SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 446

**SIMULACIJA MEHANIČKOG CRTAČA MATEMATIČKIH FUNKCIJA REALIZIRANOG ZUPČANICIMA**

Valentino Vukelić

Zagreb, svibanj 2022.

Sadržaj

[Uvod 2](#_Toc103452550)

[1. Mehaničke komponente matematičkih operacija 3](#_Toc103452551)

[1.1. Diferencijal 3](#_Toc103452552)

[1.1.1. Način rada diferencijala 4](#_Toc103452553)

[1.2. Množitelj 4](#_Toc103452554)

[1.2.1. Način rada množitelja 5](#_Toc103452555)

[2. Sintaksno parsiranje 7](#_Toc103452556)

[2.1. Gramatika za parsiranje matematičkih izraza 7](#_Toc103452557)

[3. Modeliranje i digitalni zapis virtualnih predmeta 9](#_Toc103452558)

[3.1. Prikaz geometrije poligonima 9](#_Toc103452559)

[Zaključak 10](#_Toc103452560)

[Literatura 11](#_Toc103452561)

[Sažetak 12](#_Toc103452562)

[Summary 13](#_Toc103452563)

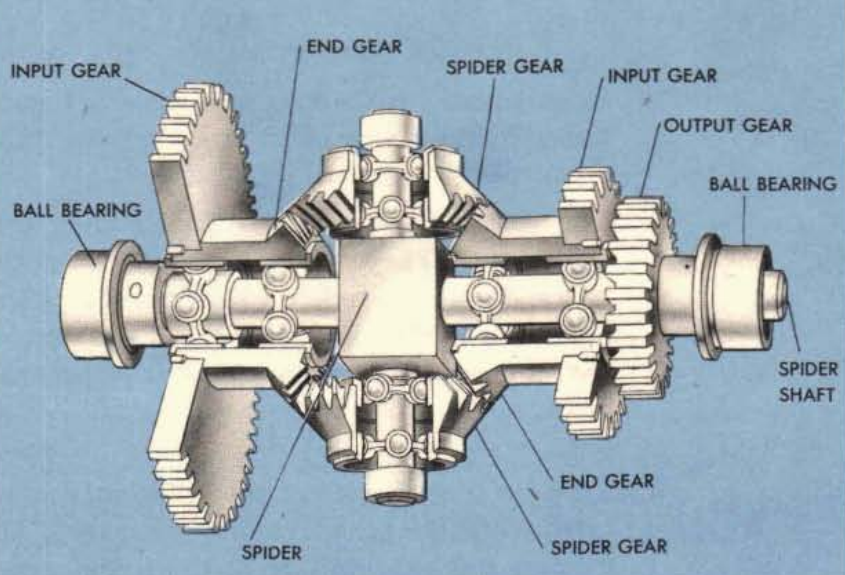
# Uvod

# Mehaničke komponente matematičkih operacija

Postoji više načina za kodiranje vrijednosti u zupčanicima. Za potrebe ovog rada vrijednost se kodira kao broj okretaja zupčanika od početne rotacije. Za modeliranje i izračun racionalnih funkcija, potrebno je mehanički realizirati pet matematičkih operacija. To su zbrajanje, oduzimanje, množenje, dijeljenje i potenciranje. Sve mehaničke komponente iskorištene su iz [1]. Operaciju zbrajanja moguće realizirati diferencijalom. Istu mehaničku komponentu se može iskoristiti za oduzimanje ako se desni pribrojnik pomnoži s minus jedan. Za množenje i dijeljenje iskorišteni su mehanički množitelji (engl. *multiplier*). Pošto su racionalne sastavljene od polinoma, a polinomi imaju samo cjelobrojne potencije varijable *x*, onda se potenciranje može predstaviti nizom množenja.

## Diferencijal

Diferencijal se koristi za zbrajanje rotacije dva zupčanika, pa tako i za zbrajanje dvije vrijednosti. Sastoji se od: (Slika 1) dva ulazna zupčanika (engl. *input gear*), izlaznog zupčanika (engl. *ouput gear*), dva prijenosna zupčanika (engl. *end gears*), dva planetarna konična zupčanika (engl. *spider gears*) i osovine oblika križa (engl. *spider shaft*).



Slika Dijagram diferencijala [1]

*Input* zupčanici su čvrsto povezani s njihovim *end* zupčanicima, odnosno okretanjem *input* zupčanika direktno se okreče i *end* zupčanik. Planetarni *spider* zupčanici su zaduženi za okretanje *spider* osovine, odnosno planetarnim okretanjem *spider* zupčanika okreće se i *spider* osovina. *Output* zupčanik čvrsto je povezan sa *spider* osovinom, odnosno okretanjem *spider* osovine, direktno se okreće i *output* zupčanik.

### Način rada diferencijala

Rad diferencijala lakše je shvatiti pojednostavljenjima. Ako su se oba ulaza okrenula isti broj puta (okreću se istom brzinom), *spider* zupčanici se neće okretati oko svojih osi, odnosno neće se kotrljati po *end* zupčanicima nego će biti zaključani za njih i okretati se planetarno oko *spidera*, istom brzinom kao i ulazni zupčanici. Tako će se *spider* osovina, pa i *output* zupčanik okrenuti isti broj puta kao i ulazni zupčanici. Važno je primijetiti da se *output* zupčanik okrene duplo manje puta nego zbroj koji se traži, pa je na njega potrebno dodati još jedan zupčanik s duplo manje zubaca da se dobije točan zbroj.

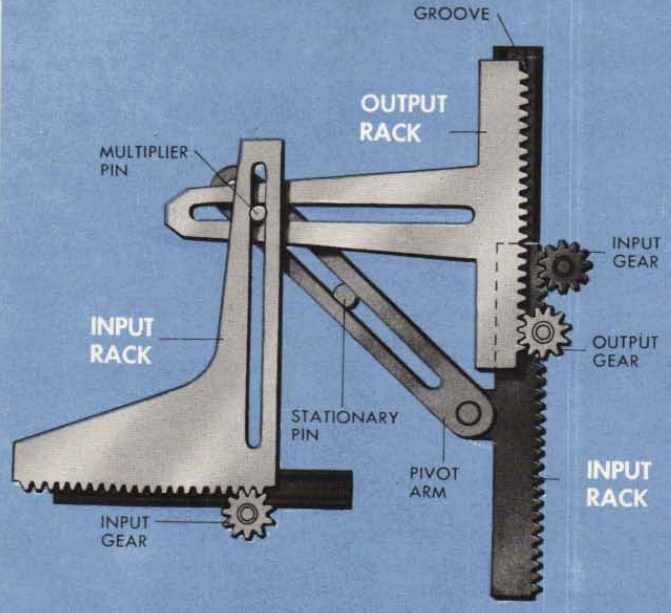
U slučaju da se oba ulaza okreću istom brzinom ali u suprotnim smjerovima, *spider* zupčanici će se okretati oko svojih osi, ali će planetarno stajati na mjestu, odnosno neće okretati *spider* osovinu, pa će rezultat biti nula.

Vidljivo je sada kako okretanjem jednog ulaza brže ili sporije od drugog okreće *spider* zupčanike i oko svoje osi, i oko planetarne osi, te se tako dobije točan zbroj na izlaznom zupčaniku.

## Množitelj

Množitelj se koristi za računanje operacije množenja i dijeljenja. Sastoji se od (Slika 2): dvije ulazne zupčaste letve (engl. *input racks*), izlazne zupčaste letve (engl. *output rack*), stacionarnog *pina* (engl. *stationary pin*), zakretne ruke(engl. *pivot arm*) i *pina* za množenje (engl. *multiplier pin*).

*Pivot arm* se slobodno može okretati i pomicati oko *stationary pina*, a povezan je s desnim *input rackom* koji mu kontrolira rotaciju. *Multiplier* *pin* prolazi kroz sva tri utora, dva na *rackovima* i jedan na *pivot armu*. Bit *multiplier pina* je da pomiče *output rack* kako se pomiču dva *input racka*.



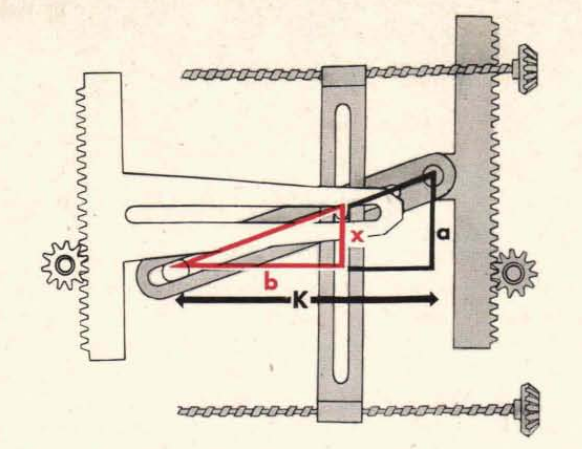
Slika Dijagram množitelja [1]

### Način rada množitelja

Množitelj računa operaciju množenja uz pomoć sličnih trokuta. Trokuti su pravokutni trokuti između *stationary pina* i stožera *pivot arma* kao što je prikazano na dijagramu (Slika 1). Duljina donje katete manjeg trokuta je pomak donjeg *input racka* od *stationary* *pina*, a duljina desne katete je pomak *output racka* od *stationary pina*. Duljina donje katete većeg trokuta je konstanta, odnosno udaljenost *stationary pina* i stožera *pivot arma*, a duljina desne katete je pomak bočnog *input racka* od *stationary pina*. Sada se po poučku proporcionalnosti stranica sličnih trokuta dobiva formula (1).

(1)

Vidi se da je rezultat (pomak *output racka* - *x*) umnožak ulaza (pomak *input rackova* - *a* i *b*), ali je podijeljen s konstantom *K*, tako da je potrebno na *output rack* staviti zupčanik određene veličine da se izlazna vrijednost smanji za faktor *K*.



Slika Dijagram načina rada množitelja [1]

Također je važno primijetiti da množitelj ima fizička ograničenja na veličinu vrijednosti ulaza jer je ograničen dužinom *rackova*, za razliku od diferencijala, a rješenje tog problema objašnjeno je kasnije.

Dijeljenje je postignuto tako da se zamjene jedan ulazni *rack* i izlazni *rack*. S tim da je zbog jednostavnosti prvo izračunata recipročna vrijednost djelitelja, pa je zatim rezultat toga pomnožen s djeljenikom. Dijeljenje odnosno računanje recipročne vrijednosti ima još veća fizička ograničenja, osim apsolutnog gornjeg limita ima i apsolutni donji limit, te nije moguće dijeliti s nulom, odnosno proći kroz nju tijekom mijenjanja predznaka.

# Sintaksno parsiranje

Da bi se generirao mehanizam zupčanika s prije spomenutim komponentama, potrebno je parsirati dobiveni matematički izraz odnosno formulu. Izraz je potrebno pretvoriti u sintaksno stablo gdje su čvorovi matematički izrazi, a listovi su konstante ili varijabla *x*.

Postoje različite tehnike sintaksnog parsiranja [2]:

* odozgo prema dolje (engl. *top-down*)
* prediktivno (engl. *predicting parsing*) – LL(1)
* **metoda rekurzivnog spusta** (engl. *recursive descent parsing*)
* odozdo prema gore (engl. *bottom-up*)

Razlika *top-down* i *bottom-down* parsiranja je što *top-down* generira stablo s vrha, odnosno od korijenskog čvora, dok *bottom-up* generira stablo počevši od listova.

Gramatika sintaksnog parsera sastoji se od jedne ili više produkcija. Produkcija se sastoji od nezavršnog znaka gramatike koji prelazi u niz završnih ili nezavršnih znakova. Primjer jedne produkcije je:

(2)

U produkciji (2) nezavršni znakovi su *P* i *F*, dok je završni znak . Završni znak predstavlja list u sintaksnom stablu, a nezavršni znakovi predstavljaju čvorove, te se stablo generira po produkcijama gramatike.

Metoda rekurzivnog spusta parsira ulazni niz tako da za svaki nezavršni znak postoji funkcija koja primjenjuje njegove produkcije. Za nezavršne znakove produkcije konzumira znak na ulazu, dok za nezavršne znakove rekurzivno poziva njihove funkcije.

## Gramatika za parsiranje matematičkih izraza

Pošto neki matematički operatori imaju prednost nad drugim te mogu biti lijevo ili desno asocijativni, važno točno sastaviti produkcije gramatike da to i poštuju. Tako je Theodore Norvell definirao sljedeća pravila po kojoj je sastavio gramatiku [3]:

* zagrade imaju prednost nad svim ostalim operatorima
* potenciranje ima prednost nad unarnim - i binarnim operatorima /, \*, - i +
* \* i / imaju prednost nad unarnim - i binarnim - i +
* unarni - ima prednost nad binarnim - i +
* potenciranje je desno asocijativno dok su svi ostali binarni operatori lijevo asocijativni

Zatim je poštivajući ta pravila sastavio gramatiku koju je moguće implementirati metodom rekurzivnog spusta:

(3)

(4)

(5)

(6)

gdje {} zagrade označavaju da se izraz u njima ponavlja nula ili više puta, a izraz *a* označava konstantu, odnosno decimalni broj. Produkcija (3) je početna produkcija te označava *expression* koji se sastoji od niza zbrajanja ili oduzimanja *termova*. Pošto je to prva produkcija, bit će i najplića u sintaksnom stablu pa će tako imati najmanju prednost. Također je pravilo lijeve asocijativnosti poštovano pri implementaciji rekurzivne metode tog nezavršnog znaka. Na isti način je definiran i *term* (4) koji se sastoji od niza množenja i dijeljenja *powera*. Moguće je i izostaviti znak operatora pa se to smatra kao množenje tako da je moguće parsirati izraz 2*x*. Nezavršni znak *power* (5) definiran je desnom rekurzijom, tako da poštuje pravilo lijeve asocijativnosti. Na kraju je definirana produkcija za *factor* (6) koji može biti negirani *term* (unarni -), *expression* između zagrada ili konstanta.

# Modeliranje i digitalni zapis virtualnih predmeta

Za modeliranje predmeta u memoriji računala postoji više metoda, a sve se nalaze u spektru između dvije krajnosti [4]: parametarskih metoda i metoda jediničnih elemenata.

Parametarske metode za digitalni zapis koriste predefinirane funkcije ili oblike, koje se onda preciznije oblikuju parametrima. Na primjer definirana je funkcija kugle čiji su parametri radijus i centar u 3D prostoru.

Metoda jediničnih elemenata koristi poligone ili male 3D ćelije (engl. *volume elements, voxels*), smještene u trodimenzionalnom prostoru koji zajedno čine neki veći model.

## Prikaz geometrije poligonima

Prikaz geometrije poligonima najčešći je način zapisa virtualnih predmeta u računalu [4]. Danas je većina sklopovlja optimizirana za iscrtavanje virtualnih scena iz niza poligona i to najčešće trokuta. Modeli su sastavljeni od niza trokuta, a svaki trokut se sastoji od 3 vrha (engl. *vertex*) i 3 brida(engl. *edge*). Točan zapis u memoriji je da se za svaki vrh zapišu njegove tri koordinate, a za svaki trokut se nakon toga zapišu 3 indeksa prije zapisanih vrhova koji čine taj trokut.

Pošto je nezgodno ručno zapisivati koordinate vrhova, najčešće se koriste pomoćni alati koji to rade automatski uz navođenje dizajnera. No nekad je to potrebno raditi parametarski odnosno proceduralno u stvarnom vremenu izvođenja programa (engl. *real time*) i to se zove proceduralna generacija (engl. *procedural generation*). U ovom radu je ta tehnika potrebna za generiranje modela zupčanika bilo koje zadane veličine (radijusa, debljine, broja zubaca, veličine zubaca…).

# Zaključak

# Literatura

1. Navy Department Bureau of Ordnance. *Basic Fire Control Mechanisms*. Washington, D.C.: rujan 1944.
2. John Smith, *Syntax Analysis: Compiler Top Down & Bottom Up Parsing Types*, 14. travnja 2022., <https://www.guru99.com/syntax-analysis-parsing-types.html>, 11. svibnja 2022.
3. Theodore Norvell, *Parsing Expressions by Recursive Descent*, 1999., <https://www.engr.mun.ca/~theo/Misc/exp_parsing.htm>, 11. svibnja 2022.
4. Igor S. Pandžić, Tomislav Pejša, Krešimir Matković, Hrvoje Benko, Aleksandra Čereković, Maja Matijašević, *Virtualna okruženja: interaktivna 3D grafika i njene primjene*, 1. izdanje, Zagreb: Element, 2011.

# Sažetak

Naslov, sažetak, ključne riječi (na hrvatskom jeziku)

Sažetak opisuje sadržaj rada, prepričan u stotinjak riječi.

# Summary

Title, summary, keywords