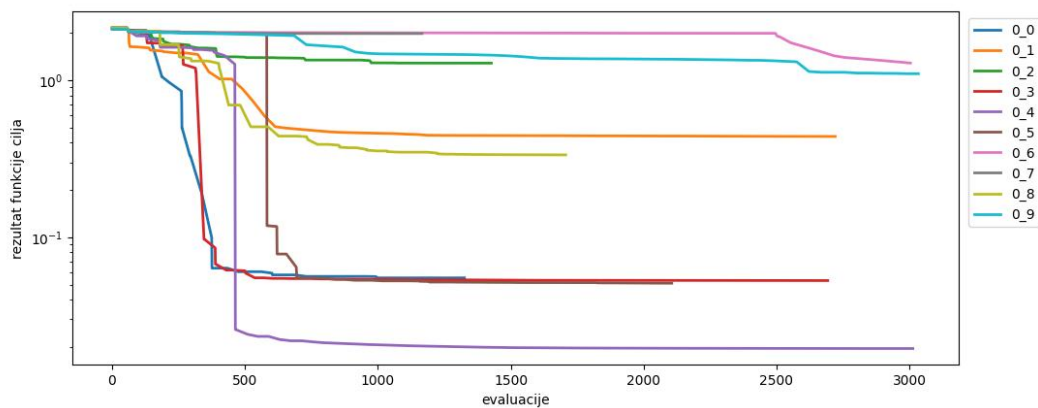
6.1 Rezultati optimizacije prvog primjera

6.1.1 Iznos funkcije cilja kod lokalnih optimizacija svakog stupnja optimizacije

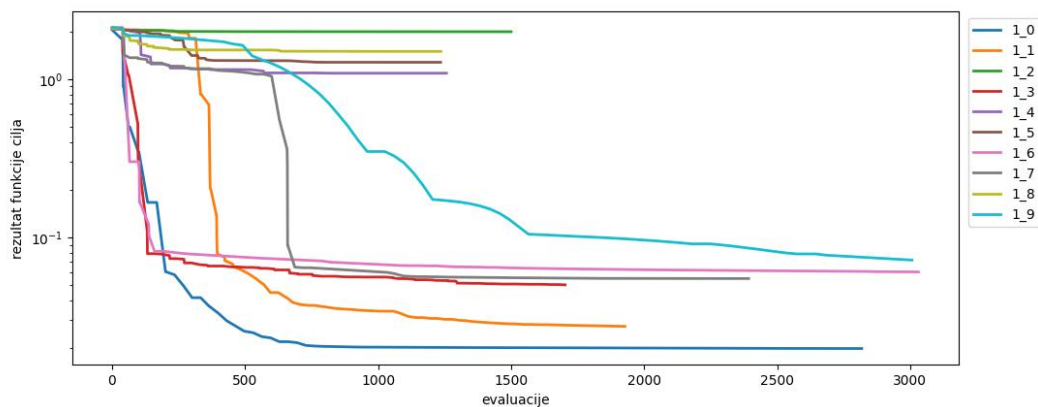
Kako se krećemo kroz pojedine stupnjeve optimizacije, zbog smanjenog broja varijabli možemo primjetiti sve bržu konvergenciju rezultata optimizacija. Također, smanjenjem nepoželjnih greda (smanjenjem broja optimizacijskih varijabli) možemo primjetiti rijede zastoje kod optimizacije (stagnacija optimizacije zbog postizanja lokalnog minimuma neovisno o točnosti rješenja).

Neke od lokalnih optimizacija nisu konvergirale. Zbog nasumične prirode postavljanja početnih vektora optimizacijskih varijabli postavljenog problema, postoji mogućnost da početne vrijednosti kod nekih lokalnih optimizacija prouzroče konvergenciju u lokalni minimum čija vrijednost uvelike odstupa od optimalnog rješenja. U višim stupnjevima optimizacije primjećuje se manja učestalost ovakvih grešaka zbog manjeg broja optimizacijskih varijabli.

Poglavlje 6. Rezultati

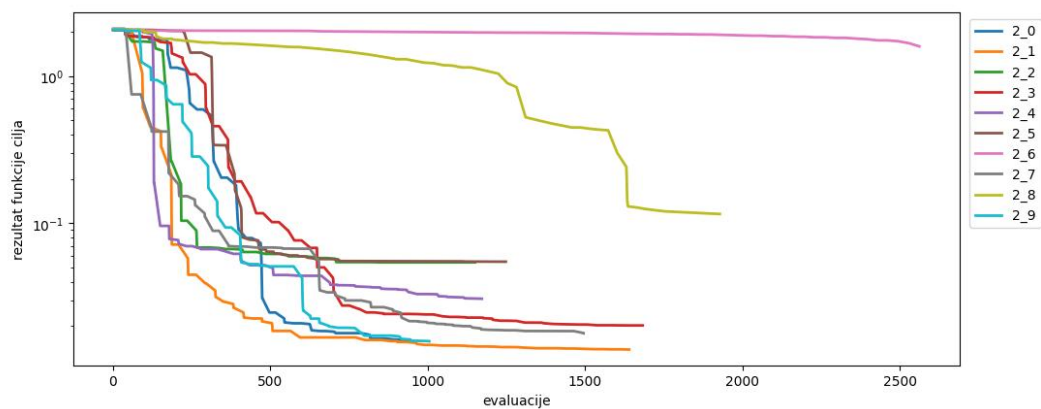


Slika 6.1 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija prvog stupnja algoritma. Vrijednosti svih lokalnih optimizacija su konvergirale. Ovaj stupanj lokalnih optimizacija optimizira vektor od 28 varijabli.

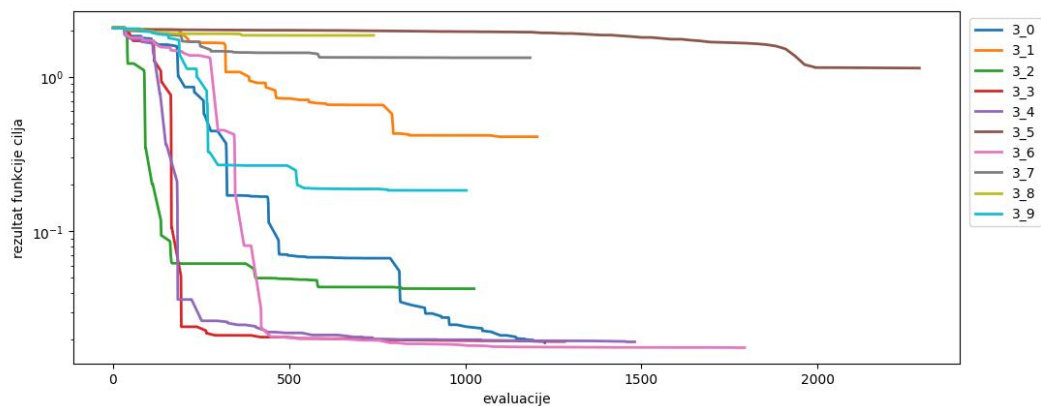


Slika 6.2 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija drugog stupnja optimizacije. Ovaj stupanj optimizira vrijednosti 19 optimizacijskih varijabli. Za razliku od prvog stupnja, više rezultata počinje težiti istoj vrijednosti.

Poglavlje 6. Rezultati

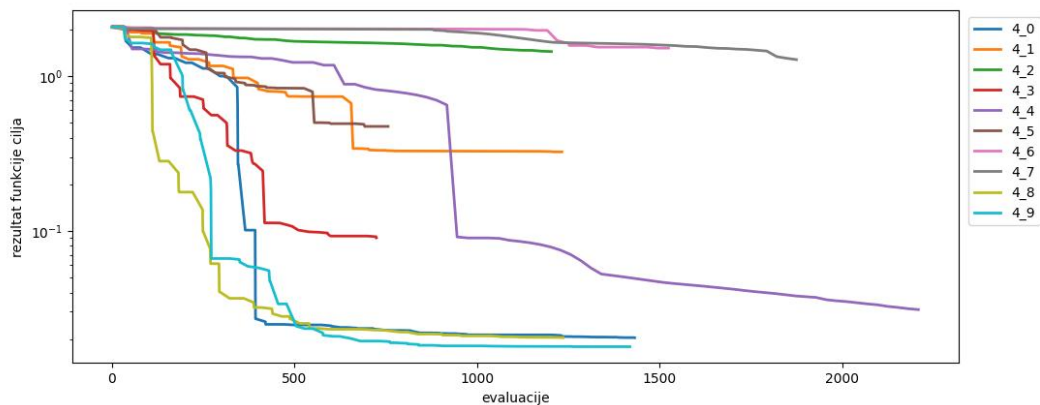


Slika 6.3 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija trećeg stupnja algoritma. Ovaj stupanj optimizira vrijednosti 18 optimizacijskih varijabli. Za razliku od prva dva stupnja, rezultati vrlo brzo konvergiraju, te samo dvije lokalne optimizacije nisu konvergirale.

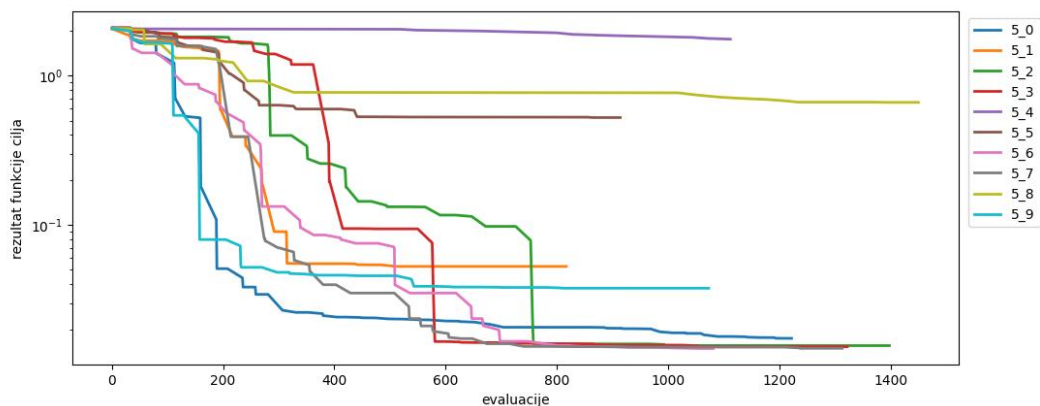


Slika 6.4 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija četvrtog stupnja algoritma. Stupanj optimizira vrijednosti 16 optimizacijskih varijabli. Za razliku od prethodnog stupnja, vidljivo je kako veliki broj lokalnih optimizacija nije konvergiralo u blizinu minimalne vrijednosti funkcije cilja.

Poglavlje 6. Rezultati



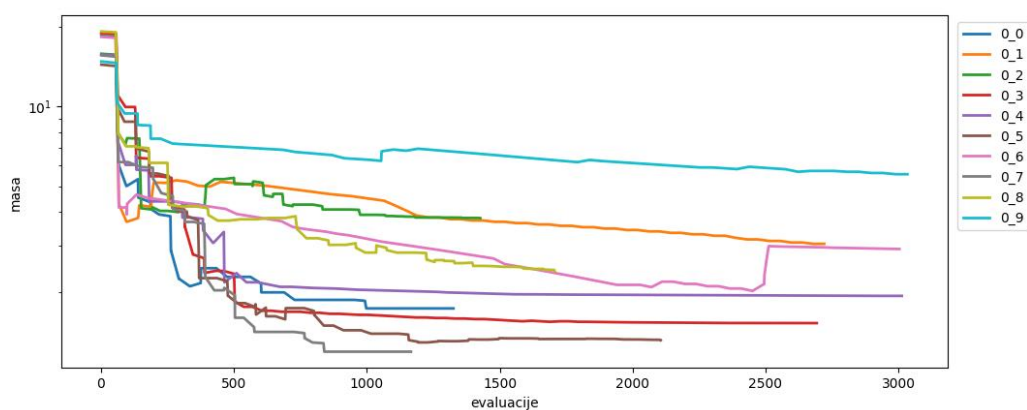
Slika 6.5 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija petog stupnja algoritma. Algoritam je zadržao sve optimizacijske varijable, te nastavlja optimizaciju s 16 optimizacijskih varijabli.



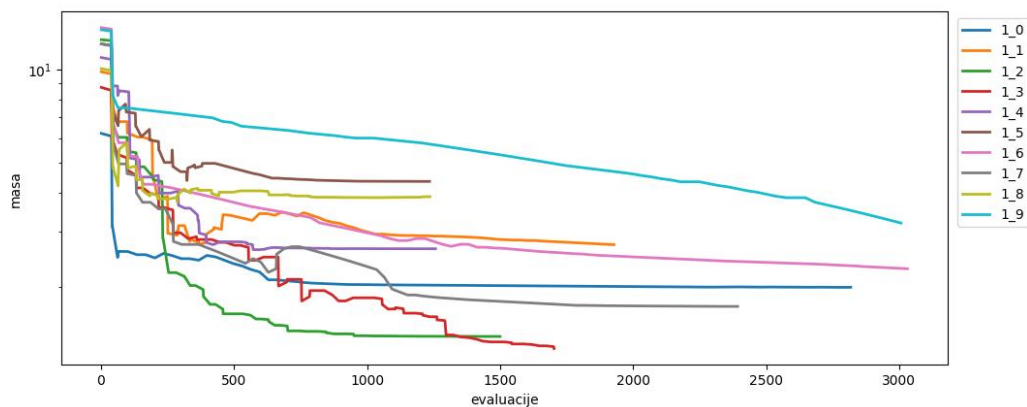
Slika 6.6 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija šestog stupnja algoritma. Kao i kod prethodnog stupnja, optimizira se 16 optimizacijskih varijabli. Na temelju zadržavanja svih 16 optimizacijskih varijabli kroz protekla 3 stupnja algoritma, može se pretpostaviti kako daljnjeg smanjenja optimizacijskog vektora neće biti, te se algoritam zautavlja.

6.1.2 Iznos mase kod lokalnih optimizacija svakog stupnja optimizacije

Masa mehanizma iako ima manji utjecaj na iznos funkcije cilja, unutar svih optimizacija mora konvergirati te se konstantno smanjivati. Trend minimizacije mase može se primijetiti kod svih stupnjeva algoritma.

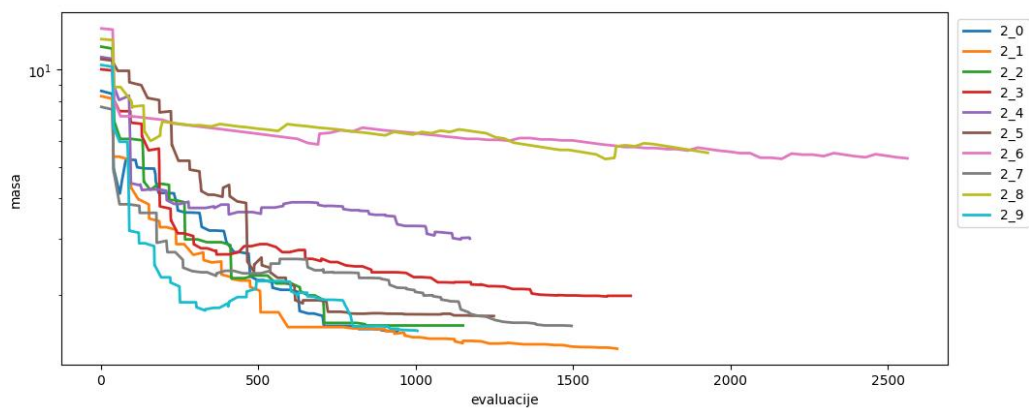


Slika 6.7 Graf prikazuje iznos mase mehanizma kod prvog stupnja algoritma.

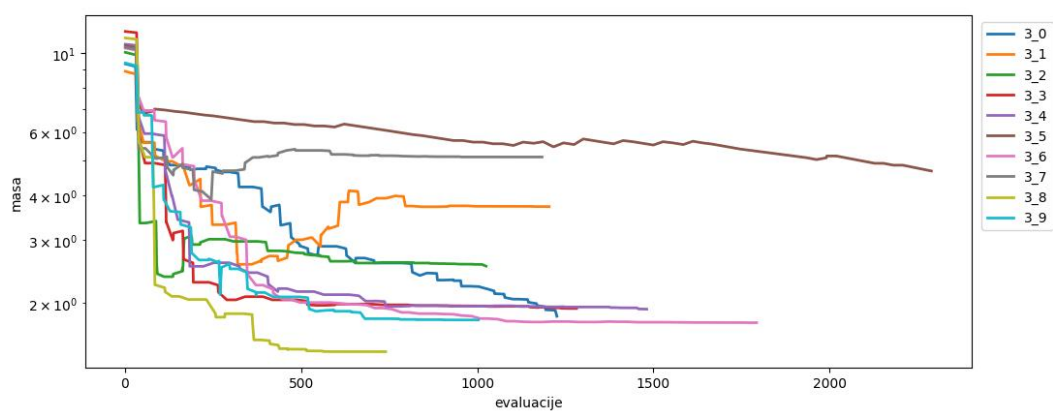


Slika 6.8 Graf prikazuje iznos mase mehanizma kod drugog stupnja algoritma.

Poglavlje 6. Rezultati

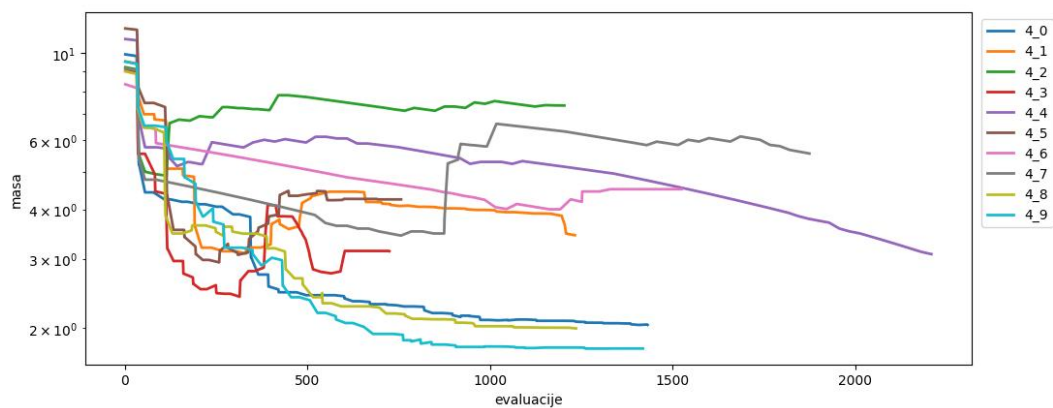


Slika 6.9 Graf prikazuje iznos mase mehanizma kod trećeg stupnja algoritma.

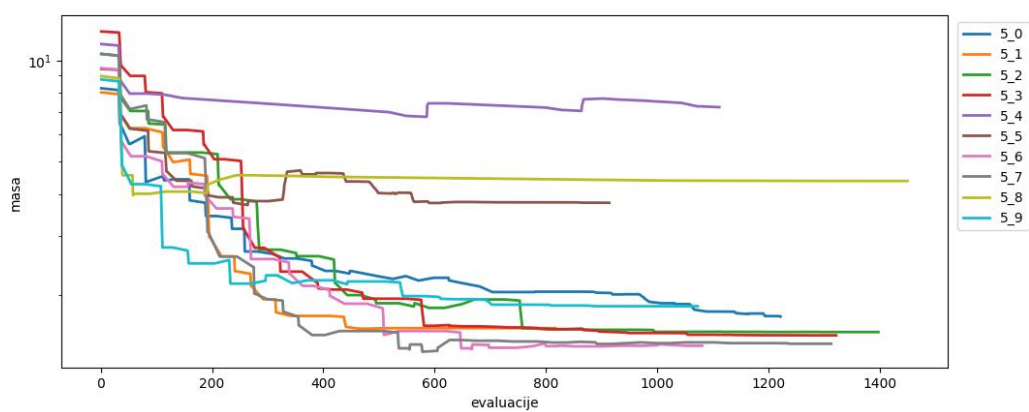


Slika 6.10 Graf prikazuje iznos mase mehanizma kod četvrtog stupnja algoritma.

Poglavlje 6. Rezultati



Slika 6.11 Graf prikazuje iznos mase mehanizma kod petog stupnja algoritma.



Slika 6.12 Graf prikazuje iznos mase mehanizma kod šestog stupnja algoritma.

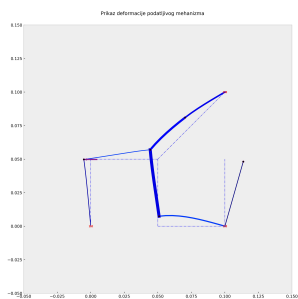
6.1.3 Rješenja optimizacija unutar svakog stupnja algoritma

Sljedeći grafovi prikazuju izgled mehanizma na temelju svake lokalne optimizacije unutar stupnja algoritma. Na temelju ovih grafova može se potvrditi konvergencija rješenja funkcija cilja unutar stupnjeva algoritma. Također, na temelju sljedećih grafova vidljivo je zbog čega su određene lokalne optimizacije u početnim stupnjevima algoritma divergirale ili konvergirale u vrlo visoku vrijednost.

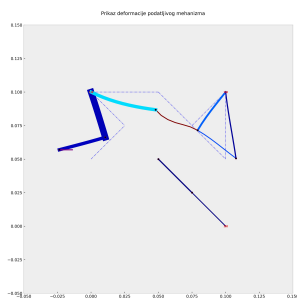
Kako je opisano prethodno u radu, jedno od ograničenja je maksimalno *Von Mises* naprezanje. Boja pojedinih greda na sljedećim grafovima definira naprezanje određene grede. Postizanje crvene boje označava ulazak u “kritično područje” koje svakako ne prelazi granicu elastičnosti mehanizma, ali predstavlja moguće probleme kod izrade i korištenja mehanizma. Postizanje kritičnog naprezanja označava moguće pojave trajnih deformacija nakon višekratne uporabe mehanizma, kao i moguće pozicije loma u slučaju zadavanja većeg opterećenja.

Na temelju posljednjeg stupnja algoritma moguće je odabrati dizajn koji se dobiva kao rješenje 6 od 10 lokalnih optimizacija posljednjeg stupnja kako zadovoljava sve zadane početne i krajnje uvijete analiziranog problema.

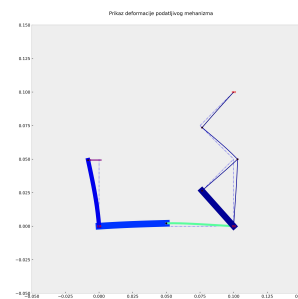
Poglavlje 6. Rezultati



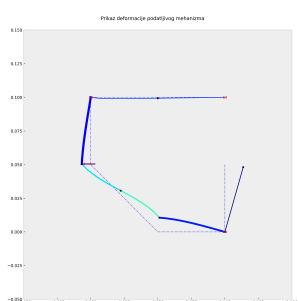
1. Lokalna optimizacija



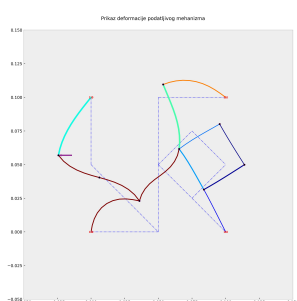
2. Lokalna optimizacija



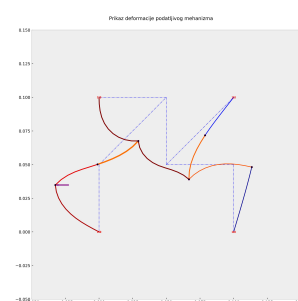
3. Lokalna optimizacija



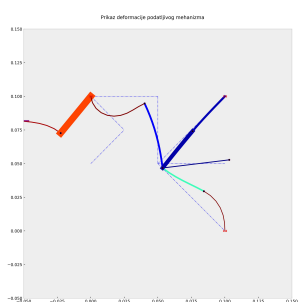
4. Lokalna optimizacija



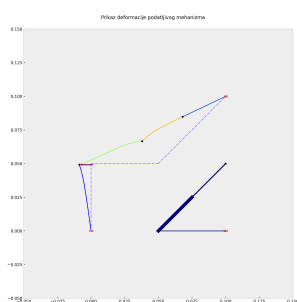
5. Lokalna optimizacija



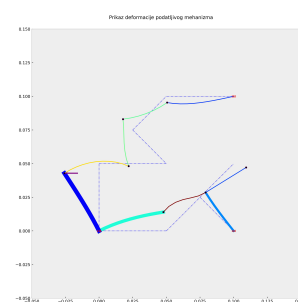
6. Lokalna optimizacija



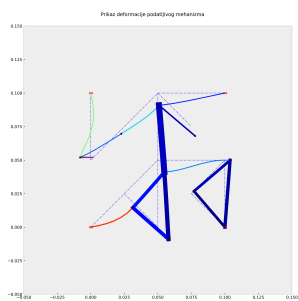
7. Lokalna optimizacija



8. Lokalna optimizacija



9. Lokalna optimizacija

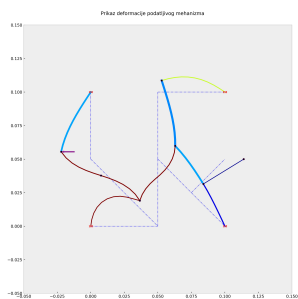


10. Lokalna optimizacija

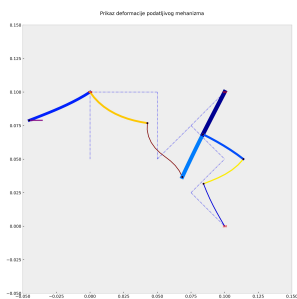
Slika 6.13 Prikaz rezultata prvog stupnja optimizacije.

Od 10 MSGS lokalnih optimizacija, svaka je optimizacija pronašla lokalni minimum. Pokazujući kako je problem koji riješavamo svakako multimodalan s većim brojem lokalnih optimuma. Dva rješenja (lokalna optimizacija 2 i optimizacija 8) su *nemoguće* rješenja kod kojeg je oslonac koji se ne smije maknuti iz proračuna odvojen od ostatka mehanizma. Također, kod tri rješenja (5, 6, 7) grede ulaze u definirano kritično područje naprezanja.

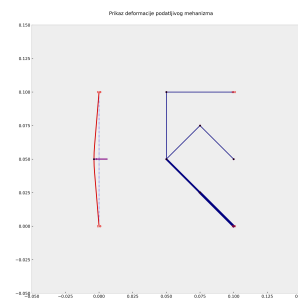
Poglavlje 6. Rezultati



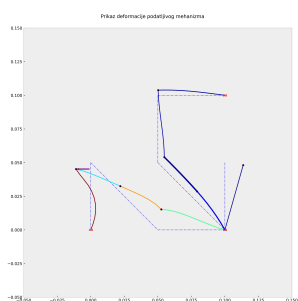
1. Lokalna optimizacija



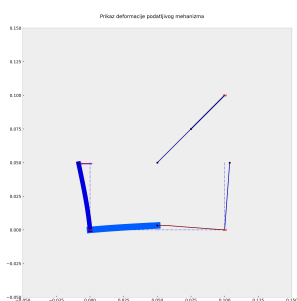
2. Lokalna optimizacija



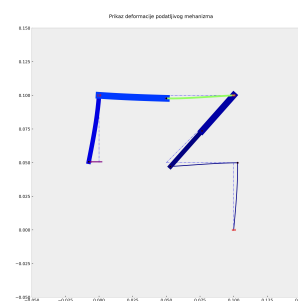
3. Lokalna optimizacija



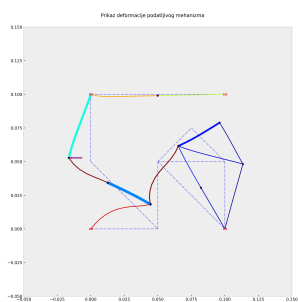
4. Lokalna optimizacija



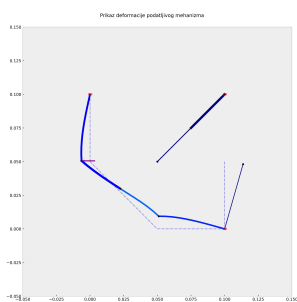
5. Lokalna optimizacija



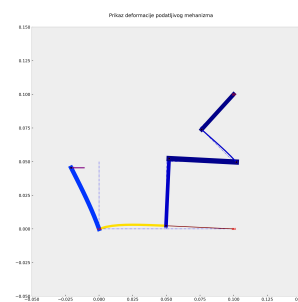
6. Lokalna optimizacija



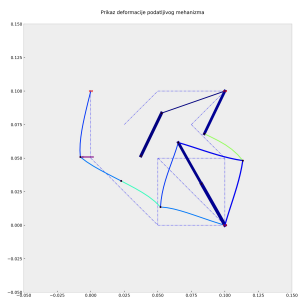
7. Lokalna optimizacija



8. Lokalna optimizacija



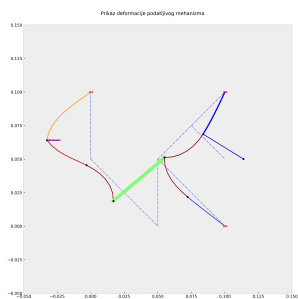
9. Lokalna optimizacija



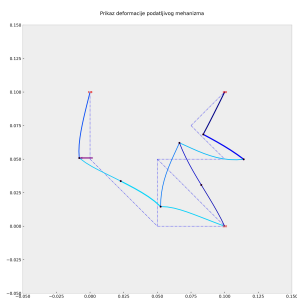
10. Lokalna optimizacija

Slika 6.14 Prikaz rezultata drugog stupnja optimizacije. Vidljivo je kako već sada par lokalnih optimizacija kreće u relativno istu minimalnu vrijednost. Ponovo tri lokalne optimizacije dobivaju “nemoguće” rješenje, dok dvije valjane optimizacije ulaze u kritično područje.

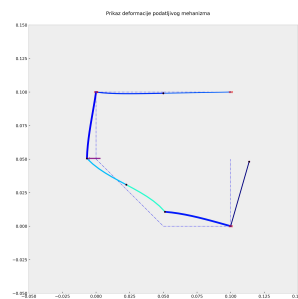
Poglavlje 6. Rezultati



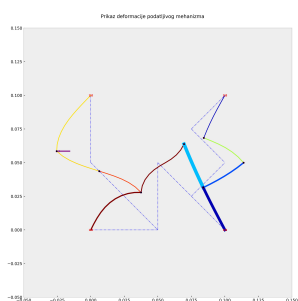
1. Lokalna optimizacija



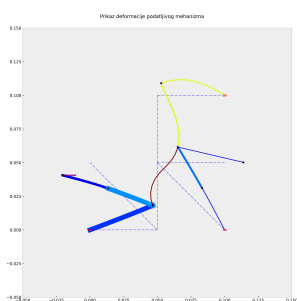
2. Lokalna optimizacija



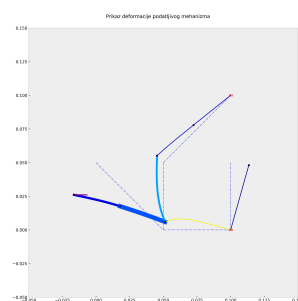
3. Lokalna optimizacija



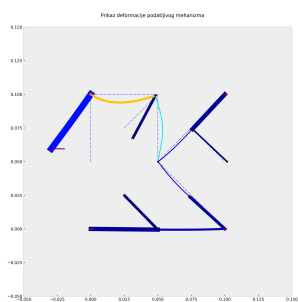
4. Lokalna optimizacija



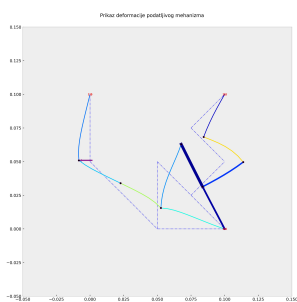
5. Lokalna optimizacija



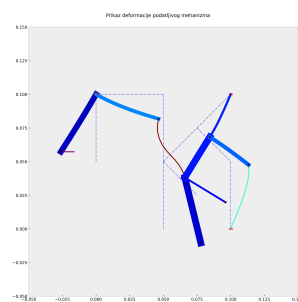
6. Lokalna optimizacija



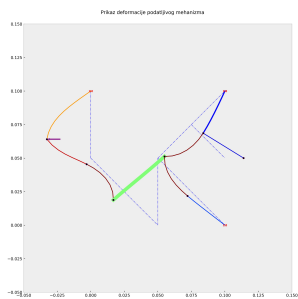
7. Lokalna optimizacija



8. Lokalna optimizacija



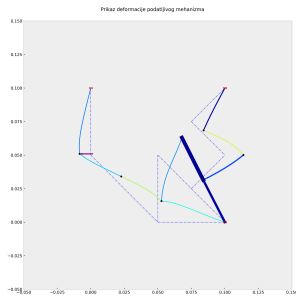
9. Lokalna optimizacija



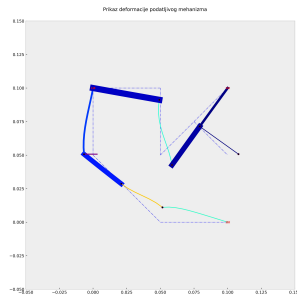
10. Lokalna optimizacija

Slika 6.15 Prikaz rezultata trećeg stupnja optimizacije. Kod ovog stupnja vidimo kako više nema “nemogućih” rješenja, već su sva rješenja valjana. Također sve manje greda ulazi u kritično područje.

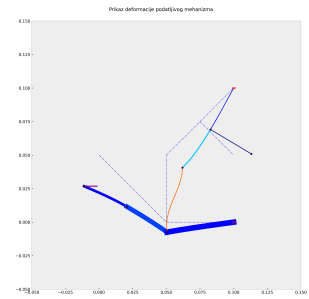
Poglavlje 6. Rezultati



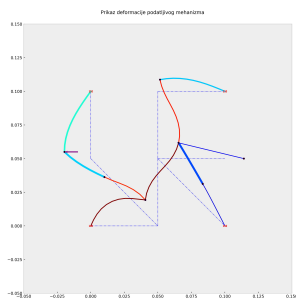
1. Lokalna optimizacija



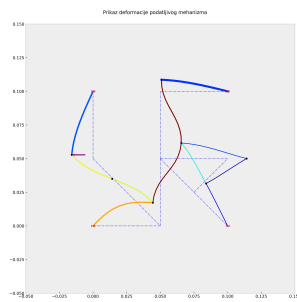
2. Lokalna optimizacija



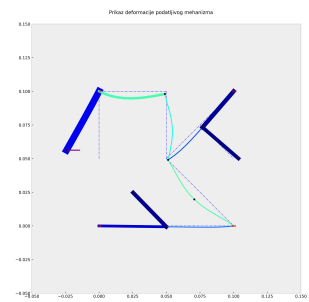
3. Lokalna optimizacija



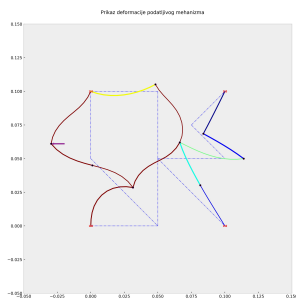
4. Lokalna optimizacija



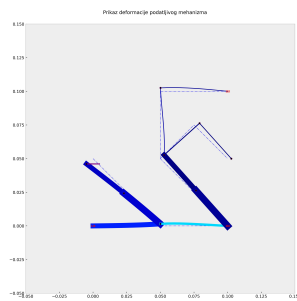
5. Lokalna optimizacija



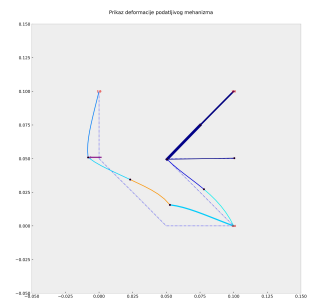
6. Lokalna optimizacija



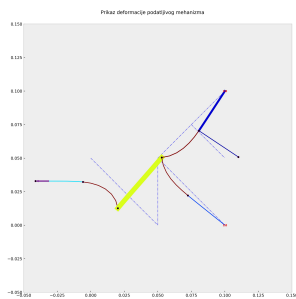
7. Lokalna optimizacija



8. Lokalna optimizacija



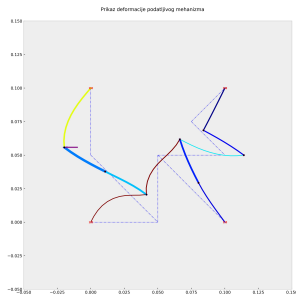
9. Lokalna optimizacija



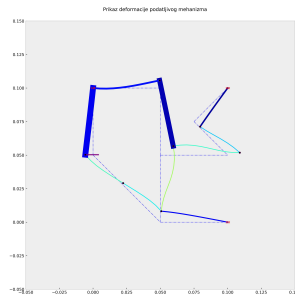
10. Lokalna optimizacija

Slika 6.16 Prikaz rezultata četvrtog stupnja optimizacije. Lokalne optimizacije sve se više kreću prema jednakom rješenju (zbog manjeg broja optimizacijskih varijabli).

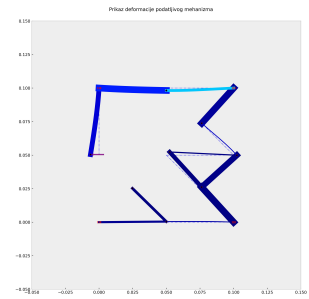
Poglavlje 6. Rezultati



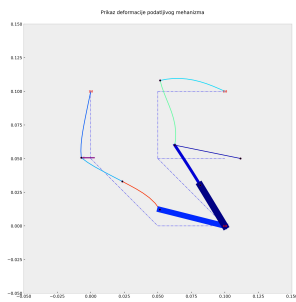
1. Lokalna optimizacija



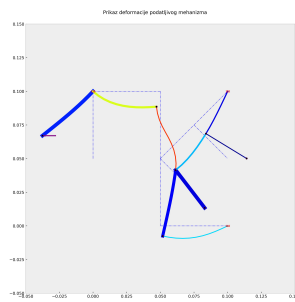
2. Lokalna optimizacija



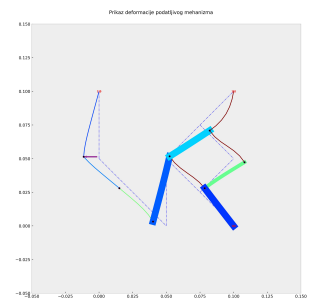
3. Lokalna optimizacija



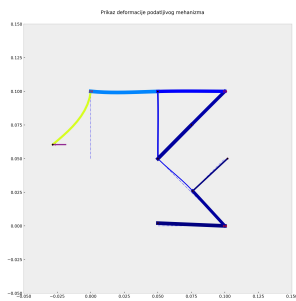
4. Lokalna optimizacija



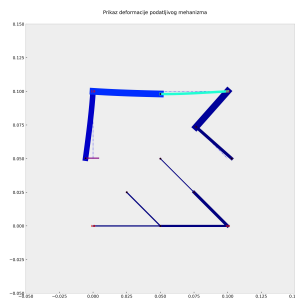
5. Lokalna optimizacija



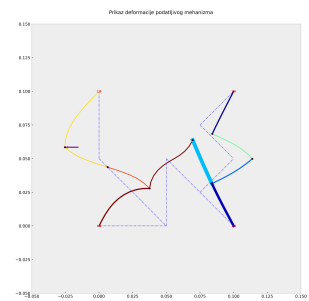
6. Lokalna optimizacija



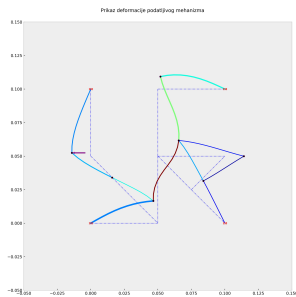
7. Lokalna optimizacija



8. Lokalna optimizacija



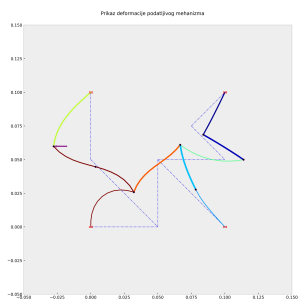
9. Lokalna optimizacija



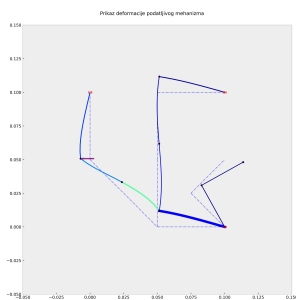
10. Lokalna optimizacija

Slika 6.17 Slike prikazuju peti stupanj algoritma. Jedna od lokalnih optimizacija daje “nemoguće” rješenje

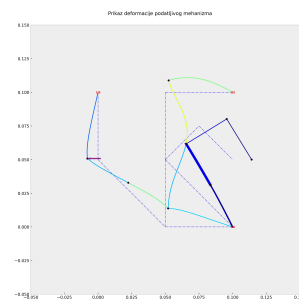
Poglavlje 6. Rezultati



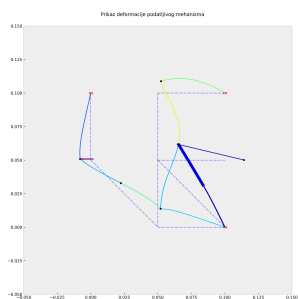
1. Lokalna optimizacija



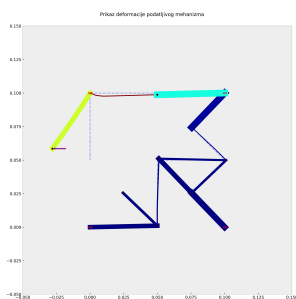
2. Lokalna optimizacija



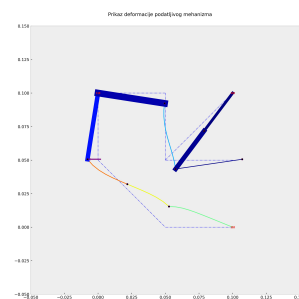
3. Lokalna optimizacija



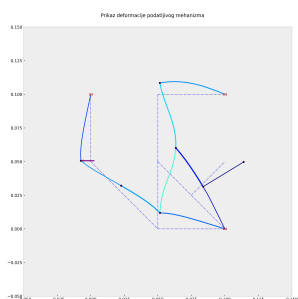
4. Lokalna optimizacija



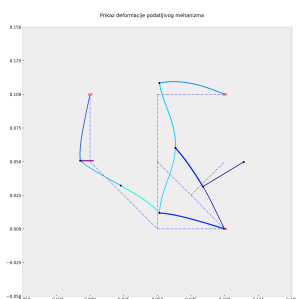
5. Lokalna optimizacija



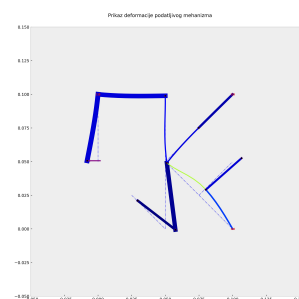
6. Lokalna optimizacija



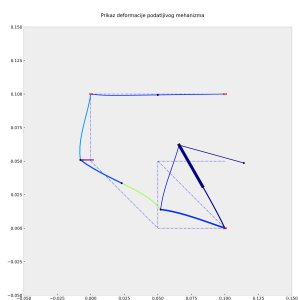
7. Lokalna optimizacija



8. Lokalna optimizacija



9. Lokalna optimizacija



10. Lokalna optimizacija

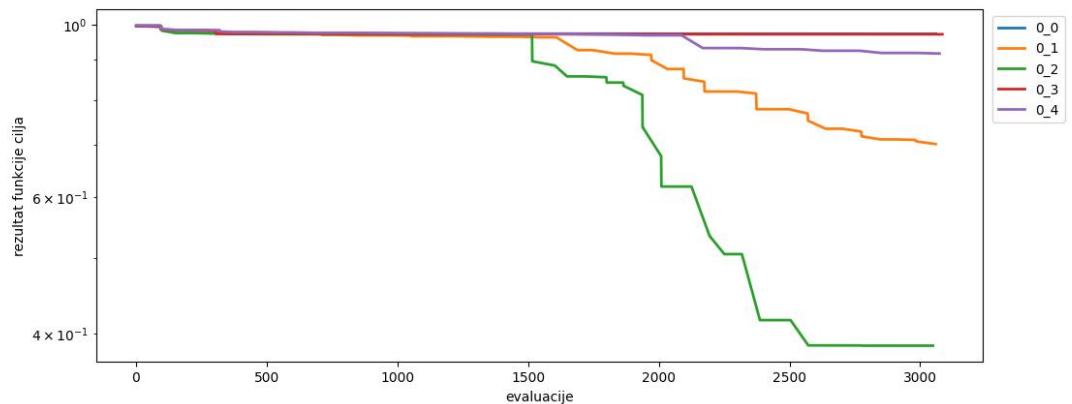
Slika 6.18 Posljednji stupanj prikazuje definitivan odabir krajnjeg dizajna podatljivog mehanizma. Kako se primjetilo kod rezultata funkcije cilja, u posljednjem stupnju algoritma 6 lokalnih optimizacija postiže vrlo slično rješenje. Kako se proračun zaustavlja, odabire se upravo taj dizajn mehanizma.

6.2 Rezultati drugog primjera

6.2.1 Iznos funkcije cilja kod lokalnih optimizacija svakog stupnja optimizacije

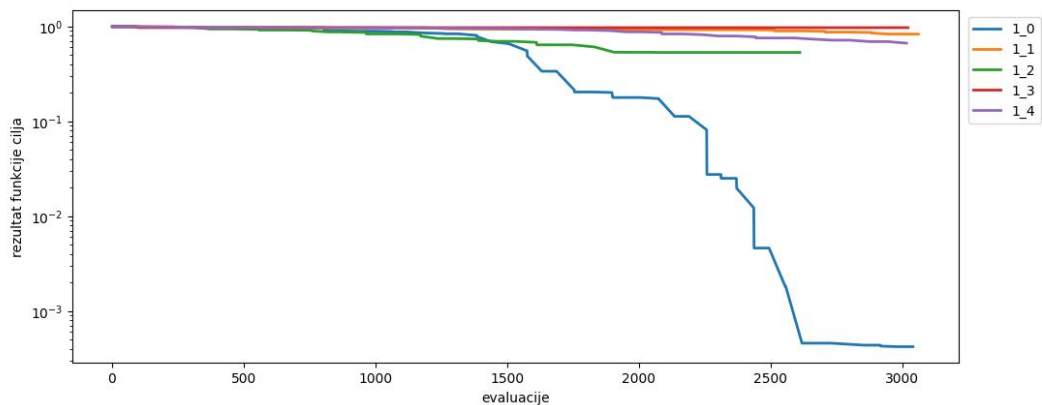
Zbog smanjenog broja varijabli može se primjetiti sve brža konvergencija rezultata optimizacija kao i kod prvog primjera.

Neke od lokalnih optimizacija nisu konvergirale. Zbog nasumične prirode postavljanja početnih vektora optimizacijskih varijabli postavljenog problema, postoji mogućnost da početne vrijednosti kod nekih lokalnih optimizacija prouzroče konvergenciju u lokalni minimum čija vrijednost uvelike odstupa od optimalnog rješenja. U višim stupnjevima optimizacije primjećuje se manja učestalost ovakvih grešaka zbog manjeg broja optimizacijskih varijabli.

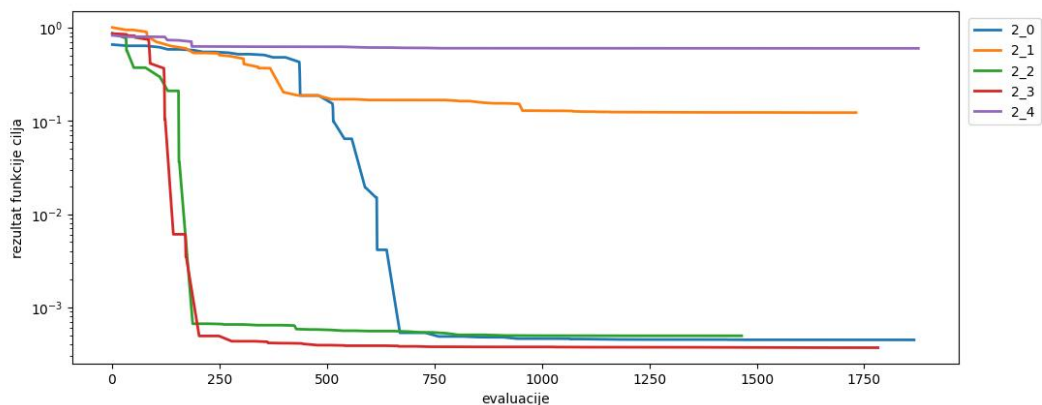


Slika 6.19 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija prvog stupnja optimizacije. Dvije lokalne optimizacije su konvergirale, ali s vrlo visokim iznosom funkcije cilja. Problem se sastoji od 47 optimizacijskih varijabli.

Poglavlje 6. Rezultati

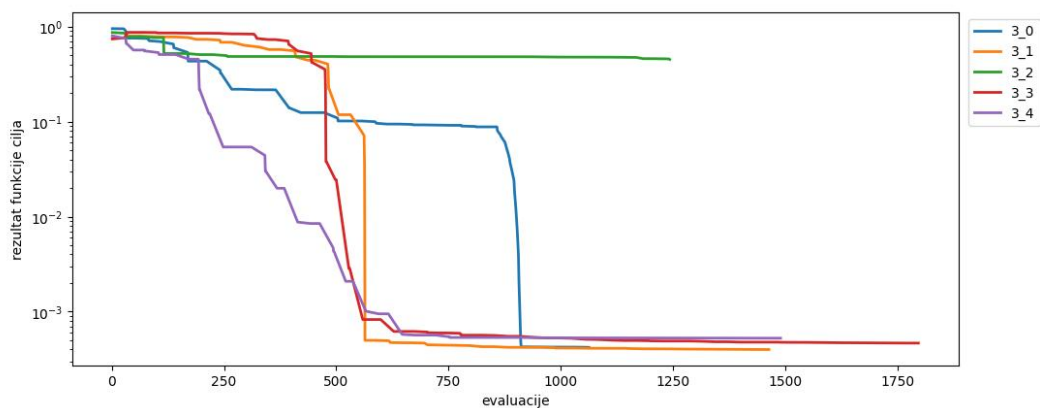


Slika 6.20 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija drugog stupnja algoritma. Kako se u drugom stupnju broj optimizacijskih varijabli smanjio samo za jednu vrijednost (46 optimizacijskih varijabli) kao i kod prvog stupnja vidimo da niti jedna optimizacija nije konvergirala.

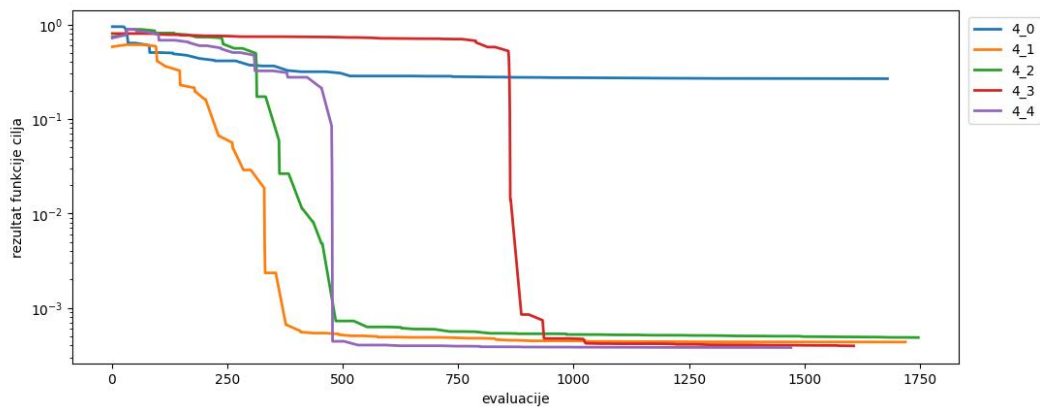


Slika 6.21 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija trećeg stupnja algoritma. U ovome stupnju koristi se 16 optimizacijskih varijabli. Nagli pad broja optimizacijskih varijabli razlog je nagle konvergencije rezultata za razliku od prijašnjih optimizacija.

Poglavlje 6. Rezultati

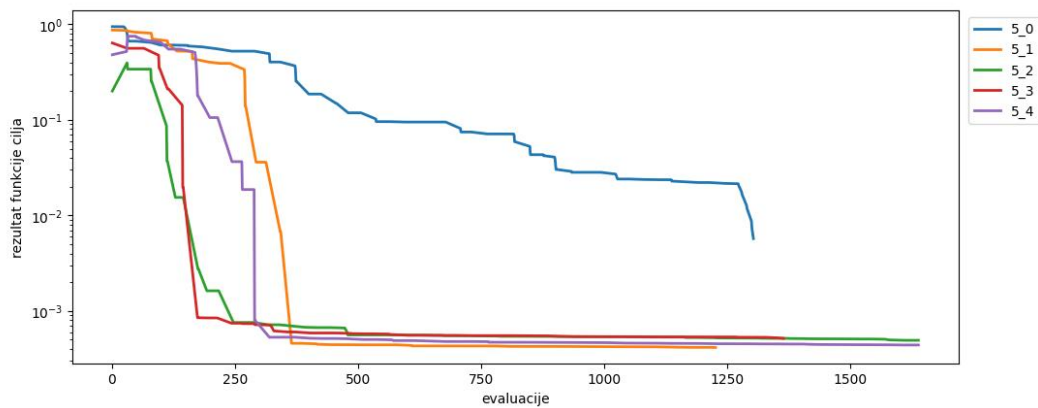


Slika 6.22 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija četvrtog stupnja optimizacije.

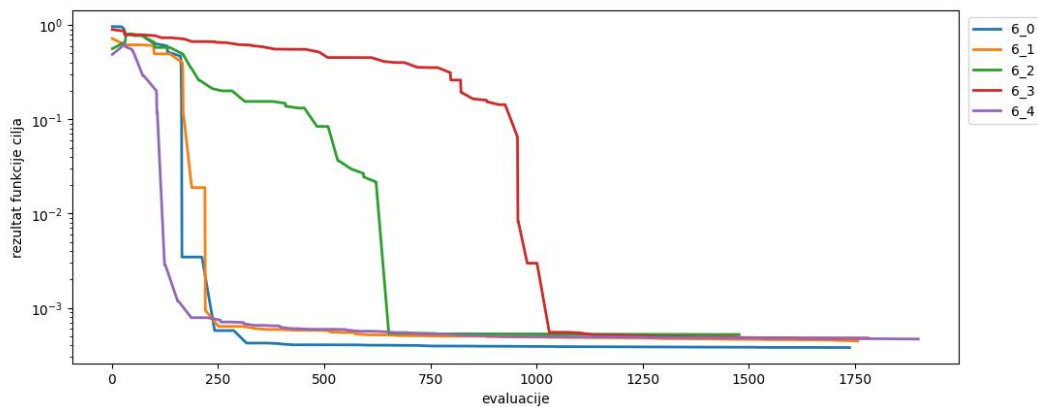


Slika 6.23 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija petog stupnja optimizacije.

Poglavlje 6. Rezultati

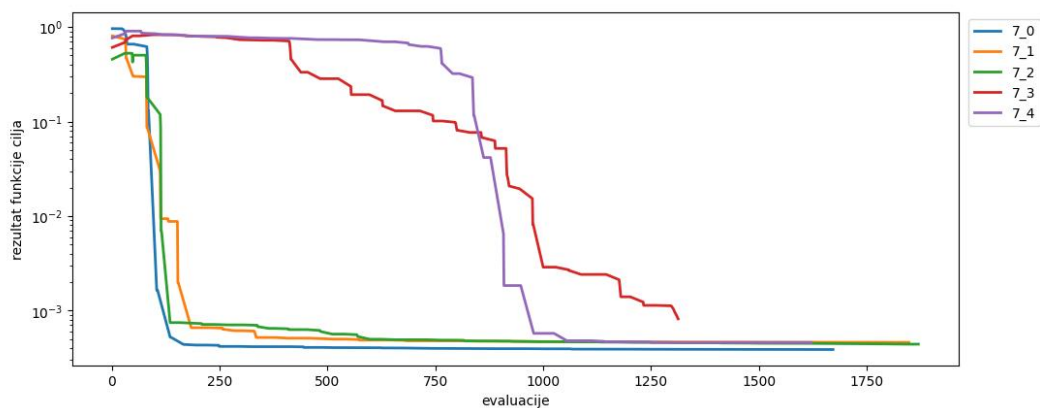


Slika 6.24 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija šestog stupnja optimizacije.

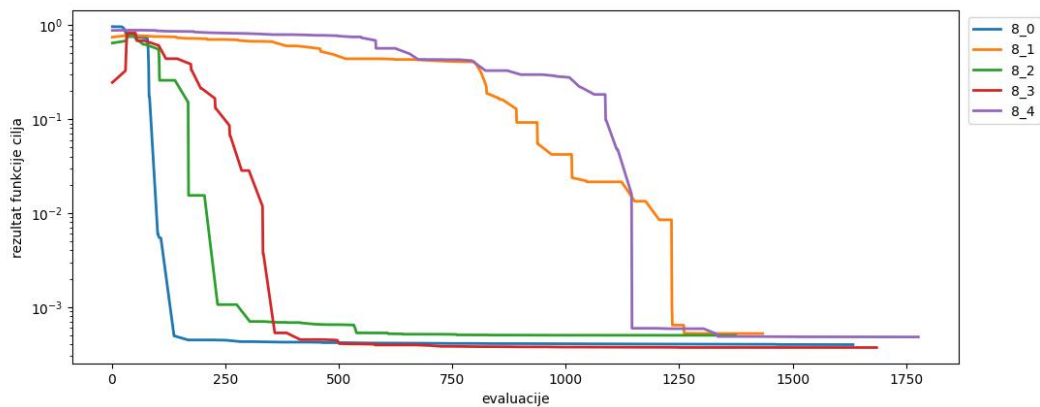


Slika 6.25 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija sedmog stupnja optimizacije.

Poglavlje 6. Rezultati

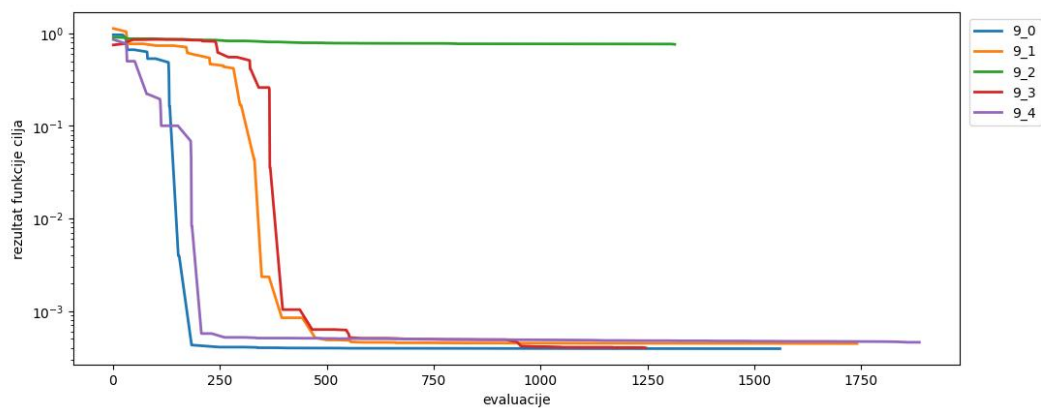


Slika 6.26 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija osmog stupnja optimizacije.



Slika 6.27 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija devetog stupnja optimizacije.

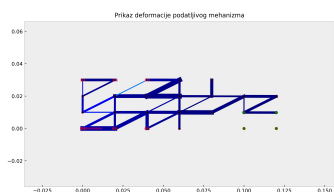
Poglavlje 6. Rezultati



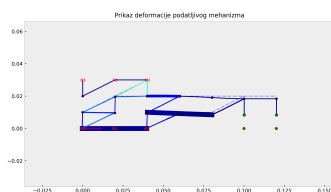
Slika 6.28 Graf prikazuje iznos funkcije cilja tijekom lokalnih optimizacija desetog stupnja optimizacije. Uvjerljivo su sve lokalne optimizacije ovog stupnja konvergirale. Sve lokalne optimizacije osim jedne postigle su jednaki rezultat. Algoritam se zaustavlja.

6.2.2 Rješenja optimizacija unutar svakog stupnja algoritma

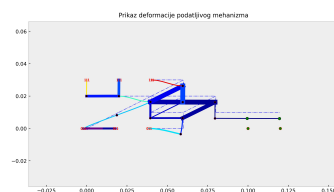
Prikazani su rezultati strukturalne analize rezultata lokalnih optimizacija svakog stupnja algoritma.



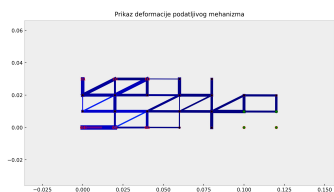
1. Lokalna optimizacija



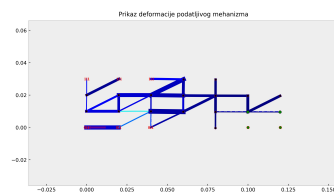
2. Lokalna optimizacija



3. Lokalna optimizacija



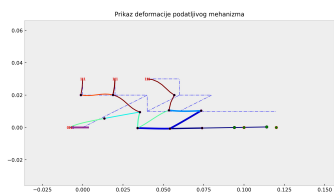
4. Lokalna optimizacija



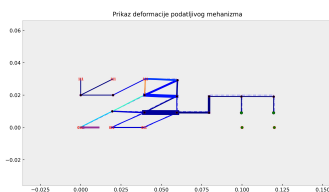
5. Lokalna optimizacija

Slika 6.29 Slike prikazuju rješenja prvog stupnja algoritma. Kako se moglo pretpostaviti na temelju grafova iznosa funkcije cilja, samo je jedna lokalna optimizacija uspjela postići zadane pomake iako uz moguće prekoračenje debljina greda u kritično područje.

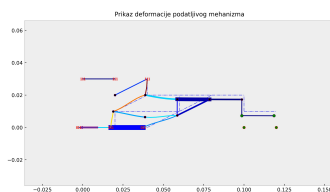
Poglavlje 6. Rezultati



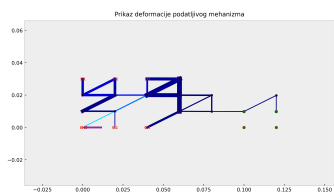
1. Lokalna optimizacija



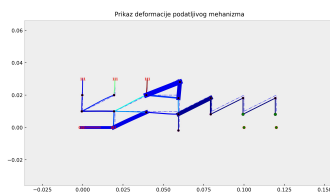
2. Lokalna optimizacija



3. Lokalna optimizacija

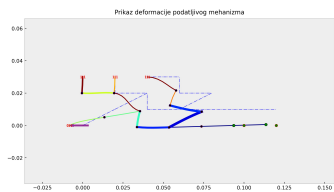


4. Lokalna optimizacija

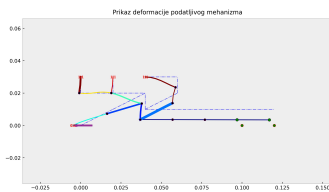


5. Lokalna optimizacija

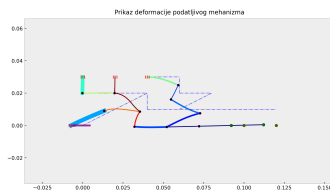
Slika 6.30 Kao i kod prvog stupnja algoritma, lokalne optimizacije sve konvergiraju u drugačija rješenja. Većina lokalnih optimizacija nije postigla optimalno traženo rješenje.



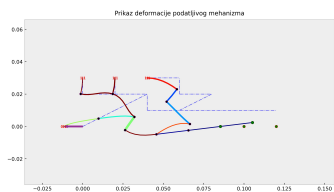
1. Lokalna optimizacija



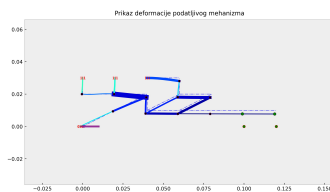
2. Lokalna optimizacija



3. Lokalna optimizacija



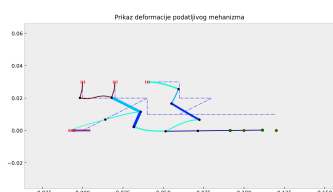
4. Lokalna optimizacija



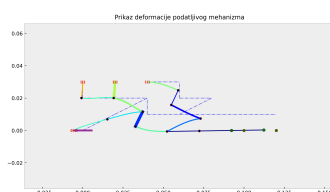
5. Lokalna optimizacija

Slika 6.31 Prikazan je treći stupanj algoritma. Nakon velikog smanjenja broja optimizacijskih varijabli većina rješenja prikazuje zadovoljene uvjete optimizacije. Kod većine rješenja doduše grede ulaze u kritično područje.

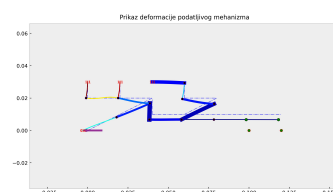
Poglavlje 6. Rezultati



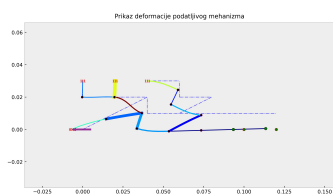
1. Lokalna optimizacija



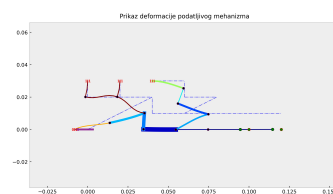
2. Lokalna optimizacija



3. Lokalna optimizacija



4. Lokalna optimizacija



5. Lokalna optimizacija

Slika 6.32 Prikazan je četvrti stupanj algoritma. U ovome stupnju nije moguće maknuti više optimizacijskih varijabli (greda). Micanjem bilo koje grede mičemo poveznicu između točaka s zadanim rubnim uvjetima potrebnim za rješavanje optimizacijskog problema. Svaki idući stupanj algoritma koristi jednaki broj optimizacijskih varijabli, tako da nema potrebe prikazivati ostale rezultate. Zaključuje se da je optimizator postigao zadovoljavajuće rješenje.

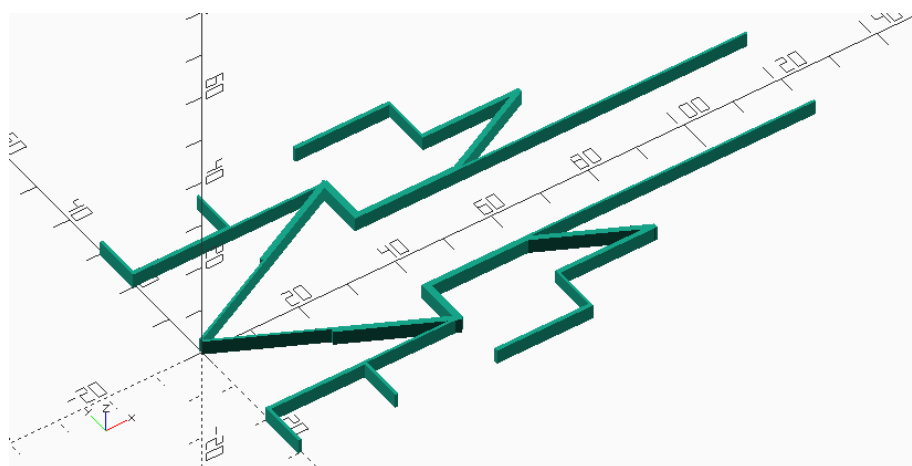
6.3 Odabir dizajna mehanizma i izrada prototipa mehanizama

Na temelju prethodno opisanih rezultata moguće je odrediti završni dizajn podatljivih mehanizma iz postavljenih primjera. Korištenjem mogućnosti brzog prototipiranja dizajnirani mehanizmi mogu se lako ispisati pomoću 3D pisača. Dobiveni prototipi mehanizma pokazuju ispravnost proračuna te ispravnost modela dizajniranog mehanizma.

6.3.1 3D ispis podatljivih mehanizama

Temeljem zadanih geometrijskih karakteristika mehanizama te dobivenih debljina greda rešetke nakon optimizacije podatljivog mehanizma, automatiziranim postupkom postavlja se računalni 3D model mehanizma.

Postupak izrade mehanizma parametriziran je i automatiziran korištenjem *Python* programskog jezika te *Python* biblioteke *openpyscad* kojim se dobiveni rezultati i zadana geometrija prevode u *OpenSCAD* kod. Korištenjem programskog paketa *OpenSCAD* sada je moguće dobiti 3D model dobivenog mehanizma.



Slika 6.33 Prikazuje se dobiveni 3D računalni model jednog od zadanih mehanizama korištenjem *OpenSCAD*ovog *post-processinga*.

Poglavlje 6. Rezultati

Dobiveni 3D računalni model mehanizma prikaz je matematičkog modela mehanizma i ne uključuje uklještenja i oslonce. Zbog toga je često potrebno dobiveni mehanizam dodatno promijeniti nadodajući potrebne ne-strukturalne elemente (oslonce, uklještenja, jednostavnih hvatišta sile, itd.). Postavljanje potrebnih elemenata moguće je prije samog ispisa korištenjem programa za 3D modeliranje (u ovom slučaju se koristi programski paket *FreeCAD*) ili nakon 3D printanja korištenjem dodatnih alata. Sve promijene dobivenog podatljivog mehanizma opisat će se naknadno.

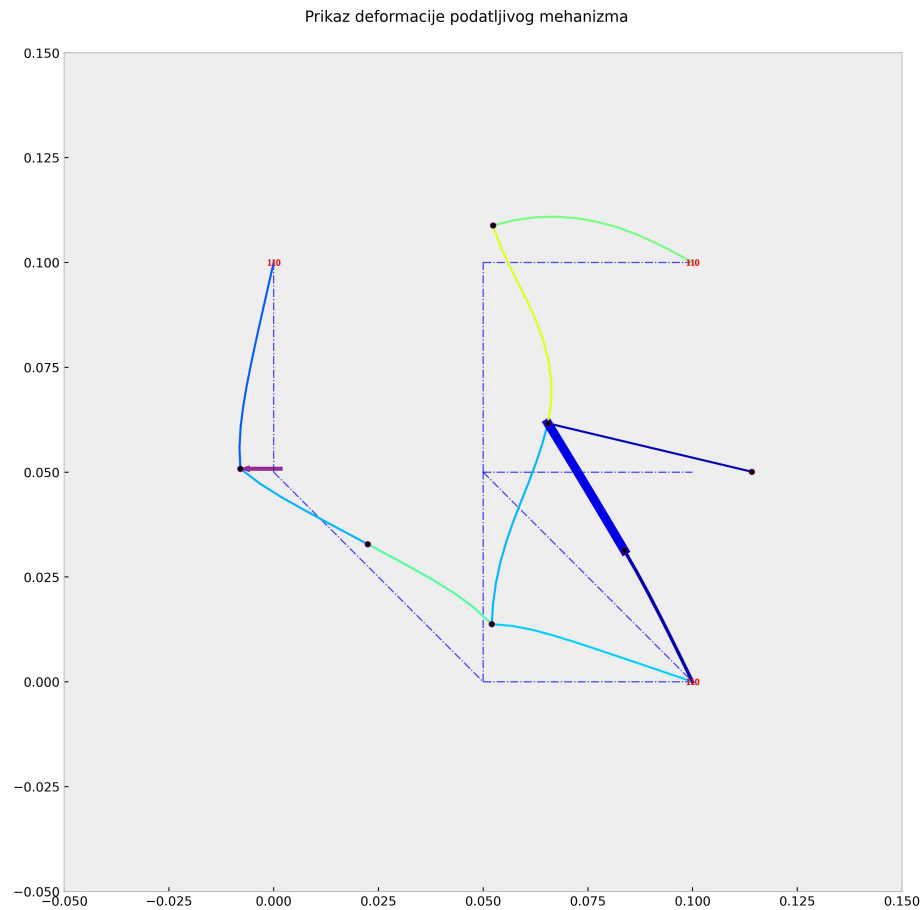
Potrebno je uzeti u obzir kako 3D ispis nikako nije savršena metoda izrade podatljivih mehanizama, već se samo koristi kod izrade prototipa mehanizma koji se kasnije može izraditi trajnijom i kvalitetnijom metodom, primjerice lijevanjem plastike.

6.3.2 Primjer 1. - Invertor pomaka

Kako u posljednjem stupnju optimizacije ponovo imamo različita rješenja zadanog problema, odabire se rješenje oje se pojavljuje u više od 50% primjera. Također zbog prirode optimizacije s nasumičnim početnim vektorom, potrebno je odabrati najbolje rješenje na temelju iznosa funkcije cilja. Vidljivo je iz grafa 6.6 kako je lokalna optimizacija 5_3 optimizacija s najboljom vrijednosti funkcije cilja te se taj mehanizam odabire kao rješenje zadanog problema.

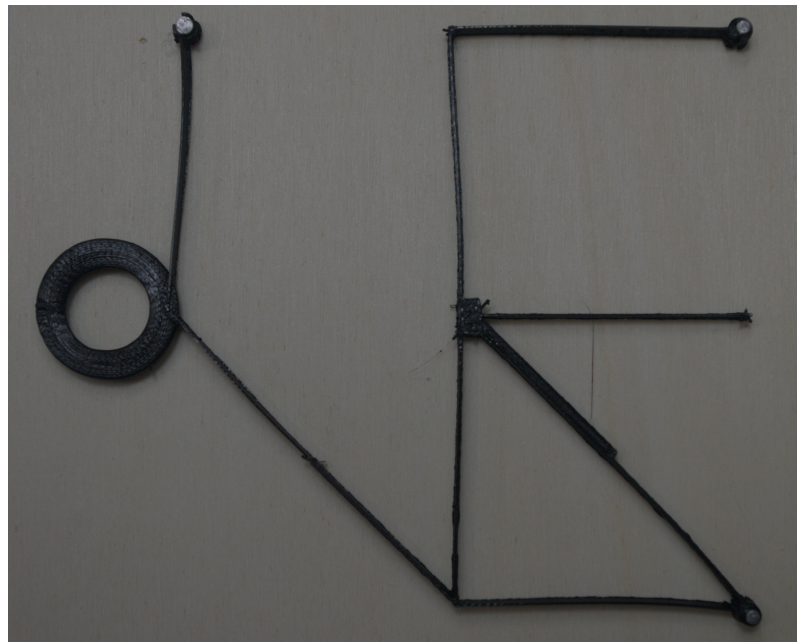
Kod ovog je rješenja na dobiveni računalni 3D model potrebno nadodati jednostavan način prihvata hvatišta sile, te riješiti problem definiranja stvarnih oslonaca koji se ne mogu isprintati na 3D pisaču.

Poglavlje 6. Rezultati



Slika 6.34 Graf prikazuje odabrani dizajn mehanizma na temelju iznosa funkcije cilja. Prikazuje se rezultat 4. lokalne optimizacije 5. stupnja algoritma. Može se uočiti kako niti jedna greda ne ulazi u kritično područje naprezanja.

Poglavlje 6. Rezultati



Slika 6.35 Slika prikazuje nedeformirani prototip odabranog mehanizma izrađen na 3D pisaču. Zbog lakšeg nanošenja opterećenja (sile) dodaje se prsten na hvatištu sile. Također, problem ispisa oslonaca riješen je postavljanjem šupljih valjaka na čvorove mehanizma na kojima se nalaze oslonci. Cijeli mehanizam se pomoću čavlića zabija u drvo, što omogućuje postavljanje oslonca na tražene čvorove.



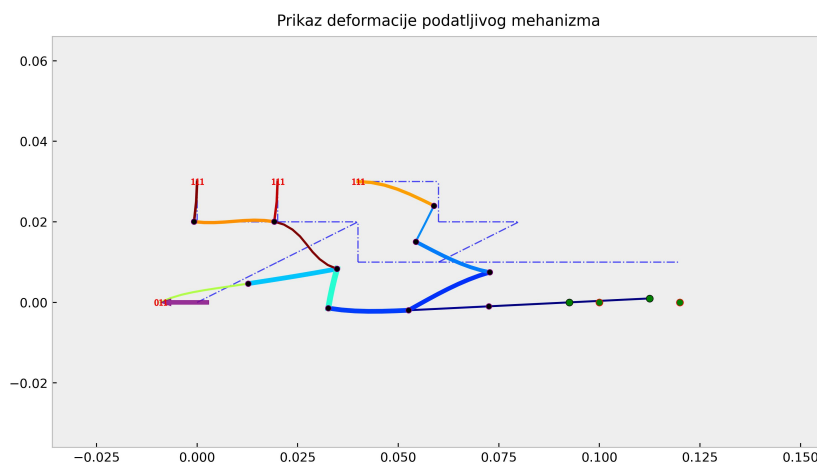
Slika 6.36 Slika prikazuje deformirani prototip odabranog mehanizma izrađen na 3D pisaču. Uspoređujući dobiveni rezultat dizajna podatljivog mehanizma i isprintani mehanizam vidljivo je kako su fizikalni aproksimirani model i strukturalni proračun pravilno predvidjeli ponašanje fizikalnog modela.

6.3.3 Primjer 2. - Hvataljka

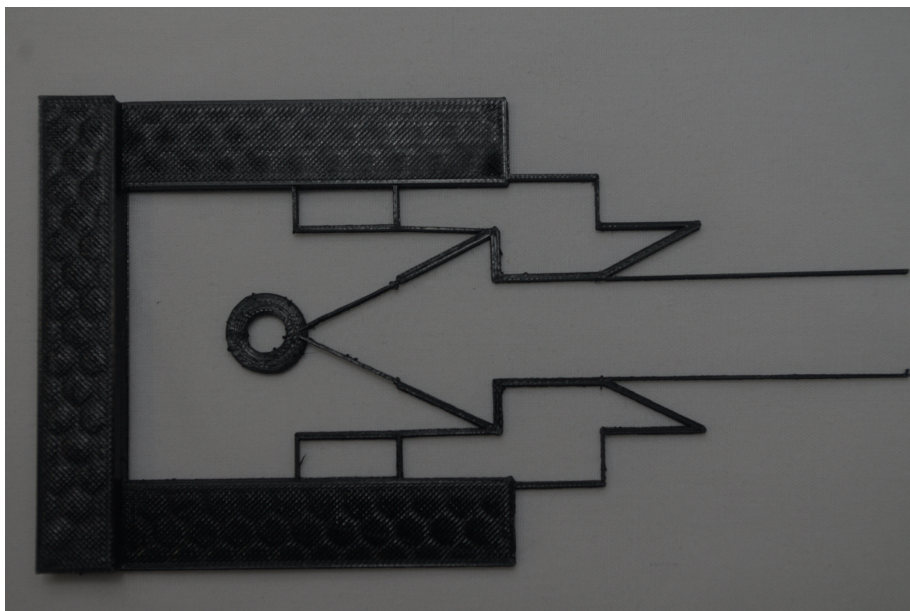
Za razliku od prvog primjera, kod drugog se primjera unutar manje stupnjeva algoritma dobiva zadovoljavajuće rješenje. Također kako sve lokalne optimizacije u posljednjem stupnju algoritma ukazuju na jednako strukturalno rješenje. To se rješenje odabire kao konačno rješenje dizajna podatljivog mehanizma. Svaka optimizacija doduše dobiva drugačiji raspored debljina greda, ali je geometrijska struktura mehanizama jednaka u svim lokalnim optimizacijama posljednjeg stupnja algoritma.

Za razliku od prethodnog primjera, optimizacija dizajna podatljivog mehanizma u ovome slučaju optimizira i dobiva rezultate deformacije polovice mehanizma. Kako je proračun pokrenut postavljanjem uvjeta simetrije kod izrade prototipa mehanizma potrebno je nadodati drugu polovicu mehanizma zrcaljenjem izračunate polovice preko osi simetrije definirane kod početne definicije bazne rešetke ovog primjera.

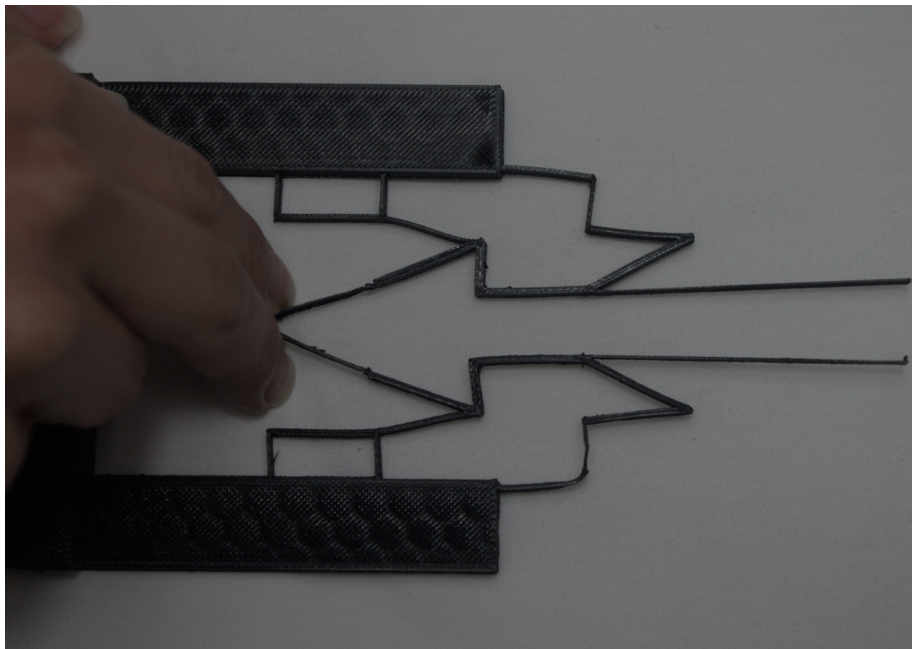
Kako se u ovom primjeru postavljaju uklještenja, moguće je definirati sve dodatne ne-strukturalne dijelove mehanizma prije samog ispisa mehanizma na 3D pisaču. Tako se dodatno postavlja kruti okvir na kojem se nalaze uklještenja, kao i torus (prsten) na hvatištu sile za jednostavnije nanošenje opterećenja (sile).



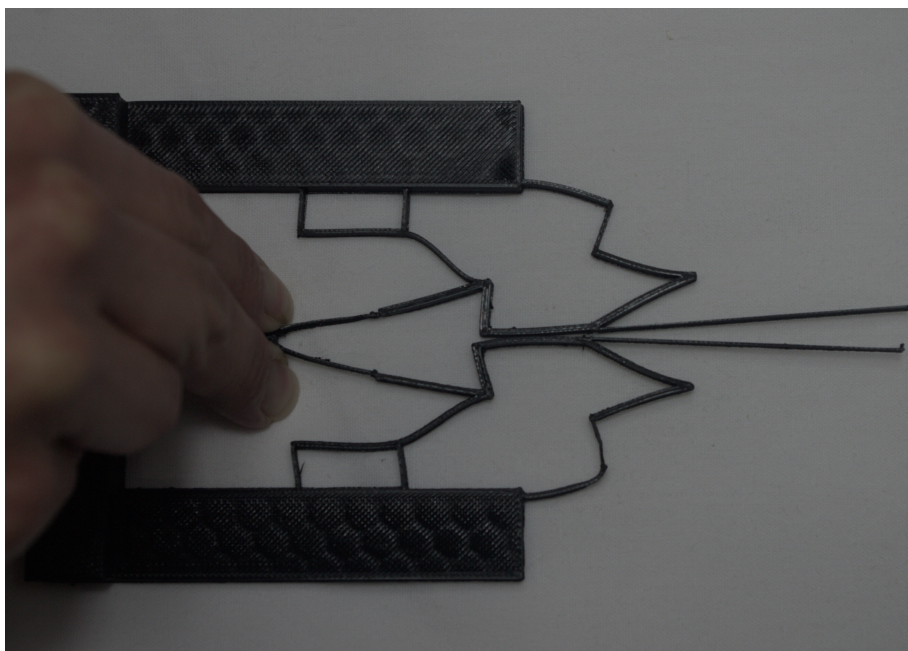
Slika 6.37 Graf prikazuje odabrani dizajn mehanizma na temelju iznosa funkcije cilja. Prikazuje se rezultat 0. lokalne optimizacije 11. stupnja algoritma.



Slika 6.38 Nedeformirani oblik prototipa mehanizma hvataljke izrađenog na 3D pisaču. Zbog lakšeg nanošenja sile, dodan je prsten povezan s hvatištem sile. Također postavlja se i kruti okvir na koji se kruto spajaju čvorovi modela kod kojih je definirano uklještenje.



Slika 6.39 Slika prikazuje deformirani prototip mehanizma opterećenog manjom silom nego što je zadana u modelu. Vidljivo je kako dobivene deformacije mehanizma prate trend deformiranja dobiven strukturalnom analizom mehanizma.



Slika 6.40 Slika prikazuje deformirani prototip mehanizma opterećen punim iznosom sile. Zbog geometrijskih nepravilnosti uvedenih 3D ispisom deformacije polovica nisu jednake, te deformacija mehanizma nije u potpunosti osno simetrična kako je bilo predviđeno kod postavljanja primjera. Također točke na desnom kraju hvataljke nisu postigle zadani pomak. Pretpostavlja se da zbog uvjeta minimizacije mase grede na krajevima hvatalje postaju vrlo tanke (minimalna moguća veličina grede) što u matematičkom modelu ne igra nikakvu ulogu, ali kod prototipa mehanizma prouzrokuje pogrešno deformirani mehanizam. Podebljanjem opisanih greda neće se promijeniti deformacija mehanizma, ali će se osigurati konstantna paralelnost greda tijekom deformacije mehanizma (kako je predviđeno matematičkim modelom).

7 Zaključak

Cilj ovog rada je postavljanje algoritma koji temeljem numeričke optimizacije daje dizajnirani optimizirani podatljivi mehanizam. Korištenjem parametriziranog modela rešetkastog podatljivog mehanizma automatizirani postupak pretvara zadani problem zajedno s početnim i rubnim uvjetima u sustav podoban za analizu i optimizaciju. Opisani optimizacijski pristup se može koristiti kod strukturalnih problema gdje se traži optimalni sustav koji se sastoji od podskupa elemenata strukture.

Spoj lokalnih *MSGs* optimizacija i algoritma za micanje “suvišnih” optimizacijskih varijabli pokazao se uspješan kod rješavanja problema definiranih u radu. Kod oba primjera vidljiva je konvergencija u optimalno rješenje. Kod drugog primjera algoritam je spustio broj varijabli na minimalni mogući broj varijabli, gdje micanjem bilo koje od varijabli postizemo prekidanje veze između zadanih oslonaca, opterećenja te čvora kod kojeg tražimo određeni pomak. Na temelju toga može se zaključiti kako je dizajn koji je algoritam postigao optimalan, te se takav dizajn odabire kao završni dizajn podatljivog mehanizma. Za razliku od toga, u prvome primjeru moguće je primijetiti kako je potrebna daljnja intervencija za određivanje optimalnog mehanizma. Jedna od mogućnosti je svakako ponovo pokretanje optimizacijskog postupka, ali kako veći dio rješenja lokalnih optimizacija posljednjeg stupnja algoritma ukazuje na jednako rješenje možemo to rješenje uzeti kao završni dizajn podatljivog mehanizma.

Potrebno je uzeti u obzir da postupak opisan u ovom radu nije prikladan za korištenje kod svakog optimizacijskog problema, kako je jedna od bitnih karakteristika micanje određenog broja optimizacijskih varijabli između stupnjeva algoritma. Ako je kod promatranog slučaja fizikalno nemoguće izbaciti varijablu iz proračuna, a da se proračun nastavi, ovaj pristup gubi smisao, kako će se u takvom primjeru svaki stupanj sastojati od jednakog broja varijabli i nasumičnih lokalnih optimizacija čiji rezultati u tako definiranom slučaju nemaju utjecaj na krajnje rješenje.

Dobiveni prototipi mehanizma pokazatelji su točnosti strukturalne optimizacije i zadanog matematičkog modela. Kako je nemoguće izbjeći greške kod 3D

Poglavlje 7. Zaključak

ispisa, ispisani mehanizam moguće je preopteretiti i izazvati puknuće mehanizma na kritičnim spojevima pojedinih greda (koje ulaze u kritično područje) zbog nesavršenog spoja ispisanih greda. Također ispisom mehanizma dobiva se uvid u poboljšanje dizajna određenog numeričkom optimizacijom.

Iako su se u ovome radu koristile vrijednosti materijalnih i geometrijskih karakteristika vezanih uz ispis na 3D printeru, u parametriziranom postupku moguće je lako definirati drugačije karakteristike problema omogućujući tako automatizirani dizajn velikog broja podatljivih mehanizama.

Bibliografija

- [1] L. L. Howell, *Compliant Mechanisms*. Wiley-Interscience, 2001.
- [2] M. Wang, D. Ge, L. Zhang, and J. L. Herder, “Micro-scale realization of compliant mechanisms: Manufacturing processes and constituent materials—a review,” *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, sep 2021.
- [3] CALCULIX: A Three-Dimensional Structural Finite Elemente Program. , s Interneta, <http://www.calculix.de> [s Interneta; 27.04.2023].
- [4] S. Ivić, D. S., and G. L. Indago, Python 3 biblioteka za numeričke proračune. Tehnički fakultet sveučilišta u Rijeci. , s Interneta, <https://gitlab.com/sivic/indago> [s Interneta; 27.04.2023].
- [5] Stranice Centra za napredno računanje i modeliranje. Sveučilište u Rijeci. , s Interneta, <https://cnrm.uniri.hr/hr/bura/> [s Interneta; 27.04.2023].
- [6] Programski paket za parametrizirano postavljanje 3D CAD modela. , s Interneta, <https://openscad.org/> [s Interneta; 27.04.2023].
- [7] Python biblioteka za automatizirano generiranje OpenSCAD koda. , s Interneta, <https://pypi.org/project/openpyscad/> [s Interneta; 27.04.2023].
- [8] Programski paket za postavljanje 3D CAD modela. , s Interneta, <https://www.freecad.org/> [s Interneta; 27.04.2023].
- [9] Stranice Zajednice Tehničke Kulture Ludbreg. Zajednica Tehničke Kulture grada Ludbrega. , s Interneta, <https://www.ztkl.hr/> [s Interneta; 27.04.2023].
- [10] Programski paket za automatizirano postavljanje g-koda FlashForge 3D pisača. , s Interneta, <https://www.flashforge.com/product-detail/40> [s Interneta; 27.04.2023].

Bibliografija

- [11] J. Bradfield, M. Lombard, J. Reynard, and S. Wurz, "Further evidence for bow hunting and its implications more than 60 000 years ago: Results of a use-trace analysis of the bone point from klasies river main site, south africa," *Quaternary Science Reviews*, vol. 236, p. 106295, may 2020.
- [12] J. Brnić and G. Turkalj, *Nauka o čvrstoći II*. ZIGO, 2006.
- [13] J. Sorić, *Uvod u numeričke metode u strojarstvu*. Fakultet strojarstva i brodogradnje, 2009.
- [14] —, *Metoda konačnih elemenata*. Golden marketing - Tehnička knjiga, 2004.
- [15] N. Avgoustinov, *Modelling in mechanical engineering and mechatronics*. Springer.
- [16] M. Katalenić. Program korišten kod izrade diplomskog rada. , s Interneta, <https://github.com/mkatalenic/compliant-mechanism-optimal-design>
- [17] matplotlib, Python 3 biblioteka za vizualizaciju podataka. , s Interneta, <https://matplotlib.org/> [s Interneta; 27.04.2023].

Sažetak

Ovaj rad obrađuje temu dizajna podatljivih mehanizama pomoću numeričke optimizacije. Model podatljivog mehanizma se parametrizira pomoću *Python* programskog jezika, te se strukturalna analiza pomoću *Metode konačnih elemenata* provodi korištenjem programskog paketa *Calculix*. Vršiti se strukturalna optimizacija podatljivog mehanizma korištenjem modificirane težinske sume rješenja *Multi-Scale Grid Search* lokalnih optimizacija. Iterativnim se postupkom smanjuje broj optimizacijskih varijabli kako bi se odredio krajnji dizajn podatljivog mehanizma. Prototip krajnjeg dizajna podatljivog mehanizma ispisuje se pomoću 3D pisaača u svrhu analize dizajna.

Ključne riječi — Podatljivi mehanizam, strukturalna optimizacija, *Calculix*, MKE, Multi-Scale Grid Search

Abstract

This thesis addresses the design of compliant mechanisms using numerical optimization methods. A mathematical model of the compliant mechanism is parameterized using *Python* programming language, and structural analysis using the *Finite element method* is carried out using *Calculix*. Structural optimization of the mechanism is carried out using a modified version of the weighted sum of the results of a number of *Multi-Scale Grid Search* local optimizations. An iterative process reduces the number of optimization variables until the desired design of the compliant mechanism is reached. The final design of the mechanism is prototyped using a 3D printer to further analyze its design.

Keywords — Compliant mechanism, structural optimization, *Calculix*, FEM, Multi-Scale Grid Search