

Prototipna optimizacija uz pomoć CAD alata otvorenog koda

Pavešić, Danijel

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:483680>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-20**



Repository / Repozitorij:

[University North Digital Repository](#)

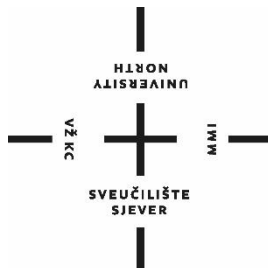


zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE SJEVER
SVEUČILIŠNI CENTAR VARAŽDIN



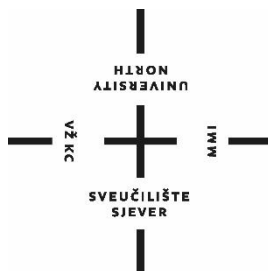
DIPLOMSKI RAD br. 117/STR/2024

PROTOTIPNA OPTIMIZACIJA UZ POMOĆ
CAD ALATA OTVORENOG KODA

Danijel Pavešić, 4187/336

Varaždin, rujan 2024. godine

SVEUČILIŠTE SJEVER
SVEUČILIŠNI CENTAR VARAŽDIN
Diplomski sveučilišni studij Strojtarstvo



DIPLOMSKI RAD br. 117/STR/2024

PROTOTIPNA OPTIMIZACIJA UZ POMOĆ
CAD ALATA OTVORENOG KODA

Student:

Danijel Pavešić, 4187/336

Mentor:

izv.prof.dr.sc. Sven Maričić

Varaždin, rujan 2024. godine

Prijava diplomskog rada

Definiranje teme diplomskog rada i povjerenstva

ODJEL

STUDIJ

PRISTUPNIK

MATIČNI BROJ

DATUM

KOLEGIJ

NASLOV RADA

NASLOV RADA NA
ENGL. JEZIKU

MENTOR

ZVANJE

ČLANOVI POVJERENSTVA

1.

2.

3.

4.

5.

VŽKC

MMI

Zadatak diplomskog rada

BROJ

OPIS

ZADATAK URUČEN

POTPIS MENTORA

Predgovor

Prije svega želio bih se zahvaliti profesoru Svenu Maričiću na trudu, vremenu i volji koje je uložio za pomoć pri izradi diplomskog rada. Također bih se želio zahvaliti svim profesorima, asistentima, predavačima i kolegama koji su prenijeli svoje znanje na mene tokom preddiplomskog te diplomskog studija.

Ovaj diplomski rad predstavlja ukupno znanje stečeno kroz svo školovanje pa sam iz tog razloga izabrao ovo izazovno područje.

Nakraju bih se posebno zahvalio svojoj obitelji koji su mi omogućili studij i bili podrška kroz cijelo školovanje.

Sažetak

Ovaj diplomski rad istražuje primjenu FreeCAD-a, CAD alata otvorenog koda, za optimizaciju industrijskih prototipa poput laboratorijskog stola i pokretne trake. Cilj rada bio je prikazati kako se putem makronaredbi i Python skriptiranja mogu automatizirati procesi dizajniranja i prilagodbe modela. FreeCAD je korišten za izradu 3D prototipa laboratorijskog stola i pokretne trake, dok su Python skripte omogućile optimizaciju dimenzija i funkcionalnosti.

Optimizacija laboratorijskog stola uključivala je prilagodbu dimenzija gornje ploče i nogu, dok je optimizacija pokretne trake bila složenija, zahtijevajući prilagodbu dimenzija U profila, izradu rupa za valjke te određivanje njihovog broja i rasporeda. Rezultati pokazuju da kombinacija FreeCAD-a i Python skripti pruža visoku razinu prilagodljivosti i učinkovitosti u dizajnu, omogućujući brzu prilagodbu modela ovisno o specifičnim zahtjevima korisnika.

Ključne riječi: FreeCAD, prototipiranje, Python, makronaredbe, laboratorijski stol, pokretna traka

Abstract

This thesis explores the application of FreeCAD, an open-source CAD tool, for optimizing industrial prototypes such as tables and conveyor belts. The aim of the work was to demonstrate how macro commands and Python scripting can automate the processes of designing and customizing models. FreeCAD was used to create 3D models of the table and conveyor belt, while Python scripts enabled the optimization of dimensions and functionality.

The optimization of the table involved adjusting the dimensions of the top surface and legs, while the optimization of the conveyor belt was more complex, requiring adjustments to the dimensions of U profiles, the creation of holes for rollers, and determining their number and arrangement. The results show that the combination of FreeCAD and Python scripts provides a high level of flexibility and efficiency in design, allowing for rapid model adjustments based on specific user requirements.

Keywords: FreeCAD, prototyping, Python, macros, laboratory table, conveyor belt.

Popis korištenih kratica

CAD	engl. <i>Computer-Aided Design</i> Računalno potpomognuto projektiranje
3D	3 dimenzije
FEM	engl. <i>Finite Element Method</i> Metoda konačnih elemenata
STEP	engl. <i>Standard for the Exchange of Product Data</i> Standard za razmjenu podataka o proizvodu
IGES	engl. <i>Initial Graphics Exchange Specification</i> Specifikacija za razmjenu početne grafike
STL	engl. <i>Stereolithography</i> Stereolitografija
SVG	engl. <i>Scalable Vector Graphics</i> Skalabilna vektorska grafika
DXF	engl. <i>Drawing Exchange Format</i> Format za razmjenu crteža
OBJ	engl. <i>Object File Format</i> Format datoteke objekta
2D	2 dimenzije
API	engl. <i>Application Programming Interface</i> Sučelje za programiranje aplikacija

Sadržaj

1.	Uvod.....	1
2.	CAD alati otvorenog koda	2
2.1.	FreeCAD.....	2
2.1.1.	Glavne značajke FreeCAD-a.....	2
2.1.2.	Usporedba FreeCAD-a sa ostalim programima	4
2.1.3.	Primjena FreeCAD-a	5
2.1.4.	Preuzimanje programa.....	6
3.	Izrada prototipa pokretne traka	9
3.1.	Upoznavanje FreeCAD programa	9
3.2.	Korišteni alati i ograničenja.....	13
3.3.	Izrada CAD modela	16
3.4.	Python u FreeCAD-u	25
3.4.1.	Upotreba makronaredbi u FreeCAD-u za optimizaciju laboratorijskog stola i pokretne trake	27
3.4.2.	Program za optimizaciju prototipa laboratorijskog stola	29
3.4.3.	Program za optimizaciju prototipa pokretne trake	33
4.	Primjeri.....	42
4.1.	Primjeri optimiziranih prototipa laboratorijskih stolova	42
4.2.	Primjeri optimiziranih prototipa pokretnih traka.....	43
5.	Zaključak.....	46
6.	Literatura	48
7.	Popis slika	49
8.	Popis tablica	52

1. Uvod

U ovom diplomskom radu istražuje se primjena CAD (engl. Computer-Aided Design) alata otvorenog koda u procesu dizajniranja i optimizacije inženjerskih projekata. S obzirom na brzi tehnološki razvoj i rastuću popularnost otvorenog softvera, alati poput FreeCAD-a postali su pristupačni inženjerima, arhitektima i dizajnerima. Glavna svrha rada je prikazati kako se uz pomoć FreeCAD-a mogu učinkovito dizajnirati i optimizirati industrijski prototipi, konkretno laboratorijski stol i pokretna traka.

Cilj ovog rada je prikazati prednosti korištenja otvorenih CAD alata u izradi modela te kako se Python programiranjem i makronaredbama može dodatno unaprijediti funkcionalnost tih alata. Optimizacija dizajna laboratorijskog stola i pokretne trake temeljena je na prilagodbi parametara modela, čime se omogućuje brzo i jednostavno mijenjanje dimenzija prema specifičnim zahtjevima korisnika. Metodologija uključuje izradu 3D (3 dimenzije) modela, njihovu optimizaciju, putem Pythona.

Uvod u temu, svrhu i ciljeve diplomskog rada pružit će uvid u važnost ovog pristupa za suvremenu inženjersku praksu, dok će detaljan pregled korištenih alata i metoda olakšati razumijevanje praktičnih rezultata.

2. CAD alati otvorenog koda

CAD alati otvorenog koda su softverski programi za računalno potpomognuto projektiranje čiji je izvorni kod slobodno dostupan javnosti. To znači da svatko može pregledati, preuzeti, mijenjati i distribuirati izvorni kod ovih alata, što pruža velike prednosti u pogledu fleksibilnosti, prilagodljivosti i troškova.

Najpoznatiji CAD alati otvorenog koda su:

- FreeCAD
- Blender
- LibreCAD
- OpenSCAD
- SolveSpace

Za izradu diplomskog rada korišten je program FreeCAD u verziji 0.21.2.

2.1. FreeCAD

FreeCAD je 3D alat otvorenog koda, prvenstveno osmišljen za dizajn stvarnih objekata. FreeCAD karakterizira parametarsko modeliranje, podrška za više formata datoteka, integracija sa skenerima i 3D pisačima, korištenje skripti i makroa te njegovo korisničko sučelje. Sve karakteristike FreeCAD-a će biti opisane kasnije u radu. [4]

2.1.1. Glavne značajke FreeCAD-a

Modularna arhitektura

FreeCAD ima modularnu arhitekturu koja omogućava korisnicima proširivanje funkcionalnosti putem različitih modula (radnih ploča). Svaka radna ploča pruža specifične alate za određene aspekte dizajniranja i modeliranja, što uključuje: [4]

- *Mesh Workbench*: Rad s poligonalnim mrežama.
- *Raytracing Workbench*: Stvaranja slike iz 3D modela dodavanjem osvjetljenja, sjena, tekstura i drugih vizualnih efekata kako bi se dobila fotorealistična slika.
- *Spreadsheet Workbench*: Kreiranje i manipulacija tabličnim podacima.

- FEM (engl. *Finite Element Method*) *Workbench*: Alati za analizu konačnih elemenata.

Parametarsko modeliranje

Parametarsko modeliranje je ključna značajka FreeCAD-a koja omogućava korisnicima stvaranje modela kontroliranih parametrima. Ova značajka omogućava jednostavne izmjene modela kroz prilagodbu parametara, što automatski ažurira cijeli model. Na primjer, promjenom visine, širine i duljine cigle, cijeli model se prilagođava novim dimenzijama. [4]

Podrška za više formata datoteka

FreeCAD podržava različite formate datoteka, uključujući STEP (engl. *Standard for the Exchange of Product Data*), IGES (engl. *Initial Graphics Exchange Specification*), STL (engl. *Stereolithography*), SVG (engl. *Scalable Vector Graphics*), DXF (engl. *Drawing Exchange Format*), OBJ (engl. *Object File Format*) i mnoge druge, omogućavajući jednostavnu razmjenu modela s drugim CAD softverima. Ova značajka je posebno korisna za inženjere i dizajnere koji rade s različitim alatima. [4]

Integracija sa skenerima i 3D pisačima

FreeCAD omogućava integraciju s 3D skenerima i pisačima, što olakšava proces prototipiranja i proizvodnje. To omogućava korisnicima da brzo prelaze s dizajna na fizičke prototipove. [4]

Skripte i makroi

FreeCAD podržava automatizaciju zadataka putem skripti i makroa, s glavnim programskim jezikom za skriptiranje Python. Ovo omogućava korisnicima da prilagode i prošire funkcionalnost softvera, što je posebno korisno za napredne korisnike koji žele automatizirati složene zadatke. [4]

Korisničko sučelje

FreeCAD nudi prilagodljivo korisničko sučelje s mogućnošću stvaranja prilagođenih alatnih traka i radnih prostora. Korisnici mogu prilagoditi izgled i funkcionalnost sučelja prema svojim potrebama, što doprinosi efikasnosti i jednostavnosti korištenja. [4]

2.1.2. Usporedba FreeCAD-a sa ostalim programima

CAD alati otvorenog koda postaju sve popularniji zbog svoje pristupačnosti, fleksibilnosti i mogućnosti prilagodbe specifičnim potrebama korisnika. Ovi alati omogućuju korisnicima kreiranje, modificiranje, analiziranje i optimiziranje dizajna na različite načine, ovisno o njihovim funkcionalnostima i namjenama.

U tablici 1. prikazani su nam spomenuti CAD alati otvorenog koda te njihove karakteristike kao što su namjena, vrste modeliranja i njihova primjena u industriji.

Alat	FreeCAD	Blender	LibreCAD	OpenSCAD	SolveSpace
Namjena	Parametarsko 3D modeliranje, inženjerske i arhitektonske aplikacije	3D animacija, vizualizacija	2D CAD	Programski orijentiran 3D program	2D crtanje, 3D modeliranje, mehaničke simulacije
Modeliranje	Parametarsko modeliranje	Slobodno formiranje modela, napredno animiranje	2D CAD, crtanje, geometrijske konstrukcije	Modelira se opisivanjem oblika i dimenzija unutar skripti	Brzo 2D crtanje, ekstrudiranje u 3D, mehanizmi
Primjena	Inženjerski projekti, arhitektonski nacrti, simulacije	Animacija, vizualizacija, dizajn igara	Tehničko crtanje	Funkcionalni dijelovi, prilagođene komponente	Manje kompleksni projekti, izrada prototipa, simulacija mehanizama
Zahtjevi hardvera	Windows 7 i macOS 10.12 ili novije, Linux Dvojezgreni procesor Min. 4 GB RAM 2 GB za pohranu	Windows 8 i macOS 10.13 ili noviji, Linux Dvojezgreni procesor Min. 4 GB RAM 500 MB za pohranu	Windows XP i macOS 10.7 ili noviji, Linux Jednojezgreni procesor 1 GB RAM 30-50 MB za pohranu	Windows 7 i macOS 10.11 ili noviji, Linux Dvojezgreni procesor 2 GB RAM 50 MB za pohranu	Windows 7 ili noviji, macOS, Linux Jednojezgreni procesor 512 GB RAM 100 MB za pohranu

Tablica 1. Usporedba FreeCAD-a i ostalih alata [4] [5][6][7][8]

Svaki CAD alat sa svojim specifičnim karakteristikama i primjenama, pružaju različite mogućnosti za korisnike u različitim industrijama. FreeCAD se ističe kao alat za parametarsko modeliranje namijenjen inženjerskim i arhitektonskim aplikacijama, dok je Blender više orijentiran prema animaciji i vizualizaciji. LibreCAD je fokusiran na tehničko crtanje u 2D (2 dimenzije) prostoru, OpenSCAD omogućuje programski orijentirano modeliranje, a SolveSpace kombinira brzo 2D crtanje s mogućnostima 3D modeliranja i mehaničkih simulacija.

Odabir odgovarajućeg alata ovisi o specifičnim potrebama projekta i preferencijama korisnika. Svaki od ovih alata nudi jedinstvene prednosti i funkcionalnosti, te može značajno doprinijeti uspjehu dizajnerskih i inženjerskih projekata.

2.1.3. Primjena FreeCAD-a

FreeCAD je moćan alat za parametarsko 3D modeliranje koji se koristi u različitim industrijama i primjenama. Njegova dostupnost kao besplatnog alata s otvorenim kodom čini ga pristupačnim širokom spektru korisnika, od profesionalnih inženjera do hobiista. [4]

Primjena FreeCAD-a: [4]

- **Inženjerska primjena**
 - Ovaj alat omogućuje stvaranje i uređivanje 3D modela mehaničkih dijelova i sklopova. Parametarsko modeliranje daje inženjerima fleksibilnost da jednostavno prilagode dizajne specifičnim zahtjevima, što značajno povećava učinkovitost u procesu projektiranja.
- **Arhitektura**
 - U arhitekturi se koristi za izradu preciznih planova zgrada i drugih građevinskih objekata. S mogućnostima za 2D nacрте i 3D modele, softver olakšava vizualizaciju i planiranje složenih građevinskih projekata.
- **Obrazovanje**
 - Budući da je besplatan i otvorenog koda, široko je prihvaćen u obrazovnim institucijama kao učinkovit alat za podučavanje osnova CAD-a i 3D modeliranja. Omogućuje studentima stjecanje praktičnih vještina bez potrebe za ulaganjem u skupe softverske licence.

- **Hobi**
 - Fleksibilnost i prilagodljivost ovog alata privlače brojne hobiste koji ga koriste za projekte poput izrade modela, prototipiranja i sl. Njegove mogućnosti su idealne za kreativne i tehnički orijentirane projekte.
- **Podrška i zajednica**
 - Aktivna zajednica korisnika i programera pruža opsežnu podršku kroz forume, dokumentaciju i tutorijale. Redovita ažuriranja donose nove značajke i poboljšanja, što omogućava kontinuirani razvoj i prilagodbu korisničkim potrebama.

2.1.4. Preuzimanje programa

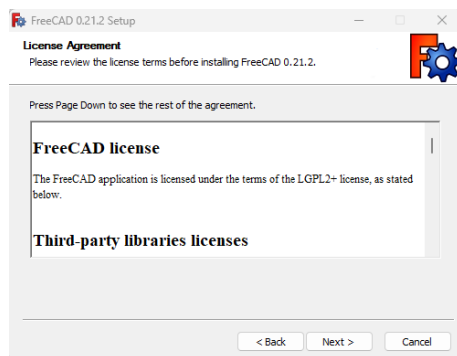
Preuzimanje programa moguće je na [6]. Ovdje možemo preuzeti program za Windows, Mac i Linux sustave.

Nakon preuzimanja instalacijskog programa, kojeg možete pronaći u preuzimanjima ili na mjestu gdje se spremaju sva preuzimanja s interneta, pokrenite ga. Pokretanjem instalacijskog programa otvara se početni prozor. Lijevim klikom miša na „Next >“ započinjemo instalaciju programa.



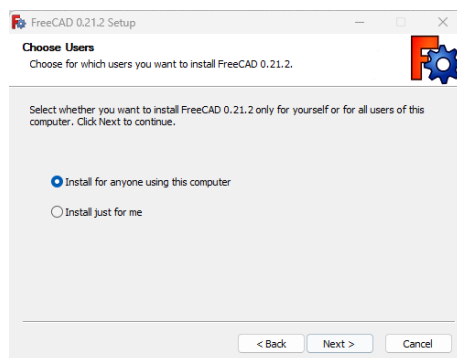
Slika 2.1 Početni prozor instalacijskog programa

U idućem koraku instalacijski program upoznaje korisnika sa uvjetima licence, kao što vidimo na slici 2.2 Kao i u prethodnom koraku, lijevim klikom miša pritisnemo „Next >“, kako bi nastavili s instalacijom programa.



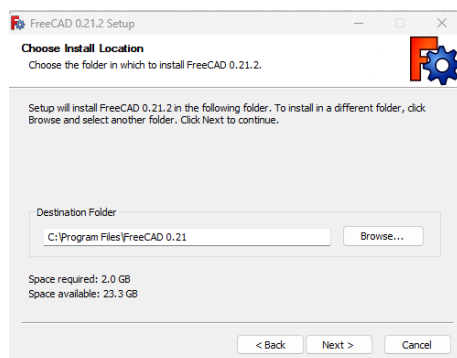
Slika 2.2 Uvjeti licence

Nakon prelaska na idući korak, korisnik odabire hoće li na tom računalu koristiti program samo on ili svi korisnici računala. Nakon što odaberemo jednu od te dvije opcije, kliknemo na „Next >“, kako bi nastavili s instalacijom.



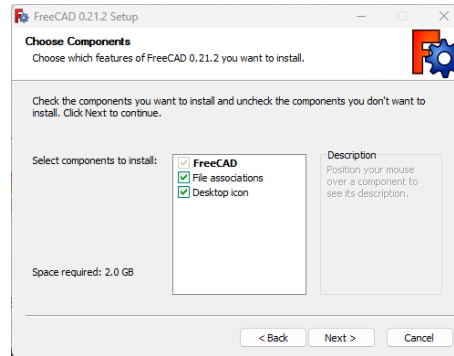
Slika 2.3 Odabir korisnika programa

U idućem koraku odabiremo mjesto na računalu gdje će se instalirati FreeCAD program. Važno je odabrati disk gdje imamo dovoljno prostora za pohranu (2.0 GB). Ako je odabrano mjesto zadovoljavajuće, kliknemo lijevim klikom miša na „Next >“.



Slika 2.4 Odabir mjesta preuzimanja na disku

Posljednji korak prije instalacije je odabir „File associations“, ako želimo da nam FreeCAD automatski otvara datoteke koje su njegovog tipa. Ako želimo imati prečac za program na radnoj površini, odabiremo „Desktop icon“. Klikom na „Next >“ počinje instalacija programa.



Slika 2.5 Odabir komponenti koje želimo preuzeti

Zadnji prozor koji se otvara nakon instalacije prikazuje opciju „Launch FreeCAD“ ako želimo da se program pokrene nakon klika na „Finish“. U donjem dijelu prozora nalazi se poveznica na novosti i savjete vezane uz FreeCAD.



Slika 2.6 Završni prozor instalacijskog programa

Nakon instalacije program možemo pokrenuti dvoklikom lijeve tipke miša na ikonu programa. Ako je kod instalacije odabran „Desktop icon“, prečac za program nalazi se na radnoj površini. Ikona programa prikazana je na slici 2.8.



Slika 2.7 Ikona FreeCAD programa

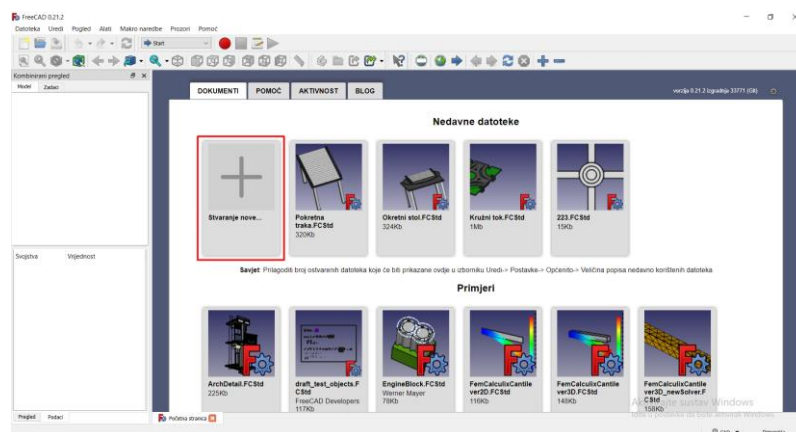
3. Izrada prototipa pokretne traka

Prije nego što započnemo s detaljnim opisom praktičnog dijela izrade prototipa pokretne trake, važno je naglasiti značaj ovog rada. Pokretna traka je ključna komponenta u mnogim industrijskim procesima, omogućujući efikasan prijenos materijala i proizvoda u proizvodnim linijama. Projektiranje i izrada prototipa pokretne trake zahtijevaju precizno planiranje, modeliranje i testiranje kako bi se osigurala funkcionalnost i pouzdanost u stvarnim uvjetima. Kod izrade prototipa pokretne neki dijelovi su pojednostavljeni kako bi mogli prikazati mogućnosti FreeCAD-a u pogledu optimizacije dizajna. Također ćemo se osvrnuti na izazove s kojima sam se susreo tijekom izrade i kako sam ih prevladalo uz pomoć ovog alata. Cilj ovog diplomskog rada je pokazati kako se moderni CAD alati mogu koristiti za unapređenje industrijskih procesa i razvoj inovativnih rješenja.

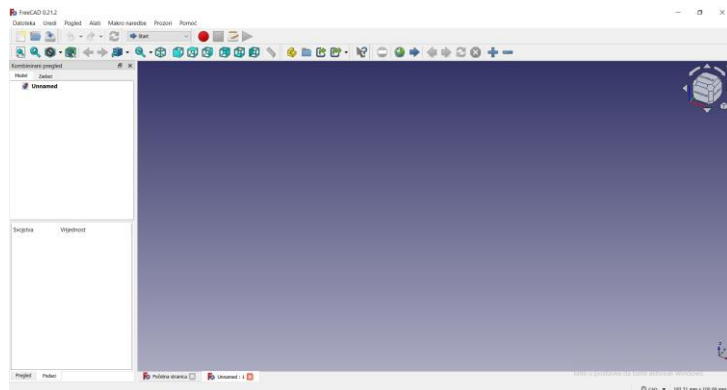
3.1. Upoznavanje FreeCAD programa

Kako bi lakše pratili izradu prototipa pokretne trake, prvo će biti opisan program, alati, ograničenja i slično koje sam koristio.

Nakon pokretanja programa, otvara se početni prozor, kao što je prikazano na slici 3.1. Ovaj prozor nudi pristup nedavnim datotekama i primjerima modela, omogućujući brzi pregled i izbor. Za početak izrade novog modela, kliknite lijevom tipkom miša na ikonu „Stvaranje nove...“. Ovaj klik preusmjerava u radno okruženje programa, kao što je prikazano na slici 3.2, gdje se može započeti s kreiranjem svog prototipa.

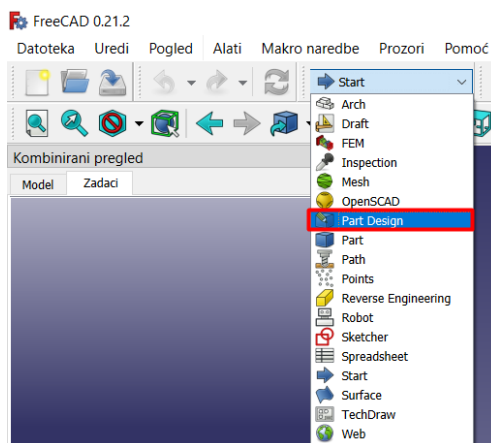


Slika 3.1 Početni prozor FreeCAD programa



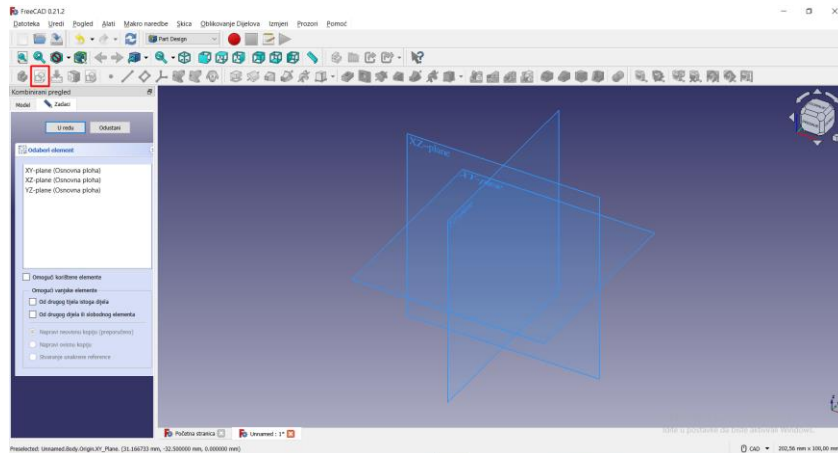
Slika 3.2 Radno okruženje

Nakon ulaska u radno okruženje, potrebno je odabrati modul za crtanje. U padajućem izborniku odabire se „Part Design“ kako bi se krenulo s kreiranjem željenog dijela. Ovaj modul omogućuje stvaranje i uređivanje 3D modela pomoću različitih alata i funkcija specifičnih za dizajn dijelova.



Slika 3.3 Odabir modula za crtanje

Kada se otvori radni prozor, klikom na ikonu „Napravite skicu“ započinje izrada skice za prvi dio željenog komada. U ovom koraku, kao što je prikazano na slici 3.4, potrebno je odabrati plohu na kojoj se želi krenuti s crtanjem.



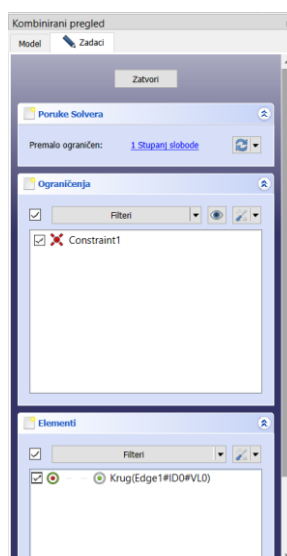
Slika 3.4 Početak skiciranja

Nakon odabira plohe za crtanje, otvara se novi prozor za crtanje. Na slici 3.5, crveni obrub označava geometrijske alate koje koristimo za crtanje, kao što su točka, crta, krug i slično. Zeleni obrub označava ograničenja, kao što su okomito, vodoravno, jednako i slično, koja moramo postaviti kako bi skica bila spremna za daljnje radnje. Plavim obrubom označeni su pomoćni alati, kao što su kreiranje simetričnog dijela, kopiranje i slično.



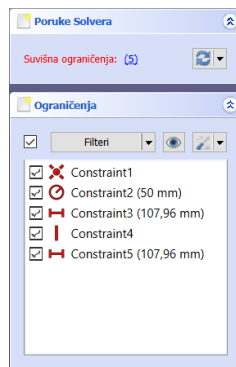
Slika 3.5 Alati koji se koriste za skiciranje

Na lijevoj strani prozora, Solver prikazuje koliko stupnjeva slobode ima skica, što je potrebno riješiti pomoću ograničenja. Ispod poruka softvera prikazana su sva ograničenja koja postoje na skici te elementi skice.



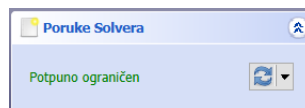
Slika 3.6 Poruke solvera, ograničenja i elementi skice

Ako postoji previše ograničenja, Solver će prijaviti grešku, kao što je prikazano na slici 3.7.



Slika 3.7 Poruka solvera o suvišnim ograničenjima

Ako je skica u potpunosti ograničena, Solver će to prikazati porukom „Potpuno ograničen“. Kada se prikaže ova poruka, skica je spremna za idući postupak.



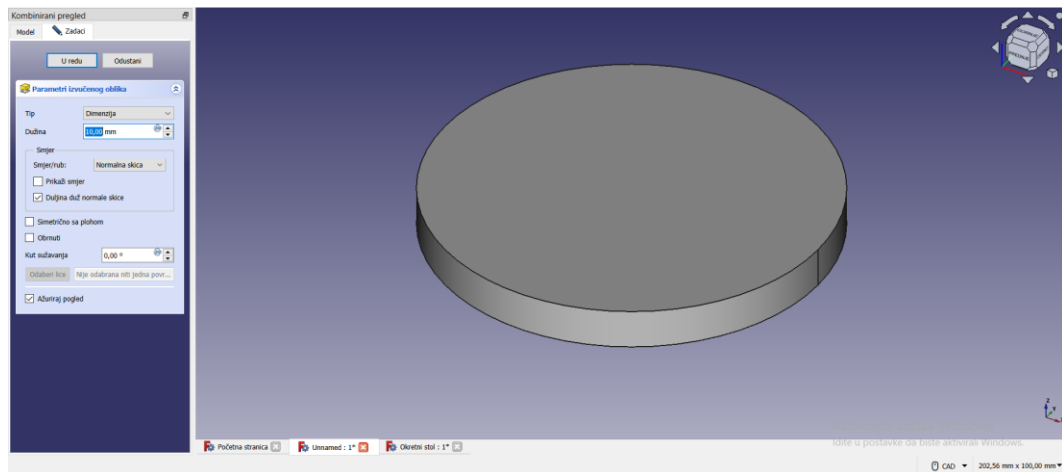
Slika 3.8 Poruka solvera o potpuno ograničenoj skici

Nakon što se završi s kreiranjem skice i klikne na „zatvori“, odabire se što se želi učiniti sa skicom. Na slici 3.9 crvenim obrubom prikazani su alati za kreiranje tijela, kao što su izvučeni oblik, kreiranje spirale i slično, te alati za oduzimanje dijelova tijela, kao što su utor, rupa i slično. Zelenim obrubom prikazani su alati za kreiranje istog dijela, kao što su zrcaljenje, linearni uzorak i slično. Plavim obrubom prikazani su dodatni alati, kao što su ravni obrub, zaobljeni obrub i slično. Crnim obrubom označeni su dodatni alati za mjerenje duljine, kuta i slično.



Slika 3.9 Alati za kreiranje 3D modela

Nakon odabira opcije za rad s skicom, otvara se prozor koji prikazuje sučelje kao na slici 3.10. Na lijevoj strani ekrana nalazi se panel gdje je moguće postaviti dimenzije ovisno o alatu koji je odabran.



Slika 3.10 Primjer 3D modela i parametara izvučenog oblika

3.2. Korišteni alati i ograničenja

Kako bi lakše pratili izradu CAD prototipa, prvo će biti opisani korišteni alati i ograničenja koja su bila korištena kod izrade.

Korišteni alati

- Izvučeni oblik (engl. *Extrude*)
 - Alat izvučeni oblik koristi se za pretvaranje dvodimenzionalne skice u trodimenzionalni objekt. Proces uključuje ekstruziju skice duž zadane osi. Parametri ekstruzije uključuju visinu ekstruzije, smjer ekstruzije (koji može biti normalan na skicu ili u nekom drugom smjeru) te opcije poput simetrične ekstruzije s obzirom na skicu.
- Utor (engl. *Cut*)
 - Alat utor koristi se za izradu udubljenja ili utora u trodimenzionalnom tijelu. Proces uključuje uklanjanje materijala iz tijela prema definiranoj skici. Parametri alata omogućuju određivanje dubine, smjera te mogućnosti poput prolaska kroz cijelo tijelo ili ograničenog utora.
- Linearni uzorak (engl. *Linear Pattern*)
 - Alat linearni uzorak omogućava ponavljanje određenog uzorka ili objekta duž jedne ili dvije osi u zadanom broju ponavljanja i na određenim udaljenostima. Ovaj alat je posebno koristan za stvaranje redova identičnih elemenata, poput rupa, utora ili nosača, na točno definiranim intervalima.

Korištena ograničenja

- Dimenzijska ograničenja (engl. *Dimensional constraints*): Ova ograničenja se koriste za definiranje točnih dimenzija linija, lukova, kružnica i drugih geometrijskih oblika. Na primjer, možete postaviti dimenzije duljine, širine ili promjera elemenata.
- Geometrijska ograničenja (engl. *Geometric constraints*): Ove vrste ograničenja koriste se za održavanje određenih geometrijskih svojstava u modelu, kao što su paralelnost, okomitost, tangencnost, simetrija ili koncentričnost.
- Odnosna ograničenja (engl. *Relational constraints*): Ovo su ograničenja koja definiraju odnose između različitih elemenata modela. Na primjer, možete definirati udaljenost između dva objekta, položaj jednog objekta u odnosu na drugi ili fiksirati određenu točku na objektu da bude na određenom mjestu.
- Ograničenja parametara (engl. *Parameter constraints*): Ova ograničenja omogućuju vam definiranje parametara u modelu koji se mogu mijenjati, a da se pri tome održava konzistentnost geometrije ili odnosa.
- Ostala ograničenja: FreeCAD također podržava različite specifične vrste ograničenja kao što su ograničenja za modeliranje elemenata kao što su zupčanici, lančanici ili prilagodbe za određene vrste specifičnih inženjerskih aplikacija.

U tablici 2. prikazana su sva ograničenja unutar FreeCAD-a.

	Postavlja ograničenje podudaranja između točaka.
	Fiksira točku na objekt.
	Postavlja okomito ograničenje linije ili između dviju točaka.
	Postavlja horizontalno ograničenje linije ili između dviju točaka.
	Postavlja paralelno ograničenje između linija.
	Postavlja okomito ograničenje između linija.
	Postavlja tangentno ograničenje između linija.
	Postavlja ograničenje jednakosti.
	Postavlja ograničenje simetrije između dvije točke s obzirom na neku liniju.
	Blokira pomicanje odabranog ruba.
	Postavlja vodoravno i okomito ograničenje na tjemenoj točki.
	Postavlja horizontalnu udaljenost između dviju točaka ili krajeva linija.
	Postavlja vertikalnu udaljenost između dviju točaka ili krajeva linija.
	Postavlja duljinu linije ili razmak između linije i tjemene točke ili razmak između dva kruga.
	Postavlja ograničenje promjera ili radijusa na luk ili krug.
	Postavlja ograničenje kuta linije ili između linija.

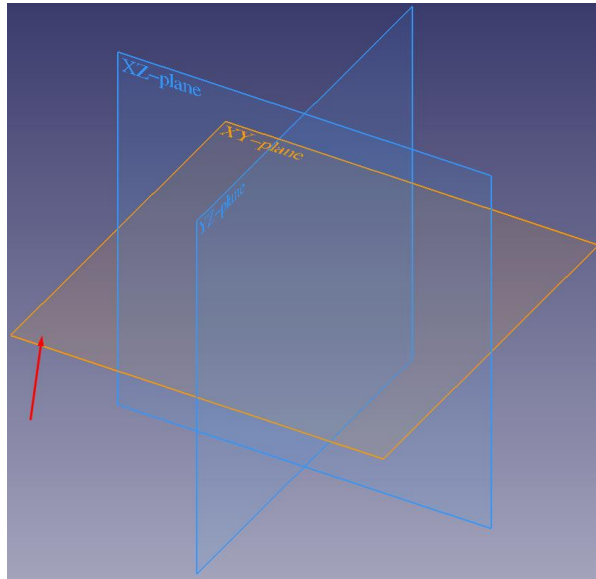
Tablica 2. Ograničenja u FreeCAD-u [4]

3.3. Izrada CAD modela

U nastavku će biti detaljno opisani koraci za izradu 3D prototipa pokretne trake.

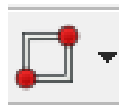
Kako bi se kreirao CAD model pokretne trake, potrebno je prvo stvoriti novi model, kao što je prethodno opisano u tekstu.

Izrada pokretne trake započinje izradom postolja. Za izradu skice postolja, odabire se XY ploha, prikazana na slici 3.11.

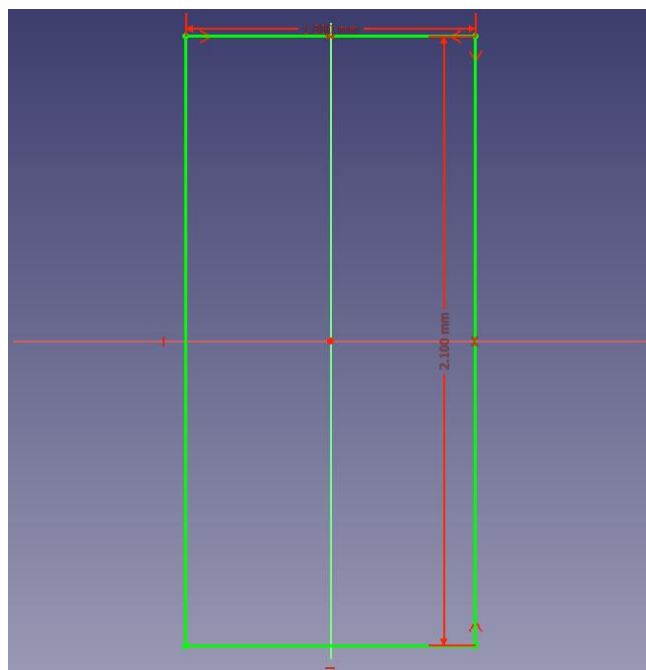


Slika 3.11 Prikaz odabira plohe

Na slici 3.13 prikazana je skica postolja. Za izradu pravokutnika korišten je alat sa slike 3.12, koji automatski postavlja okomita i vodoravna ograničenja na stranice pravokutnika. Kako bi pravokutnik uvijek bio u sredini plohe, izbrisana su ograničenja okomitosti i vodoravnosti s gornje i desne stranice te su na ta mjesta postavljena ograničenja simetričnosti. Dimenzije pravokutnika su postavljene na 1000 mm širine i 2100 mm dužine.



Slika 3.12 Alat za crtanje pravokutnika

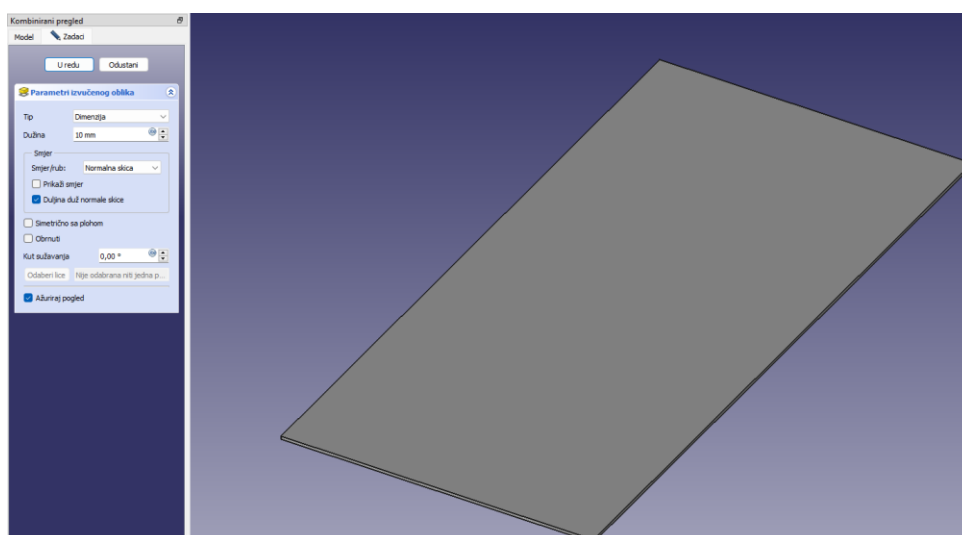


Slika 3.13 Potpuna skica postolja

Za izradu izvučenog dijela korišten je alat prikazan na 3.14. Na slici 3.15 prikazan je izvučeni oblik s postavljenim parametrima.



Slika 3.14 Alat za izradu izvučenog oblika



Slika 3.15 Izvučeni oblik postolja i postavljeni parametri

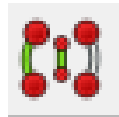
Nakon izrade postolja, izrađuje se skica za nosače valjaka na XZ plohi. Nosači valjaka su U oblika i dimenzija prikazanih na slici 3.16. Sve linije nosača su

vodoravne ili okomite. Debljina nosača je 10 mm, a visina 120 mm. Gornji i donji dio nosača su uvijek jednake duljine, pa je između točaka postavljena okomitost radi dobivanja automatskih dimenzija.

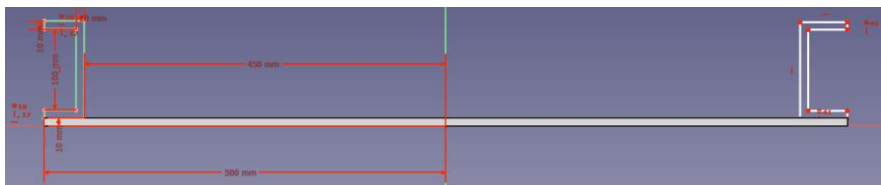


Slika 3.16 Skica U nosača valjaka

Za izradu nosača s druge strane označuje se prvi nosač i krajnja okomita linija koordinatnog sustava te se pomoću alata za stvaranje simetričnog dijela, prikazanog na slici 3.17, stvara drugi nosač.



Slika 3.17 Alat za kreiranje simetrije



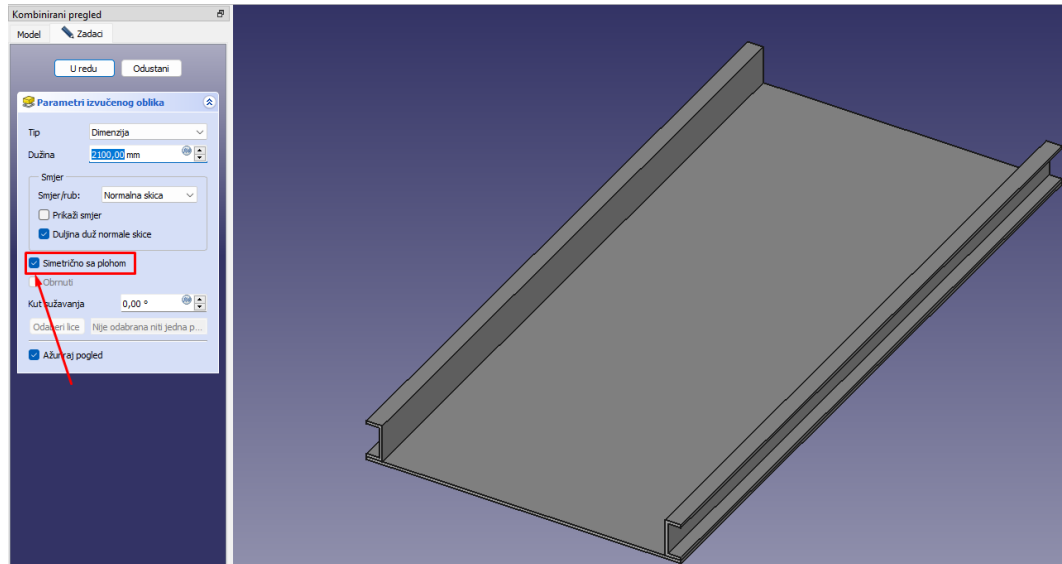
Slika 3.18 Skica nakon kreiranja simetričnog U nosača

Nakon stvaranja simetričnog dijela, postavljaju se identična ograničenja dimenzija na drugi nosač. Kako bi nosač bio na rubu postolja, postavljena je dimenzija od 500 mm (pola širine postolja). Unutarnji dio nosača također je ograničen posebnom dimenzijom od 450 mm (500 mm - 50 mm, širina nosača). Krajnja skica prikazana je na slici 3.19.



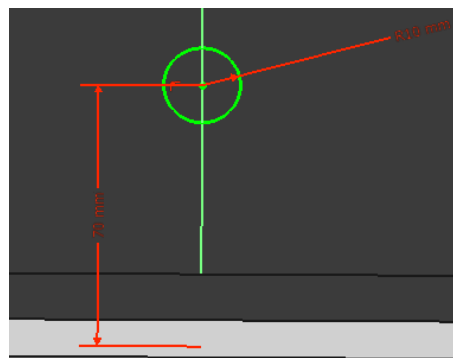
Slika 3.19 Potpuno ograničena skica nosača

Za izradu izvučenog dijela korišten je isti alat kao i za postolje. Kako bi se izvučeni oblik izradio simetrično s obje strane plohe, mora se označiti „Simetrično s plohom“, kao što je prikazano na slici 3.20.



Slika 3.20 Izvučeni oblik i parametri U nosača

Nakon izrade nosača, potrebno je izraditi provrte za valjke. Za izradu skice odabire se YZ ploha te se na sredinu postavlja rupa radijusa 10 mm. Skica i ograničenja prikazani su na slici 3.21.

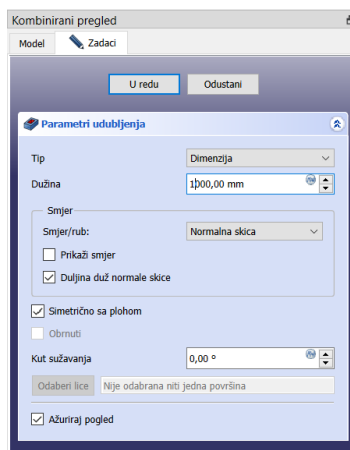


Slika 3.21 Skica utora u koje će se postavljati valjci

Za izradu rupa s obje strane korišten je alat „Utor“, prikazan na slici 3.22. Parametri za izradu rupa vidljivi su na slici 3.23.



Slika 3.22 Alat za kreiranje utora

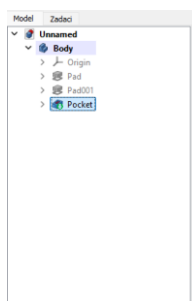


Slika 3.23 Parametri za kreiranje utora

Kako bi se identične rupe izradile duž cijele dužine nosača, koristi se alat „Linearni uzorak“, prikazan na slici 3.24. Prvo se odabire „Pocket“ jer je to dio od kojeg se želi napraviti linearni uzorak, a zatim se klikne na ikonu alata. Drugi način je klik na alat za izradu linearnog uzorka, zatim odabir „Pocket“.

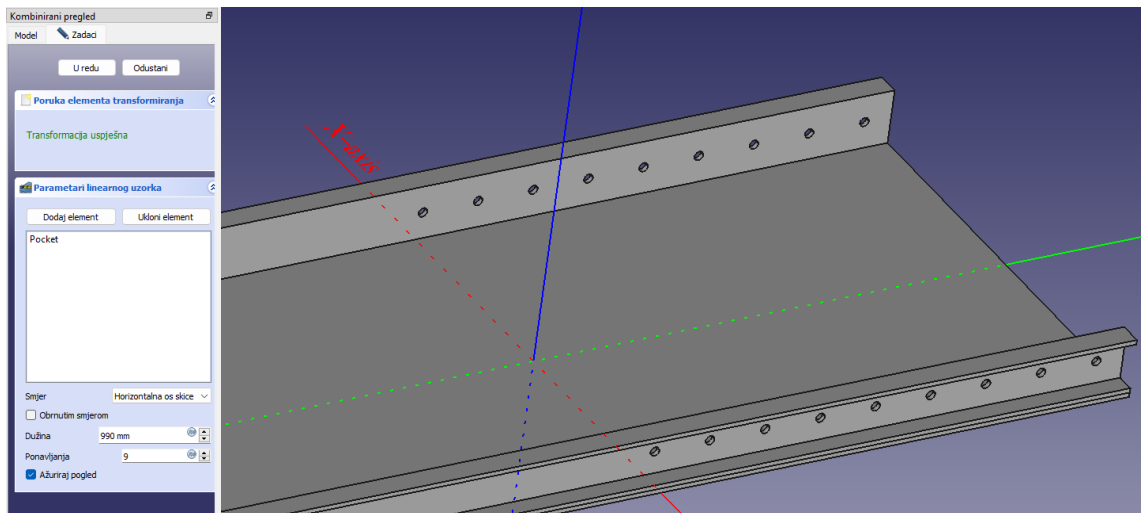


Slika 3.24 Alat za kreiranje linearnog uzorka



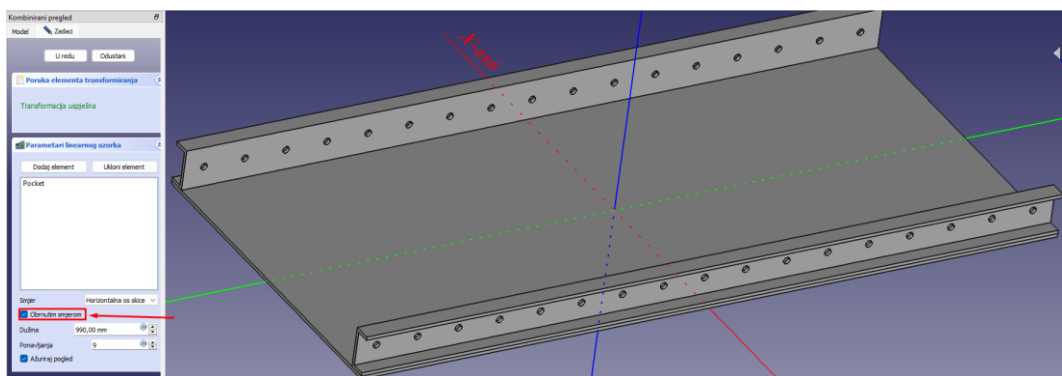
Slika 3.25 Prikaz izrađenog „Pocketa“

Parametri koji se postavljaju uključuju smjer (postavljen automatski), dužinu po kojoj se žele rasporediti uzorci i broj uzoraka. Skice su postavljene tako da s obje strane ima po 8 uzoraka na udaljenosti od 110 mm, jer će valjci biti 100 mm. Parametri su prikazani na slici 3.26.



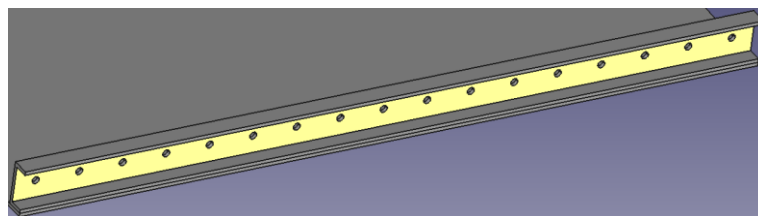
Slika 3.26 Linearni uzorak i njegovi parametri

Za izradu rupa i s druge strane ponavlja se postupak, ali se dodatno odabire obrnuti postupak. Parametri su prikazani na slici 3.27.



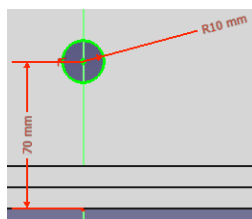
Slika 3.27 Linearni uzorak s druge strane i njegovi parametri

Nakon izrade rupa, prelazi se na izradu nosača za valjke koji idu u utore. Odabire se ploha nosača, kao na slici, i klikne na ikonu za izradu skice kako bi se radila skica na toj plohi.



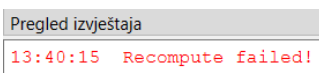
Slika 3.28 Odabir plohe na kojoj želimo raditi skicu

Zbog ograničenja programa da ništa ne može slobodno letjeti u zraku, potrebno je postaviti promjer nosača isti kao i promjer rupe.

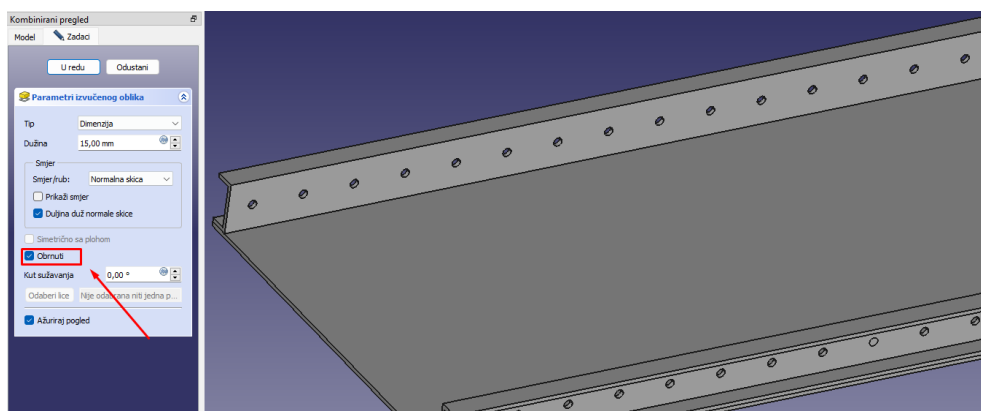


Slika 3.29 Skica nosaa valjka koji idu u utore

Nakon zatvaranja skice i pokušaja izrade izvučenog dijela, program prikazuje grešku, kao na slici 3.30. Greška se pojavila jer dio letio u zraku i nije povezan ni s čim, ali nakon odabira „Obrnuti“, izvučeni dio se izvlači na drugu stranu i prolazi kroz rupu. Na slici 3.31 prikazani su parametri izvučenog oblika.

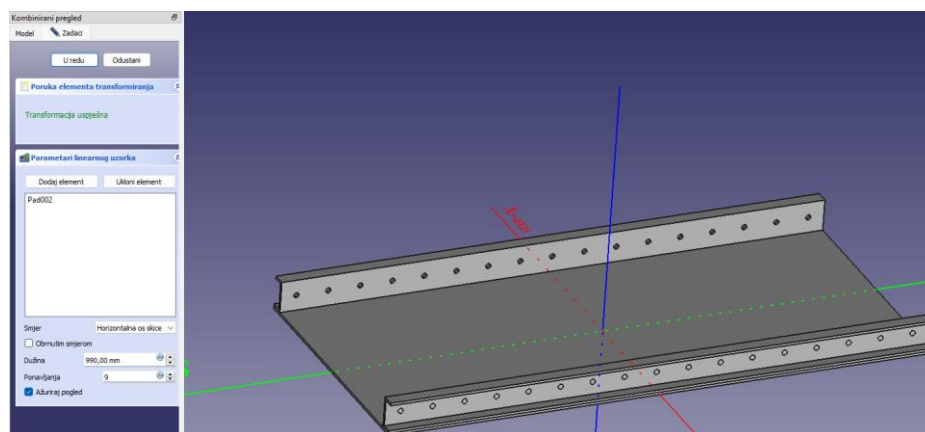


Slika 3.30 Greška kod kreiranja izvučenog dijela



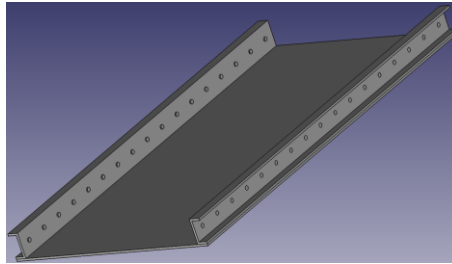
Slika 3.31 Izvučeni oblik i njegovi parametri

Pomoću linearnog uzorka izrađuju se nosači s lijeve i desne strane nosaa, koristeći isti postupak kao za utore. Parametri su prikazani na slici 3.32.



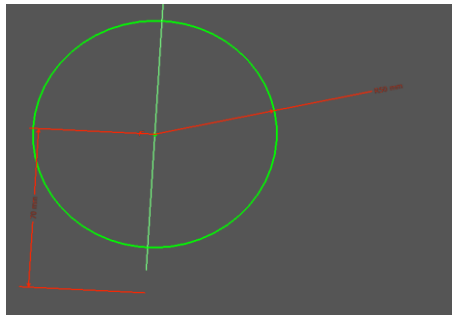
Slika 3.32 Prikaz nosaa s lijeve i desne strane U nosaa te njihovi parametri

Isti postupak se ponavlja na drugom U nosaču.



Slika 3.33 Prikaz nosača valjaka na oba U nosača

Nakon izrade nosača s obje strane, izrađuju se valjci po kojima se proizvodi kreću u industriji. Ponovno se izrađuje skica na YZ plohi. Promjer valjaka je 100 mm, što je prikazano na slici 3.34.



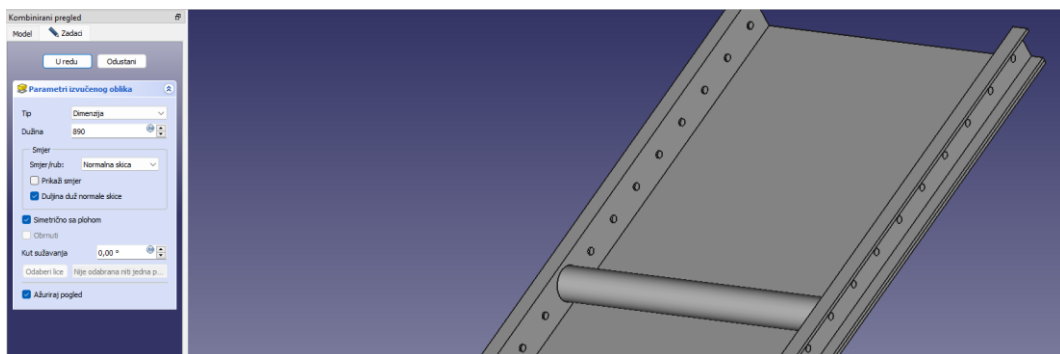
Slika 3.34 Skica valjaka

Nakon pokušaja izrade izvučenog dijela, program ponovno prikazuje grešku jer izvučeni dio leti u zrak.



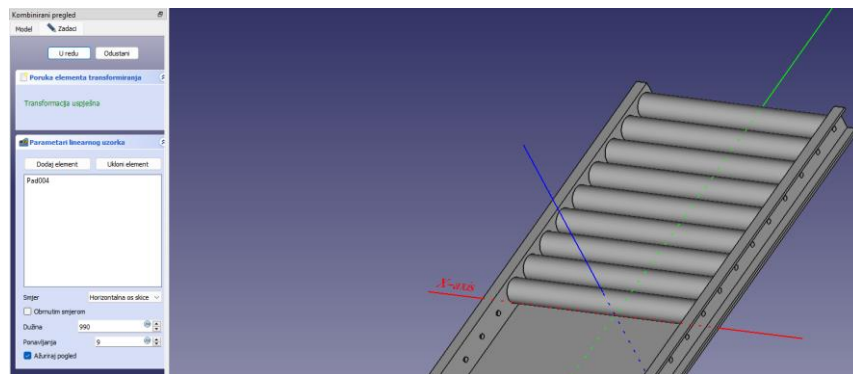
Slika 3.35 Greška kod izrade valjaka

Nakon postavljanja dimenzije tako da dio dodiruje neki dio pokretne trake, on se automatski pojavljuje. Parametri za izradu valjka prikazani su na slici 3.36.



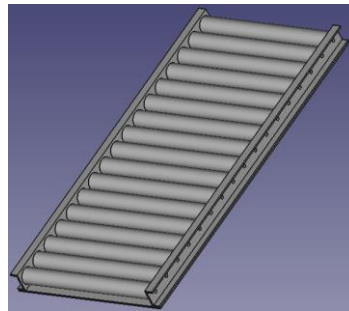
Slika 3.36 Izvučeni oblik i parametri za izradu osnovnog valjka

Za izradu ostalih valjaka koristi se isti postupak kao za nosače valjaka i utore. Parametri su prikazani na 3.37.



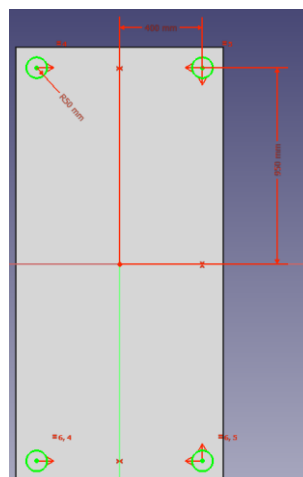
Slika 3.37 Linearni uzorak i parametri za izradu valjaka

Na slici 3.38 prikazana je gotova pokretna traka bez nogu.



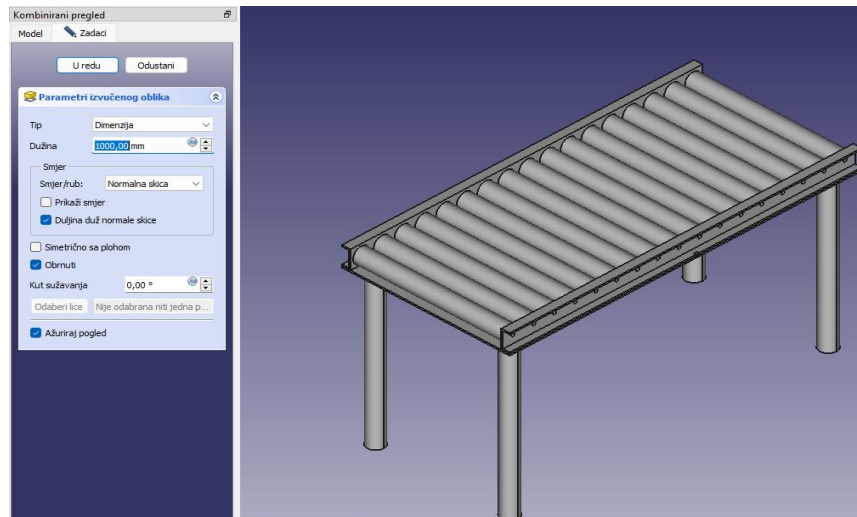
Slika 3.38 Pokretna traka bez nogu

Za reguliranje visine pokretne trake, potrebno je postaviti oslonce ili noge. Postavljene su okrugle noge promjera 100 mm i udaljene 100 mm od ruba trake. Kako ne bi bilo potrebno postavljati svaku nogu posebno, sve su noge postavljene jednakih promjera i simetrične su s obzirom na koordinatni sustav.



Slika 3.39 Skica oslonca ili nogu

Osim dimenzija, potrebno je odabrati „Obrnuti“ kako bi noge bile izvučene na drugu stranu, a ne kroz valjke. Nakon postavljanja izvučenog dijela, gotova pokretna traka prikazana je na slici 3.40.



Slika 3.40 Parametri za izradu nogu i gotova pokretna traka

3.4. Python u FreeCAD-u

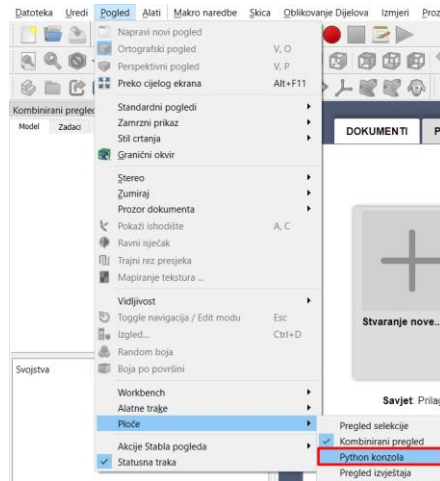
Python je programski jezik otvorenog koda koji se široko koristi zbog svoje jednostavnosti i svestranosti. U FreeCAD-u, Python se koristi za skriptiranje, omogućavajući korisnicima kontrolu nad gotovo svakim aspektom softvera.

Python se ističe po: [10]

- Jednostavnosti učenja: Python je poznat po svojoj jednostavnosti, što ga čini pristupačnim za korisnike koji nemaju prethodno iskustvo u programiranju.
- Ugrađeno skriptiranje: Python je ugrađen u mnoge popularne aplikacije poput Blendera, Inkscapea i GRASS-a, što ga čini poznatim alatom među korisnicima.
- Kontrola i prilagodba: Korištenjem Pythona, korisnici mogu kontrolirati gotovo svaku značajku FreeCAD-a, stvarati nove objekte, modificirati geometriju, analizirati sadržaje i stvarati nove alate i panele sučelja.

FreeCAD uključuje naprednu Python konzolu koja je dostupna putem izbornika (Pogled -> Ploče -> Python konzola), što je prikazano na slici 3.57. Ova konzola omogućuje korisnicima izvođenje raznih operacija, provjeru oblika na probleme i

automatizaciju ponavljajućih zadataka. Konzola također prikazuje Python kod koji se izvršava putem alatnih traka, omogućujući korisnicima učenje Pythona promatrajući kod u akciji.



Slika 3.41 Putanja za otvaranje Python konzole

Nakon klika na Python konzola ona nam se otvara na dnu ekrana kao na slici 3.58.

```
Python konzola
>>> App.getDocument('pokretna_traka').getObject('Pad004').TaperAngle = 0.000000
>>> App.getDocument('pokretna_traka').getObject('Pad004').UseCustomVector = 0
>>> App.getDocument('pokretna_traka').getObject('Pad004').Direction = (1, -0, 0)
>>> App.getDocument('pokretna_traka').getObject('Pad004').ReferenceAxis = (App.getDocument('pokretna_traka').getObject('Sketch005'), ('N_Axis'))
>>> App.getDocument('pokretna_traka').getObject('Pad004').AlongSketchNormal = 1
>>> App.getDocument('pokretna_traka').getObject('Pad004').Type = 0
>>> App.getDocument('pokretna_traka').getObject('Pad004').OptoFace = None
>>> App.getDocument('pokretna_traka').getObject('Pad004').Reversed = 0
>>> App.getDocument('pokretna_traka').getObject('Pad004').Midplane = 1
>>> App.getDocument('pokretna_traka').getObject('Pad004').Offset = 0
>>> App.getDocument('pokretna_traka').recompute()
>>>
```

Slika 3.42 Python konzola na dnu

FreeCAD-ov Python interpreter omogućava korisnicima interakciju s različitim modulima softvera putem skripti: [10]

- Moduli i funkcije: Korisnici mogu uvoziti Python module i koristiti njihove funkcije unutar FreeCAD-a. Na primjer, funkcija definirana u Python datoteci može se uvesti i koristiti unutar FreeCAD interpretera.
- Pomoć i dokumentacija: Pythonova funkcija help() i naredba dir() su korisne za istraživanje dostupnih modula i njihovih funkcija. FreeCAD pruža Python pomoćnu opciju u izborniku Help, nudeći dokumentaciju u stvarnom vremenu za sve dostupne Python module.

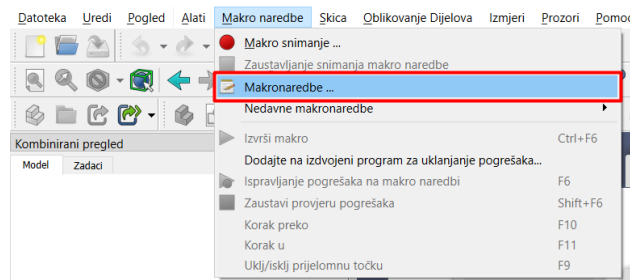
3.4.1. Upotreba makronaredbi u FreeCAD-u za optimizaciju laboratorijskog stola i pokretne trake

FreeCAD omogućuje korisnicima prilagođavanje i proširenje funkcionalnosti softvera kroz makronaredbe, što predstavlja važan alat za automatizaciju zadataka i prilagođavanje procesa dizajniranja. Makronaredbe su skripte koje bilježe radnje korisnika u FreeCAD-u te ih je moguće ponovno izvršiti u bilo kojem trenutku, čime se značajno smanjuje vrijeme potrebno za izvođenje ponavljajućih operacija. One su napisane u Python programskom jeziku, što omogućuje korisnicima naprednu manipulaciju objektima i prilagodbu različitih aspekata modela. [11]

U ovom diplomskom radu, makronaredbe su korištene za optimizaciju dizajna laboratorijskog stola i pokretne trake. Optimizacija laboratorijskog stola zahtijevala je prilagodbu dimenzija gornje ploče, u skladu sa specifičnim zahtjevima korisnika, kao i izmjene razmaka između nogu stola te promjene polumjera nogu. S obzirom na složenost pokretne trake, optimizacija ovog sustava uključivala je definiranje dimenzija U profila koji nose valjke, izradu otvora za postavljanje valjaka, te određivanje njihovog broja u odnosu na ukupnu duljinu pokretne trake. Također, bilo je potrebno postaviti duljine valjaka, kao i učestalost ponavljanja valjaka i nosača koji idu u otvore na U profilima duž trake. Pisanje Python programa omogućilo je automatizaciju složenih proračuna i simulacija koje bi inače zahtijevale više ručnih intervencija.

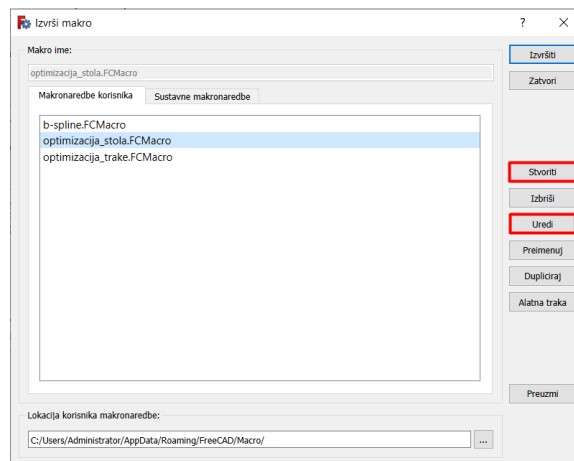
Prednost korištenja makronaredbi i Pythona u FreeCAD-u očituje se u mogućnosti ponovne upotrebe razvijenih skripti u različitim projektima, čime se ne samo štedi vrijeme nego i osigurava dosljednost rezultata. Ovaj pristup omogućuje inženjerima i dizajnerima da brzo evaluiraju više scenarija dizajna bez potrebe za ručnim mijenjanjem svakog parametra, što čini proces dizajniranja efikasnijim i preciznijim.

Na slici 3.59 prikazan je postupak za otvaranje prozora za kreiranje, brisanje, uređivanje i slično makronaredbi.



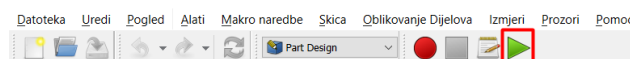
Slika 3.43 Otvaranja prozora za rad s makronaredbama

Postupak stvaranja nove makronaredbe u FreeCAD-u izuzetno je jednostavan. U novom prozoru odabire se opcija „Stvori“ te se unosi željeni naziv makronaredbe. Nakon toga otvara se prazna makronaredba u kojoj korisnik može upisivati Python program za daljnje izvršavanje. Za uređivanje postojeće makronaredbe potrebno je odabrati željenu makronaredbu i zatim kliknuti na opciju „Uredi“.



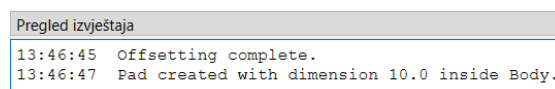
Slika 3.44 Prozor za rad s makronaredbama

Kada je gotovo programiranje, makronaredba se pokreće klikom na ikonu prikazanu na slici 3.61.



Slika 3.45 Izvršavanje makronaredbe

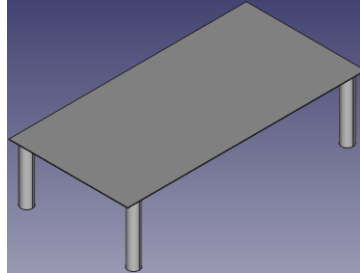
Ako program sadrži greške ili je postavljeno ispisivanje rezultata, automatski se otvara prozor „Pregled izvještaja“, kao što je prikazano na slici 3.62.



Slika 3.46 Pregled izvještaja na dnu

3.4.2. Program za optimizaciju prototipa laboratorijskog stola

Program za optimizaciju prototipa laboratorijskog stola bit će opisan tokom koda. Optimizirana je širina i dužina stola te njegova visina. Na slici 3.47 prikazan je prototip laboratorijskog stola.



Slika 3.47 Prototip laboratorijskog stola

```
# Uvoz potrebnih modula iz FreeCAD-a
import FreeCAD as App
import FreeCADGui as Gui
import Sketcher
from PySide import QtGui
```

„FreeCAD as App“ modul omogućava pristup API-ju (engl. *Application Programming Interface*) za manipulaciju dokumentima u FreeCAD-u. „FreeCADGui as Gui“ omogućava rad s grafičkim sučeljem FreeCAD-a. „Sketcher“ je modul koji omogućava rad sa skicama, dok „QtGui“ omogućava kreiranje grafičkog sučelja za unos podataka.

```
# Funkcija za unos vrijednosti putem dijaloga
def get_user_input(prompt):
    value, ok = QtGui.QInputDialog.getDouble(None, "Unos
vrijednosti", prompt, decimals=2) # Otvara dijalog za unos
decimalnog broja
    if ok:
        return value # Vraća unesenu vrijednost ako je unos
uspješan
    else:
        raise ValueError("Unos je prekinut od strane
korisnika.") # Podiže iznimku ako je unos prekinut
```

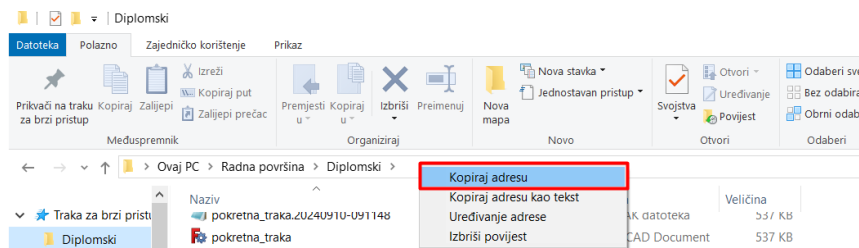
„def get_user_input“ definira funkciju koja prima tekst kao parametar. Ova funkcija kreira prozor za unos decimalnog broja s dvije decimale. Ako korisnik klikne „ok“, funkcija vraća unesenu vrijednost. U suprotnom, ako korisnik prekinu unos, ispisuje se poruka o grešci „Unos je prekinut od strane korisnika“.

```

# Otvaranje postojećeg modela
model_path =
'C:/Users/Administrator/Desktop/Diplomski/stol.FCStd' #
Zamijeniti s odgovarajućom putanjom
App.open(model_path) # Otvara FreeCAD dokument s navedenom
putanjom
doc = App.activeDocument() # Dohvaća aktivni dokument

```

Sljedeći korak u programu je otvaranje modela koji je prethodno izrađen. U varijablu „model_path“ unosi se lokacija modela na računalu. Desnim klikom na putanju otvaraju se opcije, kao što je prikazano na slici 3.47, gdje je moguće kopirati adresu s računala. Nakon kopiranja, potrebno je promijeniti „\“ u „/“ te na kraju dodati naziv modela s nastavkom „FCStd“. Funkcija „App.open“ otvara model čija je lokacija definirana varijablom „model_path“. U sljedećem koraku, funkcija „doc = App.activeDocument()“ dohvaća aktivni dokument za daljnju manipulaciju.



Slika 3.48 Kopiranje lokacije modela

```

# Unos vrijednosti od korisnika za duljinu, širinu i visinu
pokretne trake
try:
    duljina_stola = get_user_input("Unesite duljinu stola:")
# Unosi duljinu stola
    sirina_stola = get_user_input("Unesite širinu stola:")
# Unosi širinu stola
    visina_stola = get_user_input("Unesite visinu stola:")
# Unosi visinu stola
except ValueError as e:
    print(e) # Ispisuje poruku o grešci
    raise SystemExit # Prekida izvršenje programa

```

Nakon otvaranja dokumenta, koristi se funkcija „get_user_input“ za prikupljanje željenih vrijednosti od korisnika. Te vrijednosti se pohranjuju pod nazivima „duljina_stola“, „sirina_stola“ i „visina_stola“. U slučaju greške, ispisuje se poruka o grešci i program se prekida.

```

# Pronalazimo skicu koju želimo izmijeniti (u ovom slučaju
Sketch)
sketch = doc.getObject('Sketch') # Dohvaća objekt skice s
imenom 'Sketch'

```

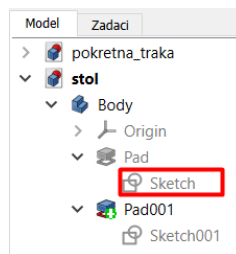


```

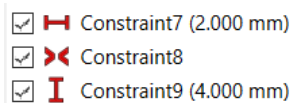
constraint7_index = 6 # Postavlja indeks za Constraint7-
širina
constraint9_index = 8 # Postavlja indeks za Constraint9-
duljina

```

Kako bi se omogućilo mijenjanje vrijednosti ograničenja unutar skice, potrebno je prvo dohvatiti željenu skicu pomoću funkcije „doc.getObject('Naziv skice)“. Naziv skice prikazan je na slici 3.48 i ograničenja unutar skice na slici 3.49. Za prilagodbu određenih ograničenja, potrebno je definirati njihove indekse. Indeksi ograničenja 7 i 19, koji predstavljaju širinu i dužinu, pohranjeni su pod nazivima „constraint7_index“ i „constraint9_index“. Njihove vrijednosti su broj ograničenja minus jedan jer indeksi počinju od nule, dok ograničenja počinju od 1.



Slika 3.49 Naziv skice



Slika 3.50 Ograničenja ploče stola

```

# Postavljanje širine i duljine stola
try:
    sketch.setDatum(constraint7_index,
App.Units.Quantity(f'{sirina_stola} mm')) # Postavlja
Constraint7 na širinu stola
    sketch.setDatum(constraint9_index,
App.Units.Quantity(f'{duljina_stola} mm')) # Postavlja
Constraint9 na dljinu stola
except Exception as e:
    print("Greška pri promjeni dimenzija ploče stola:", e)
# Ispisuje poruku o grešci kod postavljanja dimenzija
raise SystemExit # Prekida izvršenje programa

```

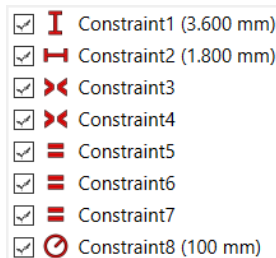
Kada je skica definirana, kao i ograničenja koja je potrebno mijenjati, postavljaju se vrijednosti tih ograničenja na željene dimenzije. U ovom slučaju to su širina i duljina stola. Kako bi se omogućilo praćenje svakog dijela koda, dodana je poruka o grešci kako bi se identificiralo gdje je u kodu došlo do problema.

```

# Pronalazi skicu (Sketch001)
sketch001 = doc.getObject('Sketch001') # Dohvaća skicu s
imenom Sketch001
# Unesite odgovarajuće indekse za
constraint1_index = 0 # Postavlja indeks za Constraint1
koji predstavlja duljinu između nogu
constraint2_index = 1 # Postavlja indeks za Constraint2
koji predstavlja širinu između nogu
constraint8_index = 7 # Postavlja indeks za Constraint8
koji predstavlja radius nogu

```

Nakon što su postavljene duljina i širina stola, dohvaća se skica nogu. Kada je skica dohvaćena, postavljaju se indeksi ograničenja za razmak između nogu (ograničenja 1 i 2) te za radius nogu (ograničenje 8).



Slika 3.51 Ograničenja nogu stola

```

# Postavljanje ograničenja za noge
try:
    sketch001.setDatum(constraint1_index,
App.Units.Quantity(f'{duljina_stola - duljina_stola/10}
mm')) # Postavlja duljinu između nogu
    sketch001.setDatum(constraint2_index,
App.Units.Quantity(f'{sirina_stola - sirina_stola/10} mm'))
# Postavlja širinu između nogu
    sketch001.setDatum(constraint8_index,
App.Units.Quantity(f'{duljina_stola/40} mm')) # Postavlja
radius nogu
except Exception as e:
    print("Greška pri promjeni ograničenja nosača:", e) #
Ispisuje poruku o grešci kod postavljanja dimenzija nogu
    raise SystemExit # Prekida izvršenje programa

```

Postavljanjem indeksa ograničenja koja želimo mijenjati možemo krenuti sa promjenom njihove vrijednost. Noge će biti postavljene tako da uvijek budu u kutu stola bez obzira na vrijednost varijabli (9/10 širine i duljine). Kako noge nebi bile pretanke ili predebele postavljeno je proizvoljno da njihova debljina ovisi o dužini stola (duljina/40).

```

# Pronalazimo postojeći izvučeni oblik nogu
pad001 = doc.getObject('Pad001') # Dohvaća objekt Pad001

```

Kako bi se mogla kontrolirati visina stola, mora se dohvatiti izvučeni oblik nogu.

```
# Postavlja dimenzije izvučenog oblika nogu stola
try:
    pad001.Length = visina_stola # Postavlja duljinu nogu
na visinu stola
except Exception as e:
    print("Greška pri postavljanju nosača:", e) # Ispisuje
poruku o grešci
    raise SystemExit # Prekida izvršenje programa
```

Nakon dohvaćanja izvučenog oblika nogu, postavlja se njegova duljina na vrijednost varijable „visina_stola“.

```
# Ponovno računanje dokumenta
doc.recompute()
# Spremanje dokumenta s novim izmjenama
doc.saveAs('C:/Users/Administrator/Desktop/Diplomski/stol.FC
Std') # Zamijeniti s odgovarajućom putanjom
```

„doc.recompute()“ ponovno preračunava model kako bi se primijenile sve promjene. Kako bi se sačuvao novi model, koristi se „doc.saveAs (...)“.

```
# Prikaz rezultata
Gui.activeDocument().activeView().viewAxometric() #
Postavlja pogled na aksonometrijski prikaz
Gui.SendMsgToActiveView("ViewFit") # Prilagođava prikaz
veličini modela
```

Na kraju programa postavlja se aksonometrijski prikaz modela i prilagođava prikaz veličini modela. Aksonometrijski prikaz je metoda koja se koristi za prikazivanje 3D modela na 2D papiru ili ekranu, pri čemu se očuvavaju proporcije modela.

3.4.3. Program za optimizaciju prototipa pokretne trake

Program za optimizaciju trake je sličan programu za optimizaciju laboratorijskog stola uz postavljanje nekoliko više izvučenih oblika, ograničenja itd. Za lakše praćenje koda dodani su komentari na svaku liniju koda.

```
# Uvoz potrebnih modula iz FreeCAD-a
import FreeCAD as App
import FreeCADGui as Gui
import Sketcher
from PySide import QtGui
import math
```

Uvezeni su moduli kao i u prethodnom programu uz dodatak modula „math“. Modul „math“ koristi se za matematičke funkcije, u ovom slučaju zaokruživanje kod broja ponavljanja.

```
# Funkcija za unos vrijednosti putem dijaloga
def get_user_input(prompt):
    value, ok = QtGui.QInputDialog.getDouble(None, "Unos
    vrijednosti", prompt, decimals=2) # Otvara dijalog za unos
    decimalnog broja
    if ok:
        return value # Vraća unesenu vrijednost ako je unos uspješan
    else:
        raise ValueError("Unos je prekinut od strane korisnika.") #
        Podiže iznimku ako je unos prekinut
```

Kreirana je funkcija za uzimanje vrijednost kao i u prethodnom programu.

```
# Otvaranje postojećeg modela
model_path =
'C:/Users/Administrator/Desktop/Diplomski/pokretna_traka.FCS
td' # Zamijeniti s odgovarajućom putanjom
App.open(model_path) # Otvara FreeCAD dokument s navedenom
putanjom
doc = App.activeDocument() # Dohvaća aktivni dokument
```

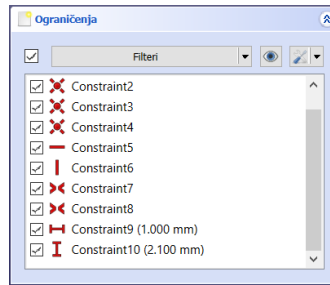
Dohvaćanje modela je također jednako uz promjenu naziva modela.

```
# Unos vrijednosti od korisnika za duljinu, širinu i visinu
pokretne trake
try:
    duljina_trake = get_user_input("Unesite duljinu trake:") #
    Unosi duljinu trake
    sirina_trake = get_user_input("Unesite širinu trake:") #
    Unosi širinu trake
    visina_trake = get_user_input("Unesite visinu trake:") #
    Unosi visinu trake
except ValueError as e:
    print(e) # Ispisuje poruku o grešci
    raise SystemExit # Prekida izvršenje programa
```

Na isti način, uz malu promjenu teksta, uzimamo vrijednosti koje korisnik želi.

```
# Pronalazimo skicu koju želimo izmijeniti (u ovom slučaju
Sketch)
sketch = doc.getObject('Sketch') # Dohvaća objekt skice s
imenom 'Sketch'
constraint9_index = 8 # Postavlja indeks za Constraint9-
širina
constraint10_index = 9 # Postavlja indeks za Constraint10-
duljina
```

Otvora se skica „Sketch“ te se postavljaju indeksi za ograničenja 9 i 10.



Slika 3.52 Ograničenja unutar skice "Sketch"

```
# Postavljanje širine i duljine trake
try:
    sketch.setDatum(constraint9_index,
    App.Units.Quantity(f'{sirina_trake} mm')) # Postavlja
    Constraint9 na širinu trake
    sketch.setDatum(constraint10_index,
    App.Units.Quantity(f'{duljina_trake} mm')) # Postavlja
    Constraint10 na duljinu trake
except Exception as e:
    print("Greška pri promjeni dimenzija nosača:", e) # Ispisuje
    poruku o grešci kod postavljanja dimenzija
    raise SystemExit # Prekida izvršenje programa
```

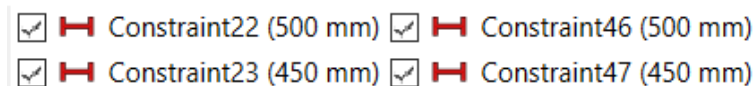
Kada je skica definirana, kao i ograničenja koja je potrebno mijenjati, postavljaju se vrijednosti tih ograničenja na željene dimenzije. U ovom slučaju to su širina i duljina trake.

```
# Pronalazi skicu (Sketch001)
sketch001 = doc.getObject('Sketch001') # Dohvaća skicu s
imenom Sketch001
```

Dohvaća „Sketch001“ koja je skica U profila.

```
# Unesite odgovarajuće indekse za vanjska i unutarnja
ograničenja nosača
constraint23_index = 22 # Postavlja indeks za Constraint22-
unutarnji
constraint22_index = 21 # Postavlja indeks za Constraint23
constraint47_index = 46 # Postavlja indeks za Constraint44-
unutarnji
constraint46_index = 45 # Postavlja indeks za Constraint45
```

Dohvaćanje ograničenja koja se žele mijenjati.



Slika 3.53 Ograničenja koja želimo mijenjati

```

# Postavljanje ograničenja za U profile
try:
sketch001.setDatum(constraint22_index,
App.Units.Quantity(f'{sirina_trake / 2} mm')) # Postavlja
vanjsko ograničenje
sketch001.setDatum(constraint46_index,
App.Units.Quantity(f'{sirina_trake / 2} mm')) # Postavlja
vanjsko ograničenje
sketch001.setDatum(constraint23_index,
App.Units.Quantity(f'{sirina_trake / 2 - 50} mm')) #
Postavlja unutarnje ograničenje
sketch001.setDatum(constraint47_index,
App.Units.Quantity(f'{sirina_trake / 2 - 50} mm')) #
Postavlja unutarnje ograničenje
except Exception as e:
print("Greška pri promjeni ograničenja nosača:", e) #
Ispisuje poruku o grešci kod promjene ograničenja nosača (U
profila)
raise SystemExit # Prekida izvršenje programa

```

Promjena postavljenih ograničenja, kako bi U profili bili na rubu pokretne trake.

Nakraju ispisuje grešku kako bi znali u kojem dijelu koda je došlo do pogreške.

```

# Pronalazimo skicu (Sketch006)
sketch006 = doc.getObject('Sketch006') # Dohvaća skicu s
imenom Sketch006

```

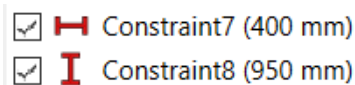
Pronalazi „Sketch006“ koja je skica nogu.

```

# Unesite odgovarajuće indekse za širinu i duljinu između
nogu trake
constraint7_index = 6 # Postavlja indeks za širinsku
udaljenost nogu
constraint8_index = 7 # Postavlja indeks za duljinsku
udaljenost nogu

```

Dohvaćanje ograničenja koja se mijenjaju.



Slika 3.54 Ograničenja u "Sketch006" koja se mijenjaju

```

# Postavljanje širine i duljine između nogu trake
try:
sketch006.setDatum(constraint8_index,
App.Units.Quantity(f'{duljina_trake / 2 - 100} mm')) #
Postavlja duljinu između nogu trake
sketch006.setDatum(constraint7_index,
App.Units.Quantity(f'{sirina_trake / 2 - 100} mm')) #
Postavlja širinu između nogu trake
except Exception as e:

```

```
print("Greška pri promjeni ograničenja nogu:", e) # Ispisuje
poruku o grešci kod postavljanja ograničenja za noge trake
raise SystemExit # Prekida izvršenje programa
```

Promjena ograničenja između nogu kako bi uvijek bile u kutu.

```
# Pronalazimo postojeći izvučeni oblik U profila (Pad001)
pad001 = doc.getObject('Pad001') # Dohvaća objekt Pad001
```

Nakon postavljanja dimenzija u skicama, prelazi se na postavljanje 3D dijelova.

Prvo se dohvaća izvučeni oblik U profila pod nazivom „Pad001“.

```
# Postavlja dimenzije izvučenog oblika U profila
try:
pad001.Length = duljina_trake # Postavlja duljinu U profila
na duljinu trake
pad001.Midplane = True # Omogućava simetrično izvlačenje
except Exception as e:
print("Greška pri postavljanju nosača:", e) # Ispisuje
poruku o grešci
raise SystemExit # Prekida izvršenje programa
```

Postavlja se duljina izvučenog dijela na duljinu trake definirana varijablom `duljina_trake`. Kako bi izvučeni oblik bio izvučen simetrično na obje strane, dodaje se „`pad001.Midplane = True`“.

```
# Pronalazi izvučeni oblik nogu
pad005 = doc.getObject('Pad005') # Dohvaća objekt Pad002
```

Nakon izrade izvučenog oblika U profila postavlja se izvučeni oblik nogu. Dohvaća se „Pad005“ kako bi se postavila visinu trake.

```
# Postavljanje dimenzija nogu
try:
pad005.Length = visina_trake # Postavlja duljinu nogu na
visinu trake
except Exception as e:
print("Greška pri postavljanju visine trake:", e) # Ispisuje
poruku o grešci kod postavljanja visine trake
raise SystemExit # Prekida izvršenje programa
```

Kako bi visina trake bila prema korisnikovim željama, postavlja se dužina izvučenog dijela na vrijednost varijable „`visina_trake`“.

```
# Pronalazi rupe na U profilima
pocket = doc.getObject('Pocket') # Dohvaća objekt Pocket
```

Nakon postavljanja dimenzija trake, kreće se sa postavljanjem valjaka. Prvo dohvaćamo rupe „Pocket“ u koje idu valjci.

```

# Postavljanje dimenzije rupa
try:
pocket.Length = sirina_trake # Postavlja duljinu rupe na
širinu trake
pocket.Midplane = True # Omogućava simetrično izvlačenje
except Exception as e:
print("Greška pri postavljanju srednje rupe na nosaču:", e)
# Ispisuje poruku o grešci kod postavljanja duljine rupa
raise SystemExit # Prekida izvršenje programa

```

Rupe prolaze kroz oba U profila, pa se postavlja izrada rupa simetrično sa skicom na duljini koja odgovara vrijednosti varijable „sirina_trake“

```

# Pronalazi linearne uzorke lijevih rupa na nosaču
linear_pattern = doc.getObject('LinearPattern') # Dohvaća
objekt LinearPattern

```

Kako bi rupe bile raspoređene po cijelom U profilu, mora se postaviti linearni uzorak, koji se zato dohvaća iz modela.

```

# Računanje broja ponavljanja
ponavljanja = math.floor((duljina_trake / 2) / 110) # Računa
broj ponavljanja i zaokružuje na cijeli broj

```

Ovaj dio koda izračunava potreban broj ponavljanja duž nosača. Zbog ob+vog dijela trebao je modul „math“, jer se koristi funkcija koja izračunava i zaokružuje broj ponavljanja.

```

# Postavljanje parametara za linearne uzorke lijevih rupa
try:
linear_pattern.Length = ponavljanja * 110 # Postavljanje
duljine linearnih uzoraka
linear_pattern.Occurrences = ponavljanja # Postavljanje
broja ponavljanja
except Exception as e:
print("Greška pri postavljanju lijevih rupa na nosaču:", e)
# Ispisuje poruku o grešci kod postavljanja linearnih
uzoraka rupa lijevo
raise SystemExit # Prekida izvršenje programa

```

Za postavljanje linearnog uzorka potrebno je odrediti duljinu po kojoj se uzorak proteže, što u ovom slučaju predstavlja razmak između valjaka (110 mm) pomnožen s brojem ponavljanja, te broj ponavljanja uzoraka koji će biti raspoređeni po toj duljini.

```

# Pronalazi linearne uzorke desnih rupa na nosaču
linear_pattern01 = doc.getObject('LinearPattern01') #
Dohvaća objekt LinearPattern01

```



```

# Postavljanje parametara za linearne uzorke desnih rupa
try:
linear_pattern001.Length = ponavljanja * 110 # Postavlja
duljinu linearnog uzorka na ponavljanja * 110 mm
linear_pattern001.Occurrences = ponavljanja # Postavlja broj
ponavljanja

except Exception as e:
print("Greška pri postavljanju desnih rupa na nosaču:", e) #
Ispisuje poruku o grešci kod postavljanja linearnih uzoraka
rupa desno
raise SystemExit # Prekida izvršenje programa

```

Budući da je jednaki linearni uzorak i prema drugoj strani, postavljamo ga na isti način kao i prethodno.

```

# Pronalazi linearne uzorke desnih nosača za valjke na
desnom nosaču
linear_pattern002 = doc.getObject('LinearPattern002') #
Dohvaća objekt LinearPattern002

```

Nakon što su postavljene rupe u U profilima, potrebno je postaviti i linearne uzorke nosača koji idu u te rupe. U nastavku se dohvaćaju jedan po jedan linearni uzorak tih nosača i postavljaju im se vrijednosti kao i rupama.

```

# Postavljanje parametara desnih nosača za valjke na desnom
nosaču
try:
linear_pattern002.Length = ponavljanja * 110 # Postavlja
duljinu linearnog uzorka
linear_pattern002.Occurrences = ponavljanja # Postavlja broj
ponavljanja
except Exception as e:
print("Greška pri postavljanju desnih nosača za valjke na
desnom nosaču:", e) # Ispisuje poruku o grešci
raise SystemExit # Prekida izvršenje programa

```

```

# Pronalazi linearne uzorke lijevih nosača za valjke na
desnom nosaču
linear_pattern003 = doc.getObject('LinearPattern003') #
Dohvaća objekt LinearPattern003
# Postavljanje parametara lijevih nosača za valjke na desnom
nosaču
try:
linear_pattern003.Length = ponavljanja * 110 # Postavlja
duljinu linearnog uzorka
linear_pattern003.Occurrences = ponavljanja # Postavlja broj
ponavljanja
except Exception as e:
print("Greška pri postavljanju lijevih nosača za valjke na
desnom nosaču:", e) # Ispisuje poruku o grešci
raise SystemExit # Prekida izvršenje programa

```

```

# Pronalazi linearne uzorke desnih nosača za valjke na
lijevom nosaču
linear_pattern004 = doc.getObject('LinearPattern004') #
Dohvaća objekt LinearPattern004

# Postavljanje parametara desnih nosača za valjke na lijevom
nosaču
try:
linear_pattern004.Length = ponavljanja * 110 # Postavlja
duljinu linearnog uzorka
linear_pattern004.Occurrences = ponavljanja # Postavlja broj
ponavljanja
except Exception as e:
print("Greška pri postavljanju desnih nosača za valjke na
lijevom nosaču:", e) # Ispisuje poruku o grešci
raise SystemExit # Prekida izvršenje programa

# Pronalazi linearne uzorke lijevih nosača za valjke na
lijevom nosaču
linear_pattern005 = doc.getObject('LinearPattern005') #
Dohvaća objekt LinearPattern005

# Postavljanje parametara lijevih nosača za valjke na
lijevom nosaču
try:
linear_pattern005.Length = ponavljanja * 110 # Postavlja
duljinu linearnog
linear_pattern005.Occurrences = ponavljanja # Postavlja broj
ponavljanja
except Exception as e:
print("Greška pri postavljanju lijevih nosača za valjke na
lijevom nosaču:", e) # Ispisuje poruku o grešci
raise SystemExit # Prekida izvršenje programa

# Pronalazi izvučeni oblik valjka
pad004 = doc.getObject('Pad004') # Dohvaća objekt Pad004

```

Kada su postavljeni svi nosači na oba U profila, prelazi se na postavljanje izvučenog oblika valjka, zbog čega se dohvaća „Pad004“.

```

# Postavljanje širinu valjaka
try:
pad004.Length = sirina_trake - 110 # Postavlja šrinu valjaka
pad004.Midplane = True # Omogućava simetrično izvlačenje
except Exception as e:
print("Greška pri postavljanju srednjeg valjka:", e) #
Ispisuje poruku o grešci
raise SystemExit # Prekida izvršenje programa

```

Duljina izvučenog oblika postavlja se simetrično na duljinu između nosača u rupama (varijabla „sirina_trake“-100 mm od širine U profila – 10 mm koliko su nosači izvan U profila).

```
# Pronalazi linearne uzorke desnih valjaka
linear_pattern006 = doc.getObject('LinearPattern006') #
Dohvaća objekt LinearPattern006

# Postavljanje parametara desnih valjaka
try:
linear_pattern006.Length = ponavljanja * 110 # Postavlja
duljinu linearnog
linear_pattern006.Occurrences = ponavljanja # Postavlja broj
ponavljanja
except Exception as e:
print("Greška pri postavljanju desnih valjaka:", e) #
Ispisuje poruku o grešci
raise SystemExit # Prekida izvršenje programa

# Pronalazi linearne uzorke lijevih valjaka
linear_pattern007 = doc.getObject('LinearPattern007') #
Dohvaća objekt LinearPattern007

# Postavljanje parametara lijevih valjaka
try:
linear_pattern007.Length = ponavljanja * 110 # Postavlja
duljinu linearnog uzorka
linear_pattern007.Occurrences = ponavljanja # Postavlja broj
ponavljanja
except Exception as e:
print("Greška pri postavljanju lijevih valjaka:", e) #
Ispisuje poruku o grešci
raise SystemExit # Prekida izvršenje programa
```

Postavljanje linearnih uzoraka jednako je kao kod rupa i nosača u tim rupama.

```
# Ponovno računanje dokumenta
doc.recompute()
# Spremanje dokumenta s novim izmjenama
doc.saveAs('C:/Users/Administrator/Desktop/Diplomski/pokretn
a_traka.FCStd') # Zamijeniti s odgovarajućom putanjom
```

Kao i u prethodnom programu imamo ponovno računanje modela te spremanje istog.

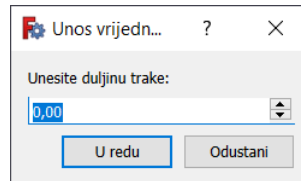
```
# Prikaz rezultata
Gui.activeDocument().activeView().viewAxometric() #
Postavlja pogled na aksonometrijski prikaz
Gui.SendMsgToActiveView("ViewFit") # Prilagođava prikaz
veličini modela
```

Na kraju se ponovno postavlja prikaz kako bi model bio najbolje prikazan.

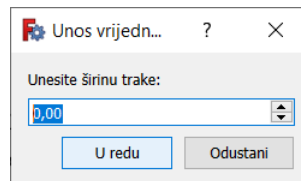
4. Primjeri

U nastavku će biti prikazani primjeri optimiziranih prototipa laboratorijskog stola i pokretne trake, kojima je preko Python programa promijenjena dimenzija ovisno o korisnikovim željama.

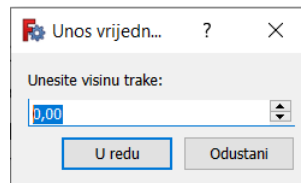
Nakon pokretanja programa korisniku se otvaraju prozori kao na slijedećim slikama, gdje upisuje željene dimenzije laboratorijskog stola ili trake.



Slika 4.1 Prvi prozor u kojem se upisuje duljina trake



Slika 4.2 Drugi prozor u kojem se upisuje širina trake

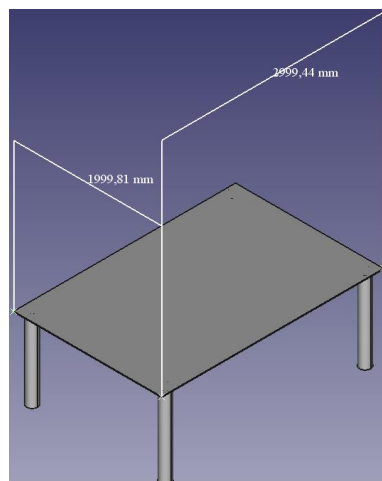


Slika 4.3 Treći prozor u kojem se upisuje visina trake

Nakon trećeg prozora program počinje preračunavati model i prikazuje nam novi optimizirani model.

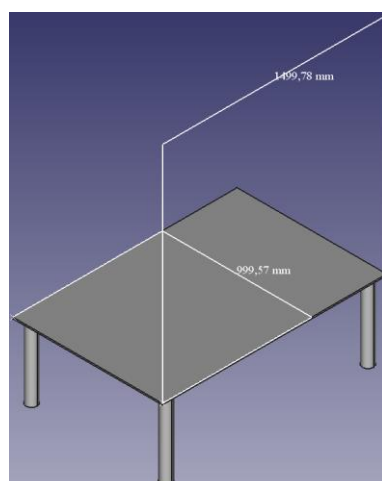
4.1. Primjeri optimiziranih prototipa laboratorijskih stolova

Kao primjer na slici 4.4 su opisane dimenzije za duljinu stola od 3000 mm, širinu 2000 mm i visinu 1100 mm.



Slika 4.4 Primjer stola sa dimenzijama 3000 mm x 2000 mm x 1100 mm

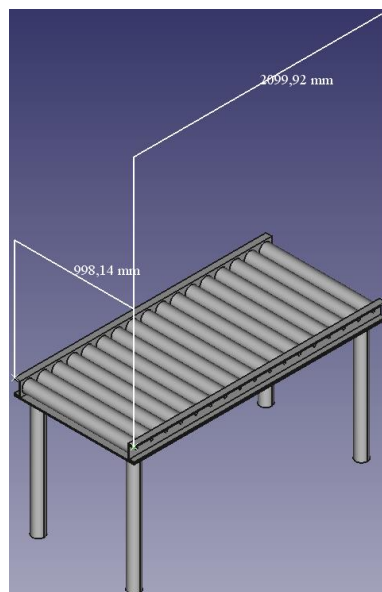
Kao primjer na slici 4.5 su opisane dimenzije za duljinu stola od 1500 mm, širinu 1000 mm i visinu 500 mm.



Slika 4.5 Primjer stola sa dimenzijama 1500 mm x 1000 mm x 500 mm

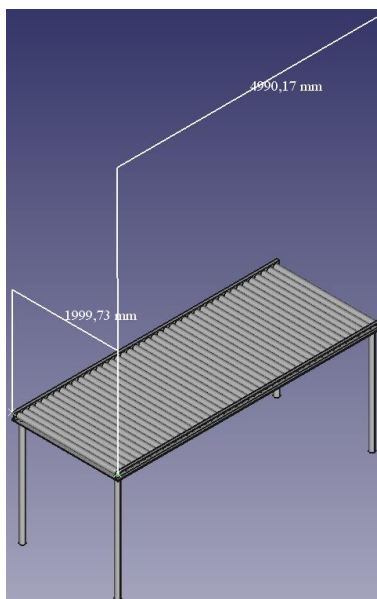
4.2. Primjeri optimiziranih prototipa pokretnih traka

Kao primjer na slici 4.6 su opisane dimenzije za duljinu trake od 2100 mm, širinu 1000 mm i visinu 1000 mm.



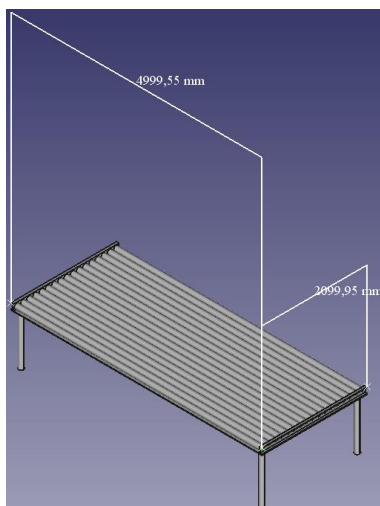
Slika 4.6 Primjer pokretne trake sa dimenzijama 2100 mm x 1000 mm x 1000 mm

Kao primjer na slici 4.7 su opisane dimenzije za duljinu trake od 5000 mm, širinu 2000 mm i visinu 2000 mm.



Slika 4.7 Primjer pokretne trake sa dimenzijama 5000 mm x 2000 mm x 2000 mm

Kao primjer na slici 4.8 su opisane dimenzije za duljinu trake od 2100 mm, širinu 5000 mm i visinu 1000 mm.



Slika 4.8 Primjer pokretne trake sa dimenzijama 2100 mm x 5000 mm x 1000 mm

5. Zaključak

Ovaj diplomski rad ističe važnost primjene CAD alata otvorenog koda, s posebnim naglaskom na FreeCAD, kao praktično i pristupačno rješenje za inženjerske projekte, posebno one u fazi prototipiranja. Kroz analizu i primjenu FreeCAD-a u izradi laboratorijskih stolova i pokretnih traka, pokazano je da otvoreni softver može konkurirati komercijalnim CAD alatima pružajući slične funkcionalnosti, ali bez visokih troškova licenciranja. U radu je također provedena usporedba FreeCAD-a s alatima poput Blendera, LibreCAD-a, OpenSCAD-a i SolveSpace-a, koji nude različite prednosti ovisno o inženjerskim zahtjevima. FreeCAD se ističe fleksibilnošću i širokom primjenjivošću, što ga čini učinkovitim alatom za dizajn i razvoj prototipova.

Jedna od ključnih prednosti FreeCAD-a jest njegova mogućnost parametarskog modeliranja, što omogućava jednostavnu prilagodbu modela u skladu sa specifičnim potrebama korisnika. Korištenje Pythona i makronaredbi dodatno je unaprijedilo proces, automatizirajući ponavljajuće zadatke i omogućavajući bržu optimizaciju modela. Na primjeru laboratorijskog stola i pokretne trake, prikazana je fleksibilnost u promjeni dimenzija i parametara modela. Python skripte omogućile su prilagodbu dijelova poput nogu i površine stola, te elemenata trake, poput U profila i valjaka, čime je postignut visok stupanj prilagodljivosti u dizajnu.

Detaljnim opisom izrade modela i programa za optimizaciju prototipa, u radu su prikazane mogućnosti FreeCAD-a, posebno u pogledu upravljanja gotovo svim aspektima modela putem makronaredbi. Ovi primjeri jasno pokazuju da se složeni modeli mogu prilagoditi i optimizirati pomoću automatiziranih skripti, što štedi vrijeme i povećava preciznost dizajna.

Primjeri optimizacije prototipa pokazuju da je moguće brzo i učinkovito kreirati funkcionalne modele s mogućnošću trenutne promjene parametara. Na taj način smanjuje se vrijeme potrebno za izradu i prilagodbu modela, što je ključno u fazi prototipiranja industrijskih proizvoda. Ova metoda može se posebno primijeniti u manjim i srednjim poduzećima, koja traže kvalitetna, ali pristupačna softverska rješenja za razvoj proizvoda.

U konačnici, rad detaljno opisuje kako CAD alati otvorenog koda, u kombinaciji s Python skriptiranjem, mogu značajno doprinijeti razvoju industrijskih prototipova.

Ovo pristupačno i moćno rješenje može olakšati rad inženjerima, arhitektima i dizajnerima, omogućujući im stvaranje i prilagodbu složenih modela uz minimalna ulaganja u softver. Očekuje se da će daljnji razvoj i poboljšanja u FreeCAD-u dodatno povećati njegovu upotrebljivost u inženjerskoj praksi, posebno u kontekstu otvorenog koda.

6. Literatura

- [1] izv.prof.dr.sc. Sven Maričić, Predavanja iz kolegija Prototipno 3D modeliranje, Sveučilište sjever, 2023./2024.
- [2] mag. ing. mech. Siniša Švoger, Predavanja iz kolegija 3D oblikovanje, Sveučilište Sjever, 2020./2021.
- [3] dr. sc. Milan Kljajin, red. prof. u tr. zv., Predavanja iz kolegija Tehnička dokumentacija, Sveučilište Sjever, 2019./2020.
- [4] Y. Van Havre, FreeCAD Manual, Geeknet, Inc., 2010.
- [5] <https://nel-media.com/blog/3d-programi/blender-program-za-izradu-3d-modela>, dostupno 6.9.2024.
- [6] <https://librecad.en.download.it/> , dostupno 6.9.2024.
- [7] <https://www.nihlibrary.nih.gov/resources/tools/openscad> , dostupno 6.9.2024.
- [8] <https://solvespace.github.io/solvespace-web/tutorial.html> , dostupno 6.9.2024.
- [9] <https://www.freecad.org/downloads.php>, dostupno 6.8.2024.
- [10] https://wiki.freecad.org/Python_scripting_tutorial, dostupno 6.9.2024.
- [11] <https://wiki.freecad.org/Macros>, dostupno 6.9.2024.
- [12] <https://github.com/FreeCAD/FreeCAD-macros>, dostupno 6.9.2024.
- [13] https://github.com/FreeCAD/FreeCAD-documentation/blob/main/wiki/FreeCAD_Scripting_Basics.md , dostupno 6.9.2024.
- [14] D. Grgić, Vizualizacija i priprema simulacija za dualsphysics rješavač, Diplomski rad, Sveučilište u Rijeci, Rijeka, 2023.
- [15] https://www.pkmachinery.com/spare-parts/conveyor-roller.html?gad_source=1&gclid=Cj0KCQjwi5q3BhCiARIsAJCfuZnljFW49xXzRX1DgAkiaXLX9qDyBz9y3F_nXUQa7pKS38MWWCmvexEaAsv9EALw_wcB , dostupno 6.9.2024.

7. Popis slika

Slika 2.1 Početni prozor instalacijskog programa	6
Slika 2.2 Uvjeti licence	7
Slika 2.3 Odabir korisnika programa	7
Slika 2.4 Odabir mjesta preuzimanja na disku	7
Slika 2.5 Odabir komponenti koje želimo preuzeti	8
Slika 2.6 Završni prozor instalacijskog programa	8
Slika 2.7 Ikona FreeCAD programa	8
Slika 3.1 Početni prozor FreeCAD programa	9
Slika 3.2 Radno okruženje	10
Slika 3.3 Odabir modula za crtanje.....	10
Slika 3.4 Početak skiciranja	11
Slika 3.5 Alati koji se koriste za skiciranje.....	11
Slika 3.6 Poruke solvera, ograničenja i elementi skice.....	11
Slika 3.7 Poruka solvera o suvišnim ograničenjima	12
Slika 3.8 Poruka solvera o potpuno ograničenoj skici.....	12
Slika 3.9 Alati za kreiranje 3D modela.....	12
Slika 3.10 Primjer 3D modela i parametara izvučenog oblika	13
Slika 3.11 Prikaz odabira plohe	16
Slika 3.12 Alat za crtanje pravokutnika.....	16
Slika 3.13 Potpuna skica postolja	17
Slika 3.14 Alat za izradu izvučenog oblika	17
Slika 3.15 Izvučeni oblik postolja i postavljeni parametri.....	17
Slika 3.16 Skica U nosača valjaka	18
Slika 3.17 Alat za kreiranje simetrije.....	18
Slika 3.18 Skica nakon kreiranja simetričnog U nosača.....	18
Slika 3.19 Potpuno ograničena skica nosača	18
Slika 3.20 Izvučeni oblik i parametri U nosača	19
Slika 3.21 Skica utora u koje će se postavljati valjci.....	19
Slika 3.22 Alat za kreiranje utora	19
Slika 3.23 Parametri za kreiranje utora.....	20
Slika 3.24 Alat za kreiranje linearnog uzorka.....	20

Slika 3.25 Prikaz izrađenog „Pocketa“	20
Slika 3.26 Linearni uzorak i njegovi parametri	21
Slika 3.27 Linearni uzorak s druge strane i njegovi parametri	21
Slika 3.28 Odabir plohe na kojoj želimo raditi skicu	21
Slika 3.29 Skica nosača valjka koji idu u utore	22
Slika 3.30 Greška kod kreiranja izvučenog dijela	22
Slika 3.31 Izvučeni oblik i njegovi parametri.....	22
Slika 3.32 Prikaz nosača s lijeve i desne strane U nosača te njihovi parametri.....	22
Slika 3.33 Prikaz nosača valjaka na oba U nosača	23
Slika 3.34 Skica valjaka.....	23
Slika 3.35 Greška kod izrade valjaka.....	23
Slika 3.36 Izvučeni oblik i parametri za izradu osnovnog valjka.....	23
Slika 3.37 Linearni uzorak i parametri za izradu valjaka	24
Slika 3.38 Pokretna traka bez nogu	24
Slika 3.39 Skica oslonca ili nogu.....	24
Slika 3.40 Parametri za izradu nogu i gotova pokretna traka	25
Slika 3.41 Putanja za otvaranje Python konzole.....	26
Slika 3.42 Python konzola na dnu	26
Slika 3.43 Otvaranja prozora za rad s makronaredbama	28
Slika 3.44 Prozor za rad s makronaredbama.....	28
Slika 3.45 Izvršavanje makronaredbe	28
Slika 3.46 Pregled izvještaja na dnu	28
Slika 3.47 Prototip laboratorijskog stola.....	29
Slika 3.48 Kopiranje lokacije modela.....	30
Slika 3.49 Naziv skice	31
Slika 3.50 Ograničenja ploče stola	31
Slika 3.51 Ograničenja nogu stola	32
Slika 3.52 Ograničenja unutar skice "Sketch"	35
Slika 3.53 Ograničenja koja želimo mijenjati.....	35
Slika 3.54 Ograničenja u "Sketch006" koja se mijenjaju	36
Slika 4.1 Prvi prozor u kojem se upisuje duljina trake	42
Slika 4.2 Drugi prozor u kojem se upisuje širina trake.....	42

Slika 4.3 Treći prozor u kojem se upisuje visina trake	42
Slika 4.4 Primjer stola sa dimenzijama 3000 mm x 2000 mm x 1100 mm	43
Slika 4.5 Primjer stola sa dimenzijama 1500 mm x 1000 mm x 500 mm	43
Slika 4.6 Primjer pokretne trake sa dimenzijama 2100 mm x 1000 mm x 1000 mm	44
Slika 4.7 Primjer pokretne trake sa dimenzijama 5000 mm x 2000 mm x 2000 mm	44
Slika 4.8 Primjer pokretne trake sa dimenzijama 2100 mm x 5000 mm x 1000 mm	45

8. Popis tablica

Tablica 1. Usporedba FreeCAD-a i ostalih alata	4
Tablica 2. Ograničenja u FreeCAD-u	15

IZJAVA O AUTORSTVU

Završni/diplomski/specijalistički rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, DANIŠEL PAVEŠIĆ (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/~~ko~~ ~~završnog/diplomskog/specijalističkog~~ (obrisati nepotrebno) rada pod naslovom PROTOTIPNA OPTIMIZACIJA UZ POMOĆ CAD (upisati naslov) te da u ALATA OTVORENOG KODA navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

(vlastoručni potpis)

Sukladno članku 58., 59. i 61. Zakona o visokom obrazovanju i znanstvenoj djelatnosti završne/diplomske/specijalističke radove sveučilišta su dužna objaviti u roku od 30 dana od dana obrane na nacionalnom repozitoriju odnosno repozitoriju visokog učilišta.

Sukladno članku 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje umjetnička djelatnost i visoko obrazovanje.