

Gimnazija Jurija Vege Idrija

Študentovska ulica 16

5280 Idrija

RAZISKOVALNA NALOGA

POSTAVITEV SISTEMA ENAČB ZA ANALIZO ACKERMANN SISTEMA V KARTINGU

Tehnika in tehnologija

Avtor:

Tim Gorenc

Mentor:

Jurij Drev, mag. ing. str.

Vrhnika, 2021

POVZETEK

Cilj raziskovalne naloge je postaviti delajoč sistem enačb, ki nam poda natančne rezultate premika konice premnika na gokartu. Ovire pri izračunu povzročajo številni koti, ki sodelujejo pri samem delovanju sistema. V sklopu raziskovalne naloge bi rad ugotovil, ali je mogoče s pomočjo CAD programa izdelati delajoč in dovolj natančen 3D model gokarta in njegovih komponent ter ali je mogoče izdelati dovolj natančen sistem enačb za izračun premika premnika glede na kot zavoja. Pri izdelavi in sestavi posameznih komponent ugotovim, da je mogoče izdelati pravilen in delajoč 3D model gokarta. S pomočjo 3D modela sem prav tako izdelal sistem enačb, ki nam ponazoriti delovanje premnika v zavoju. Glede na vstavljenе parametre variirajo tudi končni rezultati, kar je bil tudi moj cilj.

ABSTRACT

The goal of this research paper to establish a working system of equations that give us an accurate representation of the movement of the stub axle on the kart. The obstacle with calculating the movement are the numerous angles used for the system to function. In the context of the paper, I would like to find out if with the help of CAD programme, I can make a fully working and detailed enough 3D model of the kart and its components and if it is possible to produce a system of equations accurate enough to calculate the movement of the tip of the stub axle considering a turning angle. By modelling and assembling the parts I conclude that it is possible to make a valid and working 3D model of the kart. With the help of the 3D model, I also present the system of equations that illustrate the activity of the stub axle when cornering. Along with the entering parameters, the final results also change, which was my goal.

ZAHVALA

Rad bi se zahvalil podjetju Kart Republic, ki me je po potrebi priskrbelo z določenimi merami in parametri komponent za natančnejše rezultate.

Zahvala gre tudi mentorju, profesorju Juriju Drevu, mag. ing. str. in vsem, ki so me tekom naloge podpirali in spodbujali.

Za konec bi se rad zahvalil tudi profesorici slovenščine Silvi Uršič, za lektoriranje naloge in pa profesorici angleščine Klavdiji Stepančič za pregled angleškega povzetka.

Kazalo vsebine

| | | |
|--------------|--|----|
| <u>1</u> | <u>UVOD</u> | 7 |
| <u>1.1</u> | <u>Cilj naloge</u> | 7 |
| <u>1.2</u> | <u>Hipoteze</u> | 7 |
| <u>2</u> | <u>IZDELAVA 3D MODELA</u> | 8 |
| <u>2.1</u> | <u>Osnove "ackermann" sistema v kartingu</u> | 8 |
| <u>2.2</u> | <u>Izdelava modela</u> | 9 |
| <u>2.2.1</u> | <u>Izdelava okvirja</u> | 9 |
| <u>2.2.2</u> | <u>Izdelava komponent</u> | 10 |
| <u>2.3</u> | <u>Sestavljanje komponent</u> | 10 |
| <u>2.4</u> | <u>Ponastavitev parametrov</u> | 11 |
| <u>2.4.1</u> | <u>Zaostajanje koles</u> | 12 |
| <u>2.4.2</u> | <u>Stekanje koles</u> | 12 |
| <u>2.4.3</u> | <u>Previs koles</u> | 13 |
| <u>3</u> | <u>MATHCAD</u> | 14 |
| <u>3.1</u> | <u>Povezava CREO – MATHCAD</u> | 14 |
| <u>3.1.1</u> | <u>Izdelava skic na modelu</u> | 14 |
| <u>3.1.2</u> | <u>Povezava parametrov med programoma</u> | 15 |
| <u>3.2</u> | <u>Zasnova sistema enačb</u> | 16 |
| <u>3.2.1</u> | <u>Premik volanskega droga</u> | 16 |
| <u>3.2.2</u> | <u>Premik premnika</u> | 17 |
| <u>3.2.3</u> | <u>Pot konice premnika</u> | 18 |
| <u>3.3</u> | <u>Grafični prikaz rezultatov</u> | 20 |
| <u>4</u> | <u>ANALIZA REZULTATOV</u> | 22 |
| <u>4.1</u> | <u>Izdelava delujočega 3D modela</u> | 22 |
| <u>4.2</u> | <u>Matematični preračun premika premnika</u> | 22 |

| | | |
|----------|--------------------------------|----|
| <u>5</u> | <u>UGOTOVITVE IN ZAKLJUČEK</u> | 25 |
| <u>6</u> | <u>VIRI IN LITERATURA</u> | 26 |

Kazalo slik

| | |
|--|----|
| <u>Slika 1: Skica delovanja sistema ackermann</u> | 8 |
| <u>Slika 2: Prikaz ackermann sistema na gokartu</u> | 9 |
| <u>Slika 3: Zmodeliran okvir gokarta</u> | 10 |
| <u>Slika 4: Primerjava realne komponente (levo) in 3D modela (desno)</u> | 10 |
| <u>Slika 5: Sestavljen model gokarta</u> | 11 |
| <u>Slika 6: Prikaz nastavitevnih parametrov</u> | 11 |
| <u>Slika 7: Prikaz ponastavljanja zaostajanja koles</u> | 12 |
| <u>Slika 8: Prikaz ponastavljanja stekanja koles</u> | 12 |
| <u>Slika 9: Ekscenter</u> | 13 |
| <u>Slika 10: Prikaz ponastavljanja preivsa koles</u> | 13 |
| <u>Slika 11: Prikaz projecirane elipse</u> | 14 |
| <u>Slika 12: Povezava parametrov med programoma Creo (levo) in Mathcad (desno)</u> | 15 |
| <u>Slika 13: Obrazložitev prvega razloga nepravilnosti</u> | 16 |
| <u>Slika 14: Prikaz izračuna odklona izvrtine</u> | 16 |
| <u>Slika 15: Prikaz izračuna kota premika</u> | 18 |
| <u>Slika 16: Prikaz izračuna začetne pozicije po x-osi</u> | 19 |
| <u>Slika 17: Prikaz dobljene in potrebovane vrednosti</u> | 19 |
| <u>Slika 18: Prikaz rezultatov (levo) in dobljenega grafa (desno)</u> | 21 |

1 UVOD

V zadnjih 20 letih je moto šport glede same tehnologije razvoja izredno napredoval. Dandanes je že nekako samoumevno, da se komponente dirkalnega bolida najprej načrtujejo na računalniku, kjer se natančno analizirajo in tako odkrijejo možna mesta za izboljšave, in šele nato izdelajo. Tak način razvoja je neprimerno cenejši, časovno varčnejši in omogoča, da posamezne komponente izkoristimo kar se da najbolje.

1.1 Cilj naloge

Karting ima v večini zastarel način razvijanja svojih komponent in sicer s tako imenovano metodo poskusov in napak (trail and error). Časovno je ta metoda zelo zamudna in denarno potratna.

Moja ideja je, da ta način posodobim in za razvoj uporabim naprednejše metode. Sam sem se odločil za razvoj tako imenovanega "ackermann" sistema, ki je v karting svetu značilen po svoji težavnosti izračuna zaradi neštetih kotov in variacij nastavitev. Sistem sem se odločil razviti s pomočjo uporabe CAD in računskega programa.

1.2 Hipoteze

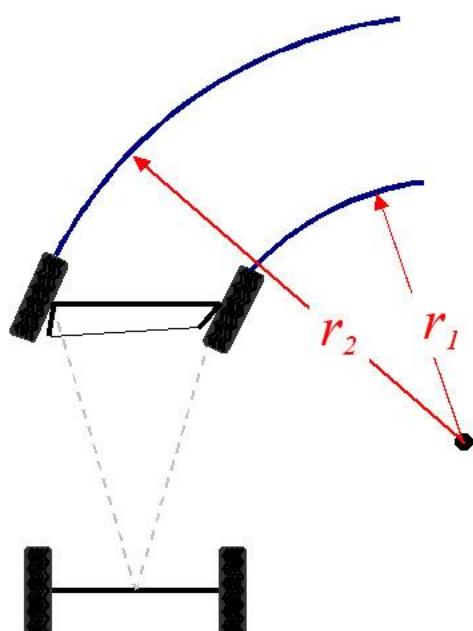
V sklopu naloge sem si zastavil dve hipotezi:

1. Mogoče je izdelati natančen in delujoč 3D model gokarta.
2. S pomočjo 3D modela je možno izpeljati matematične formule, ki nam podajo natančne rezultate.

2 IZDELAVA 3D MODELA

2.1 Osnove "ackermann" sistema v kartingu

Ackermann sistem je izumil Nemec Georg Lankensperger leta 1817, patentiral pa ga je Anglež Rudolph Ackermann, po katerem je sistem tudi dobil ime. Dandanes je sistem močno razširjen v avtomobilski industriji in prav tako se ta sistem že od samega

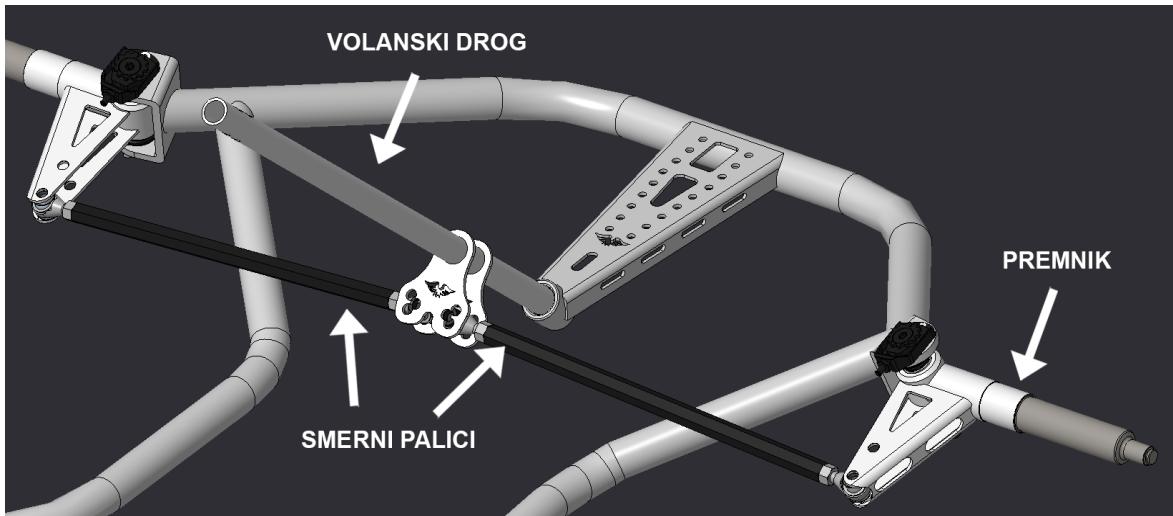


Slika 1: Skica delovanja sistema ackermann¹

začetka uporablja tudi v kartingu. Prednost sistema je, da kolo na zunanjji strani zavoja zavije manj, kot tisto na notranji strani. Ko avto vstopi v zavoj, se vsakemu izmed štirih koles izriše namišljena krožnica oz. pot, po kateri bo kolo potovalo med zavojem. Zaradi različnih pozicij koles pa je ta pot in posledično tudi radij namišljene krožnice za vsako kolo drugačna. Tako bo zunanje kolo prepotovalo veliko daljšo pot, oz. ima namišljena krožnica večji radij od tistega na notranji strani. Zato je ackermann sistem najbolj primeren za gokart, saj se tako stabilnost in vodljivost vozila močno poveča. ("Turn", 2012)

Princip med sistemoma v avtomobilu in gokartu je sicer podoben, vendar pa se način izvedbe razlikuje. Volan gokarta je pritrjen direktno na volanski drog, ki je v okvir gokarta pritrjen z očesnim ležajem. Na drogu sta privarjena dva ploščata dela jekla z različnimi pozicijami izvrtin. S smernimi palicami in dvema očesnima ležajema tako povežemo volanski drog in premnika, ki imata prav tako kot volanski drog več izvrtin za širšo možnost nastavitev. Premnik je vpet v okvir gokarta v tako imenovanih C-jih okvirja. Na vrhu le-tega se nahaja sistem za spremnjanje parametrov, kot so stekanje, previs in zaostajanje koles, ki bo kasneje v nalogi imelo zelo pomembno nalogo.

¹ <https://www.glue-it.com/wp/wp-content/uploads/2019/11/image002-1.jpg>



Slika 2: Prikaz ackermann sistema na gokartu

2.2 Izdelava modela

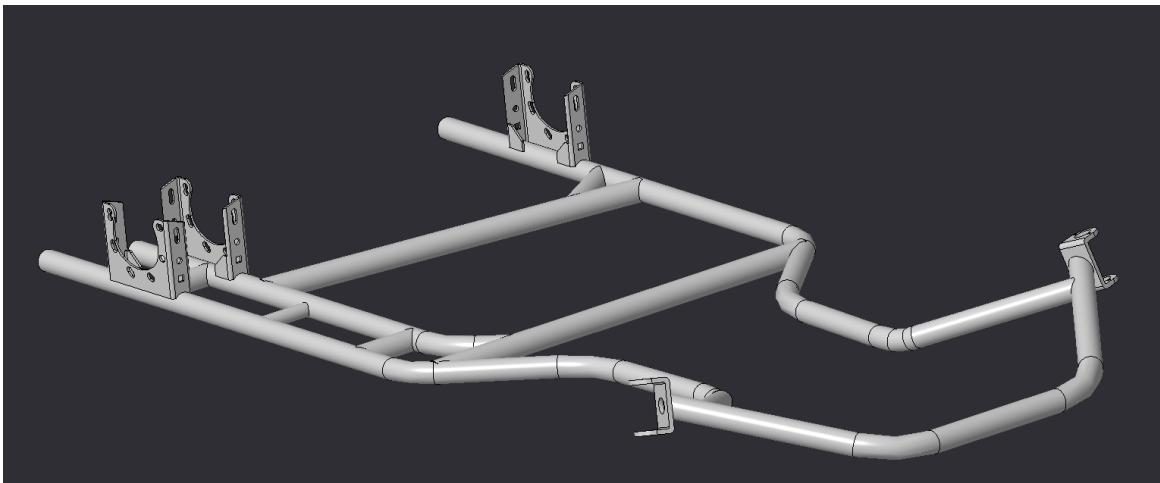
Za izdelavo vseh komponent sem uporabljal program CREO 6.0, saj sem z njim že imel predhodne izkušnje v šoli in sem kot dijak upravičen do brezplačne verzije programa Students Edition.

Sam model pa temelji na okvirju podjetja Kart Republic. Znamka je svetovno znana, njihovi okvirji pa se uporabljajo na profesionalnih tekmovanjih.

2.2.1 Izdelava okvirja

Najprej je bilo potrebno izdelati model okvirja, na katerega sem kasneje pritrdiril vse komponente. Pomembno je, kako okvir izdelam, saj že model okvirja lahko znatno vpliva na končne rezultate.

Prvi model sem zmodeliral kot vsako cev posebej in jih nato sestavil skupaj, kot bi to naredili v realnosti. Zaradi ekstremno majhnih prekrivanj cevi med seboj, na katere nisem mogel vplivati, ta način modeliranja posledično ni prišel v poštev. Po nasvetu strokovnjaka sem okvir zmodeliral še enkrat, tokrat pa okvir deluje kot enoten homogen kos. Na mojo analizo to nima vpliva, saj ne vključuje nobenih obremenitev in tako zadošča potrebam izračuna.



Slika 3: Zmodeliran okvir gokarta

2.2.2 Izdelava komponent

Na vrsto je prišlo modeliranje mnogih komponent. Pri tem sem moral biti še posebej natančen in pozoren. Nepozornost lahko zelo hitro pripelje do nepravilnih mer in posledično do prekrivanja oziroma razmika med komponentami. Taka napaka je lahko usodna za preračun, saj nam model posledično podaja nepravilne rezultate.



Slika 4: Primerjava realne komponente (levo) in 3D modela (desno)

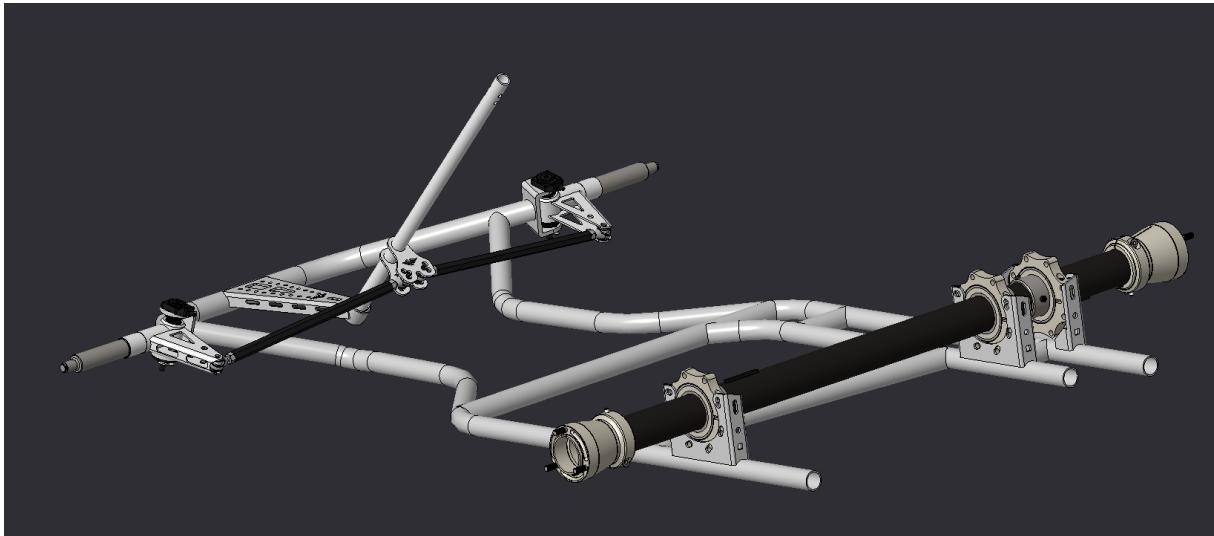
Ker mi delavnische risbe komponent niso dostopne, same komponente pa imam na voljo, sem vsako komponento posebej premeril z raznimi merili in te vrednosti vnesel v program, za kar se da natančen končni rezultat.

Še posebej težko je bilo zmodelirati premnika, saj sem moral upoštevati precej kotov, ki nastopajo, da celotni model deluje kot celota.

2.3 Sestavljanje komponent

Sestavljanje samih komponent je bil eden najzahtevnejših procesov. Komponente je bilo potrebno pravilno definirati v model, da se premikajo, kot želimo. Prav tako je bil zelo pomemben sam način njihovega sestavljanja in definiranja v model. Zaradi različnih omejitev jih je bilo potrebno sestavljati po neobičajnem vrstnem redu. V obratnem primeru bi prišlo do nepravilnega delovanja modela, kar bi imelo za posledico preveč napak. Zato sem po končanem sestavljanju s pomočjo programa

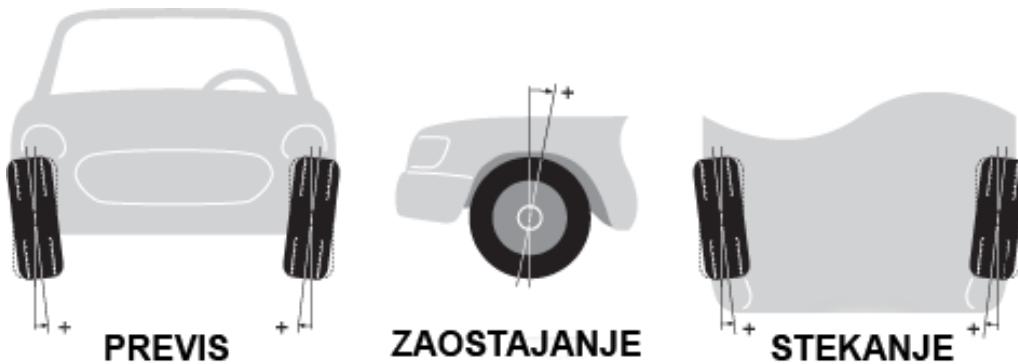
preveril pravilnost delovanja modela in tako odstranil manjša prekrivanja oziroma razmake celotnega modela.



Slika 5: Sestavljen model gokarta

2.4 Ponastavitev parametrov

Kot pri avtomobilu poznamo tudi v gokartu standardne nastavitev parametrov, ki veljajo za določene okvirje. Za okvir znamke Kart Republic so značilne nastavitev 4 mm skupnega stekanja koles, 8 mm skupnega previsa koles, za zaostajanje kolesa pa gledamo kos za nastavljanje previsa na C-ju okvirja. Standardna razdalja od minimalne pozicije naj bi bila 6 mm. Zato je bilo potrebno tudi nastavitev na modelu ponastaviti. Točnih vrednosti nisem mogel doseči zaradi omejitve natančnosti programa, vendar pa sem se jim kar se da približal.

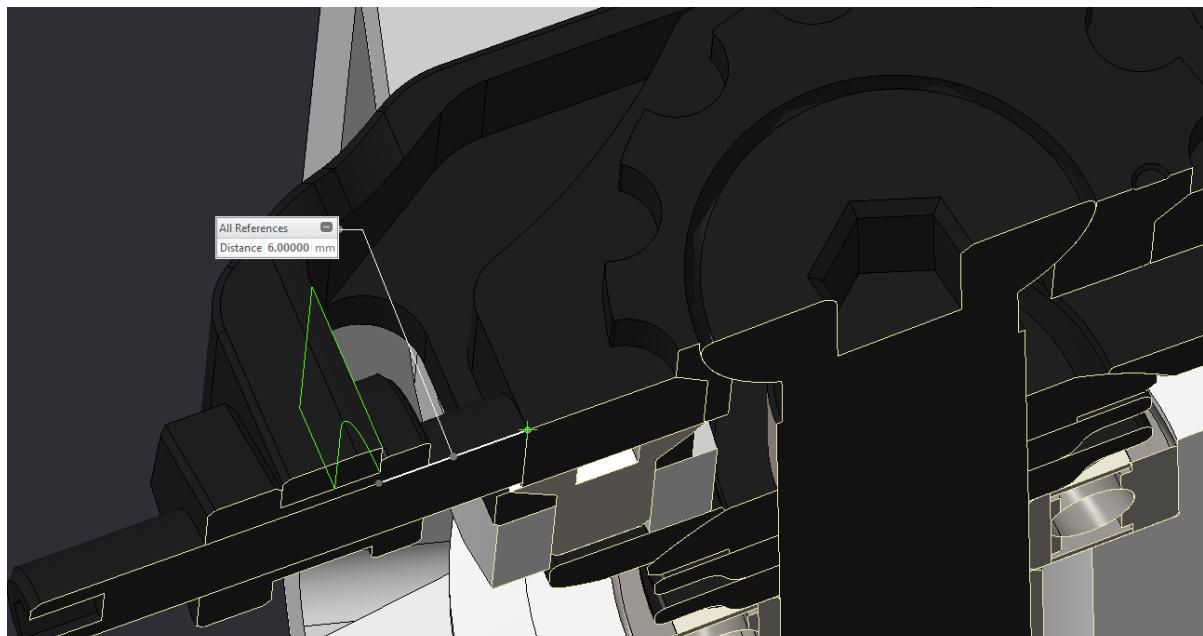


Slika 6: Prikaz nastavitvenih parametrov²

² <https://i.imgur.com/B50Zcyz.gif>

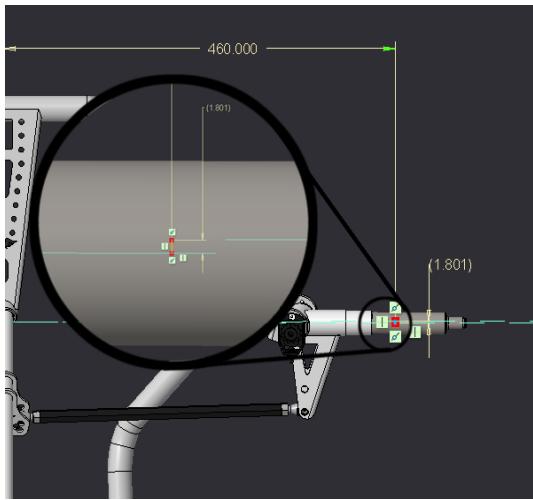
2.4.1 Zaostajanje koles

Nastavitev zaostajanja koles je bilo v primerjavi z drugimi parametri še najlažje ponastaviti. Ker sem model sestavil po pravilnem vrstnem redu, sem preprosto vnesel željeno razdaljo in dobil standardno pozicijo.



Slika 7: Prikaz ponastavljanja zaostajanja koles

2.4.2 Stekanje koles



Ponastavljanje nastavitev stekanja koles je bilo malo težje, saj sem moral ponastavljati globino očesnega ležaja v smerni palici in sem tako spremenjal stekanje koles. Po spremnjanju vrednosti sem preveril vrednost želenega parametra. Postopek sem velikokrat ponovil, dokler nisem dosegel najboljšega možnega rezultata, s katerim sem se približal standardni vrednosti.

Slika 8: Prikaz ponastavljanja stekanja koles

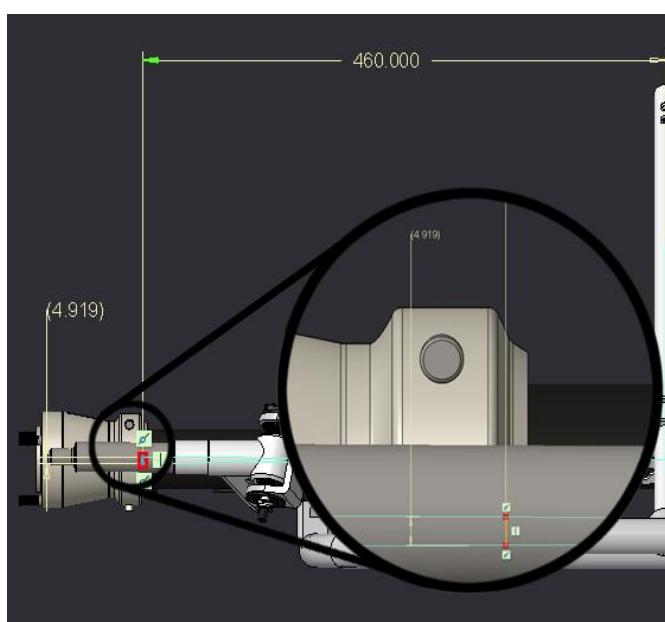
2.4.3 Previs koles

Najtežji izviv pa je bil, približati se nastavitevi previsa koles. Najprej sem zgornji ekscenter postavil na nevtralno pozicijo.

S tem sem si omogočil kasnejše spremjanje vrednosti po želji. Ker nimam na voljo delavnih risb komponent in okvirja, sem kote, ki sodelujejo pri zavijanju, izmeril s kotomerom. Seveda je posledično prišlo do merskih napak, ki pa sem jih sedaj lahko vsaj delno odstranil. Sprva sem izmeril vrednost previsa koles in ker je bila ta prevelika, sem enostavno spremenil velikost kota na modelu



Slika 9: Ekscenter



Slika 10: Prikaz ponastavljanja previsa koles

okvirja in premnika ter ponovno preveril parametre. Postopek sem ponavljal, dokler se ni vrednost previsa koles na modelu ujemala z želeno vrednostjo.

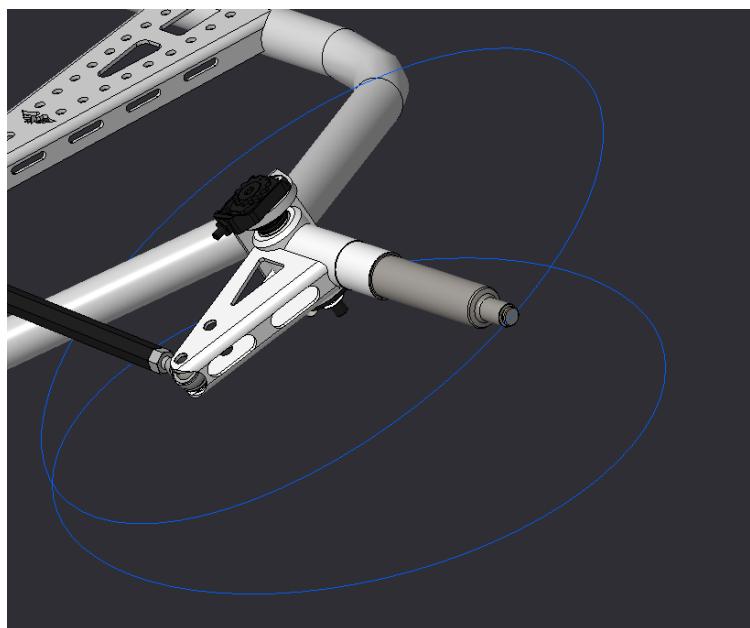
3 MATHCAD

Mathcad je program, ki nam omogoča zapisovanje raznoraznih sistemov enačb in je zmožen hitrega sprotnega računanja. Omogoča zapis vseh mogočih matematičnih operacij, poleg tega pa omogoča tudi risanje raznovrstnih grafov. Največja prednost programa je, da lahko enostavno povežemo parametre med programoma Creo in Mathcad.

3.1 Povezava CREO – MATHCAD

3.1.1 Izdelava skic na modelu

Preden sem lahko izdelal povezave želenih parametrov modela in programa Mathcad, sem na modelu izdelal tako imenovane skice. Te skice olajšajo povezavo parametrov med programoma, hkrati pa priomorejo k razumevanju samega delovanja modela in k lažji predstavi. Najtežje je bilo prikazati pot premika konice prenika. To sem storil tako, da sem najprej narisal skico pravokotno na os vrtenja prenika, saj se v tem pogledu ta premika po krožnici. Nato sem krožnico prenika projeciral na referenčno

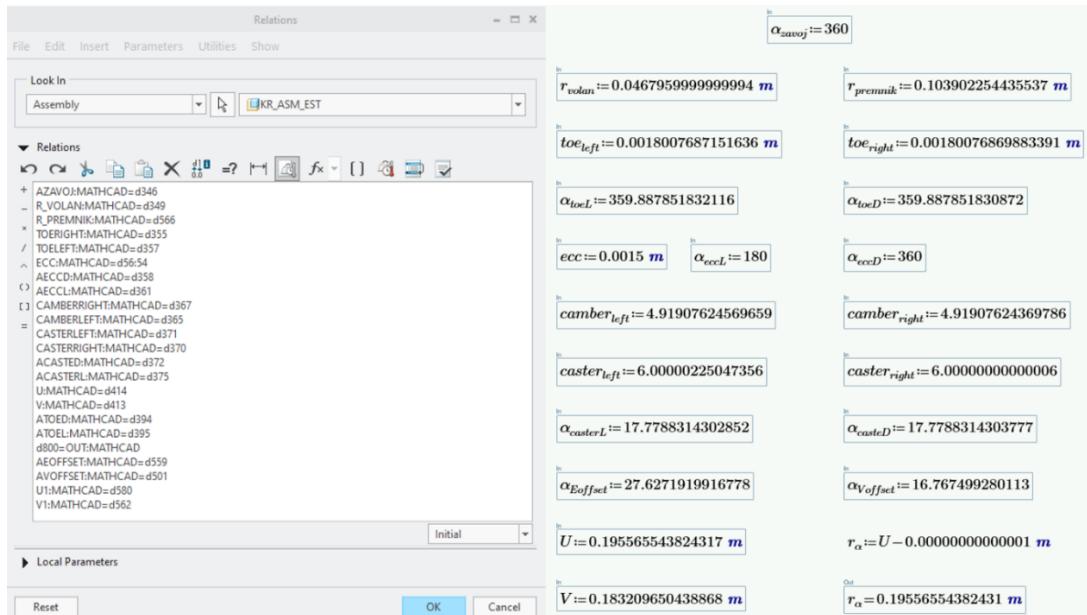


ravnino (ravnina, ki predstavlja tla) in tako dobil namesto krožnice elipso. Elipsa pa prikazuje dejansko pot premika prenika. Dobra stran tega je, da se pri spremembah enega od parametrov, avtomatsko spremeni tudi elipsa. Tako mi krožnice ni potrebno posodabljati. To velja tudi za vse ostale skice, ki sem jih naredil.

Slika 11: Prikaz projicirane elipse

3.1.2 Povezava parametrov med programoma Creo in Mathcad

Po izdelavi skic sem lahko ustvaril povezavo parametrov med programom modela Creo in programom, v katerem smo kasneje zasnovali sistem enačb Mathcad. Proses je dokaj preprost, saj imata programa to funkcijo že vgrajeno. V Mathcadu sem sprva označil, katere veličine želim, da Creo uvozi oz. katere količine želim, da Mathcad izvozi v Creo. Nato sem v programu Creo s pomočjo skic, ki sem jih naredil, določil razdalje in parametre, ki sem jih potreboval, in nato tem razdaljam in parametrom pripisal količine v Mathcadu. Vrednosti, ki sem jih povezal, predstavljajo raznorazne parametre, od kota zavoja z volanom do radijev in raznoraznih razdalij. Te mi bodo prišle prav predvsem pri računskem delu naloge, hkrati pa tudi pri sprotnem kontroliranju pravilnosti delovanja modela.

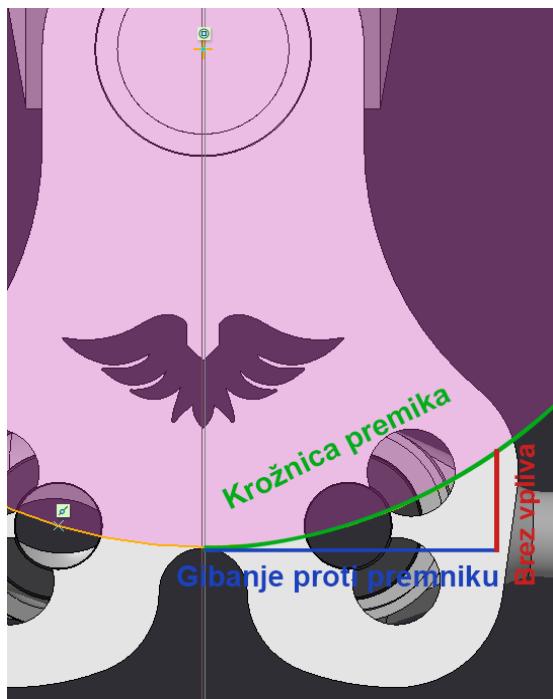


Slika 12: Povezava parametrov med programoma Creo (levo) in Mathcad (desno)

3.2 Zasnova sistema enačb

3.2.1 Premik volanskega droga

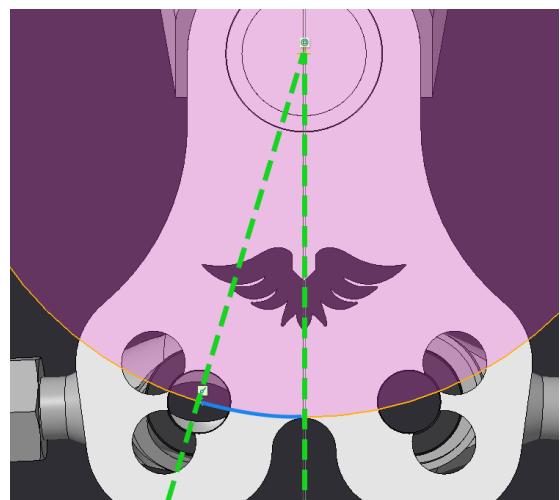
Z izračunom sem začel pri volanskem drogu. Tukaj imam podanih največ začetnih podatkov. Vem, da se volanski drog vrti okoli osi, na katero je vpet. Tako lahko izmerim radij krožnice, po kateri se premika središče izvrtine, kjer vpne smerno palico. Nato sem predpostavil, da je premik središča izvrtine na volanskem drogu enak premiku središča izvrtine, kjer vpne smerno palico na premnik. Tako sem uporabil splošno enačbo za izračun krožnega loka ($l = \frac{\pi \cdot r}{180^\circ} \cdot \alpha$).



Kasneje sem ugotovil, da ta enačba ni pravilna zaradi dveh razlogov. Prvi razlog je, da enačba ne upošteva smeri premika. S tem ko se palica premika po krožnici, pomeni, da potuje hkrati proti premniku (kar povzroči gibanje) in v smeri, ki ne vpliva na premik premnika. Na sliki lahko vidimo prikaz prvega razloga. Zelena barva prikazuje krožnico premika središča izvrtine na volanskem drogu. Modra barva prikazuje gibanje proti premniku, rdeča pa prikazuje gibanje brez nadaljnjega vpliva.

Slika 13: Obrazložitev prvega razloga nepravilnosti

Drugi razlog pa je, da prejšnja enačba ne prikaže vračanja zunanjega kolesa pri večjih kotih. Ko kot zavoja preseže približno 30° , se zunanje kolo začne vračati proti začetni poziciji, ker se izvrtina začne oddaljevati od premnika. Sprva je bilo potrebno izračunati pot odklona izvrtine glede na podan kot z navadno enačbo kotnih funkcij



Slika 14: Prikaz izračuna odklona izvrtine

$$l_{Voffset} = r_{volan} \cdot \sin \alpha_{Voffset}$$

$l_{Voffset}$ – pot odklona izvrtine

r_{volan} – sam radij izvrtine

$\alpha_{Voffset}$ – kot odklona izvrtine

Pri postopku računanja pa se enačbi za zunanje in notranje kolo razlikujeta. Za zunanje kolo uporabimo enačbo

$$l_{volan} = r_{volan} \cdot \sin (\alpha_{Voffset} + \varphi) - l_{Voffset}$$

l_{volan} – pot volanskega droga

φ – kot zavoja, ko voznik zavije z volanom

Za notranje kolo pa enačbo spremenimo tako, da dobimo

$$l_{volanI} = r_{volan} \cdot \sin (\varphi - \alpha_{Voffset}) + l_{Voffset}$$

l_{volanI} – pot volanskega droga za kolo na notranji strani zavoja

Ostale spremenljivke predstavljajo enake parametre.

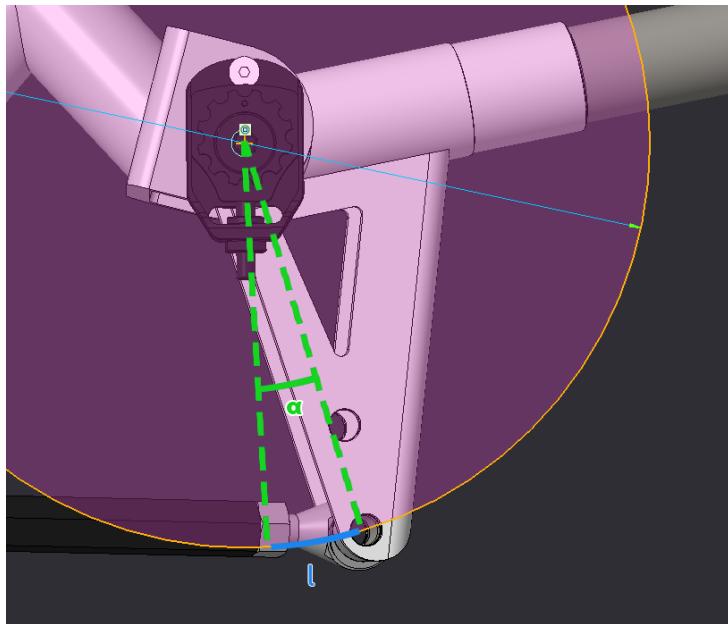
3.2.2 Premik premnika

Izračun kota premika premnika je dokaj preprost postopek. Predpostavil sem, da je premik središča izvrtine na premniku enak premiku izvrtine na volanskem drogu, po x-osi. Tega sem zračunal v zgornjem poglavju. Tako dobim obrnjeno enačbo krožnega loka

$$\alpha_{premnik} = \frac{360^\circ \cdot l_{volan}}{2 \cdot \pi \cdot r_{premnik}}$$

$\alpha_{premnik}$ – kot premika premnika

$r_{premnik}$ – radij izvrtine premnika



Slika 15: Prikaz izračuna kota premika

3.2.3 Pot konice premnika

Pomemben podatek pri izračunu prepotovane konice premnika je, da vemo, da konica potuje po elipsi (če gledamo model iz ptičje perspektive). Tako vemo, da lahko določimo pot premikanja in s kotom premnika tudi prepotovano pot.

Tako sem najprej izračunal obseg elipse z enačbo

$$ob = \pi \cdot (3(a + b) - \sqrt{(3 \cdot a + b) \cdot (a + 3 \cdot b)})$$

ob – obseg elipse

a – veliki radij

b – mali radij

s pomočjo velikega in malega radija, ki sem ga uvozil iz programa Creo.
("Perimeter", 2017)

Prvič sem pot izračunal tako, da sem preprosto uporabil enačbo za izračun krožnega loka ($l = \frac{\pi \cdot r}{180^\circ} \cdot \alpha$). Kasneje sem z nadaljnji raziskavami ugotovil, da ta enačba ne velja za elipse. Tako sem našel drug način za izračun poti po elipsi. Kar pa delo še oteži je, da elipsa ni vzporedna z začetno pozicijo premnika in tako ponovno dobimo odklon. Tega moramo upoštevati pri izračunu. Z enačbo

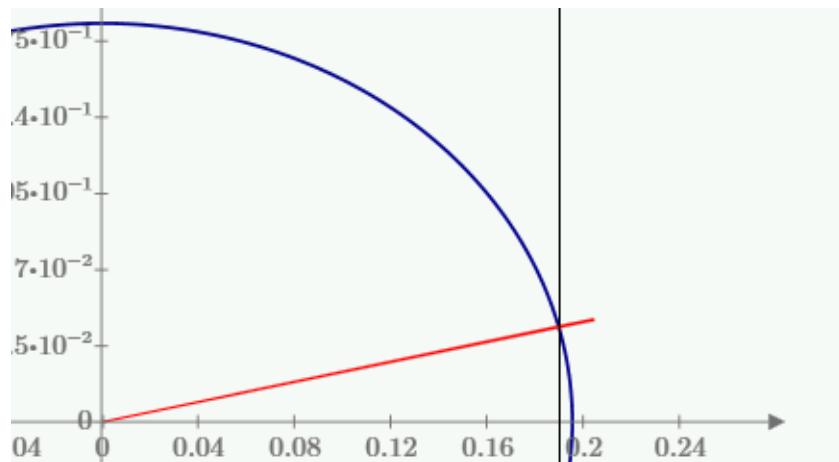
$$x_{toe} = \frac{a \cdot b \cdot \cos \alpha_{Eoffset}}{\sqrt{(b \cdot \cos \alpha_{Eoffset})^2 + (a \cdot \sin \alpha_{Eoffset})^2}}$$

x_{toe} – začetni odklon prenika na elipsi po x-osi

$\alpha_{Eoffset}$ – odklon elipse glede na začetno pozicijo prenika

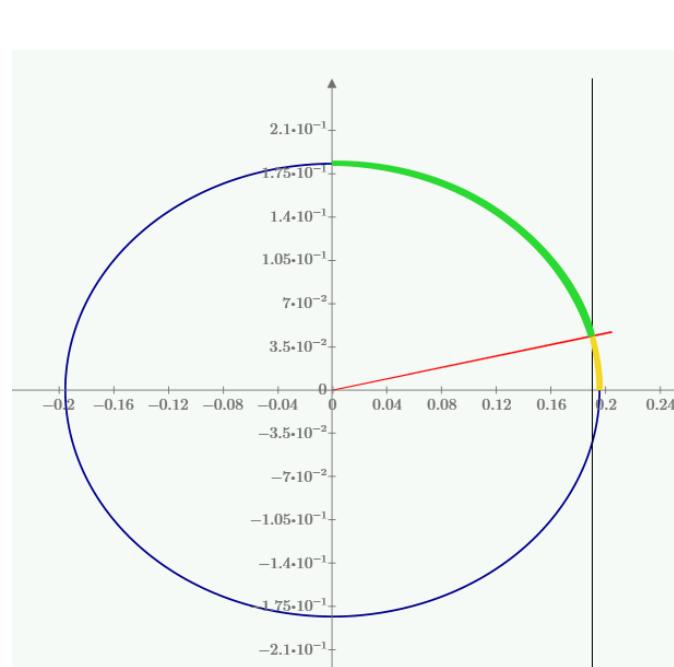
izračunamo začetno pozicijo prenika na elipsi po x-osi (na sliki 16 je pozicija prikazana s črno premico). Ta mi bo pomagala pri upoštevanju odklona.

("Calculating", 2011)



Slika 16: Prikaz izračuna začetne pozicije po x-osi

Nato pa sem z uporabo integrala izračunal dolžino loka



Slika 17: Prikaz dobljene in potrebovane vrednosti

$$s_{toe} = \int_0^{x_{toe}} \sqrt{1 + \frac{b^2 \cdot x_{toe}^2}{a^2(a^2 - x_{toe}^2)}} dx_{toe}$$

s_{toe} – dolžina loka odklona

Ta enačba pa še ne poda rezultata, ki ga potrebujem. Enačba poda dolžino loka, ki je na sliki 17 označeno z zeleno barvo. Za moj primer pa potrebujem dolžino rumenega loka. Predpostavil sem, da so vse četrtine elipse po dolžini enake dolžine. Obseg sem zato razdelil na četrtine (*lok*) in od te odštel vrednost, izračunano z

integralom. Tako sem dobil pot odklona premnika. Sedaj pa lahko izračunam tudi dejansko pot premnika pri določenem kotu z enačbo

$$arc_{toe} = lok - s_{toe}$$

arc_{toe} – dejanska pot odklona premnika

("Calculating", 2011; "Elliptic arc", 2012; "Perimeter", 2017)

Princip in enačbe za računanje dejanske poti konice premnika so si med sabo zelo podobne. Pri računanju pozicije premnika po x-osi sem sedaj kotom odklona prištel še kot premika premnika, ki sem ga izračunal v prejšnjem poglavju

$$x = \frac{a \cdot b \cdot \cos(\alpha_{Eoffset} + \alpha_{premnik})}{\sqrt{(b \cdot \cos(\alpha_{Eoffset} + \alpha_{premnik}))^2 + (a \cdot \sin(\alpha_{Eoffset} + \alpha_{premnik}))^2}}$$

x – pozicija premnika na elipsi po x-osi

Ponovno sem z določenim integralom izračunal dolžino loka, ki ga je potrebno še obrniti

$$s = \int_0^x \sqrt{1 + \frac{b^2 \cdot x^2}{a^2(a^2 - x^2)}} dx$$

s – dolžina loka

Tako sem spet od četrtine obsega odštel izračunano vrednost in nato še pot odklona

$$arc = lok - s - arc_{toe}$$

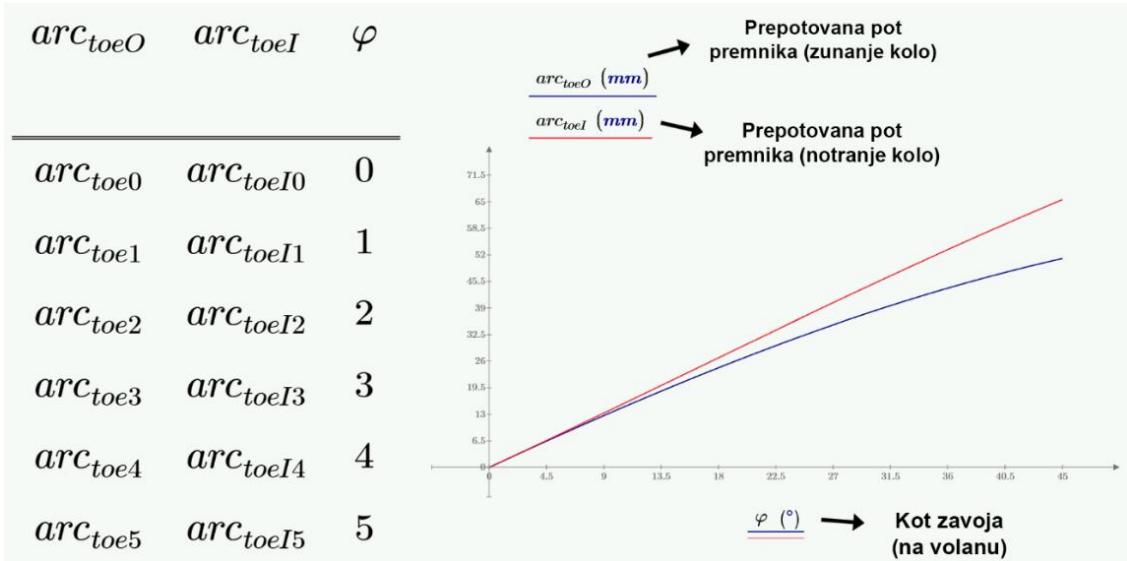
arc – prepotovana pot konice premnika

Končna enačba tako predstavlja prepotovano pot konice premnika. ("Calculating", 2011; "Elliptic arc", 2012; "Perimeter", 2017)

3.3 Grafični prikaz rezultatov

Po izdelavi sistema enačb je bilo treba izračunati vrednosti med kotoma premika volana 0° in 45° za vsako stran posebej. Zaradi programskih omejitev je bilo potrebno napisati sistem enačb za vsak kot posebej. Tako sem dobili 92 rezultatov, ki sem jih nato uredil in vnesel v tabelo. Pripisal sem jim pripadajoče kote zavoja in s tem v

programu vsakemu rezultatu določil točne koordinate. Nato sem v grafu določil veličine na oseh. Vneseni rezultati izrišejo diagram poti konice premnika.



Slika 18: Prikaz rezultatov (levo) in dobljenega grafa (desno)

4 REZULTATI

4.1 Izdelava delajočega 3D modela

Ugotovil sem, da ima Creo vse potrebne funkcije za izdelavo delajočega 3D modela. Tako mi je uspelo narediti realen in delajoč 3D model gokarta in s tem lahko potrdim hipotezo.

4.2 Matematični preračun premika premnika

S programom Mathcad mi je uspelo izdelati sistem enačb, ki mi z vsemi vhodnimi parametri poda natančen končni rezultat. Program ima vse potrebne funkcije za izdelavo sistema enačb in tako lahko hipotezo potrdim.

$$l_{Voffset} = r_{volan} \cdot \sin \alpha_{Voffset}$$

$l_{Voffset}$ – pot odklona izvrtine

r_{volan} – sam radij izvrtine

$\alpha_{Voffset}$ – kot odklona izvrtine

$$l_{volan} = r_{volan} \cdot \sin (\alpha_{Voffset} + \varphi) - l_{Voffset}$$

l_{volan} – pot volanskega droga

φ – kot zavoja, ko voznik zavije z volanom

$$l_{volanI} = r_{volan} \cdot \sin (\varphi - \alpha_{Voffset}) + l_{Voffset}$$

l_{volanI} – pot volanskega droga za kolo na notranji strani zavoja

$$\alpha_{premnik} = \frac{360^\circ \cdot l_{volan}}{2 \cdot \pi \cdot r_{premnik}}$$

$\alpha_{premnik}$ – kot premika premnika

$r_{premnik}$ – radij izvrtine premnika

$$ob = \pi \cdot (3(a + b) - \sqrt{(3 \cdot a + b) \cdot (a + 3 \cdot b)})$$

ob – obseg elipse

a – veliki radij

b – mali radij

$$x_{toe} = \frac{a \cdot b \cdot \cos \alpha_{Eoffset}}{\sqrt{(b \cdot \cos \alpha_{Eoffset})^2 + (a \cdot \sin \alpha_{Eoffset})^2}}$$

x_{toe} – začetni odklon premnika na elipsi po x-osi

α_{Eoffset} – odklon elipse glede na začetno pozicijo premnika

$$s_{toe} = \int_0^{x_{toe}} \sqrt{1 + \frac{b^2 \cdot x_{toe}^2}{a^2(a^2 - x_{toe}^2)}} dx_{toe}$$

s_{toe} – dolžina loka odklona

$$arc_{toe} = lok - s_{toe}$$

arc_{toe} – dejanska pot odklona premnika

lok – četrtina elipse

$$x = \frac{a \cdot b \cdot \cos (\alpha_{Eoffset} + \alpha_{premnik})}{\sqrt{(b \cdot \cos (\alpha_{Eoffset} + \alpha_{premnik}))^2 + (a \cdot \sin (\alpha_{Eoffset} + \alpha_{premnik}))^2}}$$

x – pozicija premnika na elipsi po x-osi

$$s = \int_0^x \sqrt{1 + \frac{b^2 \cdot x^2}{a^2(a^2 - x^2)}} dx$$

s – dolžina loka

$$arc = lok - s - arc_{toe}$$

arc – prepotovana pot konice premnika

5 UGOTOVITVE IN ZAKLJUČEK

Rezultati so že v trenutni fazi uporabni in pripravljeni na medsebojno primerjavo. Z njimi si v tej fazi lahko pomagamo tako, da rezultate z različnimi parametri oziroma komponentami primerjamo med sabo. Več kot bomo naredili primerjav in pa fizičnih testiranj, bolj si bomo z rezultati znali pomagati, saj jih bomo lažje brali in vse skupaj bo še bolj razumljivo.

Naslednji cilj je, da bi računski del naloge popolnoma osamosvojil in ga tako ločil od 3D modela. Mislim, da bi tako zmanjšal možnosti potencialnih napak in pa olajšal sam postopek izračunov. Prav tako pa bi rad model izboljšal in ga približal realnemu gokartu podjetja Kart Republic, to pa bi mi uspelo z dostopom do tehničnih risb komponent. Tako bi bil izračun identičen realnim spremembam.

Dokazal sem, da je izračun premika premnika pri določenem kotu zavoja mogoč. Z izračunom dobimo dovolj natančne rezultate in grafični prikaz, ki nam omogoča primerjavo rezultatov z različnimi komponentami in parametri. Mislim, da bi s pomočjo modela in sistema enačb znatno pospešili proces razvoja komponent in pa tudi razvoja nastavitev na samem gokartu. Za optimalno delovanje sistema pa bi bilo sistem enačb potrebno popolnoma osamosvojiti od modela, saj bi bilo modeliranje vsake nove komponente časovno potratno delo. Sami izračuni pa so vseeno že v trenutni fazi zelo uporabni in mislim, da je to velik korak k modernizaciji razvojne tehnologije v karting svetu. S to nalogo se razvoj samih komponent lahko neizmerno pospeši, hkrati pa močno izboljša učinkovitost samih komponent in gokarta kot celote. Do sedaj je bilo potrebno rezultate neke spremembe predvidevati in nato te spremembe tudi preizkusiti na progi. S pomočjo takega preračuna pa lahko predvidevanja v trenutku preizkusimo. Naknadno lahko spremembe še dodatno izpopolnimo, da iz komponent izvlečemo kar se da največjo učinkovitost le-teh. Vse to nam omogoča preračun, kar pred tem ni bilo mogoče narediti.

6 VIRI IN LITERATURA

Calculating a Point that lies on an Ellipse given an Angle. (2011). Na *math.stackexchange.com*. Pridobljeno 28. januar 2021 s <https://math.stackexchange.com/questions/22064/calculating-a-point-that-lies-on-an-ellipse-given-an-angle>

Elliptic arc length. (2012). Na *keisan.casio.com*. Pridobljeno 28. januar 2021 s <https://keisan.casio.com/keisan/lib/real/system/2006/1343722259/Elliptic%20arc%20length.pdf>

Perimeter of an ellipse. (2017). Na *mathisfun.com*. Pridobljeno 28. januar 2021 s <https://www.mathsisfun.com/geometry/ellipse-perimeter.html>

Turn, turn, turn: The great fib about who really invented Ackermann steering. (2012). Na *hemmings.com*. Pridobljeno 28. januar 2021 s <https://www.hemmings.com/stories/2012/04/17/turn-turn-turn-the-great-fib-about-who-really-invented-ackermann-steering>