برنامه‌ریزی منطقی برای تشخیص عیب در شبکه‌های حلقوی توزیع

محمدق سجودی
مجتمع سازندگی و آموزش آذربایجان

چکیده :

اولین هدف در عملکرد یک سیستم توزیع تأمین نیروی مصرفی مشترکین و برآوردن بار درخواستی آنان میباشد. با توجه به اینکه احتمال عیب در شبکه‌های ولتاژ بالا و شبکه‌های ولتاژ فوق العاده بالا ، پائین است ، احتمال وقوع عیب در شبکه‌های توزیع نسبتا" زیاد است. وقتی در یک شبکه توزیع خطائی اتفاق می‌افتد باید فورا" تشخیص داده شده و رفع گردد تا نیروی مصرفی بتواند هر چه سریعتر اصلاح شود. این مقاله یک مدل منطقی را برای تشخیص معایب در شبکه‌های حلقوی توزیع نشان میدهد. متدها در روی مدل برای نمایش ترکیب شبکه ، معایب ، علائم معایب و نارسائیها به کار گرفته میشود. قوانین مربوط به عیب‌یابی از روی روابط بین خطاها و علائم مربوطه توسعه یافته‌اند که شامل اثرات نهائی ناشی از عیوب باتریها و علائم مدار باز سیستمهای حفاظتی فیدرها میباشد. یک برنامه منطقی که بر مبنای مدل منطقی توسعه یافته است به وسیله یک زبان مقدماتی که در جهت هدف میباشد اجرا شده و توانائی سیستم برنامه منطقی برای تشخیص عیب از طریق کاربرد آن روی شبکه توزیع ۲۰ کیلوولتی به نمایش گذاشته شده‌است.

شرح مقاله :

وقتی یک خطایی در یک شبکه حلقوی توزیع اتفاق می‌افتد ، باید فورا" محل

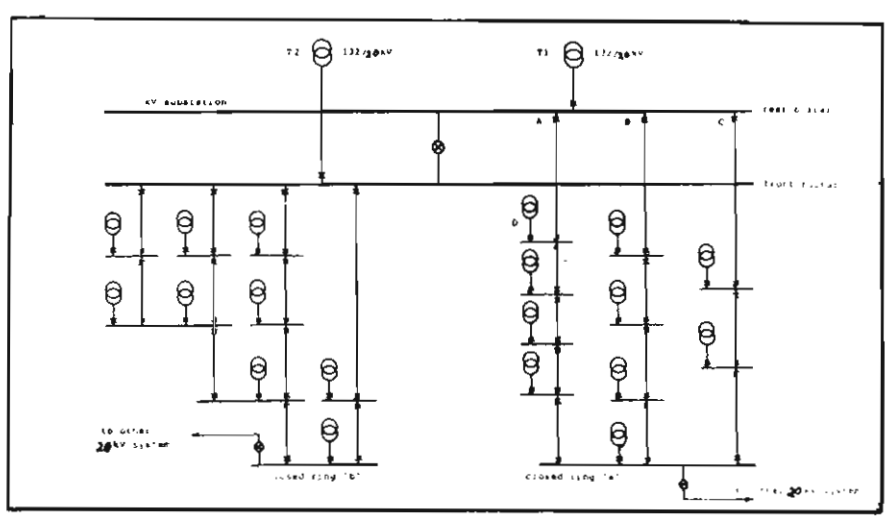
عیب تشخیص داده شده و از شبکه ایزوله گردد، تا اینکه وضعیت بار مصرفی اصلاح شده و زمان قطعی برق کاهش یابد. مهندسین عملیات معمولا" در تشخیص عیب از روی معلومات و گزارشات علایم خطا که در اثر آنها کلیدها قطع شده‌اند، اقدامات خود را شروع میکنند. مهندسین از روی علائم ظاهر شده و با بکارگیری عملیات استدلالی محل خطا را تعیین میکنند. با وجود اینکه این روش رضایت بخش‌است ، عملیات استدلال انسانی همیشه موفق نیست و بعضی عوامل مانند نواقص باتریها وسیم‌های حامل اطلاعات ممکن است در محدوده عملیات مزبور نباشد، لذا زمان عیب یابی طولانی میشود. یکی از روش های کاهش زمان عیب‌یابی ، بکاربردن روش استفاده از جدول است . در این روش ، جدول علایم خطاهای مختلف تعیین شده و فهرست‌وار نوشته میشود. اما وقتی تمام معایب ممکن و نارسایی‌هایی که به نظر میرسند جمع شوند، جدولها گسترده و مختلط خواهند شد ، بنابراین هنوز عملیات استعداد پذیرش خطاهای انسانی را دارد. بعلاوه باید جدولهای مختلفی برای ترکیب شبکه‌های مختلف بوجود آید . در عمل توپولوژی شبکه توزیع به خاطر تعیین شبکه مکررا" تغییر می‌یابد و تعداد زیادی از ترکیب‌های مختلف به وجود می‌آید. در این روش جدولهای زیادی برای تحت پوشش قرار دادن تمام ترکیب‌های ممکن شبکه‌های مختلف نیاز هست . لذا عملی نمی باشد. اخیرا" روش اساسی و تخصصی اسلوب‌شناسی سیستم برای عیب‌یابی شبکه‌های ولتاژ خیلی بالا و برای سایر قسمتهایی مانند نقشه‌کشی سیستم قدرت ، کنترل توان راکتیو ، سطح ولتاژ و کنترل عملیات آلارمهای اتاق فرمان به کار برده شده است . اما کاربرد روشهای برنامه‌ریزی منطقی (کامپیوتری) برای عیب‌یابی در شبکه‌های حلقوی توزیع مورد توجه قرار نگرفته است . این مقاله براساس تحقیقات برنامه‌ریزی منطقی یک مدل منطقی را برای تشخیص معایبی که ممکن است در شبکه‌های توزیع حلقوی اتفاق افتد ، ارائه میدهد. مدل شامل نمایش توپولوژی‌های شبکه، مؤلفه‌های شبکه، معایب، نارسائیها و علائم مربوطه میباشد. با بکار بردن استدلال مؤثر و مدلل ، قوانین تشخیص عیوب فیدر ، عیوب شینه‌ها و عیوب ترانسفورماتورها توسعه داده میشوند و به وسیله زبان مقدماتی کامپیوتری مناسب اجرا میگردد. برنامه مزبور میتواند در یک روش مستقیم برای تشخیص معایب از روی علایم معلوم بکار برده شود و همچنین از آن میتوان به روش معکوس نیز برای یافتن علایمی از یک خطای معین استفاده نمود. فایده و توانائی نرم‌افزار مورد بحث از طریق کاربرد آن برای مسائل عیب‌یابی در یک شبکه توزیع ۲۰ کیلوولتی حلقوی معلوم می‌گردد.

۱ - معایب ، علایم و نقص‌ها :

شبکه‌های حلقوی توزیع بسته ، به طور نرمال و به خاطر اطمینان بـه تـغذیه‌آن برای تـغذیه قسمت اعـظم منـاطق مسکـونی و مـراکز اصلی بـار پـذیرفته شده است. (شکـل۱) تـرکیب مـدار حـلقوی ۲۰ کـیلوولتی را نشان مـیدهد . تـوان از طـریق ترانسفورماتورهای قدرت ۱۳۲/۲۰ کیلوولتی T_1 و T_2 به ترتیب به شبکه‌های حـلقوی a و b تحویل میشود. هر مدار حلقوی مـیتواند دو ، سه یـا چهـار مـدار فـیدر مـوازی داشته باشد و هر مدار فیدر به کمک سیم‌های حامل اطلاعات حفاظت میشود. هر مـدار موازی همچنین توسط یک رله جریان زیاد حفاظت میشود. رله جریان زیاد روی شیـنه ۲۰ کیلوولتی پست نصب میشود. توان از طریق تـرانسفورماتورهای تـوزیع مـوجود در مدارهای حلقوی به مصرف کننده تحویل میشود.

۱-۱- انواع عیب و علایم :

انواع مختلف عیبها که ممکن است در شبکـه‌های حـلقوی تـوزیع اتفـاق افـتد، خطاهای شینه ، فیدر و ترانسفورماتور مـیباشد. وقـتی در شبکـه a در شکـل (۱) یک عیب شینه اتفاق می‌افتد ، از علایم آن قطع کلیدها در محلهای A ، B و C به خـاطر عملکرد رله‌های جریان زیاد در پست مـیباشد. وقـتی یک خطـا در یک فـیدر اتفـاق می‌افتد ، کلیدهای طرفین فیدر به طور نرمال محل عیب را از سیستم جدا میکنند، بـه عنوان مثال تحت شرایط نرمال یک خطای ترانسفورماتور در مـحل D در شکـل (۱) از طریق بـاز شدن کلید ، ترانسفورماتور از سیستم جدا خواهد شد. علایم خطاهایی که بـا بـاز شدن کلید ظاهر میشوند معمولا" نتیجه مستقیم اتفاق افتادن خطاها درشبکه‌است.



شکل ۱- پست ۲۰ کیلوولت وشبکه‌های حلقوی

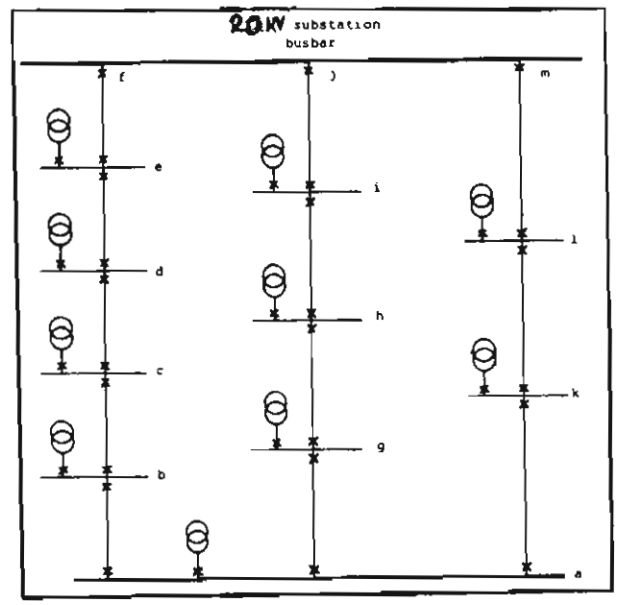
در تحقیقات سیستم به روش اساسی علایم ناشی از معایب با استفاده از یک دسته روش‌های مختلف تجزیه و تحلیل میکردند. این علایم با قوانین و تجارب ماخوذه از حوادث اتفاق افتاده قبلی و یا با ترکیبی از آنها تطبیق داده میشود، سپس معایب بوجودآمده مجددا" با معایب دیگر مقایسه و تجزیه و تحلیل میشود. عملیات موقعی پایان می‌یابد که عده‌ای از خطاهای مورد نظر پیدا شده‌اند . یکی از مشکلات این روش تحقیق آن است که قوانین و روش‌ها در آن بینهایت پیچیده و مختلط میتواند باشد، اگر خطاهای پنهانی داشته باشیم. با توجه به روش مراجعه به جدول ، معلوم میشود که این تحقیق به تعداد زیادی از قوانین برای استخراج تمام خطاهای اتفاق افتاده ممکن و ترکیبات علایم نیازمنداست .

۲-۱- نقص‌ها :

ممکن است نواقصی از قبیل اشکالات باتری و قطع سیم‌های حامل اطلاعات در سیستم شبکه بدون هیچگونه علامت وآلارمی وجود داشته باشد. وقتی در یک شبکه خطایی وجود دارد ، خطاها به صورت یک گروه علایم ظاهر میشوند که وضعیت را با حالت سالم متفاوت میکند. در شبکه نشان داده شده در شکل (۲) اگر یک خطای اتصال کوتاه در فیدر cd باشد و یک عیب باتری در شینه c باشد، کلید فیدر در c باز نمیکند ، در حالیکه مدارشکن (کلید) در d روی فیدر cd باز خواهد کرد. این باعث میشود که رله‌های جریان زیاد در محل‌های j و m برای فرمان دادن به کلیدها عمل کنند. اگر در شینه c نقص باتری موجود باشد ، عیب ترانسفورماتور در شینه c از سیستم ایزوله نخواهد شد، یعنی عیب به صورت پنهانی به سیستم وصل خواهد بود. وقتی سیم حامل اطلاعات مربوط به سیستم حفاظتی فیدر gh باز شده باشد، یک خطا روی فیدر cd به خوبی منجر به قطعی کلیدهای محل c و d میشود، تا اینکه کلیدهای روی فیدر gh قطع شوند. یعنی باز شدن سیم حامل اطلاعات فیدر gh مانع از باز شدن کلیدهای آن فیدر خواهد شد.

۲ - نمایش‌ها :

روش‌های عیب‌یابی در بخش ۴ توضیح داده خواهد شد. در این بخش خلاصه مختصری از هماهنگی زبان پرولاگ (PROLOG) داده شده و نحوه اجرای دستورات آن بیان میشود ، و این دستورات با روش‌های توسعه برای نمایش شبکه‌ها ، معایب ، نقص‌ها و علایم دنبال میشود.



شکل ۲- یک نمونه از شبکه حلقوی توزیع

۱-۲- دستورات پرولاگ (PROLOG) و نحوه اجرای آنها :

این زبان که اینک مطرح است توسط بخش کامپیوتر دانشگاه ادینبرک دنبال میشود. یک دستور برنامه PROLOG به شکل زیر میباشد:

$$A : - B_1, \ldots B_n . \qquad (1)$$

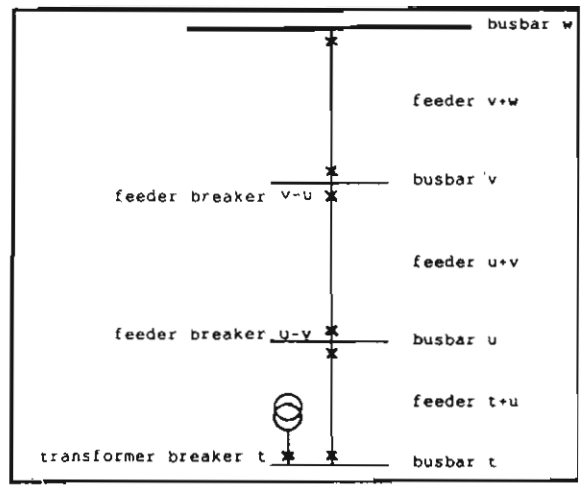
که A و B پیش فرض‌ها و مسندات هستند و سمبل - : معنی "اگر" میدهد . پیش فرض A عنوان (سر) موضوع نامیده میشود و پیش فرض‌های طرف راست علامت (-:) نیز شامل بدنه موضوع است . معنی دستور (۱) این است که : اگر تمام موضوعات B درست باشد ، پس عنوان موضوع A درست است . موضوع A شامل یک یکسان کننده است که با یک عامل سطح پایین یک لیست از مراحل ابتدایی را به عنوان مباحث شروع میکند. دستور (۱) گاهی به عنوان یک قانون مطرح میشود. اگر بدنه دستور خالی باشد پس همواره سرموضوع (A) درست خواهد بود و دستور یک واقعیت نامیده میشود. در مقابل یک دستور بدون سر یک هدف (GOAL) نامیده میشود. مقصود از اجرای یک برنامه PROLOG یافتن یک عده از متغیرهایی هست که هر کدام یک هدف درست است . طی مدت اجرا، متغیرهای درست از اهداف کوچک موضوعات B در زمان ظهورشان مورد ارزیابی قرار میگیرند. در یک برنامه هر کدام از اهداف B با یک دستور دیگر و با همان سر توسط یکسان ساز یکی میشوند.

یکسان سازی روشی است که در آن تعدادی از متغیرها که با یکدیگر مساوی هستند پیدا میشوند. وقتی که روش یکسان سازی موفقیت آمیز است ، اهداف B توسط بدنه دستور یکسان سازی جابجا میشوند ومتغیرهای آن معرفی میشوند. عمل ارزیابی در قالب دستور یکنواخت‌سازی به پیش فرض‌های اهداف اعمال میشود. این رویه تکرار میشود تا اینکه پیش فرض‌های اهداف در دستور هدف اصلی ارزیابی شده باشد و آن به عنوان استراتژی تحقیق اولین عمق اطلاق میشود. اگر روند یک هدف اشتباه باشد، هدف ماقبل آن باید فورا" مجددا" ارزیابی شود قبل از اینکه روند ارزیابی ادامه یابد . این رویه ارزیابی روش رد گم کردن (BACKTRACKING) نام دارد. تحقیق اولین عمق ، رد گم کردن و یکنواخت کردن مکانیسم‌های اصلی اجرای زبان PROLOG هستند . این مکانیسم‌ها و خواص آنها کاملا" در آخرین توسعه نمایش شبکه‌ها و عمومی کردن عیب‌یابی روش PROLOG در این مقاله بکار گرفته شده است.

۲-۲- نمایش شبکه :

در برخی از امور اخیر نمایش یک شبکه ، هم ارتباط انفرادی واقعیات و هم یک لیست از گره‌های متصل به هم مورد استفاده قرار میگیرد. این نمایش طولی است در تعریف توپولوژی شبکه خطای انسانی وارد شود. یک تحقیق جدید برای نمایش شبکه‌ها و مؤلفه‌های شبکه ذیلا" با بسط و توسعه ارائه شده است. شینه در یک شبکه حلقوی با گره‌ها نمایش داده شده‌اند و مدارهای موازی فیدر با ساقه‌هایش نمایش داده شده‌اند . هر ساقه شامل یک توالی گره‌هاست. گره‌ها در یک ساقه در یک خط راست با لبه‌ها به هم دیگر متصل میشوند. یک فیدر در یک شبکه با یک کناره تعیین میشود که با عبارت u + v نشان داده میشود ، مطابق شکل (۳) سمبل ذره‌ای هر u و v برای نمایش گره‌ها مورد استفاده قرار میگیرد، در حالیکه عمل + برای نشان دادن اتصال بین دو گره بکار میرود. ساختمان شبکه کامل با واقعیات ساقه‌ها در شبکه نمایش داده شده است . کاربرد روش فوق برای نمایش سه ساقه ۲۰ کیلوولتی شبکه حلقوی شکل (۲) با حقایق ذیل تعریف میشود :

(۲a)	شبکه	(a+b+c+d+e+f)
(۲b)	شبکه	(a+g+h+i+j)
(۲c)	شبکه	(a+k+l+m)



شکل ۳- نمایش شبکه و مؤلفه‌ها

رله‌های جریان زیاد با کره نشان داده میشود. برای تشخیص توسط رله‌های جریان زیاد از شینه(باس‌بار) حقایق زیادی لازم میشود. واقعیت‌های اضافی برای شبکه شکل (۲) اینها هستند.

(۳a) رله جریان زیاد (f)

(۳b) رله جریان زیاد (j)

(۳c) رله جریان زیاد (m)

در محل هر باس‌بار دو کلید فیدر و یک کلید ترانسفورماتور وجود دارد. یک کلید ترانسفورماتور توسط نام کره باس‌بار مربوطه نشان داده میشود که به آن متصل است و یک کلید فیدر با u - v نشان داده میشود. بیان u - v بیشتر نشان میدهد که کلید در پایان u از فیدر u + v قرار داده شده است شکل (۳). این روش نمایش را نشان میدهد.

۳-۲- نمایش خطا، نقص‌ها و علایم :

خطاهای فیدر با نام‌های فیدرها که خطا در کناره آنها اتفاق افتاده است نمایش داده میشود . برای مثال a + b یک خطای فیدر را معرفی میکند که روی فیدر میان گرههای a و b اتفاق افتاده است . یک خطا روی باس‌بار به وسیله کره باس‌بار خطا نمایش داده میشود. یک باتری ناقص بنام کره باس‌باری که در آن قرار دارد نامیده میشود. یک سیم حامل اطلاعات باز شده در سیستم حفاظتی

یک فیدر با نام دو کناره اتصالی مجاور نمایش داده میشود. بنابراین یک خطای سیم حامل اطلاعات بصورت g + h یعنی سیم حامل اطلاعات بین گره‌های g و h باز شده است . علائم با نام‌های مدار شکن‌ها (کلیدها) و رله‌های جریان زیاد که عمل کرده‌اند و زمین میشوند در یک لیست علامت تعریف میشوند . لیست علامت [c-b,j,m] ، برای مثال نشان میدهد که کلید c - b و رله‌های جریان زیاد j , m در شبکه شکل (۲) عمل کرده‌اند.

۳ - فرموله کردن تشخیص عیب :

اساس فرموله کردن تشخیص عیب قوانینی است که ارتباط خطاها و علائم ناشی از آنها را بیان میکنند. در قوانین عیب‌یابی توسعه یافته ذیل فرض شده است که فقط یک خطای تکی ممکن است در یک زمان اتفاق افتاده باشد. آشار یک عیب تک ناشی از باتری و یک خطای تکی ناشی از باز شدن سیم حامل اطلاعات مشمول این روش است.

۳-۱- خطاهای فیدر :

تحت شرائط طبیعی ، یک خطا روی یک فیدر منجر به قطعی کلیدهای ترمینالهای فیدر میگردد . یک دستور، زبان PROLOG ، معرف خطای فیدر به شکل زیر بیان میشود:

```
feederfault(X + Y,S) : - innet(X + Y),only(S,X - Y,Y - X).        (۴)
```

در دستور (۴) پیش فرض innet نشان میدهد که فیدر X + Y ، در شبکه تحت آزمایش است . اگر درست باشد، پس پیش فرض only نشان میدهد که فقط علایم ، آنها هستند که در لیست علامت S موجودند . برای یک خطا روی فیدر X+Y علایم باید فقط قطعی کلیدهای X - Y و Y - X باشد . اما این دستور باطل است وقتی که یک اشکال باتری هم روی گره X و هم روی گره Y موجود باشد بدون توجه به اینکه آیا قطعی سیم راهنماهست یا نه . بنابراین دستور (۴) به صورت زیر تغییر می‌یابد :

```
feederfault(X + Y,S,B,P) : - innet(X + Y),only(S,X - Y,Y - X), B = nil.
                                                                  (۵)
```

که در آن علامت = یکی بودن عبارات طرفین آن علامت را نشان می‌دهد. سمبل‌های B و P جانشین خطای باتری و خطای سیم حامل اطلاعات میباشند.

۱-۱-۳- گنجایش خطای سیم حامل اطلاعات :

با یک خطای سیم حامل اطلاعات باز شده در سیستم حفاظتی فیدر u + v ، یک خطای روی فیدر X + Y باعث میشود کلیدهای X - Y,Y - X,U - V و V - U قطع شوند و بدین ترتیب وجود باتریها در کره U و V الزام دارد. دستور نقص برای این حالت به شرح زیر است :

```
defect(X + Y,S,B,P) : - innet(P),not match(P,X + Y),P = U + V,
member(U - V,S) , member(V - U,S), B = nil.                    (۶)
```

جائیکه پیش فرض not match بررسی میکند که فیدر با یک خطای سیم حامل اطلاعات معرفی شده با مقدار P ، فیدر خطای X + Y نیست و سیم حامل اطلاعات U + V است . پس تعداد پیش فرض‌ها تائید میکند که قطعی کلیدهای u - v و v - u در لیست علایم S است .وقتی که یک اشکال باتری در پایان U در اثر نقص سیم حامل اطلاعات u - v وجود دارد، کلید u + v پایدار میماند . دستورکلی عیب به صورت زیر است :

```
defect(X + Y,S,B,P) : - innet(P),not match(P,X + Y),P = U + V,
B = U , B\ = = X , B\ = = Y , member(V - U,S).                 (۷)
```

در حالی که = = / معرفی میکند که جمله‌ها در دو طرف آن برابر نیستند . به همان ترتیب آثار خرابی باتری در کره V به وسیله دستور زیر بدست می‌آید :

```
defect(X + Y,S,B,P) : - innet(P),not member(P,X + Y),P = U+ V,
B = V , B\ = = X , B\ = = Y , member(U - V,S).                 (۸)
```

استفاده از سمبل ، برای "or" دستورات (۶) و (۸) میتواند ترکیب شده و داشته باشیم :

```
defect(X + Y,S,B,P) : - innet(P),not match(P,X + Y),P = U + V,
((member(U - V,S) , member(V - U,S));
(B = U , B\ = = X , B\ = = Y , member(V - U,S);                (۹)
(B = V , B\ = = X , B\ = = Y , member(U - V,S))).
```

روش پیدا کردن یک خطای فیدر با آثار یک نقص باتری و از طرف دیگر یک عیب سیم حامل اطلاعات میتواند به صورت زیر باشد:

```
feederfault(X + Y,S,B,P) : - innet(X + Y), member(X - Y,S),
defect(X + Y,S,B,P).                                          (۱۰)
```

از ترکیب دستورات خطای فیدر در (۵) و(۱۰) داریم :

```
feederfault(X + Y,S,B,P) : - innet(X + Y),
((only(S,X - Y,Y - X) , B = nil),
(member(X - Y,S) , member(Y - X,S),                    (11)
dfect(X + Y,S,B,P))).
```

۲-۱-۳- خرابی باتری در یک خروجی فیدر :

هر فیدری که رویش خطا اتفاق افتاده باشد ، ممکن است خرابی باتری نیز داشته باشد . بر اثر یک خطا روی فیدر c+d در شکل (۲) و نیز خرابی باتری در گره C ، کلید d - c باز میکند وکلیدهای j و m نیز به دلیل ایجاد شدن جریان خطا در مدارهای jc و mc یا عملکرد رله‌های جریان زیاد j و m باز میکنند. علایم در نقاط j و m با روش تحقیق گرهی که تحقیق در باره گرههای شبکه را از گره C شروع میکند بدست می‌آید. که این متد در قسمت بعدی توضیح داده میشود. روش PROLOG برای یافتن یک خطای فیدر توام با خرابی باتری در انتهای آن فیدر مطابق دستور زیر میتواند بکار برود:

```
feederfault(X + Y,S,B,P) : - innet(X + Y), cases (X,Y,X1,Y1),
member(X1 - Y1,S) , B = Y1 , nodesearch(Y1,[X1],S,B,P).    (12)
```

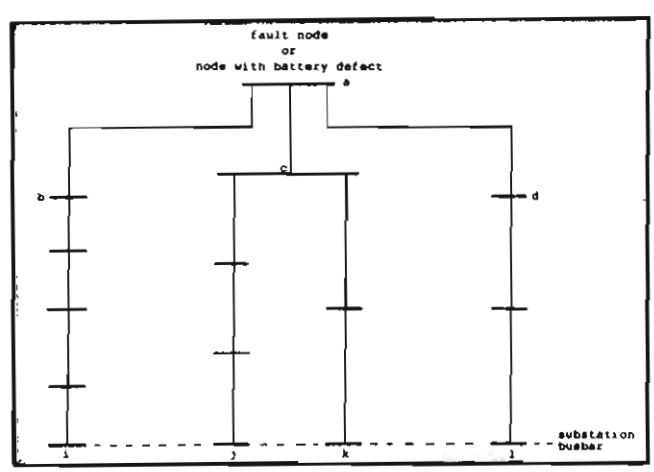
عبارات در دستور (۱۲) این واقعیت را نشان میدهد که خرابی باتری ممکن است در گره X یا Y در فیدر X + Y وجود داشته باشد . گره‌هایی که باتری معیوب دارند با حرف X_1 یا Y_1 و کلیدی که باز شده با عبارت $X_1 - Y_1$ مشخص میشود، همان طور که در جمله MEMBER مشاهده میشود. به خاطر خرابی باتری در گره Y_1 که با عبارت $B = Y_1$ نمایش داده میشود. کلید $Y_1 - X_1$ بسته باقی می‌ماند . این منجر به بروز علایمی میشود که همان علایم در زمان اتفاق خطا در گره Y_1 که در مدار بین گره X_1 وباس بار پست است ظاهر میشود. برای یافتن علایم، در روش تحقیق گرهی در دستور (۱۲) گره Y_1 به عنوان گره مبنا در نظر گرفته میشود و تحقیق از آن شروع میشود. علایم عیب نامرئی سیم حامل اطلاعات در مدار نیز با روش تحقیق گرهی میتواند پیدا شود. دستورات (۱۱) و(۱۲) یک شکل کاملی از قوانین عیب یابی یک فیدر میباشد.

۲-۳- تحقیق گره‌های شبکه برای یافتن علایم :

شکل (۴) یک گراف درختی از مدارهای فیدر یک شبکه حلقوی را نشان میدهد که از یک گره که خرابی باتری دارد یا خطایی در آن اتفاق افتاده به طرف باس بارهای پست که در آنجا رله‌های جریان زیاد i,j,k,l قرارداده شده‌اند ، منشعب میشوند . برای تعیین دستگاههای عملیاتی رله‌های جریان زیاد، گره‌های درخت آن قدر مورد بررسی قرارمیگیرد تا تمام رله‌های جریان زیاد تعیین شوند. دستورپرولاک زیر گره Z را به عنوان یک گره همسایه گره X جستجو میکند:

nextneighbour(X,L,Z) : - innet(X + Z), not member(Z,L). (۱۳)

L لیستی از گره‌هاست که قبلا" جستجو شده‌اند . عبارت INNET به طور یکسان در زبان پرولاک برای معرفی Z از روی واقعیات شبکه‌ها مورد استفاده قرار میگیرد. برای درخت (شکل۴) وقتی گره A را به عنوان گره X انتخاب میکنیم ، گره Z میتواند گره b گره c یا گره d باشد.



شکل ۴- گراف درخت یک شبکه حلقوی (k,j, i و l رله‌های جریان زیاد هستند)

با شروع در گره X ، علایم در امتداد یک خط از درخت ، بین X و باس بار پست با بکار بردن روش پیوسته وبسته زیر تعیین میشود:

nodescarch(X,-,S,-,-) : - overcurrentrelay(X),!,member(X,S). (۱۴)

```
nodesearch(X,L,S,B,P) : nextneighbour(X,L,Z),!,nodesearch(Z,[X|L],S,B,P)
```
(۱۵)

تحقیق دستور (۱۵) به وسیله دستور (۱۴) تعیین میشود که یک رله جریان زیاد پیدا میشود که علامت مربوط به آن در لیست S وجود دارد. سمبل ! در دو دستور قبلی یک مکانیسم کنترل است که CUT نام دارد که از تغییر مکان محل حوادث جلوگیری میکند . سمبل - در بحث دستور (۱۴) نشان میدهد که این دستور مستقل از مقادیر متغیرهای B,L و P است . برای جستجوی تمام شاخه‌های درخت ، یک سطح بسیار بزرگی از پیوستگی بسته در دستور (۱۵) به شرح زیر ایجاد شده است :

```
nodesearch(X,L,S,B,P) : - nextneighbour(X,L,Z),!,
nodesearch(Z,[X|L],S,B,P) , nodesearch(X,[Z|L],S,B,P).
```
(۱۶)

دستور بالا هم میتواند با دستور (۱۴) ، و هم با دستور زیر خاتمه داده شود وقتی که تمام گره‌های درخت جستجو شده باشند :

```
nodesearch(-,-,-,).
```
(۱۷)

۱-۲-۳- دخالت خرابی سیم حامل اطلاعات :

وقتی خطای فیدر ناشی از سیم حامل اطلاعات در سیستم حفاظتی فیدر پیش می‌آید ، علامتهای فوق العاده خطا بسته به حالتهای زیر ظاهر خواهد شد :

الف - کلیدهای Z - X و X - Z در صورت عدم وجود خرابی باتری در X و Z باز میکنند.

ب - کلید X - Z باز میکند وقتی که باتری در X معیوب باشد.

ج - کلید Z - X باز میکند وقتی که باتری در Z معیوب باشد.

علایم ناشی از عیب سیم حامل اطلاعات در حالات الف ، ب و ج در بالا میتواند در دستور (۱۶) یکسان تلقی شوند. مطابق دستور ذیل :

```
nodesearch(X,L,S,B,P) : - nextneighbour(X,L,Z,),!,
((P= X + Z),
((nonvar(B), B\ = = X, member(X - Z,S),
B\ = = Z , member(Z-X, S)),
(B = X,member(Z - X,S)),
(B = Z,member(X - Z,S)));
nodesearch(Z[X|L], S,B,P,)),
nodesearch(X,[Z|L],S,B,P).
```
(۱۸)

ساختارعبارت nonvar آزمایش میکند که آیا B معرفی شده است یا نه. پس دستورهای (۱۴) و (۱۸) و (۱۷) رویه تحقیق کامل و پیوسته مرتبط را ارائه میکند که شامل آثار معایب باز شدن سیم حامل اطلاعات و خرابی باتریها است .

۳-۳- اشکال درترانسفورماتور و باس‌بار:

وقتی یک خطا در برخی از باس‌بارهای شبکه حلقوی اتفاق می‌افتد، جریان‌های خطا در تمام مدارات فیدر که به باس‌بار معیوب وصل هستند جاری میشود. فیدرها بارشان زیاد میشود و رله‌های جریان زیاد پست عمل میکنند . این علایم میتوانند با دستورات روش تحقیق کره تعیین شوند. حق تقدم استفاده از این دستورات برای شینه‌هایی (باس‌بارها) میباشد که یک کره از یک شبکه باشد و این مطلب به وسیله دستور زیر بررسی میشود:

```
node(X) : - setof(A,Y^ innet(A+Y),N), member(X,Y).        (۱۹)
```

عبارت setof در دستور بالا یک جمله ساختاری در تفسیر کننده زبان پرولاک است وآن تمام کرههای شبکه موجود در لیست N را فراهم می‌آورد . عبارت member برای بررسی اینکه کره A در شبکه داده شده هست یا نه، مورد استفاده قرار میگیرد. با استفاده از دستورات (۱۴) و(۱۷) تا(۱۹) یک خطای باس‌بار به شرح زیر پیدا میشود:

```
busfault(X,S,B,P) : - node(X),nodesearch(X,[],S,B,P),
not overcurrentrelay(X).                                  (۲۰)
```

درعبارت nodesearch در (۲۰) ، کره X کره شروع است که تحقیق از آن شروع میشود. لیست L نخست به صفر تنظیم میشود. X به عنوان یک باس‌بار پست که بودن آن در یک شبکه ملاحظه نمیشود و آن توسط جمله not overcurrentrelay در دستور(۲۰) کنار گذاشته میشود. برای یک عیب روی یک ترانسفورماتور کلید ترانسفورماتور باز میکند. وقتی یک خرابی باتری روی باس‌باری که ترانسفورماتور به آن متصل است به وجود می‌آید، کلید بسته باقی می‌ماند ، به همان ترتیب در مقابل عیب باس‌بار نیز کلید باز نمیشود. یک روش عیب‌یابی برای یک ترانسفورماتور معیوب همچنین میتواند به شرح زیر بیان شود:

```
trfault(X,S,B,P) : - node(X)
((S = [X], B = nil);
(B = X, nodesearch(X,[ ],S,B,P))),
not overcurrentrelay(X).                                  (۲۱)
```

۴ - نتایج واجرای برنامه منطقی :

قوانین عیب‌یابی و نمایش توسعه یافته شبکه که در بخشهای (۳) و (۴) توضیح داده شد ، به طور کاملا" عمومی بوده و میتواند به شبکه‌های حلقوی با توپولوژیهای مختلف اعمال شود . برای ترکیبهای مختلف ، فقط حقایق شبکه (۲) و (۳) در بخش (۲-۳) نیاز به تعریف دارد. در این بخش ، برنامه منطقی توسعه یافته به یک شبکه ۲۰ کیلوولتی توزیع در شکل (۲) اعمال میشود.

۴-۱- یافتن عیب از روی علائم :

یکی از کاربردهای اصلی برنامه منطقی توسعه یافته ، تشخیص عیب‌ها در شبکه‌های حلقوی از روی لیست داده شده علایم میباشد. برای شبکه شکل (۲)، اگر یک لیست علامت [c-b,j , m] داده شود، پس محل عیب در فیدر باسئوال کردن از سیستم توسط هدف (GOAL) زیر میتواند تعیین شود:

```
? - feederfault(X,[c - b,j,m],B,P).
```

پس از اجرا، سیستم به یک عده از متغیرهایی به شرح زیر مبدل میشود:

X = b + c , B = b, and p = -

این یکی از حل‌هااست . مقدارX یک خطا را در فیدر b + c نشان میدهد و مقدار یک خرابی باتری را در گره b معرفی میکند. اما آنجا خرابی سیم حامل اطلاعات وجود ندارد ، چون p معرفی نشده است . حل درست است زیرا خرابی باتری در باس‌بار b مانع از باز شدن کلید c - b شده است و در نتیجه ، کلیدهای قدرت ، در j و m با عملکرد رله‌های جریان زیاد در آن محل‌ها باز کرده‌اند. مدارهای فیدر در بین c و f هیچ جریان خطایی را حمل نمکنند، چون کلید c - b باز شده است .

به همان ترتیب ، سئوالات میتواند با بکارگیری اهداف (goals) در مورد خطای شینه (busfault) و خطای ترانسفورماتور(trfault)برای تشخیص معایب باس‌بار و ترانسفورماتور درست بشود. برای یافتن تمام خطاهای ممکن که ممکن است در شبکه اتفاق افتد، قوانین عیب‌یابی میتواند برای سه نوع عیب ترکیب شده و تشکیل یک نوع دستور جدید findfault به شکل زیر بدهد:

```
findfault(S,LF,BF,TF,B,P) : - ( feederfault(LF,S,B,P);
busfault(BF,S,B,P);
trfault(TF,S,B,P).                                    (۲۲)
```

مقادیر متغیرهای LF , BF و TF محلهای عیب فیدر، عیب باس بار وعیب ترانسفورماتور را به ترتیب نشان میدهند . برای آخرین گروه از علایم ، تمام عیوب ممکن و نواقص ناشی از آنها میتوانند با اجرای هدف (GOAL) زیر پیدا شوند :

```
? - findfault ([c - b,j,m],LF,BF,TF,B,P).
```

جدول (۱) خلاصه‌ای از حل ۹ عیب را که در سیستم ایجاد شده بود نشان میدهد. اولین حل مربوط به عیب یک فیدر است و عیناً" همانند آنچه قبلاً" یافته شده ، میباشد. حل تمام معایب ممکن باس‌بار از سطر دوم تا هشتم جدول داده شده‌است. این عیب‌ها یک عیب مشترک باتری در باس‌بار b و یک عیب مشترک سیم حامل اطلاعات در امتداد فیدر c + b دارند . با همان اشکالات ، آخرین حل مربوط به یک خطای ترانسفورماتور در باس‌بار b میباشد . از این نتایج میتوان استنباط نمود که میبایست در گره b یک عیب باتری وجود داشته و نیز یک اشکال سیم حامل اطلاعات به احتمال قوی در سیستم حفاظت فیدر c + b وجود دارد.

اشکال سیم حامل اطلاعات	اشکال باتری	ترانسفورماتور	شینه (باس‌بار)	فیدر	نوع عیب
-	b	-	-	b+c	فیدر
b+c	b	-	l	-	باس‌بار
b+c	b	-	k	-	
b+c	b	-	a	-	
b+c	b	-	i	-	
b+c	b	-	h	-	
b+c	b	-	g	-	
b+c	b	-	b	-	
b+c	b	b	-	-	ترانسفورماتور

جدول ۱-نمونه‌ای از حلهای هدف findfault

۲-۴- یافتن علایم از روی عیب‌ها :

کاربرد دیگر برنامه لاجیک توسعه یافته ، یافتن تمام علایم و خطاهای مربوطه از روی یک خطای معلوم با اجرای آن به روش معکوس میباشد . برای یک خطای فیدر c + b در شبکه شکل (۲)، علایم و خطاها میتوانند با پرسش زیر پیدا شوند :

```
? - findfault(S,c + b,nil,nil,B,P).
```

دراین پرسش ، به‌جای LF, c + b قرار داده میشود. خطاهای باس‌بار و ترانسفورماتور باتبدیل BF و TF به nil از پرسش کنار گذاشته میشود. چهل وپنج حل که به وسیله سیستم پیدا شده ، وجود دارد و در جدول (۲) خلاصه شده‌است . اولین حل متناظر با حالت نرمالی است که تریپ کلیدهای b - c و c - b از خرابی روی فیدر c + b تفکیک میشود. وقتی که در b یک عیب باتری وجود دارد، کلید c - b باز میکند ولی کلید b - c بسته می‌ماند. جریانهای خطا در مدارهای ja، ab و ma جاری میشود. رله‌های جریان زیاد در j و m عمل میکنند. این در حل دوم ، در جدول (۲) نشان داده شده است . علایم در حل سوم به خاطر وضعیت عیب داده شده و خرابی باتری در گره c میباشد.

بقیه حلها میتواند در دو گروه طبقه‌بندی شوند. گروه اول ۳۱ حل است که به خاطر ترکیب آثار روی فیدر خطای c + b و عیوب ممکن که در فیدر خطا نیستند ، گره آنها می‌باشد. گروه دوم شامل یازده (۱۱) حل است که در (جدول ۲) مشاهده میشود، علایم عیب را با یک عیب باتری در انتهای فیدر معیوب و یک عیب سیم حامل اطلاعات مقایسه میکند.

۲-۴- کاربردهای دیگر :

موقعی که دستور عیب‌یابی (FINDFAULT) در (۲۲) میتواند به عنوان یک تعریفی از روابط با تمام بحثهایش منظور شود، میتواند با برخی از بحثهایش نیز ایجاد سئوال نماید که به صورت دو مورد کاربرد تعریف شده در بخشهای ۵-۱ و۵-۲ معرفی میشود . همه متغیرها در بحثهای دستور عیب‌یابی کماکان میتوانند یکجا معرفی شوند قبل از آنکه خواسته شوند. دستورات زبان پرولاگ از (۴) تا (۲۱) در بخش ۴ به این خاطر توسعه داده شده‌اند که این نیاز مخصوص (یافتن عیب) با کنترل دقیق و راحت و نفی کردن توسط پرولاگ برآورده میشود. به خاطر عمومیت سیستم توسعه یافته ، از آن میتوان برای تهیه جداولی استفاده کرد که

علایم	نقص باتری	نقص سیم حامل اطلاعات
[c-b,b-c]	-	-
[c-b,j,m]	b	-
[b-c,f]	c	-
[c-b,b-c,e-f,f-e]	_	e+f
[c-b,b-c,f-e]	e	e+f
[c-b,b-c,e-f]	f	e+f
[c-b,b-c,d-e,e-d]	_	d+e
[c-b,b-c,e-d]	d	d+e
[c-b,b-c,d-e]	e	d+e
[c-b,b-c,c-d,d-c]	-	c+d
[c-b,b-c,c-d]	d	c+d
[c-b,b-c,a-b,b-a]	-	a+b
[c-b,b-c,b-a]	a	a+b
[c-b,b-c,i-j,j-i]	-	i+j
[c-b,b-c,j-i]	i	i+j
[c-b,b-c,i-j]	j	i+j
[c-b,b-c,h-i,i-h]	-	h+i
[c-b,b-c,i-h]	h	h+i
[c-b,b-c,h-i]	i	h+i
[c-b,b-c,g-h,h-g]	-	g+h
[c-b,b-c,h-g]	g	g+h
[c-b,b-c,g-h]	h	g+h
[c-b,b-c,a-g,g-a]	-	a+g
[c-b,b-c,g-a]	a	a+g
[c-b,b-c,a-g]	g	a+g
[c-b,b-c,l-m,m-l]	-	l+m
[c-b,b-c,m-l]	l	l+m
[c-b,b-c,l-m]	m	l+m
[c-b,b-c,k-l,l-k]	-	k+l
[c-b,b-c,l-k]	k	k+l
[c-b,b-c,k-l]	l	k+l
[c-b,b-c,a-k,k-a]	-	a+k
[c-b,b-c,k-a]	a	a+k
[c-b,b-c,a-k]	k	a+k
[c-b,a-b]	b	a+b
[c-b,a-g,g-a,m]	b	a+g
[c-b,g-h,h-g,m]	b	g+h
[c-b,h-i,i-h,m]	b	h+i
[c-b,i-j,j-i,m]	b	i+j
[c-b,j,a-k,k-a]	b	a+k
[c-b,j,k-l,l-k]	b	k+l
[c-b,j,l-m,m-l]	b	l+m
[b-c,d-c]	c	d+c
[b-c,d-e,e-d]	c	d+e
[b-c,e-f,f-e]	c	e+f

جدول ۲- علایم و نارسایی‌های مربوط به یک عیب روی فیدر bc در (شکل۲)

میتوانند علایم و عیوب مختلف را به هم ارتباط دهند و برای بحث عیوب در مورت نیاز از جدول یادداشت مطابق توضیح (بخش ۱) استفاده میشود. کاربرد دیگر سیستم توسعه یافته تشخیص عیبها از روی علایم ویژه ناقص است ، که این با تشکیل یک هدف (GOAL) جدید شامل عیب‌یابی بدون مرحله و یک دستور SUBSETTEST که برای صاف کردن این حلها بکار رفته و به علایم خاصی مربوط نیستند، انجام می‌یابد. اگر برای مثال قطع کلید a - b فقط تنها علامت معلوم باشد، تمام خطاهای ممکن میتوانند از دستور زیر پیدا شوند:

```
? - findfault(S,LF,BF,TF,B,P),
subsettest([b - a],S).
```

سپس سیستم برمی‌گردد به حل خطاهایی که در آنها قطع کلید a - b یکی از عناصر لیست علایم S بود. میتوان از روی حلها، احتمال یک خطا یا نقص معین را با شمردن تعداد اتفاق افتادن‌هایش تخمین زد. این مخصوصا" در عده‌ای از موقعیتهای عملی مفید است وقتی که یک لیست کامل علایم موجود نباشد.

نتیجه :

یک دستور لاجیک عمومی برای تشخیص عیب در شبکه‌های حلقوی توزیع توسعه داده شده است و با استفاده از برنامه‌ریزی منطقی زبان پرولاک خاتمه یافته‌است. روش نمایش شبکه مطرح در این مقاله کاملا" ساده وانعطاف‌پذیراست. برخی تغییرات در توپولوژی شبکه را میتوان با تغییر دستورات منطق شبکه انجام داد. روش تعریف ساختمان شبکه‌ها از خواص مجتمع و غیر حساس عبارتهای زبان پرولاک استخراج میشود، که هر کدام بطور صریح و کامل رویه‌های تحقیق باس‌بارهای شبکه و فیدرهای شبکه را توصیه میکنند. رفتار شبکه‌های حلقوی توزیع که در آنها عیب‌های فیدر، باس‌بار یا ترانسفورماتور هست با استفاده از روابط بین خطا و علامتهایشان به صورت فرمول در می‌آید. این روابط به شکل قوانین عیب‌یابی پرولاک که در (بخش ۴) ارائه شدند بیان میشوند. آثار نامرئی خرابی باتری و باز شدن سیم حامل اطلاعات در سیستم شبکه نیز مشمول فرمول‌بندی میباشد. وقتی که یک لیست علایم معین میشود، برنامه منطقی سیستم ارائه شده در این مقاله نه تنها عیب را مشخص میکند بلکه تمام معایب و نارساییهایی احتمالی را که ممکن است در شبکه اتفاق افتد، معین میکند . در این روش احتمال عدم موفقیت در تشخیص عیب بسیار کم است . زمان لازم اجرا برای یافتن جوابهای آزمایش نمونه ارائه

شده در (بخش ۵-۱) توسط کامپیوتر (VAX - 750) ۱۵ ثانیه میباشد. سیستم حاضر همچنین میتواند برای ایجاد کلیه علامتهای ممکن و نواقص جمعی ، مطابق وضعیت عیب داده شده مورد استفاده قرار گیرد .

ارزش این سیستم تشخیص عیب دارای دو قسمت است . اولین کاربرد آن سرعت بخشیدن برای تعیین عیب شبکه توسط مهندسین عملیاتی است . دومین کاربرد آن یک هدف آموزشی است که توسط مهندسین آموزشی و علمی مورد استفاده قرار میگیرد. با وجود اینکه برنامه منطقی در این مقاله با زبان برنامه سی - پرولاک (C -PROLOG) نوشته شده است و در کامپیوتر (VAX - 750) قابل پردازش است ، همچنین میتواند به سادگی به وسیله تفسیرکنندگان پرولاک تجارتی توسط سیستم‌های کامپیوترهای شخصی (PC) انجام شود.

منابع :

۱- کتاب مرجع C -PROLOG و VERSION1.5 دانشگاه ادینبرگ (۱۹۸۶)

۲- تعداد کثیری از مجلات سیستمهای قدرت و دیسکتهای مربوطه در زمینه زبان PROLOG

۳- تعدادی از مجلات IEEE