

P R I P R A V N I S K A N A L O G A

V Ljubljani, 27.6.1988

Vrhovec Borut

Vrhovec Borut

KAZALO

I. SIMULACIJA REGULACIJE SINHRONSKEGA MOTORJA	1
1. Matematični model splošnega trifaznega izmeničnega motorja	1
2. Matematični model trifaznega sinhronskega motorja s trajnimi magneti	3
3. Obravnava sinhronskega motorja v koordinatah polja	5
4. Delovanje sinhronskega motorja s trajnimi magneti	7
5. Simulacija sinhronskega motorja s trajnimi magneti	9
6. Močnostni pretvornik	10
7. Merjenje tokov, pozicije in izračun hitrosti	11
8. Digitalna regulacija sinhronskega motorja v koordinatah polja	13
9. Simulacijski program	15
10. Vpliv vzorčnih časov in ločljivosti AD pretvorbe na regulacijo motorja	17
11. Rezultati simulacije različnih režimov delovanja motorja :	22
11.1. Delovanje nad nazivno hitrostjo	22
11.2. Reverzija hitrosti	24
11.3. Obremenitev motorja	27
11.4. Ustavljanje motorja	29
II. VARSTVO PRI DELU	30
III. DNEVNIK	38
IV. LITERATURA	40

SIMULACIJA REGULACIJE SINHRONKEGA MOTORJA

1. MATEMATIČNI MODEL SPLOŠNEGA TRIFAZNEGA IZMENIČNEGA MOTORJA

Po W. Leonhardu (L2) sem prevzel model splošnega trifaznega izmeničnega motorja. Model velja pri naslednjih pogojih :

- cilindrični rotor in stator s trifaznim simetričnim navitjem z izoliranim ničliščem,
- rotorsko in statorsko navitje z enakim številom ovojev,
- enakomerna zračna reža,
- sinusna porazdelitev magnetnega polja po rotorju in statorju,
- zanemarljive izgube železa in vrtilinčnih tokov.

Matematični model sestavljajo električne in mehanske enačbe.

Električni del :

$$u_s(t) = R_s i_s(t) + L_s \frac{d i_s(t)}{dt} + M \frac{d}{dt} \operatorname{Im}(i_r(t) e^{j\epsilon}) \quad (1.1)$$

$$u_r(t) = R_r i_r(t) + L_r \frac{d i_r(t)}{dt} + M \frac{d}{dt} \operatorname{Im}(i_s(t) e^{-j\epsilon}) \quad (1.2)$$

$$m_d = \frac{2}{3} M \operatorname{Im}(i_s(t) (i_r(t) e^{j\epsilon})^*) \quad (1.3)$$

$$u_s(t) = u_{s1}(t) + u_{s2}(t) e^{j\gamma} + u_{s3}(t) e^{j2\gamma} \quad (1.4)$$

$$u_r(t) = u_{r1}(t) + u_{r2}(t) e^{j\gamma} + u_{r3}(t) e^{j2\gamma} \quad (1.5)$$

$$i_s(t) = i_{s1}(t) + i_{s2}(t) e^{j\gamma} + i_{s3}(t) e^{j2\gamma} \quad (1.6)$$

$$i_r(t) = i_{r1}(t) + i_{r2}(t) e^{j\gamma} + i_{r3}(t) e^{j2\gamma} \quad (1.7)$$

kjer je :

- $i_s(t)$ je "vektor" trifaznega statorskega in $i_r(t)$ rotorskega toka,
- $u_s(t)$ in $u_r(t)$ sta vektorja trifazne statorske in rotorske napetosti,
- $u_{s1}(t)$, $u_{s2}(t)$ in $u_{s3}(t)$ so fазne statorske, $u_{r1}(t)$, $u_{r2}(t)$ in $u_{r3}(t)$ pa rotorske napetosti,
- $i_{s1}(t)$, $i_{s2}(t)$ in $i_{s3}(t)$ so fazni statorski, $i_{r1}(t)$, $i_{r2}(t)$ in $i_{r3}(t)$ pa rotorski tokovi,
- γ je prostorska premaknitev navitij (120°),
- L_s je statorska, L_r rotorska in M medsebojna induktivnost,
- R_s je statorska in R_r rotorska upornost,
- ϵ je položaj rotorja glede na mirujoči stator,
- m_d je električni navor motorja.

Mehanski del:

$$J \frac{dw}{dt} = m_d(t) - m_b(t) - b w \quad (1.8)$$

$$\frac{d\epsilon}{dt} = w$$

(1.9).

kjer je :

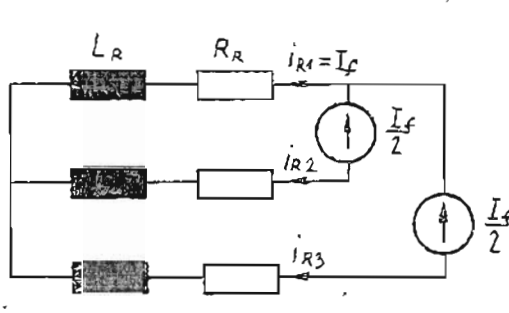
- J vstrajnostni moment motorja,
- w kotna hitrost motorja,
- m_b bremenski navor,
- b koeficient viskoznega trenja ležajev.

Model motorja je dinamičen: zato velja za poljubno obliko statorskih in rotorskih tokov ter za spremenljiv navor in hitost. Za obravnavo delovanja motorja v stacionarnih razmerah, t.j. pri enakomernem bremenskem navoru in sinusnih električnih veličinah, se zgornje enačbe nekoliko poenostavijo.

2. MATEMATIČNI MODEL TRIFAZNEGA SINHRONSKEGA MOTORJA S TRAJNIMI MAGNETI

Matematični model splošnega trifaznega izmeničnega motorja je uporaben za poljubni izmenični motor. Glede na lastnosti obravnavanega motorja se enačbe iz prvega poglavja ustrezno spremenijo.

Za sinhronski motor s trajnimi magneti na rotorju predlaga Leonhard /1/ naslednje rotorsko nadomestno vezje (slika 2.1).



Sl. 2.1. Električno nadomestno vezje rotorja

Iz vezja dobimo vektor rotorskega toka

$$i_r(t) = I_f - \frac{I_f}{2} e^{j\tau} - \frac{I_f}{2} e^{j2\tau} = \frac{3}{2} I_f \quad (2.1)$$

Enačba (1.1) preide v :

$$u_s(t) = R_s i_s(t) + L_s \frac{d i_s(t)}{dt} + j\omega \frac{3}{2} e^{j\epsilon} \quad (2.2)$$

Kjer smo z $\frac{3}{2}$ označili produkt $M I_f$. Rotorska napetostna enačba (1.2) zaradi trajnih magnetov na rotorju odpade. Enačba za električni navor (1.3) dobi z upoštevanjem (2.1) obliko :

$$m_d = \frac{3}{2} \operatorname{Im}(i_s e^{-j\epsilon}) \quad (2.3)$$

Mehanski enačbi (1.8 in 1.9) ostaneta nespremenjeni.

Pri obravnavi delovanja motorja v stacionarnih razmerah vstavimo vrednosti faznih statorskih napetosti

$$u_{s1}(t) = \frac{\sqrt{2}}{2} (U_s e^{j\omega t} + U_s^* e^{-j\omega t}) = \sqrt{2} U_s \cos(\omega t + \alpha_1) \quad (2.4)$$

$$u_{s2}(t) = u_{s1}\left(t - \frac{\tau}{\omega}\right) \text{ in} \quad (2.5)$$

$$u_{s3}(t) = u_{s1}\left(t - \frac{2\tau}{w}\right) \quad (2.6)$$

v enačbo (1.4) in dobimo vektor statorske trifaze napetosti za stacionarne razmere

$$\underline{u}_s(t) = \frac{3}{\sqrt{2}} \underline{u}_s e^{j\omega t} \quad (2.7)$$

kjer je

$$\underline{u}_s = U_s e^{j\alpha_1} \quad (2.8)$$

kompleksor z amplitudo enako efektivni vrednosti fazne statorske napetosti U_s in poljubnim faznim kotom α_1 , odvisnim od izbranega trenutka za čas nič. Podobno dobimo preko faznih statorskih tokov

$$i_{s1}(t) = \sqrt{2} I_s \cos(\omega t + \alpha_2) \quad (2.9)$$

$$i_{s2}(t) = i_{s1}\left(t - \frac{\tau}{w}\right) \quad (2.10)$$

$$i_{s3}(t) = i_{s1}\left(t - \frac{2\tau}{w}\right) \quad (2.11)$$

in enačbe (1.6) vektor statorskega trifaznega toka

$$\underline{i}_s(t) = \frac{3}{\sqrt{2}} \underline{i}_s e^{j\epsilon} \quad (2.12)$$

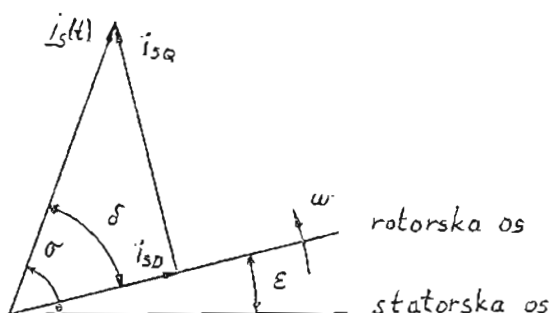
kjer je $\epsilon (= \omega t)$ pozicija rotorja in

$$\underline{i}_s = I_s e^{j\alpha_2} \quad (2.13)$$

Kompleksor z amplitudo enako efektivni vrednosti faznega toka I_s in poljubnim faznim kotom α_2 , odvisnim od izbranega trenutka za čas nič.

3. OBRAVNAVA SINHRONSKEGA MOTORJA V KOORDINATAH POLJA

F. Blascke je prvi predlagal obravnavo izmeničnih motorjev v koordinatah polja. Izmenične časovno spremenljive veličine motorja opazujemo v vrtečem se koordinatnem sistemu pripetem na rotorsko magnetno polje. Pri sinhronskem motorju je rotorsko magnetno polje pripeto na rotor, lega rotorja pa nam je znana. Slika (3.1) prikazuje razstavitev vektorja trifaznega toka na dve enosmerni komponenti pripeti na rotor.



Slika 3.1. Poljske komponente statorskega toka

Tok i_{sd} je vzporeden rotorski osi tok i_{sq} pa je pravokoten nanjo. $\epsilon(t)$ je kot zasuka rotorja glede na stator, $\sigma(t)$ kot vektorja trifaznega toka, $\delta(t)$ pa bremenski kot. Iz zgornje slike izhajaja zveza :

$$i_s(t) = i_s e^{j\sigma} = (i_{sd} + j i_{sq}) e^{j\epsilon} \quad (3.1)$$

S prevedbo trifaznih statorskih tokov na dvofazne ob upoštevanju izoliranosti ničlišča in zveze (3.1) dobimo enačbe transformacije polja :

$$i_{sa} = \frac{3}{2} i_{s1}(t) \quad (3.2)$$

$$i_{sb}(t) = \frac{\sqrt{3}}{2} (i_{s2}(t) - i_{s3}(t)) \quad (3.3)$$

$$i_{sd}(t) = i_{sa} \cos \epsilon + i_{sb} \sin \epsilon \quad (3.4)$$

$$i_{sq}(t) = i_{sb} \cos \epsilon - i_{sa} \sin \epsilon \quad (3.5)$$

Enačbi (3.2) in (3.3) imenujemo združitev faz, enačbi (3.4) in (3.5) pa demodulacija. i_{sa} in i_{sb} sta komponenti dvofaznega sistem. Podobno dobimo enačbe inverzne transformacije polja :

$$i_{sa}(t) = i_{sd} \cos \epsilon - i_{sq} \sin \epsilon \quad (3.6)$$

4. DELOVANJE SINHRONKEGA MOTORJA S TRAJNIMI MAGNETI

Sinhroski motorji z močnimi trajnimi magneti so zaradi svoje dinamike zelo primerni za servopogone. Od njih se zahteva poleg delovanja pod osnovno hitrostjo tudi delovanje visoko nad njo. Ker je rotorsko magnetno polje povzročeno s trajnimi magneti, je slabljenje polja težko. Uvajati moramo tako statorsko polje, ki bo poleg potrebnega navora povzročalo tudi slabljenje rotorskega polja. Obravnava v koordinatah polja omogoča preprosto reguliranje motorja. Iz enačb (2.3) in (3.1) dobimo izraz

$$m_d = \tilde{\Phi} i_{sq} \quad (4.1)$$

Ki pove, da je navor motorja sorazmeren komponenti i_{sq} . Iz slike (3.1) je razvidno, da komponenta i_{sd} povzroča polje v smeri rotorskega polja, zato lahko služi za slabljenje.

Statorska napetostna enačba (2.2) je pomembna pri dimenzioniranju močnostnega pretvornika. Vanjo vstavimo enačbi za statorski tokovni in napetostni vektor v stacionarnih razmerah (2.7 in 2.12) in dobimo izraz

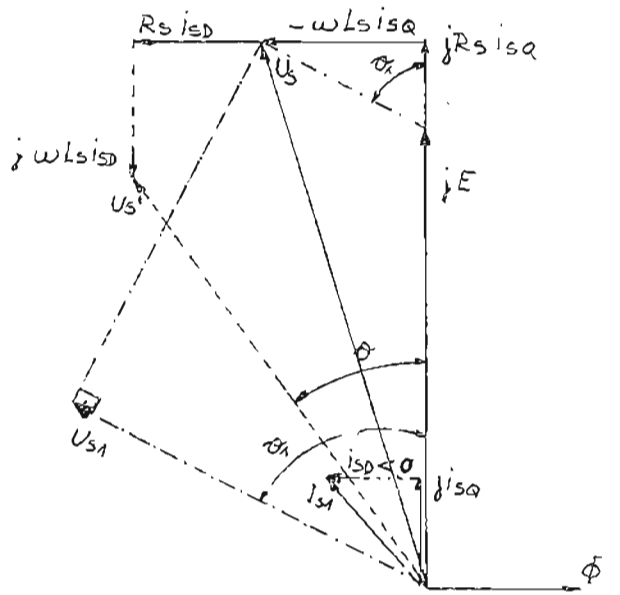
$$U_s = j(E + \omega L_s i_{sd} + R_s i_{sq}) + R_s i_{sd} - \omega L_s i_{sq} \quad (4.2)$$

kjer je

$$E = \frac{1}{\sqrt{2}} \omega \tilde{\Phi} \quad (4.3)$$

efektivna vrednost inducirane napetosti v faznem navitju. Toka i_{sd} in i_{sq} sta poljski komponenti faznega toka in imata vrednost

$$i_{sq} = \frac{\sqrt{2}}{3} i_{sq}, \quad i_{sd} = \frac{\sqrt{2}}{3} i_{sd} \quad (4.4)$$



Slika 4.1. Kazalci v faznem statorskem navitju

Iz enačbe (4.2) izhaja kazalčni diagram (slika 4.1), ki dobro ponazarja razmere v faznem navitju med delovanjem motorja. Pri delovanju pod osnovno hitrostjo je inducirana napetost dovolj majhna, da je U_S v dovoljenih mejah, in ni potrebno slabljenje polja, zato imamo le komponento toka I_{sq} . Izkoristek motorja je dober, ker se ves tok troši za navor.

Pri delovanju nad osnovno hitrostjo inducirana napetost precej naraste. Pri neki hitrosti močnostni pretvornik ne more več dajati dovolj velike napetosti U_S . Čez te obrate lahko motor zavrtimo le, če oslabimo rotorsko magnetno polje in tako zmanjšamo inducirano napetost. To dosežemo z uvajanjem negativne komponente toka I_{sd} . Kot vidimo na sliki (4.1) (črtkano), se potrebna statorska napetost U_S' res zmanjša, vendar pa je smiselno večanje negativnega toka I_{sd} le do te mere, da je kot kazalca napetosti, θ manjši od θ_1 . Če gremo čez kot θ_1 začne napetost U_{S1} naraščati. Slabljenje polja zahteva zelo velik tok I_{sd} . Ker pretvornik lahko daje le nek največji efektivni fazni tok I_{smax} , se mora komponenta I_{sq} ustrezno zmanjšati po enačbi

$$I_{sq} = \sqrt{I_{smax}^2 - I_{sd}^2}, \quad (4.5)$$

zato navor motorja nad nazivnimi obrati pade. Izkoristek motorja je slab, saj se večina toka porablja za slabljenje polja.

Pri regulaciji komponente I_{sd} potrebujemo velikost statorske fazne napetosti

$$U_S = \sqrt{(E + \omega L_S I_{sd} + R_S I_{sq})^2 + (R_S I_{sd} - \omega L_S I_{sq})^2} \quad (4.6)$$

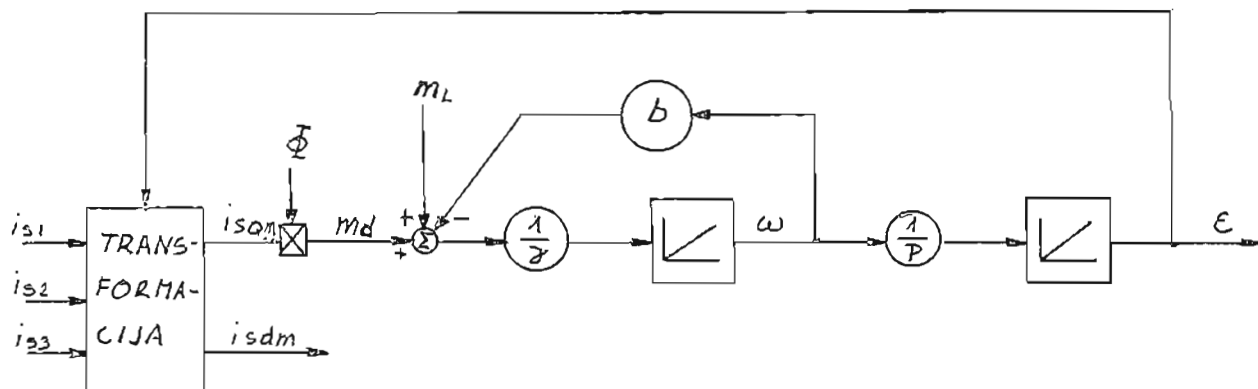
Enačbi za izračun kotov θ in θ_1 izhajata iz kazalčnega diagrama (slika 4.1) :

$$\theta = \text{Arctg} \left(\frac{R_S I_{sd} - \omega L_S I_{sq}}{E + R_S I_{sq} + \omega L_S I_{sd}} \right), \quad (4.7)$$

$$\theta_1 = \text{Arctg} \left(\frac{\omega L_S}{R_S} \right), \quad (4.8)$$

5. SIMULACIJA SINHRONSKEGA MOTORJA S TRAJNIMI MAGNETI

Slika (5.1) prikazuje simulacijsko shemo sinhrnskega motorja s trajnimi magneti v koornatah polja.



Slika 5.1. simulacijski model sinhronskega motorja

Shema izhaja iz enačb: (1.8), (1.9), (3.2), (3.3), (3.4), (3.5) in (4.1).

Pasivno breme smuliramo tako :

- da je pri pozitivni hitrosti motorja $m_1 = -(m_b + m_{trd})$,
- da je pri negativni hitrosti motorja $m_1 = m_b + m_{trd}$,
- da se pri preveliki obremenitvi motor ustavi in se ne zavrti v nasprotno smer.

Pri tem je m_b čisti bremenski navor, m_{trd} pa konstantno (suho) trenje. Poleg tega upoštevamo, na poenonostavljen način, tudi lepenje. Dokler motor stoji in je navor motorja manjši od navora lepenja, se motor ne premakne, m_{trd} pa je enak nič.

Da preprečimo prekoračitve dovoljenih vrednosti komponent i_{sq} in i_{sd} ter kota θ_1 , vključimo na regulacijski strani omejitve. Program nam morebitne prekoračitve teh vrednosti javi.

Fazne statorske napetosti računamo po enačbah

$$u_{s1}(t) = \sqrt{2} U_s \cos \epsilon, \quad (5.1)$$

$$u_{s2}(t) = \sqrt{2} U_s \cos(\epsilon - \tau), \quad (5.2)$$

$$u_{s3}(t) = \sqrt{2} U_s \cos(\epsilon - 2\tau), \quad (5.3)$$

medfazne pa

$$u_{s12} = u_{s1} - u_{s2}, \quad (5.4)$$

$$u_{s23} = u_{s2} - u_{s3}, \quad (5.5)$$

$$u_{s31} = u_{s3} - u_{s1}, \quad (5.6)$$

Vsota trenutni faznih napetosti je v vsakem trenutku enaka nič.

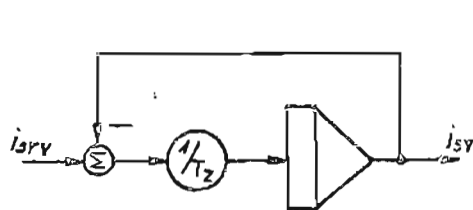
6. MOČNOSTNI PRETVORNIK

Za dobro reguliranje motorja potrebujemo trifazne izmenične veličine spremenjive frevence, amplitude in faze. Za napajanje servomotorjev se uporabljajo transistorski mostiči, krmiljeni s pulzno širinskimi modulatorji (PWM). Naloga PWM-a je tako vklapljanje transistorjev, da imajo izhodne veličine čim manj višjih harmonskih komponent.

V našem programu upoštevamo močnostni napajalnik kot sistem prvega reda v vsaki fazni tokovni veji. Na osnovi enačbe

$$T_z \frac{di_{sv}}{dt} - i_{sv} = i_{srv}, \quad v = 1, 2, 3 \quad (6.1)$$

dobimo simulacijsko shemo (slika 6.1).



Slika 6.1. Sistem prvega reda

Časovna konstanta T_z je reda milisekunde ali manj. Za bolj podrobno obravnavo je potrebna simulacija PWM-a in transistorskega mostiča.

7. MERJENJE TOKOV, POZICIJE IN IZRAČUN HITROSTI

Za zaprtozančno regulacijo sinhronskega motorja potrebujemo povratne informacije o tokvih, poziciji, hitrosti in napetosti.

Zaradi izoliranosti ničlišča statorskega navitja merimo le dva toka, tretjega pa izračunamo po enačbi :

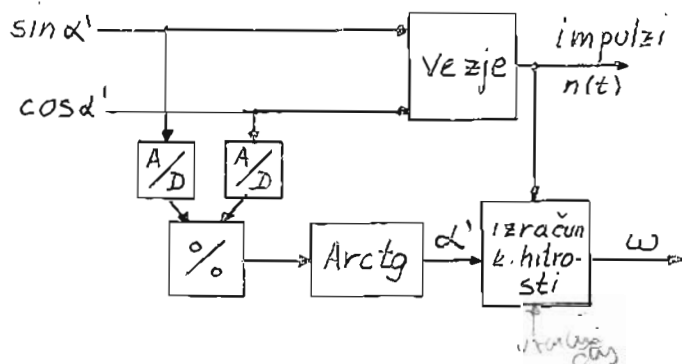
$$i_{s3}(t) = -i_{s1}(t) - i_{s2}(t) \quad (7.1)$$

Tokovni senzor lahko ponazorimo s sistemom prvega reda (slika 6.1), njegov izhod pa preko AD pretvornika vodimo v računalnik.

Velika dinamika izmeničnih servomotorjev zahteva kratke čase vzorčenja, kratki vzorčni časi pa bi, če bi uporabili navadni inkrementalni dajalnik, povzročili slabo ločljivost hitrosti. Zato uporabimo dajalnik s sinusnim in kosinusnim analognim izhodom.

Pozicijo dobimo preko posebnega vezja (oziroma čipa), ki ob določenih časih primerja harmonična signala in da na vsako periodo analognih signalov dvajset impulzov. Na ta način lahko dobimo več kot dvajsettisoč impulzov na obrat motorja.

Pri visokih obratih bi lahko dobili dovolj točno hitrost motorja že samo iz števila impulzov. Pri majhnih hitrostih to ne pride v poštev, ker bi dobili na vzorčni čas premalo impulzov. Ta problem rešimo, kot kaže slika 7.1, z izračunom pozicije med dvema impulzoma. Arctg ulomka, preko AD pretvorbe dobljenih vrednosti harmoničnih signalov, nam da kot α .

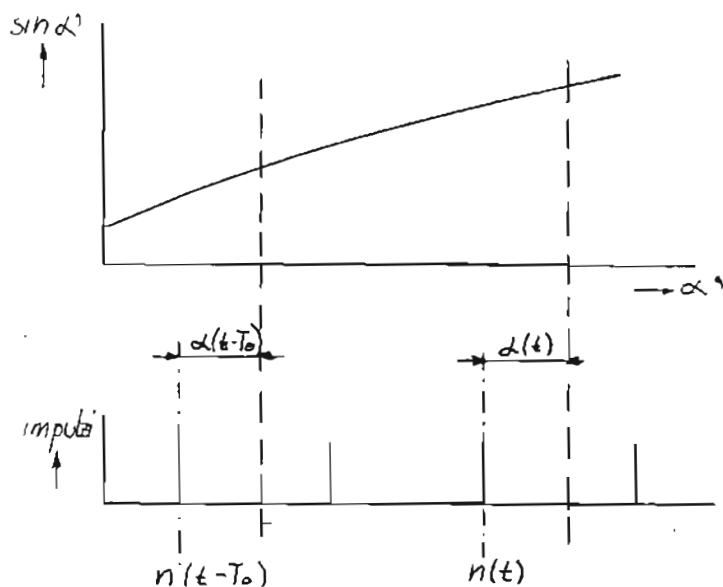


Slika 7.1. Princip izračuna hitrosti

Kotno hitrost motorja nato dobimo po enačbi

$$\omega = \frac{n(t) - n(t-T_0) - \alpha(t-T_0) + \alpha(t)}{T_0} \quad (7.2)$$

Pomen kotov $\alpha(t)$ in $\alpha(t-T_0)$ kaže slika 7.2.



Slika 7.2. Pozicija med dvema impulzoma dajalca

Na točnost izračunane hitrosti pri majhnih obratih vpliva predvsem kvaliteta AD pretvrbe, ki pri visokih obratih ni pomembna. Napaka izračuna hitrosti pri močnih pospeševanjih in zaviranjih je odvisna predvsem od vzorčnega časa. Simulacijski program nam poleg merjenih in resničnih vrednosti tokov, pozicije in hitrosti računa tudi njihovo razliko.

8. DIGITALNA REGULACIJA SINHRONSKEGA MOTORJA V KOORDINATAH POLJA

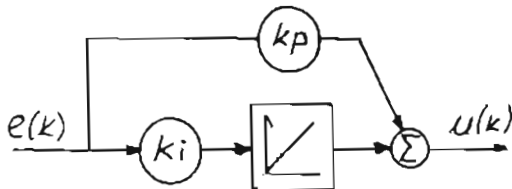
Regulacija izmeničnih servomotorjev je zaradi velikih dinamičnih zahtev, spremenljivih parametrov, kompliciranih omejitvenih pogojev in obravnave v poljskih koordinatah izvedljiva le v digitalni obliki. To pomeni, da se povratne informacije (tokovi, pozicija in hitrost) digitalizirajo in privedejo v računalnik, kjer se z ostalimi podatki matematično obdelajo. Rezultat tega procesa so signali, ki krmilijo PWM ali pa direktno odpirajo transistorje (če je PWM izveden programsko). Digitalni regulatorji so v bistvu matematične enačbe v računalniku. Uporabili smo digitalne PI regulatorje z integracijo izvedeno po trapezni integracijski metodi, ki jo opisuje enačba

$$y(k) = T_0 \left(\frac{x(0) + x(k)}{2} + \sum_{i=1}^{k-1} x(i) \right), \quad (8.1)$$

kjer je

- $y(k)$ vrednost odvisne spremenljivke ob času kT_0 ,
- $x(k)$ vrednost neodvisne spremenljivke in
- T_0 vzorčni čas.

Slika 8.1 predstavlja blok shemo PI regulatorja.



Slika 8.1. PI regulator

Iz slike 8.1 in enačbe (8.1) izhaja enačba diskretnega PI regulatorja

$$u(k) = K_p e(k) + K_i T_0 \left(\frac{e(0) + e(k)}{2} + \sum_{i=1}^{k-1} e(i) \right) \quad (8.2)$$

kjer je :

- $e(k)$ razlika med željeno in doseženo vrednostjo regulirane veličine ob času kT_0 ,
- T_0 vzorčni čas,
- $u(k)$ regulirna veličina (izhodna veličina regulatorja) ob času kT_0 ,
- K_p ojačenje proporcionalnega dela PI regulatorja in
- K_i ojačenje integracijskega dela PI regulatorja.

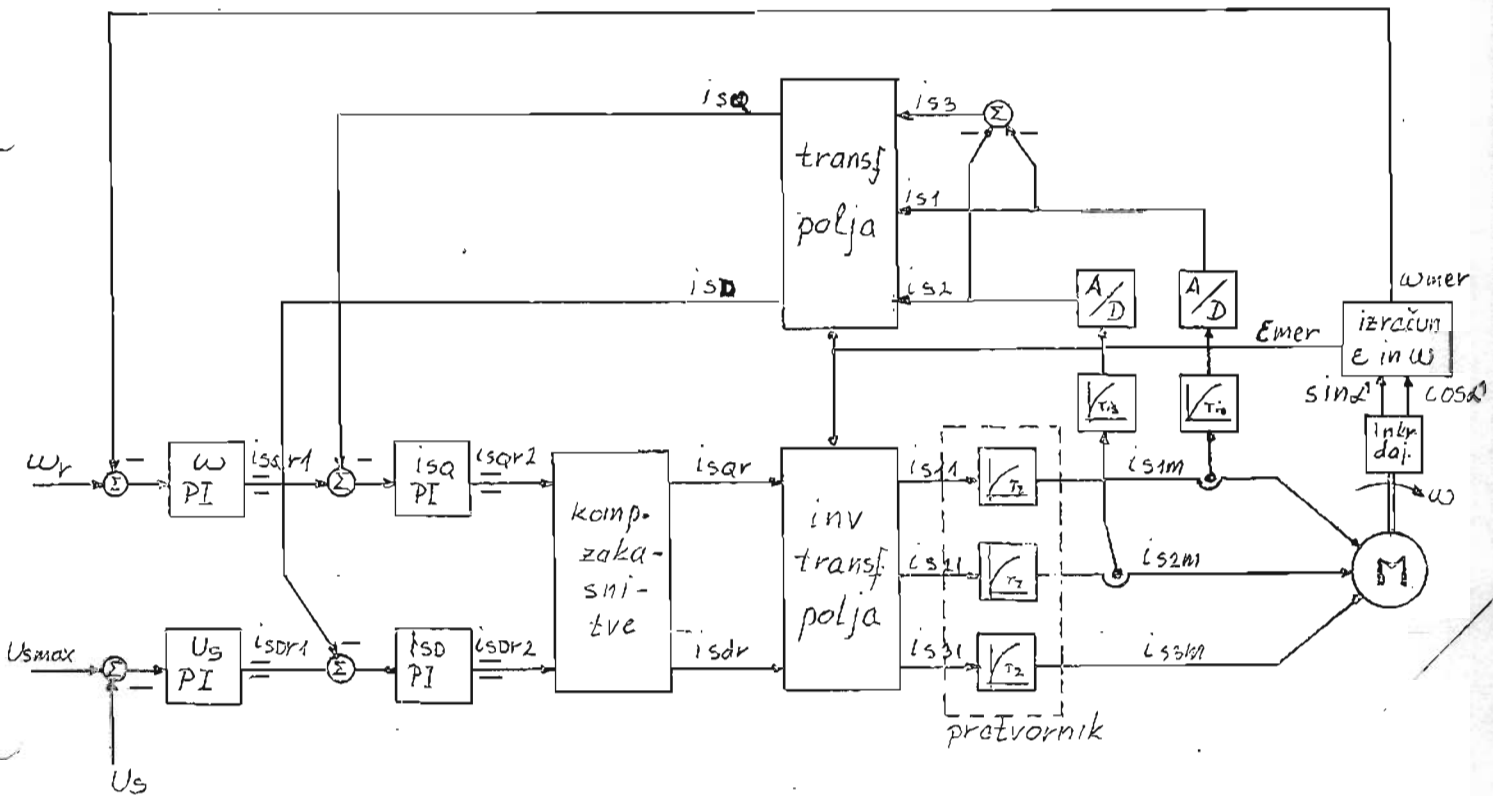
Za računanje z digitalnim računalnikom je bolj primerna rekurzivna formula PI regulatorja, ki jo dobimo iz razlike enačb (8.2) za čas k in $k-1$:

$$u(k) = u(k-1) + a_0 e(k) + a_1 e(k-1) \quad (8.3)$$

$$a_0 = k_p + \frac{k_i T_o}{2}$$

$$a_1 = -k_p + \frac{k_i T_o}{2}$$

Slika 8.2 prikazuje simulacijsko shemo regulacije.



Slika 8.2. Regulacija sinhronskega motorja

Uporabili smo princip kaskadne regulacije. Zunanji, počasnejši regulacijski zanki (hitrostna in napetostna), dajeta referenco notranjima (tokovnjima). Izhodi regulatorjev so omejeni. Izhod isdr1 je omejen na negativne vrednosti, absolutno manjše od največje amplitude vektorja trifaznega statorskega toka

$$i_{smax} = \frac{3}{\sqrt{2}} i_{sfmax} \quad (8.4)$$

Kjer je i_{sfmax} efektivna vrednost največjega dovoljenega faznega toka. i_{sqr1} je po absolutni vrednosti omejen z enačbo

$$i_{sq1} \leq \sqrt{i_{smax2}^2 - i_{sd2}^2} \quad (8.5)$$

Podobno sta omejena tudi izhoda tokovnih regulatorjev i_{sdr2} po enačbi (8.4) in i_{sqr2} po (8.5).

Če pri izračunu vrednosti i_{sq1} in i_{sd1} ne upoštevamo zakasnitve močnostnega pretvornika, pride pri visokih obratih do takšnih napak tokov, da motorja ne obvladamo več. Na preprost način se lahko izpeljejo enačbe za izračun referenc tokov, ki upoštevajo zakasnitev pretvornika :

$$i_{sqr} = i_{sqr2} + wT_k i_{sdr2} \quad (8.6)$$

$$i_{sdr} = i_{sdr2} - wT_k i_{sqr2} \quad (8.7)$$

kjer je T_k enak časovni konstanti močnostnega pretvornika (poglavje 6).

Ker sta notranji regulacijski zanki hitrejši od zunanjih, jih vzorčimo pogosteje. T_{0n} je vzorčni čas pozicije, kompenzacije zakasnitve pretvornika, transformacije in inverzne transformacije polja ter notranjih regulatorjev. T_{0z} je vzorčni čas izračuna hitrosti, napetosti U_s in zunanjih regulatorjev. Motor, močnostni pretvornik in tokovna senzorja smo vzorčili z zelo majhnim vzorčnim časom dt , integracijo pa smo izvedli kar preprosto po pravokotniški metodi zadnje diference. Vzorčne čase T_{0z} , T_{0n} in dt ter parametre regulatorjev lahko med posameznimi teki simulacijskega programa spreminjamo.

9. SIMULACIJSKI PROGRAM

Program za simulacijo sinhronskega motorja reguliranega v koordinatah polja je napisan v programskem jeziku FORTRAN 77. Sestavljen je iz treh glavnih delov.

Prvi del omogoča :

- izbiro file-a s parametri
- spremembo vseh glavnih parametrov motorja, napajalnika, regulatorjev, inkrementalnega dajalca, reference hitrosti, bremena itd,
- izbiro načina simulacije.

Drugi del je sama simulacija.

Tretji del pripravi rezultate v obliki primerni za AGP grafiko :

- izbere število in lego grafov na ekranu,
- rezultate namenjene za grafično predstavitev shrani v datoteke POD1, POD2 in POD3,
- na podlagi največjih vrednosti rezultatov normira ordinate grafov,
- določi napise ordinatnih in abscisnih osi.

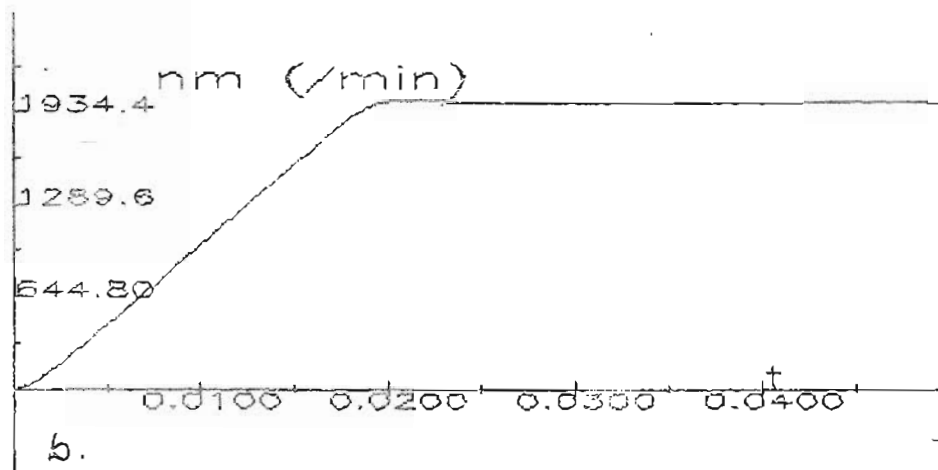
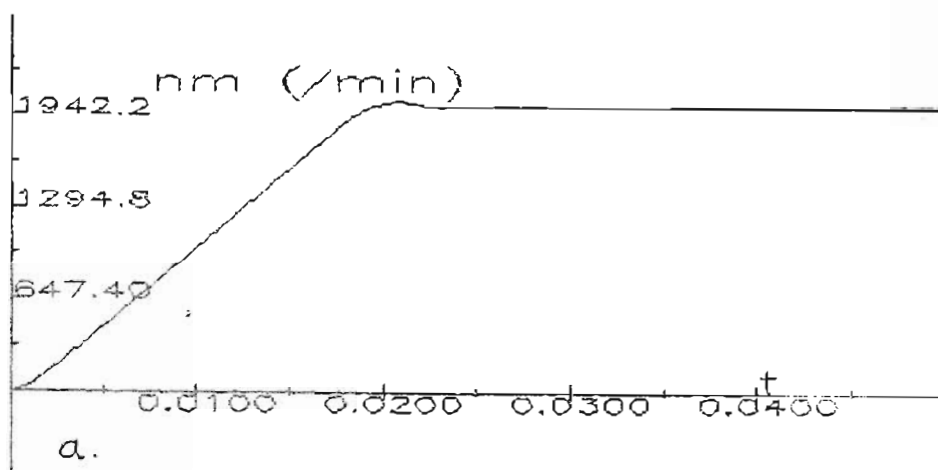
Od izbire načina delovanja je odvisno katere veličine se grafično predstavijo. Možni načini simulacije so:

- normalno delovanje
- optimizacija hitrostnega regulatorja,
- optimizacija regulatorja napetosti U_s ,
- optimizacija tokovnih regulatorjev.

Pri prvih treh simulacijah so razlike le v izpisanih rezultatih za grafiko (spremljamo poteka tistih veličin, ki so pomembne za izbrani način). Pri optimizaciji tokovnih regulatorjev pa odklopimo zunanji regulacijski zanki (hitrostno in napetostno), izberemo referenci tokov i_{sdr1} in i_{sdr2} ter zavremo rotor.

10. VPLIV VZORČNIH ČASOV IN LOČLJIVOSTI AD PRETVORNIKOV NA REGULACIJO MOTORJA

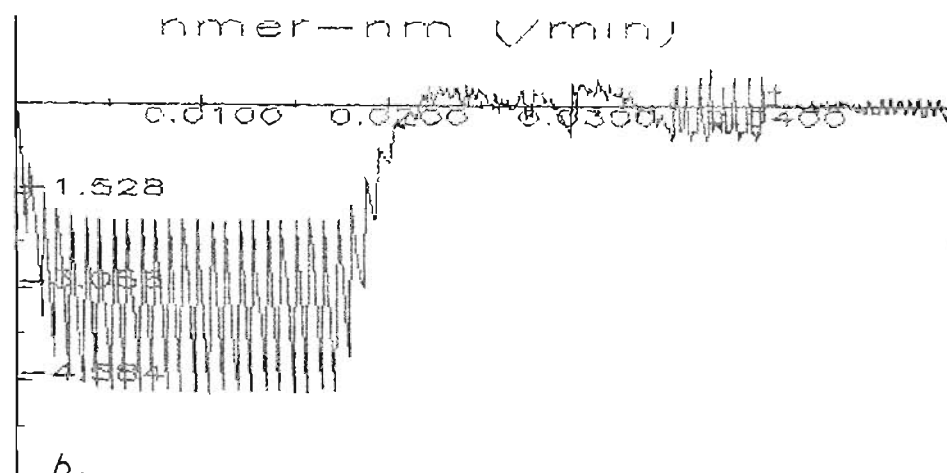
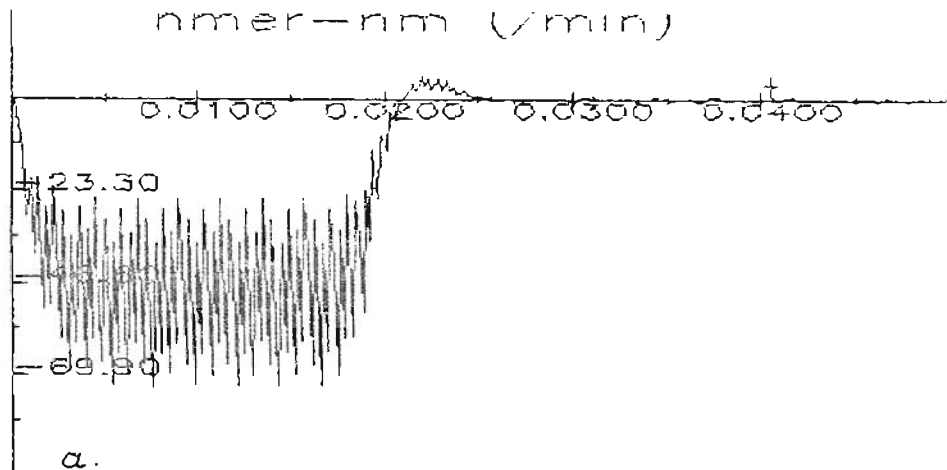
Vpliv vzorčnih časov T_{0z} in T_{0n} se kaže predvsem v napakah merjenih velečin pri hitrih spremembah (pospeševanja, zaviranja, obremenitve). Slika 10.1.a. kaže pospeševanje motorja od mirovanja do nazivnih obratov, pri vzorčnih časih $T_{0n} = 150 \mu s$ in $T_{0z} = 450 \mu s$, slika 10.1.b. pa pri $T_{0n} = T_{0z} = dt$. Čas vzorčenja mehanike dt je bil v obeh primerih enak $5 \mu s$.



Sliki 10.1. a, b Pospeševanje do nazivnih obratov

Pri realnih razmerah (a), zaradi daljših vzorčnih časov hitrost pri enakih parametrih regulatorjev malo bolj preniha, kot v primeru b.

Napako izračuna hitrosti za primer a kaže slika 10.2.a, za primer b pa slika 10.2.b. Na slikah se vidi, da je absolutna napaka največja pri pospeševanju in je skoraj ves čas pospeševanja, zaradi konstantnega pospeška, enaka. Podobno je z izračunom komponent toka i_{sq} in i_{sd} .

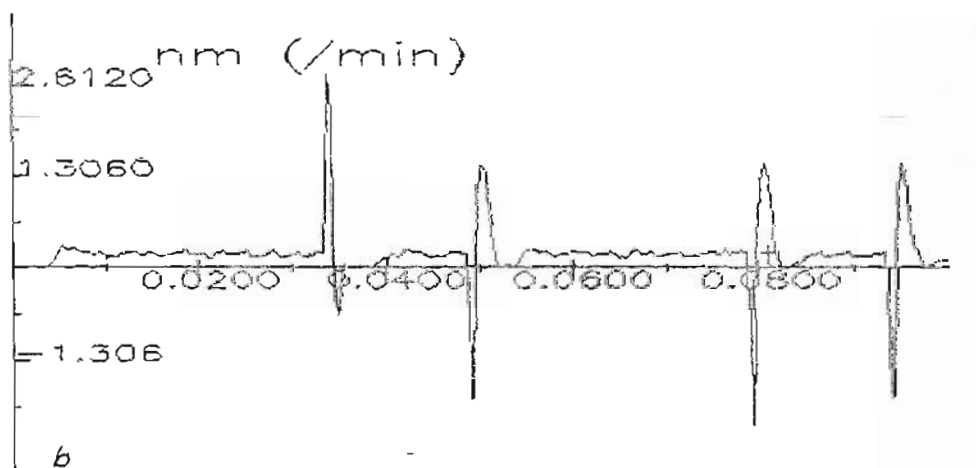
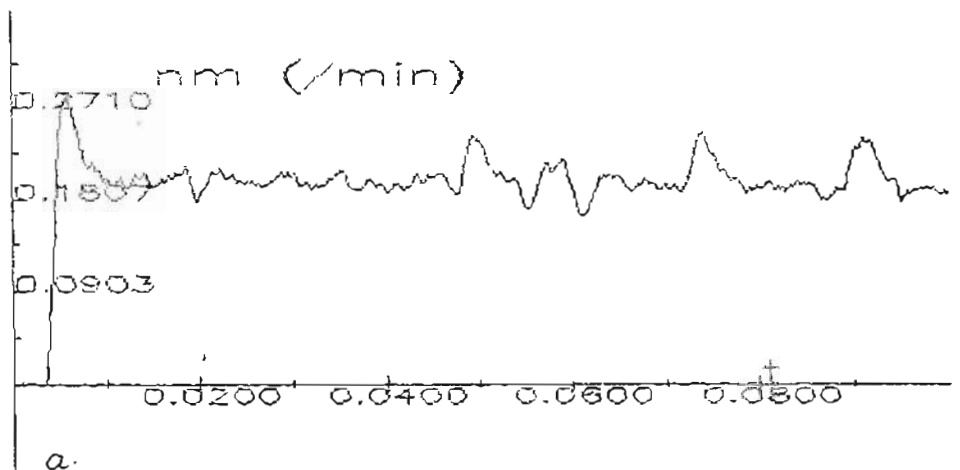


Sliki 10.2.a,b Razlika med merjeno in realno hitrostjo motorja pri različnih vzorčnih časih

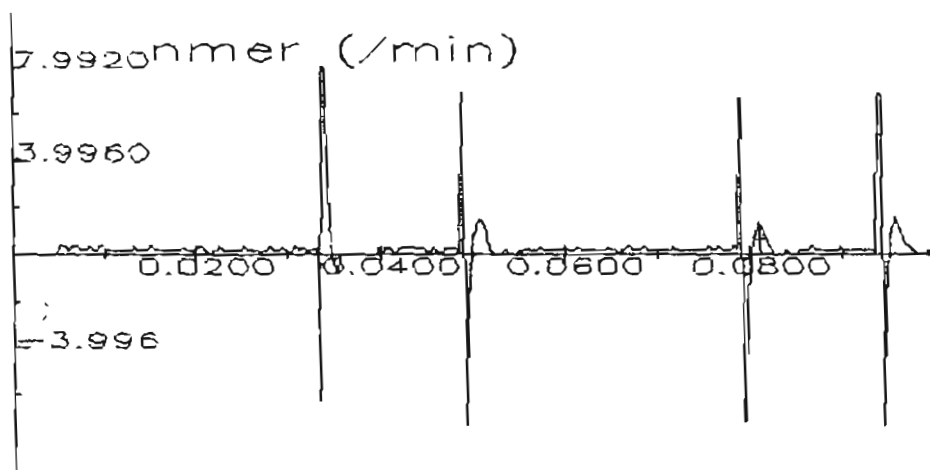
Vpliv ločljivosti AD pretvorbe izhodov inkrementalnega dajalca na točnost izračuna hitrosti se najbolje vidi pri majhnih hitrostih. W.Schumacher /3/ navaja, da je potrebna ločljivost hitrosti motorjev za servopogone okrog 0.2 o/min. Sliki 10.3.a in b kažeta odziv hitrosti motorja na stopnico 0.2 o./min. pri 12 bitnem AD pretvorniku (a) in 8 bitnem (b in c). Na sliki c se vidi, da prihaja, zaradi slabše ločljivosti AD pretvornika, do napak v izračunu hitrosti. Pri prvi špiči smo izračunali

negativno hitrost, čeprav se je motor vrtel v drugi smeri, zato je dobil regulator I_{sq} večjo referenco in zavrtel motor visoko nad željeno hitrost.

Simulirali smo z vzorčnimi časi kot pri primeru 10.1.a in z 12 bitno pretvorbo merjenih tokov. Na slikah 10.3.a in b se vidi, da se motor ne zavrti takoj, ampak šele po nekem času. Vzrok je lepenje. Ker je referenca hitrosti majhna, parametri regulatorjev pa niso zelo veliki, traja nekaj časa, da dobi motor večji moment od momenta lepenja. Ko motor potegne moment trenja pade, zato dobimo relativno velik prenehaj hitrosti.

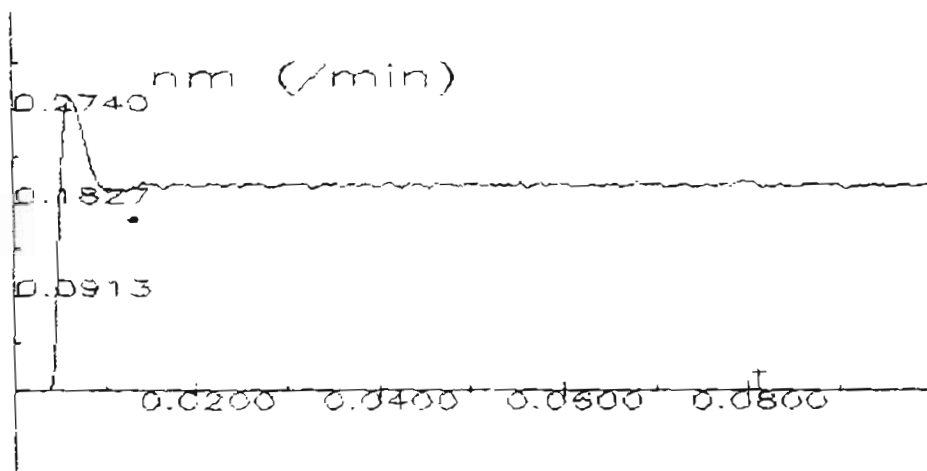


Sliki 10.3.a in b Odziv motorja na stopnico hitrosti 0.20/min pri 12 in 8 bitni AD pretvorbi sig. inkr. dajalca

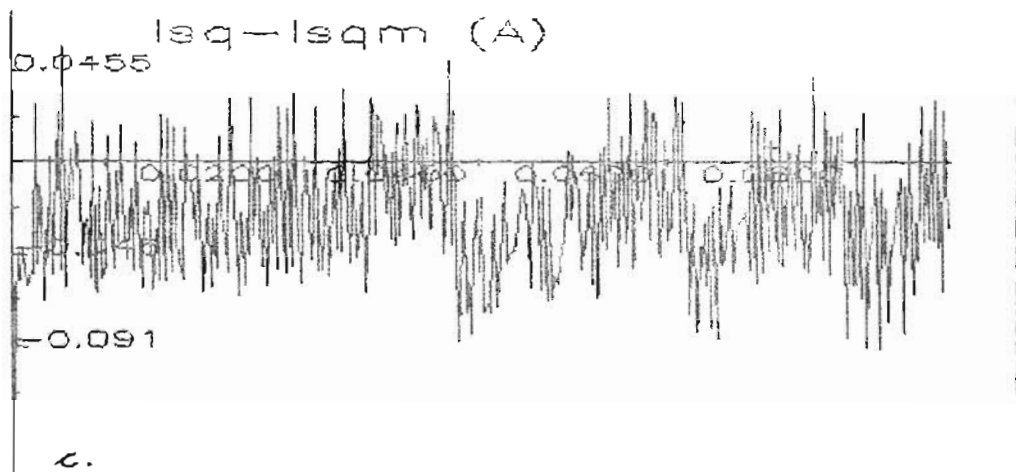
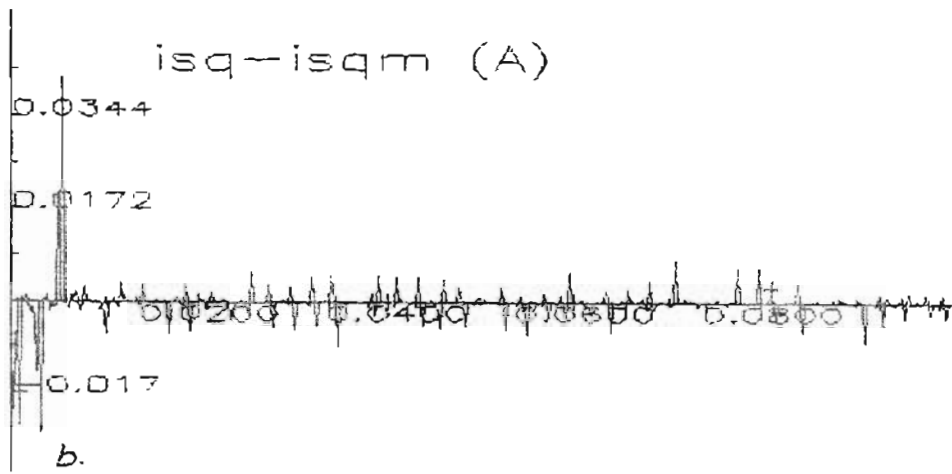


Slika 10.3.c Hitrost merjena preko 8 bit. AD pretvornika

Na nazobčanost hitrosti (slika 10.3.a) vpliva, poleg napake izračuna hitrosti in vzorčnih časov, tudi kvaliteta AD pretvorbe merjenih tokov. Slika 10.4.a kaže odziv hitrosti na enako stopnico kot na slikah 10.3, vendar brez napake AD pretvorbe tokov. Zaradi manjše napake merjenih tokov (slika b) je nihanje toka isam manjše, zato je potek hitrosti gladkejši. Za primerjavo vidimo na sliki 10.4.c napako merjenih tokov pri primeru s slike 10.3.a.



Slika 10.4.a Potek hitrosti brez napake AD pretvorbe tokov



Sliki 10.4.b in c Vpliv AD pretvorbe na napako tokov pri zelo majhni hitrosti motorja

11. REZULTATI SIMULACIJE RAZLIČNIH REŽIMOV DELOVANJA MOTORJA

Kot zgornje simulacije so tudi spodnje izvedene s podatki:

Motor

- nazivni moment $m_n = 10.4 \text{ Nm}$,
- nazivna medfazna napetost $U_n = 220 \text{ V}$,
- nazivni fazni tok $I_{sfn} = 8.5 \text{ A}$,
- vstrajnostni moment $J = 0.0053 \text{ kgm}^2$,
- viskozno trenje ležajev $b = 0.003 \text{ Nms/rad}$,
- fazna upornost $R_s = 0.54 \Omega$,
- fazna induktivnost $L_s = 5.4 \text{ mH}$,

Pretvornik

- največja medfazna napetost $U_{max} \approx 220 \text{ V}$,
- največji fazni tok $I_{fmax} = 51 \text{ A}$ ($6 \times I_{sfn}$ motorja),
- časovna konstanta $T_z = 1 \text{ ms}$,

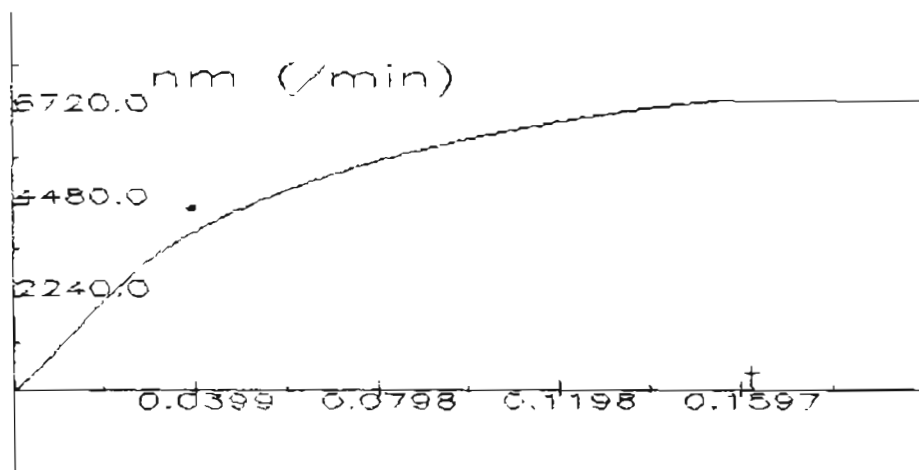
Inkrementalni dajalec

- število period sinus-a in cosinus-a na obrat motorja $n_{per} = 1000$.

Vzorčni časi in AD pretvorniki so enaki kot v prejšnjih primerih.

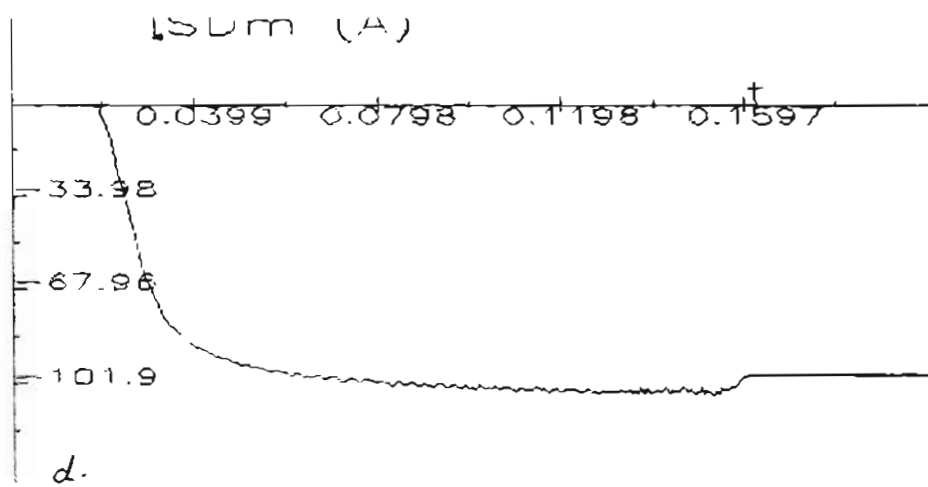
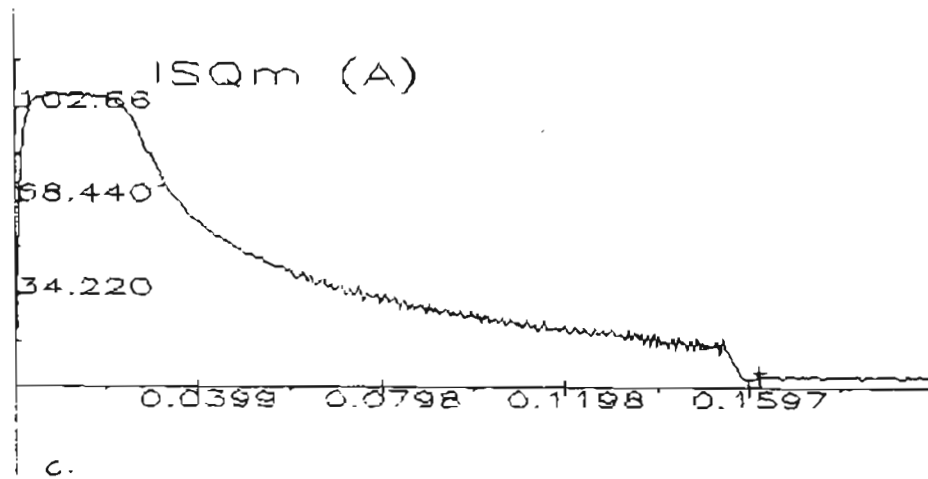
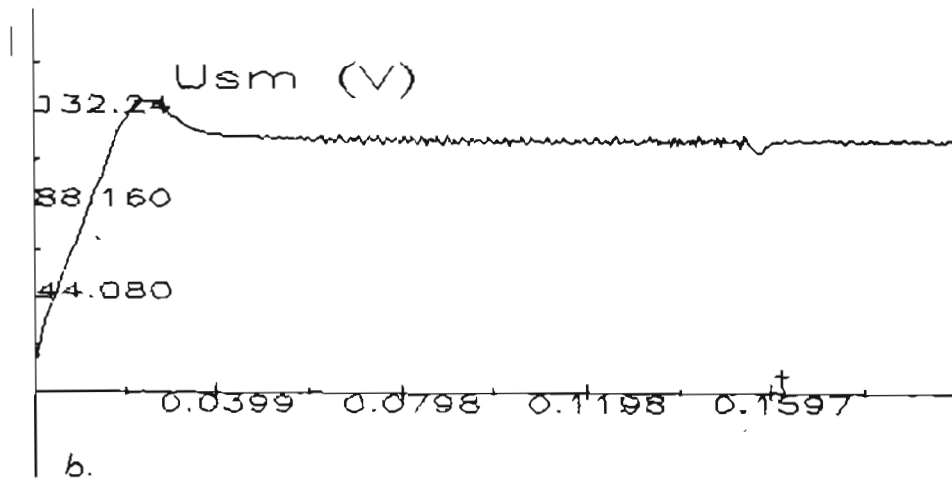
11.1 Delovanje nad nazivno hitrostjo

Slike 11.1 kažejo odziv motorja na stopnico hitrosti 7000 o/min.



Slika 11.1.a hitrost motorja

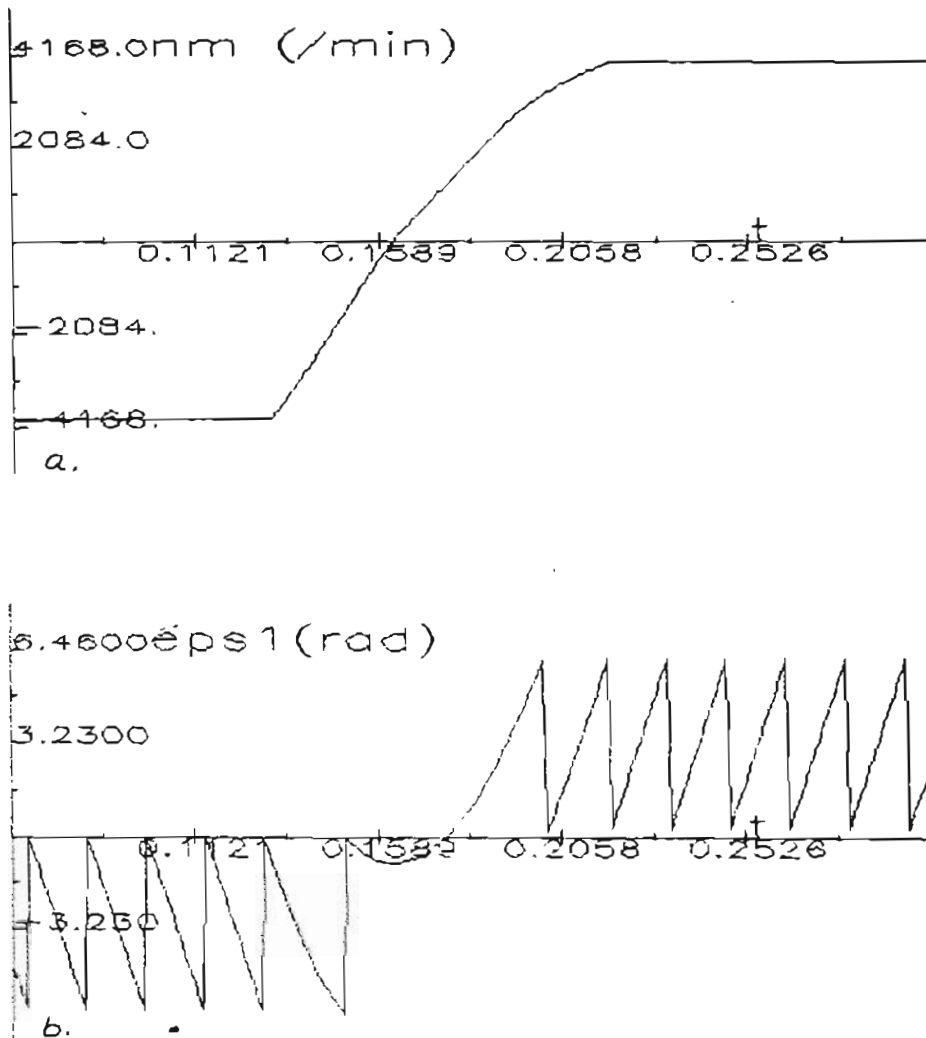
Kot smo že omenili, ima motor v območju slabljenja manjši navor, zato nad nazivnimi obrati hitrost počasneje narašča. Ko U_s doseže največjo dovoljeno vrednost, začne naraščati komponenta I_{sd} , I_{sq} pa ob enem pada (slike 11.1. b, c in d). Prenihaja napetosti U_s ne moremo povsem odpraviti, ker večji parametri regulatorja U_s povzročajo nihanje.



Slike 11.1.b,c,d fazna napetost in komponenti toka

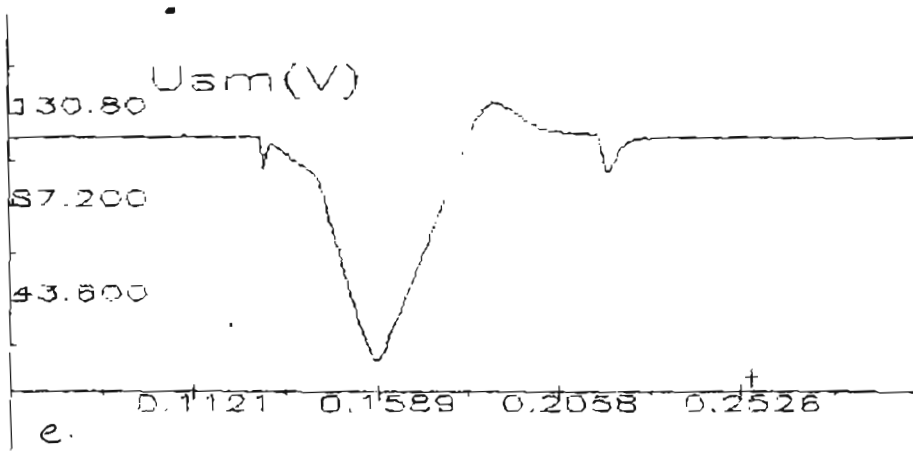
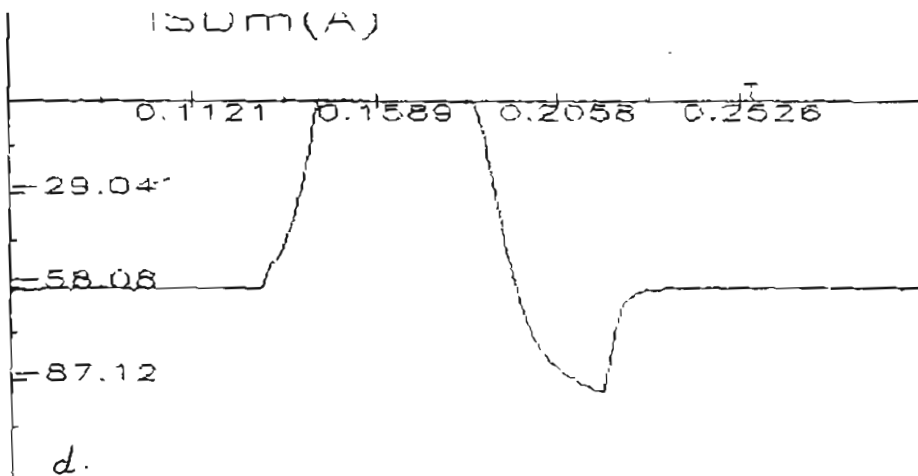
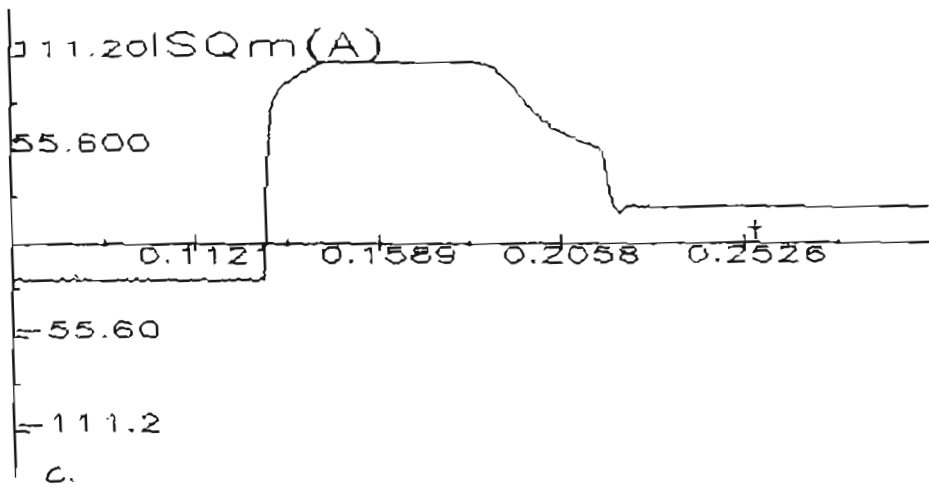
11.2. Reverzija hitrosti

Sliki 11.2.a in b kažeta potek hitrosti in pozicije motorja pri reverziranju z -4000 o/min na +4000 o/min.

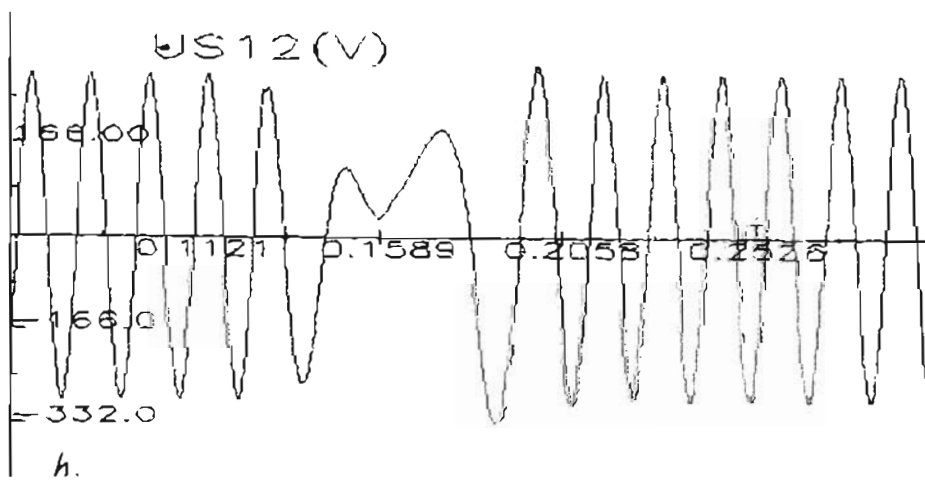
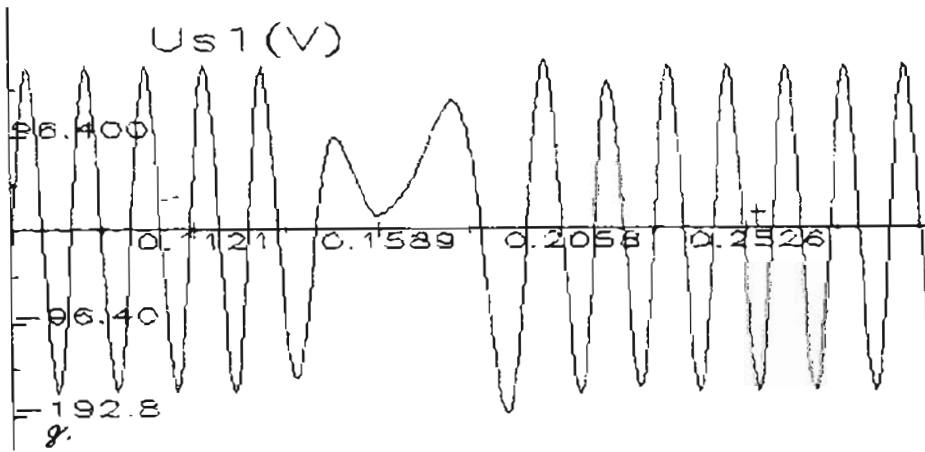
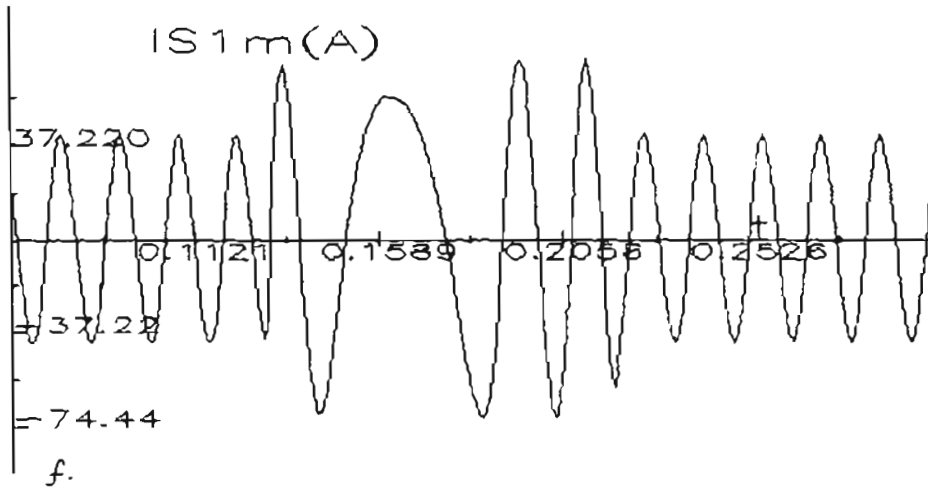


Sliki 11.2.a in b Reverziranje

Na slikah 10.2.c,d in e vidimo poteke poljskih komponent toka in fazne napetosti, na 10.2. f, g in h pa tok in napetost prve faze ter medfazno napetost U_{S12} .



Slide 11.2.c, d in e

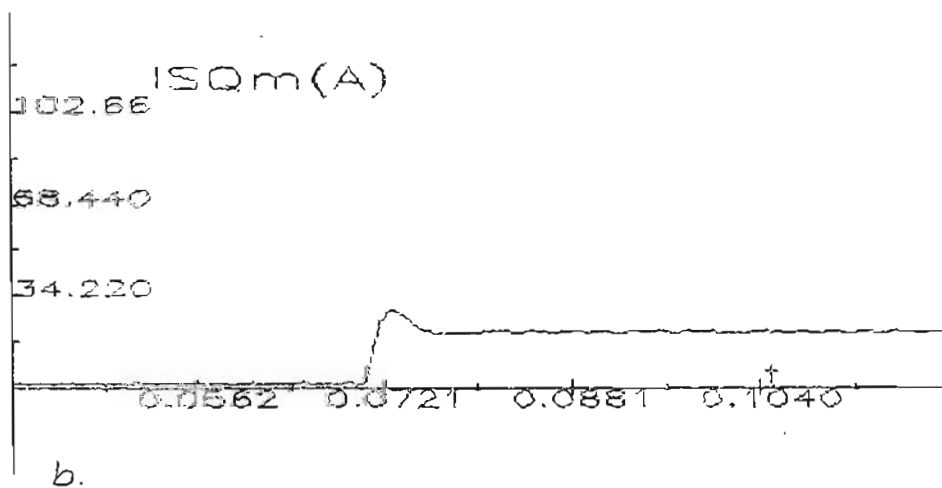
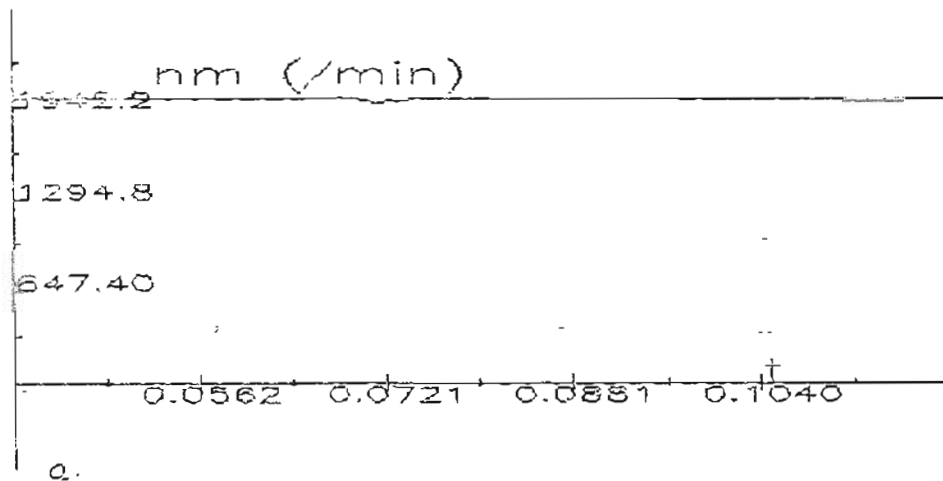


slike 11.2. f, g in h

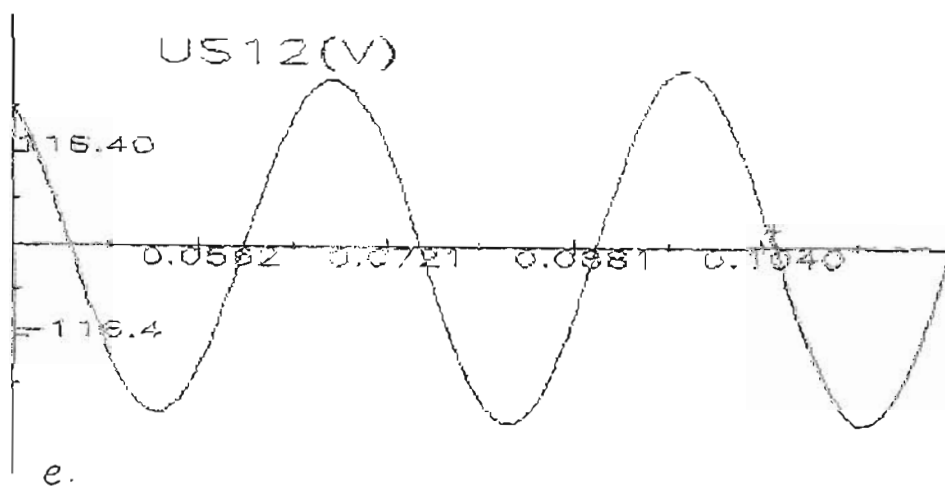
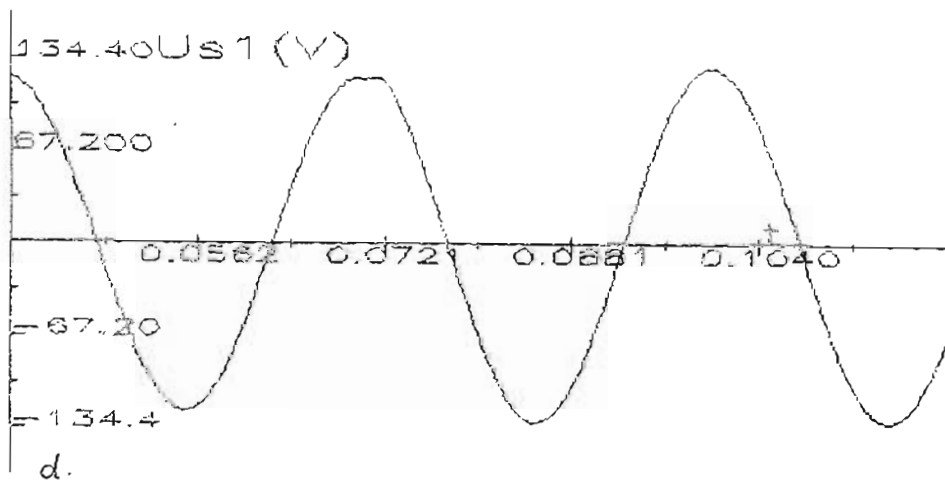
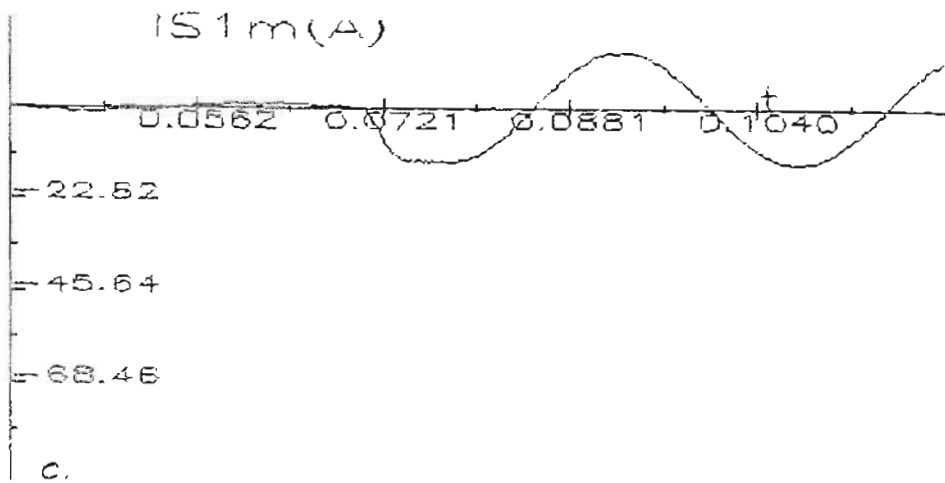
11.3. Obremenitev motorja

Potek hitrosti in komponente i_{sq} pri obremenitvi z 11 Nm kažeta sliki 11.3.a in b. Hitrost v trenutku obremenitve malo pade, vendar regulacija takoj poveča komponento i_{sq} in motor spet ujame željeno hitrost.

Na slikah 10.3. c, d in e so poteki izmeničnih veličin na motorju : faznega toka, fazne in medfazne napetosti. Fazni tok po obremenitvi naraste, napetosti pa se tudi nekoliko povečata.

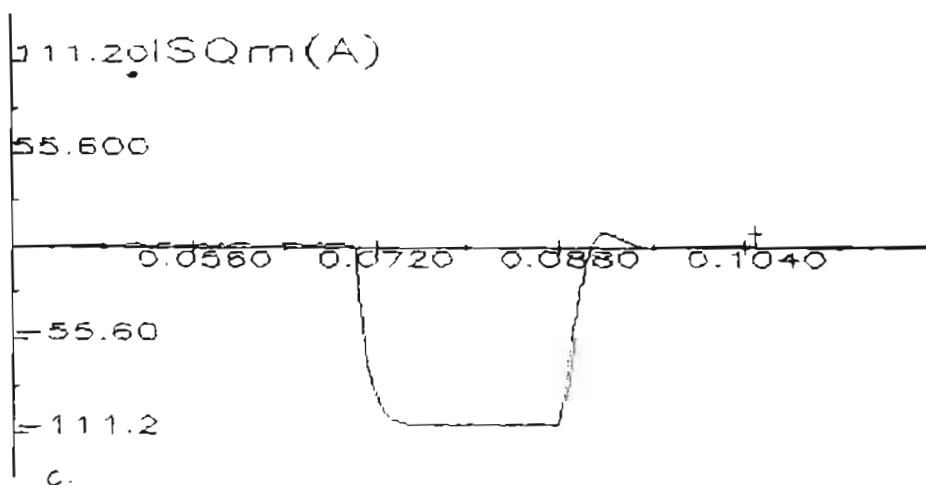
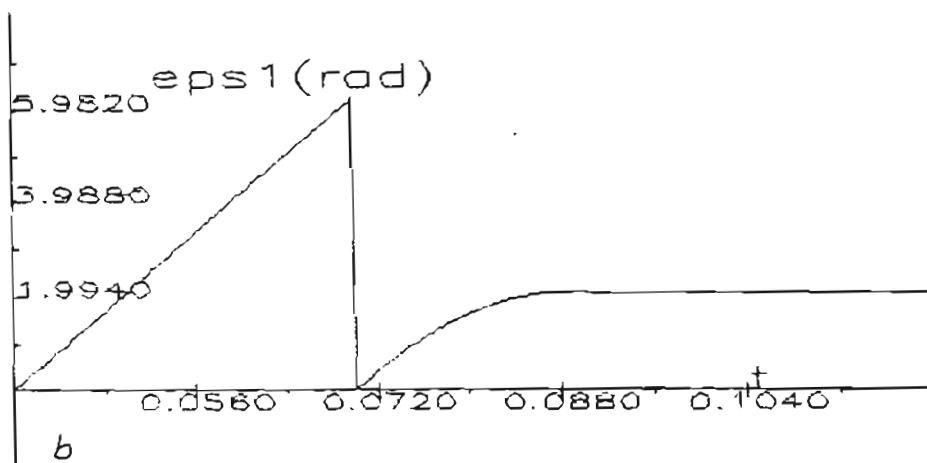
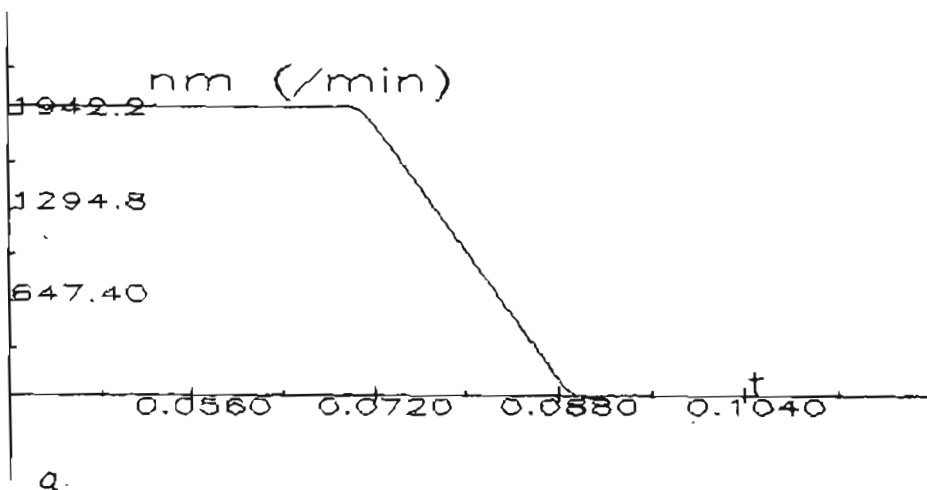


Sliki 11.3.a,b Obremenitev motorja



Slike 11.3. c, d, e Izmenične veličine na obremenjenem motorju

11.4. Ustavljanje motorja



Slike 11.3a,b,c Potek hitrosti, pozicije in komponente i_{sq} pri ustavljanju motorja

V A R S T V O P R I D E L U

ISKRA - AVTOMATI
SLUŽBA ZA VARNOST PRI DELU

Predmet: Prilpaktiška
na Soga

Skladno s ...
(U adri: ...)

Opomba: B.P.

Predloženo opravljeno:
Podpis:

24. 6. 1988

LISTINE VARSTVA PRI DELU

Namen varstva pri delu je poleg zaščite delavcev pred poškodbami pri delu tudi preprečitev zdravstvenih okvar, kot posledic dolgotrajnega delovanja škodljivosti, ki spremljajo delo. Zato so v zakonu o varstvu pri delu dosedanjo pisno izjavo zamenjale listine varstva pri delu.

Zakon o varstvu pri delu določa naslednje listine:

- 11. člen določa, da mora OZD in skupnost, ki projektira sredstva za delo, priložiti k tehnični dokumentaciji ELABORAT, v katerem opredeli nevarnosti in škodljivosti za poškodbe ali zdravstvene okvare delavcev, ter navede ukrepe za njihovo odpravljanje oziroma omejitav. Elaboratu mora biti priložena tudi PISNA IZJAVA o tem, kateri predpisani varstveni ukrepi, normativi, standardi in tehnični predpisi so bili upoštevani pri projektiranju sredstev za delo. (to določa 12. člen tega zakona)

- 13. člen določa da mora OZD in skupnost, ki projektira in proizvaja sredstva za delo, izdati SPRIČEVALO O VARNOSTI, v katerem mora navesti vse predpise, ki jih je upoštevala, in roke, v katerih je potrebno opravljati preiskave in preglede v skladu s členom 27.

- 14. člen pravi da mora OZD in skupnost, ki izbere ponudbo za uvoz sredstev za delo, pred uvozom ugotoviti, ali so sredstva za delo izdelana v skladu s predpisanimi varstvenimi ukrepi, normativi, standardi in tehničnimi predpisi in o tem izdati PISNO IZJAVO. Če ni predpisanih varstvenih ukrepov, normativov, standardov in tehničnih predpisov, mora pridobiti STROKOVNO OCENO o tem, ali sredstvo za delo izpolnjuje zahteve glede varstva pri delu.

- 15. člen določa, da mora OZD in skupnost opremiti vsako novo izdelano sredstvo za delo z NAVODILOM ZA VARNO UPORABO, PREIZKUŠANJE IN VZDRŽEVANJE, napisano v slovenskem jeziku.

Od leta 1974 je neizdaja listin varstva pri delu gospodarski prestopok zbiroma kaznivo dejanje. Vsaka OZD je dolžna izdati listine. Če jih sama ne more izdelati naroči izdelavo listin pri drugi specializirani OZD ali strokovni službi. Listine, ki so jih izdale OZD na podlagi zakona o varstvu pri delu, preverja organ inšpekcije dela. Če se dvomi v ustreznost izdelave listine, lahko zahtevamo strokovno oceno listine, ki jo poda pooblaščen OZD.

Moja pripravniška naloga, simulacija, je prva stopnja projektiranja izmeničnega motorskega pogona. Da bo ta projekt zaživel, bo potrebno najprej zgraditi močnostni pretvornik in potreben hardware, nabaviti motorje (iz uvoza) in napisati software. Ko bo sistem preizkušen, bo treba napisati ustrezno razvojno konstrukcijsko dokumentacijo. Ta dokumentacija bo morala vsebovati tudi po zakonu predpisane listine varstva pri delu.

Pri izmenični motorski pogon bodo sestavljali trije ločeni deli in sicer:

- regulacijski panel
- transistorski močnostni pretvornik
- motor

V nadaljevanju bom podal izjavo o upoštevanju varstvenih ukrepov in normativov ter navodilo za varno uporabo, preizkušanje in vzdrževanje transistorskega močnostnega pretvornika.

D L.

NA PODLAGI 13. ČLENA O VARSTVU PRI DELU (URADNI LIST SRS, ŠT. 32/74, 16/80, 25/86 IN 47/86 P.B.) IZDAJAMO

S P R I Č E V A L O O V A R N O S T I

1. NAZIV ORGANIZACIJE ZDRUŽENEGA DELA:

OZD ISKRA AVTOMATIKA, LJUBLJANA, STEGNE 15b

PROJEKTANT: TOZD Razvojni inštitut

IZVAJALEC: TOZD AVTOMATSKE IN VARILNE NAPRAVE

2. OSNOVNI TEHNIČNI PODATKI:

NAZIV IZDELKA IN TIP: MOP-Tr-6

KODA IZDELKA: 430 055 900

PRIKLJUČNA NAPETOST/FREKVENCA/MOČ 3x380V/50Hz, 3kW

STOPNJA MEHANSKE ZAŠČITE: IP 00

TEMPERATURA OKOLICE: V DELOVANJU OD 0° DO 45°, PRI

SKLADIŠČENJU OD -30° DO +70°

DIMENZIJE: 162 x 264 x 195

MASA: 10 kg

3. PRI IZDELAVI MOP-Tr-6

SMO UPOŠTEVALI POGOJE NAVEDENE V ELABORATU PROJEKTA

OZ. V TEHNIČNI DOKUMENTACIJI ŠT....

IN NASLEDNJE VARNOSTNE, SPLOŠNE IN TEHNIČNE UKREPE IN ZAHTEVE:

- PRAVILNIK O VARSTVENIH UKREPIH ZOPER NEVARNOSTI ELEKTRIČNEGA TOKA V DELOVNIH KRAJIH (UR. LIST ŠT. 107/47),
- PRAVILNIK O SPLOŠNIH UKREPIH IN NORMATIVIH ZA VARSTVO PRI DELU Z DELOVNIMI PRIPRAVAMI (UR. LIST ŠT. 18/67),
- PRAVILNIK O TEHNIŠKIH UKREPIH IN POGOJIH ZA NAPELJEVANJE ELEKTROENERGETSKIH INSTALACIJ V STAVBAH (UR. LIST SFRJ ŠT. 43/66),
- PRAVILNIK O TEHNIČNIH UKREPIH ZA ELEKTROENERGETSKE INSTALACIJE V INDUSTRIJI (UR. LIST SFRJ ŠT. 2/73),
- PRAVILNIK O TEHNIČNIH ZAHTEVAH IN PREIZKUŠANJU ELEKTRIČNE OPREME INDUSTRIJSKIH STROJEV (JUS N.53.001 TOČKE 13.1, 13.2, 13.3 IN 13.4, UR. LIST SFRJ ŠT. 10/87)
- POVEZOVALNI SISTEM MED NUMERIČNIMI KRMILNIMI NAPRAVAMI IN INDUSTRIJSKIMI STROJI (JUS N.53.010, UR. LIST SFRJ ŠT. 10/87).

5. PROIZVAJALEC ZAGOTAVLJA, DA JE MOP-Tr-6 VARNO ZA UPORABO OB DOSLEDNEM UPOŠTEVANJU NAVODIL ZA VARNO UPORABO, PREIZKUŠANJE IN VZDRŽEVANJE, KI SO PRILOŽENA K DOKUMENTACIJI.

DATUM: 24.6. 1988

ODGOVORNI PROJEKTANT:

ODGOVORNI IZVAJALEC:

ZIG:

NA PODLAGI 15. ČLENA ZAKONA O VARSTVU PRI DELU (URADNI LIST SRS,
ŠT. 32/74, 16/80, 25/86 IN 47/86 p.b) IZDAJAMO

NAVODILO

ZA VARNO UPORABO, PREIZKUŠANJE IN VZDRŽEVANJE

1. NAZIV ORGANIZACIJE ZDRUŽENEGA DELA:

OZD ISKRA AVTOMATIKA, LJUBLJANA, STEGNE 15b

PROJEKTANT: TOZD Razvojni inštitut

IZVAJALEC: TOZD AVTOMATSKE IN VARILNE NAPRAVE

2. NAZIV IZDELKA IN TIP: MOP-Tr-6

KODA IZDELKA: 430 055 900

3. TEHNIČNI PODATKI:

LETO IZDELAVE: 30.5. 1988

SERIJSKA ŠTEVILKA : 1

PRIKLJUČNA NAPETOST/FREKVENCA/MOČ 3x380V/50Hz/3kW

STOPNJA MEHANSKE ZAŠČITE: IP 00

TEMPERATURA OKOLICE: V DELOVANJU OD 0° DO 45°, PRI

SKLADIŠČENJU OD -30° DO +70°

DIMENZIJE: 162 x 264 x 195

MASA: 10 kg

ŽIVLJENSKA DOBA: 5 let

4. PRI IZDELAVI NAPRAVE SO BILI UPOŠTEVANI NASLEDNJI TEHNIŠKI
PREDPISI, JUGOSLOVANSKI STANDARDI IN PREDPISI O VARSTVU PRI
DELU:

- TEHNIŠKI PREDPIS ZA ELEKTRIČNO OPREMO INDUSTRIJSKIH STROJEV
(JUS N.S3.001, UR. LIST SFRJ ŠT. 10/87),
- TEHNIŠKI PREDPIS ZA STOPNJO ZAŠČITE ZA ELEKTRIČNE APARATE
(JUS N.A5.070),
- PRAVILNIK O TEHNIŠKIH UKREPIH IN POGOJIH ZA NAPELJEVANJE
ELEKTROENERGETSKIH INSTALACIJ V STAVBAH (UR. LIST SFRJ
ŠT.43/66),
- PRAVILNIK O TEHNIŠKIH UKREPIH ZA ELEKTROENERGETSKE
INSTALACIJE V INDUSTRIJI (UR.LIST ŠT. 2/73),

- PRAVILNIK O VARSTVENIH UKREPIH ZOPER NEVARNOST ELEKTRIČNEGA TOKA V DELOVNIH KRAJIH (UR. LIST SFRJ ŠT. 107/47)
- PRAVILNIK O SPLOŠNIH UKREPIH IN NORMATIVIH ZA VARSTVO PRI DELU Z DELOVNIMI PRIPRAVAMI (UR. LIST št. 18/67)
- POVEZOVALNI SISTEM MED NUMERIČNIMI KRMILNIMI NAPRAVAMI IN INDUSTRIJSKIMI STROJI (JUS. N.S3.010, UR. LIST SFRJ ŠT.10/87)

5. NAČIN VARNE UPORABE, PREIZKUŠANJA IN VZDRŽEVANJA: sledijo po navodilih v tehnični dokumentaciji izdelka

- MONTAŽO, SERVIS IN VZDRŽEVANJE LAHKO IZVAJA LE STROKOVNO USPOSOBLJENA IN POOBLAŠČENA OSEBA V SKLADU S TEHNIŠKO DOKUMENTACIJO IZDELKA.

DATUM: 24.6. 1988

ODGOVORNI PROJEKTANT:

ODGOVORNI IZVAJALEC:

ZIG:

2

D N E V N I K

Pripravnništvo sem začel prvega oktobra 1987.

Najprej sem delal na področju preučevanja kinematike in dinamike robotov, kmalu pa sem začel delati simulacijo regulacije sinhronskega motorja, kar je tudi vsebina moje pripravniške naloge.

V oktobru sem študiral dinamiko in kinematiko robotov. Prebral sem knjigi:

- D.J.Tod: Fundamentals of Robot Technology, Kogan Page 1986 in
- Richard P.Paul: Robot Manipulators : Mathematics, Programming and Control, MIT Press 1981

Novembra sem začel delo na področju simulacije delovanja in regulacije sinhronskih motorjev. Najprej sem vzel zelo poenostavljen sistem, ki sem ga, ob študiju knjige W.Leonarda: Control of Electrical Drives in člankov s tega področja ter posvetovanjih s sodelavci, postopoma dopolnjeval. To delo je trajalo vse do konca pripravništva. Poleg tega pa sem se v času pripravništva ukvarjal tudi z drugimi stvarmi.

Ker sem simuliral regulacijo sinhronskega motorja na računalniku HP-9500 v programskem jeziku Fortran 77, sem predelal knjigi:

- Mc. Gilton, R. Morgan: UNIX System in
- S. Pollack: Fortran 77

V januarju sem napisal simulacijska programa za določevanje časa ustavljanja in pojemka enosmernih in sinhronskih motorjev, v odvisnosti od zunanjih uporov.

Februarja sem se udeležil tečaja jezika C.

Marca sem začel po člankih študirati osnove pulzno širinske modulacije (PWM). Naslednji korak pri obravnavi delovanja sinhronskega motorja bo verjetno simulacija močnostnega napajalnika krmiljenega s PWM-om.

Aprila sem se ukvarjal s študijem večpolnih izmeničnih motorjev in navijanja motorjev.

Maja sem se osredotočil na inkrementalne dajalnike pozicije in metode izračuna hitrosti.

Ker je del teorije motorjev in regulacij skupne vsem motorjem , sem delno obdelal tudi osnove enosmernih in asinhronskih motorjev.

V času pripravništva sem se udeležil obveznega programa usposabljanja pripravnikov.

S potekom pripravništva sem zadovoljen. V svojem delovnem okolju se dobro počutim. Pri sodelavcih sem vedno našel na razumevanje in prijateljsko pomoč. Pogoji za delo so dobri. Vedno sem lahko delal na računaniku, imel dostop do dobre literature in se vključil tudi v informacijski sistem INDOK centra Iskre.

IV. LITERATURA

- /1/ W. Leonhard : CONTROL OF ELECTRICAL DRIVES, Springer-Verlag
1984
- /2/ W. Leonhard : MICROCOMPUTER CONTROL OF HIGH DYNAMIC
PERFORMANCE AC DRIVES - A SURVEY, Automatica Vol. 20, pp
1-19, 1986
- /3/ W. Schumacher : FULLY DIGITAL CONTROL OF INDUCTON MOTOR AS
SERVO DRIVE, Članek

PROGRAM SINHR

digitalna reg. sinh. motorja s tok. senzorjem in inkr. dajalcem

IMPLICIT NONE

```

INTEGER*2 STE1,SSTE1
INTEGER*4 ng1,ng,nx,npx,ny,npj,nx1(18),npx1(18),ny1(18),npj1(18)
INTEGER*4 i,ii,jj,i2,i2k,i3,i3k,i4,i5,i6,i7,i10,ndat,ost
INTEGER*4 stik,izp,stevn,stezv,adi,adepts
INTEGER*4 nper,s4,s5,s6
INTEGER*4 is13,is23,ixa,ixb,ieps
REAL*4 ymax,ymin,vl,vr,vd,vu,tnr1,nr1,isdmax,p
REAL*4 t,dt,om,omr,nr,pi,errh,kih,kph,isqr1,a,ismax,isdr,isqr2
REAL*4 errq,isq,kiq,kpq,isqr,errd,isd,isdr1,kid,kpd
REAL*4 isar,isbr,eps,islr,is2r,is3r,islp,is2p,is3p,is1,is2,is3,tz
REAL*4 isa,isb,us1,rs,ls,us2,gama,us3,us12,us23,us31,usum,tb1,m1
REAL*4 mb,mb1,mtrs,mtrd,mtrd1,omlp,om1,b,j,n,isdr2,tfin,nmax,ernn
REAL*4 fi0,kpu,kiu,usr,usmax,E,nn,erru,Us,amm
REAL*4 ymin1(18),ymax1(18),x1,x2,x5,x6,x7,most
REAL*4 isn,usn,s1,s2,s3,yme1,ymeps,ymdus,ydth,md
REAL*4 g1,g2,g3,g4,g5,g6,g7,iisd,iisq,iisdm,iisqm
REAL*4 q0h,q1h,q0q,q1q,q0d,q1d,q0u,q1u,errhp,errqp,errdp,errup
REAL*4 t0z,t0n,tk,tis,fakisq,mdmax,isdr1min,iisdr1min
REAL*4 is11,is21,is31,is12,is22,aml
REAL*4 is12p,is22p,l1,isfm,isdmmx,isdmmx1,isqmmx,isqmmx1
REAL*4 deps,eps1,eps2,xa,xb,deleps,deleps1,alfa,alfa1,alfa1x,l11
REAL*4 osteps,epsx,om2
REAL*4 isam,isbm,isqm,isdm,am,Usmot,x11,x21,E1,them,the1m,dthem
REAL*4 Disq,Disd,Dus,Dn,Dna,Depsm,Dkot,errumot,errdm,errqm
REAL*4 nmer,hit0
REAL*4 tb2,tb3,tb4,mb2,mb3,mb4,tnr2,tnr3,tnr4,nr2,nr3,nr4,ism
CHARACTER*20 text2,tx2(18)
CHARACTER*5 text1,tx1(18)
CHARACTER STE*2,IME*5,IIME*5
CHARACTER*8 a1,a2,a3,a33,a4,a44,a5,a55,a6,a7,a8,a9
CHARACTER*11 par
    
```

EQUIVALENCE (STE,STE1)

***** inicializacije

```

DATA is11,is21,is31,is12,is22,om1,om2,omlp/8*0./
DATA gama,pi,s1,s2,s3/2.0943951,3.141592654,2*1.,0./
DATA eps1,eps2,isq,isd/4*0./
DATA isum,isqm,isdm,the1m,them,dthem,isdr1min/7*0./
DATA i4,i5,i6,i7,stik,s4,s5,s6/8*0./
DATA errh,errq,errd,erru/4*0./
DATA isqr1,isqr2,isdr1,isdr2,aml/5*0./
DATA epsx,om,alfa1x,isqmmx,isdmmx/5*0./
DATA Dn/1*0./
    
```

```

DATA ng1/18/
DATA (ymax1(i),i=1,18)/18*0./
DATA (ymin1(i),i=1,18)/18*0./
DATA ny1/18*6/
    
```

```
DATA npy1/18*4/
DATA nx1/18*7/
DATA npx1/18*4/
DATA par/'psinhr'/
```

```
WRITE(6,'(//////////,TR3,"DIGITALNA REG. SINH.MOT.,
# UPOSTEVANJE TOK. SENZORJA in INKR. DAJALCA")')
```

***** zamenjava fila par

```
WRITE(6,'(//,TR10,NN,"ZAMENJAVA PODAT. FILA (Y/*...)")')
READ(5,'(A8)') a1
IF(a1.EQ.'y'.OR.a1.EQ.'Y')THEN
  WRITE(6,'($,R11)')par
  WRITE(6,'($," -> ")')
  READ(5,'(A11)')par
ENDIF
```

***** branje par

```
OPEN(10,ERR=1,ACCESS='SEQUENTIAL',FILE=par,STATUS='NEW')
GOTO 2
OPEN(10,ACCESS='SEQUENTIAL',FILE=par,STATUS='OLD')
CONTINUE
```

```
READ(10,'(7E11.5)') nn, isn, usn, Ls, Rs, fi0, J
READ(10,'(7E11.5)') b, usmax, isfm, tz, kph, kih, kpq
READ(10,'(7E11.5)') kiq, kpd, kid, kpu, kiu, tfin, dt
READ(10,'(I11,6E11.5)') izp, tb1, mb1, nr, tnr1, nr1, isdmax
READ(10,'(2E11.5,2I11,2E11.5,I11)') mtrs, p, stevn, stevz, tk, tis, adi
READ(10,'(2I11,5E11.5)') adeps, nper, fakisq, tb4, mb4, mtrd1, hit0
READ(10,'(7E11.5)') tnr2, nr2, tnr3, nr3, tnr4, nr4, tb2
READ(10,'(3E11.5)') mb2, tb3, mb3
CLOSE(10)
```

***** sprememba parametrov

```
WRITE(6,'(//,TR10,NN,"SPREMEMBA PARAMETROV (Y/*...)")')
READ(5,'(A8)') a1
IF(a1.EQ.'y'.OR.a1.EQ.'Y')THEN
```

```
WRITE(6,'(//,TR10,"MOTOR")')
PRINT 15,'Na(/m) =',nn,'Isn(A) =',isn,'Usn(U) =',usn,'Ls(H) =',
#Ls
PRINT 15,'Rs(ohm) =',Rs,'Fi(Us) =',fi0,'J(kgm2) =',J,'B(km2/s =',b
PRINT 15,'Mtrs(Nm =',mtrs,'Mtrd(Nm =',mtrd1,'p(par) =',p
```

```
WRITE(6,'(//,TR10,"PRETVORNIK")')
WRITE(6,15)'Isfm(A) =',isfm,'Usmax(U =',usmax,'Tz(s) =',tz,'Idmax
#(A =',isdmax
```

```
WRITE(6,'(//,TR10,"REGULATORJI")')
PRINT 15,'Kph =',kph,'Kih =',kih,'Kpu =',kpu,'Kiu =',
#,kiu
PRINT 15,'Kpq =',kpq,'Kiq =',kiq,'Kpd =',kpd,'Kid =',
#,kid
```

C WRITE(6, '(/,TR10,"INKR. DAJALEC")')

PRINT 16, 'adepts(b=' ,adepts

PRINT 15, 'nper(i =',nper

C WRITE(6, '(/,TR10,"OSTALO")')

PRINT 15, 'tfin(s)=' ,tfin, 'dt(s) =',dt, 'tb1(s) =',tb1, 'mb1(Nm)='

#,mb1

PRINT 15, 'tb2(s) =',tb2, 'mb2(Nm)=' ,mb2, 'tb3(s) =',tb3, 'mb3(Nm)='

#,mb3

PRINT 15, 'tb4(s) =',tb4, 'mb4(Nm)=' ,mb4

PRINT 15, 'nr(/m) =',nr, 'tnr1(s)=' ,tnr1, 'nr1(/m)=' ,nr1, 'tnr2(s)='

#,tnr2

PRINT 15, 'nr2(/m) =',nr2, 'tnr3(s)=' ,tnr3, 'nr3(/m)=' ,nr3, 'tnr4(s)

#=' ,tnr4

PRINT 15, 'nr4(/m) =',nr4

PRINT 15, 'tis(s) =',tis, 'fakisq =',fakisq

PRINT 15, 'hit0(/m=' ,hit0, 'tk(s) =',tk

PRINT 16, 't0n(*dt=' ,stevn, 't0z(*dt=' ,stevz, 'izp(*dt=' ,izp, 'adi(b

#it=' ,adi

C 15 FORMAT(\$,4(2X,A8,E10.4))

16 FORMAT(\$,4(2X,A8,I10))

C ***** branje

PRINT*, ' '

100 WRITE(6, '(\$,"parameter ")')

READ(5, '(A8,\$)') a6

IF(a6.EQ.'nn'.OR.a6.EQ.'Nn')THEN

WRITE(6, '(\$,E11.5)')nn

WRITE(6, '(\$," -> ")')

READ*,nn

GOTO 100

ELSEIF(a6.EQ.'isn'.OR.a6.EQ.'Isn')THEN

WRITE(6, '(\$,E11.5)')isn

WRITE(6, '(\$," -> ")')

READ*,isn

GOTO 100

ELSEIF(a6.EQ.'usn'.OR.a6.EQ.'Usn')THEN

WRITE(6, '(\$,E11.5)')usn

WRITE(6, '(\$," -> ")')

READ*,usn

GOTO 100

ELSEIF(a6.EQ.'ls'.OR.a6.EQ.'Ls')THEN

WRITE(6, '(\$,E11.5)')Ls

WRITE(6, '(\$," -> ")')

READ*,Ls

GOTO 100

ELSEIF(a6.EQ.'rs'.OR.a6.EQ.'Rs')THEN

WRITE(6, '(\$,E11.5)')Rs

WRITE(6, '(\$," -> ")')

READ*,Rs

GOTO 100

ELSEIF(a6.EQ.'fi'.OR.a6.EQ.'Fi')THEN

WRITE(6, '(\$,E11.5)')fi0

WRITE(6, '(\$," -> ")')

READ*,fi0

```
GOTO 100
ELSEIF(a6.EQ.'j'.OR.a6.EQ.'J')THEN
  WRITE(6, '($, E11.5)')J
  WRITE(6, '($, " -> ")')
  READ*, J
  GOTO 100
ELSEIF(a6.EQ.'b'.OR.a6.EQ.'B')THEN
  WRITE(6, '($, E11.5)')b
  WRITE(6, '($, " -> ")')
  READ*, b
  GOTO 100
ELSEIF(a6.EQ.'usmax'.OR.a6.EQ.'Usmax')THEN
  WRITE(6, '($, E11.5)')usmax
  WRITE(6, '($, " -> ")')
  READ*, usmax
  GOTO 100
ELSEIF(a6.EQ.'isfm'.OR.a6.EQ.'Isfm')THEN
  WRITE(6, '($, E11.5)')isfm
  WRITE(6, '($, " -> ")')
  READ*, isfm
  GOTO 100
ELSEIF(a6.EQ.'tz'.OR.a6.EQ.'Tz')THEN
  WRITE(6, '($, E11.5)')tz
  WRITE(6, '($, " -> ")')
  READ*, tz
  GOTO 100
ENDIF
IF(a6.EQ.'Kph'.OR.a6.EQ.'kph')THEN
  WRITE(6, '($, E11.5)')kph
  WRITE(6, '($, " -> ")')
  READ*, kph
  GOTO 100
ELSEIF(a6.EQ.'Kih'.OR.a6.EQ.'kih')THEN
  WRITE(6, '($, E11.5)')kih
  WRITE(6, '($, " -> ")')
  READ*, kih
  GOTO 100
ELSEIF(a6.EQ.'Kiq'.OR.a6.EQ.'kiq')THEN
  WRITE(6, '($, E11.5)')kiq
  WRITE(6, '($, " -> ")')
  READ*, kiq
  GOTO 100
ELSEIF(a6.EQ.'Kid'.OR.a6.EQ.'kid')THEN
  WRITE(6, '($, E11.5)')kid
  WRITE(6, '($, " -> ")')
  READ*, kid
  GOTO 100
ELSEIF(a6.EQ.'Kpd'.OR.a6.EQ.'kpd')THEN
  WRITE(6, '($, E11.5)')kpd
  WRITE(6, '($, " -> ")')
  READ*, kpd
  GOTO 100
ELSEIF(a6.EQ.'Kpq'.OR.a6.EQ.'kpq')THEN
  WRITE(6, '($, E11.5)')kpq
  WRITE(6, '($, " -> ")')
  READ*, kpq
```

```
GOTO 100
ELSEIF(a6.EQ.'Kpu'.OR.a6.EQ.'kpu')THEN
  WRITE(6,'($,E11.5)')kpu
  WRITE(6,'($," -> ")')
  READ*,kpu
  GOTO 100
ELSEIF(a6.EQ.'Kiu'.OR.a6.EQ.'kiu')THEN
  WRITE(6,'($,E11.5)')kiu
  WRITE(6,'($," -> ")')
  READ*,kiu
  GOTO 100
ELSEIF(a6.EQ.'Tfin'.OR.a6.EQ.'tfin')THEN
  WRITE(6,'($,E11.5)')tfin
  WRITE(6,'($," -> ")')
  READ*,tfin
  GOTO 100
ELSEIF(a6.EQ.'Dt'.OR.a6.EQ.'dt')THEN
  WRITE(6,'($,E11.5)')dt
  WRITE(6,'($," -> ")')
  READ*,dt
  GOTO 100
ELSEIF(a6.EQ.'Izp'.OR.a6.EQ.'izp')THEN
  WRITE(6,'($,111)')izp
  WRITE(6,'($," -> ")')
  READ*,izp
  GOTO 100
ELSEIF(a6.EQ.'tb1'.OR.a6.EQ.'Tb1')THEN
  WRITE(6,'($,E11.5)')tb1
  WRITE(6,'($," -> ")')
  READ*,tb1
  GOTO 100
ELSEIF(a6.EQ.'mb1'.OR.a6.EQ.'Mb1')THEN
  WRITE(6,'($,E11.5)')mb1
  WRITE(6,'($," -> ")')
  READ*,mb1
  GOTO 100
ENDIF
IF(a6.EQ.'nr'.OR.a6.EQ.'Nr')THEN
  WRITE(6,'($,E11.5)')nr
  WRITE(6,'($," -> ")')
  READ*,nr
  GOTO 100
ELSEIF(a6.EQ.'tnr1'.OR.a6.EQ.'Tnr1')THEN
  WRITE(6,'($,E11.5)')tnr1
  WRITE(6,'($," -> ")')
  READ*,tnr1
  GOTO 100
ELSEIF(a6.EQ.'nr1'.OR.a6.EQ.'nr1')THEN
  WRITE(6,'($,E11.5)')nr1
  WRITE(6,'($," -> ")')
  READ*,nr1
  GOTO 100
ELSEIF(a6.EQ.'idmax'.OR.a6.EQ.'Idmax')THEN
  WRITE(6,'($,E11.5)')idmax
  WRITE(6,'($," -> ")')
  READ*,idmax
```

```

GOTO 100
ELSEIF(a6.EQ.'mtrs'.OR.a6.EQ.'Mtrs')THEN
  WRITE(6,('$,E11.5')mtrs
  WRITE(6,('$," -> "')'
  READ*,mtrs
  GOTO 100
ELSEIF(a6.EQ.'p'.OR.a6.EQ.'P')THEN
  WRITE(6,('$,E11.5')p
  WRITE(6,('$," -> "')'
  READ*,p
  GOTO 100
ELSEIF(a6.EQ.'t0n'.OR.a6.EQ.'T0n')THEN
  WRITE(6,('$,I11')stevn
  WRITE(6,('$," -> "')'
  READ*,stevn
  GOTO 100
ELSEIF(a6.EQ.'t0z'.OR.a6.EQ.'T0z')THEN
  WRITE(6,('$,I11')stevz
  WRITE(6,('$," -> "')'
  READ*,stevz
  GOTO 100
ELSEIF(a6.EQ.'tk'.OR.a6.EQ.'Tk')THEN
  WRITE(6,('$,E11.5')tk
  WRITE(6,('$," -> "')'
  READ*,tk
  GOTO 100
ELSEIF(a6.EQ.'tis'.OR.a6.EQ.'Tis')THEN
  WRITE(6,('$,E11.5')tis
  WRITE(6,('$," -> "')'
  READ*,tis
  GOTO 100
ELSEIF(a6.EQ.'adi'.OR.a6.EQ.'Adi')THEN
  WRITE(6,('$,I11')adi
  WRITE(6,('$," -> "')'
  READ*,adi
  GOTO 100
ELSEIF(a6.EQ.'adepts'.OR.a6.EQ.'Adepts')THEN
  WRITE(6,('$,I11')adepts
  WRITE(6,('$," -> "')'
  READ*,adepts
  GOTO 100
ENDIF
IF(a6.EQ.'nper'.OR.a6.EQ.'Nper')THEN
  WRITE(6,('$,I11')nper
  WRITE(6,('$," -> "')'
  READ*,nper
  GOTO 100
ELSEIF(a6.EQ.'fakisq'.OR.a6.EQ.'Fakisq')THEN
  WRITE(6,('$,E11.5')fakisq
  WRITE(6,('$," -> "')'
  READ*,fakisq
  GOTO 100
ELSEIF(a6.EQ.'mtrd'.OR.a6.EQ.'Mtrd')THEN
  WRITE(6,('$,E11.5')mtrd1
  WRITE(6,('$," -> "')'
  READ*,mtrd1

```

```

GOTO 100
ELSEIF(a6.EQ.'hit0'.OR.a6.EQ.'Hit0')THEN
  WRITE(6,'($,E11.5)')hit0
  WRITE(6,'($," -> ")')
  READ*,hit0
  GOTO 100
ELSEIF(a6.EQ.'tb2'.OR.a6.EQ.'Tb2')THEN
  WRITE(6,'($,E11.5)')tb2
  WRITE(6,'($," -> ")')
  READ*,tb2
  GOTO 100
ELSEIF(a6.EQ.'mb2'.OR.a6.EQ.'Mb2')THEN
  WRITE(6,'($,E11.5)')mb2
  WRITE(6,'($," -> ")')
  READ*,mb2
  GOTO 100
ELSEIF(a6.EQ.'mb3'.OR.a6.EQ.'Mb3')THEN
  WRITE(6,'($,E11.5)')mb3
  WRITE(6,'($," -> ")')
  READ*,mb3
  GOTO 100
ELSEIF(a6.EQ.'tb3'.OR.a6.EQ.'Tb3')THEN
  WRITE(6,'($,E11.5)')tb3
  WRITE(6,'($," -> ")')
  READ*,tb3
  GOTO 100
ELSEIF(a6.EQ.'mb4'.OR.a6.EQ.'Mb4')THEN
  WRITE(6,'($,E11.5)')mb4
  WRITE(6,'($," -> ")')
  READ*,mb4
  GOTO 100
ELSEIF(a6.EQ.'tb4'.OR.a6.EQ.'Tb4')THEN
  WRITE(6,'($,E11.5)')tb4
  WRITE(6,'($," -> ")')
  READ*,tb4
  GOTO 100
ENDIF
IF(a6.EQ.'tnr2'.OR.a6.EQ.'Tnr2')THEN
  WRITE(6,'($,E11.5)')tnr2
  WRITE(6,'($," -> ")')
  READ*,tnr2
  GOTO 100
ELSEIF(a6.EQ.'nr2'.OR.a6.EQ.'Nr2')THEN
  WRITE(6,'($, I11)')nr2
  WRITE(6,'($," -> ")')
  READ*,nr2
  GOTO 100
ELSEIF(a6.EQ.'tnr3'.OR.a6.EQ.'Tnr3')THEN
  WRITE(6,'($,E11.5)')tnr3
  WRITE(6,'($," -> ")')
  READ*,tnr3
  GOTO 100
ELSEIF(a6.EQ.'nr3'.OR.a6.EQ.'Nr3')THEN
  WRITE(6,'($,E11.5)')nr3
  WRITE(6,'($," -> ")')
  READ*,nr3

```



```

GOTO 100
ELSEIF(a6.EQ.'tnr4'.OR.a6.EQ.'Tnr4')THEN
  WRITE(6,'($,E11.5)')tnr4
  WRITE(6,'($," -> ")')
  READ*,tnr4
  GOTO 100
ELSEIF(a6.EQ.'nr4'.OR.a6.EQ.'Nr4')THEN
  WRITE(6,'($,E11.5)')nr4
  WRITE(6,'($," -> ")')
  READ*,nr4
  GOTO 100
ENDIF
C
C
C
C
C ***** Optimizacija regulatorjev
C
C WRITE(6,'(//,TR10,NN,"OPTIMIZACIJA REGULATORJEV (y/*...)"')
C READ(5,'(A8)') a2
C IF(a2.EQ.'y'.OR.a2.EQ.'Y')THEN
C
C ***** tokovna regulatorja
C WRITE(6,'(/,TR15,NN,"Tokovna regulatorja (y/*...)"')
C READ(5,'(A8)') a3
C IF(a3.EQ.'y'.OR.a3.EQ.'Y')THEN
C   s1=0.
C   s2=s1
C   WRITE(6,'(TR20,NN,"sprememba parametrov (y/*...)"')
C   READ(5,'(A8)') a33
C   IF(a33.EQ.'y'.OR.a33.EQ.'Y')THEN
C     WRITE(6,'(TR25,NN,"kpg =")')
C     WRITE(6,'($,E11.5)')kpg
C     WRITE(6,'($," -> ")')
C     READ*,kpg
C     WRITE(6,'(TR25,NN,"kiq =")')
C     WRITE(6,'($,E11.5)')kiq
C     WRITE(6,'($," -> ")')
C     READ*,kiq
C     WRITE(6,'(TR25,NN,"kpd =")')
C     WRITE(6,'($,E11.5)')kpd
C     WRITE(6,'($," -> ")')
C     READ*,kpd
C     WRITE(6,'(TR25,NN,"kid =")')
C     WRITE(6,'($,E11.5)')kid
C     WRITE(6,'($," -> ")')
C     READ*,kid
C   ENDIF
C ENDIF
C
C ***** hitrostni regulator
C WRITE(6,'(/,TR15,NN,"hitrostni regulator (y/*...)"')
C READ(5,'(A8)') a4
C IF(a4.EQ.'y'.OR.a4.EQ.'Y')THEN
C   s1=1.
C   s2=s1

```

```

WRITE(6,'(TR20,NN,"sprememba parametrov (y/*...)"')
READ(5,'(A8)') a44
IF(a44.EQ.'y'.OR.a44.EQ.'Y')THEN
  WRITE(6,'(TR25,NN,"kph ="')
  WRITE(6,'($,E11.5)')kph
  WRITE(6,'($," -> ")')
  READ*,kph
  WRITE(6,'(TR25,NN,"kih ="')
  WRITE(6,'($,E11.5)')kih
  WRITE(6,'($," -> ")')
  READ*,kih
ENDIF
IF(a3.EQ.'y'.OR.a3.EQ.'Y') a3='n'
ENDIF

```

C
C

```

***** regulator Us
WRITE(6,'(//,TR15,NN,"regulator Us (y/*...)"')
READ(5,'(A8)') a5
IF(a5.EQ.'y'.OR.a5.EQ.'Y')THEN
  s1=1.
  s2=s1
  IF(a4.NE.'y'.OR.a4.NE.'Y')s3=1.
  WRITE(6,'(TR20,NN,"sprememba parametrov (y/*...)"')
  READ(5,'(A8)') a55
  IF(a55.EQ.'y'.OR.a55.EQ.'Y')THEN
    WRITE(6,'(TR25,NN,"kpu ="')
    WRITE(6,'($,E11.5)')kpu
    WRITE(6,'($," -> ")')
    READ*,kpu
    WRITE(6,'(TR25,NN,"kiu ="')
    WRITE(6,'($,E11.5)')kiu
    WRITE(6,'($," -> ")')
    READ*,kiu
  ENDIF
  IF(a3.EQ.'y'.OR.a3.EQ.'Y') a3='n'
ENDIF

```

C
C

```

***** referenci tokovnih regulatorjev
IF(a3.EQ.'y'.OR.a3.EQ.'Y')THEN
  WRITE(6,'(//,TR15,"reference tokov : ")')
  WRITE(6,'(TR20,NN,"Isqr(A) ="')
  READ*,isqr1
  WRITE(6,'(TR20,NN,"Isdr(A) ="')
  READ*,isdri
ENDIF
ENDIF

```

C
C

```

IF(a3.EQ.'y'.OR.a3.EQ.'Y') GOTO 110

```

C
C
C

```

***** sprememba ref. hitrosti
WRITE(6,'(//,TR10,NN,"SPREMEMBA REFERENCE HITROSTI (y/*...)"')
READ(5,'(A8)') a8
IF(a8.EQ.'y'.OR.a8.EQ.'Y')THEN
  WRITE(6,'(TR15,NN,"nr(/min) ="')
  WRITE(6,'($,E11.5)')nr

```

```
WRITE(6, '( $, " -> ") ' )  
READ*,nr
```

```
WRITE(6, '(TR15,NN,"tnr1(s) =") ' )  
WRITE(6, '( $,E11.5) ' )tnr1  
WRITE(6, '( $, " -> ") ' )  
READ*,tnr1  
WRITE(6, '(TR15,NN,"nr1(/min) =") ' )  
WRITE(6, '( $,E11.5) ' )nr1  
WRITE(6, '( $, " -> ") ' )  
READ*,nr1
```

```
WRITE(6, '(TR15,NN,"tnr2(s) =") ' )  
WRITE(6, '( $,E11.5) ' )tnr2  
WRITE(6, '( $, " -> ") ' )  
READ*,tnr2  
WRITE(6, '(TR15,NN,"nr2(/min) =") ' )  
WRITE(6, '( $,E11.5) ' )nr2  
WRITE(6, '( $, " -> ") ' )  
READ*,nr2
```

```
WRITE(6, '(TR15,NN,"tnr3(s) =") ' )  
WRITE(6, '( $,E11.5) ' )tnr3  
WRITE(6, '( $, " -> ") ' )  
READ*,tnr3  
WRITE(6, '(TR15,NN,"nr3(/min) =") ' )  
WRITE(6, '( $,E11.5) ' )nr3  
WRITE(6, '( $, " -> ") ' )  
READ*,nr3
```

```
WRITE(6, '(TR15,NN,"tnr4(s) =") ' )  
WRITE(6, '( $,E11.5) ' )tnr4  
WRITE(6, '( $, " -> ") ' )  
READ*,tnr4  
WRITE(6, '(TR15,NN,"nr4(/min) =") ' )  
WRITE(6, '( $,E11.5) ' )nr4  
WRITE(6, '( $, " -> ") ' )  
READ*,nr4
```

ENDIF

```
*****  
***** sprememba obremenitve
```

```
110 WRITE(6, '( //,TR10,NN,"SPREMEMBA OBREMNITVE (y/*...)' ) ' )
```

```
READ(5, '(A8) ' ) a7  
IF(a7.EQ.'y'.OR.a7.EQ.'Y')THEN  
WRITE(6, '(TR15,NN,"tb1(s)=") ' )  
WRITE(6, '( $,E11.5) ' )tb1  
WRITE(6, '( $, " -> ") ' )  
READ*,tb1  
WRITE(6, '(TR15,NN,"mb(Nm)=") ' )  
WRITE(6, '( $,E11.5) ' )mb1  
WRITE(6, '( $, " -> ") ' )  
READ*,mb1
```

```
WRITE(6, '(TR15,NN,"tb2(s)=") ' )  
WRITE(6, '( $,E11.5) ' )tb2
```

```
WRITE(6, '($, " -> ")')  
READ*, tb2  
WRITE(6, ' (TR15, NN, "mb2(Nm)=")')  
WRITE(6, '($, E11.5)') mb2  
WRITE(6, '($, " -> ")')  
READ*, mb2
```

```
WRITE(6, ' (TR15, NN, "tb3(s)=")')  
WRITE(6, '($, E11.5)') tb3  
WRITE(6, '($, " -> ")')  
READ*, tb3  
WRITE(6, ' (TR15, NN, "mb3(Nm)=")')  
WRITE(6, '($, E11.5)') mb3  
WRITE(6, '($, " -> ")')  
READ*, mb3
```

```
WRITE(6, ' (TR15, NN, "tb4(s)=")')  
WRITE(6, '($, E11.5)') tb4  
WRITE(6, '($, " -> ")')  
READ*, tb4  
WRITE(6, ' (TR15, NN, "mb4(Nm)=")')  
WRITE(6, '($, E11.5)') mb4  
WRITE(6, '($, " -> ")')  
READ*, mb4
```

ENDIF

```
***** sprememba casov
```

```
WRITE(6, ' (//, TR10, NN, "SPREMEMBA CASOV (y/*...)"')  
READ(5, '(A8)') a9  
IF(a9.EQ.'y'.OR.a9.EQ.'Y') THEN  
WRITE(6, ' (TR15, NN, "tfin(s)=")')  
WRITE(6, '($, E11.5)') tfin  
WRITE(6, '($, " -> ")')  
READ*, tfin  
WRITE(6, ' (TR15, NN, "dt(s)=")')  
WRITE(6, '($, E11.5)') dt  
WRITE(6, '($, " -> ")')  
READ*, dt  
WRITE(6, ' (TR15, NN, "t0n(*dt)=")')  
WRITE(6, '($, I11)') stevn  
WRITE(6, '($, " -> ")')  
READ*, stevn  
WRITE(6, ' (TR15, NN, "t0z(*dt)=")')  
WRITE(6, '($, I11)') stevz  
WRITE(6, '($, " -> ")')  
READ*, stevz  
WRITE(6, ' (TR15, NN, "izpis(*dt)=")')  
WRITE(6, '($, I11)') izp  
WRITE(6, '($, " -> ")')  
READ*, izp
```

ENDIF

```
***** vpis parametrov
```

```
OPEN(10,ACCESS='SEQUENTIAL',FILE=par,STATUS='OLD')
```

```
WRITE(10,'(7E11.5)') nn,ian,usn,La,Rs,fi0,J
WRITE(10,'(7E11.5)') b,usmax,isfm,tz,kph,kih,kpq
WRITE(10,'(7E11.5)') kiq,kpd,kid,kpu,kiu,tfin,dt
WRITE(10,'(I11,6E11.5)') izp,tb1,mb1,nr,tnr1,nr1,isdmax
WRITE(10,'(2E11.5,2I11,2E11.5,I11)') mtrs,p,stevn,stevez,tk,tis,adi
WRITE(10,'(2I11,5E11.5)') adeps,nper,fakisq,tb4,mb4,mtrd1,hit0
WRITE(10,'(7E11.5)') tnr2,nr2,tnr3,nr3,tnr4,nr4,tb2
WRITE(10,'(3E11.5)') mb2,tb3,mb3
CLOSE(10)
```

|||||

***** glavni del programa

***** odprtje datotek pod-

```
ng=ng1
ndat=ng/6
ost=MOD(ng,6)
IF(ndat.EQ.0.AND.ost.EQ.0)THEN
  WRITE(6,'(//////////,TR20,"NI PODATKOV ZA GRAFIKO")')
  GOTO 1000
ELSEIF(ost.NE.0)THEN
  ndat=ndat+1
ENDIF
```

```
i4=0
DO i=1,ndat
  STE1=2**13+2**12+i*2**8
  IME='ggg'//STE
  OPEN(i,ERR=10,ACCESS='SEQUENTIAL',FILE=IME,STATUS='NEW')
  GOTO 20
  OPEN(i,ACCESS='SEQUENTIAL',FILE=IME,STATUS='OLD')
CONTINUE
ENDDO
```

simulacija

```
t0z=dt*stevez
t0n=dt*stevn
ismax=isfm*3./SQRT(2.)

i5=izp
i6=stevez
i7=stevn

mdmax=ismax*fi0

l1=isfm*SQRT(2.)/2.**(adi-1)

l11=2.**(adeps-1)
deps=2.*pi/(20.*nper)
```

```
DO WHILE(t.LE.tfin)
```

//////////////////////////////////// t0n

```
IF(i7.EQ.stevn)THEN
```

```
***** merjena pozicija
ieps=eps1/deps
eps=ieps*deps
```

```
***** odzemanje tokov
~~~~~ AD kvantizacija
is13=is12/11
is1=is13*11
```

```
is23=is22/11
is2=is23*11
```

```
~~~~~ izracun is3
is3=-is1-is2
```

```
***** transformacija polja
~~~~~ redukcija faz
isa=3./2.*is1
isb=SQRT(3.)/2.*(is2-is3)
```

```
~~~~~ demodulacija
isq=isb*COS(eps)-isa*SIN(eps)
isd=isa*COS(eps)+isb*SIN(eps)
```

```
***** omejitveni pogoj a
IF(ABS(isd).GT.ABS(fakisq*isdmax))THEN
  a=0.
ELSE
  a=SQRT((fakisq*isdmax)**2-isd**2)
ENDIF
```

```
ENDIF
```

//////////////////////////////////// t0z

```
IF(i6.EQ.stevz)THEN
```

```
i6=0
```

```
***** izracun hitrosti
~~~~~ izracun deleps
```

```
deleps=eps-epsx
IF(epsx.LT.0.)THEN
  IF(deleps.GT.pi) deleps=-2*pi+deleps
ELSEIF(epsx.GT.0.)THEN
  IF(deleps.LT.-pi) deleps=2*pi+deleps
ENDIF
```

```
epsx=eps
```

```

C
C      eps2=eps1*nper
C
C      ~~~~~ AD analog sig. a in b
C      ixa=sin(eps2)*l11
C      xa=ixa/l11
C      ixb=cos(eps2)*l11
C      xb=ixb/l11
C
C      ~~~~~ izracun in popravek kota alfa
C      alfa=ATAN2(xa,xb)
C
C      IF(eps2.GT.0.)THEN
C        IF(alfa.LT.0.) alfa=alfa+pi
C        IF(xa.LT.0.) alfa=alfa+pi
C        IF(alfa.EQ.0..AND.xb.EQ.-1.) alfa=pi
C      ELSEIF(eps2.LT.0.)THEN
C        IF(alfa.GT.0.) alfa=alfa-pi
C        IF(xa.GT.0.) alfa=alfa-pi
C        IF(alfa.EQ.0..AND.xb.EQ.-1.) alfa=-pi
C      ENDIF
C
C      ~~~~~ kot alfa1
C      alfa1=AMOD(alfa,pi/10.)/nper
C      IF(xb.EQ.0..AND.ABS(xa).NE.1.)THEN
C        IF(om.GT.0.) alfa1=deps
C        IF(om.LT.0.) alfa1=-deps
C      ENDIF
C
C      ~~~~~ hitrost
C      om=(alfa1-alfa1x+deleps)/t0z
C
C      alfa1x=alfa1
C
C      IF(s1.EQ.1.)THEN
C
C        ***** sprememba reference hitrosti
C        IF(t.GE.tnr4)THEN
C          nr=nr4
C        ELSEIF(t.GE.tnr3)THEN
C          nr=nr3
C        ELSEIF(t.GE.tnr2)THEN
C          nr=nr2
C        ELSEIF(t.GE.tnr1)THEN
C          nr=nr1
C        ENDIF
C
C        omr=nr*2.*pi/60.
C
C        ***** regulator hitrosti
C        errhp=errh
C        errh=omr-om
C
C        q0h=kph+kih*t0z/2.

```

```
q1h=kih*t0z/2.-kph  
isqr1=isqr1+q0h*errh+q1h*errhp
```

```
***** omejitev isqr1  
IF(isqr1.GT.a)THEN  
  isqr1=a  
ELSEIF(isqr1.LT.-a)THEN  
  isqr1=-a  
ENDIF
```

```
ENDIF
```

```
IF(s2.EQ.1.)THEN
```

```
***** izracun Us  
iisd=SQRT(2.)/3.*isd  
iisq=SQRT(2.)/3.*isq
```

```
E=1./SQRT(2.)*om*fi0  
x1=iisq*Rs+E+iisd*om*Ls  
x2=-om*Ls*iisq+Rs*iisd  
Us=SQRT(x1**2+x2**2)
```

```
***** regulator Us  
errup=erru  
erru=usmax-Us
```

```
q0u=kpu+kiu*t0z/2.  
q1u=kiu*t0z/2.-kpu  
isdrl=isdrl+q0u*erru+q1u*errup
```

```
***** izracun isdrlmin  
x6=om*Ls*E+2.*om*Ls*Rs*iisq  
x7=Rs**2.-om**2.*Ls**2.  
iisdrlmin=x6/x7  
isdrlmin=3./SQRT(2.)*iisdrlmin
```

```
***** omejitev isdrl  
IF(isdrl.LT.-ABS(isdmax)) isdrl=-ABS(isdmax)  
IF(erru.LT.0..AND.isdrl.LT.isdrlmin) isdrl=isdrlmin  
IF(isdrl.GT.0.) isdrl=0.
```

```
ENDIF
```

```
ENDIF
```

```
//////////////////////////////////// t0n
```

```
IF(i7.EQ.stevn)THEN  
  i7=0
```

```
***** regulator isq  
errqp=errq  
errq=isqr1-isq
```



```
C
C
C      q0q=kpq+kiq*t0n/2.
C      q1q=kiq*t0n/2.-kpq
C      isqr2=isqr2+q0q*errq+q1q*errqp
C
C      ***** omejitev isqr2
C      IF(isqr2.GT.a)THEN
C        isqr2=a
C      ELSEIF(isqr2.LT.-a)THEN
C        isqr2=-a
C      ENDIF
C
C      ***** regulator isd
C      errdp=errd
C      errd=isdr1-isd
C
C      q0d=kpd+kid*t0n/2.
C      q1d=kid*t0n/2.-kpd
C      isdr2=isdr2+q0d*errd+q1d*errdp
C
C      ***** omejitev isdr2
C      IF(isdr2.GT.ABS(isdmax))THEN
C        isdr2=ABS(isdmax)
C      ELSEIF(isdr2.LT.-ABS(isdmax))THEN
C        isdr2=-ABS(isdmax)
C      ENDIF
C
C      ***** kompenzacija zakasntve inverterja
C      isdr=isdr2-om*tk*isqr2
C      isqr=isqr2+om*tk*isdr2
C
C      ***** inverzna transformacija polja
C      ~~~~~ modulacija
C      isar=isdr*COS(eps)-isqr*SIN(eps)
C      isbr=isqr*COS(eps)+isdr*SIN(eps)
C
C      ~~~~~ razcepitev faz
C      is1r=2./3.*isar
C      is2r=-1./3.*isar+1./SQRT(3.)*isbr
C      is3r=-1./3.*isar-1./SQRT(3.)*isbr
C
C      ENDIF
C
C      ////////////////////////////////////// dt
C
C      ***** tranz. konverter
C      is1p=(is1r-is1l)/tz
C      is1l=is1l+is1p*dt
C
C      is2p=(is2r-is2l)/tz
C      is2l=is2l+is2p*dt
C
C      is3p=(is3r-is3l)/tz
C      is3l=is3l+is3p*dt
C
```

```

C ***** tok. senzor
C is12p=(is11-is12)/tis
C is12=is12+is12p*dt
C
C is22p=(is21-is22)/tis
C is22=is22+is22p*dt
C
C ***** motor
C ***** transformacija polja motorja
C ~~~~~ redukcija faz
C isam=3./2.*is11
C isbm=SQRT(3.)/2.*(is21-is31)
C
C ~~~~~ demodulacija
C isqm=isbm*COS(eps1)-isam*SIN(eps1)
C isdm=isam*COS(eps1)+isbm*SIN(eps1)
C
C ***** nasicenje pretvornika
C ~~~~~ isdm
C IF (ABS(isdm).GT.ismax) THEN
C   isdmmax1=(ABS(isdm)-ismax)*100./ismax
C   IF (isdmmax1.GT.isdmmax) isdmmax=isdmmax1
C   s5=1
C ENDIF
C
C ~~~~~ isqm
C am=SQRT(ismax**2-isdm**2)
C IF (ABS(isqm).GT.am) THEN
C   isqmmax1=(ABS(isqm)-am)*100./am
C   IF (isqmmax1.GT.isqmmax) THEN
C     isqmmax=isqmmax1
C     am1=am
C   ENDIF
C   s4=1
C ENDIF
C
C ***** statično trenje
C md=isqm*fi0
C IF (om2.EQ.0.) THEN
C   mtrd=0.
C   IF (ABS(md).LE.mtrs) md=0.
C ELSE
C   mtrd=mtrd1
C ENDIF
C
C ***** pasivno breme + dinamično trenje
C ~~~~~ vlijupevanje zunanjih bremen
C IF (t.GE.tb4) THEN
C   mb=ABS(mb4)+ABS(mtrd)
C ELSE IF (t.GE.tb3) THEN
C   mb=ABS(mb3)+ABS(mtrd)
C ELSE IF (t.GE.tb2) THEN
C   mb=ABS(mb2)+ABS(mtrd)
C ELSE IF (t.GE.tb1) THEN
C   mb=ABS(mb1)+ABS(mtrd)
C ELSE

```

```

        mb=ABS(mtrd)
ENDIF
C
C
C ~~~~~ celotni bremenski navor
IF (ABS(om2).LT.hit0.AND.mb.GT.mdmax) THEN
    om1=0.
ELSE
    IF (om2.GT.0.) THEN
        m1=-mb
    ELSEIF (om2.LT.0.) THEN
        m1=mb
    ENDIF
    om1p=(md-b*om2+m1)/J
    om1=om1+om1p*dt
ENDIF

```

```

C
C ***** k. hitrost in pozicija rotorja
om2=om1/p

```

```

C
C eps1=eps1+om2*dt
IF (eps1.GE.2.*pi) eps1=eps1-2.*pi
IF (eps1.LE.-2.*pi) eps1=eps1+2.*pi

```

|||||

```

C
C ***** izracun max vrednosti rezultatov
C ***** in izpis rezultatov posameznih tekov

```

```

IF (i5.EQ.izp) THEN
    i5=0

```

```

C
C ***** izracun vrednosti za izpis

```

```

Disq=isq-isqm
Disd=isd-isdm
nmer=om*60./(2.*pi)
n=om2*60./(2.*pi)
Dna=nmer-n
IF (n.NE.0.) Dn=Dna/n*100.
Depsm=eps-eps1
Dkot=alfa-AMOD(eps2,2.*pi)

```

```

errn=nr-n

```

```

C
C ***** preostali navor

```

```

amm=(fakisq*isdmax)**2.-isdmax**2.
IF (amm.GT.0.) THEN
    most=(SQRT(amm)-ABS(isqm))*fi0
ELSE
    most=0.
ENDIF

```

```

C
C ***** Usmot

```

```

iisdm=SQRT(2.)/3.*isdmax
iisqm=SQRT(2.)/3.*isqm

```

```

E1=1./SQRT(2.)*om2*fi0
x11=iisqm*Rs+E1+iisdm*om2*Ls
x21=-om2*Ls*iisqm+Rs*iisdm
Usmot=SQRT(x11**2+x21**2)

```

```

errumot=Usmax-Usmot
Dus=Us-Usmot

```

```

***** dthem
IF(isdm.NE.0..AND.isqm.NE.0.)THEN
  the1m=ATAN(ABS(om2)*Ls/Rs)
  them=ATAN2(ABS(x21),ABS(x11))
  dthem=the1m-them
  IF(dthem.LT.0..AND.ABS(n).GE.ABS(nn)) s6=1
ENDIF

```

```

errqm=isqr1-isqm
errndm=isdr1-isdm

```

```

***** trenutne napetosti
~~~~~ fazne
us1=SQRT(2.)*Usmot*COS(eps1)
us2=SQRT(2.)*Usmot*COS(eps1-gama)
us3=SQRT(2.)*Usmot*COS(eps1-2.*gama)

```

```

~~~~~ medfazne
us12=us1-us2
us23=us2-us3
us31=us3-us1

```

```

~~~~~ vsota
usum=us1+us2+us3
IF(usum.LT.0.001) usum=0.

```

```

***** vsota tokov
isum=is11+is21+is31
IF(ABS(isum).LT.0.001) isum=0.

```

```

***** opis rezultatov v pod. file

```

```

IF(s1.EQ.1.)THEN
  IF(s3.EQ.0.)THEN

```

```

***** normalni tek
~~~~~ v datoteki pod1
WRITE(1, '(7E11.5)') t,n,errn,eps1,isqm,isdm,Usmot
IF(n.GT.ymax1(1)) ymax1(1)=n
IF(n.LT.ymin1(1)) ymin1(1)=n
IF(errn.GT.ymax1(2)) ymax1(2)=errn
IF(errn.LT.ymin1(2)) ymin1(2)=errn
IF(eps1.GT.ymax1(3)) ymax1(3)=eps1
IF(eps1.LT.ymin1(3)) ymin1(3)=eps1
IF(isqm.GT.ymax1(4)) ymax1(4)=isqm
IF(isqm.LT.ymin1(4)) ymin1(4)=isqm

```

```
IF(isdm.GT.ymax1(5)) ymax1(5)=isdm
IF(isdm.LT.ymin1(5)) ymin1(5)=isdm
IF(Usmot.GT.ymax1(6)) ymax1(6)=Usmot
IF(Usmot.LT.ymin1(6)) ymin1(6)=Usmot
```

```
~~~~~ v datoteki pod2
WRITE(2,'(7E11.5)') t,is11,is21,us1,us2,us12,usum
IF(is11.GT.ymax1(7)) ymax1(7)=is11
IF(is11.LT.ymin1(7)) ymin1(7)=is11
IF(is21.GT.ymax1(8)) ymax1(8)=is21
IF(is21.LT.ymin1(8)) ymin1(8)=is21
IF(us1.GT.ymax1(9)) ymax1(9)=us1
IF(us1.LT.ymin1(9)) ymin1(9)=us1
IF(us2.GT.ymax1(10)) ymax1(10)=us2
IF(us2.LT.ymin1(10)) ymin1(10)=us2
IF(us12.GT.ymax1(11)) ymax1(11)=us12
IF(us12.LT.ymin1(11)) ymin1(11)=us12
IF(usum.GT.ymax1(12)) ymax1(12)=usum
IF(usum.LT.ymin1(12)) ymin1(12)=usum
```

```
~~~~~ v datoteki pod3
WRITE(3,'(7E11.5)') t,n,nmer,Dn,Dna,Disq,most
IF(n.GT.ymax1(13)) ymax1(13)=n
IF(n.LT.ymin1(13)) ymin1(13)=n
IF(nmer.GT.ymax1(14)) ymax1(14)=nmer
IF(nmer.LT.ymin1(14)) ymin1(14)=nmer
IF(Dn.GT.ymax1(15)) ymax1(15)=Dn
IF(Dn.LT.ymin1(15)) ymin1(15)=Dn
IF(Dna.GT.ymax1(16)) ymax1(16)=Dna
IF(Dna.LT.ymin1(16)) ymin1(16)=Dna
IF(Disq.GT.ymax1(17)) ymax1(17)=Disq
IF(Disq.LT.ymin1(17)) ymin1(17)=Disq
IF(most.GT.ymax1(18)) ymax1(18)=most
IF(most.LT.ymin1(18)) ymin1(18)=most
```

ELSE

***** pri optimizaciji reg. Us

```
~~~~~ v datoteki pod1
WRITE(1,'(7E11.5)') t,n,isqm,isdm,Usmot,errumot,dthem
IF(n.GT.ymax1(1)) ymax1(1)=n
IF(n.LT.ymin1(1)) ymin1(1)=n
IF(isqm.GT.ymax1(2)) ymax1(2)=isqm
IF(isqm.LT.ymin1(2)) ymin1(2)=isqm
IF(isdm.GT.ymax1(3)) ymax1(3)=isdm
IF(isdm.LT.ymin1(3)) ymin1(3)=isdm
IF(Usmot.GT.ymax1(4)) ymax1(4)=Usmot
IF(Usmot.LT.ymin1(4)) ymin1(4)=Usmot
IF(errumot.GT.ymax1(5)) ymax1(5)=errumot
IF(errumot.LT.ymin1(5)) ymin1(5)=errumot
IF(dthem.GT.ymin1(6)) ymax1(6)=dthem
IF(dthem.LT.ymin1(6)) ymin1(6)=dthem
```

```
~~~~~ v datoteki pod2
WRITE(2,'(7E11.5)') t,n,Usmot,Dus,dthem,isdrlmin,eps1
IF(n.GT.ymax1(7)) ymax1(7)=n
IF(n.LT.ymin1(7)) ymin1(7)=n
```

```

IF(Usmot.GT.ymax1(8)) ymax1(8)=Usmot
IF(Usmot.LT.ymin1(8)) ymin1(8)=Usmot
IF(Dus.GT.ymax1(9)) ymax1(9)=Dus
IF(Dus.LT.ymin1(9)) ymin1(9)=Dus
IF(dthem.GT.ymax1(10)) ymax1(10)=dthem
IF(dthem.LT.ymin1(10)) ymin1(10)=dthem
IF(isd.GT.ymax1(11)) ymax1(11)=isd
IF(isd.LT.ymin1(11)) ymin1(11)=isd
IF(eps1.GT.ymax1(12)) ymax1(12)=eps1
IF(eps1.LT.ymin1(12)) ymin1(12)=eps1

```

```

~~~~~ v datoteki pod3
WRITE(3,'(7E11.5)') t,n,Depsm,Dkot,Disq,Disd,Dus
IF(n.GT.ymax1(13)) ymax1(13)=n
IF(n.LT.ymin1(13)) ymin1(13)=n
IF(Depsm.GT.ymax1(14)) ymax1(14)=Depsm
IF(Depsm.LT.ymin1(14)) ymin1(14)=Depsm
IF(Dkot.GT.ymax1(15)) ymax1(15)=Dkot
IF(Dkot.LT.ymin1(15)) ymin1(15)=Dkot
IF(Disq.GT.ymax1(16)) ymax1(16)=Disq
IF(Disq.LT.ymin1(16)) ymin1(16)=Disq
IF(Disd.GT.ymax1(17)) ymax1(17)=Disd
IF(Disd.LT.ymin1(17)) ymin1(17)=Disd
IF(Dus.GT.ymax1(18)) ymax1(18)=Dus
IF(Dus.LT.ymin1(18)) ymin1(18)=Dus

```

```

ENDIF
ELSE

```

```

***** pri optimizaciji tokovnih reg.
~~~~~ v datoteki pod1
WRITE(1,'(7E11.5)') t,isqm,errqm,n,isdm,errdm,eps1
IF(isqm.GT.ymax1(1)) ymax1(1)=isqm
IF(isqm.LT.ymin1(1)) ymin1(1)=isqm
IF(errqm.GT.ymax1(2)) ymax1(2)=errqm
IF(errqm.LT.ymin1(2)) ymin1(2)=errqm
IF(n.GT.ymax1(3)) ymax1(3)=n
IF(n.LT.ymin1(3)) ymin1(3)=n
IF(isdm.GT.ymax1(4)) ymax1(4)=isd
IF(isdm.LT.ymin1(4)) ymin1(4)=isd
IF(errdm.GT.ymax1(5)) ymax1(5)=errdm
IF(errdm.LT.ymin1(5)) ymin1(5)=errdm
IF(eps1.GT.ymax1(6)) ymax1(6)=eps1
IF(eps1.LT.ymin1(6)) ymin1(6)=eps1

```

```

~~~~~ v datoteki pod2
WRITE(2,'(7E11.5)') t,is11,is21,is31,us1,us2,us3
IF(is11.GT.ymax1(7)) ymax1(7)=is11
IF(is11.LT.ymin1(7)) ymin1(7)=is11
IF(is21.GT.ymax1(8)) ymax1(8)=is21
IF(is21.LT.ymin1(8)) ymin1(8)=is21
IF(is31.GT.ymax1(9)) ymax1(9)=is31
IF(is31.LT.ymin1(9)) ymin1(9)=is31
IF(us1.GT.ymax1(10)) ymax1(10)=us1
IF(us1.LT.ymin1(10)) ymin1(10)=us1
IF(us2.GT.ymax1(11)) ymax1(11)=us2

```

```

IF(us2.LT.ymin1(11)) ymin1(11)=us2
IF(us3.GT.ymax1(12)) ymax1(12)=us3
IF(us3.LT.ymin1(12)) ymin1(12)=us3

```

```

~~~~~ v datoteki pod3
WRITE(3,'(7E11.5)') t,n,Disq,Disd,isam,isum,isum
IF(n.GT.ymax1(13)) ymax1(13)=n
IF(n.LT.ymin1(13)) ymin1(13)=n
IF(Disq.GT.ymax1(14)) ymax1(14)=Disq
IF(Disq.LT.ymin1(14)) ymin1(14)=Disq
IF(Disd.GT.ymax1(15)) ymax1(15)=Disd
IF(Disd.LT.ymin1(15)) ymin1(15)=Disd
IF(isam.GT.ymax1(16)) ymax1(16)=isam
IF(isam.LT.ymin1(16)) ymin1(16)=isam
IF(isbm.GT.ymax1(17)) ymax1(17)=isbm
IF(isbm.LT.ymin1(17)) ymin1(17)=isbm
IF(isum.GT.ymax1(18)) ymax1(18)=isum
IF(isum.LT.ymin1(18)) ymin1(18)=isum

```

ENDIF

ENDIF

```

t=t+dt
i5=i5+1
i6=i6+1
i7=i7+1

```

ENDDO

```

DO i=1,ndat
CLOSE(i)

```

ENDDO

```

*****
***** napisi osi grafou

```

```

IF(s1.EQ.1.)THEN
IF(s3.EQ.0.)THEN

```

***** normalni tek

```

tx2(1)='nm (/min) '
tx2(2)='Errnm (/min) '
tx2(3)='eps1(rad) '
tx2(4)='ISQm(A) '
tx2(5)='ISDm(A) '
tx2(6)='Usm(U) '
tx2(7)='IS1m(A) '
tx2(8)='IS2m(A) '
tx2(9)='Us1(U) '
tx2(10)='Us2(U) '
tx2(11)='US12(U) '
tx2(12)='Usum(U) '
tx2(13)='nm (/min) '
tx2(14)='nmer (/min) '
tx2(15)='(nmer-nm)/nm (%) '
tx2(16)='nmer-nm (/min) '
tx2(17)='isq-isqm (A) '

```

```
tx2(18)='Most (Nm)'  
ELSE
```

```
***** pri optimizaciji Us
```

```
tx2(1)='nm (/min)'  
tx2(2)='ISQm (A)'  
tx2(3)='ISDm (A)'  
tx2(4)='Usm (V)'  
tx2(5)='ErrUs (V)'  
tx2(6)='th1m-thm (rad)'  
tx2(7)='nm (/min)'  
tx2(8)='Usm (V)'  
tx2(9)='Dus (V)'  
tx2(10)='dthem (rad)'  
tx2(11)='isdrlmin (A)'  
tx2(12)='eps1 (rad)'  
tx2(13)='nm (/min)'  
tx2(14)='eps-eps1 (/rad)'  
tx2(15)='alfa-"eps2" (rad)'  
tx2(16)='Isq-Isqm (A)'  
tx2(17)='Isd-Isdm (A)'  
tx2(18)='Us-Usm (V)'
```

```
ENDIF
```

```
ELSE
```

```
***** pri optimizaciji Isq in Isd
```

```
tx2(1)='ISQm(A)'  
tx2(2)='ErrISQm(A)'  
tx2(3)='nm(/min)'  
tx2(4)='ISDm(A)'  
tx2(5)='ErrISDm(A)'  
tx2(6)='EPSm(rad)'  
tx2(7)='IS1m(A)'  
tx2(8)='IS2m(A)'  
tx2(9)='IS3m(A)'  
tx2(10)='US1(V)'  
tx2(11)='US2(V)'  
tx2(12)='US3(V)'  
tx2(13)='nm (/min)'  
tx2(14)='Isq-Isqm (A)'  
tx2(15)='Isd-Isdm (A)'  
tx2(16)='Isam (A)'  
tx2(17)='Isbm (A)'  
tx2(18)='vsota tokov'
```

```
ENDIF
```

```
***** napis abcis
```

```
DO i=1,18  
tx1(i)="t"  
ENDDO
```

```
***** odprtje datotek pod-
```

```
i4=0  
DO i=1,ndat  
STE1=2**13+2**12+i*2**8
```



```

IIME='pod'//STE
OPEN(i,ERR=30,ACCESS='SEQUENTIAL',FILE=IIME,STATUS='NEW')
GOTO 40
30 OPEN(i,ACCESS='SEQUENTIAL',FILE=IIME,STATUS='OLD')
40 IF(i.NE.ndat)THEN
    ng=6
    i2k=2
    i3k=3
ELSE IF(ost.EQ.0)THEN
    ng=6
    ELSE
        ng=ost
ENDIF
C ***** opis dimenzij grafov
C ***** ng
WRITE(i,'(I2)')ng
v1=0.
vr=.45
C
IF(i.EQ.ndat)THEN
    IF(ost.GT.3)THEN
        i2k=2
        ost=ost-3
    ELSEIF(ost.EQ.0)THEN
        i2k=2
    ELSE
        i2k=1
    ENDIF
ENDIF
C
DO i2=1,i2k
    vu=1.
    vd=.68
C
    IF(i.EQ.ndat)THEN
        IF(i2.EQ.i2k.AND.ost.NE.0)THEN
            i3k=ost
        ELSE
            i3k=3
        ENDIF
    ENDIF
C
    DO i3=1,i3k
        i4=i4+1
C
        ***** dolocitev max. vr. osi grafov
        i10=INDEX(tx2(i4),'Err')
C
        IF(i10.NE.0)THEN
            ~~~~~ max in min vr. diagramov napak
            IF(ABS(ymax1(i4)).GT.ABS(ymin1(i4)))THEN
                ymax=ABS(ymax1(i4))*0.05
                ymin=-ymax
            ELSEIF(ymax1(i4).EQ.0..AND.ymin1(i4).EQ.0.)THEN
                ymax=1.
                ymin=-ymax

```

```

ELSE
  ymax=ABS(ymin1(i4))*0.05
  ymin=-ymax
ENDIF

```

```

ELSEIF(ymax1(i4).GT.0.)THEN
  IF(ymin1(i4).GT.-ymax1(i4)*0.3)THEN
    ymax=ymax1(i4)*1.3
    ymin=-ymax1(i4)*1.3
  ELSEIF(-.3*ymin1(i4).GT.ymax1(i4))THEN
    ymax=-ymin1(i4)*0.3
    ymin=ymin1(i4)*1.3
  ELSE
    ymax=ymax1(i4)*1.3
    ymin=ymin1(i4)*1.3
  ENDIF
ELSEIF(ymax1(i4).EQ.0..AND.ymin1(i4).EQ.0.)THEN
  ymax=1.
  ymin=-1.
ELSE
  ymax=-ymin1(i4)*.3
  ymin=ymin1(i4)*1.3
ENDIF

```

```

nx=nx1(i4)
npx=np1(i4)
ny=ny1(i4)
npy=np2(i4)
text1=tx1(i4)
text2=tx2(i4)

```

```

***** ymin,ymax,vl,vd,vu,npx,nx,npy,ny,text1,text2
WRITE(i,'(2E10.3)') ymin,ymax
WRITE(i,'(4E10.3)') vl,vr,vd,vu
WRITE(i,'(4I2)') nx,npx,ny,npy
WRITE(i,'(A20)') text2
WRITE(i,'(A5)') text1
vu=vu-.34
vd=vd-.34

```

```

ENDDO
vl=vl+.55
vr=vr+.55

```

```

ENDDO
ENDDO

```

```

***** prepis rezultatov iz ggg- v pod-

```

```

DO i=1,ndat
  ii=i+20
  STE1=2**13+2**12+i*2**8
  IME='ggg'//STE
  OPEN(ii,ACCESS='SEQUENTIAL',FILE=IME,STATUS='OLD')

```

```

101 READ(ii,'(7E11.5)',END=102) g1,g2,g3,g4,g5,g6,g7
WRITE(i,'(7E11.5)') g1,g2,g3,g4,g5,g6,g7
GOTO 101

```

```

102     CLOSE(ii,STATUS='DELETE')
        CLOSE(i)
        ENDDO

C
C *****
C ***** izpisi na ekranu
C ***** prekoracitve
PRINT*, ' '
PRINT*, ' '
IF(s4.EQ.1.OR.s5.EQ.1.) WRITE(6,('NASICENJE PRETVORNIKA :'))
IF(s5.EQ.1) WRITE(6,('ismd gre cez ismax ",F9.4," %")) isdmmax

IF(s4.EQ.1) WRITE(6,('pri am=",F9.4," je isqm=",F9.4," % vecji'))
#) am1, isqmmax

PRINT*, ' '
IF(s6.EQ.1) WRITE(6,('PREKORACITEV KOTA THE1M ! '))

C
C ***** vsebine datotek
PRINT*, ' '
PRINT*, ' '
WRITE(6,('TR5,NL,"Rezultati so v datotekah :'))
PRINT*, ' '

IF(s1.EQ.1.)THEN
  IF(s3.EQ.0.)THEN

C
C ~~~~~ normalni tek
WRITE(6,('pod1 : nm(/min), ErrN(/min), epsm(r), Isqm(A), Isdm
#(A), Usm(U)'))
WRITE(6,('pod2 : Is1m(A), Is1(A), Is2m(A), Us1(U), Us12(U), U
#sum(U)'))
WRITE(6,('pod3 : nm(min), n(min), Dn(/min), Dna(/min), Disqm,
# most(Nm)'))
ELSE

C
C ~~~~~ pri optimizaciji Us
WRITE(6,('pod1 : nm(/min), Isqm(A), Isdm(A), Usm(U), ErrUsm(U
#), dthem(rad)'))
WRITE(6,('pod2 : nm(/min), Dus(v), dthem(rad), isdr1min(A), e
#psm(rad)'))
WRITE(6,('pod3 : nm(min), Deps(rad), Dkot(rad), Disq(A), Disd
#(A), Dus(U)'))
ENDIF
ELSE

C
C ~~~~~ pri optimizaciji Isq in Isd
WRITE(6,('pod1 : Isqm(A), ErrIsqm(a), nm(/min), Isdm(A), ErrIsd
#m(A), epsm(rad)'))
WRITE(6,('pod2 : Is1m(A), Is2m(A), Is3m(A), Us1(U), Us2(U), Us3
#(U)'))
WRITE(6,('pod3 : nm(rad), Disq(A), Disd(A), isam(A), isbm(A), i
#sum(A)'))
ENDIF

```

```
PRINT*, ' '  
PRINT*, ' '  
PRINT*, ' '
```

```
C  
1000 END
```

PROGRAM ZAVIRANJE

```

C
INTEGER*2 ng,nx,npx,ny,npj,nx1(2),npx1(2),ny1(2),npj1(2)
INTEGER*2 i,i5,izp
REAL*4 y,ymax,ymin,vl,vr,vd,vu,b,kor,p,la,mc,vez
REAL*4 t,dt,om1,om,omp,om0,ra,rz,k,kt,ke,jm,jb,tfin,n0,n,a
CHARACTER*12 text2
CHARACTER*6 text1
CHARACTER a1*8,a2*11,a6*8

```

```

C
EQUIVALENCE (STE,STE1)

```

```

C
DATA pi,omp,om,om1,i5/3.14159,3*0.,0/
DATA ny,npj,nx,npx/4,0,6,2/

```

```

C
WRITE(6,'(//////////,TR5,"ZAVIRANJA SINHRONSKEGA MO
#TORJA ")')

```

```

C
***** sprememba parametrov

```

```

C
OPEN(10,ERR=1,ACCESS='SEQUENTIAL',FILE='par',STATUS='NEW')
GOTO 2
1 OPEN(10,ACCESS='SEQUENTIAL',FILE='par',STATUS='OLD')
2 CONTINUE

```

```

C
READ(10,'(7E11.5)') n0,la,ra,rz,kt,p,kor
READ(10,'(6E11.5,I11)') jm,jb,b,mc,dt,tfin,izp
READ(10,'(A11)') a2
CLOSE(10)

```

```

C
*****

```

```

WRITE(6,'(///,TR10,NN,"SPREMEMBA PARAMETROV (Y/*...)")')
READ(5,'(A8)') a1
IF(a1.EQ.'y'.OR.a1.EQ.'Y')THEN

```

```

C
WRITE(6,'(/,TR10,"PARAMETRI :")')
PRINT*,' '
PRINT*,' N0(/min), Ra(om), Rz(om), kt(Nm/A), Jb(kgm2)*1E-4'
PRINT*,' '
PRINT*,' Jm(kgm2)*1E-04, b(Nms/rad), dt(sec), tfin(sec), izp
#(/dt)'
PRINT*,' '
PRINT*,' kor( ), p( pari), La(Us/A), Mc(Nm), vez(Z ali
#T)'

```

```

C
***** branja parametrov

```

```

C
PRINT*,' '
100 WRITE(6,'($,"parameter ")')
READ(5,'(A8,$)') a6
IF(a6.EQ.'n0'.OR.a6.EQ.'N0')THEN
WRITE(6,'($,E11.5)')n0
WRITE(6,'($," -> ")')
READ*,n0
GOTO 100

```

```
ELSEIF(a6.EQ.'ra'.OR.a6.EQ.'Ra')THEN
  WRITE(6,'($,E11.5)')ra
  WRITE(6,'($," -> ")')
  READ*,ra
  GOTO 100
ELSEIF(a6.EQ.'rz'.OR.a6.EQ.'Rz')THEN
  WRITE(6,'($,E11.5)')rz
  WRITE(6,'($," -> ")')
  READ*,rz
  GOTO 100
ELSEIF(a6.EQ.'kt'.OR.a6.EQ.'Kt')THEN
  WRITE(6,'($,E11.5)')kt
  WRITE(6,'($," -> ")')
  READ*,kt
  GOTO 100
ELSEIF(a6.EQ.'ke'.OR.a6.EQ.'Ke')THEN
  WRITE(6,'($,E11.5)')ke
  WRITE(6,'($," -> ")')
  READ*,ke
  GOTO 100
ELSEIF(a6.EQ.'jm'.OR.a6.EQ.'Jm')THEN
  WRITE(6,'($,E11.5)')jm
  WRITE(6,'($," -> ")')
  READ*,jm
  GOTO 100
ELSEIF(a6.EQ.'jb'.OR.a6.EQ.'Jb')THEN
  WRITE(6,'($,E11.5)')jb
  WRITE(6,'($," -> ")')
  READ*,jb
  GOTO 100
ELSEIF(a6.EQ.'tfin'.OR.a6.EQ.'Tfin')THEN
  WRITE(6,'($,E11.5)')tfin
  WRITE(6,'($," -> ")')
  READ*,tfin
  GOTO 100
ELSEIF(a6.EQ.'dt'.OR.a6.EQ.'Dt')THEN
  WRITE(6,'($,E11.5)')dt
  WRITE(6,'($," -> ")')
  READ*,dt
  GOTO 100
ELSEIF(a6.EQ.'izp'.OR.a6.EQ.'Izp')THEN
  WRITE(6,'($,11)')izp
  WRITE(6,'($," -> ")')
  READ*,izp
  GOTO 100
ELSEIF(a6.EQ.'b'.OR.a6.EQ.'B')THEN
  WRITE(6,'($,E11.5)')b
  WRITE(6,'($," -> ")')
  READ*,b
  GOTO 100
ELSEIF(a6.EQ.'kor'.OR.a6.EQ.'Kor')THEN
  WRITE(6,'($,E11.5)')kor
  WRITE(6,'($," -> ")')
  READ*,kor
  GOTO 100
ELSEIF(a6.EQ.'p'.OR.a6.EQ.'P')THEN
```

```
WRITE(6, '( $, E11.5)') p
WRITE(6, '( $, " -> ")')
READ*, p
GOTO 100
ELSEIF(a6.EQ. 'la'.OR.a6.EQ. 'La') THEN
WRITE(6, '( $, E11.5)') la
WRITE(6, '( $, " -> ")')
READ*, la
GOTO 100
ELSEIF(a6.EQ. 'mc'.OR.a6.EQ. 'Mc') THEN
WRITE(6, '( $, E11.5)') mc
WRITE(6, '( $, " -> ")')
READ*, mc
GOTO 100
```

```
ENDIF
IF(a6.EQ. 'vez'.OR.a6.EQ. 'Uez') THEN
WRITE(6, '( $, A11)') a2
WRITE(6, '( $, " -> ")')
READ(5, '( A11)') a2
IF(a2.EQ. 'T'.OR.a2.EQ. 't'.OR.a2.EQ. 'trikot') THEN
vez=2./3.
GOTO 100
ELSEIF(a2.EQ. 'Z'.OR.a2.EQ. 'z'.OR.a2.EQ. 'zvezda') THEN
vez=0.5
GOTO 100
ELSE
PRINT*, 'katera vezava ? (T ali Z)'
READ(5, '( A11)') a2
GOTO 100
ENDIF
ENDIF
```

```
ENDIF
```

```
OPEN(10,ERR=111,ACCESS='SEQUENTIAL',FILE='par',STATUS='NEW')
GOTO 112
111 OPEN(10,ACCESS='SEQUENTIAL',FILE='par',STATUS='OLD')
112 CONTINUE
WRITE(10, '(7E11.5)') n0, la, ra, rz, kt, p, kor
WRITE(10, '(6E11.5, I11)') jm, jb, b, mc, dt, tfin, izp
WRITE(10, '(A11)') a2
CLOSE(10)
```

```
*****
*****
```

```
***** graficni del
```

```
WRITE(6, '(//, "Rezultati so v datoteki pod : N(/min), a(rad/s2)")')
OPEN(1,ERR=10,ACCESS='SEQUENTIAL',FILE='pod',STATUS='NEW')
GOTO 20
10 OPEN(1,ACCESS='SEQUENTIAL',FILE='pod',STATUS='OLD')
```

```
*****
om0=2.*pi*n0/60.
om=om0
```

```

C      *****
C
20     ng=2
       WRITE(1, '(I2)') ng
C
       vl=0.
       vr=1.
       text1='t'
C
       vu=1.
       vd=.55
       IF(n0.GE.0.)THEN
           ymax=1.2*n0
           ymin=0.
       ELSE
           ymax=0.
           ymin=1.2*n0
       ENDIF
       text2='N(/min)'
       WRITE(1, '(2E10.3)') ymin,ymax
       WRITE(1, '(4E10.3)') vl,vr,vd,vu
       WRITE(1, '(4I2)') nx,npx,ny,npj
       WRITE(1, '(A20)') text2
       WRITE(1, '(A5)') text1
C
       vu=.45
       vd=0.
       WRITE(6, '(/, $, "amax =")')
       IF(n0.LT.0.)THEN
           k=kor**2*kt**4*om0**2/(3.*((ra*vez+rz)**2+(p*vez*la*om0)**2))
           y=-(k+mb+mc+b*om0)/(jm+jb)*1.E+04
           ymax=1.2*y
           ymin=0.
           PRINT*,y
       ELSE
           k=kor**2*kt**4*om0**2/(3.*((ra*vez+rz)**2+(p*vez*la*om0)**2))
           y=-(k+mb+mc+b*om0)/(jm+jb)*1.E+04
           ymin=1.2*y
           ymax=0.
           PRINT*,y
       ENDIF
       text2='a(rad/s2)'
       WRITE(1, '(2E10.3)') ymin,ymax
       WRITE(1, '(4E10.3)') vl,vr,vd,vu
       WRITE(1, '(4I2)') nx,npx,ny,npj
       WRITE(1, '(A20)') text2
       WRITE(1, '(A5)') text1
C
       ***** glavni del programa
       om0=2.*pi*n0/60.
       om=om0
C
       DO WHILE(t.LE.tfin)
           t=t+dt
           i5=i5+1
C

```



```
k=kor**2*kt**4*om**2/(3.*((ra*vez+rz)**2+(p*vez*la*om)**2))
omp=-(k+mb+mc+b*om)/(jm+jb)*1.E+04
a=omp
om1=om1+omp*dt
IF (ABS(om1).GE.ABS(om0)) THEN
  om=0.
ELSE
  om=om0+om1
ENDIF
n=om*60./(2*pi)
IF (MOD(i5,izp).EQ.0.) THEN
  WRITE(1,'(3E11.5)') t,n,a
ENDIF
ENDDO
CLOSE(1)
END
```

1000