

A.Sauhats A.Dolģicers

RELEJAIZSARDZĪBU VIRTUĀLĀS PĀRBAUDES

RTU EEF ENERĢĒTIKAS INSTITŪTS
RĪGA 2003

Ievads

Saturs

1. Relejaizsardzību virtuālās pārbaudes	4
1.1. Ievads	4
1.2. Virtuālās testēšanas organizēšana.....	5
1.3. Pārbaudi imitējošo elektrisko lielumu (signālu) ģenerācija.	7
1.4. Programmu pakete.....	8
1.5. Secinājumi.....	12
2. Virtuālā statistisko pārbažu metode.....	13
2.1. Ievads	13
2.2. Programmēšanas kļūdu konstatēšanas uzdevuma formulējums	14
2.3. Monte-Karlo metode un kļūdu meklēšanas procedūras realizācija	15
2.4. Ideālas ierīces darbības modelēšana	17
2.5. Procesu un iestatījumu uzdošanas procedūra.....	19
2.6. Piemēri ar programmu kļūdām, kas izsauc retas izpausmes atteices	22
2.7. Secinājumi.....	25

1. Relejaizsardzību virtuālās pārbaudes

Virtuālās pārbaudes metoda būtība. Testēšanas procesa organizēšana. Programmu pakete, procesu piemēri. Virtualā zinātniski-pētnieciskā laboratorija.

1.1. Ievads

Vispārīgā gadījumā sarežģītu kontroles un vadības mikroprocesorsistēmu pārbaudei nepieciešams radīt speciālu, dārgu laboratoriju un veikt ilgstošas, darbietilpīgas, augsti kvalificēta personāla izpildītas pārbaudes /85/.

Problēmu daļēji risina iebūvētie pašpārbaudes bloki, kas ļauj konstatēt bojājumu rašanos vairumā no aparātdaļas blokiem. Izstrādāti arī specializēti ar datoru vadāmi standi, kas nodrošina uzdotu strāvu un spriegumu izmaiņas procesu ģenerāciju un RA un A ierīču testēšanu /86/. Taču šādu standu izmaksas ir augstas, bet iegūto strāvu, spriegumu un loģisko signālu daudzums visai ierobežots. Sarežģītu aizsardzību pārbaudē, pat izmantojot datorvadāmus standus, rodas ievērojamas problēmas. Šīs problēmas vēl daudzkārt pieaug ierīču tehniskās izstrādes etapā. Pieredze rāda, ka sintezēto ierīču darbības pārbaudes stadijā bieži nepieciešams izdarīt korekcijas. Pie tam, pat nelielu programmatūras korekciju gadījumā, iespējamās kļūdas, kas parādās visneiedomājamākajās vietās. Jebkura izmaiņa, kas izdarīta programmatūrā, rada nepieciešamību pilnā apjomā atkārtot darbietilpīgo testēšanu. Dabiskā tieksme samazināt izdevumus uz testēšanas apjoma samazināšanas rēķina rada ievērojamas kļūdas. Var minēt piemērus, kad kļūdas parādās tikai pēc vairākiem gadiem, ekspluatējot desmitiem ierīču, un rezultātā rada ievērojamus zaudējumus, ko izsauc visu procesoru programmējamo atmiņu nomaiņas nepieciešamība. RA un A elektromehānisko un uz analogo pusvadītājelementu bāzes radīto ierīču struktūru veido paralēli strādājoši bloki, kas ļauj tos pārbaudīt atsevišķi, piemēram, atsevišķi pārbaudīt palaišanas un bloķēšanas blokus, atsevišķi pārbaudīt dažādu fāžu mērierīces, kā arī dažādu aizsardzības pakāpju mērierīces. Mikroprocesorierīces atsevišķus programmas blokus realizē pēc kārtas ar nobīdi laikā. Pie tam atsevišķu bloku pārbaude ir iespējama tikai apejot (izlaižot) nepārbaudāmos blokus. Šāda pārbaude ir nepilnīga.

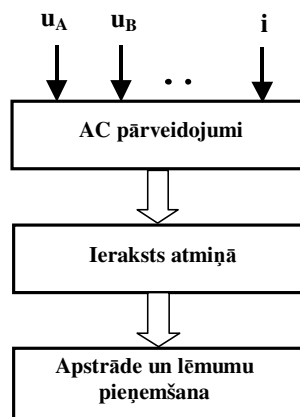
Ierīču izstrādes stadijā, lai atvieglotu visai darbietilpīgo programmu ievadīšanu mikroprocesorā, šīs programmas parasti modelē un skenē ar personālā datora palīdzību. Taču šis paņēmiens nespēj pilnīgi aizvietot reālās iekārtas pārbaudi.

Virtuālās testēšanas organizēšana

Tālāk parādīts, ka modernā dator tehnika ļauj ar visai maziem papildus izdevumiem atrisināt aizsardzību testēšanas problēmu mikroprocesorierīču projektēšanas stadijā, gandrīz pilnīgi automatizējot atkārtotās pārbaudes.

1.2. Virtuālās testēšanas organizēšana

Jebkuras mikroprocesorierīces darbības algoritmu var raksturot ar virknē slēgtiem blokiem (sk. Att. 1.1).

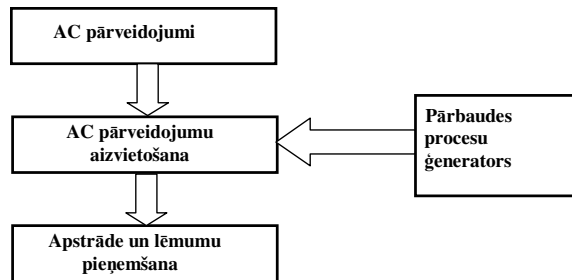


Att. 1.1 Mikroprocesorierīces darbības algoritma fragments

1. Ieejas strāvu un spriegumus analogciparu (ACP) pārveidotāja, kā arī ieejas loģisko signālu ievada bloku.
2. Signālu ierakstīšanu atmiņā.
3. Pārveidojumu un ievadīto rezultātu apstrādes programmu bloku.

Virtuālās testēšanas būtību raksturo pārbaudāmās ierīces funkcionēšanas algoritma papildinājums ar papildus programmu bloku, kas aizvieto iepriekšminētā pirmā bloka pārveidošanas rezultātus ar mikroprocesora atmiņā ierakstītiem pārbaudes procesu imitējošiem elektriskiem lielumiem. Acīmredzami, ka šāda testēšana ir piemērota programmatūras pārbaudei un ļauj noteikt gan tieši programmas kļūdas, gan aparātu atteices, kas izraisa nepareizu programmu izpildi. Pilnīgi netiek pārbaudīts pirmais no nosauktajiem blokiem. Taču šī bloka darbības pārbaude viegli izdarāma ar reālu, bet ļoti vienkāršu testu palīdzību: pārbaudot sinhronu strāvu un spriegumu pārveidojuma rezultātus.

Virtuālās testēšanas realizācijai nepieciešams (sk.Att. 1.2):



Att. 1.2 Virtuālo pārbažu algoritma fragments

1. Papildināt mikroprocesorierīces ar atmiņas bloku, kura apjoms atļautu uzglabāt pietiekama ilguma procesu raksturojošos elektriskos lielumus.
2. Nodrošināt procesu (to raksturojošo signālu) saņemšanu no ārējā avota un ierakstīšanu atmiņā.
3. Organizēt pārbaudi imitējošo procesu ģenerāciju un pārvadi.

Jāatzīmē, ka pirmo divu noteikumu realizācija nerada grūtības. Mūsdienīgas ierīces, kā likums, nodrošinātas ar atmiņu, kas paredzēta avārijas procesu signālu uzglabāšanai (ciparu oscilografēšana). Pārbažu laikā šo atmiņu vai tās daļu var izmantot testa signālu sēriju uzglabāšanai. Tajā pat laikā mikroprocesorierīces apgādā arī ar interfeisiem, kas realizē datu apmaiņas funkciju ar ārējiem datoriem/62/. Tādējādi, lai realizētu virtuālās pārbaudes, nepieciešams izstrādāt divu līmeņu atbalsta (papildus) programmas:

- Programmu ieejas signālu analogciparu pārveidojumu aizvietošanai ar personālā datora iepriekš izskaitļotām vērtībām (pārbaudes laikā signāli var līdzināties nullei). Šo programmu realizē mikroprocesorierīce un to var sintezēt tikai izstrādes stadijā.
- Programmu pārbaudi imitējošo procesu ģenerācijai.

Šādu programmu izstrāde praktiski nav saistīta ar testējamo iekārtu īpatnībām un tā veicama ievērojot konkrētā aizsargājamā objekta specifiku.

Pārbaudi imitējošo elektrisko lielumu (signālu) ģenerācija

1.3. Pārbaudi imitējošo elektrisko lielumu (signālu) ģenerācija

Pašreizējā augstu attīstītā elektromagnētisko un elektromehānisko pārejas procesu teorija, kā arī jau izstrādātā daudzveidīgā, energosistēmās un projektēšanas organizācijās plaši pielietotā programmatūra ļauj atrast pārbaudi imitējošo procesu raksturojošos elektriskos lielumus. Vairums šo programmu atrod strāvu un spriegumu lielumus vektoru veidā. Tieša šo vektoru izmantošana virtuālo pārbaudu veikšanai nav piemērota, taču tie viegli pārveidojami sinusoidālu signālu raksturojošās ciparu virknes veidā, ievērojot kvantēšanas (nolasījumu) periodu un iegūstot attiecīgu pakāpjveida (režģa) funkciju. Bez tam sinusoidas jāpārveido tā, lai tās atbilstu konkrētās testējamās ierīces analogciparu pārveidotajā izmantojamam kodam (skaitļošanas sistēmai, vērtīgo ciparu skaitam). Abi šie pārveidojumi ir elementāri un viegli realizējami. Ātrdarbīgu relejaizsardzību pārbaudēs var rasties nepieciešamība ģenerēt pārejas procesus ievērojot ne vien uzspiestās, bet arī brīvās rimstošās komponentes. Šai gadījumā zināmās programmas ataino procesus režģa funkciju veidā, kuras virtuālo pārbaudu veikšanai tālāk jāpārveido atbilstošā ACP kodā. Pie tam jāievēro, ka vairumam mikroprocesorierīču ir analogie ieejas filtri. Šie filtri slāpē augstākās harmoniskās, tie ir elementāri vienkārši gan struktūras ziņā (R-C ķēde), gan ietekmes modelēšanas nozīmē (inerciāls posms).

Pārbaudi imitējošo procesu elektriskos lielumus var iegūt arī citā veidā no ciparu oscilogrammām, ko ierakstījuši avārijas procesu mikroprocesorreģistratori vai pašas ciparu relejaizsardzības. Lai nodrošinātu oscilogrammu izmantošanas iespēju nepieciešams:

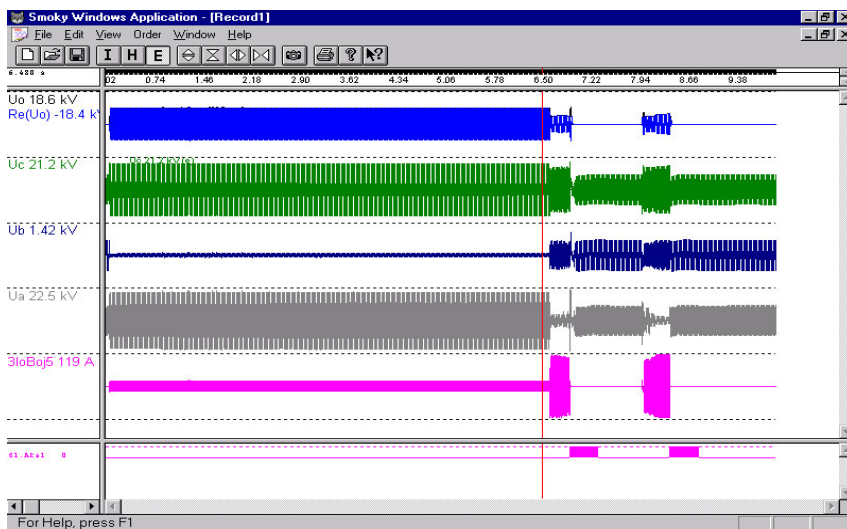
- Radīt iespēju pārveidot sākotnējo ierakstu izmainot, ja nepieciešams, tā mērogu, diskreditācijas frekvenci (piemēram, izmantojot interpolāciju).
- Nodrošināt analogo un loģisko signālu izmaiņas procesu papildināšanu un pārveidošanu.
 - Radīt interesanto un informatīvo laika intervālu "izgriešanas un salīmēšanas" iespēju, kas dod iespēju formēt daudzveidīgus, reti sastopamus, bet reālus procesus. Piemēram: īsslēguma rašanās tūlīt pēc jaudas slēdža ieslēgšanas, viena īsslēguma veida pāreja citā u.t.t.

Minēto pārveidojumu veikšanai nepieciešama atbilstoša programmatūra, kas varētu apstrādāt COMTRADE standartā (starptautiskais standarts digitālo oscilogrammu kodēšanai) dotās oscilogrammas.

Relejaizsardzības un automātikas ierīces visbiežāk pārbauda ar tradicionālajām metodēm, pievadot testējamai ierīcei noteiktus loģiskos signālus, strāvas un spriegumus, kuru vērtības iegūst ar regulējamiem strāvas un sprieguma avotiem. Acīm redzams, ka arī šādu pārbaudi var veikt virtuāli. Šim nolūkam jārada programmatūra, kas ierakstītu atmiņā operatora realizētos ieejas signālus.

1.4. Programmu pakete

Virtuālās laboratorijas programmu paketes pamatā ir Rīgas Tehniskajā universitātē izstrādātais ciparu oscilogrammu attēlošanas un apstrādes programmu komplekss SMOKY/88,89/, kas veidots ar darba autora līdzdalību. Šī kompleksa izvēli galvenokārt nosaka tas, ka Baltijas valstu energoobjektu lielākajā daļā uzstādīti REMI tipa reģistratori /98/, kuru oscilogrammas apstrādā izmantojot SMOKY programmas. SMOKY programma izvietota Enerģētikas un Elektrotehnikas fakultātes internetlapā: <http://www.eef.rtu.lv/remis> un ir pieejama energosistēmu darbiniekiem un studentiem. Tā rezultātā uzkrājusies visai plaša dažādu procesu bibliotēka, ko var izmantot testa signālu sintēzē. Lai nodrošinātu iespēju izmantot citu tipu ierīču COMTRADE standartā reģistrētās oscilogrammas, papildus uzstādīts translators no COMTRADE formāta, kā arī nodrošināta jau iepriekš minēto “izgriešanas un salīmēšanas” funkciju izpilde. Vairāku dažādos laika intervālos fiksētu oscilogrammu sintēzes piemērs parādīts Att. 1.3. attēlā. Testa process atbilst vienfāzīga īsslēguma pārejai divfāžu īsslēgumā uz zemi un nesekmīgas automātiskās atkalieslēgšanas pārejai divkāršā īsslēgumā uz zemi.

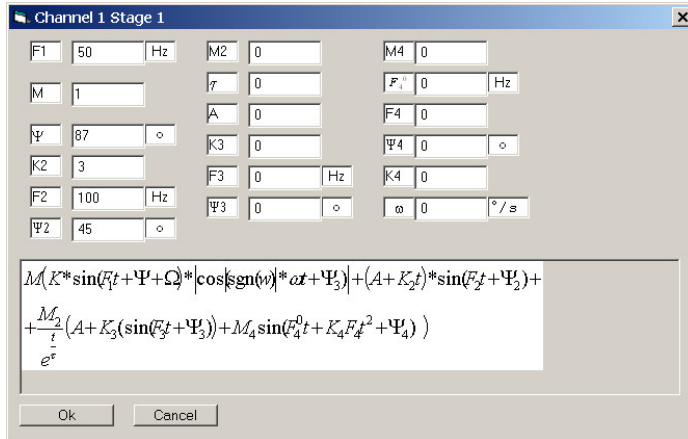


Att. 1.3 Dažādos laika intervālos fiksētu oscilogrammu sintēzes piemērs

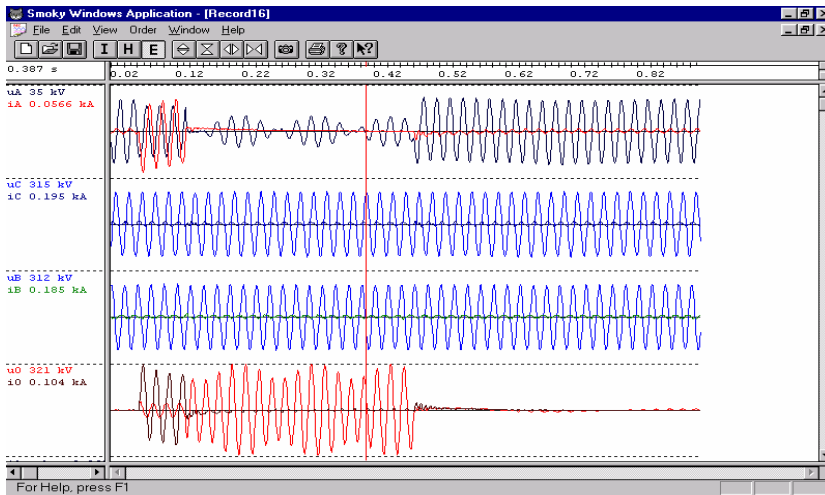
Lai izmainītu signālus vai to harmoniskās sastāvdaļas laikā, radīts speciāls interfeiss. Tas ļauj uzdot signālus (harmoniskās) sarežģītu laika funkciju veidā. Interfeisa logs dots attēlā

Programmu pakete

4.4. Pie tam nav nepieciešama programmēšanas valodu izmantošana. Rodas plašas iespējas uzdot procesus secīgos laika intervālos. Tā Att. 1.5 dots testa procesa ieraksts, kurš ilustrē sarežģītu testa procesu radīšanas iespēju.



Att. 1.4 Sarežģīta signāla formēšanas interfeiss

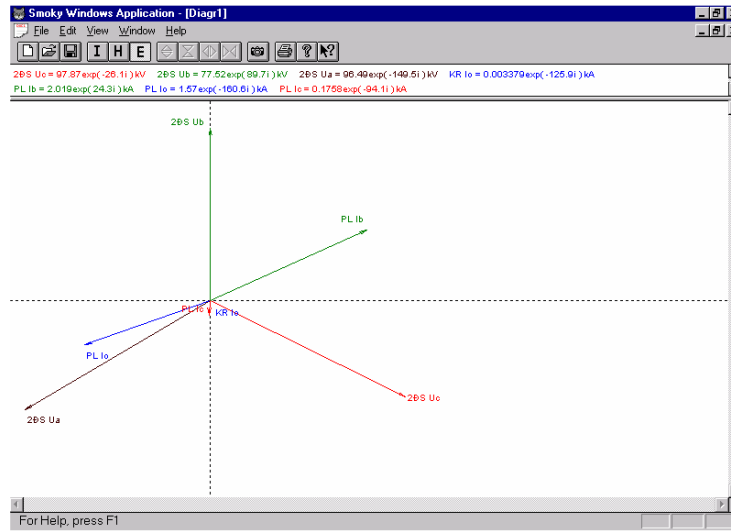


Att. 1.5 Sarežģīta testa procesa piemērs

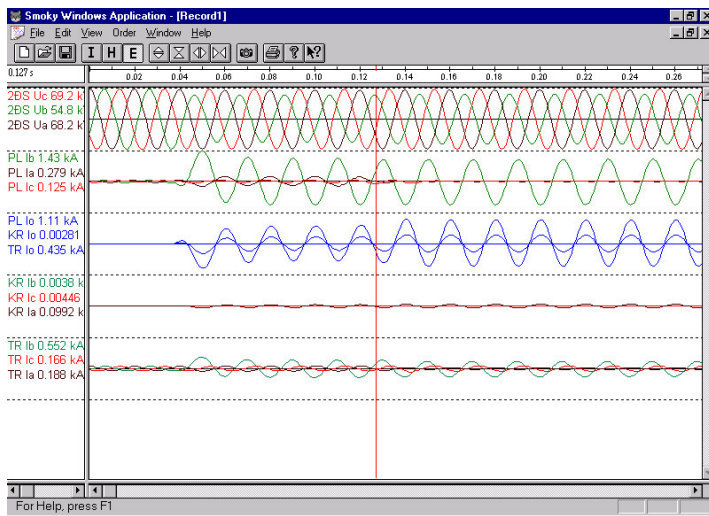
Lai izmantotu dažādus energosistēmu modeļus (šos modeļus aprakstošās programmas), radīts interfeiss, kas ļauj modeļos iegūstamos kompleksos elektriskos lielumus pārveidot oscilogrammu formātā. Attiecīgi attēlā Att. 1.6 parādīta ar energosistēmas modeļa

Relejaizsardzību virtuālās pārbaudes

palīdzību iegūtā strāvu un spriegumu vektoru diagramma vienfāzes īsslēguma gadījumā, bet attēlā Att. 1.7 tās pašas strāvas un spriegumi pēc translācijas oscilogrammas formā.



Att. 1.6 Vektoru diagramma vienfāzes īsslēguma gadījumā



Att. 1.7 Vienfāzes īsslēguma strāvas un spriegumi

Programmu pakete

Periodisku atkārtotu pārbaūžu (testu) veikšanai nepieciešams radīt datu bāzi, kas noteiktu testu secību un pārbaudāmās ierīces paraugreakciju uz šiem testiem.

1.5.Secinājumi

1. RA un PA mikroprocesorierīču darbības algoritmu īpatnības un aparātu struktūra atļauj pielietot virtuālo pārbažu metodi. Pie tam, atkrīt nepieciešamība izmantot sarežģītus un dārgus pārbažu standus.
2. Virtuālo pārbažu metode var izmantot reālo vai izrēķināto, izmantojot rūpnieciskas programmu paketes, procesu ierakstus.
3. Virtuālo pārbažu metode aprobēta sarežģīto mikroprocesorierīču pārbaudes gaitā: kompleksās augstsprieguma līniju aizsardzības REDI , asinhronās gaitas novēršanas automātikas AGNA un ct.

2. Virtuālā statistisko pārbaūžu metode

Mikroprocesoru tehnikas īpašības veicina programmu sarežģītības pieaugumu. Programmas kļūdu meklēšanas grūtības. Kļūdu meklēšanas algoritma struktūrshēma. Monte-Karlo metode un kļūdu atrašana. Nepieciešamais pārbaūžu skaits. Programmas kļūdu piemēri.

2.1. Ievads

Augstāk jau atzīmēts, ka uz mikroprocesoru tehnikas bāzes izveidotajām energosistēmu relejaizsardzības un automātikas ierīcēm ir virkne pozitīvu īpašību un priekšrocību. Tomēr šīs ierīces sev līdzī atnesa arī problēmas, kas sakņojas mikroprocesortehnikas pozitīvajās īpašībās:

1. Funkcionēšanas algoritma uzdošana programmas veidā. Tieši šis paņēmiens veicina viena mērķuzdevuma ierīču variantu un modifikāciju ātru maiņu. Ierīcēs tiek ievestas arvien jaunas, visbiežāk otršķirīgas, servisa funkcijas, paplašinātas iespējas.
2. Augstāzīgu mikroprocesorelementu zemas cenas. Šī īpašība veicina agrāko atsevišķo ierīču funkciju integrāciju.
3. Informācijas apstrādes ciparu paņēmiens. Šī paņēmiena priekšrocības salīdzinājumā ar analogo ir vispārzināmas un tiek lietderīgi izmantotas.
4. Iespēja viegli veikt iestatījumu uzdošanas un maiņas procedūru, iestatījumu izmantošana algoritma struktūras izmaiņai (loģiskās funkcijas, atsevišķu funkciju ievēšana vai izslēgšana).

Zemāk tiks parādīts, ka atzīmētās īpašības var radīt nopietnas grūtības vērtējot aizsardzības un automātikas ierīču efektivitāti. Grūtību pamatā ir pārbaūžu sarežģītība vērtējot ierīces piemērotības pakāpi paredzēto prasību nodrošināšanai. Mikroprocesortehnikas pārbaūžu nepieciešamību diktē divi galvenie iemesli:

1. Aparatūras atteices;
2. Programmatūras kļūdas.

Mikroprocesortehnikas aparatūras atteices, pateicoties ražošanas tehnoloģijas sasniegumiem, ir samērā reta parādība, kas samērā viegli identificējama ar speciāliem līdzekļiem un metodēm /51/. Turpretim programmas kļūdas atrast grūti /85/. Citos gadījumos veiksmīgi pielietojamais kļūdu atrašanas paņēmiens – pārbaudes ekspluatācija nav piemērots relejaizsardzībai, jo tās darbībā nostrādes pieprasījuma plūsmas intensitāte ir necīgi maza. Ātrā programmatūras variantu maiņa vēl vairāk pasvīturo pārbaudes ekspluatācijas nepiemērotību programmas kļūdu meklējumos. Atliek vispusīgu, sarežģītu, dārgu un darbietilpīgu laboratorijas pārbaūžu veikšanas ceļš. Zemāk parādīts, ka funkciju integrācija, informācijas programmas-ciparu apstrādes paņēmienienu lietojums un lielais iestatījumu skaits sarežģī un sadārdzina ierīču pārbaudes.

- ◆ Funkciju integrācija noved pie sarežģītākām programmām. Realizējot programmatūras produktus, diemžēl jāsecina, ka kļūda jauniestatītajā modulī var traucēt citu, jau pārbaudīto, moduļu darbību.
- ◆ Informācijas apstrādes ciparu paņēmieni lielā mērā pakļauti nelineāriem efektiem, īpaši kļūdu gadījumos. Analogās tehnikas struktūrās bija daudz lineāro bloku, kuru fizikālās būtības dēļ bija pietiekami veikt pārbaudi tikai dažos to raksturlīkņu punktos. Izmantojot ciparu principus rodas pārpildījuma efekti, notiek operācijas pa bitiem, tiek realizēta aritmētika ar uzdoto bitu skaitu, kas situāciju vēl vairāk pasliktina. Var rasties kļūdas ar ļoti retu izpausmi.
- ◆ Iestatījumu skaita pieaugums pie pārbaudēm rada papildprasības. No teiktā izriet, ka pārbaudot ciparu ierīces funkcionēšanu jāveic pārbaudes ar dažādām iestatījumu kombinācijām un palielinoties iestatījumu skaitam šo kombināciju skaits strauji aug.

Meklējamās programmēšanas kļūdas var dalīt divās grupās:

1. Tiešās programmas kļūdas. Šo kļūdu rezultātā tiek izjaukts programmētāja iedomātais algoritms.
2. Algoritma kļūdas. Izstrādājot relejaizsardzības un automātikas ierīces rodas dažādas detalizētības pakāpes algoritmi.

Vairumā gadījumu sākotnēji izstrādā izejas algoritmu, kurā operē ar relejaizsardzības klasiskajiem jēdzieniem un kategorijām: strāvas, sprieguma, pretestības mērelementu nostrāde, to atgriešanās, loģiskās funkcijas, nostrādes un atgriešanās kavējumi utt. Tādu algoritmu efektivitāti un racionalitāti daudzu gadu laikā un daudzos gadījumos apstiprinājusi ekspluatācijas prakse un daudziem kvalificētiem ekspertiem pierastā terminoloģija. Realizētais un programmējams algoritms operē ar AC-pārveidojumu rezultātiem, komandām un operācijām, kas definētas konkrētajam procesoram un citiem elementiem, kas ietilpst programmējamajā aparātā. Starp šiem diviem nosauktajiem algoritmu tipiem ir liela atšķirība. Otrais ir daudzskārt sarežģītāks, tā izstrādes gaitā iespējamas kļūdas, kas novedīs pie sākotnēji iepļānotā algoritma neizpildes. Tāda tipa kļūdas kopā ar pašas programmas kļūdām jāatrod. Tālāk vadīsimies no pieņēmuma, ka izejas algoritms ir bez kļūdām un tādēļ nav to meklēšanas uzdevuma.

2.2. Programmēšanas kļūdu konstatēšanas uzdevuma formulējums

Relejaizsardzības ierīču funkcionēšana atkarīga:

- No kontrolējamo procesu gaitas.
- No programmā noteiktas kontrolējamās procesus atspoguļojošu ciparu virknējumu (strāvu, spriegumu AC-pārveidojumi, loģisko signālu vērtības) apstrādes procedūras.
- No iestatījumu vērtību kopuma.

Pie kam ierīcēm izvirza divas galvenās prasības:

1. Nostādāt pie nostrādes pieprasījuma.
2. Nenostrādāt pie nenostrādes pieprasījuma.

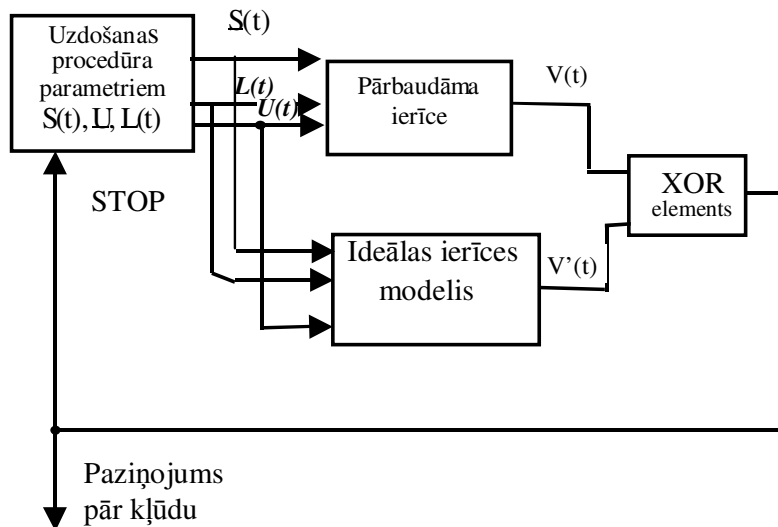
Tiek uzskatīts, ka pārbaudāmās ierīces iestatījumi izvēlēti saskaņā ar nostrādes un nenostrādes prasībām. Minēto prasību neizpildes gadījumā rodas atbilstošas atteices:

- ◆ nostrādes atteice (pie nostrādes pieprasījuma);
- ◆ lieka nostrāde (pie pretējās prasības).

Relejaizsardzības ierīču kontrolētie procesi atbilst varbūtības procesu klasei, jo tos nosaka daudzi gadījumrakstura faktori (bojājuma veids un vieta, attālinājums, slodze, tīkla konfigurācija u.c.).

Pārbaudāmās ierīces funkcionēšanas programma un iestatījumi tiek doti projektēšanas procesā un parasti ir determinēti. Tomēr teiktais atbilst patiesībai tikai tad, ja nav programmas kļūdu. Ja tādas ir, tad komandu izpildes secība un sastāvs, kā arī iestatījumi var būt atkarīgi no kontrolējamajiem procesiem. Tādēļ kļūdu konstatēšanas uzdevumā visu trīs faktoru grupas, kas nosaka pārbaudāmās ierīces funkcionēšanu, var tikt uzskatītas kā gadījumrakstura. Pie kam, saskaņā ar identifikācijas teorijā pieņemto terminoloģiju / /, pārbaudāmo ierīci var uzskatīt par nezināmas struktūras objektu ar novērojamām ieejām un izejām. Bet kļūdu meklēšanas uzdevumu var pārstāvēt attēlā 5.1 parādītā procedūra.

Monte-Karlo metode un kļūdu meklēšanas procedūras realizācija



Att. 2.1. Kļūdu meklēšanas procedūras struktūrshēma

$\underline{S}(t)$ – kontrolējamie analogie procesi;

$L(t)$ – kontrolējamie loģiskie procesi;

\underline{U} - iestatījumu kopa;

$V(t), V'(t)$ – pārbaudāmās un ideālās ierīces izejas signāli.

Attēlā Att. 2.1 parādītās procedūras praktiskai izmantošanai svarīgākie ir kontrolējamo procesu $\underline{S}(t)$ un iestatījumu \underline{U} izveidošanas procedūras jautājumi, kā arī “ideālas ierīces” izstrādāšana. Atzīmēsim arī, ka elements “izslēdzošais VAI”, kas ģenerē kļūdas paziņojumu, ir vienkāršots, tā kā reāli iespējamās pilnībā pieļaujamas laika novirzes pie signālu $V(t)$ un $V'(t)$ parādīšanās. Zemāk aplūkota signālu $V(t)$ un $V'(t)$ nepieļaujamas atšķirības atpazīšanas reāla procedūra.

2.3. Monte-Karlo metode un kļūdu meklēšanas procedūras realizācija

$\underline{S}(t)$ un \underline{U} uzdošanas procedūrai jānodrošina tādu realizāciju $\underline{S}(t)$ un \underline{U} kombināciju meklēšana, kas noved pie atteices parādīšanās. Tātad jārealizē procesu un iestatījumu realizācijas variantu kombinēšanas procedūra. Tādas procedūras izveidošanai eksistē divi galvenie varianti:

1. Izmantojot deterministisko pieeju.
2. Ar statistisko pieeju.

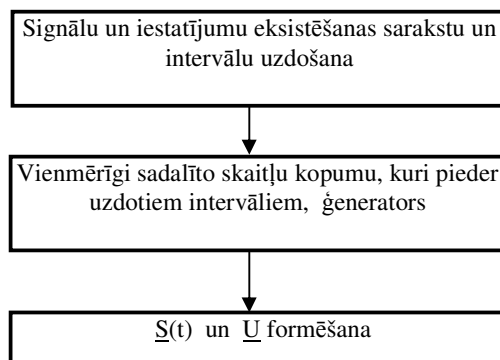
Daudzmēru uzdevumos otrajai pieejai, kas izveidota balstoties uz Monte-Karlo metodes, ir būtiskas priekšrocības /46/, tā nodrošina ātrāku un drošāku atrisinājumu /46/.

Meklējuma procedūru (gadījumrakstura kombinēšana) var izveidot divējādi:

1. Kontrolējamos procesus $\underline{S}(t)$ uzdodošo gadījumrakstura parametru laukā. Atsevišķā gadījumā variē spriegumu un strāvu amplitūdas un fāzes. Atteices šajā laukā visbiežāk meklē aizsardzības un automātikas ierīču laboratorijas pārbaužu laikā.
2. Aizsargājamo objektu modeļu parametru laukā.

No realizācijas darbietilpīguma un kļūdu meklēšanas efektivitātes viedokļa abi ceļi ir aptuveni līdzvērtīgi. $\underline{S}(t)$ un \underline{U} uzdošanas algoritmu struktūrshēmas parādītas attēlā 5.2 un attēlā 5.3. Abos gadījumos izmanto pieņēmumu par gadījumrakstura parametru vienmērīga sadalījuma likumu.

Atzīmēsim, ka aplūkojamajā uzdevumā šāds pieņēmums ir pilnīgi korekts, jo mērķis ir atrast procesu un iestatījumu jebkuru kombināciju, pie kuras parādās kļūda*.



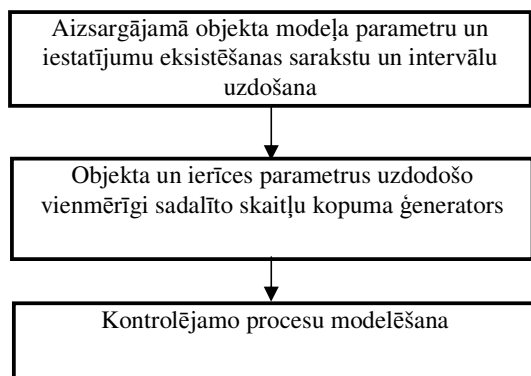
Att. 2.2 Kļūdu meklējums signālu parametru laukā

Organizējot aplūkojamās procedūras rodas jautājums par pārbažu skaitu, kas ar iepriekš uzdotu ticamību garantē atteižu konstatēšanu.

Pieņemam, ka atsevišķas pārbaudes laikā atteice var izpausties ar varbūtību Θ .

Atradīsim varbūtību, ka n neatkarīgu pārbažu rezultātā atteice parādīsies vienu reizi.

Att. 2.3 Kļūdu meklējums objekta modeļa parametru laukā



Parbaudes izpilda līdz pirmās atteices faktam. Šajā gadījumā pārbažu skaitam X , kas jāveic līdz pirmajai atteicei, būs sadalījums saskaņā ar Fari likumu (Paskāla sadalījuma likuma īpašs gadījums /95/). Tad:

$$P(X; 1; \Theta) = \Theta(1 - \Theta)^x, \quad (2.1)$$

kur

$P(X; 1; \Theta)$ – atteices parādīšanās varbūtība pie X -tās pārbaudes. Atradīsim varbūtību, ka atteices konstatēšanai nepieciešams ne vairāk par X pārbaudēm:

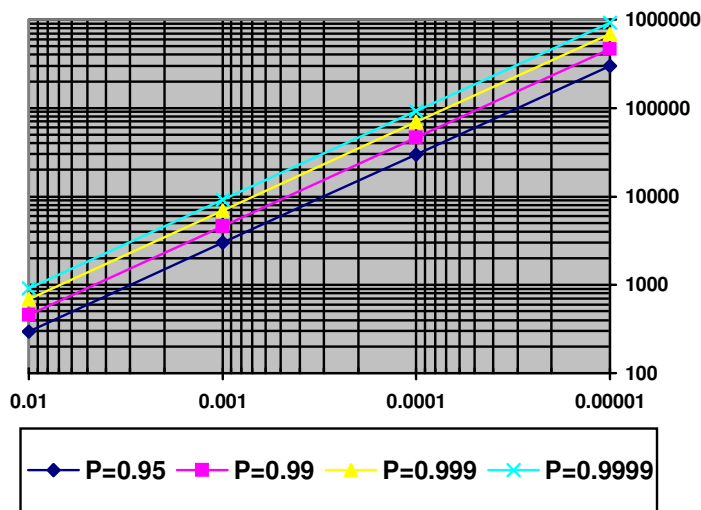
* Vērtējot aizsardzības ierīču efektivitāti tiek aprēķinātas atteižu varbūtības. Šajos gadījumos pieņēmums par vienmērīgā sadalījuma likumu var radīt lielas kļūdas.

Ideālas ierīces darbības modelēšana

$$P(X < x) = \sum_{i=0}^{x-1} \Theta(1 - \Theta)^i. \quad (2.2)$$

Organizējot pārbaudes loģiska ir sekojoša jautājuma nostādne:

- izvēlēties pārbaudu skaitu X , pie kura pat reti eksistējošas atteices varētu konstatēt ar varbūtību, kas tuva 1. Atkarība X no P un Θ , kas atbilst tādai nostādnei, parādīta attēlā Att. 2.4.



Att. 2.4 pārbaudu skaitu X atkarība no P un Θ

Ticamības P un varbūtības Θ vērtības izvēle jāveic vadoties no pārbaudu izmaksām un ekonomiskās situācijas, kas radīsies izlaižot produkciju ar programmas kļūdām (tām parādoties jāveic visu ražoto ierīču programmatūras nomaiņa, rodas zaudējumi sakarā ar ražotājfirmas prestiža kritumu). Aptuvenu varbūtības Θ var pieņemt balstoties uz prasības, ka programmas kļūdu radītās atteices varbūtībai jābūt mazākai par aparatūras radītās atteices varbūtību. Tādējādi, pieņemot, ka moderno mikroprocesorierīču darbumūžs T ir 25 gadi, var atrast (pieņemot, ka atteices rodas saskaņā ar eksponenciālo likumu) atteices varbūtību viena ekspluatācijas gada laikā

$$P_{omk} = 1 - e^{-\frac{1}{T}} = 0,039, \quad (2.3)$$

ko var pieņemt par pamatu uzdodot Θ izteiksmē (2.1).

2.4. Ideālas ierīces darbības modelēšana

Programmas kļūdu meklēšanai saskaņā ar attēla 5.1 struktūrshēmu tikai tad ir jēga, ja droši un nekļūdīgi realizēts ideālas ierīces modelis. Tādēļ šī modeļa realizācijas programmai jābūt daudz vienkāršākai par pārbaudāmo mikroprocesorrealizāciju. Šo prasību var izpildīt izmantojot izejas algoritmus un projektēšanas tehnisko uzdevumu. Pie kam ierīces darbības programmēšanai jāizmanto augsta līmeņa valodas. Personālajam datoram paredzētās programmas var izstrādāt izmantojot dubultās precizitātes aritmētiku un neekonomējot programmas un operatīvo atmiņu, atkrīt nepieciešamība vadīt daudzus elementus un blokus, kas ietilpst mikroprocesorierīces sastāvā. Tādēļ programmas, kas modelē izejas algoritmus, izrādās relatīvi vienkāršas un tās

viegli rediģēt un pārbaudīt. Atzīmēsim, ka daudzos gadījumos algoritmisko kļūdu konstatēšanai mikroprocesorierīču projektēšanas procesu sāk ar modelējošās programmas radīšanu un tās darbības efektivitātes pārbaudi ar aizsargājamā objekta ciparu modeļa palīdzību. Šajā gadījumā organizējot statistiskās virtuālās pārbaudes var izmantot gatavus programmu blokus, kas modelē izejas algoritmu.

2.5. Procesu un iestatījumu uzdošanas procedūra

Virtuālo pārbažu organizēšanai nepieciešamā procesu un iestatījumu uzdošanas procedūra pirmkārt atkarīga no pārbaudāmās ierīces mērķuzdevuma, kontrolējamajiem signāliem un tādēļ to var izstrādāt un programmēt tikai konkrētām ierīcēm. Zemāk aplūkotas tādu procedūru izveidošanas īpatnības divām mikroprocesorierīcēm:

110-330 kV elektropārvades līniju kompleksai aizsardzībai un automātiskai REDI;
asinchronā režīma novēršanas un likvidācijas automātiskai AGNA /101/.

Atbilstoši divu veidu prasībām, ko izvirza relejaizsardzības un automātikas ierīcēm, aplūkojamo procedūru dalīsim divās daļās, kas ģenerē procesus:

- ◆ Izsaucošus nostrādes pieprasījumus.
- ◆ Nostrādi nepieprasošus.

Dalījums ir nosacīts, jo iespējamas procesu pārejas no vienas klases uz otru. Aplūkots procedūras uzdošanas gadījums parametru laukā, kas uzdod kontrolējamus procesus.

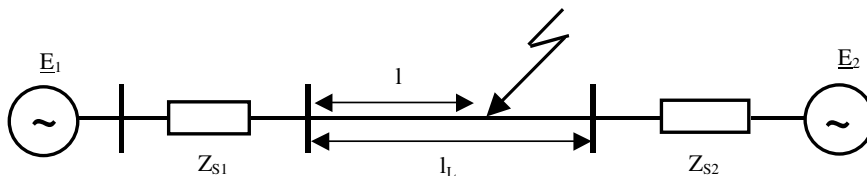
Aplūkojamās ierīces funkciju analīze un nostrādes pieprasījuma situācija ļauj izdalīt dažus raksturīgus laika periodus:

1. Pirmsavārijas (slodzes) režīms. Režīmu raksturo strāvu un spriegumu simetrija. Zināmi procesu sinusoidālu parametru eksistences diapazoni.
2. Avārijas režīms
3. Bezstrāvas pauze
4. Automātiskā atkalieslēgšana (AAI)
5. Pēcavārijas režīms

Režīms	Pirmsavārijas režīms	Avārijas režīms	Bezstrāvas pauze	Automātiskā atkalieslēgšana (AAI)	Pirmsavārijas režīms
	Avārijas režīms				
Pārbaudāmās funkcijas	Palaišanas bloks		Slēdža atteices aizsardzība		
	Bloķēšana pie sprieguma ķēžu bojājumiem		Mērīšanas elementu atgriešanās		
	Bloķēšana pie svārstībām		AAI loģika Atslēgšanas paātrinājums pēc nesekmīgas AAI		
	Mirusās zonas novēršana		Sinhronisma uztveršana		
	Strāvas un distantie mērīšanas elementi		Loģisko signālu vadības funkcijas		
	Loģisko signālu vadības funkcijas		Procesu un notikumu reģistrācija, signalizācija		
	Laika kavējumi				
	Pakāpju nostrādes laiks				
Procesu un notikumu reģistrācija, signalizācija					

Tabula 5.1 Aplūkojamās ierīces funkciju laika periodi

Tabulā 5.1 parādīto procesu modelēšanu var veikt izmantojot vienu no praktiski ekspluatējamiem programmu kompleksiem, kas modelē elektromagnētiskos un elektromehāniskos pārejas procesus /87/. Tomēr universālu programmu izmantošana rada nepamatoti lielus skaitļošanas laika patēriņus. Skaitļošanas laika samazināšanai un RA un PA ierīču pārbažu realizēšanai izstrādātas vienkāršotu shēmu (parādītas attēlos Att. 2.5 un Att. 2.6) elektromagnētisko un elektromehānisko pārejas procesu modelēšanas programmas.



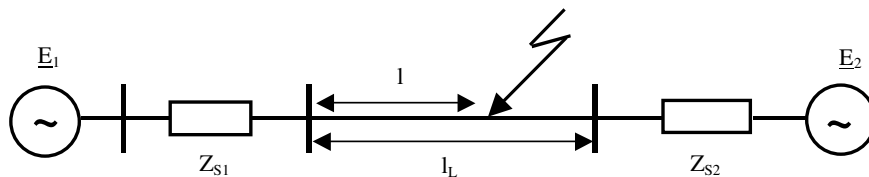
Att. 2.5 Ierīču REDI pārbaudei izmantotā tīkla shēma

Attēlā 5.5 parādītā shēma nodrošina procesu modelēšanu režīmiem, kas parādīti tabulā 5.1. Kā gadījumparametri pieņemti: energosistēmu ekvivalento EDS vektoru \underline{E}_1 , \underline{E}_2 , \underline{E}_3 , energosistēmu ekvivalentās pretestības \underline{Z}_{S1} un \underline{Z}_{S2} (tiešās, pretsecības un nulsecības), attālums līdz bojājuma vietai, līniju parametri, mijinduktivitātes pretestības, bojājuma veids u.c. Pieņemts vienmērīgais gadījumparametru sadalījuma likums, bet tā parametrus (eksistences intervālu) pēc līniju un tīklu parametru statistisko datu analīzes uzdod lietotājs.

Procesu un iestatījumu uzdošanas procedūra

Procesu modelēšanā izmantoti plaši pazīstamie vienādojumi un simetrisko komponentu metode /87/.

Attēlā Att. 2.6 parādītā energosistēmas divmašīnu ekvivalentā shēma izmantota asinhronā režīma novēršanas un likvidācijas automātikas AGNA pārbaudei.

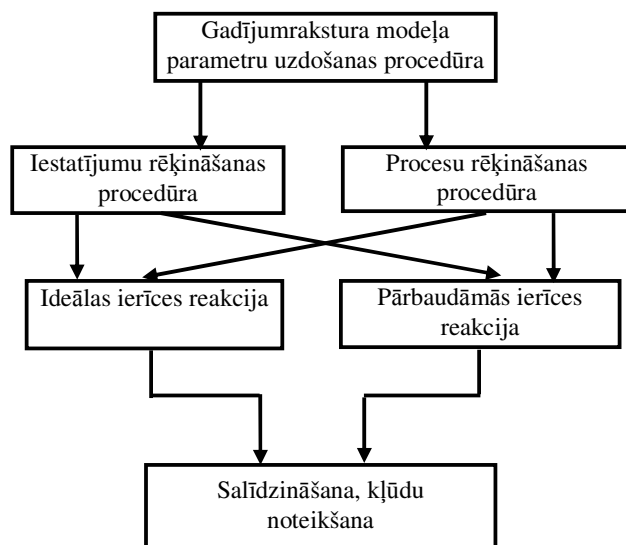


Att. 2.6 Tīkla shēma, ko izmanto ierīces AGNA pārbaudei

Šī shēma ļauj modelēt slodzes režīmu, trīsfāžu īsslēgumu un vienkāršotu asinhrono režīmu.

Asinhronā režīma vienkāršota modelēšana veikta balstoties uz hipotēzes par divu frekvenču stacionāru režīmu, kad energosistēmas E_1 ģeneratori rotē ar frekvenci ω_1 , bet energosistēmas E_2 – ar frekvenci ω_2 . Šajā gadījumā aprēķini kļūst vienkāršāki un tos var veikt ar superpozīcijas metodi. Par gadījumperimetriem pieņemti visi attēlā 5.6 parādītie parametri, frekvences ω_1 un ω_2 kā arī pieņemti vienmērīgā sadalījuma likumi.

Attēlā 5.7 parādīti virtuāli-statistisko pārbažu veikšanas programmas galvenie bloki. Attēlos 5.5 un 5.6 parādīto vienkāršāko shēmu gadījumos iestatījumu izvēles procedūras izveidošana nerada principiālas grūtības.



Att. 2.7 Virtuāli-statistisko pārbažu veikšanas algoritma fragments

Ideālās un pārbaudāmās ierīces darbības rezultātu salīdzināšanas procedūra attēlā Att. 2.7 atbilst elementam “izslēdzošais VAI” attēlā Att. 2.1 Īstenībā rodas nepieciešamība izmantot sarežģītāku

divu ierīču funkcionēšanas rezultātu salīdzināšanas procedūru. Tādai nepieciešamībai ir sekojoši iemesli:

1. Ideālās un pārbaudāmās ierīces reakcijas (nostrādes, atgriešanās) var nebūt vienlaicīgas. Atšķirību ievērošana veidojot aplūkojamo procedūru parasti grūtības nerada, jo zināmi maksimālie (maksimāli pieļaujamie) nostrādes un atgriešanās laiki.
2. Abu ierīču reakcijas var atšķirties pie kontrolējamo procesu tuvināšanās robežai starp nostrādes un nenostādes nosacījumiem. Šajā gadījumā fiksējot reakciju atšķirību pietiek pārbaudīt nosacījumu attālinājumu no atbilstošajām robežām. Attālinājumu, kurā pieļautas atšķirības, var uzdot vadoties no tehniskajiem nosacījumiem.

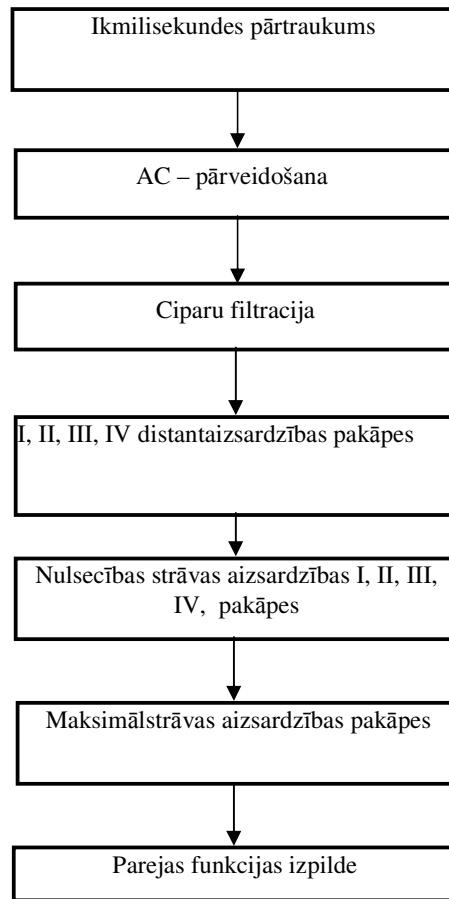
2.6. Piemēri ar programmu kļūdām, kas izsauc retas izpausmes atteices

Zemāk aplūkosim divu tipu kļūdas, kas konstatētas projektējamo ierīču pārbažu beigu stadijā. Abos gadījumos kļūdas konstatētas veicot virtuālās statistiskās pārbaudes un abos gadījumos programmas kļūdas parādījās pie noteiktām, reti sastopamām kontrolējamo procesu un iestatījumu kombinācijām.

1. Piemērs. 110-330 kV EPL kompleksā aizsardzība un automātika REDI. Attēlā Att. 2.8 parādīts ierīces funkcionēšanas algoritma fragments, saskaņā ar ko:

- ik pēc milisekundes notiek programrealizēts pārtraukums, kas noved pie visu kontrolējamo signālu AC-pārveidošanas un iegūto rezultātu ciparu filtrēšanas;
- notiek laikā secīga visu distantaizsardzību pakāpju, maksimālstrāvas (virzītās, vai sprieguma ķēžu bojājuma gadījumā nevirzītās), nulsecības strāvas nostrādes nosacījumu pārbaude. Visas šīs pārbaudes veic laika intervālos starp pārtraukumiem. Konstatējot nostrādes nosacījumus pakāpei vai zonai ar mazāku numuru atkrīt, protams, nepieciešamība pārbaudīt nostrādes nosacījumus pakāpēm ar augstāku numuru. Programmētāja uzdevums bija dotās milisekundes robežās nodrošināt visas nepieciešamās pārbaudes visu nepieciešamo datu sagatavošanu. Šajā nolūkā tika veikta nepieciešamo komandu izvēle un uzskaitē, to izpildes laika aprēķins, lai nodrošinātu to izpildi atvēlētajā laika diapazonā.

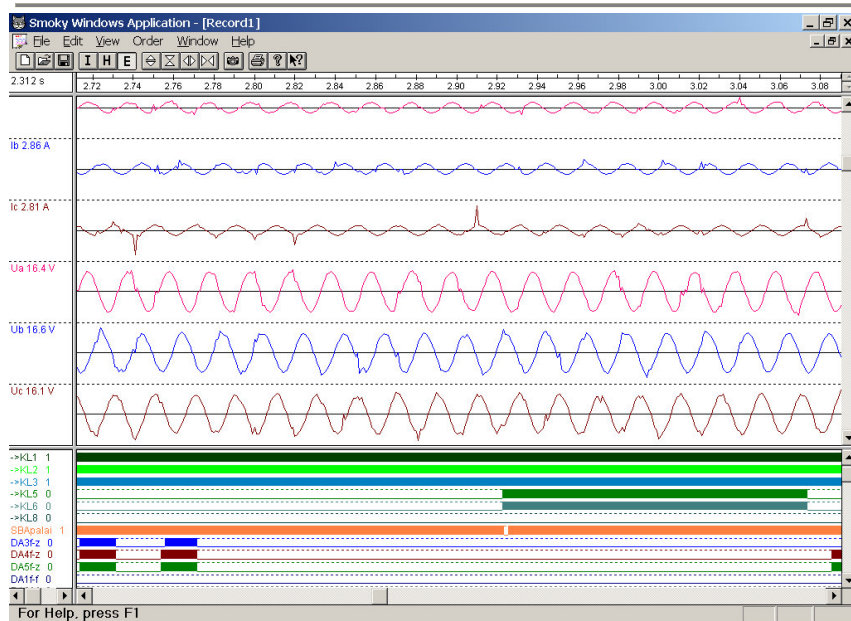
Piemēri ar programmu kļūdām, kas izsauc retas izpausmes atteices



Att. 2.8 110-330 kV EPL kompleksās aizsardzības un automātikas funkcionēšanas algoritma fragments

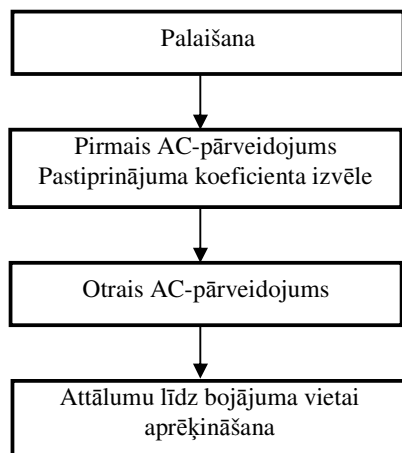
Lielā zarošanās skaita dēļ uzdevums izrādījās visai sarežģīts.

Pieļautās kļūdas būtība bija apstākļi, ka nebija ņemts vērā viens režīms – procesu atbilstība vienlaicīgi piektajai zonai distantaizsardzībai pret īsslēgumiem uz zemi, maksimālstrāvas aizsardzības (nevirzītajai, ko ievēd pie sprieguma ķēžu bojājumiem) ceturtajai pakāpei un nulsecības strāvas aizsardzības ceturtajai pakāpei. Pie distantaizsardzības zonu lieliem laika kavējumiem izrādījās, ka procesors nepaspēj sakārtot datus, kas nepieciešami iepriekšējās milisekundes aprēķiniem (pārtraukums tika izpildīts pirms beidzās visu datu sagatavošana). Pie pārtraukumiem atliktās operācijas krājās, kas pie lieliem laika kavējumiem noveda pie atteicēm. Pie kam atteices radās programmas daļā, kas šķita labi nostrādāta un neatkarīga no citām programmas daļām – ACP vadībā. Traucētā AC-pārveidojuma procesa rezultātu oscilogramma parādīta attēlā 5.9.



Att. 2.9 Traucētā AC-pārveidojuma oscilogramma

Skaidri redzami signāla kropļojumi, ko radījusi nolasījumu “izkrišana”. Redzamas distanta aizsardzības 3., 4., 5. zonas liekas nostrādes. Tomēr ciparu filtrācijas rezultātā kropļotie procesi tika izlīdzināti un atteices izpaudās tikai gadījumos, kad procesi atbilda darbības zonai tuvu izvietotiem punktiem.



Att. 2.10 Mikroprocesorfiksatora funkcionēšanas algoritma fragments nosakot attālumu līdz bojājuma vietai

Tādējādi pieļautās kļūdas izpaušmei bija nepieciešama sekojošu nosacījumu sakritība:

- ievērojami aizsardzību laika kavējumi;
- procesu atbilstība zonām ar maksimālajiem numuriem;
- pretestību un strāvu vektoru novietojums tuvu zonas robežām;
- bojājums sprieguma ķēdēs.

Šādu nosacījumu rašanās varbūtība pārbaudes ekspluatācijas laikā ir ļoti maza. Veicot statistiskās virtuālās pārbaudes atteice parādījās pēc apmēram tūkstoš pārbaudēm.

2. piemērs. Mikroprocesorindikator, kas fiksē attālumu līdz bojājuma vietai /21, 22, 23/. Attēlā Att. 2.10 parādīts ierīces funkcionēšanas algoritma fragments, saskaņā ar kuru:

Secinājumi

- tiek veikta katra signāla divkārtēja AC-pārveidošana. Pie kam pie pirmās pārveidošanas signālu ACP pievada caur pastiprinātāju ar mazu pastiprinājuma koeficientu. Notiek signāla lieluma rupjš novērtējums, tiek izvēlēts pastiprinājuma koeficients, pie kura ACP pievadāmais signāls ir tuvs maksimāli pieļaujamam un notiek atkārtots, precīzs AC-pārveidojums. Aprēķinos izmanto AC-pārveidojuma rezultāta reizinājumu ar pastiprinājuma koeficientu. Aprakstītais divkārtējais pārveidošanas paņēmiens ļāva paaugstināt pārveidošanas precizitāti signāliem ar lielu dinamisko izmaiņas diapazonu.
- realizē ciparu filtrāciju un aprēķina attālumu līdz bojājuma vietai balstoties uz signālu parametru un iestatījumu novērtējumiem /22/.

Kļūdas būtība bija nepareizi izvēlēts kārtu skaits veicot dažas aritmētiskās darbības (tika izmantota veselo skaitļu aritmētika /45/). Izrādījās, ka pie dažām signālu vērtību un iestatījumu kombinācijām izpildot aritmētiskās darbības rodas pārpildījums. Tādēļ skaitļošanas rezultāti tika rupji izkropļoti. Interesanti atzīmēt, ka kropļojumi radās pie signālu vērtībām to eksistences diapazona vidū.

Aplūkotais piemērs interesants ar to, ka apstiprina sekojošu faktu. Desmitiem aparātu ar aprakstīto kļūdu tika veiksmīgi ekspluatēti energosistēmās (kļūda neizpaudās).

2.7.Secinājumi

1. Mikroprocesoru elementu bāzes īpašības un tehniskie raksturojumi veicina RA un PA programmu sarežģītības pieaugumu, funkciju apvienojumu, iestatījumu skaita pieaugumu un, kā sekas, rada programmas kļūdu meklēšanas problēmu.
2. Kļūdu meklēšanu var realizēt salīdzinot pārbaudāmās un ideālās (programceļā realizētās) ierīces reakcijas.
3. Pārbaudes procesa automatizēšanai piedāvāts izmantot Monte-Karlo metodi. Dota pārbaudu skaita izvēles metodika.
4. Izstrādāti vienkāršoti energosistēmas modeļi ar autora līdzdalību izstrādāto sarežģīto mikroprocesorierīču pārbaudēm ar virtuālās statistiskās pārbaudes metodi:
5. REDI – kompleksai RA un PA ierīcei, kas realizē elektropārvades līniju visu veidu relejaizsardzību un automātiku.
6. AGNA – kompleksai RA un PA ierīcei, kas realizē asinhronā režīma novēršanas un likvidācijas funkcijas.
7. Sarežģītu RA un PA ierīču izstrādē reāli pieļautu kļūdu ietekmes un izpausmes nosacījumu analīze apstiprināja ilgstošu un visaptverošu pārbaudu veikšanas nepieciešamību.

3. Literatūras saraksts

- [86] P. Bonanni, E. Carta, L. Valtriani. Automatic line protection testing: SDH equipment with optical fibre amplifier systems, IEEE IMTC' 94, May, Hamamatsu.
- [87] Лосев С.Б., Чернин А.Б. Расчет электромагнитных переходных процессов для релейной защиты на линиях большой протяженности. –М. Энергия, 1972.
- [88] A.Sauhatas, T.Loman, A.Utan, A.Dolgicer, "A disturbance recording system", International Symposium on Modern electric power system, Wroclaw, Poland, september 1996. p.p 314-316
- [89] А. Саухатас, А. Долгицер, А. Утанс, Г.Пашнин, В.Шабанов. Система регистрации аварийных электроэнергетических процессов Всероссийская научно-техническая конференция Информационные и кибернетические системы управления и их элементы. Уфа 1997 .p.p 252-253.