

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Sveučilišni preddiplomski studij strojarstva

Završni rad

**PRIMJENA INTERAKTIVNE VIZUALIZACIJE ZA
ANALIZU UTJECAJA OBLIKA LOPATICA NA
RAD HIDRAULIČNOG STROJA**

Rijeka, srpanj 2011.

Arsen Sušanj

0069042117

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Sveučilišni preddiplomski studij strojarstva

Završni rad

**PRIMJENA INTERAKTIVNE VIZUALIZACIJE ZA
ANALIZU UTJECAJA OBLIKA LOPATICA NA
RAD HIDRAULIČNOG STROJA**

Mentor: doc. dr. sc. Zoran Čarija

Komentor: mr. sc. Marko Čavrak

Rijeka, srpanj 2011.

Arsen Sušanj

0069042117

TEHNIČKI FAKULTET

Povjerenstvo za završne ispite
Preddiplomskog sveučilišnog studija strojarstva

Klasa: 602-04/11-01/26
Ur. br.: 2170-15-12-11-3

Rijeka, 17. 02. 2011.

Z A D A T A K

za završni rad

Pristupnik/pristupnica: **Arsen Sušanj**

JMBAG: 0069042117

Naziv zadatka: **Primjena interaktivne vizualizacije za analizu utjecaja oblika lopatice na rad hidrauličkog stroja**

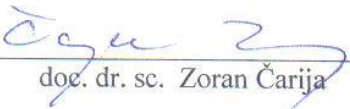
Sadržaj zadatka:

Razviti programsku aplikaciju za proračun radnih karakteristika hidrauličkog stroja te vizualizaciju ulaznih i izlaznih trokuta brzina. Potrebno je omogućiti interaktivnu manipulaciju trokutima brzina te realno u vremenu proračunavanju karakteristika. Za razvijanje programske aplikacije koristiti Matlab ili neki drugi programski jezik.

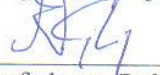
Zadano: 21.02.2011.

**Predati: 14.07.2011. ili
13.09.2011.**


Mentor:


doc. dr. sc. Zoran Čarija


Predsjednik Povjerenstva:


Red. prof. dr. sc. Roberto Žigulić

Komentor:


mr. sc. Marko Čavrak

Zadatak preuzeo dana 27.2.2011


(potpis pristupnika/pristupnice)

Dostaviti:

- Pristupniku/pristupnici
- Mentoru
- Predsjedniku Povjerenstva
- Djelovodi Povjerenstva
- Službi studentske evidencije


IZJAVA

Sukladno članku 11. Pravilnika o završnom radu i završnom ispitu na preddiplomskom sveučilišnom studiju Tehničkog fakulteta Sveučilišta u Rijeci, izjavljujem da sam samostalno izradio završni rad u razdoblju od 21. veljače 2011. do 12. srpnja 2011.

Rad sam izradio prema zadatku Povjerenstva za završne ispite Preddiplomskog sveučilišnog studija strojarstva pod vodstvom mentora doc. dr. sc Zorana Čarije i komentora mr. sc. Marka Čavraka i uz navedenu literaturu.

Arsen Sušanj

Mar. br.: 0069042117



(potpis)

ZAHVALA

Zahvaljujem se mentoru doc. dr. sc. Zoranu Čariji i komentoru dr. sc. Marku Čavraku na podršci, realizaciji, korisnim diskusijama i savjetima tijekom izrade završnog rada. Svojim idejama, prijedlozima i pomoći omogućavali su mi bržu i kvalitetniju izradu završnog rada, posebno u dijelu koji je vezan za izradu programa u Matlab-u. Također se zahvaljujem na pažnji koju su mi posvetili i otvorenosti na suradnju tijekom mog dosadašnjeg studiranja na Tehničkom fakultetu

Sadržaj

1.	Uvod.....	2
1.1.	Cilj i koraci rješavanja zadatka	2
2.	Turbostrojevi	3
2.1.	Podjele turbostrojeva.....	5
2.2.	Turbo strojevi-opći pojmovi.....	6
2.3.	Eulerova jednadžba turbostrojeva	8
3.	Proračun specifične snage turbostroja.....	14
3.1.	Ulazni trokut brzina.....	14
3.2.	Izlazni trokut brzina	15
3.3.	Specifična snaga hidrauličnog stroja.....	16
4.	Opis programa	17
5.	Funkcije koje su korištene u programu:	21
5.1.	Verzija02	21
5.2.	Verzija02_OpeningFcn	22
5.3.	Verzija02_OutputFcn	22
5.4.	Pushbutton_Callback.....	22
5.4.1.	Pushbutton4_Callback.....	22
5.4.2.	Pushbutton5_Callback.....	24
5.4.3.	Pushbutton3_Callback.....	24
5.5.	Edit_Callback	25
5.6.	Edit_CreateFcn.....	25
5.7.	Punjenje_Datoteke	25
5.8.	start_testfcn	27
5.9.	stop_testfcn.....	30
5.10.	testfcn	30
5.11.	circle	36
6.	Zaključak.....	37
7.	Literatura	38
8.	Prilozi	39
8.1.	Argumenti.....	39
8.2.	Originalni kod programa (VERZIJA02)	40
8.2.1.	VERZIJA02.....	40
8.2.2.	Punjenje_Datoteke	46
8.2.3.	start_testfcn	50
8.2.4.	stop_testfcn.....	53
8.2.5.	testfcn	54
8.2.6.	circle	65

1. Uvod

Svakim danom na svijetu živi sve više ljudi. Napredovanjem raznih grana tehnologije potrebno nam je sve više energije kako bi se obezbjedilo korištenje i pokretanje te tehnologije. Zbog toga se sve više koriste obnovljivi izvori energije, poput energije sunca, vjetra i vode. Iskorištavanje energije vodenog potencijala ekonomski je konkurentno proizvodnji električne energije iz fosilnih i nuklearnih goriva, zato je hidroenergija najznačajniji obnovljivi izvor energije. U zadnjih trideset godina proizvodnja u hidroelektranama je utrostručena, a njen udio povećan je za 50%. Proizvodnja u hidroelektranama ima određena tehnička i prirodna ograničenja, a jedno od njih je i zahtjev za postojanjem obilnog izvora vode tokom cijele godine jer je skladištenje električne energije vrlo skupo. Hidroelektrane ili hidroelektrične centrale su postrojenja u kojima se potencijalna energija vode najprije pretvara u kinetičku energiju njezinog strujanja, a potom u mehaničku energiju vrtnje vratila turbine, te konačno u električnu energiju u električnom generatoru. Kako bi se postigla što veća iskoristivost pretvorbe potencijalne energije vode u kinetičku energiju potrebno je odabrati pravilni oblik lopatica turbine pomoću kojeg ćemo dobivati najveću specifičnu snagu što je ujedno i moj glavni cilj u ovom radu.

1.1. Cilj i koraci rješavanja zadatka

Zadatak je razviti programsku aplikaciju za proračun radnih karakteristika hidrauličnog stroja te vizualizirati ulazne i izlazne trokute brzina. Potrebno je omogućiti interaktivnu manipulaciju izlaznih trokuta brzina te realno u vremenu proračunati geometrijske karakteristike turbine kao i ostale vezane veličine poput specifične snage i sl.. Za razvijanje programske aplikacije koristiti programski jezik Matlab. Treba napraviti dio programa koji će pomicanjem određenih točaka davati interaktivno dobivene vrijednosti za relativnu brzinu izlaznog trokuta brzina kuta « β_2 ». Nakon toga treba pomoću grafičkog sučelja, GUI-a, kojeg omogućuje Matlab, napraviti novi prozor prilikom pokretanja programa u kojem će se korisniku omogućiti upisivanje određenih vrijednosti, interaktivno pomicanje točaka na grafu te ispis dobivenih vrijednosti. Krajnji cilj je dobiti interaktivnu ovisnost promjene specifične snage turbostroja ovisno o ulaznom i izlaznom kutu rotora.

2. Turbostrojevi

Na granici čvrstog tijela fluid uvijek djeluje nekom silom na stjenke čvrstog tijela bilo da je u mirovanju ili gibanju. Ukoliko se čvrsto tijelo giba u fluidu javit će se sile koje će nadalje vršiti rad, a koji će s obzirom na gibanje čvrstog tijela biti pozitivan ili negativan.

S obzirom na prirodu sila koje sa javljaju hidrauličke strojeve možemo podijeliti na statičke i dinamičke strojeve.

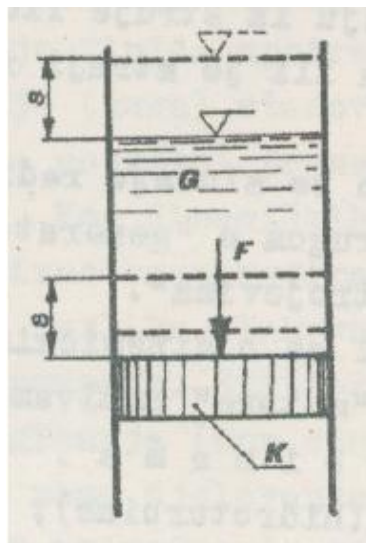
Statički strojevi to su oni strojevi kod kojih je strujanje načelno periodičko, a izmjena energije se ne vrši silama hidrodinamičkog napona (već silama tlaka i potiskivanjem trenja).

Dinamički strojevi kod kojih je strujanje stacionarno, a najveći dio promjene energije struje dolazi od rada hidrodinamičkih sila uzgona (optjecanje uzgonskih tijela).

Primjer statičkog stroja je vertikalna cijev zatvorena odozdo stapom nad kojim se nalazi kapljevina težine G . Izvrši li se pomak stap nagore za put s , obaviti će se rad

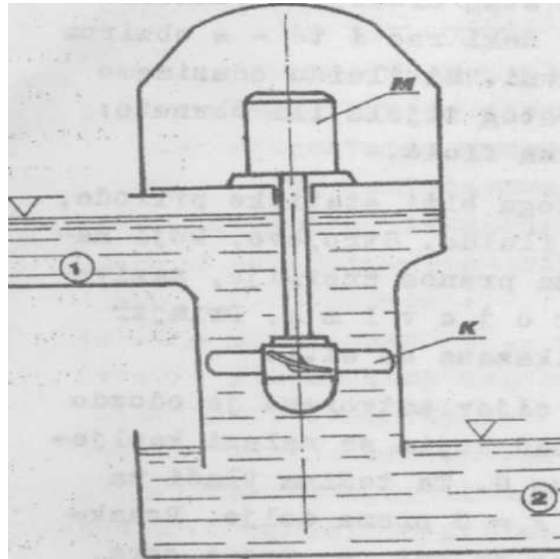
$$W = G \cdot s \quad (2.1)$$

Time će se povećati potencijalnu energiju položaja fluida koji je pomaknut vertikalno naviše za visinu s . S druge strane, dozvolimo li da težina G fluida pomakne stap naniže, to će se dio energije fluida prenijeti na stap dok će se istovremeno energija fluida smanjiti za isti iznos.



Slika 1-Statički stroj

S druge strane kod gibanja čestica fluida javljaju se i druge sile povezane s gibanjem fluida. Strojevi koji koriste takve sile nazivaju se dinamički strojevi. Princip rada dinamičkog stroja prikazana je na sljedećoj slici.



Slika 2 – Dinamički stroj

Spremnici sadrže tekućinu čije se razine nalaze na različitim visinama, a povezana su vertikalnom spojnom cijevi u kojoj se nalazi radno kolo s lopaticama. Radno kolo povezano je s električnim strojem.

Ukoliko se s električnim strojem pogoni radno kolo ono će podizati kapljevину iz donjeg spremnika u gornji. U tom slučaju se troši energija (električna energija potrebna za pokretanje elektromotora) koja se preko vratila i lopatica kola predaje tekućini. Višak energije tekućine se očituje u višoj potencijalnoj energiji fluida koji se nalazi u gornjem spremniku.

Suprotan slučaj prethodnome je kada kapljevina teče iz gornjeg spremnika u donji. U tom slučaju tok fluida će pokretati radno kolo uslijed čega će se ostvariti okretni moment na radnom kolu. Kako je električni stroj čvrsto vezan vratilom sa radno kolo to će za krajnju posljedicu imati proizvodnju električne energije električnim strojem.

2.1. Podjele turbostrojeva

Općenito možemo zaključiti da ćemo ukoliko povećamo energiju fluida takve strojeve nazvati radnim ili motorima ili neturbinama, a ukoliko smanjimo energiju takvu fluida tada ćemo takve strojeve nazivati pogonjenim strojevima ili turbinama.

Dinamički pogonjenim strojevima podrazumijevaju se turbine (od lat. turbinus - virovit, koji ima vir). Ovisno o mediju s kojim rade turbine je nadalje moguće podijeliti na vodne, parne i plinske (termičke ili toplinske) i zračne (vjetroturbine).

Dinamički radnim strojevima se podrazumijevaju crpke, ventilatori, puhala i kompresori.

Hidrauličkim strojevima nazivamo sve strojeve koji rade s praktički nestišljivim fluidom (kapljevnom ili plinom do tlaka kod kojeg ga još možemo smatrati nestišljivim).

Radi toga će se u nastavku proučavati:

- vodne turbine* - strojevi za proizvodnju mehaničke energije koja se putem generatora pretvara u električnu energiju
- crpke* - strojeve za transport kapljevine
- ventilatori* - strojeve za transport plinova pod niskim tlakom.

Po konstrukciji strojeve dijelimo na rotacione i nerotacione, a ova podjela se gotovo podudara s podjelom na dinamičke i statičke strojeve.

Rotacionim strojevima podrazumijevamo sve one strojeve kod kojih se pretvorba energije zbiva uz pomoć rotora (često puta se koristi naziv kolo), za razliku od nerotacionih strojeva gdje se pretvorba zbiva na neki drugi način (u prvom redu pravocrtnim gibanjem klipa).

Kod statičkih je strojeva za pomak fluida iskorišteno neko drugo tijelo (npr. stap ili klip) dok je kod dinamičkih strojeva tu ulogu preuzela lopatica stroja. Lopatica rotora svojom rotacijom ostvaruje razliku tlaka koja za posljedicu ima strujanja fluida u željenom smjeru.

Turbostrojevima nazivamo sve strojeve kod kojih se pretvorba energije u rotoru zbiva po Eulerovim zakonima turbostrojeva.

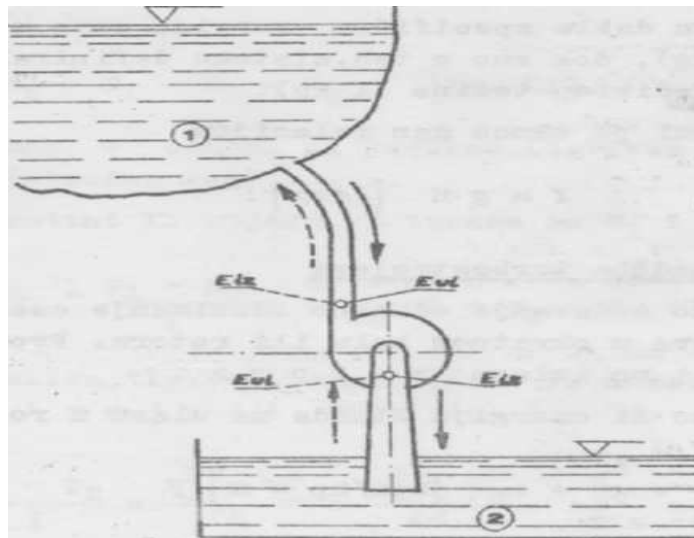
Danas se u pravilu gotovo isključivo u inženjerskoj praksi koriste samo dinamički strojevi dok se statički strojevi koriste u specijalnim slučajevima (npr. u čeličanama gdje za pogon određenih linija potrebno razviti vrlo visok tlak ulja).

Odnos hidrauličkih i toplinskih strojeva

Osnovna je razlika u gustoći medija. Kod hidrauličkih pogonjenih strojeva (vodnih turbina) koji rade s vodom, gustoća je reda veličine 10^3 puta veća nego kod toplinskih strojeva. Toplinske turbine imaju više stupnjeva rotora (do 60-ak) sa ulaznim tlakom od oko 250 bara za razliku od vodnih turbina koje imaju jedan rotor i ulazni tlak vode do 1.5-50 bara.

2.2. Turbo strojevi-opći pojmovi

Turbostrojevi, ovisno o vrsti, fluidu oduzimaju ili dodaju energiju. Na slici Sl. 2.1 je prikazan sustav sa dva spremnika međusobno spojenih cjevovodom u koji je ugrađen turbostroj. Ukoliko fluid (u pravilu se radi o vodi) struji iz gornjeg spremnika u donji spremnik tada se takav turbostroj naziva turbinom. Ukoliko je strujanje suprotnog smjera tada se takav stroj naziva crpkom. Potrebno je napomenuti da crpka može raditi i tako da dodatno gura fluid iz gornjeg spremnika u donji, ali su takvi slučajevi u praksi rijetki.



Slika 3- Princip rada turbostroja

Bez obzira na smjer strujanja specifična energija na ulaznom presjeku je jednaka

$$e_{ul} = \frac{p_{ul}}{\rho} + \frac{v_{ul}^2}{2} + g \cdot z_{ul} \quad (2.2)$$

dok je na izlazu jednaka

$$e_{izl} = \frac{p_{izl}}{\rho} + \frac{v_{izl}^2}{2} + g \cdot z_{izl} \quad (2.3)$$

Energija je jednaka umnošku specifične energije i masenog protoka.

$$E = m \cdot e[W] \quad (2.4)$$

Prema tome razlika energije koju turbostroj daje ili prima od fluida je jednaka

$$\Delta E_{TS} = E_{izl} - E_{ul} = \pm m \cdot Y \quad (2.5)$$

odnosno

$$\Delta e_{TS} = e_{izl} - e_{ul} = \pm Y = \pm g \cdot H \quad (2.6)$$

Gdje je H [m] neto pad ukoliko se radi o turbini odnosno napor ukoliko se radi o crpki, a Y [J / kg] specifični rad turbostroja. Prema tome raspoloživa energija za turbinu je jednaka

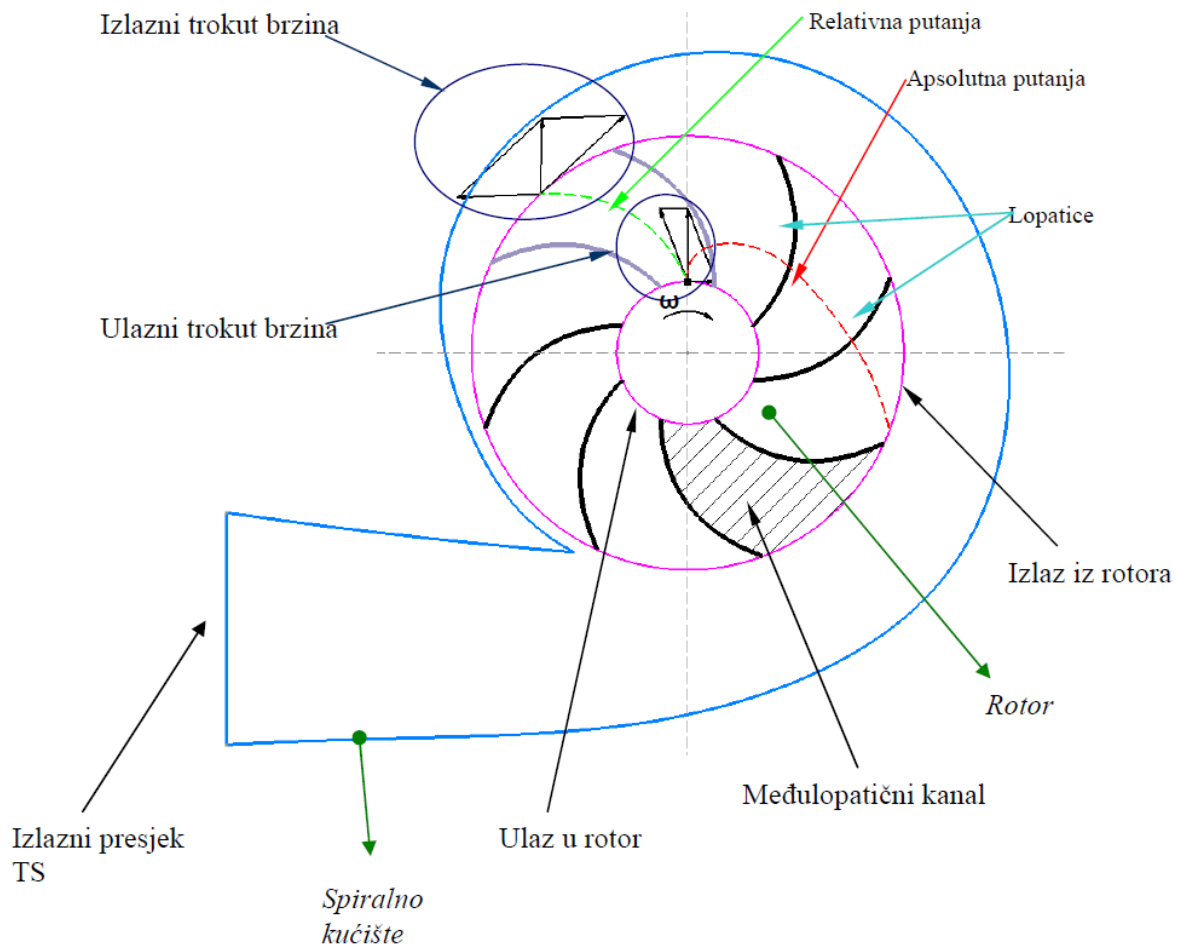
$$e_{ul} - e_{izl} = Y_T \quad (2.7)$$

odnosno za radne strojeve je dobavljena energija jednaka

$$e_{izl} - e_{ul} = Y_{RS} \quad (2.8)$$

2.3. Eulerova jednadžba turbostrojeva

Dodavanje ili oduzimanje energije se kod turbostrojeva odvija pokretnim dijelom stroja - rotorom. Shematski prikaz jednog takvog stroja je prikazano na sljedećoj slici.



Slika 4-Spiralno kućište

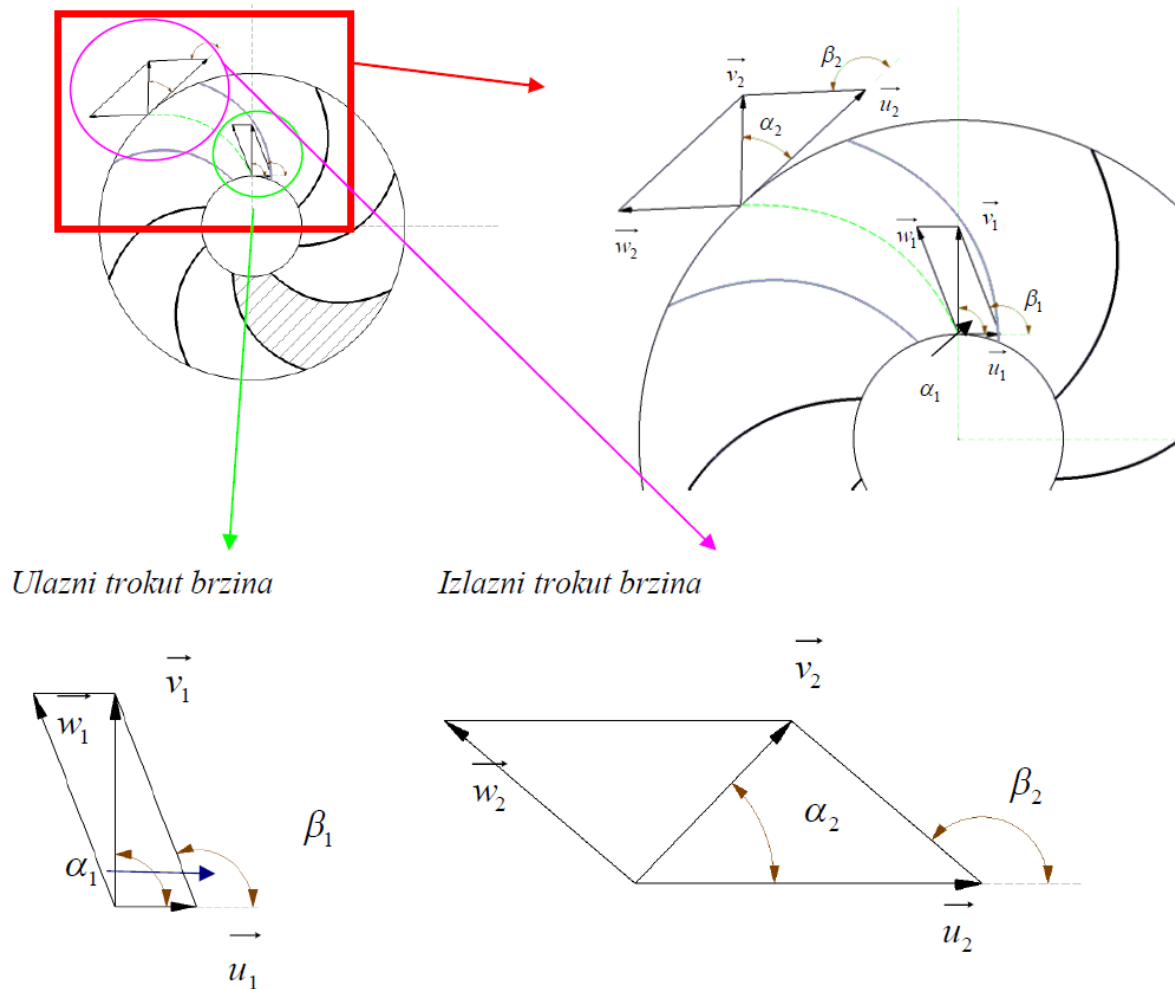
Prikazan je radijalni turbostroj kod kojega se protok fluida ostvaruje u radijalnom smjeru.

Razlikujemo:

Apsolutna putanja - je putanja čestice fluida koja se vidi praćenjem fluidne čestice iz nepomičnog koordinatnog sustava

Relativna putanja - je putanja čestice fluida koja nastaje praćenjem iz pomičnog koordinatnog sustava. Takav koordinatni sustav je kod turbostrojeva vezan uz rotor crpke/turbine i rotira jednakom brzinom kao i rotor turbostroja (TS). U idealnom slučaju (kakav će se ovdje analizirati) oblik relativne putanje odgovarati izgledu lopatice.

U nastavku će se analizirati promjena energije na ulaznom i izlaznom presjeku iz rotora. Na tim presjecima će se postaviti tzv. trokuti brzina koji će poslužiti pri analizi pretvorbe energije.



Slika 5-Ulazni i izlazni trokuti brzina

Na prethodnoj slici vidimo poligon brzina vektorski prikazane na ulaznom i izlaznom trokutu brzina:

- \vec{u} - obodna brzina
- \vec{v} - apsolutna brzina
- \vec{w} - relativna brzina

Kao što se vidi razlikujemo apsolutne, relativne i obodne brzine koje su međusobno povezane sljedećim vektorskim zbrojem.

$$\vec{v} = \vec{v} + \vec{w} \quad (2.9)$$

prema kojem je apsolutna brzina jednaka vektorskom zbroju relativne i obodne brzine.

Vektori brzina, apsolutnih i relativnih, su zakrenuti s obzirom na obodnu brzinu na pripadajućem presjeku za određeni kut kojeg nazivamo

α - kut apsolutne brzine

β - kut relativne brzine

Indeksi 1 i 2 se koriste za pripadajuće presjeke i to,

1- za ulazni presjek

2- za izlazni presjek

npr. \vec{v}_1 - apsolutna brzina na ulazu

npr. α_1 - kut apsolutne brzine na ulazu

Ukoliko je specifična energija na ulaznom presjeku označi s e_1 , a na izlaznom presjeku sa e_2 tada je energija koju je turbostroj primio (primio - pod uvjetom da je $e_1 > e_2$) jednaka

$$\Delta e = e_1 - e_2 \quad (2.10)$$

Ta se energija utrošila na gubitke strujanja fluida kroz rotor ($h_{gub,rot}$) i na koristan rad (H) kola.

$$\frac{\Delta e}{g} = \frac{e_1 - e_2}{g} = H_k + h_{gub,rot} \quad (2.11)$$

Odnosno za turbine

$$H_k = \frac{e_1 - e_2}{g} - h_{gub,rot} \quad (2.12)$$

Prima li fluid energiju (kao kod crpke) kada je $e_2 > e_1$ tada vrijedi

$$H_k = \frac{e_1 - e_2}{g} + h_{gub,rot} \quad (2.13)$$

Dakle u ovome slučaju se dovedena energija koristi na savladavanje hidrauličkih gubitaka u rotoru crpke i za povećanje energije fluida (napora crpke).

Općenito se može zapisati daje

$$\pm H_k = \frac{e_1 - e_2}{g} + h_{gub,rot} \quad (2.14)$$

pri čemu se znak „+“ odnosi na turbine, a znak „-“ na crpke.

Ukoliko analiziramo specifični rad turbostroja tada se može napisati da je odnosno

$$\mp Y = e_1 - e_2 - e_{gub} \quad (2.15)$$

„ - ” radni strojevi (crpke)

„ + ” gonjeni strojevi (turbine)

Općenito je specifična energija (energija po jediničnom masenom protoku) jednaka

$$e = \frac{v^2}{2} + \frac{p}{\rho} + g \cdot z \quad (2.16)$$

Uzimajući u obzir odgovarajuće indekse slijedi da je:

$$\mp Y = \frac{v_1^2 - v_2^2}{2} + \frac{p_1 - p_2}{\rho} + g \cdot (z_1 - z_2) - e_{gub} \quad (2.17)$$

Izraz predstavlja specifični rad TS za realni fluid, budući su uzeti u obzir gubici, izveden u mirujućem koordinatnom sustavu.

Ukoliko se zakon očuvanja energije analizira u pomičnom (rotirajućem) koordinatnom sustavu tada slijedi

$$0 = e_1 - e_2 - e_{gub} \quad (2.18)$$

Specifična energija fluidne čestice u rotirajućem koordinatnom sustavu, u kojem nema razmjene energije ($Y = 0$), je jedinka

$$e = \frac{w^2}{2} + \frac{p}{\rho} + g \cdot z - \frac{u^2}{2} \quad (2.19)$$

Uzimajući u obzir odgovarajuće indekse slijedi da je razlika tlaka jednaka:

$$\frac{p_1 - p_2}{\rho} = \frac{w_2^2 - w_1^2}{2} - \frac{u_2^2 - u_1^2}{2} + g \cdot (z_2 - z_1) + e_{gub} \quad (2.20)$$

Uvrštavanjem prethodnih izraza slijedi da je:

$$\mp Y = \frac{v_1^2 - v_2^2}{2} - \frac{w_1^2 - w_2^2}{2} + \frac{u_1^2 - u_2^2}{2} \quad (2.21)$$

Ovo je važna Eulerova jednadžba turbostrojeva kojom se pokazuje da je razmjena energije (primanje ili davanje) ovisno o promjeni apsolutnih, relativnih i obodnih brzina u rotirajućem kanalu. Osim toga iz prethodnog izraza zaključujemo da je za poznavanje razmijenjene energije dovoljno poznavati stanje na ulaznom i izlaznom presjeku turbostroja.

Treći oblik osnovne jednadžbe TS moguće je dobiti korištenjem kosinusovog poučka

$$w_x^2 = v_x^2 + u_x^2 - 2 \cdot v_x \cdot u_x \cdot \cos(\alpha_x) \quad (2.22)$$

gdje indeks x odgovara ulaznom (1) ili izlaznom (2) presjeku.

$$\mp Y = \frac{v_1^2 - v_2^2}{2} - \frac{v_1^2 + u_1^2 - 2 \cdot v_1 \cdot u_1 \cdot \cos(\alpha_1) - v_2^2 + u_2^2 - 2 \cdot v_2 \cdot u_2 \cdot \cos(\alpha_2)}{2} + \frac{u_1^2 - u_2^2}{2} \quad (2.23)$$

$$\mp Y = u_1 \cdot v_1 \cdot \cos(\alpha_1) - u_2 \cdot v_2 \cdot \cos(\alpha_2) \quad (2.24)$$

Kako je projekcija apsolutne brzine u smjeru obodne brzine jednaka

$$v_u = v \cdot \cos(\alpha) \quad (2.25)$$

slijedi

$$\mp Y = u_1 \cdot v_{1u} - u_2 \cdot v_{2u} \quad (2.26)$$

Ovo je drugi, uobičajeni, oblik Eulerove jednadžbe turbostrojeva.

Uzimajući u obzir da je obodna brzina jednaka umnošku polumjera i kutne brzine

$$u = R \cdot \omega \quad (2.27)$$

slijedi još jedan izraz.

$$\mp Y = \omega \cdot (R_1 \cdot v_{1u} - R_2 \cdot v_{2u}) \quad (2.28)$$

Kod Eulerovih jednadžbi turbostrojeva važno je uočiti sljedeće:

- izrazi za Y_k potpuno su neovisni o smjeru strujanja i smjeru rotacije kola
- bitni su samo uvjeti na ulaznom i izlaznom presjeku kola
- izrazi vrijede za idealni i realni fluid (gubici su uzeti u obzir kod izvođenja)
- izraz Y_k je potpuno je neovisan o fluidu (karakteristikama fluida).

Ukupna snaga kola jednaka je

$$P_k = m \cdot Y^* = \omega \cdot M_k \quad (2.29)$$

Odnosno moment kola jednak je

$$M_k = \frac{m \cdot Y^*}{\omega} \quad (2.30)$$

3. Proračun specifične snage turbostroja

Kako bi mogli izračunati specifičnu snagu turbostroja potrebno je poznavati određene parametre na ulaznom i izlaznom trokutu brzina koji utječu na konačni rezultat.

3.1. Ulazni trokut brzina

Najprije treba izračunati kutnu brzinu koja ovisi o broju okretaja turbostroja

$$\omega = \frac{2 \cdot n \cdot \pi}{60} \left[s^{-1} \right] \quad (3.1)$$

Zatim treba izračunati obodnu brzinu koja proporcionalna kutnoj brzini i unutarnjem radijusu kojeg unosi sam korisnik kao ulazni podatak u program

$$u_1 = r_1 \cdot \omega \left[\frac{m}{s} \right] \quad (3.2)$$

Površina poprečnog presjeka je zavisna je o unutarnjem promjeru i visine turbostroja koju također unosi sam korisnik kao ulazni podatak

$$A_1 = D_1 \cdot \pi \cdot b_1 \left[m^2 \right] \quad (3.3)$$

Apsolutna brzina se izračunava iz protoka koji zadaje korisnik i izračunate površine poprečnog presjeka

$$v_1 = \frac{Q}{A_1} \left[\frac{m}{s} \right] \quad (3.4)$$

Kut apsolutne brzine je unaprijed zadan

$$\alpha_1 = 90^0 \quad (3.5)$$

što utječe na sljedeći uvjet da je meridijalna komponenta apsolutne brzine jednaka apsolutnoj brzini

$$v_{1m} = v_1 \left[\frac{m}{s} \right] \quad (3.6)$$

Zatim računamo relativnu brzinu koju dobivamo iz sljedećag izraza, a proizlazi iz vektorskog zbroja brzina

$$w_1 = \sqrt{u_1^2 + v_1^2} \left[\frac{m}{s} \right] \quad (3.7)$$

Na ulaznom trokutu brzina potrebno je još izračunati kut relativne brzine prema sljedećem izrazu koristeći vrijednosti apsolutne i obodne brzine

$$\beta_1 = \pi - \operatorname{atg} \left(\frac{v_1}{u_1} \right) \quad (3.8)$$

3.2. Izlazni trokut brzina

Kutna brzina je jednaka na ulaznom i izlaznom trokutu brzina stoga nam je potrebno za računanje obodne brzine samo još veličina vanjskog promjera koju dobivamo od korisnika

$$u_2 = r_2 \cdot \omega \left[\frac{m}{s} \right] \quad (3.9)$$

U sljedećem izrazu postavlja se još jedan uvjet koji nam omogućuje neštojednostavniji proračun, a to je da su medijalne komponente apsolutne brzine jednake na ulaznom i izlaznom trokutu brzina

$$v_{2m} = v_{1m} \left[\frac{m}{s} \right] \quad (3.10)$$

Kut relativne brzine β_2 dobiva se iz geometrije interaktivnom manipulacijom izlaznog trokuta brzina

Obodna komponenta relativne proporcionalna je medijalnoj komponenti apsolutne brzine i zavisi o kutu relativne brzine

$$w_{2u} = \frac{v_{2m}}{\operatorname{tg}(\pi - \beta_2)} \left[\frac{m}{s} \right] \quad (3.11)$$

Također se računa i obodna komponenta apsolutne brzine kao razlika obodne brzine i njezine komponente relativne brzine

$$v_{2u} = u_2 - w_{2u} \left[\frac{m}{s} \right] \quad (3.12)$$

Apsolutna brzina se dobiva iz svoje meridijalne i obodne komponente

$$v_2 = \sqrt{v_{2m}^2 + v_{2u}^2} \left[\frac{m}{s} \right] \quad (3.13)$$

Kut apsolutne brzine nije točno određen kao u ulaznom trokutu brzina te se računa na sljedeći način:

$$\alpha_2 = \operatorname{atg} \frac{v_{2m}}{v_{2u}} \quad (3.14)$$

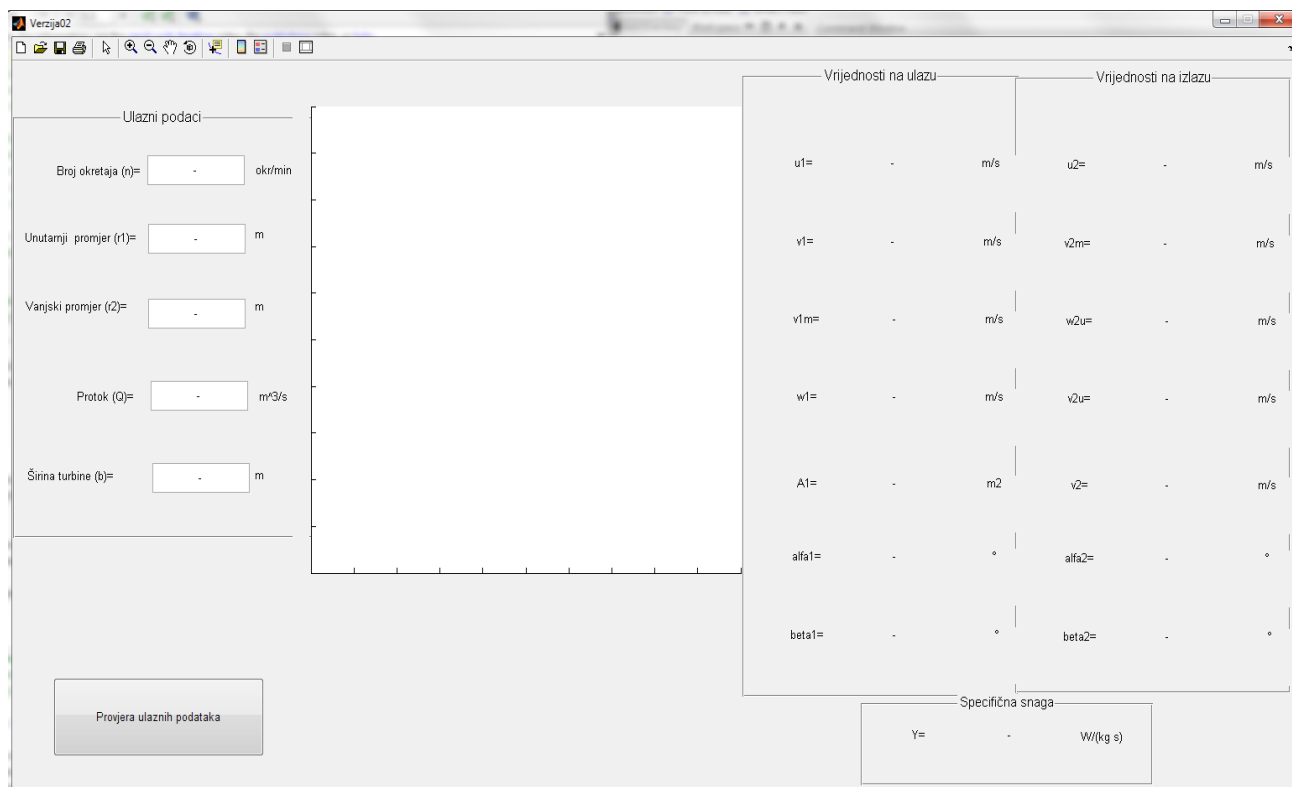
3.3. Specifična snaga hidrauličnog stroja

Specifična snaga hidrauličnog stroja ovisi o svim prethodno proračunatim vrijednostima, a proporcionalno je direktno vezana uz vrijednosti na izlaznom trokutu brzina, a to su obodna brzina i obodna komponenta apsolutne brzine.

$$Y = u_2 \cdot v_{2u} \left[\frac{J}{kg} \right] \quad (3.15)$$

4. Opis programa

Program je napravljen tako da se pri pokretanju osnovne/glavne funkcije «Verzija02» otvara prozor koji je napravljen korištenjem grafičkog sučelja GUI.



Slika 6- Grafičko sučelje u GUI-u

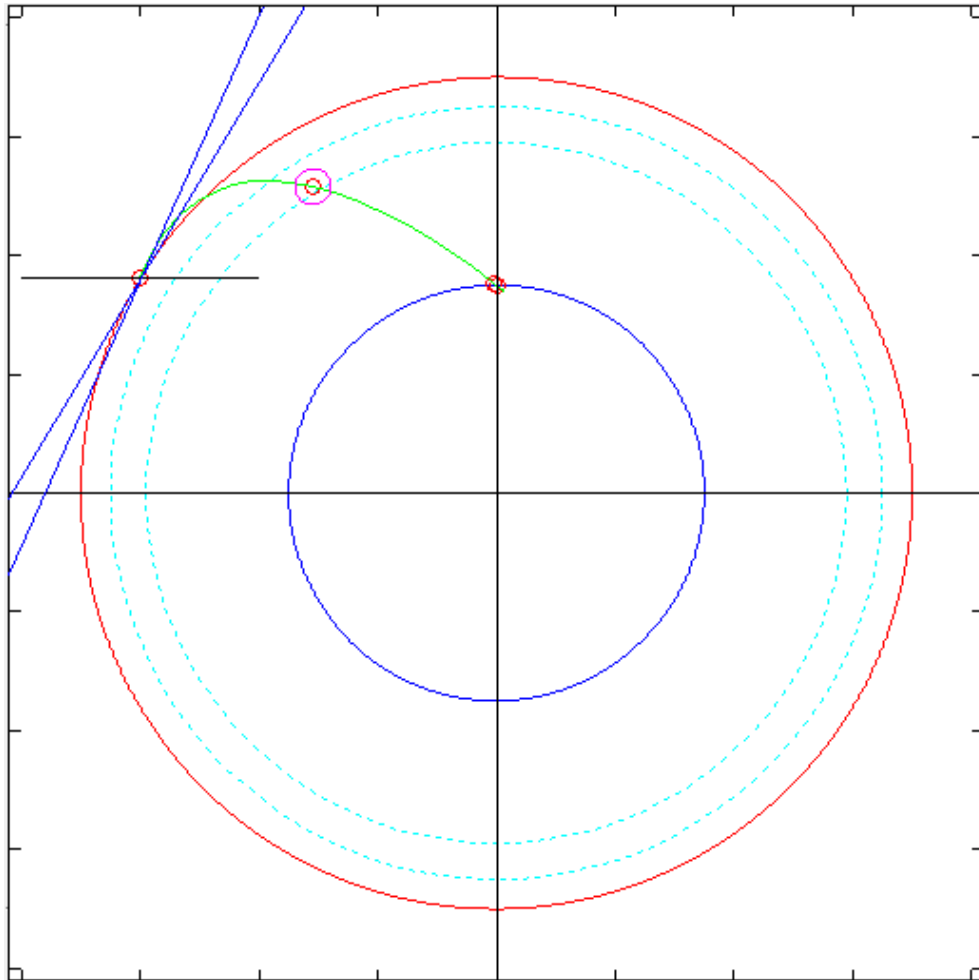
Na lijevoj strani prozora nalaze se prazna mjesta u koja korisnik upisuje podatke koji se od njega traže. To su takozvani «ulazni podaci». Ispod tog bloka za upis podataka nalazi se «pushbutton» čijim klikom provjeravamo dali su podaci zadovoljili postavljene uvjete. Ako uvjeti nisu zadovoljeni za pojedine podatke tada se korisnika ponovno pita da upiše određene vrijednosti. Zadovoljavanjem uvjeta otvara se novi «pushbutton» koji na klik mišem pokreće program.

Ulazni podaci

Broj okretaja (n)=	<input type="text" value="450"/>	okr/min
Unutarnji promjer (r1)=	<input type="text" value="1"/>	m
Vanjski promjer (r2)=	<input type="text" value="2"/>	m
Protok (Q)=	<input type="text" value="45"/>	m ³ /s
Širina turbine (b)=	<input type="text" value="0.2"/>	m

Slika 7-Blok u koji korisnik unosi tražene vrijednosti

Pokretanjem interaktivnog proračuna najprije se u središnjem dijelu prozora prikazuje graf koji predstavlja vizualni dio ulaznih i izlaznih trokuta brzina. Kružnica plave boje predstavlja unutarnji radijus « r_1 », a crvene boje vanjski radijus « r_2 ». Svijetlo plava boja predstavlja ograničeno područje u kojem se može pomicati središnju točku, ili klikom na vanjsku točku pojavljuje se područje po kojem se dopušta pomicanje te točke dok su dvije točke oko unutarnjeg radijusa fiksirane te se ih ne može pomicati.



Slika 8-Graf za interaktivno računanje geometrijom

Zelenom bojom prikazana je *spline* krivulja koja sadrži sve točke, tj. prolazi kroz njih. Na mjestu gdje *spline* krivulja siječe vanjsku kružnicu radijusa r_2 , tj. na poziciji gdje se nalazi vanjska točka prikazana su tri pravca koja sazržavaju tu točku. Pravac paralelan X-osi (crna boja), tangenta na kružnicu (plava boja) koja s navedenim pravcem zatvara kut « φ_2 », te još jedan pravac koji također s navedenim pravcem zatvara kut « φ_1 ».

S desne strane grafa nalaze se dva bloka s proračunatim vrijednostima na ulaznim i izlaznim trokutima brzina. Zbog fiksnih točaka na unutarnjem radijusu za pojedine unesene korisnikove podatke vrijednosti na ulaznom trokutu su konstantne, dok se na izlaznom trokutu vrijednosti interaktivno mijenjaju ovisno o kutu « β_2 » kojije zavisano o kutevima « ϕ_1 » i « α_2 ».

Ulazni trokut brzina			Izlazni trokut brzina		
u1=	47.1239	m/s	u2=	94.2478	m/s
v1=	47.7465	m/s	v2m=	47.7465	m/s
v1m=	47.7465	m/s	w2u=	449.725	m/s
w1=	67.0849	m/s	v2u=	-355.477	m/s
A1=	0.942478	m ²	v2=	358.669	m/s
alfa1=	90	°	alfa2=	7.64999	°
beta1=	134.624	°	beta2=	173.894	°
Specifična snaga					
Y=	33502.9	W/(kg s)			

Slika 9-Prikaz vrijednosti na ulaznom i izlaznom trokutu brzina te specifična snaga

5. Funkcije koje su korištene u programu:

- Verzija02
- Verzija02_OpeningFcn
- Verzija02_OutputFcn
- Pushbutton_Callback
- Edit_Callback
- Edit_CreateFcn
- Punjenje_Datoteke
- start_testfcn
- stop_testfcn
- testfcn
- circle

5.1. Verzija02

Ova funkcija stvara «Verzija02.fig», korištenjem GUI-a grafičkog sučelja, što znači da je osnova na kojoj se stvara grafika potrebna za izvođenje programa. «Verzija02.fig» je file kojeg kreira sam Matlab, a u njemu se nalaze informacije koje su potrebne za stvaranje cijelog prozora koji se kreira u GUI-u, tj informacije o veličini prozora, pozicija pojedinih dijelova unutar prozora, te njihova veličina. Glavna funkcija sastoji se od podfunkcija koje pokreću grafički dio programa, funkcija za unos argumenata od strane korisnika(edit-funkcije) i funkcija koje postepeno pokreću određene dijelove programa (pushbutton) ovisno o korisniku. Unutar same funkcije poziva se više podfunkcija. Neke od njih su definirane unutar same funkcije, a to su: «Verzija02_OpeningFcn», «Verzija02_OutputFcn», «Pushbutton:Callback», «Edit_Callback», «Edit_CreateFcn». Svaka od tih funkcija i njezine svrhe bit će objašnjene u nastavku.

5.2. Verzija02_OpeningFcn

Nalazi se unutar glavne funkcije «Verija02» od 38-50 reda koda. Ovo je podfunkcija koja se poziva glavnom funkcijom. U njoj se nalazi kod za ono što će se korisniku prikazati pri samom pokretanju programa. Od tri «pushbutton-a» koja se koriste u programu samo će jedan bit vidljiv pri samom pokretanju programa, a ostali će se pojavljivati ovisno o tijeku programa.

Primjer kako se onemogućuje vidljivost «pushbutton-a»:

```
00000048 set(handles.pushbutton3, 'Visible', 'Off');  
00000049 set(handles.pushbutton5, 'Visible', 'Off');
```

5.3. Verzija02_OutputFcn

Podfunkcija pozvana glavnom funkcijom koja nam omogućuje da prikazujemo izlazne podatke, npr podatke koje smo dobili izračunavanjem za ulazne i izlazne trokute brzina. Nalazi se od 51-54 reda koda glavne funkcije.

5.4. Pushbutton_Callback

Ova podfunkcija se također poziva unutar glavne funkcije, ali nju poziva sam korisnik. Za njezino pokretanje treba kliknuti mišem na pravokutnik koji je naznačen u prozoru koji se otvori pokretanjem programa. Unutar samog programa postoje tri ovakve funkcije, ali rade na isti princip zbog čega sam ih opisao sve zajedno.

5.4.1. Pushbutton4_Callback

Ova podfunkcija nalazi se od 112-170 reda koda glavne funkcije. Provjerava dali su upisani svi podaci i dali zadovoljavaju postavljene uvjete. «br» je varijabla koja ima svrhu brojača koja u početku ima vrijednost «0».

```
00000115 br=0;  
00000116  
00000117 if get(handles.edit1, 'String')== '-'  
00000118     set (handles.edit1, 'String', 'upisi vrijednost');  
00000119 else  
00000120     br=br+1;  
00000121 end
```

Pomoću if-funkcije i naredbe «get» za čitanje podataka koje korisnik unosi u mjesto predviđeno za to provjerava se dali je unesen podatak. Primjer je pokazan samo za provjeru prvog «Edit-text-a», jer se provjera za preostala četiri vrši na identičan način. To se radi tako

da za svaki pojedini unos u «Edit-text» treba provjeriti dali je jednak stringu '-' koji se u početku tamo nalazi. Ukoliko je tvrdnja točna naredbom «set» u «Edit-text» se vrši upisivanje stringa «upisi vrijednost» te brojač zadržava jednaku vrijednost, inače se brojač povećava za vrijednost «1».

Ako je brojač manji od «5» znači da barem u jednom polju nije upisana vrijednost, a koja je potrebna za daljnje izvođenje programa, te ovaj «pushbutton4» postaje nevidljiv, a omogućuje se vidljivost «pushbutton5» koji se u prozoru nalazi na istom mjestu i jednake je veličine tako se ne primjećuje da se ne manipulira istim.

«r1» i «r2» su varijable koje predstavljaju unutarnji i vanjski promjer te im se pridružuju vrijednosti iz drugog i trećeg «Edit-text-a» na sljedeći način:

```
00000148 r1=get(handles.edit2,'String');
00000149 r1=str2double(r1);
00000150 r2=get(handles.edit3,'String');
00000151 r2=str2double(r2);
```

«str2double» je naredba koja pretvara string u broj jer «get» naredbom sve podatke dobivamo u obliku stringa bez obzira dali su oni možda brojevi.

Sljedeća if petlja provjerava dali je «r1» veći od «0.75 r2».

```
00000158 if r1 > (0.75*r2)
00000159     r1='prevelik r1';set(handles.edit2,'String',r1);
00000160     r2='preмали r2';set(handles.edit3,'String',r2);
00000161     set(handles.pushbutton5, 'Visible', 'On');
00000162     set(handles.pushbutton4, 'Visible', 'Off');
00000163 end
```

Ako je tvrdnja točna sa «bushbutton-ima» se događa isto ono sto je već objašnjeno u prošlom logičkom uvjetu tj if petlji, ali se još dodatno daju upozorenja korisniku , a to su stringovi «prevelik r1» i «preмали r2» koji se ispisuju u polja «edit-text-a» u koja se inače korisnik upisuje vrijednosti unutarnjeg i vanjskog promjera

Ako je brojač «br=5» i ako je «r1<0.7500000001 r2», «pushbutton4» postaje nevidljiv, a «pushbutton3» postane vidljiv. Primjer kako je to zapisano u Matlabu:

```
00000165 if br==5
00000166     if r1 < (0.7500000001*r2)
00000167         set(handles.pushbutton3, 'Visible', 'On');
00000168         set(handles.pushbutton4, 'Visible', 'Off');
00000169     end
00000170 end
```

5.4.2. Pushbutton5_Callback

Ova podfunkcija nalazi se od 174-224 reda koda glavne funkcije. Provjerava dali su upisani svi podaci i dali zadovoljavaju postavljene uvjete. Njega je moguće pozvati tek pri drugom pokušaju upisivanja traženih vrijednosti tj kad «pushbutton4» postane nevidljiv kako je to već ranije objašnjeno u tumačenju prethodne podfunkcije.

Ovaj blok naredbi ima identičnu svrhu i funkcionira na identičan način kao i već objašnjeni prošli blok.

Ima nekolicinu razlika, a jedan od njih je da brojac u početku poprma vrijednost «br=5».

```
00000181 br=5
00000182 str1=get(handles.edit1, 'String');
00000183 if isnan(str2double(str1))
00000184     br=br-1
00000185 end
```

U poljima za koja se vrše provjere se više ne nalaze crtice nego već uneseni brojevi od strane korisnika ili stringovi «upisi vrijednost», «preмали r2» ili «preveliki r1», koji služe kao uputa korisniku. Zbog toga se i sama provjera vrši drugim naredbama. «str1» je varijabla koja poprma vrijednost onoga sta se nalazi u polju «Edit-text-a». Naredba «isnan» provjerava dali je naredba «str2double» uspjela vrijednost varijable «str1» pretvoriti u broj ili je ona zaista string. Ako je tvrdnja točna znači da varijabla ne sadrži broj nego string. Tada se brojač umanjuje za jedan, što će rezultirati u daljnjim provjerama ponovno vraćanje korisnika na novo upisivanje podataka kako je to već objašnjeno za podfunkciju «pushbutton4».

5.4.3. Pushbutton3_Callback

Nalazi se od 55-70 reda koda glavne funkcije. Pokreće sljedeću podfunkciju «Punjenje_Datoteke»u koju šalje i vrijednosti potrebnih varijabli «r1» i «r2»koje je korisnik upisivao u polja koja su za to predviđena.

```
00000060 r1=get(handles.edit2, 'String');
00000061 r1=str2double(r1);
00000062 r2=get(handles.edit3, 'String');
00000063 r2=str2double(r2);
00000064 % r1 i r2 se uzimaju iz unesenih vrijednosti jer su potrebni
kao varijable
00000065 % za pozivanje podfunkcije "Punjenje_Datoteke"
00000066 Punjenje_Datoteke(handles, r1, r2)
```

5.5. Edit_Callback

Podfunkcija koja nam omogućuje korištenje podataka koje unosi korisnik. U programu ima pet takvih podfunkcija jer je potreban unos vrijednosti pet varijabli.

```
00000071 function edit1_Callback(hObject, eventdata, handles)
```

One su definirane unutar glavne funkcije od 71-100 reda koda.

5.6. Edit_CreateFcn

Podfunkcija pomoću koje se korisniku mogu predati neke informacije. U programu se nalazi pet takvih funkcija jer postoji pet polja za unos vrijednosti potrebnih varijabli, tj tih istih pet polja se mogu koristiti i za davanje nekih obavijeti korisniku.

```
00000072 function edit1_CreateFcn(hObject, eventdata, handles)
00000073 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
00000074     set(hObject,'BackgroundColor','white');
00000075 end
```

One su definirane unutar glavne funkcije od 72-104 reda koda

5.7. Punjenje_Datoteke

Unutar ove funkcije izračunavaju se prvobitne vrijednosti točaka kroz koje se crta *spline*, te definira neke globalne varijable koje su potrebne za daljnje izvođenje programa koji se nastavlja pozivanjem sljedećih podfunkcija. Podfunkcija se nalazi u zasebnom file-u, a poziva se glavnom funkcijom na način kako je već ranije objašnjeno pod podnaslovom «pushbutton3».

Vrijednosti lokalnih varijabli «r1» i «r2» koje su dobivene iz glavne funkcije koristimo za crtanje kružnica koje predstavljaju unutarnji i vanjski promjer.

```
00000024 r2t=7;
00000025 r1t=r1*7/r2;
00000026 circle([0,0],r1t,100000,'-');
00000027 hold on;
00000028 circle([0,0],7,100000,'r-');
```

«r2t» je varijabla prema čijoj vrijednosti ($r2t=7$) se crta vanjski radijus, a «r1t» varijabla prema čijoj vrijednosti se crta unutarnji radijus, a računa se prema omjeru stvarnih vrijednosti unutarnjeg i vanjskog radijusa. Te vrijednosti se koriste samo za crtanje kako bi cijela slika stala u graf definirane veličine.

```
00000032 axis([-8.2 8.2 -8.2 8.2]);
```

Prikazano je kako se u Matlabu definiraju granice grafa, a vrijednosti iznose od «-8,2 – 8,2 po X-osi» i «-8,2 – 8,2 po Y-osi»

Globalne varijable koje se koriste u više podfunkcija treba navesti na sljedeći način:

```
00000034 global l k t1 t2 t3 t4;
00000043 l=0;
00000044 k=0;
00000045 xy = [];
```

Kako bi globalne varijable bilo moguće kasnije koristiti treba odrediti kakvog su tipa te odrediti početnu vrijednost na način kako je prikazano.

Prva točka koja je fiksna nalazi se na koordinatama «x=0»i «y=r1t».

```
00000050 i=1;
00000051 xi=0;
00000052 mn1=xi;
00000055 yi=(r1t^2-xi^2)^(1/2);
00000056 mn2=yi;
00000057 h11=plot(xi,yi,'ro');
00000058 xy(:,i) = [xi;yi];
00000059 XY(1,1)=xi;
00000060 XY(1,2)=yi;
```

Koordinate točaka se pohranjuju u matricu «XY». Za sve točke spremanje koordinata vrši se na isti način, X-koordinate u prvi stupac, a Y-koordinate u drugi.

Druga fiksna točka (u programu navedena kao druga dodana) računa se pomoću kuta relativne brzine «beta1» korištenjem trigonometrije. Navedeni kut potrebno je izračunati korištenjem podataka koji je unio korisnik.

```
00000065 Q=get(handles.edit4,'String');
00000066 Q=str2double(Q);
00000067 b=get(handles.edit5,'String');
00000068 b=str2double(b);
00000069 A1=2*r1*pi*b;
00000070 v1=Q/A1;
00000071 n=get(handles.edit1,'String');
00000072 n=str2double(n);
00000073 omega=2*pi*n/60;
00000074 u1=r1*omega;
```

```

00000075 beta1=pi-atan(v1/u1);
00000076 ddy=abs(0.05*sin(pi-beta1));
00000077 ddx=abs(0.05*cos(pi-beta1));
00000078
00000079 xi=mn1-ddx;
00000080 yi=mn2+ddy;

```

«mn1» i «mn2» su varijable u kojima se pamte koordinate prve točke, jer je druga fiksna točka pomaknuta po X-osi za «ddx» i po Y-osi za «ddy».

Druga točka ima točno određene koordinate, tj o ne zavise ni o jednoj varijabli:

```

00000087 xi=-3.35;
00000088 yi=4.92;

```

Treća točka ima određenu X-koordinatu, dok se Y-koordinata računa prema formuli koja je ovisna o «r2t».

```

00000096 xi=-6;
00000097 yi=(49-xi^2)^(1/2);

```

Krivulja «*spline*» je određena sljedećim varijablama: t-broj točaka, xy-matrice koja sadrži koordinate točaka kroz koje prolazi, ts-broj dijelova od kojih se sastoji.

```

00000105 t= 1:i;
00000106 ts = 1:0.1:i;
00000107 xys = spline(t,xy,ts);
00000108 h10=plot(xys(1,:),xys(2,:), 'g-');

```

Navedena krivulja crta se u zelenoj boji.

Na kraju ove podfunkcije pozivaju se dvije podfunkcije ovisno dali se tipka piša stisne («start_testfcn») ili otpusti («stop_testfcn»).

```

00000117 XYo=XY;
00000118 set(gcf,'WindowButtonDownFcn',{@start_testfcn,XYo,handles});
00000119 set(gcf,'WindowButtonUpFcn',{@stop_testfcn});

```

«XYo» je lokalna varijabla koja se koristi u pozvanim podfunkcijama, a sadrži koordinate točaka kroz koje prolazi «*spline*» krivulja.

5.8. start_testfcn

Ova podfunkcija pokreće se svaki put kad ponovno kliknemo tipkom miša na grafu. Poziva se unutar podfunkcije «Punjenje_Datoteke». U njoj se uzima pozicija (koordinata točke) u koju

smo kliknuli, te se provjerava dali se ta točka nalazi u željenom području (područje vanjske ili srednje točke), te ako je u blizini (unutar krušnice radijusa 0.25) u ovisnosti oko koje točke (vanjske ili srednje) program se nastavlja izvoditi pozivom sljedeće podfunkcije.

Pokretanjem podfunkcije uzimaju se trenutne koordinate točke na koju je korisnik kliknuo mišem i spremaju se u varijablu «XYs».

```
00000025     XYs = get(gca, 'CurrentPoint');
```

Globalne varijable koje su korištete su: l, k, t1, t2, t3, t4, h1, h2, h3, h4, h5, h6, h7, h8, h18. «l» i «k» su brojači koji broje koliko puta smo kliknuli na vanjsku točku (l) te koliko puta na srednju točku (k).

«t1», «t2», «t3», «t4» su varijable u koje se spremaju koordinate točke u kojoj je korisnik otpustio tipku miša tj koordinate točke u kojoj se posljednjoj nalazio. Vrijednosti poprimaju u podfunkciji «testfcn». «t1» i «t2» su x i y koordinate srednje točke, a «t3» i «t4» su x i y koordinate vanjske točke.

Provjeravanjem dali su «l» ili «k» veći od jedan provjeravamo dali već postoje traženi objekti ili ih treba definirati kojeg su tipa kako bi ih Matlab mogao koristiti. Ako je jedna od varijabla veća od jedan znači da je korisnik već ranije pomicao jednu od točaka inače je program tek pokrenut.

```
00000040 if (l>0) || (k>0)
00000041     h1 = findobj(gca, 'Type1', 'line1');
00000050 else
00000051     h1=plot([0 0],[0 0], 'k');
00000060 end
```

Navedni isječak iz skupa redova koda prikazuje samo za primjer jedan objek «h1».

Kako bi se onemogućila besmislena pomicanja točki kao npr da se točka pomakne izvan grafa ,a pokazivaš pozicije miša se još uvijek nalazi na grafu morao sam ograničiti područje na kojem će program prihvaćati koordinate pozicije pokazivača miša. Područje je ograničeno na prostor između kružnica radijusa 7.2 i 5.9. Te kružnice prikazane su cijanidnom bojom.

```
00000064 if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)<7.2
00000065     if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)>5.9
```

Ako su zadovoljeni uvjeti da se točka nalazi u navedenom području program se nastavlja.

Ako se želi pomicati vanjsku točku pozicija pokazivača miša se mora nalaziti izvan kružnice radijusa 6.49999, te se brojač «l» povećava za jedan.

```
00000067      if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)>6.49999
00000068          l=1+1;
```

Ako ovo nije prvo pomicanje točaka od pokretanja programa tj « $k>0.5$ » i « $l>1.5$ » ($l>1.5$ zbog toga što smo trenutno pomicanje točke već pribrojili) koordinate srednje i vanjske točke poprimaju vrijednosti pozicija gdje su se nalazile pri zadnjem pomicanju korištenjem podfunkcije «testfcn»

```
00000071      if k>0.5
00000072          XYo(2,1)=t1;
00000073          XYo(2,2)=t2;
00000074      end
00000075      if l>1.5
00000076          XYo(3,1)=t3;
00000077          XYo(3,2)=t4;
00000078      end
```

Kako bi se omogućilo pomicanje točke samo ako kliknemo na nju napravljena su još neka dodatna ograničenja. Međutim kad je točka područje koje je nemoguće pogoditi pokazivačem položaja miša dopušteno je polje tolerancije oko točke veličine 0.25.

```
00000080      if abs (XYs (1,1))< (abs (XYo(3,1))+0.25)
00000081          if abs (XYs (1,1))>(abs (XYo(3,1))-0.25)
00000082              if abs (XYs (1,2))<(abs (XYo(3,2))+0.25)
00000083                  if abs (XYs (1,2))>(abs (XYo(3,2))-0.25)
```

Ako su ispunjeni svi navedeni uvjeti poziva se sljedeća podfunkcija «testfcn» koja prikazuje pomicanje vanjske točke

```
00000084
set(gcf, 'WindowButtonMotionFcn', {@testfcn, XYs, XYo, handles});
```

Ako se klikne na poziciju koja se nalazi na području između kružnica radijusa 5.9 i 6.49999 omogućit će se pokretanje srednje točke na identičan način kako je već objašnjeno za vanjsku točku.

Primjer pomicanja srednje točke prikazan je na «slici 3» koja se nalazi na mjestu gdje sam objašnjavao kakav prozor se otvori pri samom pokretanju programa.

Ako nisu zadovoljeni navedeni uvjeti klikom miša program uzima nove koordinate za koje opet provjerava sve uvjete.

```
00000117          XYs = get(gca, 'CurrentPoint');
```

5.9. stop_testfcn

Ova funkcija prekida interaktivno računanje, a prekida se otpuštanjem tipke miša. Poziva se unutar podfunkcije «Punjenje_Datoteke»

```
00000001 function stop_testfcn(src,eventdata)
00000015     set(gcf, 'WindowButtonMotionFcn', []);
00000016 end
```

5.10. testfcn

Ova funkcija pokreće se svakim pomakom miša dok je stisnuta lijeva tipka. Poziva se unutar podfunkcije «start_testfcn». Omogućuje pomicanje krajnje i srednje točke što uzrokuje promjenu *spline-a* jer on sadrži te točke. Promjenom spline-a najprije geometrijski, a potom i analitički računaju se vrijednosti parametara na ulaznom i izlaznom trokutu brzina na turbinskoj lopatici

Globalni i lokalni argumenti koji se koriste već su ranije objašnjeni u podfunkcijama «Punjenje_Datoteke» i «start_testfcn».

Pokretanjem ove podfunkcije najprije se traže određeni objekti koje se treba pobrisati: «h1» je prva točka koja se nalazi na unutarnjoj kružnici, «h2» je srednja točka, «h3» je vanjska točka, «h4» je spline krivulja, «h5» je mala kružnica oko točke koja se pomiče, «h6» je pravac kroz vanjsku točku paralelan s X-osi, «h7» je pravac koji s prethodnim objektom zatvara kut « ϕ_1 », «h8» je pravac koji s objektom «h6» zatvara kut « ϕ_2 », «h18» je druga dodana točka koja ovisi o kutu beta1. Primjer jednog pronalaženja i brisanja objekta:

```
00000043          h18 = findobj(gca, 'Type18', 'line18');
00000044          delete (h1,h2,h3,h4,h5,h6,h7,h8, h18);
```

Objekti «h14» i «h15» definirani su kao točke kako bi se u nastavku programa mogli pozivati jer za pozivanje moraju biti određenog tj definiranog tipa.

```
00000048          h14=plot([0 0],[0 0], 'k');
00000049          h15=plot([0 0],[0 0], 'k');
```

U nastavku se crtaju kružnice radijusa kojeg je sam korisnik unio u program. Čitanje i korištenje informacija koje korisnik unosi objašnjeno je već ranije kod glavne funkcije «Verzija02».

```
00000060     circle([0,0],r1t,100000,'-');
00000061     hold on;
00000062     circle([0,0],7,100000,'r-');
00000063     circle([0,0],6.5,100,'c:');
```

Kružnica cijanidne boje polumjera 6.5 čini granicu područja pomicanje srednje i vanjske točke

Ako smo u podfunkciji «start_testfcn» kliknuli na točku u području izvan radijusa kružnice 6.49999, znači da pomičemo vanjsku točku te se crta kružnica cijanidne boje radijusa 7.2 koja označava vanjsku granicu područja u kojem požemo pomicati navedenu točku.

```
00000068     if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)>6.49999
00000071         h15=circle([0,0],7.2,100,'c:');
00000074         h14 = findobj(gca,'Type14','line14');
00000075         delete (h14);
```

Kada pomičemo vanjsku točku objekt «h15» se predefinira i nacрта, a objekt «h14» se briše jer nam on sada nije potreban.

Nakon toga se uzimaju koordinate trenutne pozicije i provjerava se dali se još uvijek krećemo unutar prostora koji je predviđen za pomicanje vanjske točke.

```
00000082         if ((pos(1,1))^2+(pos(1,2))^2)^(1/2)<7.2
00000083             if ((pos(1,1))^2+(pos(1,2))^2)^(1/2)>6.49999
```

Ako su uvjeti ispunjeni računa se diferencija pomaka, tj razlika točke u koju smo kliknuli i točke u kojoj se trenutno nalazimo za svaku os posebno.

```
00000086          dx = XYs(1,1) - pos(1,1);
00000087          dy = XYs(1,2) - pos(1,2);
```

Izračunata razlika oduzima se od koordinata vanjske točke i dobivaju se koordinate gdje je nova pozicija vanjske točke

```
00000088          XYo(3,1)= XYo(3,1) - dx;
00000089          XYo(3,2)= XYo(3,2) - dy;
```

Stvaranjem i korištenjem nove matrice koordinata točaka «xy» ostavljamo u nastavku programa«XY» matricu nepromjenjivom.

```
00000095          xy = [];
00000096          xy(1,1)=XY(1,1);
00000097          xy(1,2)=XY(4,1);
00000098          xy(1,3)=XY(2,1);
00000099          xy(1,4)=XY(3,1);
00000100          xy(2,1)=XY(1,2);
00000101          xy(2,2)=XY(4,2);
00000102          xy(2,3)=XY(2,2);
00000103          xy(2,4)=XY(3,2);
```

Varijable «a» i «b» nam služe samo za jednostavniji zapis i manipulaciju tkoordinatama vanjske točke.

```
00000105          a=xy(1,4);
00000106          b=xy(2,4);
```

Ako je točka i nakon pomicanja ostala u području koje je predviđeno za njezino pomicanje stvaraju se objekti «h1», «h18», «h2», «h3», «h4» i «h5», inače se traži nova pozicija te se izračunava razlika pomaka kako je već objašnjeno.

```
00000108          if (a^2+b^2)^(1/2)<7.2
00000109              if (a^2+b^2)^(1/2)>6.5
00000112              h1=plot(xy(1,1),xy(2,1),'ro');
00000113              h18=plot(xy(1,2),xy(2,2),'ro');
00000114              h2=plot(xy(1,3),xy(2,3),'ro');
00000115              h3=plot(xy(1,4),xy(2,4),'ro');
00000116              xys = spline(t,xy,ts);
00000117              h4=plot(xys(1,:),xys(2:,:),'g-');
```

```

00000118
00000119
    h5=circle([xy(1,4),xy(2,4)],0.3,10000,'m-');
00000120
                                hold on

```

Varijable «t3» i «t4» poprimaju x i y vrijednosti koordinate točke ukoliko je ona u dopuštenom području.

```

00000145
                                t3=pos(1,1);
00000146
                                t4=pos(1,2);

```

Na potpuno isti način funkcionira pomicanje srednje točke kako je to već opisano za vanjsku točku uz par sitnih detalja koji je razlikuju. Dopušteno područje pomicanja srednje točke je između kružnica radijusa 6.49999 i 5.9. Umjesto objekta «h15» predefinira se i stvara objekt «h14», a briše objekt «h15».

Koordinatni sustav se crta radi bolje preglednosti i ljepšeg izgleda grafa, iako za samo funkcioniranje programa on nije potreban.

```

00000236    plot([-8.1 8.1],[0 0], 'k');
00000237    plot([0 0],[-8.1 8.1], 'k');

```

Za dobivanje traženog kuta « β_2 » potrebno je izračunati još dva kuta, a to su kutevi « ϕ_1 » i « ϕ_2 »

Kut « ϕ_1 » se računa prema zadnje dvije točke kojima prolazi spline krivulja. To je kut koji zatvara pravac koji je prolazi kroz navedene točke i pravac koji je paralelan X-osi ali prolazi vanjskom točkom. Računa se tako da koristimo trigonometrijsku funkciju «arcus tangens» i iskoristimo razlike točaka po X i Y-osi. Matlab sve kuteve računa u radijanima stoga su pretvoreni u stupnjeve jer se u toj jedinici prikazuju korisniku.

```

00000242    Nxy=size(xys,2);
00000243    DX=xys(1,Nxy)-xys(1,Nxy-1);
00000244    DY=xys(2,Nxy)-xys(2,Nxy-1);
00000245    kutFil=atan(DY/DX)*(180/3.14);

```

Kut « ϕ_2 » je kut između tangente na vanjsku kružnicu i pravca koji je paralelan X-osi, a prolazi vanjskom točkom. Računa se tako da se najprije izračunaju x i y-koordinata koje čine točku koja je sadržana u tangenti «xtan i ytan», te se trigonometrijskom funkcijom «arcus tangens» izračuna traženi kut.

```

00000248    if XYo(3,1)<0

```

```

00000249         xtan=XYo(3,1)+1;
00000250     end
00000251     if XYo(3,1)>0
00000252         xtan=XYo(3,1)-1;
00000253     end
00000254
00000255     if XYo(3,1)==0
00000256         ytan=XYo(3,2);
00000257     else
00000258         ytan=(-XYo(3,1)/XYo(3,2)*(xtan-XYo(3,1)))+XYo(3,2);
00000259     end
00000260     DY2=ytan-XYo(3,2);
00000261     DX2=xtan-XYo(3,1);
00000262     kutFi2=atan(DY2/DX2)*(180/3.14);

```

Kut « β_2 » se računa kao razlika kuteva « φ_1 » i « φ_2 » koja se oduzima od 180° .

```

00000266     beta2=180-abs((kutFi2)-(kutFi1));
00000267     set(handles.text59,'String',beta2);

```

Crtanje objekta «h6» što ga čini pravac paralelan s X-osi te prolazi vanjskom točkom bez obzira dali pomicali vanjsku ili srednju točku.

```

00000271     xtocka1=XYo(3,1)-2;
00000272     xtocka2=XYo(3,1)+2;
00000273     ytocka=XYo(3,2);
00000274     h6=plot([xtocka1 xtocka2],[ytocka ytocka],'k');

```

Crtanje objekta «h7» što ga čini pravac koji s prethodnim objektom zatvara kut « φ_1 ». Duljina prikazanog pravca po X-osi («xpra1» i «xpra2») ovisi o položaju vanjske točke, a duljine po Y-osi («ypra1» i «ypra2») se računaju prema jednadžbi pravca koja prolazi kroz dvije točke.

```

00000279     if XYo(3,2)<3
00000280         if XYo(3,2)>(-3)
00000281             xpra1=XYo(3,1)-1.5;
00000282             xpra2=XYo(3,1)+1.5;
00000283         else
00000284             xpra1=XYo(3,1)-3;
00000285             xpra2=XYo(3,1)+3;
00000286         end
00000287     else
00000288         xpra1=XYo(3,1)-3;
00000289         xpra2=XYo(3,1)+3;

```

```

00000290     end
00000291     ypra1=(DY/DX*(xpra1-XYo(3,1)))+XYo(3,2);
00000292     ypra2=(DY/DX*(xpra2-XYo(3,1)))+XYo(3,2);
00000293     h7=plot ([xpra1 xpra2],[ypra1 ypra2]);

```

Objekt «h8» je tangenta na vanjsku kružnicu koja prolazi vanjskom točkom te s objektom «h6» zatvara kut « φ_2 ». Duljina prikazane tangente po X-osi («xtan1» i «xtan2») ovisi o položaju vanjske točke, a duljine po Y-osi («ytan1» i «ytan2») se računaju prema jednadžbi pravca koja prolazi kroz dvije točke.

```

00000299     if XYo(3,2)<3
00000300         if XYo(3,2)>(-3)
00000301             xtan1=XYo(3,1)-1.5;
00000302             xtan2=XYo(3,1)+1.5;
00000303         else
00000304             xtan1=XYo(3,1)-3;
00000305             xtan2=XYo(3,1)+3;
00000306         end
00000307     else
00000308         xtan1=XYo(3,1)-3;
00000309         xtan2=XYo(3,1)+3;
00000310     end
00000311     ytan1=(-XYo(3,1)/XYo(3,2)*(xtan1-XYo(3,1)))+XYo(3,2);
00000312     ytan2=(-XYo(3,1)/XYo(3,2)*(xtan2-XYo(3,1)))+XYo(3,2);
00000313     h8=plot ([xtan1 xtan2],[ytan1 ytan2]);

```

Ova podfunkcija još računa i sve vrijednosti na ulaznim i izlaznim trokutima brzina te ih ispisiuje na mjesta koja su predviđena za svaku vrijednost, a nalaze se na desnom dijelu prozora kada se pokrene program. (slika 19.). sljedećim primjerom ću pokazati kako se koristi vrijednost koju je korisnik unio u program te kako se određena vrijednost ispisiuje na ekran.

```

00000338 % b-visina rotora
00000339 b=get(handles.edit5,'String');
00000340 b=str2double(b);
00000341
00000342 % omega-kutna brzina
00000343 omega=2*pi*n/60;
00000344 u1=r1*omega;
00000345 set(handles.text32,'String',u1);

```


5.11. circle

Ova podfunkcija nalazi se u zasebnoj datoteci, a poziva se u podfunkcijama: «Punjenje_Datoteke» i «testfcn». Njezina je zadaća crtanje kružnica koje se definiraju središtem kružnice, radijusom i brojem crtica određene boje koje iscrtavaju nesavršenu kružnicu.

Primjer njezinom pozivanja u podfunkciji «Punjenje_Datoteke»:

```
00000028 circle([0,0],7,100000,'r-');
```

Ovim primjerom iscrtava se kružnica sa središtem u koordinatama $S(0,0)$, radijusom $r=7$ i 100000 crtica crvene boje.

6. Zaključak

Krajnji cilj ovog završnog rada je izračunati specifičnu snagu turbostroja koja ovisi o određenim brzinama na ulaznom i izlaznom trokutu brzina. Jedna od potrebnih veličina je i kut relativne brzine « β_2 ». Ova veličina se razlikuje od ostalih po tome što se dobiva iz geometrije interaktivnom manipulacijom izlaznih trokuta brzina. Ujedno se u realnom vremenu proračunava i tražena specifična snaga koja je usko povezana sa kutem relativne brzine na izlaznom trokutu brzina.

Manipulacija izlaznim trokutom brzina vrši se pomicanjem određenih točaka koje određuju izgled krivulje «spline». Ta krivulja predstavlja oblik lopatice u dvodimenzionalnom obliku. Prema tome možemo vidjeti tijekom pomicanja točaka, tj manipulacije izlaznih trokuta brzina u kakvom obliku lopatice dobivamo najveću specifičnu snagu.

Podaci koje korisnik unosi, izlazni rezultati proračuna te grafički prikaz lopatice prikazani su u grafičkom sučelju GUI-u

Smatram da sam program ima još mnogo načina za njegovo poboljšanje kako bi se dobili kvalitetniji podaci pri simulaciji oblika lopatice. Jedan od načina je da povećamo broj točaka koje bi utjecale na oblik krivulje koja se prikazuje. Na taj način bi mogli dobiti realniji oblik lopatice te njezin stvarniji utjecaj na specifičnu snagu.

Sljedeći način je da ne postavimo uvjet u ulaznom trokutu brzina da je kut apsolutne brzine « $\alpha_1=90^\circ$ » te da se i na taj način približimo realnijoj vrijednosti rezultata.

Postoji još jedan način za koji smatram da je mnogo zahtjevniji i složeniji od prethodno navedenih, a to je da se oblik lopatice prikazuje u realnom 3D obliku

-

7. Literatura

1. Pečornik, Miroslav, Osnovi hidrauličkih strojeva, Sveučilište u Zagrebu, Strojarsko-brodogradjevni fakultet Rijeka, Rijeka, 1973.
2. Version 7.9 (R2009b) MATLAB Software Documentation, MathWorks Inc., 2009.
3. <http://www.mathworks.com/matlabcentral/fileexchange/30182-2d-mohrs-circle/content/circle.m>
4. <http://hr.wikipedia.org/>

8. Prilozi

8.1. Argumenti

1. *hObject*- objekt koji u programu predajemo kao «handles»
2. *eventdata*
3. *handles*- objekt u kojem se nalaze sve informacije koje možemo dobiti iz grafičkog sučelja ili koje možemo pridodati tom sučelju
4. *varargin*
5. *r1*- unutarnji polumjer
6. *r1t*- unutarnji polumjer, ali skalirana veličina kako bi se mogao prikazivati na grafu
7. *r2*-vanjski promjer
8. *br*- koristi se u «pushbutton4» i «pushbutton5» za provjeravanje dali je korisnik unio sve potrebne informacije
9. *l,k* -brojaci koji broje koliko puta smo kliknuli na vanjsku točku (l), i koliko puta na srednju točku (k), služe kako bi mogli dijelove programa prilagoditi za pokretanje pri prvom prolazu, odnosno za sljedeće prolaze kad se određeni objekti točno definiraju
10. *t1,t2,t3,t4* -varijable u koje se spremaju koordinate točke u kojoj je korisnik otpustio tipku miša
t1,t2- x i y koordinate srednje točke
t3,t4- xi y koordinate krajnje točke
11. *XYo* -matrica u kojoj se nalaze koordinate točaka koje se koriste za crtanje spline-a
12. *h1 h2 h3 h4 h5 h6 h7 h8 h18* -objekti plotanja
13. *XYs* -varijabla u kojoj se nalaze koordinate točke na koju smo kliknuli mišem

8.2. Originalni kod programa (VERZIJA02)

Kao prilog prilažem cijeli program koji se sastoji od više funkcija, kako je već opisano, a pojedine funkcije se nalaze svaka u svoj istoimenoj datoteci.

8.2.1. VERZIJA02

```
00000001 function varargout = Verzija02(varargin)
00000002 %% Verzija02
00000003 % Ime funkcije: Verzija02
00000004 % Datum trenutne verzije: 13-Jun-2011
00000005 % Autor: Arsen Sušanj
00000006 %Ova funkcija stvara Verzija02.fig, korištenjem GUI-a grafičkog
        sučelja.
00000007 %     Sastoji se od funkcija koje ju pokreću, funkcija za unos
        argumenata od strane
00000008 %     korisnika(edit-funkcije) i funkcija koje postepeno pokreću
        određene dijelove
00000009 %     programa (pushbutton)
00000010 % Argumenti:
00000011 %         Varargin-varijabla
00000012 % Pozivi:
00000013 %     funkcije koje poziva: Verzija02_OpeningFcn,
        Verzija02_OutputFcn,
00000014 %         pushbutton3_Callback,pushbutton4_Callback,
        pushbutton5_Callback
00000015 %         Punjenje_Datoteke
00000016 %     funkcije od kojih je pozvana: -
00000017 % Revizije:
00000018
00000019 %%
00000020 gui_Singleton = 1;
00000021 gui_State = struct('gui_Name',       mfilename, ...
00000022                    'gui_Singleton',  gui_Singleton, ...
00000023                    'gui_OpeningFcn', @Verzija02_OpeningFcn, ...
00000024                    'gui_OutputFcn',  @Verzija02_OutputFcn, ...
00000025                    'gui_LayoutFcn',  [] , ...
00000026                    'gui_Callback',   []);
00000027 if nargin && ischar(varargin{1})
00000028     gui_State.gui_Callback = str2func(varargin{1});
00000029 end
```

```
00000030
00000031 if nargin
00000032     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
00000033 else
00000034     gui_mainfcn(gui_State, varargin{:});
00000035 end
00000036
00000037
00000038 %% OpeningFcn
00000039 % ova funkcija se pokreće pri samom pokretanju programa prije nego
00000040 % "Verzija02" bude vidljiva
00000041 clc;
00000042 function Verzija02_OpeningFcn(hObject, eventdata, handles,
    varargin)
00000043
00000044 handles.output = hObject;
00000045 set(hObject, 'toolbar', 'figure');
00000046 guidata(hObject, handles);
00000047
00000048 set(handles.pushbutton3, 'Visible', 'Off');
00000049 set(handles.pushbutton5, 'Visible', 'Off');
00000050
00000051 %% --- Outputs from this function are returned to the command line.
00000052 function varargout = Verzija02_OutputFcn(hObject, eventdata,
    handles)
00000053 varargout{1} = handles.output;
00000054
00000055 %% pushbotton3 "Interaktivno računanje ostalih vrijednosti"
00000056 % pokreće interaktivno računanje ostalih vrijednosti, te poziva
    sljedeću
00000057 % podfunkciju "Punjenje_Datoteke"
00000058 function pushbutton3_Callback(hObject, eventdata, handles)
00000059
00000060 r1=get(handles.edit2, 'String');
00000061 r1=str2double(r1);
00000062 r2=get(handles.edit3, 'String');
00000063 r2=str2double(r2);
00000064 % r1 i r2 se uzimaju iz unesenih vrijednosti jer su potrebni kao
    varijable
00000065 % za pozivanje podfunkcije "Punjenje_Datoteke"
00000066 Punjenje_Datoteke (handles, r1,r2)
```

```
00000067
00000068 %% edit
00000069 % edit1-edit5 omogućuju unos podataka koji se zahtijevaju od
        korisnika te
00000070 % njihovo daljnje korištenje
00000071 function edit1_Callback(hObject, eventdata, handles)
00000072 function edit1_CreateFcn(hObject, eventdata, handles)
00000073 if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
00000074     set(hObject,'BackgroundColor','white');
00000075 end
00000076
00000077
00000078 function edit2_Callback(hObject, eventdata, handles)
00000079 function edit2_CreateFcn(hObject, eventdata, handles)
00000080 if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
00000081     set(hObject,'BackgroundColor','white');
00000082 end
00000083
00000084
00000085 function edit3_Callback(hObject, eventdata, handles)
00000086 function edit3_CreateFcn(hObject, eventdata, handles)
00000087 if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
00000088     set(hObject,'BackgroundColor','white');
00000089 end
00000090
00000091
00000092 function edit4_Callback(hObject, eventdata, handles)
00000093 function edit4_CreateFcn(hObject, eventdata, handles)
00000094 if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
00000095     set(hObject,'BackgroundColor','white');
00000096 end
00000097
00000098
00000099 function edit5_Callback(hObject, eventdata, handles)
00000100 function edit5_CreateFcn(hObject, eventdata, handles)
00000101 if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
```

```
00000102     set(hObject,'BackgroundColor','white');
00000103 end
00000104
00000105 %% pushbutton4 "provjera ulaznih podataka"
00000106 %provjerava dali su upisani svi podaci koje treba korisnik upisati.
        Ako nesto nedostaje
00000107 %u taj "Edit Text" se ispisuje "upisi podatke", inače se
        pojavljuje sljedeći
00000108 % pushbotton3 "Interaktivno računanje ostalih vrijednosti"
00000109 % br je brojac koi se povećava za jedan za svaki "edit text" koji
        je
00000110 % razkičit od "-" te ako je on jednak broju mjesta za unos
        vrijednosti,
00000111 % program se nastavlja izvoditi
00000112 function pushbutton4_Callback(hObject, eventdata, handles)
00000113 %----provjera dali su upisane vrijednosti ili jos uvijek stoje
        crtice, tj
00000114 %korisnik nije upisao vrijednost
00000115 br=0;
00000116
00000117 if get(handles.edit1,'String')== '-'
00000118     set (handles.edit1, 'String','upisi vrijednost');
00000119 else
00000120     br=br+1;
00000121 end
00000122 %-----
00000123 if get(handles.edit2,'String')== '-'
00000124     set (handles.edit2, 'String','upisi vrijednost');
00000125 else
00000126     br=br+1;
00000127 end
00000128 %-----
00000129 if get(handles.edit3,'String')== '-'
00000130     set (handles.edit3, 'String','upisi vrijednost');
00000131 else
00000132     br=br+1;
00000133 end
00000134 %-----
00000135 if get(handles.edit4,'String')== '-'
00000136     set (handles.edit4, 'String','upisi vrijednost');
00000137 else
```



```
00000138     br=br+1;
00000139 end
00000140 %-----
00000141 if get(handles.edit5,'String')== '-'
00000142     set(handles.edit5, 'String','upisi vrijednost');
00000143 else
00000144     br=br+1;
00000145 end
00000146 %-----
00000147
00000148 r1=get(handles.edit2,'String');
00000149 r1=str2double(r1);
00000150 r2=get(handles.edit3,'String');
00000151 r2=str2double(r2);
00000152
00000153 if br<5
00000154     set(handles.pushbutton5, 'Visible', 'On');
00000155     set(handles.pushbutton4, 'Visible', 'Off');
00000156 end
00000157
00000158 if r1 > (0.75*r2)
00000159     r1='prevelik r1';set(handles.edit2,'String',r1);
00000160     r2='premali r2';set(handles.edit3,'String',r2);
00000161     set(handles.pushbutton5, 'Visible', 'On');
00000162     set(handles.pushbutton4, 'Visible', 'Off');
00000163 end
00000164
00000165 if br==5
00000166     if r1 < (0.7500000001*r2)
00000167         set(handles.pushbutton3, 'Visible', 'On');
00000168         set(handles.pushbutton4, 'Visible', 'Off');
00000169     end
00000170 end
00000171
00000172
00000173
00000174 %% pushbutton5 "provjera ulaznih podataka"
00000175 % provjera dali su upisane vrijednosti, tj dali je korisnik upisao
    vrijednost
00000176 % radi na principu da provjerava dali na mjestu za unos teksta
    stoji
```

```
00000177 % string "upisi vrijednost","premali r1" ili "preveliki r2"
00000178 % ako su ispunjeni zadani uvjeti dopušta se daljnje izvođenje
           programa
00000179 function pushbutton5_Callback(hObject, eventdata, handles)
00000180
00000181 br=5
00000182 str1=get(handles.edit1,'String');
00000183 if isnan(str2double(str1))
00000184     br=br-1
00000185 end
00000186 %-----
00000187 str2=get(handles.edit2,'String');
00000188 if isnan(str2double(str2))
00000189     br=br-1
00000190 end
00000191 %-----
00000192 str3=get(handles.edit3,'String');
00000193 if isnan(str2double(str3))
00000194     br=br-1
00000195 end
00000196 %-----
00000197 str4=get(handles.edit4,'String');
00000198 if isnan(str2double(str4))
00000199     br=br-1
00000200 end
00000201 %-----
00000202 str5=get(handles.edit5,'String');
00000203 if isnan(str2double(str5))
00000204     br=br-1
00000205 end
00000206 %-----
00000207
00000208 r1=get(handles.edit2,'String');
00000209 r1=str2double(r1);
00000210 r2=get(handles.edit3,'String');
00000211 r2=str2double(r2);
00000212
00000213 if (r1>(0.75*r2))
00000214     r1='prevelik r1';set(handles.edit2,'String',r1);
00000215     r2='premali r2';set(handles.edit3,'String',r2);
00000216
```

```
00000217 end
00000218
00000219 if br==5
00000220     if r1 < (0.75000001*r2)
00000221         set(handles.pushbutton3, 'Visible', 'On');
00000222         set(handles.pushbutton5, 'Visible', 'Off');
00000223     end
00000224 end
```

8.2.2. Punjenje_Datoteke

```
00000002 %% Punjenje_Datoteke
00000003 % Ime funkcije: Punjenje_Datoteke
00000004 % Datum trenutne verzije: 13-Jun-2011
00000005 % Autor: Arsen Sušanj
00000006 % Unutar ove funkcije izračunavaju se prvobitne vrijednosti točaka
      kroz
00000007 %     koje se crta spline, te definira neke globalne varijable
      koje su
00000008 %     potrebne za daljnje izvođenje programa koji se nastavlja
00000009 %     pozivanjem niže navedenih podfunkcija
00000010 % Argumenti:
00000011 %     handles -objekt u kojem se nalaze sve informacije koje
      možemo
00000012     % dobiti iz grafičkog sučelja ili koje možemo
      pridodati tom sučelju
00000013 %     r1 -polumjeri unutarnje kružnice
00000014 %     r2 -polumjeri vanjske kružnice
00000015 % Pozivi:
00000016 %     funkcije koje poziva: start_testfcn, stop_testfcn
00000017 %     funkcije od kojih je pozvana: verzija02
00000018 % Revizije:
00000019
00000020 %% fiksne kružnice
00000021 % crtanje kružnica u omjeru koji je korisnik unio u program, ali
      vanjska
00000022 % kružnica promjera r2 ce prikazuje kao r2t=7, a unutarnja
      kružnica se crta
00000023 % u omjeru r1:r2 s promjerom r1t
00000024 r2t=7;
00000025 r1t=r1*7/r2;
```

```
00000026 circle([0,0],r1t,100000,'-');
00000027 hold on;
00000028 circle([0,0],7,100000,'r-');
00000029
00000030 %% INTERACTIVE GRAPHING
00000031 % defuniranje veličine grafa i globalnih varijabli
00000032 axis([-8.2 8.2 -8.2 8.2]);
00000033 hold on
00000034 global l k t1 t2 t3 t4;
00000035 % l,k -brojaci koji broje koliko puta smo kliknuli na vanjsku točku
        (l), i
00000036     % koliko puta na srednju točku (k), služe kako bi mogli
        dijelove programa
00000037     % prilagoditi za pokretanje pri prvom prolazu, odnosno za
        sljedeće prolaze kad se
00000038     % određeni objekti točno definiraju
00000039 % t1,t2,t3,t4 -varijable u koje se spremaju koordinate točke u
        kojoj je
00000040 % korisnik otpustio tipku miša
00000041     %t1,t2- x i y koordinate srednje točke
00000042     %t3,t4- xi y koordinate krajnje točke
00000043 l=0;
00000044 k=0;
00000045 xy = []; %prvobitna matrica u koju se pohranjuju koordinate točaka
00000046 %% Točke
00000047 % Definiranje koordinata točaka koje se koriste za crtanje spline-
        a a
00000048 % koordinate se spremaju u XY-matricu
00000049 %--Prva tocka-----
00000050 i=1;
00000051 xi=0;
00000052 mn1=xi;
00000053     % xi, yi-pohranjuju se u mn1 i mn2 kako bi se njihove
        vrijednosti
00000054     % mogle koristiti pri računanju pozicije sljedeće točke
00000055 yi=(r1t^2-xi^2)^(1/2);
00000056 mn2=yi;
00000057 h11=plot(xi,yi,'ro');
00000058     xy(:,i) = [xi;yi];
00000059 XY(1,1)=xi;
00000060 XY(1,2)=yi;
```

```
00000061 %--Druga dodana tocka-----
00000062 i=2;
00000063     % Uzimanje vrijednosti koje je unio korisnik jer se njezine
           koordinate
00000064     % dobivaju pomoću kuta beta1
00000065 Q=get(handles.edit4,'String');
00000066 Q=str2double(Q);
00000067 b=get(handles.edit5,'String');
00000068 b=str2double(b);
00000069 A1=2*r1*pi*b;
00000070 v1=Q/A1;
00000071 n=get(handles.edit1,'String');
00000072 n=str2double(n);
00000073 omega=2*pi*n/60;
00000074 u1=r1*omega;
00000075 beta1=pi-atan(v1/u1);
00000076 ddy=abs(0.05*sin(pi-beta1));
00000077 ddx=abs(0.05*cos(pi-beta1));
00000078
00000079 xi=mn1-ddx;
00000080 yi=mn2+ddy;
00000081     h19=plot(xi,yi,'ro');
00000082     xy(:,i) = [xi;yi];
00000083 XY(4,1)=xi;
00000084 XY(4,2)=yi;
00000085 %--Druga tocka-----
00000086 i=3;
00000087 xi=-3.35;
00000088 yi=4.92;
00000089 h12=plot(xi,yi,'ro');
00000090 xy(:,i) = [xi;yi];
00000091 XY(2,1)=xi;
00000092 XY(2,2)=yi;
00000093 r2=(xi^2+yi^2)^(1/2);
00000094 %--Trecja tocka-----
00000095 i=4;
00000096 xi=-6;
00000097 yi=(49-xi^2)^(1/2);
00000098     h13=plot(xi,yi,'ro');
00000099     xy(:,i) = [xi;yi];
00000100 XY(3,1)=xi;
```

```
00000101 XY(3,2)=yi;
00000102
00000103 %% spline
00000104 % crta spline kroz zadane točke koji je definiran u objektu h10
00000105 t= 1:i;
00000106 ts = 1:0.1:i;
00000107 xys = spline(t,xy,ts);
00000108 h10=plot(xys(1,:),xys(2:),'g-');
00000109 hold off
00000110
00000111 %% POMICANJE
00000112 % ovdje se pozivaju podfunkcije koje služe za pokretanje (klikom
        miša) i
00000113 % zaustavljanje (otpuštanjem miša) interaktivnog grafičkog prikaza
        i
00000114 % analitičkog računanja zadanih parametara
00000115 Xo = XY(:,1);
00000116 Yo = XY(:,2);
00000117 XYo=XY;
00000118 set(gcf, 'WindowButtonDownFcn',{@start_testfcn,XYo,handles});
00000119 set(gcf, 'WindowButtonUpFcn',@stop_testfcn);
00000001 function Punjenje_Datoteke (handles, r1, r2)
```

8.2.3. start_testfcn

```
00000001 function start_testfcn(src,eventdata,XYo,handles)
00000002 %% testfcn
00000003 % Ime funkcije: testfcn
00000004 % Datum trenutne verzije: 13-Jun-2011
00000005 % Autor: Arsen Sušanj
00000006 % Ova funkcija pokreće se svaki put kad ponovno kliknemo tipkom miša
      na grafu. U
00000007 %      njoj se uzima pozicija (koordinata točke) u kojoj smo
      kliknuli, te
00000008 %      se provjerava dali se ta točka nalazi u željenom području
      (područje vanjske
00000009 %      ili srednje točke), te ako je u blizini (unutar krušnice
      radijusa
00000010 %      0.25) u ovisnosti oko koje točke program se nastavlja
      izvoditi
00000011 %      pozivom sljedeće podfunkcije
00000012 % Argumenti:
00000013     % src
00000014     % eventdata
00000015     % XYo -matrica u kojoj se nalaze koordinate točaka koje se
      koriste
00000016     % za crtanje spline-a
00000017     % handles -objekt u kojem se nalaze sve informacije koje
      možemo
00000018     % dobiti iz grafičkog sučelja ili koje možemo
      pridodati tom sučelju
00000019 % Pozivi:
00000020 %     funkcije koje poziva: testfcn
00000021 %     funkcije od kojih je pozvana: Punjenje_Datoteke
00000022 % Revizije:
00000023 %% globalne varijable
00000024     % uzimanje koordinata pozicije gdje mišem kliknemo na grafu
00000025     XYs = get(gca, 'CurrentPoint');
00000026 global l k t1 t2 t3 t4 h1 h2 h3 h4 h5 h6 h7 h8 h18;
00000027 % l,k -brojaci koje broje kolko puta smo kliknuli na vanjsku točku
      (l), i
00000028     % kolko puta na srednju točku (k), služe kako bi mogli dijelove
      programa
```

```
00000029 % prilagoditi za pokretanje pri prvom prolazu, odnosno za
           sljedeće prolaze kad se
00000030 % određeni objekti točno definiraju
00000031 % t1,t2,t3,t4 -varijable u koje se spremaju koordinate točke u kojoj
           je
00000032 % korisnik otpustio tipku miša
00000033 %t1,t2- x i y koordinate srednje točke
00000034 %t3,t4- xi y koordinate krajnje točke
00000035 % h1 h2 h3 h4 h5 h6 h7 h8 h18 -objekti plotanja
00000036 %% definiranje objekata
00000037 % ako su l i k veći od nule pronalaze se objekti inače se ih
           definira kao
00000038 % tocke u ishodištu samo kako bi ih matlab prepoznao kao objekte u
           prvom
00000039 % prolazu kroz testfcn -funkciju
00000040 if (l>0) || (k>0)
00000041     h1 = findobj(gca,'Type1','line1');
00000042     h2 = findobj(gca,'Type2','line2');
00000043     h3 = findobj(gca,'Type3','line3');
00000044     h4 = findobj(gca,'Type4','line4');
00000045     h5 = findobj(gca,'Type5','line5');
00000046     h6 = findobj(gca,'Type6','line6');
00000047     h7 = findobj(gca,'Type7','line7');
00000048     h8 = findobj(gca,'Type8','line8');
00000049     h18 = findobj(gca,'Type18','line18');
00000050 else
00000051     h1=plot([0 0],[0 0],'k');
00000052     h2=plot([0 0],[0 0],'k');
00000053     h3=plot([0 0],[0 0],'k');
00000054     h4=plot([0 0],[0 0],'k');
00000055     h5=plot([0 0],[0 0],'k');
00000056     h6=plot([0 0],[0 0],'k');
00000057     h7=plot([0 0],[0 0],'k');
00000058     h8=plot([0 0],[0 0],'k');
00000059     h18=plot([0 0],[0 0],'k');
00000060 end
00000061
00000062 %% uvjeti za pomicanje točke
00000063 % tocka ne smije biti izvan gornje i ispod donje granice područja u
           kojem se može pomicati
00000064 if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)<7.2
```



```
00000065     if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)>5.9
00000066         % uvjet za ograničenje područja u kojem se pomiče vanjska
točka
00000067         if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)>6.49999
00000068             l=l+1;
00000069             % u ovisnosti o k i l točkama se pridružuju koordinate
njihove zadnje
00000070             % pozicije uslijed pomicanja unutar podfunkcije testfcn
00000071             if k>0.5
00000072                 XYo(2,1)=t1;
00000073                 XYo(2,2)=t2;
00000074             end
00000075             if l>1.5
00000076                 XYo(3,1)=t3;
00000077                 XYo(3,2)=t4;
00000078             end
00000079             % uvjet da je unutar 0.25 oko crvenog kruga za vanjsku točku
00000080             if abs (XYs (1,1))< (abs(XYo(3,1))+0.25)
00000081                 if abs (XYs (1,1))>(abs (XYo(3,1))-0.25)
00000082                     if abs (XYs (1,2))<(abs(XYo(3,2))+0.25)
00000083                         if abs (XYs (1,2))>(abs(XYo(3,2))-0.25)
00000084
00000085                 set(gcf,'WindowButtonMotionFcn',{@testfcn,XYs,XYo,handles});
00000086             end
00000087         end
00000088     end
00000089 end
00000090 % uvjet za ograničenje područja u kojem se pomiče srednja točka
00000091 if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)<6.49999
00000092     k=k+1;
00000093     % u ovisnosti o k i l točkama se pridružuju koordinate
njihove zadnje
00000094     % pozicije uslijed pomicanja unutar podfunkcije testfcn
00000095     if l>0.5
00000096         XYo(3,1)=t3;
00000097         XYo(3,2)=t4;
00000098     end
00000099     if k>1.5
00000100         XYo(2,1)=t1;
00000101         XYo(2,2)=t2;
```

```

00000102         end
00000103     % uvjet da je unutar 0.25 oko crvenog kruga za srednju točku
00000104         if abs (XYs (1,1)) < (abs(XYo(2,1))+0.25)
00000105             if abs (XYs (1,1)) > (abs (XYo(2,1))-0.25)
00000106                 if abs (XYs (1,2)) < (abs(XYo(2,2))+0.25)
00000107                     if abs (XYs (1,2)) > (abs(XYo(2,2))-0.25)
00000108
00000109                         set(gcf, 'WindowButtonMotionFcn', {@testfcn, XYs, XYo, handles});
00000110                             end
00000111                     end
00000112                 end
00000113             end
00000114         else
00000115             % ponovno se uzimaju koordinate točke ako nije unutar
00000116             područja
00000117             % krzžnica s radijusima 7.2 i 5.9
00000118             XYs = get(gca, 'CurrentPoint');
00000119         end
00000120     end

```

8.2.4. stop_testfcn

```

00000001 function stop_testfcn(src,eventdata)
00000002 % Ime funkcije: stop_testfcn
00000003 % Datum trenutne verzije: 13-Jun-2011
00000004 % Autor: Arsen Sušanj
00000005 % Ova funkcija prekida interaktivno računanje, a pokreće se
00000006     otpuštanjem
00000007 % klika miša
00000008 % Argumenti:
00000009 %     src
00000010 %     eventdata
00000011 % Pozivi:
00000012 %     funkcije koje poziva: -
00000013 %     funkcije od kojih je pozvana: Punjenje_Datoteke
00000014 % Revizije:-
00000015
00000016     set(gcf, 'WindowButtonMotionFcn', []);
00000017 end

```

8.2.5. testfcn

```
00000001 function testfcn(src,eventdata,XYs,XYo,handles)
00000002 %% testfcn
00000003 % Ime funkcije: testfcn
00000004 % Datum trenutne verzije: 13-Jun-2011
00000005 % Autor: Arsen Sušanj
00000006 % Ova funkcija pokreće se svakim pomakom miša dok je stisnuta
    lijeva tipka.
00000007 %         Omogućuje pomicanje krajnje i srednje točke što uzrokuje
    promjenu
00000008 %         spline-a jer on sadrži te točke. Promjenom spline-a
    najprije
00000009 %         geometrijski, a potom i analitički računaju se vrijednosti
00000010 %         parametara na ulaznom i izlaznom trokutu brzina na
    turbinskoj
00000011 %         lopatici
00000012 % Argumenti:
00000013         % src
00000014         % eventdata
00000015         % XYs -varijabla u kojoj se nalaze koordinate točke na
    koju smo
00000016         %kliknuli mišem
00000017         % XYo -matrica u kojoj se nalaze koordinate točaka koje se
    koriste
00000018         % za crtanje spline-a
00000019         % handles -objekt u kojem se nalaze sve informacije koje
    možemo
00000020         % dobiti iz grafičkog sučelja ili koje možemo
    pridodati tom sučelju
00000021 % Pozivi:
00000022 %     funkcije koje poziva: -
00000023 %     funkcije od kojih je pozvana: start_testfcn
00000024 % Revizije:
00000025 %% globalne varijable
00000026     global t1 t2 t3 t4 h1 h2 h3 h4 h5 h6 h7 h8 h18;
00000027 % t1,t2,t3,t4 -varijable u koje se spremaju koordinate točke u
    kojoj je
00000028 % korisnik otpustio tipku miša
00000029         %t1,t2- x i y koordinate srednje točke
00000030         %t3,t4- xi y koordinate krajnje točke
00000031 % h1 h2 h3 h4 h5 h6 h7 h8 h18 -objekti plotanja
```

```
00000032 %% objekti
00000033 % pronalaženje i brisanje starih (postojećih) objekata i njihovo
        brisanje
00000034 % kako bi se dobila preglednost novih objekata koji će tek biti
        nacrtani
00000035         h1 = findobj(gca, 'Type1', 'line1');
00000036         h2 = findobj(gca, 'Type2', 'line2');
00000037         h3 = findobj(gca, 'Type3', 'line3');
00000038         h4 = findobj(gca, 'Type4', 'line4');
00000039         h5 = findobj(gca, 'Type5', 'line5');
00000040         h6 = findobj(gca, 'Type6', 'line6');
00000041         h7 = findobj(gca, 'Type7', 'line7');
00000042         h8 = findobj(gca, 'Type8', 'line8');
00000043         h18 = findobj(gca, 'Type18', 'line18');
00000044         delete (h1,h2,h3,h4,h5,h6,h7,h8, h18);
00000045 %%
00000046 % definiranje objekata h14 i h15 kako bi matlabu bile definirane
        kao
00000047 % poznate varijable koje može naknadno koristiti
00000048         h14=plot([0 0],[0 0], 'k');
00000049         h15=plot([0 0],[0 0], 'k');
00000050 % definiranje veličine grafa
00000051         axis([-8.2 8.2 -8.2 8.2]);
00000052         hold on
00000053 % uzimanje veličene radijusa koje su unešene u program
00000054         r1=get(handles.edit2, 'String');
00000055         r1=str2double(r1);
00000056         r2=get(handles.edit3, 'String');
00000057         r2=str2double(r2);
00000058         r1t=r1*7/r2;
00000059 % crtanje fiksnih kružnica u grafu
00000060         circle([0,0],r1t,100000, '-');
00000061         hold on;
00000062         circle([0,0],7,100000, 'r-');
00000063         circle([0,0],6.5,100, 'c:');
00000064
00000065 %% pomicanje krajnje točke
00000066 % provjera dali se točka koju je korisnik kliknuo nalazi na
        području
00000067 % pomicanja srednje ili krajnje točke
00000068         if ((XYs(1,1))^2+(XYs(1,2))^2)^(1/2)>6.49999
```

```
00000069      % crtanje kružnice koja ograničava područje pomicanja
              krajnje točke s
00000070      % njegove vanjske strane
00000071      h15=circle([0,0],7.2,100,'c:');
00000072      % pronalaženje i brisanje objekta koji se ne koristi kada je
              trenutna pozicija u
00000073      % u području vanjske točke
00000074      h14 = findobj(gca,'Type14','line14');
00000075      delete (h14);
00000076      % trenutno uzimanje pozicije gdje se nalazimo s strelicom
              miša kad
00000077      % je stisnuta lijeva tipka
00000078      pos = get(gca, 'CurrentPoint');
00000079      % uvjet koji ograničuje pomicanje vanjske točke unutar
              područja
00000080      % kružnica radijusa 7.2 i 6.4999 provjerom gdje se
              trenutno nalazi
00000081      % strelica miša
00000082      if ((pos(1,1))^2+(pos(1,2))^2)^(1/2)<7.2
00000083          if ((pos(1,1))^2+(pos(1,2))^2)^(1/2)>6.49999
00000084              % računanje razlike pozicija gdje smo kliknuli
              mišem i gdje
00000085              % se trenutno nalazimo
00000086              dx = XYs(1,1) - pos(1,1);
00000087              dy = XYs(1,2) - pos(1,2);
00000088              XYo(3,1)= XYo(3,1) - dx;
00000089              XYo(3,2)= XYo(3,2) - dy;
00000090              XY=XYo;
00000091              i=4;
00000092              t= 1:i;
00000093              ts = 1:0.1:i;
00000094
00000095              xy = [];
00000096              xy(1,1)=XY(1,1);
00000097              xy(1,2)=XY(4,1);
00000098              xy(1,3)=XY(2,1);
00000099              xy(1,4)=XY(3,1);
00000100              xy(2,1)=XY(1,2);
00000101              xy(2,2)=XY(4,2);
00000102              xy(2,3)=XY(2,2);
00000103              xy(2,4)=XY(3,2);
```

```
00000104           % a i b služe samo za jednostavniji daljni zapis
00000105           a=xy(1,4);
00000106           b=xy(2,4);
00000107           %provjera dali je točka koju pomičemo još uvijek u
predviđenom prostoru
00000108           if (a^2+b^2)^(1/2)<7.2
00000109               if (a^2+b^2)^(1/2)>6.5
00000110                   % crtanje točaka kroz koje prolazi spline
sa novim pozicijama, te
00000111                   % crtanje i samog spline-a kroz te nove
točke
00000112                   h1=plot(xy(1,1),xy(2,1),'ro');
00000113                   h18=plot(xy(1,2),xy(2,2),'ro');
00000114                   h2=plot(xy(1,3),xy(2,3),'ro');
00000115                   h3=plot(xy(1,4),xy(2,4),'ro');
00000116                   xys = spline(t,xy,ts);
00000117                   h4=plot(xys(1,:),xys(2,),'g-');
00000118
00000119                   h5=circle([xy(1,4),xy(2,4)],0.3,10000,'m-
');
00000120
00000121                   hold on
00000122           else
00000123               pos = get(gca, 'CurrentPoint');
00000124               dx = XYs(1,1) - pos(1,1);
00000125               dy = XYs(1,2) - pos(1,2);
00000126               XYo(3,1)= XYo(3,1) - dx;
00000127               XYo(3,2)= XYo(3,2) - dy;
00000128               XY=XYo;
00000129               i=4;
00000130               t= 1:i;
00000131               ts = 1:0.1:i;
00000132
00000133               xy = [];
00000134               xy(1,1)=XY(1,1);
00000135               xy(1,2)=XY(4,1);
00000136               xy(1,3)=XY(2,1);
00000137               xy(1,4)=XY(3,1);
00000138               xy(2,1)=XY(1,2);
00000139               xy(2,2)=XY(4,2);
00000140               xy(2,3)=XY(2,2);
00000141               xy(2,4)=XY(3,2);
```

```
00000141             a=xy(1,4);
00000142             b=xy(2,4);
00000143             end
00000144         end
00000145             t3=pos(1,1);
00000146             t4=pos(1,2);
00000147         else
00000148             pos = get(gca, 'CurrentPoint');
00000149         end
00000150     end
00000151 %% pomicanje srednje točke
00000152     else
00000153         % crtanje kružnice koja ograničava područje pomicanja
srednje točke s
00000154         % njegove unutarnje strane
00000155         h14=circle([0,0],5.9,100,'c:');
00000156         % brisanje objekta koji se ne koristi kada je trenutna
pozicija u
00000157         %području srednje točke
00000158         delete (h15);
00000159         % trenutno uzimanje pozicije gdje se nalazimo s strelicom
miša kad
00000160         % je stisnuta lijeva tipka
00000161         pos = get(gca, 'CurrentPoint');
00000162         % uvjet koji ograničuje pomicanje srednje točke unutar
područja
00000163         % kružnica radijusa 5.9 i 6.4999 provjerom gdje se
trenutno nalazi
00000164         % strelica miša
00000165         if ((pos(1,1))^2+(pos(1,2))^2)^(1/2)<6.49999
00000166             if ((pos(1,1))^2+(pos(1,2))^2)^(1/2)>5.9
00000167                 % računanje razlike pozicija gdje smo kliknuli
mišem i gdje
00000168                 % se trenutno nalazimo
00000169                 dx = XYs(1,1) - pos(1,1);
00000170                 dy = XYs(1,2) - pos(1,2);
00000171                 XYo(2,1)= XYo(2,1) - dx;
00000172                 XYo(2,2)= XYo(2,2) - dy;
00000173                 XY=XYo;
00000174                 i=4;
00000175                 t= 1:i;
```

```
00000176         ts = 1:0.1:i;
00000177
00000178         xy = [];
00000179         xy(1,1)=XY(1,1);
00000180         xy(1,2)=XY(4,1);
00000181         xy(1,3)=XY(2,1);
00000182         xy(1,4)=XY(3,1);
00000183         xy(2,1)=XY(1,2);
00000184         xy(2,2)=XY(4,2);
00000185         xy(2,3)=XY(2,2);
00000186         xy(2,4)=XY(3,2);
00000187         % a i b služe samo za jednostavniji daljni zapis
00000188         a=xy(1,3);
00000189         b=xy(2,3);
00000190         %provjera dali je točka koju pomičemo još uvijek u
predviđenom prostoru
00000191         if (a^2+b^2)^(1/2)<6.5
00000192             if (a^2+b^2)^(1/2)>5.9
00000193                 % crtanje točaka kroz koje prolazi spline
sa novim pozicijama, te
00000194                 % crtanje i samog spline-a kroz te nove
točke
00000195                 h1=plot(xy(1,1),xy(2,1),'ro');
00000196                 h18=plot(xy(1,2),xy(2,2),'ro');
00000197                 h2=plot(xy(1,3),xy(2,3),'ro');
00000198                 h3=plot(xy(1,4),xy(2,4),'ro');
00000199                 xys = spline(t,xy,ts);
00000200                 h4=plot(xys(1,:),xys(2,),'g-');
00000201                 h5=circle([xy(1,3),xy(2,3)],0.3,10000,'m-
');
00000202                 hold on
00000203             else
00000204                 pos = get(gca, 'CurrentPoint');
00000205                 dx = XYs(1,1) - pos(1,1);
00000206                 dy = XYs(1,2) - pos(1,2);
00000207                 XYo(2,1)= XYo(2,1) - dx;
00000208                 XYo(2,2)= XYo(2,2) - dy;
00000209                 XY=XYo;
00000210                 i=4;
00000211                 t= 1:i;
00000212                 ts = 1:0.1:i;
```



```
00000213
00000214             xy = [];
00000215             xy(1,1)=XY(1,1);
00000216             xy(1,2)=XY(4,1);
00000217             xy(1,3)=XY(2,1);
00000218             xy(1,4)=XY(3,1);
00000219             xy(2,1)=XY(1,2);
00000220             xy(2,2)=XY(4,2);
00000221             xy(2,3)=XY(2,2);
00000222             xy(2,4)=XY(3,2);
00000223             a=xy(1,3);
00000224             b=xy(2,3);
00000225             end
00000226             end
00000227             t1=pos(1,1);
00000228             t2=pos(1,2);
00000229             else
00000230             pos = get(gca, 'CurrentPoint');
00000231             end
00000232             end
00000233             end
00000234 %% koordinatni sustav
00000235 % crtanje koordinatng sustava čije je ishodište središte kružnica
00000236     plot([-8.1 8.1],[0 0], 'k');
00000237     plot([0 0],[-8.1 8.1], 'k');
00000238     hold on
00000239 %% kutevi
00000240 % izračunavanje kuteva koji nam služe za dobivanje traženog kuta
    beta2
00000241 % kut Fi1
00000242     Nxy=size(xys,2);
00000243     DX=xys(1,Nxy)-xys(1,Nxy-1);
00000244     DY=xys(2,Nxy)-xys(2,Nxy-1);
00000245     kutFi1=atan(DY/DX)*(180/3.14);
00000246
00000247 % kut Fi2
00000248     if XYo(3,1)<0
00000249         xtan=XYo(3,1)+1;
00000250     end
00000251     if XYo(3,1)>0
00000252         xtan=XYo(3,1)-1;
```

```
00000253     end
00000254
00000255     if XYo(3,1)==0
00000256         ytan=XYo(3,2);
00000257     else
00000258         ytan=(-XYo(3,1)/XYo(3,2)*(xtan-XYo(3,1)))+XYo(3,2);
00000259     end
00000260     DY2=ytan-XYo(3,2);
00000261     DX2=xtan-XYo(3,1);
00000262     kutFi2=atan(DY2/DX2)*(180/3.14);
00000263 %% beta2
00000264 % izračunavanje kuta beta2 pomoću kuteva kutFil i kutFi2, te
        ispisivanje na
00000265 % grafičkom sučelju u mjesto za to predviđeno
00000266     beta2=180-abs((kutFi2)-(kutFil));
00000267     set(handles.text59,'String',beta2);
00000268 %% pomaknuta x-os u točku
00000269 % crta se pravac paralelan X-osi koji prolazi točkom u kojoj
        spline
00000270 % dodiruje vanjsku kružnicu polumjera r2
00000271     xtocka1=XYo(3,1)-2;
00000272     xtocka2=XYo(3,1)+2;
00000273     ytocka=XYo(3,2);
00000274     h6=plot ([xtocka1 xtocka2],[ytocka ytocka],'k');
00000275     hold on
00000276 %% pravca
00000277 % pravac služi za provjeru veličine kuta kutFil te prolazi točkom
        u kojoj
00000278 % spline dira vanjsku kružnicu polumjera r2
00000279     if XYo(3,2)<3
00000280         if XYo(3,2)>(-3)
00000281             xpra1=XYo(3,1)-1.5;
00000282             xpra2=XYo(3,1)+1.5;
00000283         else
00000284             xpra1=XYo(3,1)-3;
00000285             xpra2=XYo(3,1)+3;
00000286         end
00000287     else
00000288         xpra1=XYo(3,1)-3;
00000289         xpra2=XYo(3,1)+3;
00000290     end
```

```
00000291     ypra1=(DY/DX*(xpra1-XYo(3,1)))+XYo(3,2);
00000292     ypra2=(DY/DX*(xpra2-XYo(3,1)))+XYo(3,2);
00000293     h7=plot ([xpra1 xpra2],[ypra1 ypra2]);
00000294     hold on
00000295 %% crtanje tangente
00000296 % tangenta se crta na kružnicu radijusa r2 u točki u kojoj se
           trenutno
00000297 % nalazi spline u dodiru s tom kružnicom te služi za provjeru
           veličine
00000298 % kuta kutFi2
00000299     if XYo(3,2)<3
00000300         if XYo(3,2)>(-3)
00000301             xtan1=XYo(3,1)-1.5;
00000302             xtan2=XYo(3,1)+1.5;
00000303         else
00000304             xtan1=XYo(3,1)-3;
00000305             xtan2=XYo(3,1)+3;
00000306         end
00000307     else
00000308         xtan1=XYo(3,1)-3;
00000309         xtan2=XYo(3,1)+3;
00000310     end
00000311     ytan1=(-XYo(3,1)/XYo(3,2)*(xtan1-XYo(3,1)))+XYo(3,2);
00000312     ytan2=(-XYo(3,1)/XYo(3,2)*(xtan2-XYo(3,1)))+XYo(3,2);
00000313     h8=plot ([xtan1 xtan2],[ytan1 ytan2]);
00000314     hold off
00000315
00000316     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
00000317 %% racunanje ulaznih vrijednosti
00000318 % koristeći vrijednosti koje su unesene u program od strane
           korisnika
00000319 % izračunavaju se ostale vrijednosti na unutarnjem radijusu
           lopatice (r1),
00000320 % tj na ulaznom trokutu brzina
00000321
00000322 % n-broj okretaja po minuti
00000323 n=get(handles.edit1,'String');
00000324 n=str2double(n);
00000325
00000326 % r1-unutarnji promjer
00000327 r1=get(handles.edit2,'String');
```

```
00000328 r1=str2double(r1);
00000329
00000330 % r2-vanjski promjer
00000331 r2=get(handles.edit3,'String');
00000332 r2=str2double(r2);
00000333
00000334 % Q-protok
00000335 Q=get(handles.edit4,'String');
00000336 Q=str2double(Q);
00000337
00000338 % b-visina rotora
00000339 b=get(handles.edit5,'String');
00000340 b=str2double(b);
00000341
00000342 % omega-kutna brzina
00000343 omega=2*pi*n/60;
00000344
00000345 % u1-obodna brzina
00000346 u1=r1*omega;
00000347 set(handles.text32,'String',u1);
00000348
00000349 % alfa1-kut apsolutne brzine
00000350 alfa1=90;
00000351 set(handles.text37,'String',alfa1);
00000352
00000353 % A1-površina poprečnog presjeka
00000354 A1=2*r1*pi*b;
00000355 set(handles.text36,'String',A1);
00000356
00000357 % v1-apsolutna brzina
00000358 v1=Q/A1;
00000359 set(handles.text33,'String',v1);
00000360
00000361 % w1-relativna brzina
00000362 w1=(v1^2+u1^2)^(1/2);
00000363 set(handles.text35,'String',w1);
00000364
00000365 % v1m-meridijalna komponenta brzine v1
00000366 v1m=v1;
00000367 set(handles.text34,'String',v1m);
00000368
```

```
00000369 % beta1-kut relativne brzine
00000370 beta1=pi-atan(v1/u1);
00000371 beta1=beta1*180/pi;
00000372 set(handles.text38,'String',beta1);
00000373
00000374 %% racunanje izlaznih vrijednosti
00000375 % iz poznatih vrijednosti na unutarnjem radijusu lopatica (r1) i
      interaktivnog
00000376 % geometrijskog računanja kuta beta2 izračunavaju se ostale
      vrijednosti na
00000377 % vanjskom radijusu lopatice (r2)
00000378
00000379 % u2-obodna brzina
00000380 u2=r2*omega;
00000381 set(handles.text53,'String',u2);
00000382
00000383 % v2m-meridijalna komponenta apsolutne brzine v2
00000384 v2m=v1m;
00000385 set(handles.text54,'String',v2m);
00000386
00000387 % w2u-obodna komponenta relativne brzine w2
00000388 w2u=v2m/atan(pi-(beta2*pi/180));
00000389 set(handles.text55,'String',w2u);
00000390
00000391 % v2u-obodna komponenta apsolutne brzine v2
00000392 v2u=u2-w2u;
00000393 set(handles.text56,'String',v2u);
00000394
00000395 % v2-apsolutna brzina
00000396 v2=(v2m^2+v2u^2)^(1/2);
00000397 set(handles.text57,'String',v2);
00000398
00000399 % alfa2-kut apsolutne brzine
00000400 alfa2=abs(atan(v2m/v2u)/pi*180);
00000401 set(handles.text58,'String',alfa2);
00000402
00000403 % Y-specifična snaga
00000404 Y=abs(u2*v2u);
00000405 set(handles.text71,'String',Y);
00000406 end
```

8.2.6. circle

```
00000001 function H=circle(center,radius,NOP,style)
00000002 %-----
00000003 % H=CIRCLE(CENTER,RADIUS,NOP,STYLE)
00000004 % Ova funkcija crta kružnicusa centrom koji je definiran kao
00000005 % vektor CENTER, radijusom kao skalarom RADIS, i brojem
00000006 % točaka kružnice NOP. Stil je isti kao i kod rutinskog plotanja
00000007 %
00000008 %   Primjer
00000009 %   circle([1,3],3,1000,':');
00000010 %   circle([2,4],2,1000,'--');
00000011 %
00000012 %   Zhenhai Wang
00000013 %   Version 1.00
00000014 %   December, 2002
00000015 %-----
00000016
00000017 if (nargin <3),
00000018     error('Please see help for INPUT DATA. ');
00000019 elseif (nargin==3)
00000020     style='b-';
00000021 end;
00000022 THETA=linspace(0,2*pi,NOP);
00000023 RHO=ones(1,NOP)*radius;
00000024 [X,Y] = pol2cart(THETA,RHO);
00000025 X=X+center(1);
00000026 Y=Y+center(2);
00000027 H=plot(X,Y,style);
00000028 axis square;
```