

$$Q_{g\text{-max}} = \Gamma_{11} \text{bus nx1} \rightarrow \boxed{\begin{array}{c} \text{chq} \\ - \\ \text{lim} \end{array}} \rightarrow f : \text{chq-lim}$$

$$Q_{g\text{-min}} = \Gamma_{12} \text{bus nx1} \rightarrow \boxed{\begin{array}{c} \text{chq} \\ - \\ \text{lim} \end{array}} \rightarrow f : \text{chq-lim}$$

$Q_{g\text{-min}} = f = \begin{cases} 1 & \text{if } Q_{g\text{-min}} \leq \text{chq-lim} \\ 0 & \text{otherwise} \end{cases}$

PQ=PV \rightarrow $Q_{g\text{-min}} = 2 \text{ if bus-type} = 2$

$Q_{g\text{-min}} = \begin{cases} Q_{g\text{-min}} & \text{if bus-type} = 2 \\ Q_{g\text{-max}} & \text{otherwise} \end{cases}$

bus-type = 2 \rightarrow PV \rightarrow $Q_{g\text{-min}} = 2$ if bus-type = 2

$Q_g[\text{bus-type} = 2] > q_g^{\text{max}}[\text{bus-type} = 2] \Rightarrow q_g^{\text{max-index}} = [1 \dots 1]$

$Q_g[\text{bus-type} = 2] < q_g^{\text{min}} \Rightarrow q_g^{\text{min-index}} = [-1 \dots -1]$

calc P
g-bno

bus-type(1) = 3
volt-ref
ang-vel
scalar

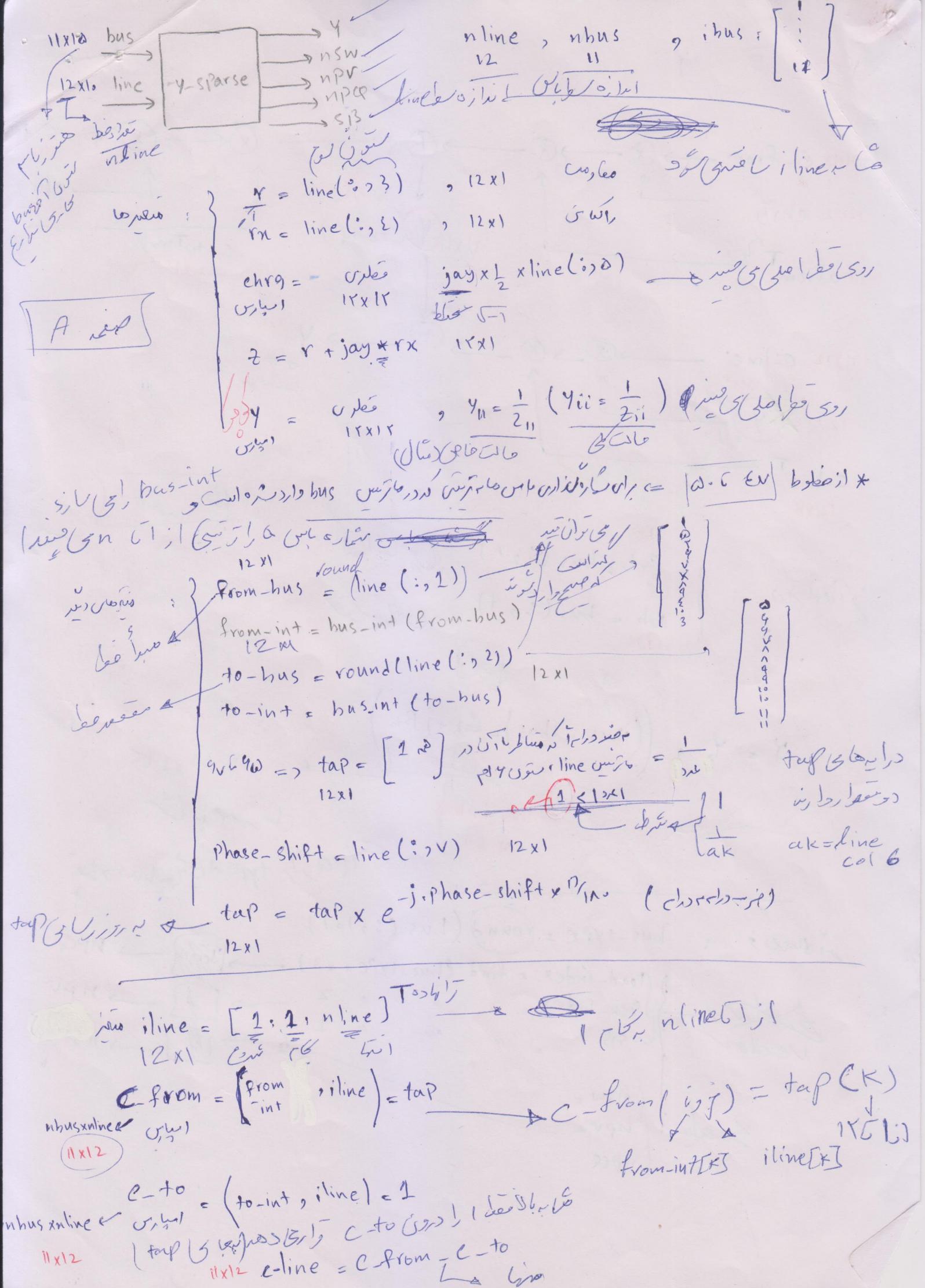
$$Q_g(\text{index} \text{ dip}) = 0 \quad (1, b, 1 \text{ ref } q_g^{\text{max-index}}) \quad \text{if max} \Rightarrow 0$$

$$Q_d(\text{ref}) = Q_d - q_g^{\text{max}}(\text{dip index})$$

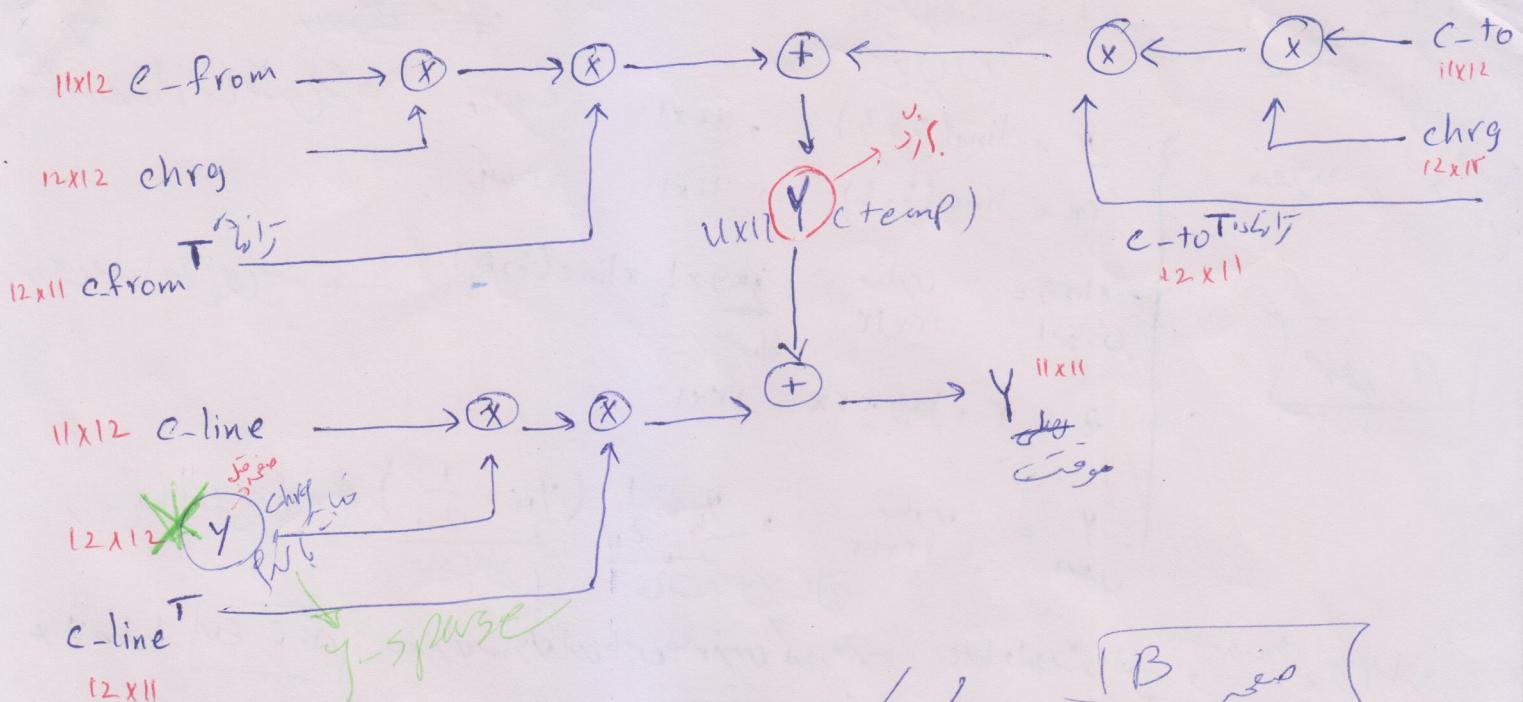
$$Q_g(\text{index} \text{ dip}) = 0 \quad (1, b, 1 \text{ ref } q_g^{\text{min-index}}) \quad \text{if min} \Rightarrow 0$$

$$Q_d(\text{ref}) = Q_d - q_g^{\text{min}}(\text{ref})$$

$f = \text{lim-flag}$



CktB :



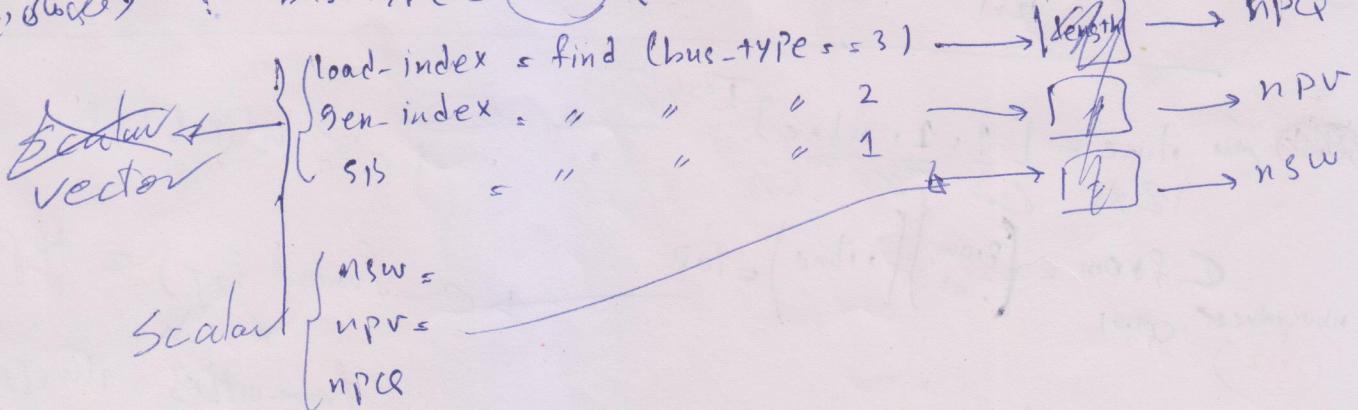
gridbus :

$$G_b = \text{bus}(:, n) \quad G_{bus}$$

$$B_b = \text{bus}(:, q) \quad B_{bus}$$

$$Y = Y + \left(\begin{matrix} (ibus, ibus) = G_b + B_b \\ \text{ibus} \times \text{ibus} \end{matrix} \right) \quad n \in \{1, K\}$$

bus-type : bus-type = round(bus(:, 10))



form-sac gl root ← ques date:

(1)

$$v_{b2} \rightarrow 11 \times 11 \quad (S+S1) \rightarrow 11 \times 11$$

$$v_{b1} S1 \rightarrow " \rightarrow (S+S1) \rightarrow 11 \times 11$$

$$N_{mag} \rightarrow 11 \times 11 \quad (1) \rightarrow 11 \times 11$$

$$(S+S1)/N_{mag} \xrightarrow{?} 11 \times 11$$

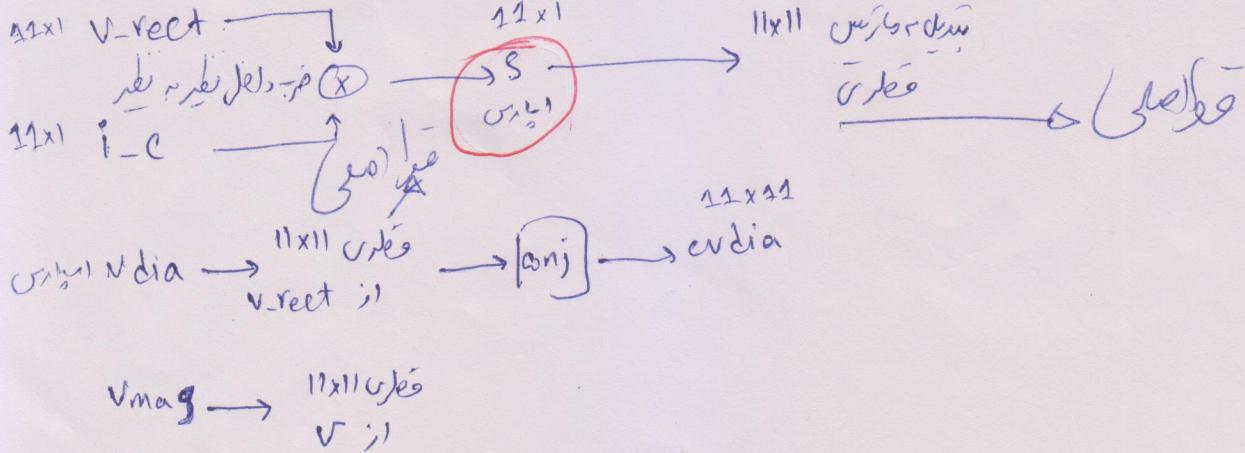
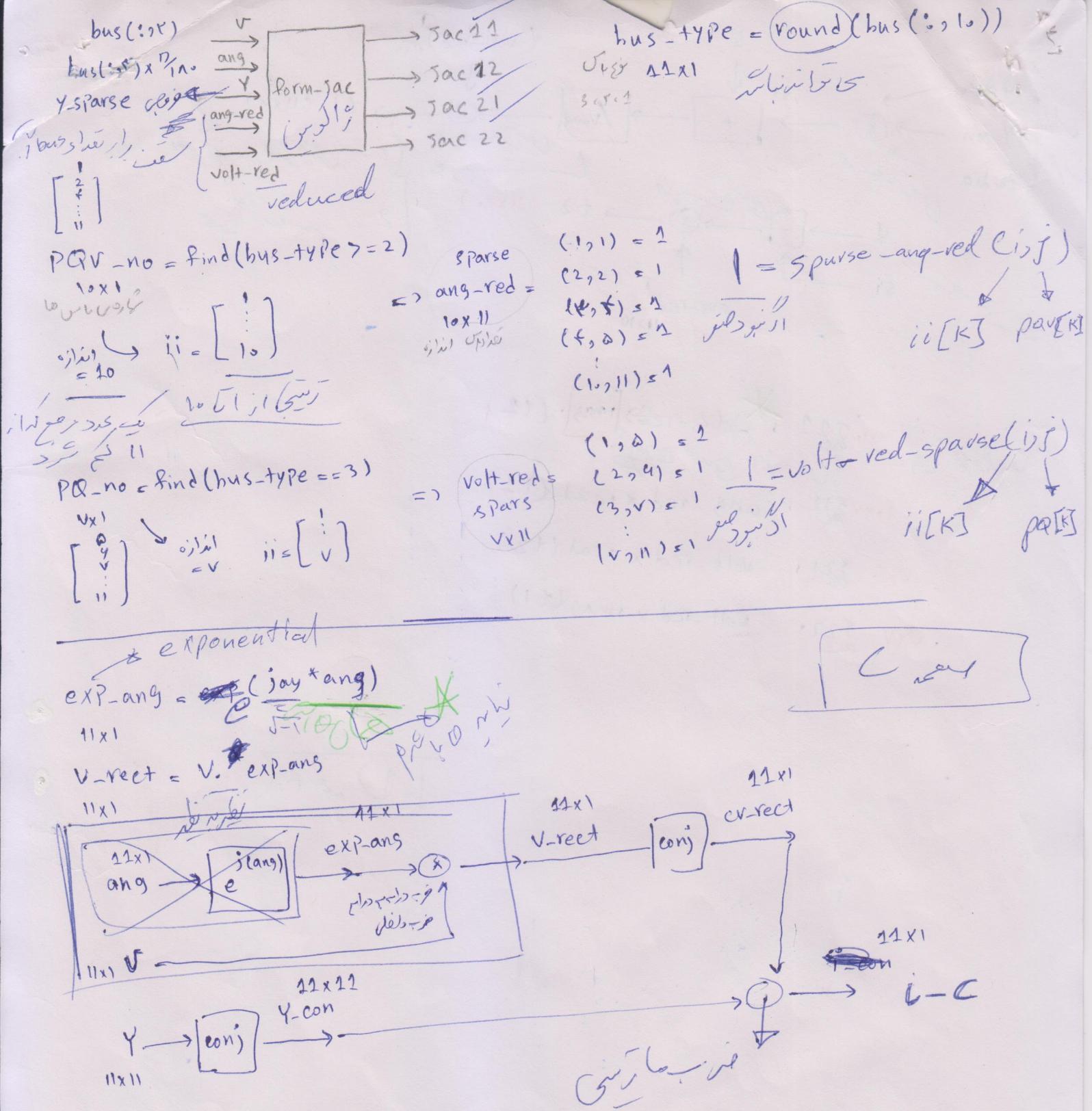
تم در این سهون ترم از روش
میگویند $(S+S1) / N_{mag}$ را

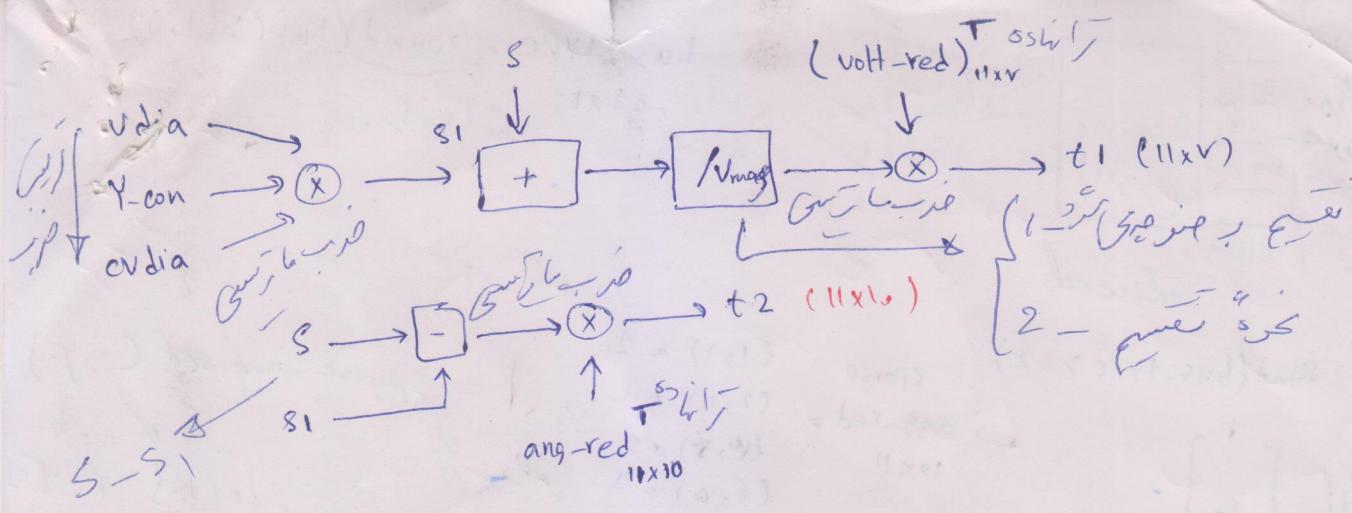
میگویند $(11 \times 11) / N_{mag}$ را

$$v_{b2} (1) = \frac{[S+S1]_{(1)}}{N_{mag} (1)} \quad 11 \times 11$$

میگویند v_{b2} را

Arman - میگویند v_{b2} را





~~lock~~ \Rightarrow $\begin{cases} 1 \times 10^6 \bar{s}_{21} : \text{ang-red} * \text{imas}(t_2) \\ 1 \times 10^6 \bar{s}_{12} : \text{ang-red} * \text{real}(t_1) \end{cases}$

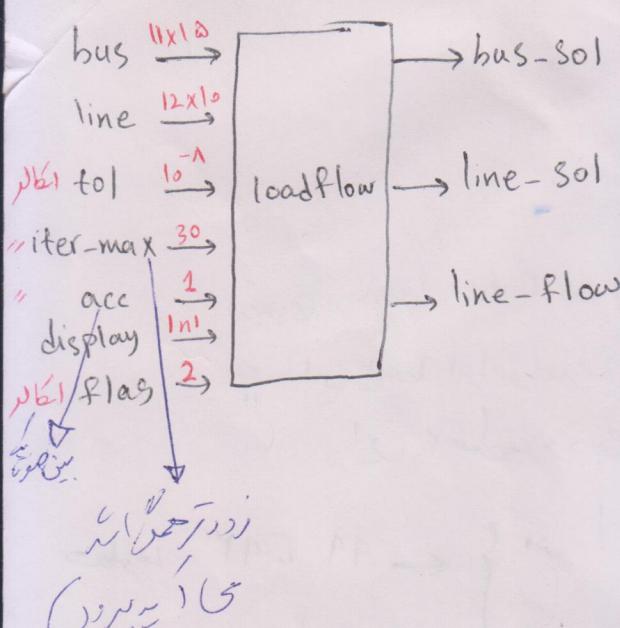
$V \times V$ $\begin{cases} \bar{s}_{21} : \text{volt-red} * \text{real}(t_2) \\ \bar{s}_{12} : \text{volt-red} * \text{imas}(t_1) \end{cases}$

$V \times V$ $\begin{cases} \bar{s}_{21} : \text{volt-red} * \text{real}(t_2) \\ \bar{s}_{12} : \text{volt-red} * \text{imas}(t_1) \end{cases}$

D_{real}

2(1)

حالة حل مجهولة بعين توان سعى قدر = loadflow

تواتر دارک افراطی ایجاد شود ← Y-Sparse , cale , form-Tac , chq1-lim 

bus: $b_{\text{لین}} \in \mathbb{C}^{11 \times 10}$
line: $b_{\text{لاین}} \in \mathbb{C}^{12 \times 10}$
tol: tolerance, $\mu_{\text{لاین}}$ (لاین), default = 10^{-1}
iter-max: max number of iteration, $\mu_{\text{لاین}}$
acc: acceleration factor, $\mu_{\text{لاین}}$ (لاین), default = 1
display = $\begin{cases} 1 & \rightarrow \text{نمایش} \\ 1 \text{ int } & \rightarrow \dots \end{cases}$ (default)
flag = $\begin{cases} 1 \text{ or } 2 & \rightarrow \text{نمایش خوب حل} \\ 1 \text{ int } & \rightarrow \dots \end{cases}$ (default = 2)

nline: $b_{\text{لاین}} \in \mathbb{C}^{12}$ و nlc: $b_{\text{لاین}} \in \mathbb{C}^{10}$

nbus: $b_{\text{لاین}} \in \mathbb{C}^{11}$ و nc1: $b_{\text{لاین}} \in \mathbb{C}^{10}$

میتوانند با هم مترادف باشند
line, bus میتوانند با هم مترادف باشند

* خطوط ایجاد شده در مورد آنها را در محدوده کاربر راهی کنند. از درون آنها 2 وارد شوند flag (باید 1 یا 2 باشد) و آنها را در محدوده کاربر راهی کنند. همین اگر در درون آنها display, acc, iter-max, tol (باید 1 یا 2 باشد) وارد شوند میتوانند با هم مترادف باشند. در حال حاضر دو این مقدار ایجاد شده اند که اینها انتخاب نمیکنند.

* خطوط ایجاد شده در محدوده کاربر راهی کنند. باعزم اینها من توان کاربر را بخوبی کنند. میتوانند وارد شوند و محدوده میتوانند با هم مترادف باشند. در حال حاضر دو این مقدار ایجاد شده اند که اینها انتخاب نمیکنند.

اعمالی نمود.

bus(:, 11) \rightarrow q1_gen_max : $b_{\text{لاین}} \in \mathbb{C}^{n_{\text{لاین}} \times 1}$ (PU)

bus(:, 12) \rightarrow q1_gen_min : $\sim \sim \sim \sim$ (PU)

bus(:, 13) \rightarrow v_rate : $\sim \sim \sim \sim$ (PU)

bus(:, 14) \rightarrow v_max : $\sim \sim \sim \sim$ (PU)

bus(:, 15) \rightarrow v_min : $\sim \sim \sim \sim$ (PU)

میتوانند \rightarrow (comment)

11 neol < 10

$$\text{if } \text{ncol} < 12 \rightarrow \begin{cases} \text{bus}(\cdot, 11) = 9999 \\ \text{bus}(\cdot, 12) = -9999 \end{cases}$$

// ncol < 13 → bus(:, 13) = 1

$$\text{bus}(:, 18) = \text{b}\omega$$

$$\text{bus} (\therefore, \text{id}) = \text{id}$$

$$U_{X1} \text{ Volt-min} = \text{bus } (v, 1\omega)$$

$$11 \times 1 \text{ Volt_max} = \text{bus}(\therefore, 1E)$$

$$\text{Volt-min}(k) = \omega / \Delta \Leftrightarrow \text{Volt-min}(\vec{k}) = \omega$$

$$\text{volt_max}(k) = 1, \omega \Leftrightarrow \text{volt_max}(k) = 0$$

$\text{bus}(k,11) = 9999 \leftarrow \text{bus}(k,11) = 0$ { $\leftarrow \log C_{100}$

bus(k,11) ← -9999 bus(k,12) ← 0

$\text{bus}(k, 13) \leq 1 \leftarrow \text{bus}(k, 13) \leq 0$

$$\left. \begin{array}{l} \text{if } \text{deg}(b) \leq \text{res } \mu \\ \text{if } \text{deg}(b) > \text{res } \mu \end{array} \right\} \begin{array}{l} \text{tap-it} = 0 \\ \text{tap-it-max} = 10 \\ \text{no-taps} = 0 \end{array}$$

$$\left\{ \begin{array}{l} \text{if } nlc < 10 \Rightarrow \\ \text{no_taps} = 1 \end{array} \right\} \text{line} \left(\therefore, (1, 9, 10) \right) = 0 \Leftrightarrow \text{line}$$

$$mm - chk = 1$$

```
while ( tap-it < tap-it-max , mn-chk = 1 )
```

$$\text{tap_it} = \text{tap_it} + 1$$

$[Y, nsw, npr, npqr, SB] = \text{Y-Sparse}(\text{bus}, \text{line})$ ← Y-Sparse \rightarrow Cisalj;

end → we gel>> bl

* میں از خلائق اے ی-Sparse میں رکھاں گے

1-②

11 x1 bus_no=bus(:,1) ← 100% corretto

١١) $V = \text{bus}(:, 2)$ ← نحوه (مقدار) (World)

$$11 \times 1 \quad \text{ans} = \text{bus}(3) \times \frac{n}{10} \leftarrow \text{زاف اول}$$

$$11 \times 1 \quad Pg = bus(^\circ, \xi) \quad \leftrightarrow \quad \text{بيان الموجات}$$

$$11 \times 1 \quad Q_{ij} = \text{bus}(\cdot, d) \leftarrow \text{out}_j \text{out}_i$$

١١ x1 P_ل = bus(:, 4) ← ماتریس اسکالر

$$11 \times 1 \quad Ql = bus(0, v) \quad \leftarrow \quad " \text{new!} \right)$$

$$11 \times 1 \quad P_{ab} = \text{bus}(:,n) \quad \leftarrow \quad \text{row} \quad \text{row}$$

$$\text{Ex: } Bb = \text{bus} (0,9) \leftarrow \text{باص} \quad \text{bus} (0,9) \leftarrow \text{باص}$$

11x) bus_type = round (bus (:,1:)) ← orig (↑ ↑ 1 ≤ 2 ≤ 3)

١١٨) $\text{Pg}_{\text{max}} \leq \text{bus} (\therefore, 11)$ ← \rightarrow بيان أسلوب حساب حداً علويًّا لـ Pg_{max}

$$11x_1 \text{ gg_min} = \text{bus}(1:9,1) \quad \leftarrow \quad " \quad " \quad " \quad " \quad \text{cut}$$

~~def~~ swing_index = find(buy_type == 1) \Leftarrow swing_index = generates

$$PQV_no = \text{find}(\text{bus_type} >= 2) \quad : PQV_no = \text{جدارهای که در مسیر ماروی را از قدر}$$

Ques 12: PQ-no = find (bus_type == 3) PQ-no = xl " " "

39) b) gen-index = " (" == 2) gen-index = 1st " = " "

$$\text{if } x_i \text{ s.w.-bno}(k) = \left\{ \begin{array}{ll} 0 & k = \text{swing-index} \\ 1 & \text{otherwise} \end{array} \right.$$

$$g - \text{knock}(k) = \begin{cases} 0 & k = \text{gen_index} \\ 1 & \text{otherwise} \end{cases}$$

$$PQR\text{-no} = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n \end{bmatrix}$$

$$PA - n\sigma = \begin{bmatrix} 0 \\ 4 \\ v \\ \vdots \\ 11 \end{bmatrix}$$

(م) فصلیوں کا لگنے والے مادے: پیکر volt-red, angr-red لگنے والے مادے *

$$\text{length}(PQR\text{-no}) = \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\text{V}\mu\ell \text{ il} = \text{length}(PQ\text{-no}) \Rightarrow \text{volt-red } (i,j) = 1 \cdot \left\{ \begin{array}{l} i = "i" \\ j = PQ\text{-no}(k) \\ \checkmark C \text{ 1;1;k} \end{array} \right.$$

Uguljst
vxi
 $\begin{bmatrix} 1 \\ 2 \\ \vdots \\ v \end{bmatrix}$

(ج) $\text{Jac} = \text{form-Jac} (\mathbf{v}, \text{ans}, \mathbf{Y}, \text{ans-red}, \text{volt-red})$, $\text{Jac} (IV \times IV)$

$$\begin{aligned} \text{1o x 1} \quad \text{red-delP} &= \text{ang-red} * \text{delP} ; \\ \text{v x 1} \quad \text{red-delQ} &= \text{volt-red} * \text{delQ} \end{aligned} \rightarrow \left[\begin{array}{c} \text{red-delP} \\ \text{red-delQ} \end{array} \right]_{\text{1o x 1}} \left[\begin{array}{c} \text{ang-red} \\ \text{volt-red} \end{array} \right]_{\text{v x 1}} \left[\begin{array}{c} \text{delP} \\ \text{delQ} \end{array} \right]_{\text{1v x 1}}$$

$$\text{temp} = \text{inv}_{\text{usg}}(\text{Jac}) * [\text{red-delP}; \text{red-delQ}]$$

length(PQR-n0) → 61 less

$$\text{def} \Delta \text{ang} = \text{ang_red}^T * \text{temp}(10, 10) \rightarrow \text{temp} \text{ out}_i$$

$$\text{del } V = \frac{\text{volt-red}^T * \text{temp}}{11 \times 1} \cdot \frac{10+1}{\text{length}} : \frac{10+V}{(PQR-\text{no})} : \frac{1}{11 \times 1} \cdot \frac{\text{length}}{(PQR-\text{no})}$$

$$V = V + acc^* \cdot delV$$

$11 \times 1 \quad 11 \times 1 \quad 26 \times 1 \quad 11 \times 1$

$$\left\{ \begin{array}{l} V = \max(V, \text{volt_min}) \\ V = \min(V, \text{volt_max}) \end{array} \right.$$

$$V_{(k)} = \begin{cases} V_{(k)} & \text{if } V_{(k)} > \text{Volt_min}(k) \\ \text{Volt_min}(k) & \text{if } V_{(k)} < \text{Volt_min}(k) \end{cases}$$

$$v_{(n)} = \begin{cases} v(n) & \text{if } n < v_{\text{left-max}}(n) \\ v_{\text{left-max}}(n) & \text{if } n > " \end{cases}$$

$$\text{if } x_1 \text{ ang} = \text{ang} + \text{acc} * \text{delAng}$$

`calc [P, CP, ...] ; [delP, delCP, ...] = calc (V, ans, ...,)`

رسانی، gen-index = find (bus-type == 2)؛ (این) که سه قوه‌ی PV

$$Q_g(\text{gen-index}) = \cancel{Q}(\text{gen-index}) + Q_l(\text{gen-index})$$

~~لماس هزارخ $P \wedge P$ نیز صورت نمیگیرد.~~ (فقط لاس از نوع $P \vee P$)

$$\lim_{t \rightarrow \infty} \text{flag} = \text{ch}^{\text{PI}} \lim (\text{qg_max}, \text{qg_min});$$

$\text{if } \text{lim_flag} = 1 \Rightarrow \text{Cwd } \text{M}(V), Qg$ plus

~~الخط~~ ~~end~~

وَعَلَمَهُ

لے گئے تھے اسی میں تو ان ہڈاروں اس ماننے

iter < iter_max و بازگشتی وابسته به اینکه calc و conv-flag = ۰ باشد

محلہ
ول

mm-chk = ٤٦ iter (٤٦) مورد وعده ایجاد شد. از آن جایی که نیش توان مدرار نمود، دراین بار

* (iter_max) conv-flag=1 iter > iter_max از طبق روش خارج خواهد شد.
* (iter_max) conv-flag=1 iter < iter_max از طبق روش خارج خواهد شد.

Let no-taps = 0 ~~no valid delay if tap~~

忽略 neighbor iter=iter_max , conv_flag==0 时停止

میں از فرم از صاف دا فارم (بڑیں داریں بڑیں) no-taps = 1
و از خوب سامنے سر خارج کرو

while $\Delta \text{Voltage} > \text{tolerance}$
 $\text{tap-it} = \text{tap-it} + 1$
 $\text{Vmax_index} = \text{find}(\text{V} == \text{Volt_max})$
 $\text{Vmin_index} = \text{find}(\text{V} == \text{Volt_min})$

$\text{gen_index} = \text{find}(\text{bus_type} == 2)$
 $\text{load_index} = \text{find}(\text{bus_type} == 3)$
 $(*) \text{gnd_index} = \text{find}(\text{bus_type} == 2, (\text{bus_type} == 2))$

$$\begin{aligned} P_g &= P + P_l \\ Q_g &= Q + Q_l \\ \text{load_index} &\Rightarrow P_l = \text{bus}(\text{gnd_index}, \text{V}) \\ Q_l &= Q_l - Q_{lg} \\ Q_{lg} &= \text{bus}(\text{gnd_index}, \text{V}) \end{aligned}$$

(*) $P_g = P + P_l$ $Q_g = Q + Q_l$

$$\begin{aligned} P_l &= P_l(\text{load_index}) = P_g - P \\ Q_l &= Q_l(\text{load_index}) = Q_g - Q_{lg} \\ P_g &= P_g(SB) = P(SB) + P_l(SB) \end{aligned}$$

$VV = V \cdot e^{j\phi}$ ϕ is angle

$$12 \times 1 \quad \text{tap_ratio}(k) = \begin{cases} 1 & \text{line}(k, u) = 0 \\ \text{line}(k, u) & \text{otherwise} \end{cases}$$

$$12 \times 1 \quad \text{Phase_shift} = \text{line}(k, v)$$

$$2-12 \quad tPS = \text{tap-ratio} \cdot e^{j\text{ay} \times \text{Phase-Shift} \times \theta / 180}$$

ضرر دارایی

$$12 \times 1 \quad \text{from-bus} = \text{line} (\therefore, 1)$$

مکانیزم تغییرات فاز

$$12 \times 1 \quad \text{to-bus} = \text{line} (\therefore, 2)$$

$$12 \times 1 \quad r = \text{line} (\therefore, 3)$$

$$12 \times 1 \quad r_{\text{tg}} = " (\therefore, 2)$$

$$12 \times 1 \quad chrg = " (\therefore, 0)$$

$$12 \times 1 \quad Z = r + j\text{ay} * r_{\text{tg}}$$

$$12 \times 1 \quad Y = \frac{1}{Z} \rightarrow (Y_k = \frac{1}{Z_k}) \quad , \quad k=1 \dots 12$$

مکانیزم تغییرات فاز

مکانیزم تغییرات فاز

$$\underline{12 \times 1} \quad MW-S(k) = VV(i) \times \text{conj} [VV(i) - tPS(k) \times VV(j) \times Y(k)]$$

$$+ VV(i) \times j \frac{chrg(k)}{r}] / [tPS(k) \times \text{conj}(tPS(k))]$$

$$i = \text{from-int}(k) \quad , \quad j = \text{to-int}(k)$$

میان to-bus, from-bus میان to-int, from-int*

$$12 \times 1$$

$$P-S = \text{real}(MW-S)$$

میان to-bus و from-bus

to Cbus from i1

$$12 \times 1$$

$$Q-S = \text{imag}(MW-S)$$

میان راسورتاتور

$$\underline{12 \times 1} \quad MW-R(k) = VV(j) \times \text{conj} [(VV(j) - VV(i) / tPS(k)) \times Y(k) + VV(j) \times j \frac{chrg(k)}{r}]$$

مکانیزم تغییرات فاز

$$P-R = \text{real}(MW-R)$$

میان to-bus و from-bus

from Cbus to

$$Q-R = \text{imag}(MW-R)$$

میان راسورتاتور

$$12 \times 1 \quad i_{\text{line}} = \begin{bmatrix} 1 \\ \vdots \\ 12 \end{bmatrix}$$

$$12 \times 1 \quad \text{line_ffrom} = [i_{\text{line}} | \text{From-bus} | \text{to-bus} | P_s | Q_s]$$

$$12 \times 1 \quad \text{line_fto} = [\text{to-bus} | \text{from-bus} | P_r | Q_r]$$

$$\text{P-loss} = \left(\frac{\text{V}_{\text{bus}}^2}{P_s} \right) + \left(\frac{\text{V}_{\text{bus}}^2}{P_r} \right)$$

$$\text{Q-loss} = \left(\frac{\text{V}_{\text{bus}}^2}{Q_s} \right) + \left(\frac{\text{V}_{\text{bus}}^2}{Q_r} \right)$$

$$11 \times 10 \quad \text{bus_sol} = [\text{bus_no} | V | \text{ang} \times \frac{180}{\pi} | P_g | Q_g | P_e | Q_e | G_b | B_b | \text{bus-type} | \text{qig_max} - \\ | \text{qig_min} | \text{bus}(1:13) | \text{volt_max} | \text{volt_min}]$$

$$12 \times 10 \quad (\text{line_sol}) = \text{line} \rightarrow \text{block?}$$

$$\text{line_flow} = \begin{cases} \text{if } 1 \rightarrow [i_{\text{line}} \text{ from-bus to-bus } P_s \text{ Q_s}] \\ \text{if } 2 \rightarrow [i_{\text{line}} \text{ to-bus from-bus } P_r \text{ Q_r}] \\ \text{if } 3 \rightarrow [\dots] \end{cases}$$

if conv-flag = 1 \Rightarrow أرجو توضيح

if display == 1 \Rightarrow أرجو توضيح \leftarrow lines

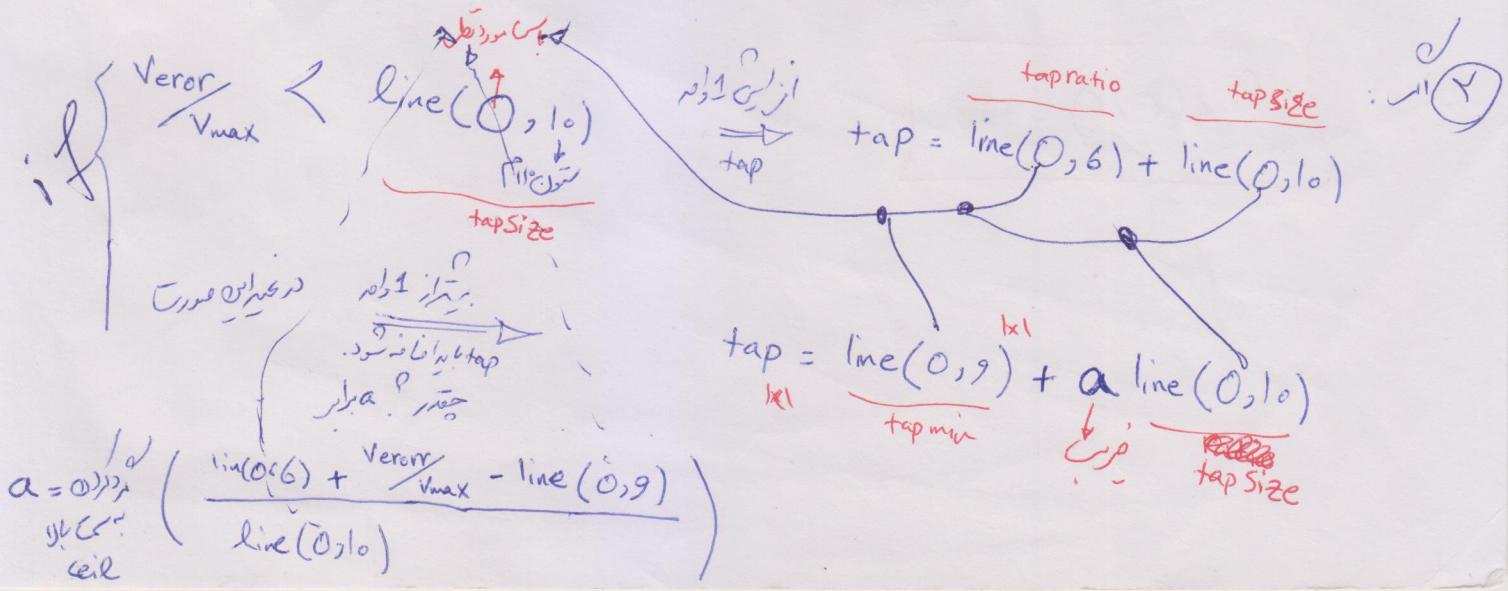
نوع ۲۷۵ : خروج و ورود خامینه ای، $\frac{1}{2}$ سطح آبایی، $\frac{1}{2}$ بار \downarrow left tap
line(8,6) \rightarrow line(8,10) \rightarrow line(8,12) \rightarrow line(8,14)
مورد جاذبه شد. مکان در دراده headflow در جویم رودخانه شود.

$$\begin{array}{l} \text{max-v-indx} \\ \hline V > \text{volt_max} \\ \downarrow \\ V < \text{volt_min} \end{array} \quad \begin{array}{l} \text{mn-chk=0} \\ \text{mn-chk=1} \end{array} \quad \begin{array}{l} \text{is max, min (W, J)} \\ \text{is max, min (U, J)} \end{array}$$

١) الـ line يُسمى بـ حـدـاـلـيـ خـوـزـيـ (سـمـ دـمـ خطـ (سـمـ دـمـ خطـ (سـمـ دـمـ خطـ (سـمـ دـمـ خطـ (line

٢) اـنـ سـمـ دـمـ خطـ تـكـنـ يـتـحـدـ طـارـطـ : فـنـ دـمـ مـدـيـ باـسـوـنـ ١٨ tap max

$$V_{\text{tx}} - V_{\max} \left(\frac{\rho}{\sigma} \right) = \text{Vererr}$$



R-4) $\text{line}(0,6) = \text{tap}$

$\text{tap} = \min(\text{line}(0,8), \max(\text{tap}, \text{line}(0,9)))$

$\text{line}(0,8)$ $\text{line}(0,9)$

*

$\Rightarrow \boxed{\frac{\text{tap ratio}}{\text{line}(0,6)} = \text{tap}}$

مقدار tap میان $\text{line}(0,8)$ و $\text{line}(0,9)$ باشد. (این دو خط را بخواهید)
 هر دو باهم اتفاق نمایند.

با احتساب: $V_{\min} - V = V_{\text{error}}$

$(V_{\min} - V) / \text{line}(0,8)$ میان $\text{line}(0,8)$ و $\text{line}(0,9)$

$$\left\{ \begin{array}{l} V_{\text{error}} / V_{\min} < \text{line}(0,10) \\ \text{tap} = \text{line}(0,6) + b \text{line}(0,10) \end{array} \right.$$

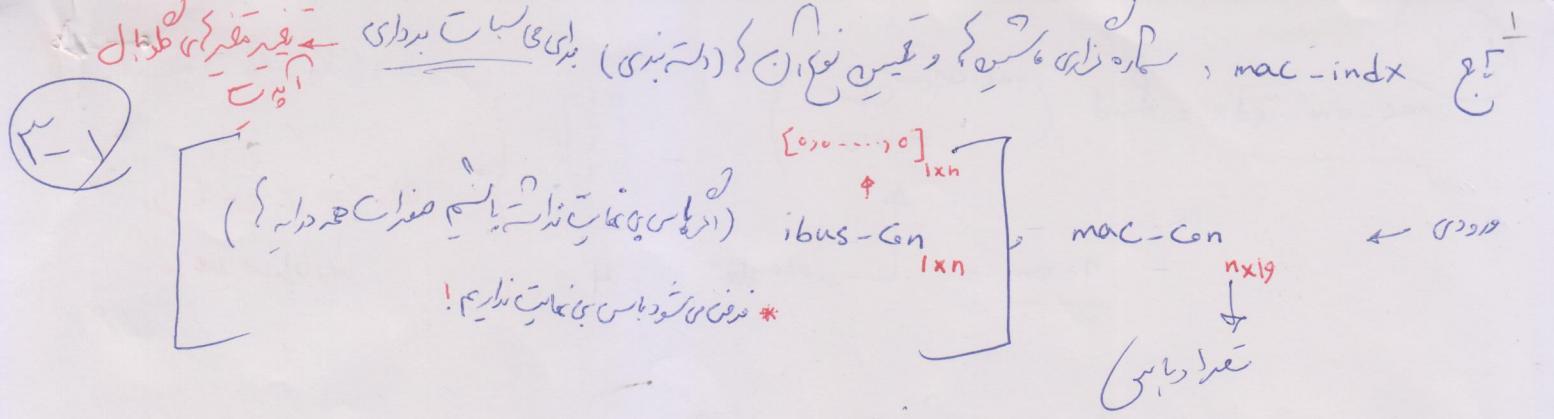
مقدار b را با $\frac{V_{\text{error}} - \text{line}(0,6)}{\text{line}(0,10)}$ محاسبه کنید.

$b =$
 $\frac{V_{\text{error}} - \text{line}(0,6)}{\text{line}(0,10)}$
 fix

$$\left(\frac{\text{line}(0,6) - V_{\text{error}}}{V_{\min} - \text{line}(0,9)} \right)$$

* $b =$ $\frac{V_{\text{error}} - \text{line}(0,6)}{\text{line}(0,10)}$ $\Rightarrow \text{tap}$

$\boxed{\text{line}(0,6) = \text{tap}}$



پس از دریافت عددی در ردیف ۱۹، سکون دارد. سکون کمتر از ۲۲ و ۲۳ نداشته است. سکون کمتر از ۲۰ نیز دارد. سکون کمتر از ۱۹ نیز دارد.

بلطفاً ابتدا از مقدار ۱۹ کمتر از سکون داشت. همچنان ۱۹ سکون داشته باشد (سکون کمتر از ۲۰ و ۲۱) سکون کمتر از ۲۰ مقدار ۱۹ را ندارد.

$$\begin{matrix} \text{mac} \\ \text{pot} \end{matrix} = \underset{nxn}{\text{zeros}}(n, 15)$$

دو سیاهی بزرگ

$$\text{macmax} = \max(\text{mac-con}(:, 1)) \quad ; \quad \text{mac-con} \text{ عده کم اول}$$

$$\text{mac-int} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \rightarrow \text{macmax} \text{ کم اول}$$

محباقلت به جزءی از mac-con در ردیف اول عدد خود را در mac-int

$$\text{mac-con} = \begin{bmatrix} 1 \\ 2 \\ 5 \\ 7 \end{bmatrix} \Rightarrow$$

$$\text{macmax} = 7$$

$$\Rightarrow \text{mac-int} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 3 \\ 0 \\ 4 \end{bmatrix}$$

برایت سیاهی از mac-con

$$\text{not-ibind} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} - \text{ibind}$$

از زیرها

$n-ib = 0$

$n-ib-em = 0$

$n-ib-trac = 0$

$n-ib-sub = 0$

منه: این سیاهی را می خواستیم!

منه: این سیاهی را می خواستیم!

لطفاً در نظر بگیرید:

$$\text{mac-em-idx} = \text{find} \left(\begin{array}{l} \text{mac-on} \\ \text{= 0} \end{array} \right) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

\uparrow

n-em = مقدار = 4

شیخ x_1 mac-em

نامه لدرستو

نامه مسخردار

$$\text{mac-tra-idx} = \text{find} \left(\begin{array}{l} \text{tra-on} \\ \neq 0, \text{mac-on} \\ = 0 \end{array} \right)$$

n-tra, مقدار

شیخ mac-tra

نامه که درستو

نامه مسخردار

$$\text{mac-sub-idx} = \text{find} \left(\begin{array}{l} \text{sub-on} \\ \neq 0 \end{array} \right)$$

n-sub = مقدار

global

شیخ mac-sub

نامه که درستو

نامه مسخردار

ایجاد شرطی برای این که هر کدام از اینها مقدار مثبت باشد

n-ib-em
n-ib-tra
n-ib-sub
n-ib

ایجاد شرطی برای وجود داشت (فرض شرطی)

[optional]

[0, 1, 0] (یک شرطی که در ورودی ماهیت داشته باشد و شرطی که در خروجی داشته باشد) \oplus ibus-on \oplus وارد شده در ماهیت

$$\text{mac-ib-idx} = \text{find} (\oplus \text{ibus-on} = 1)$$

n-ib-em: مقدار: $\left\{ \text{find} \left(\begin{array}{l} \text{mac-on} \\ = 0 \end{array} \right) \text{And } \text{ibus-on} \\ = 1 \right)$

n-ib-tra: مقدار: $\sim \left(\text{tra-idx} \sim \text{مقدار} \text{ And } // \right)$

n-ib-sub: $(\text{sub-idx} \sim \sim \text{مقدار} \text{ And } N)$

(\rightarrow). n_{em} جدول شود و $n_{em} = 3$: Flag = 0 \Rightarrow mac-em \rightarrow $n_{em} = 3$ \Rightarrow initialization: $n_{em} = 3$ \Rightarrow 3 مولود initialization: $n_{em} = 3$

mac-em-ind = 0 \leftarrow n_{em} clear: Vectorized orde \rightarrow obf (vectorized) \rightarrow n_{em} max-index

$$\text{busnum} = \text{bus-int}(\text{mac-con}(0,2)) \Rightarrow \begin{matrix} \text{busnum} \\ \text{bus-int} \end{matrix} \xrightarrow{\text{mac-con}(0,2)} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \Rightarrow \begin{matrix} \text{busnum} \\ \text{bus-int} \end{matrix} \xrightarrow{\text{mac-con}(0,2)} \begin{matrix} 2 \\ 3 \end{matrix}$$

$$\begin{aligned} \text{mac-pot}(0,1) &= \underbrace{\text{busnum}}_{\text{bus-int}} \times \underbrace{\text{ones}(n_{em}, 1)}_{[1]} \\ &\quad \text{mac-index} \\ &\quad \rightarrow \text{bus-int} \\ &\quad \cancel{\text{mac-con}(0,3)} \end{aligned}$$

mac-index
mac-pot
nxis
mac-pot

$$\text{mac-pot}(0,2) = \text{ones}(n_{em}, 1) = [1]$$

$$\text{e term}(0,1) = \underbrace{\text{bus}}_{\text{loadflow}}(\text{busnum}, 2)$$

$$\theta(0,1) = \text{bus}(\text{busnum}, 3) * \cancel{\text{mac-con}(0,2)}$$

$$Pelect(0,1) = \text{bus}(\text{busnum}, 4) * \cancel{\text{mac-con}(0,22)}$$

جای مولود فرجه \rightarrow
که θ را $\pi/180$ درadian

$$qelect(0,1) = \text{bus}(\text{busnum}, 5) * \text{mac-con}(0,23);$$

$$\text{curr} = \sqrt{2 \times \left(Pelect^2 + qelect^2 \right)} \xrightarrow{\text{e term}(0,1) * \text{mac-pot}(0,1)}$$

$$I = \frac{\sqrt{P^2 + Q^2}}{V}$$

$$\phi_{ij} = \text{atan2} \left(qelect(0,1), Pelect(0,1) \right) \Rightarrow \phi_{ij} = \tan^{-1} \left(\frac{Q}{P} \right)$$

$$V = \text{eterm}(0,1) * e^{j \cdot \text{theta}(\text{busnum}, 1)}$$

$\cancel{R-E}$

$$\text{curr} = \text{curr} \cdot e^{j (\text{theta}(\text{busnum}, 1) - \phi)}$$

$$eprime = V + j \text{mac-cn}(0,7) * \text{curr}$$

$$J_{\delta} E' = V + j X_d I$$

$$ei = eprime \rightarrow \text{copy}_{\text{curr}}^{\text{eprime}}$$

$; l$ eprime

$$\text{mac-ang}(0,1) = \text{atan}_2\left(\frac{\text{imag}(ei)}{\text{real}(ei)}\right) \rightarrow \delta = \tan^{-1}\left(\frac{\text{imag}(e')}{\text{real}(e')}\right)$$

$$\text{mac-spd}(0,1) = \text{ones}(n-\text{em}, 1)$$

$$\omega = [1] \quad \text{aics}$$

$$\text{rot} = j e^{-j(\text{mac-ang}(0,1))}$$

$$\rightarrow \text{rot} = j e^{-j\delta}$$

$$\text{PSI-re}(0,1) = \text{real}(eprime)$$

$$\text{PSE-im}(0,1) = \text{imag}(eprime)$$

$$\text{eprime} = \text{eprime} \circ * \text{rot} \rightarrow \text{ej}' = \text{real}(\vec{v})$$

$$\text{edprime}(0,1) = \text{real}(eprime) \quad \text{eq}' = \text{imag}(\vec{v})$$

$$\text{eprime}(0,1) = \text{imag}(eprime)$$

$$\text{curr} = \text{curr} * \text{rot}$$

$$\text{curr}_g(0,1) = \text{real}(\text{curr})$$

$$\text{curr}_g(0,1) = \text{imag}(\text{curr})$$

pu

$$\begin{cases} \text{curr}_d(0,1) = \frac{\text{real}(\text{curr})}{\text{mac-pot}(0,1)} \\ \text{curr}_q(0,1) = \frac{\text{imag}(\text{curr})}{\text{mac-pot}(0,1)} \end{cases}$$

$$V = V \cdot * \text{rot} \rightarrow \text{park}$$

$$\text{ed}(0,1) = \text{real}(V)$$

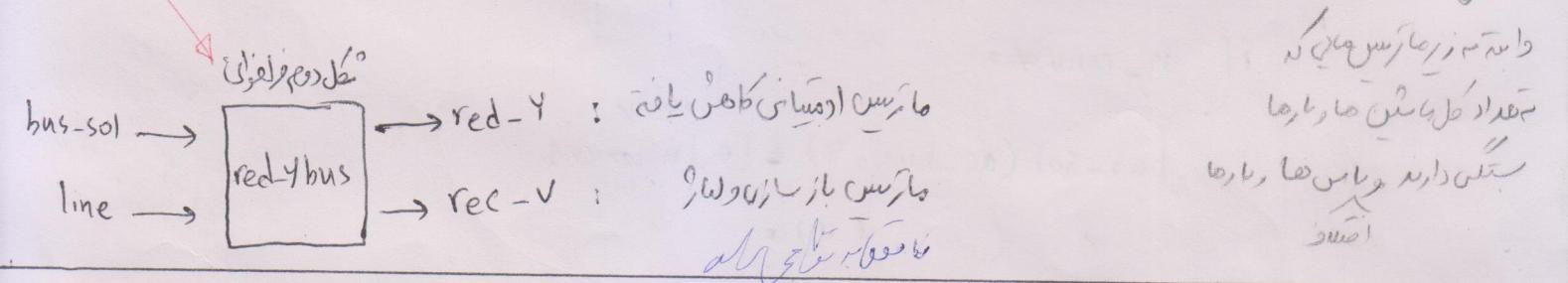
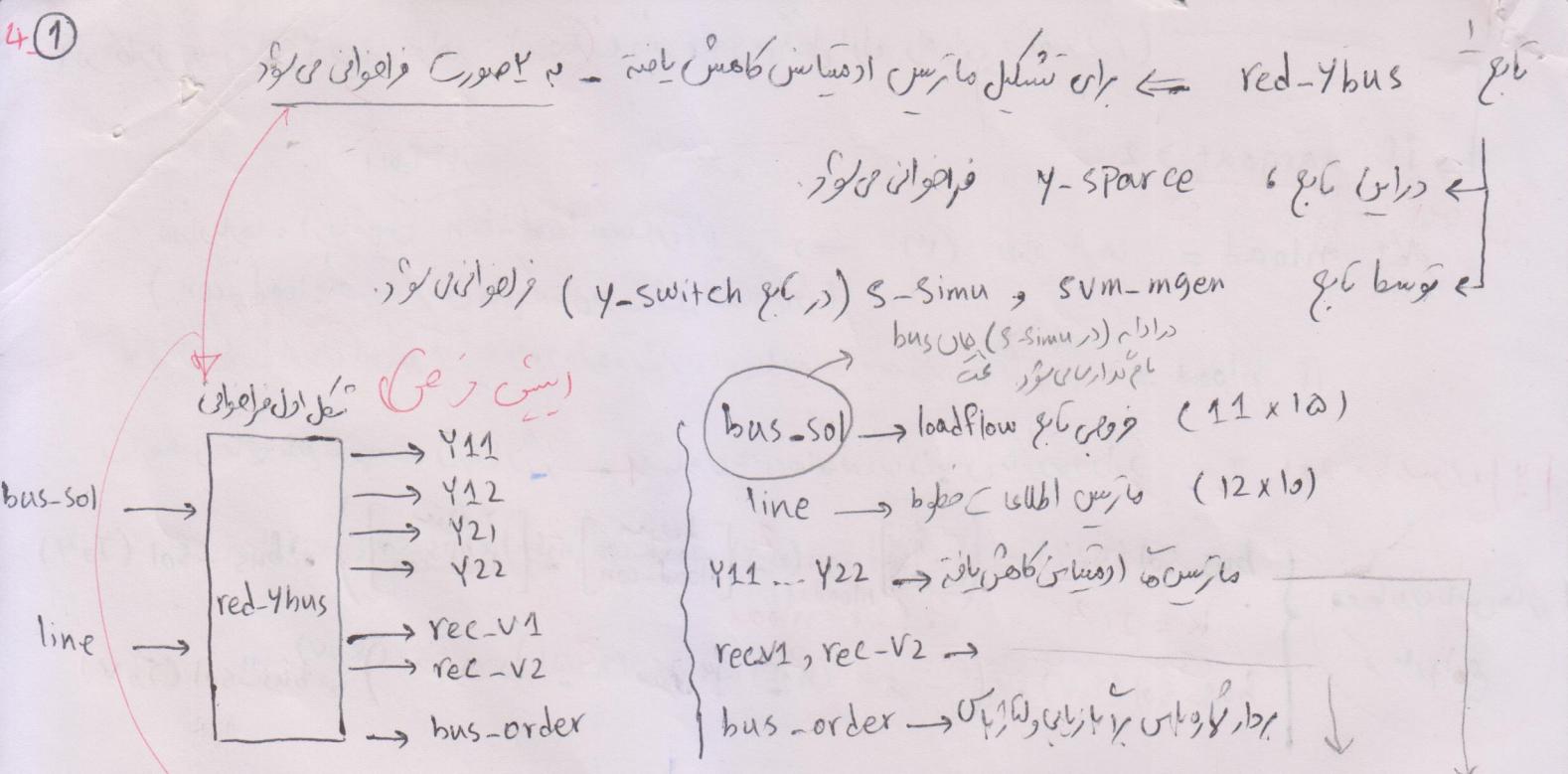
$$\text{eq}(0,1) = \text{imag}(V)$$

$$\text{Vex}(0,1) = \text{eqprime}(0,1)$$

$$\begin{cases} \text{Pmech}(0,1) = \text{Pselect}(0,1) \\ \text{Pelec} = \frac{0}{6 \cdot 10^6 \cdot 10^3} \end{cases}$$

$$\text{mac-pot}(0,1)$$

4 - ①

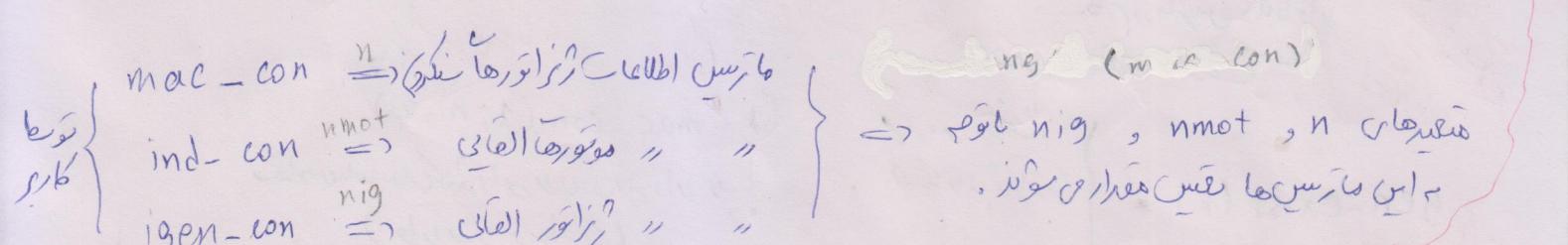


online $\xrightarrow{\text{line}}$ boksha (12) , nbus $\xrightarrow{\text{bus}}$ boksha (11)

$n \rightarrow$ نَمْوَةٌ (نَمْوَةٌ) , nmot → نَمَادِيٌّ (نَمَادِيٌّ)

$n_{\text{ig}} \rightarrow \underbrace{\text{نـيـجـ}}_{\text{نيـجـ}} (n)$; $n_{\text{tot}} \rightarrow \underbrace{\text{نـيـجـ}}_{\text{نيـجـ}} \text{ (} n + n_{\text{mot}} + n_{\text{ig}} \text{)}$

$$n\text{gm} \rightarrow \text{Levi's} + \text{Gelso} (\text{n+m})$$



مادیں اسیں بانی

$$11 \times 11 Y_{-d} = Y\text{-sparse}(\text{bus-sol}, \text{line})$$

، line) ← y-sparse واقعون
ریاضی، y-d ~ y هست

$$11 \times 1 \quad V = bus_sol(0, 2)$$

, V: J Bus; Wojciech
C Bus jas

اگر کابر بسیار زیاد باشد: (بصورت دیرهای بسیار بزرگ اول راهنمایی نماید)

if $n_{\text{argout}} > 2$

تعداد مدارها در هر چندین مدار

$n_{\text{load}} =$

تعداد مدارها

load-con

load: صفر ماتریس های میان تعرفه شده است (که خود را)

* در این مدل مدارهای ماسن ها لازم و محدود دارد. مدارهای ماسن های میان تعرفه شده است که این کار را کار نمایند.

مدارهای ماسن های میان تعرفه شده است که این کار را کار نمایند.

$$\left\{ \begin{array}{l} \text{bus-sol}(k, q) = (1 - \text{load-con}(n^2 x) - \text{load-con}(n^2 q)) \times \text{bus-sol}(k, q) \\ k = j(n^2), \quad n^2 \in 1, \dots, n_{\text{load}} \\ \text{bus-sol}(k, v) = (1 - \dots - (n^2 a)) \times \dots \times (k, v) \end{array} \right.$$

if $n_{\text{conv}} \neq 0$

$$\text{bus-sol}(\text{ac-bus}, q) = [0]_{n_{\text{conv}} \times 1}$$

، (، v) = "

end

end

با قم بعمل فعالی، این مسیر
از کاربری بود. فعلاً از این
مسیر دور خواهد شد.

$$11 \times 1 \quad P_l = \text{bus-sol} \quad \text{سیو ۶۰} \quad Q_l = \text{bus-sol}(1, v) \quad 11 \times 1$$

سیو ۱۰۰

$$\text{gen-exist} \rightarrow \begin{bmatrix} \text{دار} \\ \text{سیو} \end{bmatrix}_{11 \times 1}$$

با قم این کار را کار نمایند

* سیو اول اول bus میان را دارد.
ماسن های اینست که این کار را کار نمایند.

$$\star \rightarrow \text{بزرگترین مدار و ماسن} = \max(\text{bus-sol}(:, 1))$$

کاربری ماتریس bus دارد
که این کار را کار نمایند

کرد است

$$k = \text{mac-con}(1:n, 2)$$

سیو این ماسن های کاربری کار نمایند

(وارد شده توسط مدار)

other

سیو که کار نمایند

$$\text{gen-exist}(k) =$$

$$\left\{ \begin{array}{l} \text{کاربری درست} \\ (\text{سیو های میان}) \end{array} \right.$$

0

other

سیو که کار نمایند

$$\text{mac-con} \quad \text{سیو این ماسن که کار نمایند} = \text{سیو} (4)$$

اردرایر هفتم برای کاربری را که بجزئی های کاربری است

که رهی کاربری دارد

$\text{netgen} = i \quad \text{if} \quad \text{Part}(i) < 1 \rightarrow \text{gen-exist}(i)$ - 4- Part

\hookrightarrow نحوه زیرا عرضه محدود ندارد. (نحوه محدود)

$\text{Part}(i) = \text{gen-exist}(m) \rightarrow m = \text{bus_sol}(:, 1) \rightarrow$ نحوه

$11 \times 1 \downarrow$ 11×1

$$\begin{aligned} \text{11x1} \quad \text{Pl}(k) &= \text{Pl}(k) - \text{bus_sol}(k, \omega) \quad \left\{ k \in \text{netgen} \right. \\ \text{11x1} \quad \text{Ql}(n) &= \text{Ql}(n) - \text{bus}(n, \omega) \quad \left. \right\} \end{aligned}$$

$$(Pl - jay^* \alpha l) / v^2 \leftarrow \text{دایرکٹ} \quad \text{(رسانه مکانیکی)} \\ \gamma_l = \left\{ \begin{array}{l} \text{اسپرس مکانیکی} \\ \text{و} \quad \text{جفیق} \end{array} \right.$$

$$\text{Ex1 } \text{ra} = \underbrace{\text{mac-con}(:, 0)}_{\text{Ex1}} \times \underbrace{\text{basmva}}_{\text{Ex1}} / \text{mac-con}(:, 3)$$

n tot = 8

$$\text{Ex1 testXPP}(k) = \begin{cases} 0 & \text{if } \text{mac-con}(k, \underline{n}) = 0 \\ 1 & \text{if } \quad " \quad \neq 0 \end{cases}$$

Ex1 test &P = not(test+&P)

Ex1 test+X = not (test&X),
 مثلاً $t \& p \rightarrow \neg(t \& p)$ و $t + p \rightarrow \neg(t + p)$

$$\frac{dx_{PP}}{dt} = \frac{xd(k_1) \cdot mac_con(k_2) \cdot x_{basmva} / mac_con(k_3)}{mac_con(k_4)}, \quad exp \text{ stuck}$$

$xP // v \Rightarrow xd(x, i) = mac_con(k, v) \times n / mac_con(k))$

$$\text{Ex1 } Y = \frac{1}{ra + jxd}, \quad \text{Ex1 } Yg = \frac{1}{ra(k_1) + jxd(k_1)}, \quad k_1 \in \mathbb{N}$$

* مارسون مکانیزمات مربوط به فرآیندرها سندوں را دارد. سهول اول این مارسون کارهای فرآیندرها درست از ۱۰٪ و سهول دهم کارهای فرآیندر را که در فرآیند تولید شده اند در دارد. درینکه مفهوم فرآیندرها ۲۰٪ درست روی مارسون هاست

$$\Rightarrow \text{Perm} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \\ 4 & 0 & 0 & 0 \end{bmatrix} \quad \leftarrow \text{نحوه ایجاد}\}$$

حال وضن لعنة رکزاتور مده ۳، رسن مس هماره ۷ واردات است ایسی اری باس

* اگر روسیہ کے مالک بننے اور اُنہوں نے اپنے دارا رہنے والے مالکوں کو مارسیں تو اُنہوں نے اپنے دارا رہنے والے مالکوں کو مارنے کا حق پیدا کیا ہے؟

$$\text{Perm}_2^2 \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{rg} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}_{\text{px1}} \leftarrow \begin{array}{l} \text{اعداد ممكناً اثنان} \\ \text{لأول زوج} \\ \text{صفوة زوج} \end{array}$$

محل خود و مسیرهای پرداختی Perm نامیده می‌شوند.

$$Y_{mod} = \begin{bmatrix} y_{11} & & & \\ & y_{21} & 0 & \\ & & y_{31} & \\ & & & y_{41} \end{bmatrix} \times \text{Perm}^T$$

Ex

Ex

n-tot nbus

مقدار دهنده $y - b$ با این $x \times 11$ تکمیل شود و مجموع از درایه های آن بمساحت مقدار $(y - b)$ باشد:

$$Y - b(i,j) \leftarrow -Y \bmod (ij)$$

ex: $\begin{matrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{matrix}$

$i \in \{1, 2, 3, 4\}$
 $j \in \{1, 2, 3, 4\}$

لقد تم حذف المدخلات المكررة

$$Y_{-d(i,j)} = Y_{-d(i,j)} + m \quad \Rightarrow \quad \left\{ \begin{array}{l} m = \text{Perm} \neq Y \bmod \\ \end{array} \right.$$

علاء الدين (البيروي) m ٤٢٤
علي بن ابي طالب (رسول الله) m ٤٢٤

کی نتیجہ کا مکمل، حکومت فوڈ اے سینڈ. ٹیکنیکل سادگری کے لئے ایک بسیاری کی تحریر کرنے والے ایک صنعتی ادارہ ہے۔ اسی صنعتی ادارے کی تحریر کرنے والے ایک صنعتی ادارہ ہے۔

$$\text{mot max} = 0 \quad , \quad \text{ig max} = 0$$

$$Y-a(i,j) = Y(i,j), \quad i,j = \text{in}(k), \quad n_tot \leq 1 \leq k$$

n_tot x n_tot
x x

$$Y-c = \begin{pmatrix} Y-b \\ 0 \end{pmatrix}^T \quad (\text{Transpose})$$

11x11 11x11

if narginout <= 2

$\Leftrightarrow \text{load}_1, \text{load}_2, \text{load}_3, \dots, \text{load}_n$

(if 1)

$$Y_{12} = -\text{inv}(Y-d) \times Y-c$$

11x11 11x11 11x11

$$Y_{11} = Y-a + Y-b * Y_{12}$$

11x11 11x11 11x11 11x11

① (else)

: حملات

(if nload ≠ 0)

: حملات، توزيع (if 2)

$$\text{bus-order}(n) = \begin{cases} k=1 : nload & \text{load_on}(k, 1) \\ k=nload+1 : 11 & \text{load_off}(k, 1) \end{cases}$$

حملات و توزيع
($\omega_1 = \omega_2$)

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{bmatrix}$$

$$P(i,j) = \begin{cases} 1 & i=k, j=\text{bus-order}(k), 11 \leq i \leq k \\ 0 & \text{other} \end{cases}$$

نقطة على

$$Y-b = Y-b \times P^T, \quad Y-c = P \times Y-c, \quad Y-d = P * Y-d * P^T$$

11x11 11x11 11x11 11x11

$$k=1 : nload \Rightarrow \begin{cases} Y-b1 = Y-b(:, k) \rightarrow Y-b \text{ حملات 1 } \leq k \leq nload \\ Y-b2 = Y-b(:, k:k) \rightarrow Y-b \text{ حملات } 1 \leq k \leq nload+1 \text{ حملات } (k+1) \end{cases}$$

nload+1 : nbus

$$Y-c1 = Y-c(k, :) , \quad Y-c2 = Y-c(k:k, :)$$

2x11 2x11

$$Y-d1 = Y-d(k, :) , \quad Y-d2 = Y-d(k:k, :)$$

2x11 2x11

$$2 \times 2 Y_{-d11} = Y_{-d1}(:, k), \quad Y_{-d12} = Y_{-d1}(:, kk)$$

$$4 \times 2 Y_{-d21} = Y_{-d2}(:, k), \quad Y_{-d22} = Y_{-d2}(:, kk)$$

$$\text{rec_v2} = -\frac{\text{inv}(Y_{-d22}) \times \begin{bmatrix} 4 \times \xi & | & 4 \times 2 \\ Y_{-c2} & | & Y_{-d21} \end{bmatrix}}{4 \times 4}$$

$$\left. \begin{array}{l} \text{rec_v1} = \text{rec_v2}(:, 1:n_tot) \\ 4 \times \xi \end{array} \right\} \xleftarrow{\text{rec_v2}} \text{if } n_tot \leq 1 \text{ then}$$

$$\left. \begin{array}{l} \text{rec_v2} = \text{rec_v2}(:, \underbrace{n_tot+1}_{1 \triangleq}, \underbrace{n_tot+n_load}_4) \\ 4 \times 2 \end{array} \right\} \xleftarrow{\text{rec_v2}}$$

$$\text{ex} \quad Y_{11} = Y_a + Y_{-b2} \times \text{rec_v1}$$

$$\text{ex} \quad Y_{12} = Y_{-b1} + Y_{-b2} \times \text{rec_v2}$$

$$2 \times \xi \quad Y_{21} = Y_{-c1} + Y_{-d12} \times \text{rec_v1}$$

$$2 \times 2 \quad Y_{22} = Y_{-d11} + Y_{-d12} \times \text{rec_v2}$$

اگر کفر را نمایم (داخلی) $\left(\begin{array}{l} \text{if } n_conv \neq 0 \\ \text{else if } \end{array} \right)$

$\left(\begin{array}{l} \text{if } \\ 2-1-1 \end{array} \right)$

(جیلیکی و معمولی) بدهی اگر گذشتیم $\left(\begin{array}{l} \text{if } \\ \text{else if } \end{array} \right)$

(else) 2-1 \leftarrow سعیدی

اگر کفر نباشد

$$C_{-j4} = Y_{12} = -\text{inv}(Y_{-d}) \times Y_{-c};$$

$$\text{ex} \quad Y_{11} = Y_a + Y_{-b} \times Y_{12}$$

$$11 \times \xi \quad \text{rec_v1} = Y_{12}$$

$$Y_{12}, Y_{-21}, Y_{-22}, \text{rec_v2} \Rightarrow \text{bus-order}$$

جایزی

$$\text{bus-order} = \text{bus_sol}(:, 1)$$

تغیرات

٤-٤ mac-em (الإنت.) الباقي جناب برس flag=0 mac-tra flag

ايجي سعر n-tra ≠ 0 } ٤ مقدار معاون معاون معاون flag = 0

mac-tra - idx = 0 → mac-idx

(جودي)

if mac-on(0,14) = 0 → mac-on(0,14) = 999

لوري ١٤

busnum = bus-int(mac-on(0,2)) → [4]

mac-pot(0,1) = busnum × ones(n-tra, 1) → mac-on(0,3)

mac-pot(0,2) = ones(n-tra, 1)

eterm(0,1) = bus(busnum, 2)

theta(busnum, 1) = bus(busnum, 3) * $\frac{\pi}{180}$

poleet(0,1) = bus(busnum, 4)

qeteet(0,1) = bus(busnum, 5) * $\frac{mac-on(0,22)}{mac-on(0,23)}$

curr =

phi = $\frac{1}{2} \sin^{-1} \left(\frac{V}{mac-em} \right)$

V = mac-em

curr =

$e' = V + \underbrace{mac-on(0,5)}_{em \neq 0} + j \underbrace{mac-on(0,7)}_{curr} * curr$

$e_i = V + mac-on(0,5) + j mac-on(0,11) * curr$

mac-ang(0,1) = atan2(imag(ei), real(ei)) → tan⁻¹($\frac{imag(ei)}{real(ei)}$)

mac-Spd(0,1) = ones(n-tra, 1)

rot, psi-re, psi-im, eprime, edprime, eprime, curr, currq, currqg, curdg, curd, mac-em

$$P_{\text{mech}}(0,1) = \underbrace{p_{\text{elect}}(0,1)}_{T_C, \omega} * \underbrace{\text{mac-pot}(0,1)}_{\rightarrow} + \text{mac-on}(0,5) * \text{maccarmag} *$$

maccurrmag

$$P_m = P_e + R_i^2$$

$$\text{maccarmag} = \text{abs}(\text{curr}) \xrightarrow{\text{curr}} |curr|$$

$$V = V \cdot \times \text{rot} \xrightarrow{\text{mac em } \rightarrow P} \xrightarrow{\substack{\text{real } \\ (\nu)}} \text{eq}(0,1) = \text{imag}(V)$$

نحوه

$$b = \begin{bmatrix} 0.8 & 1 + \text{mac-on}(0,20) \\ -0.8 & " \\ 0 & " \end{bmatrix}$$

$$\text{inv-sat} = \text{inv} \left(\begin{bmatrix} 0.84 & 0.8 & 1 \\ 1 & 1 & 1 \\ 1.44 & 1.2 & 1 \end{bmatrix} \right) \xrightarrow{\text{جذب}} \text{inv-sat} = \begin{bmatrix} 12.5 & -25 & 12.5 \\ -27.5 & 50 & -22.5 \\ 15 & -24 & 10 \end{bmatrix} \xrightarrow{\text{رسخ}} \underset{3 \times 3}{\text{رسخ}}$$

$$\text{mac-pot}(0,3) = b \xrightarrow{\text{جذب}} \text{inv-sat}(1,:)^T$$

$$\text{mac-pot}(0,4) = b \xrightarrow{\text{جذب}} \text{inv-sat}(2,:)^T$$

$$\sim(0,5) = " \xrightarrow{\text{جذب}} \sim(3,:)^T$$

$$E - I_{\text{sat}} = \underbrace{\text{mac-pot}(0,3) * \text{eqprime}(0,1) * \text{eqprime}(0,1)}_{+ \text{mac-pot}(0,5)} + \underbrace{\text{mac-pot}(0,4) * \text{eqprime}(0,1)}_{**}$$

$$\text{no-satidx} = \text{find}(\text{eqprime}(0,1) < 0.8);$$

$$E - I_{\text{sat}}(\text{no-satidx}) = \text{eqprime}(0(\text{no-satidx}), 1)$$

لهم ما زلت أتمنى أن تكتب لي إساتي وتحفظه في ذاكرة جهاز الكمبيوتر

$$③ \quad 4-5. \quad V_{ex}(0,1) = E - I_{sat} + [mac_G_n(0,6) - mac_G_n(0,7)] * Curd_g(0,1)$$

$$\text{fld cur } (0,1) = V_{ex}(0,1)$$

→ mac-tra gl. abgeschr. O^u

(4)

(mactra, mac-em) flag = 0 \Rightarrow mac-sub $\not\in$

57 21
50 21

\rightarrow flag \oplus n-sub fo flag = 0 $\left\{ \begin{array}{l} \text{if } 6 \text{ is not in sub fo then} \\ \text{flag = 1} \end{array} \right.$

mac-sub-idx = 0 \rightarrow mac-sub-idx \oplus

if $\text{mac-Cn}(0,8) \neq \text{mac-Cn}(0,13) \Rightarrow \text{mac-Cn}(0,13) = \text{mac-Cn}(0,8)$

" " $(0,15) = 0 \rightarrow \text{mac-Cn}(0,15) = 000$

busnum = busint ($\text{mac-Cn}(0,2)$)

mac-pot(0,2) = ones(n-sub, 1)

mac-pot(0,8) = $\text{mac-Cn}(0,7) - \text{mac-Cn}(0,4)$

mac-pot(0,9) = $\frac{\text{mac-Cn}(0,8) - \text{mac-Cn}(0,4)}{\text{mac-pot}(0,8)}$

! \oplus mac pot \oplus mac Cn \oplus \star

mac-pot(0,7) = $\left[\text{mac-Cn}(0,6) - \text{mac-Cn}(0,7) \right] + \text{mac-pot}(0,9)$

" 10 = $\frac{\text{mac-Cn}(0,7) - \text{mac-Cn}(0,8)}{\text{mac-pot}(0,8)}$

" 6 = $\frac{\text{mac-Cn}(0,6) - \text{mac-Cn}(0,7)}{\text{mac-pot}(0,8) + \text{mac-pot}(0,10)}$

" 13 = $\text{mac-Cn}(0,12) - \text{mac-Cn}(0,11)$

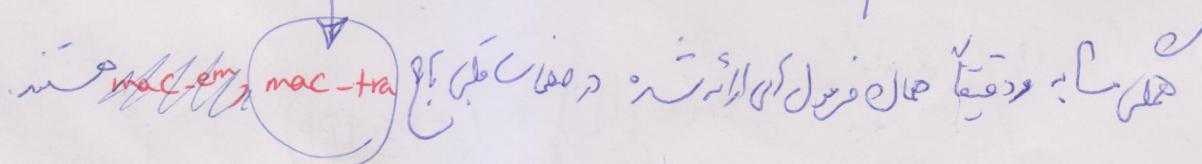
" 14 = $\left[\text{mac-Cn}(0,13) - \text{mac-Cn}(0,14) \right] / \text{mac-pot}(0,13)$

$$⑤ \quad \text{mal-pot}(0,12) = [\text{mal-Gr}(0,11) - \text{mal-Gr}(0,12)] * \text{mal-pot}(0,14)$$

$$\therefore 15 = \frac{\text{mal-pot}(0,12) - \text{mal-pot}(0,13)}{\text{mal-pot}(0,13)} *$$

$$\therefore 11 = \frac{\text{mal-Gr}(0,11) - \text{mal-Gr}(0,12)}{\text{mal-pot}(0,13) * \text{mal-pot}(0,15)} *$$

$\text{etern}(0,1)$, theta, pdeet, qelect, curr, phi, v, curr,
 ϵ^i , mac-ang, mac-spd, rot, curr, mcurmag, pselect
 curdg, curdg, curd, curq, v, ed, eq
 Pmech



الطباطبى
الفاصل
الراحل \rightarrow $\text{eqra} = \text{eq}(0,1) + \text{mal-Gr}(0,5) * \text{curdg}(0,1)$

$$\text{eqra} + \text{mal-Gr}(0,8) * \text{curdg}(0,1)$$

$$\text{psi_kd}(0,1) = \text{eqra} + \text{mal-Gr}(0,4) * \text{curdg}(0,1)$$

$$\text{eqprime}(0,1) = \text{eqra} + \text{mal-Gr}(0,7) * \text{curdg}(0,1)$$

$$\text{edra} = -\text{ed}(0,1) - \text{mal-Gr}(0,5) * \text{curdg}(0,1)$$

$$\text{psi_qpp} = \text{edra} + \text{mal-Gr}(0,13) * \text{curqg}(0,1)$$

$$\text{psi_kq}(0,1) = \text{edra} + \text{mal-Gr}(0,4) * \text{curqg}(0,1)$$

$$\text{edprime}(0,1) = \text{edra} + \text{mal-Gr}(0,12) * \text{curqg}(0,1)$$

⑥ Inv-Sat \rightarrow mac-tra position

b = ~ ~ ~

mac-pot(0,3), 4, S \rightarrow mac tra

E-Jsat, no-statidx \rightarrow ~ ~

$$Vex(0,1) = E-Jsat + \text{mac-pot}(0,6) * \left[\text{eqprime}(0,1) - \text{psikd}(0,1) \right] + \text{mac-pot}(0,7) * \text{curdg}(0,1)$$

$$fldcur(0,1) = Vex(0,1)$$

$$\text{PSi-re}(0,1) = \sin(\text{mac-ang}(0,1)) * (-\text{psi}\underline{qPP}) + \cos(\text{mac-ang}(0,1)) * \text{PSid}\underline{PP}$$

$$\text{PSi-im}(0,1) = -\cos(\text{mac-ang}(0,1)) * (-\text{psi}\underline{qPP})$$

$$+ \sin(\text{mac-ang}(0,1)) * \text{PSid}\underline{PP}$$

۴۰
* خروجی ندارد. از اطلاعات طبعی مارسین استفاده شود و با فرآوریان گنجایشی این کامپیوتر را در یک شبکه red-Ybus مارسین کنید.

D-1 : pre-fault, fault-immediate post-fault-final fault clear

(۱) ساختار یک سیستم (همایش) pre-fault

۱۶۱ nload = size (load-con (:, 1)) ← تعداد بارگذاری شده سیستم

خواهیم داشت
red-ybus $\begin{bmatrix} Y_{gPrf}, Y_{gnePrf}, Y_{ncgPrf}, Y_{neprf}, V_{rgPrf}, V_{rnePrf}, boprf \end{bmatrix}$ = red-Ybus
 $\begin{bmatrix} Y_{11} & Y_{12} & Y_{21} & Y_{22} & rec-v1 & rec-v2 & bus-order \\ ex_1 & ex_2 & ex_3 & ex_4 & ex_1 & ex_2 & 11 \times 1 \\ 11 \times 1 & 12 \times 1 & 11 \times 1 & 12 \times 1 \end{bmatrix}$ (bus, line)
 $\begin{bmatrix} s_simu, & C_wl, bus_sol \end{bmatrix}$
 $\begin{bmatrix} bus = bus_sol \\ 11 \times 1 \end{bmatrix}$

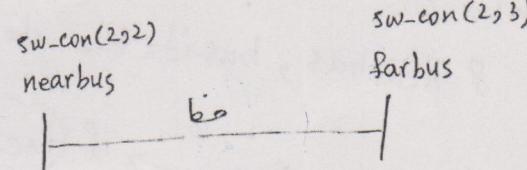
11×1 bus-intPrf ← موارد ممکن برای سیستم

۱۶۱ nbus = size (bus (:, 1)) ← تعداد بuss ها

bus-f = bus, line-f = line (میزبانی) : Fault $\begin{bmatrix} 11 \times 1 & 12 \times 1 \end{bmatrix}$ \rightarrow bus-f = bus, line-f = line (میزبانی)

۱۶۱ f-types sw-con (2, 4) → (V (1; 1) میزبانی، طبقه f-type)

(if) $f_type < f \leq f_type = V$
 ① $f_nearbus = sw_con(2, 2);$



else if $f_type = 1 \rightarrow$ خط فاصل

1-3
↳ $\left\{ \begin{array}{l} x_f = sw_con(2,4) \times sw_con(2,2) / [sw_con(2,4) + sw_con(2,2)] \\ x_f = \max(x_f, 1^{10}) \\ b_f = 1/x_f \end{array} \right.$

D-1

else if $f_type = 2 \rightarrow$ خط فاصل

1-4
↳ $\left\{ \begin{array}{l} x_f = sw_con(2,4) + sw_con(2,2) \\ x_f = \max(x_f, 1^{10}) \\ b_f = 1/x_f \end{array} \right.$

else if $f_type = 3 \rightarrow$ خط فاصل

1-2
↳ $\left\{ \begin{array}{l} x_f = sw_con(2,4) \\ x_f = \max(x_f, 1^{10}) \\ b_f = 1/x_f \end{array} \right.$

میں b_f کو 0 کے لئے $\Rightarrow bus_f(bus_idx, 9) = -b_f$

if $f_type = 4 \rightarrow$

② $f_nearbus, busidx \leftarrow$ جو جو $f_farbus = sw_con(2,3)$

لیکن $line_idx = k$ $\left\{ \begin{array}{l} \text{if } line_f(:,1)_k = f_nearbus, line_f(:,2)_k = f_farbus \\ \text{میں اس کو } line_f(:,1)_k \text{ کو } line_f(:,2)_k \text{ کے لئے} \\ \text{if } line_f(:,1)_k = f_farbus, line_f(:,2)_k = f_nearbus \end{array} \right.$

لیکن $line_idx \leftarrow line_idx$ $\left\{ \begin{array}{l} \text{if } 2-1 \\ \text{if } line_f(line_idx(1), t) = 10^V \end{array} \right.$

$line_f(line_idx(1), t) = 10^V$

line_idx کو اس کے لئے
جو جو

if f type = 0 \rightarrow (زدی قشنگ نه)

$\hookrightarrow f$ -nearbus, bus-idx ماتریس

if f type = 0 \rightarrow (زدی قشنگ نه) \leftarrow bus-idx اول است
 \leftarrow bus-f(bus-idx(1), 4) = 0 \leftarrow (زدی قشنگ نه) \leftarrow (0, 0, 0, 0) = 0

if f type = 4 \rightarrow no fault

$\hookrightarrow f$ -nearbus, bus-idx ماتریس

if f type = 4 \rightarrow (زدی قشنگ نه) \leftarrow bus-idx اول است
 \leftarrow bus-f(bus-idx(1), 4) = 0 \leftarrow (زدی قشنگ نه) \leftarrow (0, 0, 0, 0) = 0

red-ybus \hookrightarrow [Y-gf, Y-gncf, Y-nclf, Y-ncf, V-rgf, V-rncf, hof] = red-ybus(bus-f, line-f)

bus-intf \rightarrow (زدی قشنگ نه)

: post-fault (8)

if f type < 4

\hookrightarrow f -nearbus \rightarrow (زدی قشنگ نه)

f -farbus \rightarrow sw-wn(2, 3)

line-PF1 = line, bus-PF1 = bus (پلیس)

line-idx \rightarrow (زدی قشنگ نه)

f -farbus \rightarrow f -nearbus (زدی قشنگ نه) \leftarrow bus-idx

line-PF1(line-idx(1), f) = 1 \rightarrow

(زدی قشنگ نه) را می‌شناسیم

new-bus = 10 + (زدی قشنگ نه) \rightarrow (21)

max-PF1b = 1 + (زدی قشنگ نه) \rightarrow (12) \rightarrow (زدی قشنگ نه) bus-PF1

برینج سود و bus-Pf1 می باشند (12) max-Pf1b درینجا که داشتیم
 $bus-Pf1 \left(max-Pf1b, 12 \right)$ $\left\{ \begin{array}{l} \text{سوناول} = new_bus \\ \text{و} = 1 \\ \psi-v = 0 \\ q = bus-f \left(bus-idx(1), 9 \right) \\ 10 = \psi \\ \psi = 0 \quad (\text{خود راه صفر}) \end{array} \right.$

Δ-E

$$dPf1 = 1 + \frac{\text{لطفاً}}{\text{لطفاً}} \quad (13) \rightarrow \text{line-Pf1}$$

$\text{line-Pf1} \left(dPf1, 13 \right)$ $\left\{ \begin{array}{l} \text{سوناول} = new_bus \\ 2 = f-farbus \\ 3-4 = \text{line} \left(line-idx(4), \frac{4}{4} \right) \end{array} \right.$ $\text{لطفاً} : line-idx = \left[\frac{0}{4} \right]$

لطفاً \rightarrow bus-Pf1, line-Pf1
 red-Ybus \rightarrow چیزی: Y-gpf1, Y-gncpf1, Y-ncgpf1, Y-ncpf1, V-rgpf1, V-rncpf1, hopf1
 $bus-intPf1 \rightarrow$ چیزی که نوشتم

④ elseif f-type <= t & t <= 9

bus-Pf1 = bus-f

line-Pf1 = line-f

Y-gpf1 = Y-gf

Y-gncpf1 = Y-gncf

Y-ncgpf1 = Y-ncgf

Y-ncpf1 = Y-ncf

V-rgpf1 = V-rgf

V-rncpf1 = V-rncf

bus-intpf1 = bus-intf

hopf1 = hopf

⑤ elseif f-type > v

bus-Pf1 = bus

line-Pf1 = line

Y-gpf1 = Y-gprf

Y-gncpf1 = Y-gncprf

Y-ncgpf1 = Y-ncgprf

Y-ncpf1 = Y-ncprf

V-rgpf1 = V-rgprf

V-rncpf1 = V-rncprf

bus-intpf1 = bus-intprf

hopf1 = hoprf

Ⓐ if f-type < 4

clear-fault (2)

$$\text{line-Pf2} = \text{line-Pf1}$$

$$\text{line-Pf2} (\text{dR Pf1}, \epsilon) = 10^V \rightarrow \text{bus-Pf2}$$

$$\text{bus-Pf2} = \text{bus-Pf1}$$

D-F

$$\text{bus-Pf2} (\text{max-Pf1b}, \eta) = 0$$

$$\begin{array}{l} \text{bus-Pf2, line-Pf2} \\ \text{bus-Pf2, Y-gPf2, Y-gncPf2, Y-ncPf2, V-rgPf2, V-rncPf2, hopf2} \end{array}$$

Ⓐ Conditions :

$$\text{bus-Pf2} = \text{bus-Pf1}$$

$$\text{line-Pf2} = \text{line-Pf1}$$

$$Y_{-g,Pf2} = Y_{-g,Pf1}$$

$$Y_{-gncPf2} = Y_{-gncPf1}$$

$$Y_{-ncPf2} = Y_{-ncPf1}$$

$$Y_{-ncPf2} = Y_{-ncPf1}$$

$$V_{-rgPf2} = V_{-rgPf1}$$

$$V_{-rncPf2} = V_{-rncPf1}$$

$$\text{bus-intPf2} = \text{bus-intPf1}$$

$$\text{hopf2} = \text{hopf1}$$

دعا

* خوب ندارد. از اطلاعات طبعی مارس SW-con y-switch یعنی

، ما رسن (ادعیان) کسی نیست اینطور که ایسا وسیع زیر را سئول میدارد.

* sw-con میں از ما رس ہائین میں تعریف شدہ ہے کارڈر.

الطبقات: pre-fault, fault - immediate post-fault - final fault e.ear

الف) ساقن ملک (میسر) برای اینجا

$\text{load_con} = \text{size}(\text{load_con}(:, 1))$ ← داده‌های مورد نظر را بازگشایی کرده و آنرا به load_con می‌دانیم.

$$\text{red-ybus} \begin{bmatrix} Y_{gPrf}, Y_{gnePrf}, Y_{negPrf}, Y_{ncPrf}, V_{rgPrf}, V_{rnePrf}, b_0Prf \end{bmatrix} = \text{red-ybus} \dots$$

$$\begin{array}{ccccccccc} Y_{11} & Y_{12} & Y_{21} & Y_{22} & \text{rec-v1} & \text{rec-v2} & \text{bus-order} \\ \text{Ex}\Sigma & \text{Ex}\Sigma & \text{xx}\Sigma & \text{xx}\Sigma & \text{q}\times\Sigma & \text{q}\times\Sigma & \text{11x1} \\ (\text{bus}) & & & & & & & \dots & \text{(bus, line)} \\ & & & & & & & & \downarrow \\ & & & & & & & & \text{12x10} \\ & & & & & & & & \text{Q10} \end{array}$$

سازمانهای اسلامی ← ۱۱x۱۰ ۱۱x۱۰

jei nbus = size(bus(:,:1)) ← lastukščiai

bus-f=bus, line-f=line (pinie) : fault (voltage) \rightarrow becomes closer (-)
11x10 12x10

پہلی F-types SW-con(2,4) → (VC1;1) ورچیوئل، ٹیکسٹ F-type

(if) $\exists f$ $f\text{-types} \subset f \subseteq f\text{-type} = v$

sw-con(2)2
nearbus

sw-con(2,3)
farbus

① $\text{yel1f_nearbus_sw_con}(2,2)$;

bio

رسنون اول bus-f
درای اول bus-f
کیا اول bus-f
کو اول f-nearbus

سماں میں ماس کو روکنے کا خدا (عما) نے لئے
(سماں میں ماس کو روکنے کا خدا (عما) نے لئے)

کامپیوⁿ درایو از سهون \rightarrow
اول bus-f

اس کے ترتیب میں عالمی رہنمائی کا انتظامی طبقہ

لهم صلاة وسليمة على مختارين آمن العقول (فتاوى) من مسلمان سنة انت.

(if) 1-2 $\frac{1}{n}$ f-type = $\frac{1}{n} \leq v$ \leftarrow $v \leq \frac{1}{n} \sum_{i=1}^n w_i$ (عکس از مسئله ۱۰)

$$\hookrightarrow \text{bf} = 10^V$$

٤٤-١ اور فوجی نام کار، فوجی بسیں ہا
راہ میں سب مارچنے لگا گا
موجہ موجہ بسیں ہوں گا۔

4

elseif $f_type = 1 \rightarrow$ خطان فائز زیری

1-3

$$\left\{ \begin{array}{l} xf = sw_con(2,4) \times sw_con(2,2) / [sw_con(2,4) + sw_con(2,2)] \\ xf = \max(xf, 10^{-5}) \\ bf = 1/xf \end{array} \right.$$

elseif $f_type = 2 \rightarrow$ خطان فائز بالآخر

1-4

$$\left\{ \begin{array}{l} xf = sw_con(2,4) + sw_con(2,2) \\ xf = \max(xf, 10^{-5}) \\ bf = 1/xf \end{array} \right.$$

elseif $f_type = 3 \rightarrow$ خطان بالآخر

1-0

$$\left\{ \begin{array}{l} xf = sw_con(2,4) \\ xf = \max(xf, 10^{-5}) \\ bf = 1/xf \end{array} \right.$$

میله f عنوانی $\Rightarrow bus-f(bus_idx, q) = -bf$

if $f_type = 4 \rightarrow$ خطان بالآخر

② $f_nearbus, bus_idx \leftarrow$ خطان بالآخر $, f_farbus = sw_con(2,3)$

$line_idx = k$ \leftarrow خطان بالآخر

if $line-f(:,1)_k = f_nearbus \rightarrow line-f(:,2)_k = f_farbus$

if $line-f(:,1)_k = f_farbus \rightarrow line-f(:,2)_k = f_nearbus$

خطان بالآخر $\leftarrow line_idx$

$line-f(line_idx(1), t) = 10^V$ \leftarrow خطان بالآخر

خطان بالآخر \leftarrow خطان بالآخر

خطان بالآخر \leftarrow خطان بالآخر

if $f_type = 0 \rightarrow$ از دست رفتن ندار

$\hookrightarrow f_nearbus, bus_idx$ مادر قبل

اگر bus_idx معین نباشد \leftarrow if bus_idx آن است

اگر bus_idx بزرگتر از 4 باشد \leftarrow bus-f $(bus_idx(1), 4) = 0$ درین صورت مورد توجه است
درین صورت مادر اول است $\leftarrow (0, v) = 0$

اگر $f_type = 4 \rightarrow$ no fault

$\hookrightarrow f_nearbus, bus_idx$ مادر قبل

هر چیزی که $f_type = 4$ است *

هر $f_type = 4$ ای (AVR مجموعه ای از مادر و فرزند)

اگر bus_idx معین نباشد \leftarrow if f_type درین صورت مادر است

برای red_ybus $\hookrightarrow [Y_gf, Y_gncf, Y_ncgf, Y_ncf, V_rgf, V_rncf, hof] = red_ybus(bus_f, line_f)$

bus_intf \rightarrow مادر و فرزند

: POST-fault (P.
برای خطا)

if $f_type < 4$

$\hookrightarrow f_nearbus \rightarrow$ از قبل

$f_farbus \rightarrow sw_on(2, 3)$

$line_pf1 = line$, $bus_pf1 = bus$ (پلیس)

$line_idx \rightarrow$ مادر قبل

$f_farbus > f_nearbus$ \leftarrow line-idx \leftarrow if $line_idx$

$line_pf1(line_idx(1), f) = 1^v \rightarrow$

برای $line_idx$ را می‌سازد

اگر ترتیب خارجی (0, 1, 2, ..., 11) خواهد بود.

$new_bus = 10 +$ (21)

$max_pf1b = 1 +$ (12) \rightarrow

برای $line_idx$ مادرین bus-pf1

١٢) معايير bus-Pf1 ريف سطحه: bus-Pf1b

$$\text{bus-Pf1} \left(\text{max-Pf1b}, 12 \right) \quad \left\{ \begin{array}{l} \text{سون اول} = \text{new_bus} \\ \text{bus_idx} = 1 \\ \psi_v = 0 \\ q = \text{bus-f}(\text{bus-idx}(1), 9) \\ 10 = \psi \\ \text{new} = 0 \quad (\text{مودعه صفر}) \end{array} \right.$$

مقابل (ابعاد معمولة) معايير bus-Pf1b

$$\Delta Pf1 = 1 + \frac{\Delta \psi}{\Delta \psi} = (13) \rightarrow \text{line-Pf1} \quad \text{برای (معایر ۱۲) ریف سطحه}$$

$$\text{line-Pf1} \left(\Delta Pf1, 13 \right) \quad \left\{ \begin{array}{l} \text{سون اول} = \text{new_bus} = 1 \\ 2 = f_farbus \\ 3-4 = \text{line}(\text{line-idx}(1), \frac{13}{4}) \end{array} \right.$$

مقابل (ابعاد معمولة) معايير line-Pf1

ذکوازی red-thus \rightarrow bus-Pf1, line-Pf1
 \rightarrow bus: Y-gpf1, Y-gncpf1, Y-ncgpf1, Y-ncpf1, V-rgpf1, V-rncpf1, hopf1
 \rightarrow bus-intPf1 \rightarrow bus-intpf1

④ elseif f-type \leq t \leq 9

$$\text{bus-Pf1} = \text{bus-f}$$

$$\text{line-Pf1} = \text{line-f}$$

$$Y-gpf1 = Y-gf$$

$$Y-gncpf1 = Y-gncf$$

$$Y-ncgpf1 = Y-ncgf$$

$$Y-ncpf1 = Y-ncf$$

$$V-rgpf1 = V-rgf$$

$$V-rncpf1 = V-rncf$$

$$\text{bus-intpf1} = \text{bus-intf}$$

$$\text{hopf1} = \text{hopf}$$

⑤ elseif f-type $>$ 9

$$\text{bus-Pf1} = \text{bus}$$

$$\text{line-Pf1} = \text{line}$$

$$Y-gpf1 = Y-gPrf$$

$$Y-gncpf1 = Y-gncPrf$$

$$Y-ncgpf1 = Y-ncgPrf$$

$$Y-ncpf1 = Y-ncPrf$$

$$V-rgpf1 = V-rgPrf$$

$$V-rncpf1 = V-rncPrf$$

$$\text{bus-intpf1} = \text{bus-intPrf}$$

$$\text{hopf1} = \text{hoprf}$$

4-1

ج ۱ : نیروگاه داده خواهد - سرعت مقدار (مقدار) میسر میشود

10 \leftarrow $\text{نیروگاه} (r_1)$ مدل ۱ \leftarrow آفر مدل ۱ $\left\{ \begin{array}{l} \text{برای} \text{تغیر} \text{مقدار} \text{که} \text{بزرگ} \text{تر} \text{می} \text{باشد} \\ \text{ل} \leftarrow \text{تعداد} \text{اطرها} \geq \text{تعداد} \text{ماشین} \text{ها} \quad \text{و} \text{تعداد} \text{ستون} \text{ها} \end{array} \right.$

13 \leftarrow $\text{نیروگاه} (r_2)$ " " $\left\{ \begin{array}{l} \text{برای} \text{تغیر} \text{مقدار} \text{که} \text{بزرگ} \text{تر} \text{می} \text{باشد} \\ \text{برای} \text{تغیر} \text{مقدار} \text{که} \text{بزرگ} \text{تر} \text{می} \text{باشد} \end{array} \right.$

tg برابر با tg_{hydro} باشد $\left[\begin{array}{l} \text{برای} \text{تغیر} \text{مقدار} \text{که} \text{بزرگ} \text{تر} \text{می} \text{باشد} \\ \text{برای} \text{تغیر} \text{مقدار} \text{که} \text{بزرگ} \text{تر} \text{می} \text{باشد} \end{array} \right]$

برای تغیر مقدار اول سرور tg_{con} میسر میشود

معون

1: $\text{کار} \text{کرد} \text{تغیر} \text{مقدار} \text{که} \text{بزرگ} \text{تر} \text{می} \text{باشد}$ ($1 = \text{آفر مدل}$, $2 = \text{آفر مدل}$)

2: $\text{تغیر} \text{مقدار} \text{که} \text{بزرگ} \text{تر} \text{می} \text{باشد}$

3: speed set point w_f

4: steady state gain k_R

5: maximum power order T_{\max}

6: servo time constant T_S

7: governor " " T_G

8: transient gain time constant T_3

9: HP section " " T_4

10: reheater " " T_5

معون اول و دوم و سوم : $\text{تغیر} \text{مقدار} \text{که} \text{بزرگ} \text{تر} \text{می} \text{باشد}$

4: permanent droop R_P

10: servo gain k_S

5: transient " " R_t

11: governor time constant tg

6: T_{\max}

12: reset " " T_r

7: max rate limit

13: water starting time T_w

8: min " "

9: T_S

tg - idn $\mathcal{E}^{\mathbb{C}}$

دھوکہ میڈیا سینٹر میں ایک ایجاد کی جائے گی۔ اس ایجاد کا نام ایک ایجاد کی جائے گی۔

$$\left(\operatorname{tg} \text{-con} (0, 1) = 1 \right) \text{وارد} \xrightarrow{\text{با} \frac{1}{\operatorname{tg} x}} \sqrt{\operatorname{tg} \text{-con}} \text{ از مارکولک} = \overbrace{\operatorname{tg} - \operatorname{id}_X, \text{برای}}^{\text{برای}} \quad \text{برای} \quad \text{برای}$$

$$\text{لما } n - tg = 0 \Leftrightarrow \left\{ \begin{array}{l} \text{أولاً: } n = tg \\ \text{ثانياً: } tg = n \end{array} \right.$$

$$\text{tgh-idx}_{\text{b}_j \text{ مول}} = n - \text{tgh} : \text{نحوه}$$

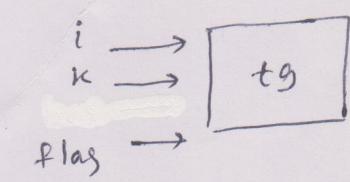
tg-con , $n\text{-tgh}$, tgh-idx , $n\text{-tg}$, tg-idx : gicols global \hookrightarrow cuso *

١- $n-tg$ \leftarrow انتل - شهد توین - کورسیا مارل

$\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{32}$

مذکورین ترین - طور مراری (عمل مبارہ ۱) : tg tg

* خریداری - خریداری پیشنهادی



i : مبارہ ترکیب

k : integer time

flag : ۰, ۱, ۲

موضع در خریداری

۵-Simul

لیست فردا خریداری

$$\underline{n} = \begin{bmatrix} \text{گزینه} \\ \text{tg-con} \\ \text{بتریب} \end{bmatrix}$$

خریداری
گزینه
طبق شرط

عمل ۱ خریداری - طور مراری (عمل ۱ خریداری)

$\text{if } (\underline{\text{Pmech}}(n, 1) > \text{tg-con}(i, 5)) : \Rightarrow$

پیام خطای عبور از حد مالی

$\text{if } (" < ") = \text{previous} = " "$

پیام خطای عبور از حد مالی با وظیفه Pmech نیست

$\text{tg}_1(\text{tg-idx}, 1) = \text{Pmech}(\underline{n}, 1)$

$\text{tg-pot}(" , 1) = \text{tg-con}(\text{tg-idx}, 1) / \text{tg-con}(\text{tg-idx}, 7)$

چند

$a_1 = \begin{bmatrix} 1 \\ 1 \\ i \\ \vdots \\ \vdots \\ \text{tg-idx} \times 1 \end{bmatrix} - \text{tg-pot}(\text{tg-idx}, 1)$

$\text{tg-pot}(\text{tg-idx}, 2) = a_1$

$\text{tg}_2(\text{tg-idx}, 1) = a_1 \cdot \text{Pmech}(n, k)$

$\text{tg-pot}(" , 3) = \text{tg-con}(\text{tg-idx}, 9) / \text{tg-con}(\text{tg-idx}, 1)$

$a_2 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ \vdots \\ \text{tg-idx} \times 3 \end{bmatrix} - \text{tg-pot}(\text{tg-idx}, 3)$

$$tg_Pot(tg_idx, f) = a_{12}$$

$$tg3(tg_idx, 1) = a_{13} \cdot P_{mech}(n, k)$$

↓↓↓↓

$$tg_Pot(tg_idx, 5) = P_{mech}(n, k)$$

$$tg_Sig(\quad , 1) = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}_{n-tg \times 1}$$

network interface computation \Leftarrow flag = 1 !!

~~Pmech~~

if $(n-tg) \neq 0 \{$

\rightarrow ~~mac~~ \rightarrow P_{mech}

$$P_{mech}(n, k) = tg3(tg_idx, k) + tg_pot(\textcircled{23}) \cdot \left[\frac{tg2 + tg_pot(\textcircled{2}, 1)}{tg1(\textcircled{2}, k)} \right]$$

↓↓↓↓

↓↓↓↓

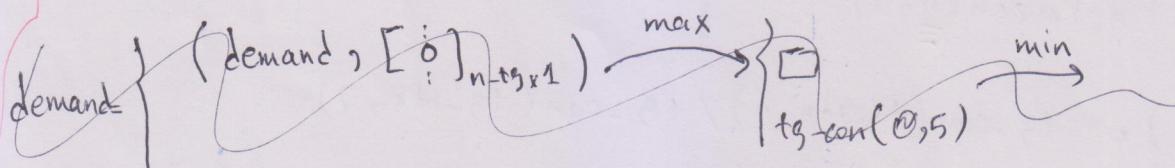
↓↓↓↓

↓↓↓↓

if $(n-tg) \neq 0 \{$

$$SPD_err = tg_con(\textcircled{2}, 3) - mac_spd(n, k)$$

$$\text{demand} = tg_Pot(\textcircled{2}, 5) + SPD_err \cdot tg_con(\textcircled{2}, 4) + tg_sig(tg_idx, k)$$



$$\text{demand} = 0 \leftarrow \text{demand} \leq tg_con(tg_idx, s) \quad ; \text{flag} = 2$$

$$dtg1(\textcircled{2}, k) = (\text{demand} - tg1(\textcircled{2}, k)) / tg_con(tg_idx, 4)$$

$$dtg2(n) = \left\{ tg1(tg_idx, k) \cdot tg_pot(\textcircled{2}, 2) - tg2(\textcircled{2}, k) \right\} / tg_con(\textcircled{2}, 7)$$

$$dtg3(n) = \left[\left(tg2(\textcircled{2}, k) + tg_pot(\textcircled{2}, 7) \cdot tg1(\textcircled{2}, k) \right) \cdot tg_pot(\textcircled{2}, 5) \right. \\ \left. - tg3(\textcircled{2}, k) \right] / tg_con(\textcircled{2}, 10)$$

~~Pmech~~

$$P_{mech}(n, k) =$$

```
function f = exc_st3(i,k,flag)
```

هدف: مدلسازی سیستم تحریک (AVR ST3)

```
global cur_re cur_im bus_int
global mac_con mac_pot mac_int
global mac_ang eqprime edprime psikd psikq
global curd curq curdg curqg fldcur
global psidpp psiqpp vex eterm theta ed eq
global pmech pelect qselect-
global mac_tra_idx mac_sub_idx n_tra n_sub
global exc_con exc_pot Efd V_R V_A V_As R_f V_TR V_B
global Efd dEfd dV_R dV_As dR_f dV_TR
global exc_sig st3_idx n_st3
global st3 TB idx st3 noTB idx st3 TR idx st3 noTR idx pss out
```

```
if n_st3~=0  
    if flag == 0; % initialization
```

```

if (n_sub==0) && (n_tra==0)
    disp('in! more tra in')
end

```

```
if i == 0; % vectorized
```

~~V_E = zeros(n_st3,1); I_N = V_E; iterm = V_E; vep = V_E; ve = V_E; F_EX = V_E;~~

`n = mac_int(exc_con(st3_idx, 2));` → *لما ندخل st3_idx في exc_con*
`n_bus = bus_int(mac_con(n, 2));`
`Efd(st3_idx, 1) = user[n];`

Efd(st3_idx,1) = vex(n,1);

```
max_lim = find(Efd(st3_idx,1) > exc_con(st3_idx,18));
```

if ~isempty(max_lim) \rightarrow max-lim

disp('Efd exceeds maximum in initialization')
end

`exc pot(st3 idx,1) = exc con(st3 idx,13).*cos(exc con(st3 idx,14)*pi/180)`

exc_pot(st3_idx,2) = exc_con(st3_idx,13).*sin(exc_con(st3_idx,14)*pi/180)

iterm = (pelect(n,1)-jay*qelect(n,1))./(eterm(n,1).*exp(-jay*theta(n_bus,1))).*mac_pot(n,1);

```
vep = eterm(n,1).*exp(jay*theta(n-bus,1)).*(exc_pot(st3_idx,1) + jay*exc_pot(st3_idx,2));
```

ve = vep+jay*(exc_con(st3_idx,15)...

```
+ (exc_pot(st3_idx,1))
V_E = abs(ve);
ve_low = find(V_E < 1e-6);
```

if ~isempty(ve_low)

```
    disp('excitation system error: no supply voltage at the following')
```

end

I_N = exc_con(st3_idx,17).*fldcur(n,1)./V_E;

low_IN=find(I_N < 0.433);

```
if ~isempty(low_IN)
```

```
F_EX(low_IN) = ones(length(low_IN),1)-0.5771*I_N(low_IN);
```

end

```
big IN=find(I N > 0.75);
```

if ~isempty(big_IN)

```
F_EX(big_IN) = 1.708*(ones(length(big_IN),1)-I_N(big_IN));
```

end

$$1.708 \times \left(\begin{bmatrix} & \\ \vdots & \\ & 1 \end{bmatrix} - \begin{bmatrix} I - N \\ & \end{bmatrix}_{\text{big } -IN \times 1} \right)_{\text{big } -IN \times 1}$$

big-IN , mid-IN , low-IN / begins [I-N] بجزء

```

mid_IN = find((I_N > 0.433) & (I_N < 0.75));
if ~isempty(mid_IN)
    F_EX(mid_IN) = [(0.75*ones(length(mid_IN),1)-(I_N(mid_IN)).^2).^0.5];
end

```

مبدأ 2 كل جزء

```

fex_error = find(F_EX<=0);
if ~isempty(fex_error)
    disp('error: F_EX zero or negative at the following generators')
end

```

↑ ←

```

V_B(st3_idx,1) = V_E.*F_EX;
V_R(st3_idx,1) = Efd(st3_idx,1)./V_B(st3_idx,1);

```

```

nKA_idx = find(exc_con(st3_idx,4) == 0);
if ~isempty(nKA_idx)
    exc_con(st3_idx(nKA_idx),4) = ones(length(nKA_idx),1); % reset to 1
end

```

↑ minimum (a,b)

```

V_A(st3_idx,1) = V_R(st3_idx,1)./exc_con(st3_idx,4)+min(...,
    exc_con(st3_idx,20),exc_con(st3_idx,19).*Efd(st3_idx(1)));
V_As(st3_idx,1) = V_A(st3_idx,1);
exc_pot(st3_idx,5) = ones(n_st3,1);

```

[]_{n_st3x1}

```

TB = st3_TB_idx;
if ~isempty(TB)
    exc_pot(st3_idx(TB),5) = exc_con(st3_idx(TB),7)./exc_con(st3_idx(TB),6);
end

```

↑

```

V_I = V_A(st3_idx,1)./exc_con(st3_idx,12);
max_VI=find(V_I > exc_con(st3_idx,10));
if ~isempty(max_VI)
    disp('V_I above maximum in initialization at')
end

```

```

min_VI = find(V_I < exc_con(st3_idx,11));
if ~isempty(min_VI)
    disp('V_I below minimum in initialization at')
end

```

```

exc_pot(st3_idx,3) = eterm(n,1)+V_I;
V_TR(st3_idx,1) = eterm(n,1);
R_f(st3_idx,1) = zeros(n_st3,1);
end %end initialization

```



```

    } eterm(n,k) = sqrt(ed(n,k).^2+eq(n,k).^2);
    pelect(n,k) = eq(n,k).*curq(n,k) + ed(n,k).*curd(n,k);
    qelect(n,k) = eq(n,k).*curd(n,k) - ed(n,k).*curq(n,k);

    item = (pelect(n,k)-jay*qelect(n,k))./...
        (eterm(n,k).*exp(-jay*theta(n_bus,k))).*mac_pot(n,1);

    vep = eterm(n,k).*exp(jay*theta(n_bus,k)).*(exc_pot(st3_idx,1) + jay*exc_pot(st3_idx,2));

    ve = vep+jay*(exc_con(st3_idx,15)...
        + (exc_pot(st3_idx,1)+jay*exc_pot(st3_idx,2)).*exc_con(st3_idx,16)).*item;

    V_E = abs(ve);

    ve_low=find(V_E < 1e-6);
    ve_norm= find(V_E >= 1e-6);

    if ~isempty(ve_norm)
        n_ven = length(ve_norm);
        n_norm = n(ve_norm);
        I_N(ve_norm) = exc_con(st3_idx(ve_norm),17).*fldcur(n_norm,k)./V_E(ve_norm);
    end

    if ~isempty(ve_low)
        disp('excitation system error: no supply voltage at')
        I_N(ve_low) = 2*ones(length(ve_low),1);
    end

    low_IN=find(I_N < 0.433);
    if ~isempty(low_IN)
        F_EX(low_IN) = ones(length(low_IN),1)-0.5771*I_N(low_IN);
    end

    big_IN=find(I_N > 0.75);
    if ~isempty(big_IN)
        bigl=length(big_IN);
        F_EX(big_IN) = 1.732*(ones(bigl,1)-I_N(big_IN));
        F_EX(big_IN)=max(F_EX(big_IN),zeros(bigl,1));
    end

    mid_IN =find((I_N > 0.433) & (I_N < 0.75));
    if ~isempty(mid_IN)
        F_EX(mid_IN) = (0.75*ones(length(mid_IN),1)-(I_N(mid_IN)).^2).^0.5;
    end

    V_B(st3_idx,k) = V_E.*F_EX;
    Efd(st3_idx,k) = min(V_R(st3_idx,k).*V_B(st3_idx,k),exc_con(st3_idx,18));
    vex(n,k) = Efd(st3_idx,k);
end
end %end interface

```

```

if flag == 2 % exciter dynamics calculation
    if i == 0 % vector calculation
        n = mac_int(exc_con(st3_idx,2));
        no_TR = st3_noTR_idx;

        if ~isempty(no_TR)
            n_nTR = n(no_TR);
            dV_TR(st3_idx(no_TR),k)=zeros(length(no_TR),1);
            V_TR(st3_idx(no_TR),k)=eterm(n_nTR,k);
        end

        TR = st3_TR_idx;
        if ~isempty(TR)
            n_TR = mac_int(exc_con(st3_idx(TR),2));
            dV_TR(st3_idx(TR),k)=(eterm(n_TR,k)-V_TR(st3_idx(TR),k))...
                ./exc_con(st3_idx(TR),3);
        end

        V_I = exc_sig(st3_idx,k) + exc_pot(st3_idx,3) - V_TR(st3_idx,k);
        V_I = V_I + pss_out(st3_idx,k);
        V_I = min(exc_con(st3_idx,10),max(V_I,exc_con(st3_idx,11)));
        V_I = exc_cn 11 <= V_I <= exc_cn 10

        no_TB = st3_noTB_idx;
        if ~isempty(no_TB)
            dV_As(st3_idx(no_TB),k) = zeros(length(no_TB),1);
            V_As(st3_idx(no_TB),k) = exc_con(st3_idx(no_TB),12).*V_I(no_TB);
            V_A(st3_idx(no_TB),k) = V_As(st3_idx(no_TB),k);
        end

        TB = st3_TB_idx;
        if ~isempty(TB)
            dV_As(st3_idx(TB),k) = (-V_As(st3_idx(TB),k)+exc_con(st3_idx(TB),12)...
                .*V_I(TB))./exc_con(st3_idx(TB),6);
            V_A(st3_idx(TB),k) = exc_pot(st3_idx(TB),5).*exc_con(st3_idx(TB),12)...
                .*V_I(TB) + (ones(length(TB),1)-...
                exc_pot(st3_idx(TB),5)).*V_As(st3_idx(TB),k);
        end

        dV_R(st3_idx,k) = (-V_R(st3_idx,k)+exc_con(st3_idx,4).*V_A(st3_idx,k)...
            -min(exc_con(st3_idx,20),exc_con(st3_idx,19)...  

            .*Efd(st3_idx,k)))./exc_con(st3_idx,5);
    end

```

7

```

max_lim=find(V_R(st3_idx,k) > exc_con(st3_idx,8));
if ~isempty(max_lim)
    V_R(st3_idx(max_lim),k) = exc_con(st3_idx(max_lim),8);
    pos_rate = find(dV_R(st3_idx(max_lim),k)>0);
    n_pos = length(pos_rate);

    if n_pos~=0
        dV_R(st3_idx(max_lim(pos_rate)),k) = zeros(n_pos,1);
    end

end

min_lim=find(V_R(st3_idx,k) < exc_con(st3_idx,9));
if ~isempty(min_lim)
    V_R(st3_idx(min_lim),k) = exc_con(st3_idx(min_lim),9);
    neg_rate = find(dV_R(st3_idx(min_lim),k)<0);
    n_neg = length(neg_rate);

    if n_neg~=0
        dV_R(st3_idx(min_lim(neg_rate)),k) = zeros(n_neg,1);
    end

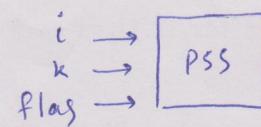
end

dEfd(st3_idx,k) = zeros(n_st3,1);
dR_f(st3_idx,k) = zeros(n_st3,1);
end % end rate calculation
end

```

function: pss(i,k,flag)

% مدلسازی پایدارساز سیستم قدرت (PSS)
% شماره ژنراتور - i : ورودی
% k - integer time
% flag - 0 - initialization
% 1 - network interface computation
% 2 - system dynamics computation
global pss_con pss_pot pss_mb_idx pss_exc_idx
global pss1 pss2 pss3 dpss1 dpss2 dpss3 pss_out
global pss_idx n_pss pss_sp_idx pss_p_idx;
global pss_T pss_T2 pss_T4 pss_T4_idx pss_noT4_idx;
global mac_con mac_int mac_spd pselect basmva



* فرآیند

{ جدول

if n_pss~=0 *جشع pss-index*
 if flag == 0; % initialization
 if i == 0; % vector computation
 pss_pot=ones(n_pss,2); *عیناً اول ماترس*
 n=pss_mb_idx; *شروع*
 if ~isempty(pss_sp_idx) *هر کدام pss-sp-idx*
 n_sp = mac_int(pss_con(pss_sp_idx,2)); *ضرف*
 pss1(pss_sp_idx,1)=mac_spd(n_sp,1);
 end *کاری شده در اینجا*
 if ~isempty(pss_p_idx)
 n_p = mac_int(pss_con(pss_p_idx,2));
 pss1(pss_p_idx,1)=pselect(n_p,1)*basmva./mac_con(n_p,3);
 end *اب پس دل / دل*
 pss2(pss_idx,1)=zeros(n-pss,1); *لارفع محتوا*
 pss3(pss_idx,1)=zeros(n_pss,1);
 pss_out(pss_exc_idx,1)=zeros(n_pss,1);
 pss_pot(:,1)=pss_con(:,5)./pss_con(:,6); *شون دوم شده*
 if ~isempty(pss_T4_idx)
 pss_pot(pss_T4_idx,2)=pss_con(pss_T4_idx,7)./pss_T4(pss_T4_idx);
 end *بردازید شد*
 end *pss-con + 0 شد*
 end *تقریباً داری*
* if flag == 1 % network interface computation
 if i == 0 % vector computation
 if n_pss~=0
 n = pss_mb_idx; % machine number vector
 var1 = zeros(n_pss,1); var2 = var1; var3 = var1;
 if length(pss_sp_idx)~=0 *ستاره اولیه*
 n_sp = mac_int(pss_con(pss_sp_idx,2));
 var1(pss_sp_idx) = mac_spd(n_sp,k)- (pss1(pss_sp_idx,k)/pss_Gn(pss_sp_idx,4))
 end *پس از هر ضرف*
 if ~isempty(pss_p_idx)
 n_p = mac_int(pss_con(pss_p_idx,2));
 var1(pss_p_idx) = pselect(n_p,k)*basmva./mac_con(n_p,3)- (pss1(pss_p_idx,k)/pss_Gn(pss_p_idx,4))
 end *شون دوم شده*
 end *کاری شده*
 end *mac*

PSS : متریک پس
PSS-Gain :

 1 " : $\frac{1}{2}$ $\frac{\text{تغیر اول}}{\text{تغیر دوم}}$ } PSS

 2 " : $\frac{1}{2}$ $\frac{\text{تغیر دوم}}{\text{تغیر اول}}$

 3 " : PSS-Gain \times Gain washout

 4 " : \rightarrow T_{washout}

 5 " : \rightarrow T_c lead 1

 6 " : \rightarrow T_c lag 1

 7 " : \rightarrow T_c lead 2

 8 " : \rightarrow T_c lag 2

 9 + max out limit

 10 - min out limit

if (pss->cn)
حالی باشد
(عرضه باشد)

$$PSS_idx = \text{find} \left(PSS_cn(i, 1) == 1 \right)$$

این تابع برای پیدا کردن اولین مکانی که PSS_cn برابر با ۱ شود.

$n - pss = \text{length}(\text{pss}-\text{id}\times)$ ~~length~~
 $\text{pss}-\text{id}\times \text{jobs}$
 $(\text{len pss} \text{ jobs})$

$pss_mb_idx = \underbrace{\text{mac_int}}_{\substack{\text{mac_intf} \\ \text{CloudIntf}}} \left(\underbrace{\text{round}}_{\substack{\text{! intf, !} \\ \text{! intf, !}}} \left(\underbrace{pss_con \left(\dots, 2 \right)}_{\substack{\text{PSS-} \\ \text{Con}}} \right) \right)$

16
for

$$\text{for } n-\text{PSS} \quad I \neq 1$$

$$j_{\text{PSS}} = 1 : n - \text{PSS} \quad \left\{ \begin{array}{l} \text{PSS-exc-idx}(j_{\text{PSS}}) = \text{find}(\text{pss_con}(\emptyset, 2) \\ \text{exc_con}(\cdot, 2); \end{array} \right.$$

إذا تم العثور على pss في المجموعة المقابلة ل exc، فـ
 exc ينتمي إلى مجموعة pss

از طرف درود درست
 رنگ باشد ناز مکمل
 سرطانی از دست
 ملائکه خداوند

$$\text{if } n - pss = 1 \circ \text{ نیز ممکن است} \quad \left\{ \begin{array}{l} PSS-T = pss_Gn(\overline{pss_idx}, 4) \\ \end{array} \right.$$

$$PSS-T2 = pss_Gn(pss_idx, 6);$$

$$PSS-T4 = \sim \sim \sim \sim$$

$$r - r - idx = \text{Find}(PSS-T4 > 0.001)$$

$$r - not4 = \sim \sim \sim \sim < 0.001$$

$$PSS-sp-idx = \text{Find}(pss_Gn(pss_idx, 1) == 1);$$

$$PSS-p-idx = \text{Find}(pss_Gn(pss_idx, 1) == 2);$$

} end

else :

$$n - pss = 0;$$

end

```
function exc_indx()
```

% (ST3 AVR) گذاری سیستم تحریک

```
global exc_pot exc_con n_exc  
global st3_TA st3_TA_idx st3_noTA_idx st3_TB st3_TB_idx st3_noTB_idx;  
global st3_TR st3_TR_idx st3_noTR_idx;
```

اگر ماتریس exc-con مقدار ۰ داشته باشد این باید باشد

```
n_st3 = 0; n_exc = 0;  
if ~isempty(exc_con)  
    st3_idx = find(exc_con(:,1) == 3);  
    if ~isempty(st3_idx); n_st3 = length(st3_idx); end  
  
    if n_st3 ~= 0  
        % TA  
        st3_TA = exc_con(st3_idx,5);  
        st3_TA_idx = find(st3_TA >= 0.001);  
        st3_noTA_idx = find(st3_TA < 0.001);  
        % TB & TC  
        st3_TB = exc_con(st3_idx,6);  
        st3_TB_idx = find(st3_TB >= 0.001);  
        st3_noTB_idx = find(st3_TB < 0.001);  
        % TR  
        st3_TR = exc_con(st3_idx,3);  
        st3_TR_idx = find(st3_TR >= 0.001);  
        st3_noTR_idx = find(st3_TR < 0.001);  
    end  
  
    n_exc = n_st3;  
    exc_pot = zeros(n_exc,5);  
end
```

ماتریس مفروضه ایجاد شده

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$n_exc \times 5$

9

البيانات المطلوبة لمحرك المولد exc-en وبيانات المولد المدخل

exc-en & مولد

Table 3 ST3 Exciter Data Format

column	data	unit
1	exciter type	3 for ST3
2	machine number	
3	input filter time constant T_R	sec
4	voltage regulator gain K_A	
5	voltage regulator time constant T_A	sec
6	voltage regulator time constant T_B	sec
7	voltage regulator time constant T_C	sec
8	maximum voltage regulator output V_{Rmax}	pu
9	minimum voltage regulator output V_{Rmin}	pu
10	maximum internal signal V_{Imax}	pu
11	minimum internal signal V_{Imin}	pu
12	first state regulator gain K_J	
13	potential circuit gain coefficient K_P	
14	potential circuit phase angle qp	degrees
15	current circuit gain coefficient K_I	
16	potential source reactance X_L	pu
17	rectifier loading factor K_C	
18	maximum field voltage E_{fdmax}	pu
19	inner loop feedback constant K_G	
20	maximum inner loop voltage feedback V_{Gmax}	pu

```
function h_sol = i_simu(k, ks, k_inc, h, bus_sim, Y_g, Y_gnc, Y_ncg, Y_nc, rec_V1, rec_V2, bo)
```

% تشکیل متغیرهای رابط شبکه : هدف
% Inputs: k - the current time step
% ks - indicates the switching times
% k_inc - the number of time steps between switching points
% h vector of time steps
% bus_sim value of bus matrix at this switching time
% Y_g - reduced Y matrix for generators
% Y_gnc - mutual reduced Y generators-nc loads
% Y_ncg - mutual reduced Y nc loads generators
% Y_nc - reduced Y matrix nc loads
% rec_V1 - voltage recovery matrix generators
% rec_V2 - voltage recovery matrix nc loads
% bo bus order for this switching time
% Output: h_sol - the time step at this value of ks
% Called by: s_simu

```
global bus_v theta bus_int
global psi_re psi_im n_mac
global cur_re cur_im
global load_con nload
```

```
flag = 1;
jay = sqrt(-1);
psi = psi_re(:,k) + jay*psi_im(:,k);
int_volt = psi;
h_sol = h(ks);
n_bus = length(bus_sim(:,1));
cur = Y_g*int_volt; % جریان تزریقی به ماشین ها
b_v(bo(nload+1:n_bus),1) = rec_V1*int_volt; % بازسازی ولتاژ ماشین ها
```

```
فرم ماتریس
if nload==0
    if k==1;
        kk = k-1;
    else
        kk=k;
    end
    vnc = bus_v(bus_int(load_con(:,1)),kk); % initial value
* vnc = nc_load(bus_sim,flag,Y_nc,Y_ncg,int_volt,vnc,1e-6,k);
```

```
% set nc load voltages
b_v(bo(1:nload),1)=vnc;
b_v(bo(nload+1:n_bus),1) = b_v(bo(nload+1:n_bus),1)+rec_V2*vnc;
cur = cur + Y_gnc*vnc; % modify generator currents for nc loads
end
```

فرم ماتریس

```
bus_v(bus_int(bus_sim(:,1)),k) = b_v;
theta(bus_int(bus_sim(:,1)),k) = angle(b_v);
cur_re(:,k) = real(cur(1:n_mac));
cur_im(:,k) = imag(cur(1:n_mac)); % جریان های ژنراتورها
```

{ زیر داده های سیستم
 (k) نیز

① if f-type < 4

clear-fault (\rightarrow)

$$\text{line-Pf2} = \text{line-Pf1}$$

$$\text{line-Pf2} (\text{dRPF1}, \epsilon) = 10^V \rightarrow \text{bus(V)}$$

$$\text{bus-Pf2} = \text{bus-Pf1}$$

$$\text{bus-Pf2} (\text{max-Pf1b}, q) = 0$$

$$\begin{array}{l} \text{bus(V)} \\ \text{bus-Pf2} \\ \text{bus(V)} \\ \text{bus(V)} \end{array} \Rightarrow \begin{array}{l} \text{bus(V), bus-Pf2, line-Pf2} \\ \text{Y-gPf2, Y-gncPf2, Y-ncPf2, V-rgPf2, V-rncPf2, hopf2} \end{array}$$

② Conditions :

$$\text{bus-Pf2} = \text{bus-Pf1}$$

$$\text{line-Pf2} = \text{line-Pf1}$$

$$Y_{-gPf2} = Y_{-gPf1}$$

$$Y_{-gncPf2} = Y_{-gncPf1}$$

$$Y_{-ncPf2} = Y_{-ncPf1}$$

$$V_{-rgPf2} = V_{-rgPf1}$$

$$V_{-rncPf2} = V_{-rncPf1}$$

$$\text{bus-intPf2} = \text{bus-intPf1}$$

$$\text{hopf2} = \text{hopf1}$$

function V_nc = nc_load(bus, flag, Y22, Y21, psi, V_o, tol, X_kad)

% Purpose: non-conforming load model, for constant power
% and constant current loads; the nonlinear
% equations are solved using a Newton's algorithm

% Input: bus - flag - Y22 - Y21 - psi - V_o - tol

% Output: V_nc

global load_con load_pot bus_int

~~global nload_idx~~

~~global eve_idx n_ccc P_eve~~

~~global busf_idx load_idx nload~~

~~global nload load_idx load_st~~

~~global nload_idx load_idx~~

if ~isempty(load_con)

jay = sqrt(-1);

if flag == 0;

j = bus_int(load_con(:,1));

$$j = \sqrt{-1}$$

$$j = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

V_nc = bus(j,2).*exp(jay*bus(j,3)*pi/180);

load_pot(:,1) = bus(j,6).*load_con(:,2) ...
+ jay*bus(j,7).*load_con(:,3);

S_cc = bus(j,6).*load_con(:,4) ...

+ jay*bus(j,7).*load_con(:,5);

load_pot(:,2) = S_cc./abs(V_nc);

load_pot(:,3) = load_pot(:,1)./V_nc./conj(V_nc);

load_pot(:,4) = S_cc./V_nc./conj(V_nc);

end

if flag == 1

length

nload = size(load_con,1);

V_nc = V_o;

lv_idx = find(abs(V_nc) <= 0.5);

hv_idx = find(abs(V_nc) > 0.5);

curr_mis = zeros(nload,1);

curr_nc = zeros(length(V_nc),1);

curr_load = Y21*psi + Y22*V_nc;

if ~isempty(hv_idx)

curr_nc = -conj([(load_pot(hv_idx,1)+load_pot(hv_idx,2)...
. *abs(V_nc(hv_idx))]./V_nc(hv_idx));

end

if ~isempty(lv_idx)

curr_nc(lv_idx) = -conj(diag(load_pot(lv_idx,3)...
+ load_pot(lv_idx,4))).*V_nc(lv_idx);

end

curr_mis = curr_nc - curr_load

[]

سازی برای
[] - []

```

count = 0;
Y22_real = real(Y22); Y22_imag = imag(Y22);
while(norm(curr_mis,'inf') > tol) → norm(A, 'inf') → مسافة المصفوفة
    if ~isempty(hv_idx)
        v_re = real(V_nc(hv_idx));
        v_im = imag(V_nc(hv_idx));
        v_mag = abs(V_nc(hv_idx));
        v_mag2 = v_mag.*v_mag;
        v1 = (v_re.*v_re - v_im.*v_im);
        v2 = v_re.*v_im;
        vp11 = -v1.*real(load_pot(hv_idx,1))-2*v2.*imag(load_pot(hv_idx,1));
        vp12 = v1.*imag(load_pot(hv_idx,1))-2*v2.*real(load_pot(hv_idx,1));
        vi11 = v_im.*v_im.*real(load_pot(hv_idx,2)) - v2.*imag(load_pot(hv_idx,2));
        vi12 = v_re.*v_re.*imag(load_pot(hv_idx,2)) - v2.*real(load_pot(hv_idx,2));
        vi21 = -v_im.*v_im.*imag(load_pot(hv_idx,2)) - v2.*real(load_pot(hv_idx,2));
        vi22 = v_re.*v_re.*real(load_pot(hv_idx,2)) + v2.*imag(load_pot(hv_idx,2));
        vp11 = vp11./v_mag2./v_mag2;
        vp12 = vp12./v_mag2./v_mag2;
        vi11 = vi11./v_mag./v_mag2;
        vi12 = vi12./v_mag./v_mag2;
        vi21 = vi21./v_mag./v_mag2;
        vi22 = vi22./v_mag./v_mag2;
        v_s11(hv_idx) = (vi11 + vp11);
        v_s12(hv_idx) = (vp12 + vi12);
        v_s21(hv_idx) = (vp12 + vi21);
        v_s22(hv_idx) = -vp11 + vi22;
    end;
    if ~isempty(lv_idx)
        v_s11(lv_idx) = -real(load_pot(lv_idx,3)+load_pot(lv_idx,4));
        v_s12(lv_idx) = imag(load_pot(lv_idx,3)+load_pot(lv_idx,4));
        v_s22(lv_idx) = v_s11(lv_idx);
        v_s21(lv_idx) = -v_s12(lv_idx);
    end
    count = count + 1;
    Jac_nc = [ Y22_real+diag(v_s11) diag(v_s12)-Y22_imag;
                Y22_imag+diag(v_s21) Y22_real+diag(v_s22) ];
    b = [ real(curr_mis) ; imag(curr_mis) ];
    x = Jac_nc\b; → جملة  $Jac\_nc \times [x] = [b]$ 
    V_nc = V_nc + x(1:nload,1) + jay*x(nload+1:2*nload,1);
    lv_idx = find(abs(V_nc)<=0.5);
    hv_idx = find(abs(V_nc) > 0.5);
    curr_load = Y21*psi + Y22*V_nc;

    if ~isempty(hv_idx)
        curr_nc = -conj((load_pot(hv_idx,1)+load_pot(hv_idx,2)...
                        .*abs(V_nc(hv_idx)))./V_nc(hv_idx));
    end

    if ~isempty(lv_idx)
        curr_nc(lv_idx) = -conj(diag(load_pot(lv_idx,3)...
                                + load_pot(lv_idx,4)))*V_nc(lv_idx);
    end
    curr_mis = curr_nc - curr_load;
    if count > 30
        disp('NC_LOAD: Newton algorithm not converged in 30 iterations')
        error('executuion terminated')
    end
end
if flag == 2
end
end

```

مكانته
مكانته
مكانته
مكانته
مكانته
مكانته
مكانته
مكانته

مكانته
مكانته
مكانته
مكانته
مكانته
مكانته
مكانته
مكانته

% S simu.m

تابع انجام مطالعات دینامیکی %

clear all

clear global

close

تیک tic تیک میز رمان شروع کنید

(optional)

```
plot now=0;
```

```
jay = sqrt(-1);
```

pst var

% input data file

* میں خود میں شامل اطلاعات شیڈ یا کارڈ میکلر شامل
bus, line, mac-cin - OSS-CIN, SWI-CIN, ...

% check for valid dynamic data file

* *sw-en*, *mac-en* بال درود و *enq* بال ماتر هرفی *pss-en* دخواهش. فون بایه اس در درودی ماتری *sys freq = 50:*

basrad = $2\pi \text{sys freq}$; — $2\pi f$

basmva = 100 : 100 MV

syn ref == 0

*ibus con = 11; ignore infinite buses in transient simulation

% solve for loadflow - loadflow parameter

`tol = 1e-8; % tolerance for convergence`

iter max = 30; % maximum number of iterations

acc = 1.0; % acceleration factor

*[bus~~.sol~~,line,line flw] = loadflow(bus,line,tol,iter max,acc,'~~n~~',2);

% initialization

```
%set indexes
```

mac indx

exc_idx;

tg_idx;

pss_idx;

ntot = n_{mac};

ngm = n_mac;

$$n-tot = n_{gm} = n_{pm} = n_{mac}$$

```

% construct simulation switching sequence as defined in sw_con
% tswitch() = sw_con(1,1);
k = 1;
n_switch = length(sw_con(:,1)); → sw-con (clocks)
k_inc = zeros(n_switch-1,1);
t_switch = zeros(n_switch,1);
h=t_switch;
for sw_count = 1:n_switch-1
    h(sw_count) = sw_con(sw_count,7);
    if h(sw_count)== 0
        h(sw_count) = 0.01; { → (جيء سير ستيم) Steptime
    end
    k_inc(sw_count) = fix((sw_con(sw_count+1,1)-sw_con(sw_count,1))/h(sw_count));
    if k_inc(sw_count)== 0
        k_inc(sw_count)=1; (جيء بار داير) → (جيء وار) اطلاعات
    end
    h(sw_count) = (sw_con(sw_count+1,1)-sw_con(sw_count,1))/k_inc(sw_count);
    t_switch(sw_count+1) = t_switch(sw_count) + k_inc(sw_count)*h(sw_count);
    t(k:k-1+k_inc(sw_count)) = t_switch(sw_count):h(sw_count):t_switch(sw_count+1)-h(sw_count);
    k=k+k_inc(sw_count);
end
k = sum(k_inc)+1; % k is the total number of time steps in the simulation
t(k) = sw_con(n_switch,1);

```

sw-con = [

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.02 \\ 0.1 & 0 & 0 & 0 & 0 & 0.02 \\ 0.15 & 0 & 0 & 0 & 0 & 0.02 \\ 1 & 0 & 0 & 0 & 0 & 0.1 \\ 3 & 0 & 0 & 0 & 0 & 0.15 \\ 5 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$h(1) = 0.02$$

$$k_inc(1) = \frac{0.1 - 0}{0.02} = 5 \xrightarrow{\text{fix}} h(1) = \frac{0.1 - 0}{5} = 0.02$$

$$h(2) = 0.02$$

$$k_inc(2) = \frac{0.15 - 0.1}{0.02} = 2.5 \xrightarrow{\text{fix}} 2 \xrightarrow{\text{fix}} h(2) = \frac{0.15 - 0.1}{2} = 0.025$$

$$h(3) = 0.02$$

$$k_inc(3) = \frac{1 - 0.15}{0.02} = 42.5 \xrightarrow{\text{fix}} 42 \xrightarrow{\text{fix}} h(3) = \frac{1 - 0.15}{42} = 0.0202$$

$$k_inc(4) = \frac{3 - 1}{0.1} = 20 \xrightarrow{\text{fix}} 20 \xrightarrow{\text{fix}} h(4) = \frac{3 - 1}{20} = 0.1$$

$$h(5) = 0.15$$

$$k_inc(5) = \frac{5 - 3}{0.15} = 4 \rightarrow h(5) = \frac{5 - 3}{4} = 0.15$$

$$k_inc = \begin{bmatrix} 5 \\ 2 \\ 42 \\ 20 \\ 4 \end{bmatrix}$$

$$t = \begin{bmatrix} 0 & 0.02 & \dots & 0.08 & 0.1 \\ 0.02 & 0.06 & \dots & 0.125 & 0.15 \\ 0.025 & 0.075 & \dots & 0.1702 & 0.1702 \\ 0.0202 & 0.0802 & \dots & 0.1905 & 0.1905 \\ 0.0202 & 0.0802 & \dots & 0.9798 & 0.9798 \end{bmatrix}$$

$n = \text{mac_cn}(\{bus\}, k)$

↑
↓
~~n~~ dummy

```

*[n dummy]=size(mac_cn) ;
n_bus = length(bus(:,1)); → n-bus = bus (كبار)
z = zeros(n,k);
z1 = zeros(1,k);
v_p = z1;
theta = zeros(n_bus+1,k); bus_v = zeros(n_bus+1,k);
mac_ang = z; mac_spd = z; dmac_ang = z; dmac_spd = z;
pmech = z; pselect = z; mac_ref = z1; sys_ref = z1;
edprime = z; eqprime = z; dedprime = z; deqprime = z;
psikd = z; psikq = z; dpsikd = z; dpsikq = z;
pm_sig = z;
z_tg = zeros(1,k);
tg1 = z_tg; tg2 = z_tg; tg3 = z_tg; tg4 = z_tg; tg5 = z_tg;
dtg1 = z_tg; dtg2 = z_tg; dtg3 = z_tg; dtg4 = z_tg; dtg5 = z_tg;
tg_sig = z_tg; if n_tg~0; z_tg = zeros(n_tg,k); end
z_pss = zeros(1,k); if n_pss~0; z_pss = zeros(n_pss,k); end
pss1 = z_pss; pss2 = z_pss; pss3 = z_pss;
dpss1 = z_pss; dpss2 = z_pss; dpss3 = z_pss;
curd = z; curq = z; curdg = z; curqg = z; fldcur = z;
ed = z; eq = z; eterm = z; qselect = z;
vex = z; cur_re = z; cur_im = z; psi_re = z; psi_im = z;
ze = zeros(1,k); if n_exc~0; ze = zeros(n_exc,k); end
V_B = ze; exc_sig = ze;
V_TR = ze; V_R = ze; V_A = ze; V_As = ze; Efd = ze; R_f = ze;
dV_TR = ze; dV_R = ze; dV_As = ze; dEfd = ze; dR_f = ze;
pss_out = ze;

sys_freq = ones(1,k);
% step 1: construct reduced Y matrices
y_switch →
disp('initializing other models')
% step 2: initialization
theta(1:n_bus,1) = bus(:,3)*pi/180;
bus_v(1:n_bus,1) = bus(:,2).*exp(jay*theta(1:n_bus,1));
flag = 0; ← Y-switch
bus_int = bus_intprf;% pre-fault system
mac_sub(0,1,bus,flag);
mac_tra(0,1,bus,flag);
mac_em(0,1,bus,flag);
pss(0,1,bus,flag);
exc_st3(0,1,bus,flag);
tg(0,1,bus,flag);

```

تعريف اول

بيانات مدخلات

بيانات خروج

بيانات مدخلات

بيانات خروج

Y-Switch

```
% initialize non-linear loads
if ~isempty(load_con).
    vnc = nc_load(bus,flag,Y_ncprf,Y_ncgprf);
else
    nload = 0;
end
```

nc-load
نکلود

```
H_sum = sum(mac_con(:,16)./mac_pot(:,1));
```

```
% step 3: perform a predictor-corrector integration
```

```
kt = 0;
```

```
ks = 1;
```

```
k_tot = sum(k_inc);
```

```
lswitch = length(k_inc);
```

```
ktmax = k_tot-k_inc(lswitch);
```

```
bus_sim = bus;
```

لایه
لایه

initialization
iniziation

```
while (kt<=ktmax) % شروع حلقه شبیه سازی
```

```
    k_start = kt+1;
```

```
    if kt==ktmax
```

```
        k_end = kt + k_inc(ks);
```

```
    else
```

```
        k_end = kt + k_inc(ks) + 1;
```

```
    end
```

```
for k = k_start:k_end
```

```
    % step 3a: network solution
```

```
    mach_ref(k) = 0;
```

```
    pmech(:,k+1) = pmech(:,k);
```

flag = 1;

flag = 1

لایه، شبیه سازی

```
    % network-machine interface
```

```
    mac_sub(0,k,bus_sim,flag);
```

```
    mac_tra(0,k,bus_sim,flag);
```

```
    mac_em(0,k,bus_sim,flag);
```

```
    % Calculate current injections and bus voltages and angles
```

```
    if k >= sum(k_inc(1:3))+1
```

```
        % fault cleared
```

```
        line_sim = line_pf2;
```

```
        bus_sim = bus_pf2;
```

```
        bus_int = bus_intpf2;
```

```
        Y1 = Y_gpf2;
```

```
        Y2 = Y_gncpf2;
```

```
        Y3 = Y_ncgpf2;
```

```
        Y4 = Y_ncpf2;
```

```
        Vr1 = V_rgpf2;
```

```
        Vr2 = V_rncpf2;
```

```
        bo = bopf2;
```

```
        h_sol = i_simu(k,ks,k_inc,h,bus_sim,Y1,Y2,Y3,Y4,Vr1,Vr2,bo);
```

```
    elseif k >=sum(k_inc(1:2))+1
```

i-simu

Y-sim

```

% near bus cleared
line_sim = line_pf1;
bus_sim = bus_pf1;
bus_int = bus_intpf1;
Y1 = Y_gpf1;
Y2 = Y_gncpf1;
Y3 = Y_ncgpf1;
Y4 = Y_ncpf1;
Vr1 = V_rgpf1;
Vr2 = V_rncpf1;
bo = bopf1;
h_sol = i_simu(k,ks,k_inc,h,bus_sim,Y1,Y2,Y3,Y4,Vr1,Vr2,bo);
elseif k>=k_inc(1)+1
    % fault applied
    line_sim = line_f;
    bus_sim = bus_f;
    bus_int = bus_intf;
    Y1 = Y_gf;
    Y2 = Y_gncf;
    Y3 = Y_ncgf;
    Y4 = Y_ncf;
    Vr1 = V_rgf;
    Vr2 = V_rncf;
    bo = bof;
    h_sol = i_simu(k,ks,k_inc,h,bus_sim,Y1,Y2,Y3,Y4,Vr1,Vr2,bo);
elseif k<k_inc(1)+1
    % pre fault
    line_sim = line;
    bus_sim = bus;
    bus_int = bus_intprf;
    Y1 = Y_gprf;
    Y2 = Y_gncprf;
    Y3 = Y_ncgprf;
    Y4 = Y_ncprf;
    Vr1 = V_rgprf;
    Vr2 = V_rncprf;
    bo = boprf;
    h_sol = i_simu(k,ks,k_inc,h,bus_sim,Y1,Y2,Y3,Y4,Vr1,Vr2,bo);
end
flag = 1
exc_st3(0,k,flag);
tg(0,k,flag);
pss(0,k,flag);

```

... 2, b, b, b, 1, g, f

```

function f = mac_tra(i,k,bus,flag)

global basmva basrad mach_ref
global psi_re psi_im cur_re cur_im bus_int
global mac_con mac_pot
global mac_ang mac_spd eprime edprime
global curd curq curdg curqq fldcur
global vex eterm theta ed eq
global pmech pselect qselect
global dmac_ang dmac_spd deprime dedprime
global n_tra mac_tra_idx pm_sig

jay = sqrt(-1);  $\sqrt{-1} = jay$ 
if n_tra == 0
    if flag == 0; % initialization
        notqp_idx = find(mac_con(mac_tra_idx,14) == 0);
        if ~isempty(notqp_idx)
            mac_con(mac_tra_idx(notqp_idx),14) = 999.0*ones(length(notqp_idx),1);
        end
        busnum = bus_int(mac_con(mac_tra_idx,2)); % bus number
        mac_pot(mac_tra_idx,1) = basmva*ones(n_tra,1)./mac_con(mac_tra_idx,3);
        mac_pot(mac_tra_idx,2) = ones(n_tra,1); % base kv
        % extract bus information
        eterm(mac_tra_idx,1) = bus(busnum,2); % terminal bus voltage
        theta(busnum,1) = bus(busnum,3)*pi/180;
        pselect(mac_tra_idx,1) = bus(busnum,4).*mac_con(mac_tra_idx,22);
        qselect(mac_tra_idx,1) = bus(busnum,5).*mac_con(mac_tra_idx,23);
        curr = sqrt(pselect(mac_tra_idx,1).^2 + qselect(mac_tra_idx,1).^2) ...
            ./ eterm(mac_tra_idx,1).*mac_pot(mac_tra_idx,1); % current magnitude
        phi = atan2(qselect(mac_tra_idx,1), pselect(mac_tra_idx,1));
        v = eterm(mac_tra_idx,1).*exp(jay*theta(busnum,1));
        curr = curr.*exp(jay*(theta(busnum,1)-phi));
        eprime = v + (mac_con(mac_tra_idx,5)+jay*mac_con(mac_tra_idx,7)).*curr;
        ei = v + (mac_con(mac_tra_idx,5)+jay*mac_con(mac_tra_idx,11)).*curr;
        mac_ang(mac_tra_idx,1) = atan2(imag(ei), real(ei));
        mac_spd(mac_tra_idx,1) = ones(n_tra,1);
        rot = jay*exp(-jay*mac_ang(mac_tra_idx,1));
        psi_re(mac_tra_idx,1) = real(eprime);
        psi_im(mac_tra_idx,1) = imag(eprime);
        eprime = eprime.*rot;
        edprime(mac_tra_idx,1) = real(eprime);
        eqprime(mac_tra_idx,1) = imag(eprime);

        curr = curr.*rot;
        mcurmag = abs(curr);
        pmech(mac_tra_idx,1) = pselect(mac_tra_idx,1).*mac_pot(mac_tra_idx,1) ...
            + mac_con(mac_tra_idx,5).*mcurmag.*mcurmag;
        curdg(mac_tra_idx,1) = real(curr);
        curqq(mac_tra_idx,1) = imag(curr);
        curd(mac_tra_idx,1) = real(curr)./mac_pot(mac_tra_idx,1);
        curq(mac_tra_idx,1) = imag(curr)./mac_pot(mac_tra_idx,1);
        v = v.*rot;
        ed(mac_tra_idx,1) = real(v);
        eq(mac_tra_idx,1) = imag(v);
        % compute saturation
        inv_sat = inv([0.64 0.8 1; 1 1 1; 1.44 1.2 1]);
        b = [0.8*ones(n_tra,1) ones(n_tra,1)+mac_con(mac_tra_idx,20) ...
            1.2*(ones(n_tra,1)+mac_con(mac_tra_idx,21))];
        mac_pot(mac_tra_idx,3) = b*inv_sat(1,:);
        mac_pot(mac_tra_idx,4) = b*inv_sat(2,:);
        mac_pot(mac_tra_idx,5) = b*inv_sat(3,:);
    end
end

```

mac_tra
bus
4
mac_idx

mac_idx

flag == 0 → initialization
flag == 0

14
mac_con = 0
value 999 ←

$$\begin{aligned} & \text{Pselect}^2 + \text{Qselect}^2 \\ & \text{eterm} \cdot \text{mac-pot} \\ & \Phi = \tan^{-1} \left(\frac{\text{Qselect}}{\text{Pselect}} \right) \end{aligned}$$

$$\text{inv-Sat} = \begin{bmatrix} 12.5 & -25 & 12.5 \\ -27.5 & 50 & -22.5 \\ 15 & -24 & 10 \end{bmatrix}_{3 \times 3}$$

$$b = \begin{bmatrix} .8 & 1 + \text{mac-con}(20) \\ .8 & " \\ .8 & " \end{bmatrix}$$

n_tra x 1

n_tra x 3

n_tra x 3

$$1.2 \times (1 + \text{mac-con}(21))$$

b constant
n_tra x 3

```

24
E_Isat = mac_pot(mac_tra_idx,3).*eqprime(mac_tra_idx,1).^2 ...
    + mac_pot(mac_tra_idx,4).*eqprime(mac_tra_idx,1)...
    + mac_pot(mac_tra_idx,5);
nosat_idx=find(eqprime(mac_tra_idx,1)<0.8);
if ~isempty(nosat_idx)
    E_Isat(nosat_idx)=eqprime(mac_tra_idx(nosat_idx),1);
end
vex(mac_tra_idx,1) = E_Isat + (mac_con(mac_tra_idx,6)-...
    mac_con(mac_tra_idx,7)).*curdg(mac_tra_idx,1);
fldcur(mac_tra_idx,1) = vex(mac_tra_idx,1);
end

if flag == 1 % network interface computation
    mac_ang(mac_tra_idx,k) = mac_ang(mac_tra_idx,k)-mach_ref(k)*ones(n_tra,1);
    psi_re(mac_tra_idx,k) = sin(mac_ang(mac_tra_idx,k)).*edprime(mac_tra_idx,k) + ...
        cos(mac_ang(mac_tra_idx,k)).*eqprime(mac_tra_idx,k);
    psi_im(mac_tra_idx,k) = -cos(mac_ang(mac_tra_idx,k)).*edprime(mac_tra_idx,k) + ...
        sin(mac_ang(mac_tra_idx,k)).*eqprime(mac_tra_idx,k);
% end of interface calculation
end

if flag == 2 % generator dynamics calculation
    curd(mac_tra_idx,k) = sin(mac_ang(mac_tra_idx,k)).*cur_re(mac_tra_idx,k) - ...
        cos(mac_ang(mac_tra_idx,k)).*cur_im(mac_tra_idx,k);
    curq(mac_tra_idx,k) = cos(mac_ang(mac_tra_idx,k)).*cur_re(mac_tra_idx,k) + ...
        sin(mac_ang(mac_tra_idx,k)).*cur_im(mac_tra_idx,k);
    curdg(mac_tra_idx,k) = curd(mac_tra_idx,k).*mac_pot(mac_tra_idx,1);
    curqg(mac_tra_idx,k) = curq(mac_tra_idx,k).*mac_pot(mac_tra_idx,1);
    E_Isat = mac_pot(mac_tra_idx,3).*eqprime(mac_tra_idx,k).^2 ...
        + mac_pot(mac_tra_idx,4).*eqprime(mac_tra_idx,k) + mac_pot(mac_tra_idx,5);
    nosat_idx=find(eqprime(mac_tra_idx,1)<0.8);
    if ~isempty(nosat_idx)
        E_Isat(nosat_idx)=eqprime(mac_tra_idx(nosat_idx),k);
    end
    fldcur(mac_tra_idx,k) = E_Isat + (mac_con(mac_tra_idx,6)-mac_con(mac_tra_idx,7))...
        .*curdg(mac_tra_idx,k);
    deqprime(mac_tra_idx,k) = (vex(mac_tra_idx,k) - fldcur(mac_tra_idx,k))./mac_con(mac_tra_idx,9);
    dedprime(mac_tra_idx,k) = (-edprime(mac_tra_idx,k) +...
        (mac_con(mac_tra_idx,11)-mac_con(mac_tra_idx,12))...
        .*curqg(mac_tra_idx,k))./mac_con(mac_tra_idx,14);
    ed(mac_tra_idx,k) = edprime(mac_tra_idx,k) - mac_con(mac_tra_idx,5).*curdg(mac_tra_idx,k)...
        + mac_con(mac_tra_idx,7).*curqg(mac_tra_idx,k);
    eq(mac_tra_idx,k) = eqprime(mac_tra_idx,k) - mac_con(mac_tra_idx,5).*curqg(mac_tra_idx,k)...
        - mac_con(mac_tra_idx,7).*curdg(mac_tra_idx,k);
    eterm(mac_tra_idx,k) = sqrt(ed(mac_tra_idx,k).^2+eq(mac_tra_idx,k).^2);
    pselect(mac_tra_idx,k) = eq(mac_tra_idx,k).*curq(mac_tra_idx,k)...
        + ed(mac_tra_idx,k).*curd(mac_tra_idx,k);
    qselect(mac_tra_idx,k) = eq(mac_tra_idx,k).*curd(mac_tra_idx,k)...
        - ed(mac_tra_idx,k).*curq(mac_tra_idx,k);
    curmag = abs(curdg(mac_tra_idx,k) + jay*curqg(mac_tra_idx,k));
    Te = pselect(mac_tra_idx,k).*mac_pot(mac_tra_idx,1) + mac_con(mac_tra_idx,5).*curmag.*curmag;
    dmac_ang(mac_tra_idx,k) = basrad*(mac_spd(mac_tra_idx,k)-ones(n_tra,1));
    dmac_spd(mac_tra_idx,k) = (pmech(mac_tra_idx,k)+pm_sig(mac_tra_idx,k) - Te ...
        - mac_con(mac_tra_idx,17).*(mac_spd(mac_tra_idx,k)-ones(n_tra,1)))./(2*mac_con(mac_tra_idx,16));
end
% end calculation of rates of change
end

```

mac-em چیزهای مذکور
در اینجا در نظر گرفته شده

```

function [f] = mac_em(i,k,bus,flag)

global basmva basrad syn_ref mach_ref sys_freq
global bus_v bus_ang psi_re psi_im cur_re cur_im bus_int
global mac_con mac_pot mac_int n_em
global mac_ang mac_spd eqprime edprime
global curd curq curdg curqq
global vex eterm theta ed eq
global pmech pselect qselect mac_em_idx
global dmac_ang dmac_spd deqprime dedprime

if n_em == 0
if flag == 0; % initialization
end

if flag == 1 % network interface computation
    mac_ang(mac_em_idx,k) = mac_ang(mac_em_idx,k) - mach_ref(k)*ones(n_em,1);
    psi_re(mac_em_idx,k) = sin(mac_ang(mac_em_idx,k)).*edprime(mac_em_idx,k) + ...
                           cos(mac_ang(mac_em_idx,k)).*eqprime(mac_em_idx,k);
    psi_im(mac_em_idx,k) = -cos(mac_ang(mac_em_idx,k)).*edprime(mac_em_idx,k) + ...
                           sin(mac_ang(mac_em_idx,k)).*eqprime(mac_em_idx,k);
end
if flag == 2 % generator dynamics calculation
    curd(mac_em_idx,k) = sin(mac_ang(mac_em_idx,k)).*cur_re(mac_em_idx,k) - ...
                           cos(mac_ang(mac_em_idx,k)).*cur_im(mac_em_idx,k);
    curq(mac_em_idx,k) = cos(mac_ang(mac_em_idx,k)).*cur_re(mac_em_idx,k) + ...
                           sin(mac_ang(mac_em_idx,k)).*cur_im(mac_em_idx,k);
    curdg(mac_em_idx,k) = curd(mac_em_idx,k).*mac_pot(mac_em_idx,1);
    curqq(mac_em_idx,k) = curq(mac_em_idx,k).*mac_pot(mac_em_idx,1);

    dedprime(mac_em_idx,k) = zeros(n_em,1);
    deqprime(mac_em_idx,k) = zeros(n_em,1);
end

ed(mac_em_idx,k) = edprime(mac_em_idx,k) ...
                  + mac_con(mac_em_idx,7).*curqq(mac_em_idx,k);
eq(mac_em_idx,k) = eqprime(mac_em_idx,k) ...
                  - mac_con(mac_em_idx,7).*curdg(mac_em_idx,k);
eterm(mac_em_idx,k) = sqrt(ed(mac_em_idx,k).^2+eq(mac_em_idx,k).^2);  $\sqrt{e_q^2 + e_d^2}$ 

pselect(mac_em_idx,k) = eq(mac_em_idx,k).*curq(mac_em_idx,k) ...
                      + ed(mac_em_idx,k).*curd(mac_em_idx,k);
qselect(mac_em_idx,k) = eq(mac_em_idx,k).*curd(mac_em_idx,k) ...
                      - ed(mac_em_idx,k).*curq(mac_em_idx,k);

dmac_ang(mac_em_idx,k) = basrad*(mac_spd(mac_em_idx,k)-ones(n_em,1));
dmac_spd(mac_em_idx,k) = (pmech(mac_em_idx,k) + pm_sig(mac_em_idx,k) ...
                           - pselect(mac_em_idx,k).*mac_pot(mac_em_idx,1)...
                           - mac_con(mac_em_idx,17).*mac_spd(mac_em_idx,k)...
                           - ones(n_em,1)) ./ (2*mac_con(mac_em_idx,16));
end
end

```

mac-sub initialization + global vars

```
function f = mac_sub(i,k,bus,flag)

global basmva basrad mach_ref sys_freq
global bus_int
global mac_con mac_pot
global mac_ang mac_spd eqprime edprime psikd psikq
global psi_re psi_im cur_re cur_im
global curd curq curdg curqg fldcur
global psidpp psiqpp vex eterm theta ed eq
global pmech pselect qselect
global dmac_ang dmac_spd deqprime dedprime dpsikd dpsikq
global n_sub mac_sub_idx pm_sig

if n_sub==0
    if flag == 0; % initialization

        uets_idx = find(mac_con(mac_sub_idx,8) ~= mac_con(mac_sub_idx,13));
        if ~isempty(uet_idx)
            mac_con(mac_sub_idx(uet_idx),13) = mac_con(mac_sub_idx(uet_idx),8);
        end

        notp_idx = find(mac_con(mac_sub_idx,14) == 0);
        if ~isempty(notp_idx)
            mac_con(mac_sub_idx(notp_idx),14) = 999.0*ones(length(notp_idx),1);
        end

        notpp_idx = find(mac_con(mac_sub_idx,15) == 0);
        if ~isempty(notpp_idx)
            mac_con(mac_sub_idx(notpp_idx),15) = 999.0*ones(length(notpp_idx),1);
            mac_con(mac_sub_idx(notpp_idx),12) = mac_con(mac_sub_idx(notpp_idx),13);
        end

        busnum = bus_int(mac_con(mac_sub_idx,2));
        mac_pot(mac_sub_idx,1) = basmva*ones(n_sub,1)./mac_con(mac_sub_idx,3);
        mac_pot(mac_sub_idx,2) = ones(n_sub,1);
        mac_pot(mac_sub_idx,8) = mac_con(mac_sub_idx,7)-mac_con(mac_sub_idx,4);

        mac_pot(mac_sub_idx,9) = (mac_con(mac_sub_idx,8)-mac_con(mac_sub_idx,4))...
            ./mac_pot(mac_sub_idx,8);
        mac_pot(mac_sub_idx,7) = (mac_con(mac_sub_idx,6)-mac_con(mac_sub_idx,7))...
            .*mac_pot(mac_sub_idx,9);
        mac_pot(mac_sub_idx,10) = (mac_con(mac_sub_idx,7)-mac_con(mac_sub_idx,8))...
            ./mac_pot(mac_sub_idx,8);
        mac_pot(mac_sub_idx,6) = (mac_con(mac_sub_idx,6)-mac_con(mac_sub_idx,7))...
            ./mac_pot(mac_sub_idx,8).*mac_pot(mac_sub_idx,10);
        mac_pot(mac_sub_idx,13) = mac_con(mac_sub_idx,12)-mac_con(mac_sub_idx,4);
        mac_pot(mac_sub_idx,14) = (mac_con(mac_sub_idx,13)-mac_con(mac_sub_idx,4))...
            ./mac_pot(mac_sub_idx,13);
        mac_pot(mac_sub_idx,12) = (mac_con(mac_sub_idx,11)-mac_con(mac_sub_idx,12))...
            .*mac_pot(mac_sub_idx,14);
        mac_pot(mac_sub_idx,15) = (mac_con(mac_sub_idx,12)-mac_con(mac_sub_idx,13))...
            ./mac_pot(mac_sub_idx,13);
        mac_pot(mac_sub_idx,11) = (mac_con(mac_sub_idx,11)-mac_con(mac_sub_idx,12))...
            ./mac_pot(mac_sub_idx,13).*mac_pot(mac_sub_idx,15);

        eterm(mac_sub_idx,1) = bus(busnum,2);
        theta(busnum,1) = bus(busnum,3)*pi/180;
        pselect(mac_sub_idx,1) = bus(busnum,4).*mac_con(mac_sub_idx,22);
        qselect(mac_sub_idx,1) = bus(busnum,5).*mac_con(mac_sub_idx,23);
        curr = sqrt(pselect(mac_sub_idx,1).^2+qselect(mac_sub_idx,1).^2) ...
            ./eterm(mac_sub_idx,1).*mac_pot(mac_sub_idx,1);
        phi = atan2(qselect(mac_sub_idx,1),pselect(mac_sub_idx,1));
        v = eterm(mac_sub_idx,1).*exp(jay*theta(busnum,1));
        curr = curr.*exp(jay*(theta(busnum,1)-phi));
        ei = v + (mac_con(mac_sub_idx,5)+jay*mac_con(mac_sub_idx,11)).*curr;
        mac_ang(mac_sub_idx,1) = atan2(imag(ei),real(ei));
        mac_spd(mac_sub_idx,1) = ones(n_sub,1);
        rot = jay*exp(-jay*mac_ang(mac_sub_idx,1));
        curr = curr.*rot;
    end
end
```

3

```

mcurmag = abs(curr);
pmech(mac_sub_idx,1) = pselect(mac_sub_idx,1).*mac_pot(mac_sub_idx,1)...
+ mac_con(mac_sub_idx,5).*{mcurmag.*mcurmag};
curdg(mac_sub_idx,1) = real(curr);
curgq(mac_sub_idx,1) = imag(curr);
curd(mac_sub_idx,1) = real(curr)./mac_pot(mac_sub_idx,1);
curq(mac_sub_idx,1) = imag(curr)./mac_pot(mac_sub_idx,1);
v = v.*rot;
ed(mac_sub_idx,1) = real(v);
eq(mac_sub_idx,1) = imag(v);

eqra = eq(mac_sub_idx,1)+mac_con(mac_sub_idx,5).*curgq(mac_sub_idx,1);
psidpp = eqra + mac_con(mac_sub_idx,8).*curdg(mac_sub_idx,1);

psikd(mac_sub_idx,1) = eqra + mac_con(mac_sub_idx,4).*curdg(mac_sub_idx,1);

eqprime(mac_sub_idx,1) = eqra + mac_con(mac_sub_idx,7).*curdg(mac_sub_idx,1);
edra = -ed(mac_sub_idx,1)-mac_con(mac_sub_idx,5).*curdg(mac_sub_idx,1);

psiqpp = edra + mac_con(mac_sub_idx,13).*curgq(mac_sub_idx,1);
psikq(mac_sub_idx,1) = edra + mac_con(mac_sub_idx,4).*curgq(mac_sub_idx,1);

edprime(mac_sub_idx,1) = edra + mac_con(mac_sub_idx,12).*curgq(mac_sub_idx,1);

inv sat inv_sat = inv([0.64 0.8 1;1 1 1;1.44 1.2 1]);
b= [0.8*ones(n_sub,1) ones(n_sub,1)+mac_con(mac_sub_idx,20)...
1.2*(ones(n_sub,1)+mac_con(mac_sub_idx,21)) ];
mac_pot(mac_sub_idx,3) = b*inv_sat(1,:)';
mac_pot(mac_sub_idx,4) = b*inv_sat(2,:)';
mac_pot(mac_sub_idx,5) = b*inv_sat(3,:');

E_Isat = mac_pot(mac_sub_idx,3).*eqprime(mac_sub_idx,1).^2 ...
+ mac_pot(mac_sub_idx,4).*eqprime(mac_sub_idx,1) + mac_pot(mac_sub_idx,5);
nosat_idx=find(eqprime(mac_sub_idx,1)<.8);
if ~isempty(nosat_idx)
    E_Isat(nosat_idx)=eqprime(mac_sub_idx(nosat_idx),1);
end
vex(mac_sub_idx,1) = E_Isat + mac_pot(mac_sub_idx,6).*{eqprime(mac_sub_idx,1)...
- psikd(mac_sub_idx,1)}+mac_pot(mac_sub_idx,7).*curdg(mac_sub_idx,1);
fldcur(mac_sub_idx,1) = vex(mac_sub_idx,1);
psi_re(mac_sub_idx,1) = sin(mac_ang(mac_sub_idx,1)).*(-psiqpp) ...
+ cos(mac_ang(mac_sub_idx,1)).*psidpp;
psi_im(mac_sub_idx,1) = -cos(mac_ang(mac_sub_idx,1)).*(-psiqpp) ...
+ sin(mac_ang(mac_sub_idx,1)).*psidpp;
end
if flag == 1 % network interface computation
    mac_ang(mac_sub_idx,k) = mac_ang(mac_sub_idx,k)-mach_ref(k)*ones(n_sub,1);

    psidpp = mac_pot(mac_sub_idx,9).*eqprime(mac_sub_idx,k)...
+ mac_pot(mac_sub_idx,10).*psikd(mac_sub_idx,k);
    psiqpp = mac_pot(mac_sub_idx,14).*edprime(mac_sub_idx,k)...
+ mac_pot(mac_sub_idx,15).*psikq(mac_sub_idx,k);
    psi_re(mac_sub_idx,k) = sin(mac_ang(mac_sub_idx,k)).*(-psiqpp)...
+ cos(mac_ang(mac_sub_idx,k)).*psidpp;
    psi_im(mac_sub_idx,k) = -cos(mac_ang(mac_sub_idx,k)).*(-psiqpp)...
+ sin(mac_ang(mac_sub_idx,k)).*psidpp;
end

if flag == 2 % generator dynamics calculation
    if i ~= 0
        psiqpp = mac_pot(mac_sub_idx,14).*edprime(mac_sub_idx,k) + ...
+ mac_pot(mac_sub_idx,15).*psikq(mac_sub_idx,k);
    end
end

```

9

```

psidpp = mac_pot(mac_sub_idx,9).*eqprime(mac_sub_idx,k) + ...
    mac_pot(mac_sub_idx,10).*psikd(mac_sub_idx,k);
curd(mac_sub_idx,k) = sin(mac_ang(mac_sub_idx,k)).*cur_re(mac_sub_idx,k) - ...
    cos(mac_ang(mac_sub_idx,k)).*cur_im(mac_sub_idx,k);
curq(mac_sub_idx,k) = cos(mac_ang(mac_sub_idx,k)).*cur_re(mac_sub_idx,k) + ...
    sin(mac_ang(mac_sub_idx,k)).*cur_im(mac_sub_idx,k);
curdg(mac_sub_idx,k) = curd(mac_sub_idx,k).*mac_pot(mac_sub_idx,1);
curqg(mac_sub_idx,k) = curq(mac_sub_idx,k).*mac_pot(mac_sub_idx,1);
mcurmag = abs(curdg(mac_sub_idx,k)+jay*curqg(mac_sub_idx,k));
E_Isat = mac_pot(mac_sub_idx,3).*eqprime(mac_sub_idx,k).^2 ...
    + mac_pot(mac_sub_idx,4).*eqprime(mac_sub_idx,k) + mac_pot(mac_sub_idx,5);
nosat_idx=find(eqprime(mac_sub_idx,1)<.8);
if ~isempty(nosat_idx)
    E_Isat(nosat_idx)=eqprime(mac_sub_idx(nosat_idx),k);
end

fldcur(mac_sub_idx,k) = E_Isat + mac_pot(mac_sub_idx,6)...
    .*eqprime(mac_sub_idx,k)-psikd(mac_sub_idx,k)) + mac_pot(mac_sub_idx,7)...
    .*curdg(mac_sub_idx,k);

deqprime(mac_sub_idx,k) = (vex(mac_sub_idx,k)-fldcur(mac_sub_idx,k))./mac_con(mac_sub_idx,9);

dpsikd(mac_sub_idx,k) = (-psikd(mac_sub_idx,k)+eqprime(mac_sub_idx,k)-mac_pot(mac_sub_idx,8)...
    .*curdg(mac_sub_idx,k))./mac_con(mac_sub_idx,10);

dedprime(mac_sub_idx,k) = (-edprime(mac_sub_idx,k) - mac_pot(mac_sub_idx,11)...
    .*edprime(mac_sub_idx,k)-psikq(mac_sub_idx,k)) - mac_pot(mac_sub_idx,12)...
    .*curqg(mac_sub_idx,k))./mac_con(mac_sub_idx,14);

dpsikq(mac_sub_idx,k) = (edprime(mac_sub_idx,k)-psikq(mac_sub_idx,k)-mac_pot(mac_sub_idx,13)...
    .*curqg(mac_sub_idx,k))./mac_con(mac_sub_idx,15);

ed(mac_sub_idx,k) = -mac_con(mac_sub_idx,5).*curdg(mac_sub_idx,k) - (psiqpp...
    -mac_con(mac_sub_idx,13).*curqg(mac_sub_idx,k));

eq(mac_sub_idx,k) = -mac_con(mac_sub_idx,5).*curqg(mac_sub_idx,k) + (psidpp...
    -mac_con(mac_sub_idx,8).*curdg(mac_sub_idx,k));

eterm(mac_sub_idx,k) = sqrt(ed(mac_sub_idx,k).^2+eq(mac_sub_idx,k).^2);

pelect(mac_sub_idx,k) = eq(mac_sub_idx,k).*curq(mac_sub_idx,k) + ed(mac_sub_idx,k).*curd(mac_sub_idx,k);

qelect(mac_sub_idx,k) = eq(mac_sub_idx,k).*curd(mac_sub_idx,k) - ed(mac_sub_idx,k).*curq(mac_sub_idx,k);

dmac_ang(mac_sub_idx,k) = basrad*(mac_spd(mac_sub_idx,k)-ones(n_sub,1));

Te = pelect(mac_sub_idx,k).*mac_pot(mac_sub_idx,1) + mac_con(mac_sub_idx,5).*mcurmag.*mcurmag;

dmac_spd(mac_sub_idx,k) = (pmech(mac_sub_idx,k)+ pm_sig(mac_sub_idx,k)-Te...
    -mac_con(mac_sub_idx,17).*(mac_spd(mac_sub_idx,k)-ones(n_sub,1))...
    -mac_con(mac_sub_idx,18).*(mac_spd(mac_sub_idx,k)-sys_freq(k)...
    *ones(n_sub,1)))./(2*mac_con(mac_sub_idx,16));

```

end
end
end

```

% step 3b: compute dynamics and integrate
flag = 2;
sys_freq(k) = 1.0;
mac_sub(0,k,bus_sim,flag);
mac_tra(0,k,bus_sim,flag); pss مُطابق
mac_em(0,k,bus_sim,flag);
pss(0,k,flag); Flag=2
exc_st3(0,k,flag);
tg(0,k,flag);

%% predictor steps
j = k+1; خطوة التكامل

%% mac
mac_ang(:,j) = mac_ang(:,k) + h_sol*dmac_ang(:,k);
mac_spd(:,j) = mac_spd(:,k) + h_sol*dmac_spd(:,k);
edprime(:,j) = edprime(:,k) + h_sol*dedprime(:,k);
eqprime(:,j) = eqprime(:,k) + h_sol*deqprime(:,k);
psikd(:,j) = psikd(:,k) + h_sol*dpsikd(:,k);
psikq(:,j) = psikq(:,k) + h_sol*dpsikq(:,k);

%% exc
Efd(:,j) = Efd(:,k) + h_sol*dEfd(:,k);
V_R(:,j) = V_R(:,k) + h_sol*dV_R(:,k);
V_As(:,j) = V_As(:,k) + h_sol*dV_As(:,k);
R_f(:,j) = R_f(:,k) + h_sol*dR_f(:,k);
V_TR(:,j) = V_TR(:,k) + h_sol*dV_TR(:,k);

%% pss
pss1(:,j) = pss1(:,k) + h_sol*dpss1(:,k);
pss2(:,j) = pss2(:,k) + h_sol*dpss2(:,k);
pss3(:,j) = pss3(:,k) + h_sol*dpss3(:,k);

%% tg
tg1(:,j) = tg1(:,k) + h_sol*dtg1(:,k);
tg2(:,j) = tg2(:,k) + h_sol*dtg2(:,k);
tg3(:,j) = tg3(:,k) + h_sol*dtg3(:,k);
tg4(:,j) = tg4(:,k) + h_sol*dtg4(:,k);
tg5(:,j) = tg5(:,k) + h_sol*dtg5(:,k);

flag = 1;
mach_ref(j) = 0;

```

Predictor: $x(i+1) = x(i) + h \Delta x$

3
predictor and predictor solution

```
% perform network interface calculations again with predicted states
mac_sub(0,j,bus_sim,flag);
mac_tra(0,j,bus_sim,flag);
mac_em(0,j,bus_sim,flag);

% Calculate current injections and bus voltages and angles
if j >= sum(k_inc(1:3))+1
    % fault cleared
    bus_sim = bus_pf2;
    bus_int = bus_intpf2;
    Y1 = Y_gpf2;
    Y2 = Y_gncpf2;
    Y3 = Y_ncgpf2;
    Y4 = Y_ncpf2;
    Vr1 = V_rgpf2;
    Vr2 = V_rncpf2;
    bo = bopf2;
    h_sol = i_simu(j,ks,k_inc,h,bus_sim,Y1,Y2,Y3,Y4,Vr1,Vr2,bo);
elseif j >= sum(k_inc(1:2))+1
    % near bus cleared
    bus_sim = bus_pf1;
    bus_int = bus_intpf1;
    Y1 = Y_gpf1;
    Y2 = Y_gncpf1;
    Y3 = Y_ncgpf1;
    Y4 = Y_ncpf1;
    Vr1 = V_rgpf1;
    Vr2 = V_rncpf1;
    bo = bopf1;
    h_sol = i_simu(j,ks,k_inc,h,bus_sim,Y1,Y2,Y3,Y4,Vr1,Vr2,bo);
elseif j >= k_inc(1)+1
    % fault applied
    bus_sim = bus_f;
    bus_int = bus_intf;
    Y1 = Y_gf;
    Y2 = Y_gncf;
    Y3 = Y_ncgf;
    Y4 = Y_ncf;
    Vr1 = V_rgcf;
    Vr2 = V_rncf;
    bo = bof;
    h_sol = i_simu(j,ks,k_inc,h,bus_sim,Y1,Y2,Y3,Y4,Vr1,Vr2,bo);
elseif k < k_inc(1)+1
    % pre fault
    bus_sim = bus;
    bus_int = bus_intprf;
    Y1 = Y_gprf;
    Y2 = Y_gncprf;
    Y3 = Y_ncgprf;
    Y4 = Y_ncprf;
```

```

Vr1 = V_rgprf;
Vr2 = V_rncprf;
bo = boprf;
h_sol = i_simu(j, ks, k_inc, h, bus_sim, Y1, Y2, Y3, Y4, Vr1, Vr2, bo);
end

h_sol = i_simu(j, ...), if 4 > i_simu then h_sol = 4 else 0
vex(:,j)=vex(:,k); cur_ord(:,j) = cur_ord(:,k);

pss(0,j,bus_sim,flag);
exc_st3(0,j,bus_sim,flag);
tg(0,j,bus_sim,flag); flag = 0;
=1

%% corrector steps
flag = 2;
mac_sub(0,j,bus_sim,flag);
mac_tra(0,j,bus_sim,flag);
mac_em(0,j,bus_sim,flag);
pss(0,j,flag);
exc_st3(0,j,flag);
tg(0,j,flag);

%% mac
mac_ang(:,j) = mac_ang(:,k) + ...
    h_sol*(dmac_ang(:,k)+dmac_ang(:,j))/2.;;
mac_spd(:,j) = mac_spd(:,k) + ...
    h_sol*(dmac_spd(:,k)+dmac_spd(:,j))/2.;;
edprime(:,j) = edprime(:,k) + ...
    h_sol*(dedprime(:,k)+dedprime(:,j))/2.;;
eqprime(:,j) = eqprime(:,k) + ...
    h_sol*(deqprime(:,k)+deqprime(:,j))/2.;;
psikd(:,j) = psikd(:,k) + ...
    h_sol*(dpsikd(:,k)+dpsikd(:,j))/2.;;
psikq(:,j) = psikq(:,k) + ...
    h_sol*(dpsikq(:,k)+dpsikq(:,j))/2.;;

%% exc
Efd(:,j) = Efd(:,k) + ...
    h_sol*(dEfd(:,k)+dEfd(:,j))/2.;;
V_R(:,j) = V_R(:,k) + ...
    h_sol*(dV_R(:,k)+dV_R(:,j))/2.;;
V_As(:,j) = V_As(:,k) + ...
    h_sol*(dV_As(:,k)+dV_As(:,j))/2.;;
R_f(:,j) = R_f(:,k) + ...
    h_sol*(dR_f(:,k)+dR_f(:,j))/2.;;
V_TR(:,j) = V_TR(:,k) + ...
    h_sol*(dV_TR(:,k)+dV_TR(:,j))/2.;;

Corrector: 
$$x(i+1) = x(i) + h \left( \frac{dx(i) + dx(i+1)}{2} \right)$$


```

%% pss

```
pss1(:,j) = pss1(:,k) + h_sol*(dpss1(:,k)+dpss1(:,j))/2.;  
pss2(:,j) = pss2(:,k) + h_sol*(dpss2(:,k)+dpss2(:,j))/2.;  
pss3(:,j) = pss3(:,k) + h_sol*(dpss3(:,k)+dpss3(:,j))/2.;
```

%% tg

```
tg1(:,j) = tg1(:,k) + h_sol*(dtg1(:,k) + dtg1(:,j))/2.;  
tg2(:,j) = tg2(:,k) + h_sol*(dtg2(:,k) + dtg2(:,j))/2.;  
tg3(:,j) = tg3(:,k) + h_sol*(dtg3(:,k) + dtg3(:,j))/2.;  
tg4(:,j) = tg4(:,k) + h_sol*(dtg4(:,k) + dtg4(:,j))/2.;  
tg5(:,j) = tg5(:,k) + h_sol*(dtg5(:,k) + dtg5(:,j))/2.;
```

end

% counter increment

```
kt = kt + k_inc(ks);  
ks = ks+1;
```

{ time
step
کل

کل سریل

پایان حلقه شبیه سازی

محاسبه جریانهای خطوط

```
V1 = bus_v(bus_int(line(:,1)),:);  
V2 = bus_v(bus_int(line(:,2)),:);  
R = line(:,3);  
X = line(:,4);  
B = line(:,5);  
tap = line(:,6);  
phi = line(:,7);  
[ilf,ilt]=line_cur(V1,V2,R,X,B,tap,phi); %% line currents
```

برای اینجا مقدار داده شده است
و باس، بین خطوط، پارامترهای سیستم را برای محاسبه در

تقریباً

tic - toc
et = toc;

رسم نمودار خروجی های مورد نظر

```
function [S1,S2] = line_pq(V1,V2,R,X,B,tap,phi)
```

هدف: محاسبه فلوی خطوط

```
% Input: V1      - from bus complex voltage matrix
%        V2      - to bus complex voltage matrix
%        R       - line resistance vector
%        X       - line reactance vector
%        B       - line charging vector
%        tap     - tap ratio vector
%        phi    - phase shifter angle vector in degrees
% Output: S1     - complex power injection matrix at from bus
%          S2     - complex power injection matrix at to bus
```

```
jay = sqrt(-1);
[nline,dummy] = size(V1);
for i = 1:nline
    if tap(i) == 0
        tap(i) = 1;
    end
end
tps = tap.*exp(jay*phi*pi/180);
tpsi = diag(ones(nline,1)./tps);
tps = diag(tpsi);
z = R + jay*X;
y = diag(ones(nline,1)./z);
chg = diag(jay*B/2);
curl = tps*(y*(tpsi*V1-V2) + chg*V1);
cur2 = y*(V2 - tpsi*V1) + chg*V2;
S1 = V1.*conj(curl);
S2 = V2.*conj(cur2);
```

```
function [l_if,l_it] = line_cur(V1,V2,R,X,B,tap,phi)
```

هدف: محاسبه جریانهای خطوط

```
% Input: V1, V2, R, X, B, tap, phi
% Output: l_if      - complex current (from bus)
%          l_it      - complex current (to bus)
```

```
jay = sqrt(-1);
[nline,dummy] = size(V1);
for i = 1:nline
    if tap(i) == 0
        tap(i) = 1;
    end
end
tps = tap.*exp(jay*phi*pi/180);
tpsi = diag(ones(nline,1)./tps);
tps = diag(tpsi);
z = R + jay*X;
y = diag(ones(nline,1)./z);
chg = diag(jay*B/2);
l_if = tps*(y*(tpsi*V1-V2)) + chg*V1;
l_it = y*(V2 - tpsi*V1) + chg*V2;
```

$$\text{tap} \cdot e^{j\frac{\pi}{180}} \rightarrow \begin{bmatrix} \frac{1}{t_{ps}(1)} & 0 \\ 0 & \frac{1}{t_{ps}(2)} \end{bmatrix}$$
nline x nline

جواب اینجا

برد خط امداد

line-pq

curl و مکاری

l-if و l-it

l-if و l-it

```

bus_new = bus;
if ~isempty(svc_con)
    if flag == 0;
        svc_pot(:,1) = svc_con(:,4).*svc_con(:,3)/basmva;
        % B_cv max on system base
        svc_pot(:,2) = svc_con(:,5).*svc_con(:,3)/basmva;
        % B_cv min on system base
        jsvc = bus_int(svc_con(:,2)); % bus number
        B_cv(:,1) = bus(jsvc,5)./(bus(jsvc,2).*bus(jsvc,2)); % initial B_cv
        bus_new(jsvc,5) = zeros(n_svc,1);
        testmxlmt=max( B_cv(:,1) > svc_pot(:,1));
        if testmxlmt
            error('SVC: BCV exceeds maximum at initialization')
        end
        testmnlmt=max(B_cv(:,1) < svc_pot(:,2));
        if testmnlmt
            error('SVC: BCV below minimum at initialization')
        end
        svc_pot(:,3) = B_cv(:,1); % store initial value of B_cv
        svc_pot(:,4) = bus(jsvc,2) + B_cv(:,1)./svc_con(:,6);% reference voltage
        svc_pot(:,5) = ones(n_svc,1);
        if ~isempty(svcll_idx)
            svc_pot(svcll_idx,5)= svc_con(svcll_idx,8)./svc_con(svcll_idx,9);
        end
        B_con(:,1) = B_cv(:,1).*{ones(n_svc,1)-svc_pot(:,5)}../svc_con(:,6);
    end
end
if flag == 1 % network interface computation
    % no interface calculation required - done in nc_load
end

if flag == 2 % exciter dynamics calculation
    % for linearization with operating condition at limits,
    % additional code will be needed
    lv_sbus=find(v_sbus<0.9 & svc_dsig(:,k)<0);
    d_sigin = svc_dsig(:,k);
    if ~isempty(lv_sbus)
        d_sigin(lv_sbus)=zeros(length(lv_sbus),1);
    end
    err = svc_sig(:,k) + svc_pot(:,4) + d_sigin - v_sbus;

    dB_con(:,k)= zeros(n_svc,1);
    if ~isempty(svcll_idx)
        nll = length(svcll_idx);
        dB_con(svcll_idx,k) = (-B_con(svcll_idx,k)+(ones(nll,1)-svc_pot(svcll_idx,5)).*err)./svc_con(svcll_idx,8);
    end
    dB_cv(:,k) = (-B_cv(:,k)+svc_con(:,6).*(err.*svc_pot(:,5)+B_con(:,k)))./svc_con(:,7);
    % anti-windup reset
    indmx = find( B_cv(:,k) > svc_pot(:,1));
    if ~isempty(indmx)
        B_cv(indmx,k) = svc_pot(indmx,1);
        indrate = find(dB_cv(indmx,k)>0);
        if ~isempty(indrate)
            % set rate to zero
            dB_cv(indmx(indrate),k) = zeros(length(indrate),1);
        end
    end
    indmn = find(B_cv(:,k) < svc_pot(:,2));
    if ~isempty(indmn)
        B_cv(indmn,k) = svc_pot(indmn,2);
        indrate = find(dB_cv(indmn)<0);
        if ~isempty(indrate)
            % set rate to zero
            dB_cv(indmn(indrate),k) = zeros(length(indrate),1);
        end
    end
end
end

```

Table 16 Data format for SVC

column	variable	unit
1	svc number	
2	bus number	
3	svc base MVA	MVA
4	maximum susceptance B_{cvmax}	pu
5	minimum susceptance B_{cvmin}	pu
6	regulator gain K_R	pu
7	regulator time constant T_R	sec
8	compensator lag time constant T_B	sec
9	compensator lead time constant T_B	sec
10	Fraction of bus B picked up by svc	

```

function f = svc_idx(svc_dc)
% determines the relationship between svc and nc loads
% checks for svc
% determines number of SVCS

f = 0; %Dummy Variable
global svc_con load_con n_svc svc_idx svcll_idx
n_svc = 0;
svc_idx = [];
if ~isempty(svc_con)
    [n_svc npar] = size(svc_con);
    svc_idx = zeros(n_svc,1); → [0] (b) 40,26
    % set defaults for lead lag
    if npar<9
        svc_con(:,8:9) = zeros(n_svc,2);
    end
    svcll_idx = find(svc_con(:,9)~=0); → SVC-CON 0/64 (JAN, 16-2010)
    for j = 1:n_svc
        index = find(svc_con(j,2)==load_con(:,1));
        if ~isempty(index)
            svc_idx(j) = index;
        else
            error('you must have the svc bus declared as a non-conforming load')
        end
    end
end

function [bus_new] = svc(i,k,bus,flag,v_sbus)
% Purpose: Static Var Compensator
% Input: k - integer time
%         bus - solved loadflow bus data
%         flag - 0 - initialization
%                 1 - network interface computation
%                 2 - generator dynamics computation
%         v_sbus - svc bus voltage
% Output: bus_new - on initialization bus_new is bus matrix
%                 with the reactive generation at the
%                 svc buses set to zero
%                 - otherwise bus_new = bus
global basmva bus_int
global svc_con n_svc svc_idx svcll_idx
global svc_pot B_cv dB_cv B_con dB_con
global svc_sig svc_dsig

```