

پیاده سازی کنترل نظارتی سیستم‌های گسسته پیشامد با استفاده از PLC

علی اکبر افزالیان

سید محمد نوربخش رضایی

گروه مهندسی کنترل، دانشگاه صنعت آب و برق (شهید عباسپور)، تهران، ایران

Afzalian[AT]pwut.ac.ir

S.M_Noorbakhsh[AT]yahoo.ca

چکیده - در این مقاله روش‌های پیاده‌سازی یک کنترل ناظر برای سیستم‌های گسسته پیشامد روی PLC بررسی شده است. در مسیر حرکت از یک مدل گسسته پیشامد که ذاتاً بصورت آسنکرون رفتار نموده و در پی وقوع سلسله پیشامدها هدایت می‌شود، به سمت PLC که دارای طبیعتی سنکرون بوده و بر اساس سیگنال‌های زمانی (نه پیشامدها) عمل می‌نماید، چندین مشکل اساسی وجود دارد که برخی از این مشکلات عبارتند از: غیر علی بودن سیستم حلقه بسته، امکان وقوع همزمان چند پیشامد در PLC و عدم تطابق آن با کنترل ناظر، چگونگی انتخاب بین مسیرهای مختلف، و غیر واقعی بودن همزمانی فرایند و PLC. در این مقاله این مشکلات بررسی شده و راه حل‌های ممکن ارائه می‌گردد. پس از آن در قالب مثالی از یک فرآیند صنعتی واقعی و با استفاده از الگوریتمی که ارائه خواهد شد، نحوه پیاده‌سازی مرحله به مرحله کنترل ناظر بر روی PLC نشان داده خواهد شد. نتایج بدست آمده از پیاده‌سازی کنترل ناظر بر روی PLC نشان می‌دهند که با استفاده از روش‌های مطرح شده می‌توان مشکلات پیاده‌سازی کنترل ناظر را کاهش داده و زمینه را برای بکارگیری بلادرنگ سیستم‌های کنترل ناظر فراهم نمود.

کلید واژه - پیاده‌سازی کنترل نظارتی، سیستم گسسته پیشامد، PLC.

۱- معرفی

سیستم‌های گسسته پیشامد^۱ دربردارنده گروه وسیعی از سیستم‌های صنعتی نظیر سیستم‌های کنترل ترافیک، سیستم‌های کارخانجات صنعتی، سیستم‌های مخابراتی، سیستم‌های قدرت، سیستم‌های پایگاه داده و... می‌باشند. برای کنترل این دسته از سیستم‌ها از تئوری کنترل نظارتی^۲ استفاده می‌شود [۱]. در دهه‌ی اخیر تحقیقات زیادی در رابطه با بسط و توسعه این تئوری صورت پذیرفته است ولی به دلیل وجود مشکلات فراوان بر سر راه پیاده‌سازی عملی این کنترلرها، هنوز تعداد اندکی از حوزه‌های کاربردی صنعتی را می‌توان مثال زد که این تئوری را مورد استفاده قرار داده باشند [۲-۴]. در این مقاله مشکلات اساسی پیاده‌سازی کنترل ناظر معرفی گردیده و مورد تجزیه و تحلیل قرار خواهند گرفت و در هر مورد نیز انواع راه حل‌های ممکن برای اجتناب از این مشکلات ارائه می‌گردد.

به دلیل گستردگی کاربرد PLC^۳ در صنعت، یکی از بهترین گزینه‌ها برای پیاده‌سازی عملی یک کنترل ناظر استفاده از PLC می‌باشد. پیاده‌سازی کنترل ناظر بر روی PLC در چندین مقاله از جمله [۳-۱۱] و [۱۲] مطرح شده است. در این مقاله با استفاده از الگوریتمی که ارائه می‌شود، نحوه پیاده‌سازی مرحله به مرحله کنترل ناظر طراحی شده برای یک فرآیند صنعتی بر روی PLC تشریح خواهد شد. همانند [۵] مزیت الگوریتم ارائه شده در این مقاله نسبت به روش‌های دیگر در این است که کد LD بدست آمده در این روش را می‌توان مستقیماً به PLC ارسال نموده و اجرا کرد.

در بخش ۲ تئوری کنترل نظارتی به صورت مختصر معرفی می‌شود. در بخش ۳ مشکلات موجود برای پیاده‌سازی عملی یک کنترل ناظر بر روی PLC دسته‌بندی گردیده و علاوه بر آن راه‌حل‌های ممکن برای فائق آمدن بر هر کدام از این مشکلات نیز ارائه می‌شود. در بخش ۴ نیز در قالب

^۳ Programmable Logic Controller

^۱ Discrete Event Systems

^۲ Supervisory Control Theory

۳- مشکلات پیاده‌سازی کنترل ناظر بر روی PLC از اوایل سال ۱۹۷۰ میلادی در کاربردهای صنعتی به عنوان جانشینی برای مدارات منطقی- رله‌ای مورد استفاده قرار گرفت. یک سیکل اسکن PLC شامل سه مرحله بصورت خواندن ورودی‌ها- اجرای برنامه - اعمال خروجی می‌باشد. برنامه‌ی PLC عبارت است از یکسری عبارات منطقی که در هر سیکل اسکن یکی پس از دیگری اجرا می‌شوند. برای نوشتن این برنامه می‌توان از انواع زبان‌های برنامه‌نویسی مخصوص این کار نظیر LD ، IL ، SFC ، FBD و ST استفاده نمود [۱۴-۱۶]. معمول‌ترین و شاید ساده‌ترین این زبان‌ها، LD می‌باشد که یک زبان برنامه‌نویسی گرافیکی سمبولیک است.

برای پیاده‌سازی کنترل ناظر بر روی PLC و یا به عبارتی تبدیل اتوماتن کنترل ناظر به یک کد LD می‌توان متناظر با هر کدام از حالت‌ها و پیشامدهای اتوماتن یک متغیر نظر گرفت. در این صورت یک مسیر یا گذر خاص را می‌توان متناظر با AND متغیر حالت مبدأ و متغیر مربوط به پیشامد آن مسیر در نظر گرفت. هنگامی که یک گذر اتفاق می‌افتد متغیر متناظر با حالت مقصد برابر یک منطقی و متغیر مربوط به حالت قبل برابر صفر منطقی می‌گردد [۱۲].



شکل ۲- نحوه‌ی پیاده‌سازی یک اتوماتن

برای ایجاد ارتباط میان وقوع پیشامد در یک اتوماتن و تغییر سطح یک سیگنال زمانی باینری در PLC می‌توان لایه‌ی بالا رونده‌ی سیگنال PLC را متناظر با وقوع پیشامد در اتوماتن در نظر گرفت. بنابراین برای تشخیص وقوع یک پیشامد کافیهست مقدار متغیر در سیکل اسکن فعلی با مقدار آن در سیکل اسکن قبلی مقایسه شود.

در بخش‌های زیر مشکلات پیاده‌سازی کنترل ناظر بر روی PLC بصورت مفصل بررسی شده و راه حل‌های ممکن ارائه می‌گردد.

۳-۱- اثر Avalanche

چنانچه در طول یک سیکل اسکن در اثر وقوع یک پیشامد خاص، چندین تغییر حالت رخ دهد مشکلی موسوم به اثر

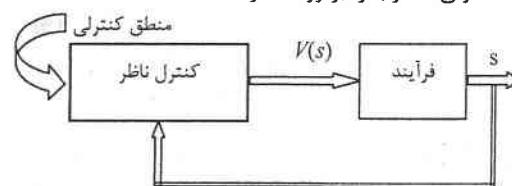
مثالی از یک فرایند واقعی نحوه‌ی پیاده‌سازی مرحله به مرحله‌ی یک کنترل ناظر بر روی PLC با استفاده از الگوریتم بکار گرفته شده ارائه می‌گردد.

۲- طراحی کنترل ناظر

یک سیستم گسسته پیشامد (DES) سیستمی است که دارای رفتار دینامیکی آنسکرون بوده و در اثر وقوع آنی سلسله پیشامدها در نمونه‌های زمانی گسسته، از یک حالت به حالت دیگر تغییر وضعیت می‌دهد [۱۳]. تکنیک کنترل نظارتی ابزار مناسبی برای کنترل این دسته از سیستم‌ها می‌باشد. برای استفاده از این تکنیک در قدم اول باید فرایند گسسته پیشامد را بصورت یک اتوماتن^۱ مدل‌سازی نمود. اتوماتن یک ۵- تایی است که بصورت $G = (Q, \Sigma, \eta, q_0, Q_m)$ تعریف می‌شود. Q مجموعه‌ی متناهی از حالت‌های سیستم گسسته پیشامد، $q_0 \in Q$ حالت اولیه و $Q_m \subseteq Q$ مجموعه‌ای شامل تمامی حالت‌های مورد نظر این سیستم می‌باشد. Σ نیز مجموعه‌ای متناهی از تمامی پیشامدها می‌باشد. حالت بعدی تحت وقوع پیشامد σ توسط تابع گذر از حالت $\eta(q, \sigma)$ مشخص می‌شود که بصورت $\eta: Q \times \Sigma \rightarrow Q$ تعریف می‌گردد. پس از مدل‌سازی فرایند، منطق کنترلی مطلوب را نیز باید بصورت یک اتوماتن بیان نمود. با داشتن اتوماتن فرایند (G) و اتوماتن منطق کنترلی (EDES)، می‌توان کنترل ناظر Supervisor را با استفاده از نرم افزار TCT [۱۳] بصورت زیر طراحی نمود:

$$\text{Supervisor} = \text{SUPCON}(G, \text{EDES})$$

همانطوری که در شکل ۱ نیز نشان داده شده، کنترل ناظر در واقع سلسله پیشامد s را که توسط فرایند تولید می‌شود دریافت نموده و با غیر فعال نمودن یکسری از پیشامدهای کنترل پذیر، باعث می‌شود که سیستم حلقه بسته منطق کنترلی مطلوب را برآورده سازد.



شکل ۱- بلوک دیاگرام کنترل ناظر

^۱ Automaton

مختلف مدل DES هستند، بصورت همزمان برابر ۱ منطقی شوند. همانند قسمت قبل با استفاده از یک متغیر میانی می توان از عدم تطابق مدل DES کنترل ناظر با نمونه ی پیاده سازی شده ی آن بر روی PLC جلوگیری نمود.

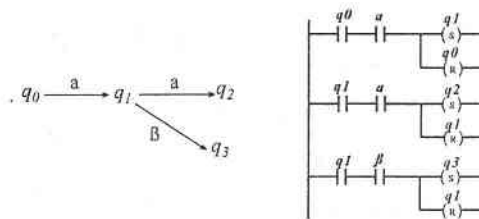
۳-۳- غیر علی بودن سیستم حلقه بسته

در تئوری کنترل نظارتی [۱] فرض می شود که همه ی پیشامدها توسط فرایند تولید می شود و کنترل ناظر با در اختیار داشتن کل پیشامدها، فقط برخی از پیشامدهای کنترل پذیر را غیر فعال می نماید. در صورتیکه در دنیای واقعی، فرایند در پاسخ به پیشامدهایی که کنترلر (ناظر) تولید نموده رفتار می نماید. بنابراین چنانچه تئوری کنترل نظارتی در عمل مورد استفاده قرار بگیرد خاصیت علی بودن سیستم حلقه بسته مورد شک قرار می گیرد. در [۴] با دید دیگری به مسئله نگریسته شده است و این طور فرض شده که پیشامدهای کنترل پذیر توسط کنترل ناظر و پیشامدهای کنترل ناپذیر توسط فرایند تولید می شوند و بدین طریق ابهام در مورد علیت سیستم حلقه بسته بخوبی برطرف می گردد.

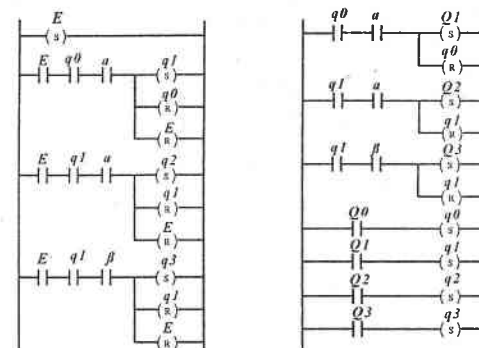
۳-۴- چگونگی انتخاب بین مسیرهای مختلف

در اتوماتان کنترل ناظر ممکن است یک حالت دارای چندین مسیر خروجی باشد. چنانچه این مسیرهای خروجی تحت پیشامدهای کنترل ناپذیر باشند، وقوع یکی از این پیشامدها در فرایند خود بخود مسیر خروجی را مشخص می نماید. مشکل وقتی ظاهر می شود که برای یک حالت، چندین مسیر خروجی تحت پیشامدهای کنترل پذیر وجود داشته باشد که باید توسط PLC (کنترل ناظر) یکی از این پیشامدها فعال شود. به عنوان نمونه شکل ۵ که قسمتی از کنترل ناظر ارائه شده در شکل ۶ مقاله ی [۱۷] است را در نظر بگیرید. در حالت q_x باید یکی از دو پیشامد کنترل پذیر w_1 و w_2 توسط کنترل ناظر (PLC) فعال شود. اگر این ناظر بصورت LD داده شده در سمت چپ شکل ۵ پیاده سازی شود، در پله ی ۳ با فرض اینکه حالت q_a فعال بوده و لبه ی بالا رونده ی f_1 تشخیص داده شود، حالت q_x فعال می شود. پس از آن در پله ی ۵ و ۶ هر دو متغیر کنترل پذیر w_1 و w_2 توسط PLC ست می شوند که مطلوب نمی باشد (معادل اینکه در اتوماتان فرایند دو مسیر خروجی یک حالت همزمان

Avalanche بوجود می آید. که با وقوع آن، PLC (کنترل ناظر) با فرایند سنکرون نخواهد بود. به عنوان مثال در ارائه شده در شکل ۳ چنانچه در ابتدای یک سیکل اسکن لبه ی بالا رونده ی α تشخیص داده شود (یعنی این پیشامد رخ داده باشد)، ناظر در انتهای آن سیکل اسکن مستقیماً از حالت q_0 به حالت q_2 خواهد رفت. برای جلوگیری از وقوع چنین مشکلی می توان ترتیب پله ها را در LD معکوس نمود [۶]. در این صورت پس از رفتن به حالت q_1 در اثر وقوع α ، این اطمینان حاصل می شود که قبل از رفتن به q_2 ، حتماً یک لبه ی پایین رونده در α رخ داده باشد. راه حل دیگر به این صورت است مطابق سمت راست شکل ۴، متناظر با هر حالت مدل گسسته پیشامد، دو متغیر در نظر گرفته شود و در پایان سیکل اسکن این دو متغیر یکسان سازی شوند [۱۱]. در روشی دیگر برای اطمینان از این که در طول هر سیکل اسکن تنها یک تغییر حالت بوجود آید، می توان از یک متغیر واسط مانند متغیر E در سمت چپ شکل ۴ استفاده نمود.



شکل ۳ - اثر Avalanche

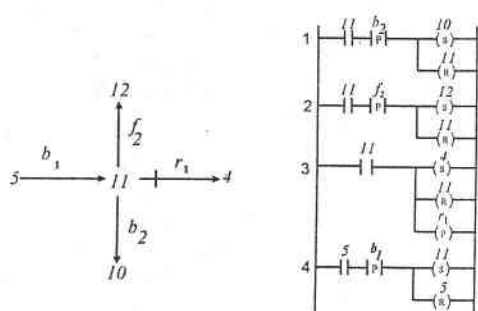


شکل ۴ - جلوگیری از اثر Avalanche

۳-۲- امکان وقوع همزمان پیشامدها در PLC

در یک مدل DES در هر لحظه از زمان فقط یک پیشامد رخ می دهد، در صورتیکه در PLC ممکن است در طول یک سیکل اسکن چندین متغیر که متناظر با پیشامدهای

فرایند فیزیکی رخ داده باشند، فرایند فیزیکی و کنترل ناظر (PLC) از حالت سنکرونی با یکدیگر خارج می‌شوند. چراکه بعد از وقوع f_2 یا b_2 ، فرایند فیزیکی نمی‌تواند در پایان سیکل اسکن دوم وقوع r_1 را ببیند. این مسئله را می‌توان بصورت یک تأخیر در انتقال دیتا در نظر گرفت. بنابراین چنانچه زبان کنترل ناظر بگونه‌ای باشد که نسبت به تأخیر غیرحساس باشد مشکل بر طرف خواهد شد. با در نظر گرفتن این مفهوم، ناظر شکل ۶ باید بتواند پیشامد r_1 را در حالت ۱۰ و b_2 را در حالت ۴ نیز قبول نماید.



شکل ۶- باید در یک پله همزمان هم پیشامد کنترل‌پذیر ست شود و هم حالت مقصد فعال شود.

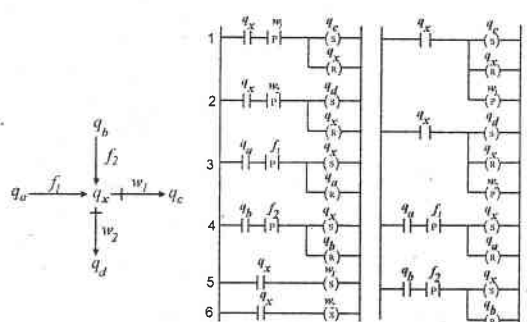
۴- پیاده‌سازی کنترل ناظر بر روی PLC

در این قسمت جهت پیاده‌سازی کنترل ناظر بر روی PLC الگوریتمی برای تبدیل یک اتوماتن به کد LD متناظر آن ارائه می‌شود. این الگوریتم با استفاده از یکسری اطلاعات مربوط به عملکرد کلی فرایند، کد LD متناظر با کنترل ناظر را بگونه‌ای بدست می‌آورد که مستقیماً قابل ارسال به PLC باشد.

در ابتدا چند مفهوم باید تعریف نمود: هر فرایندی با انجام یکسری فعالیت‌ها کار مشخصی را انجام می‌دهد و برای انجام هر کدام از این فعالیت‌ها دسته‌ای از منابع مورد استفاده قرار می‌گیرد (استاندارد IDEF3 [۱۸]). بنابراین فرایند را می‌توان بصورت مجموعه‌ای از فعالیت‌ها و منابع در نظر گرفت [۱۹]. به عنوان مثال پروسه‌ی دریل نمودن یک قطعه کار را می‌توان به صورت یک فعالیت در نظر گرفت که به منابعی نظیر ماشین ابزار و روبات نیازمند می‌باشد. بلافاصله پس از آنکه همه‌ی منابع مورد نیاز یک فعالیت اختصاص داده شد، پیشامد اولیه‌ی آن فعالیت رخ می‌دهد. بنابراین وقوع پیشامد اولیه‌ی یک فعالیت متناظر با

فعال شوند). در سیکل اسکن بعدی در پله‌ی ۱ حالت q_c فعال می‌شود و به دلیل غیر فعال شدن q_x عملاً ست شدن w_2 در سیکل اسکن قبلی در نظر گرفته نمی‌شود. مشکل هنگامی فاجعه‌آمیز خواهد بود که در عمل رفتار فرایند بگونه‌ای باشد که چنانچه w_1 و w_2 همزمان فعال شوند، w_1 نادیده گرفته شود، بنابراین فرایند فیزیکی و کنترل ناظر از حالت سنکرون بودن با یکدیگر خارج می‌شوند. برای رفع مشکل باید در یک پله همزمان هم پیشامد کنترل‌پذیر ست شود و هم حالت مقصد فعال شود. این استراتژی در LD سمت راست شکل ۵ نشان داده شده است. هر چند در این LD باز هم پله‌ی ۲ اضافی است.

بطور کلی نهایتاً نظر طراح کد LD الویت w_1 و w_2 را نسبت به یکدیگر تعیین می‌کند و به همین جهت باید پله‌هایی مربوط به پیشامدهای کنترل ناپذیر (پیشامدهای ورودی به PLC) قبل از پله‌های مربوط به پیشامدهای کنترل‌پذیر قرار داده شوند.



شکل ۵- چگونگی انتخاب بین مسیرهای مختلف

۳-۵ غیر واقعی بودن همزمانی فرایند و PLC

با آنکه می‌توان مدت زمان اجرای یک سیکل اسکن را در مقابل ثابت زمانی فرایند ناچیز تصور نمود، اما هم‌فاز بودن فرایند و کنترل ناظر (PLC) را نمی‌توان گارانتی کرد. چرا که در خلال اجرای برنامه در طول یک سیکل اسکن، PLC هیچ اطلاعی از دنیای خارج ندارد. شکل ۶ قسمتی از کنترل ناظر داده شده در شکل ۱۳ مقاله [۱۲] می‌باشد. چنانچه حالت ۵ فعال باشد و لبه‌ی بالارونده‌ی b_1 تشخیص داده شود، حالت ۱۱ فعال می‌شود. با فرض آنکه در شروع سیکل اسکن بعدی لبه‌ی بالارونده‌ی f_2 یا b_2 وجود نداشته باشد در پله‌ی ۳ حالت ۴ فعال شده و r_1 ست می‌شود. حال چنانچه در خلال اجرای سیکل اسکن دوم f_2 یا b_2 در

پیشامد داده شده در سمت چپ شکل ۸ را در نظر بگیرید. وظیفه‌ی این روبات دریل نمودن یک قطعه در خط تولید کارخانه می‌باشد. در حالت عادی چنانچه این روبات خراب شود، یک اپراتور آن را تعمیر می‌نماید. منطق کنترلی مورد نظر را به این صورت در نظر می‌گیریم که اگر روبات سه دفعه خراب شد، پس از آن فرایند دیگر راه‌اندازی نشود. با در نظر گرفتن این منطق کنترلی و با استفاده از تئوری کنترل نظارتی، کنترل‌ناظر در نرم‌افزار TCT طراحی می‌شود که در سمت راست شکل ۸ نشان داده شده است.

برای استفاده از الگوریتم ارائه شده در بخش ۴-۱، ابتدا فعالیت‌ها و منابع مشخص می‌گردند که نتیجه‌ی آن در جدول ۱ نشان داده شده است. پس از آن پیشامدهای فرایند دسته‌بندی می‌شوند که در جدول ۲ ارائه شده‌اند. در مرحله‌ی بعد در صورتیکه پیشامد اولیه‌ی یک فعالیت در اتوماتن کنترل ناظر نشان داده نشده باشد، با اضافه کردن یک حالت میانی، این پیشامد اولیه نیز در اتوماتن نشان داده خواهد شد. در اینجا پیشامد اولیه‌ی فعالیت تعمیر روبات در اتوماتن کنترل ناظر شکل ۸ نشان داده نشده است، بنابراین با اضافه نمودن حالت‌های میانی در محل‌هایی که پیشامد نهایی این فعالیت رخ داده، پیشامد اولیه را نیز در اتوماتن نشان می‌دهیم.

جدول ۱ - فعالیت‌ها و منابع بکار گرفته شده در فرایند

فعالیت‌ها	دریل نمودن قطعه کار تعمیر روبات
منابع	روبات اپراتور تعمیرکننده روبات

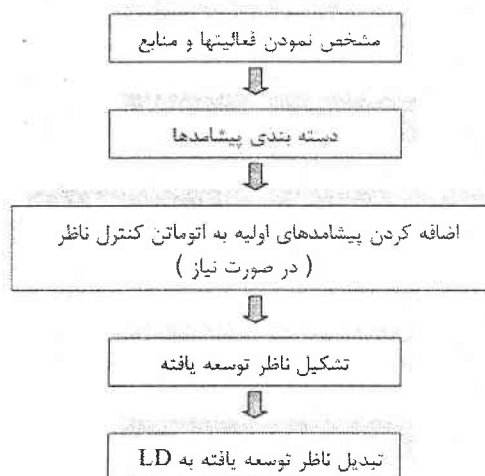
جدول ۲ - دسته بندی پیشامدهای فرایند

پیشامد	گروه پیشامد
اختصاص روبات	پیشامد اختصاص
اختصاص اپراتور تعمیرکار	منابع
آزاد شدن روبات	پیشامد عدم اختصاص منابع
آزاد شدن اپراتور تعمیرکار	پیشامد اولیه‌ی یک فعالیت ($\sigma \in \Sigma_s$)
شروع فعالیت دریل نمودن قطعه	پیشامد نهایی یک فعالیت ($\sigma \in \Sigma_f$)
شروع فعالیت تعمیر روبات	سایر پیشامدها
پایان یافتن فعالیت دریل قطعه	
پایان یافتن فعالیت تعمیر روبات	
خراب شدن روبات	

اختصاص آخرین منبع مورد نیاز برای اجرای آن فعالیت می‌باشد. تکمیل انجام یک فعالیت به منزله‌ی وقوع پیشامد پایانی آن فعالیت می‌باشد. به کار گرفتن هر کدام از منابع برای انجام یک فعالیت، معادل وقوع یک پیشامد/اختصاص منابع و آزاد نمودن آن از شرکت در اجرای فعالیت معادل وقوع یک پیشامد عدم اختصاص منابع می‌باشد. پیشامدهای اختصاص و عدم اختصاص یک منبع متناظر با خروجی‌های PLC و پیشامدهای اولیه و پیشامدهای پایانی یک فعالیت متناظر با ورودی‌های PLC می‌باشند.

۴-۱- الگوریتم تبدیل اتوماتن کنترل ناظر به LD

بلوک دیاگرام الگوریتم پیشنهاد شده در این مقاله در شکل ۷ نشان داده شده است که نمونه‌ی اصلاح شده‌ی الگوریتم ارائه شده در [۵] می‌باشد. با استفاده از این الگوریتم می‌توان کد LD بدست آمده را مستقیماً بر روی PLC ارسال نموده و آن را اجرا کرد. در قسمت ۴-۲ نحوه‌ی استفاده از این الگوریتم در قالب مثالی از یک فرایند واقعی کاملاً روشن خواهد شد.



شکل ۷ - بلوک دیاگرام الگوریتم پیاده سازی کنترل ناظر بر روی PLC

۴-۲- مثالی از پیاده‌سازی یک کنترل ناظر بر

روی PLC

برای نشان دادن چگونگی استفاده از الگوریتم فوق برای تبدیل کنترل ناظر به کد LD، مثال ساده‌ای از یک فرایند واقعی مد نظر قرار می‌گیرد. روبات R با مدل گسسته

$$Inputs_i = \begin{cases} \{x\} & \text{if } (\sigma_i \in \Sigma_s) \& (N=1) \\ \{x, \sigma_i\} & \text{o.w.} \end{cases}$$

$$O_{set_i} = \{ \{r_j \mid R_{y_i}(j)=1 \& R_x(j)=0\}, y_i \} \quad j=1,2,\dots,m$$

$$O_{reset_i} = \{ \{r_j \mid R_{y_i}(j)=0 \& R_x=1\}, x \}$$

* در رابطه‌ی بالا متغیر r_j متناظر با پیشامد اختصاص و یا عدم اختصاص زامین منبع در حالت x می‌باشد.

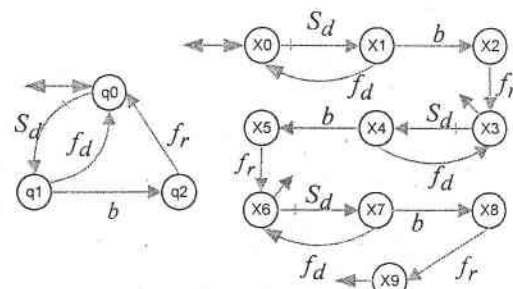
با بکارگیری این روابط برای هر کدام از حالت‌های مدل گسسته پیشامد شکل ۹، کد LD متناظر با آن بصورت شکل ۱۰ بدست خواهد آمد. به عنوان نمونه حالت $X1$ را در نظر بگیرید که از آن ۲ گذر تحت پیشامدهای f_d و b خارج می‌شوند. متناظر با این حالت، دو پله‌ی ۳ و ۴ در شکل ۱۰ به ترتیب برای گذر تحت پیشامدهای f_d و b در نظر گرفته می‌شوند. با توجه به اینکه از حالت $X1$ تحت پیشامد f_d به حالت $X0$ می‌رویم، در پله‌ی ۳ که متناظر با این پیشامد در نظر گرفته شده است، متغیرهای $X1$ و f_d را به عنوان ورودی‌ها، متغیر $X0$ را به عنوان خروجی ست شونده، و نهایتاً متغیرهای $X1$ و R_d را به عنوان خروجی‌های ریست شونده‌ی این پله در نظر می‌گیریم. یعنی:

$$Inputs_3 = \{X1, f_d\}, O_{set_3} = \{X0\}, O_{reset_3} = \{X1, R_d\}$$

با بکارگیری دستورالعمل فوق برای تمامی گذرهای مدل شکل ۹، نهایتاً کد LD بدست آمده بصورت شکل ۱۰ خواهد بود که با ارسال مستقیم آن به PLC می‌توان کنترل ناظر شکل ۸ را محقق نمود. به عنوان نمونه با اجرای پله‌ی ۵ این کد، چنانچه روبات خراب شده باشد (حالت x_2) با ست شدن O_r ، اپراتور تعمیرکار فعال می‌شود.

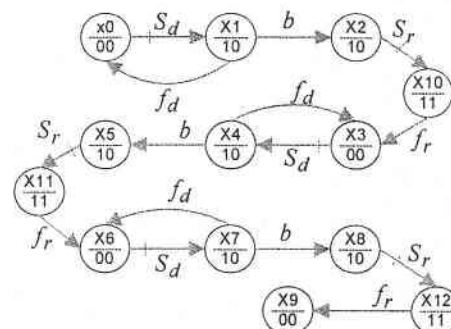
5- نتیجه گیری

در این مقاله ابتدا مشکلات اساسی فراروی پیاده‌سازی کنترل ناظر روی PLC به صورت کامل بررسی گردید. به دنبال آن در قالب مثالی از یک فرایند واقعی نشان داده شد که با داشتن یکسری اطلاعات قبلی از رفتار فرایند و استفاده از الگوریتم ارائه شده، به سادگی می‌توان مدل گسسته پیشامد کنترل ناظر را بر روی PLC پیاده‌سازی نمود. بررسی دیگرام نردبانی بدست آمده از الگوریتم مذکور، صحت پیاده‌سازی را تأیید می‌کند. امکان پیاده‌سازی کنترل ناظر روی PLC زمینه را برای بکارگیری تئوری کنترل نظارتی در کنترل بلادرنگ سیستم‌های گسسته پیشامد فراهم خواهد نمود.



شکل ۸- مدل گسسته پیشامد فرایند و کنترل ناظر آن

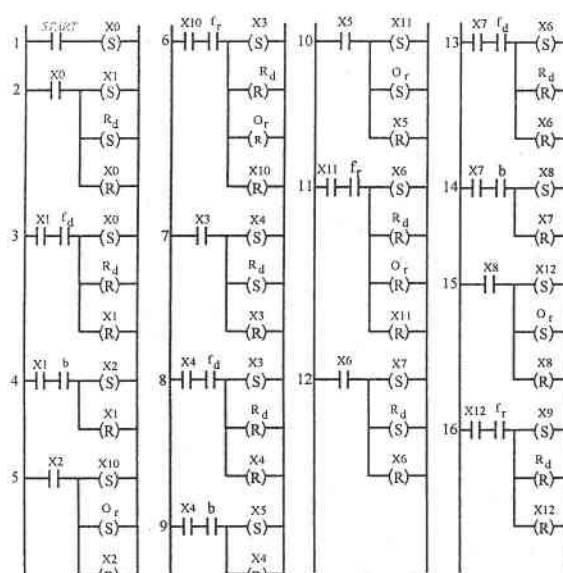
در قدم بعد بر اساس اطلاعات جداول ۱ و ۲ برای هر حالت x بردار باینری R_x با اندازه‌ی $m \times 1$ را تشکیل می‌دهیم که در آن m تعداد کل منابع موجود در فرایند می‌باشد. با نمایش این بردار در هر حالت، کنترل ناظر توسعه یافته بصورت شکل ۹ بدست می‌آید (حالت‌های نشاندار مشخص نشده‌اند). به عنوان نمونه در حالت اولیه‌ی $X0$ نیاز به هیچ کدام از منابع نمی‌باشد پس $R_{X0} = 00$ خواهد بود. با وقوع پیشامد S_d سیستم به حالت $X1$ می‌رود که در آن فعالیت دریل کردن قطعه توسط روبات انجام می‌شود، پس در این حالت به روبات نیاز می‌باشد ولی به اپراتور تعمیرکار نیازی نیست، در نتیجه $R_{X1} = 10$ خواهد بود. در حالت $X2$ که متناظر با خراب بودن روبات است، نیز فقط روبات به عنوان منبع مورد استفاده قرار می‌گیرد پس $R_{X2} = 10$.



شکل ۹- مدل گسسته پیشامد کنترل ناظر توسعه یافته

مرحله‌ی آخر تبدیل کنترل ناظر توسعه یافته به کد LD می‌باشد. برای هر کدام از حالت‌های ناظر توسعه یافته به این صورت عمل می‌نماییم: با فرض آنکه حالت x دارای N مسیر خروجی باشد متناظر با هر کدام از این N مسیر که $\eta(\sigma_i, x) = y_i \quad i=1,2,\dots,N$ یک پله در دیگرام نردبانی LD در نظر گرفته می‌شود. خروجی‌های ست شونده و ریست شونده و ورودی‌های پله‌ی i ام دیگرام نردبانی (کد LD) به این صورت در نظر گرفته می‌شوند:

- [12] B. Brandin, "The Real-Time Supervisory Control of an Experimental Manufacturing Cell," IEEE Transactions on Robotics and Automation, vol. 12, No. 1, February, 1996.
- [13] W. M. Wonham, "Notes on Control of Discrete Event Systems," University of Toronto, 2006.
- [14] R. W. Lewis, "Programming industrial control systems using IEC 1131-3," 2nd ed, London, U.K.: The Institution of Electrical Engineers, 1995.
- [15] G. Frey and L. Litz, "Formal methods in PLC programming," presented at Systems, Man, and Cybernetics, 2000 IEEE International Conference on, 2000.
- [16] E. A. Parr, "Programmable Controllers, An engineer's guide," 3rd ed. Oxford, U.K.: Newnes, 1999.
- [17] F. Charbonnier, H. Alla, R. Daivid, "The Supervised Control of Discrete-Event Dynamic Systems," IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, vol. 7, NO. 2, MARCH 1999.
- [18] Enterprise-Control Systems Integration Part I: Models and Terminology," ANSI/ISA-95.00.01-2000, ISA, 2002.
- [19] H. Darabi, M. Jafari, S.S. Manapure, "Finite Automata Decomposition for Flexible Manufacturing Systems Control and Scheduling," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, vol. 33, NO. 2, MAY 2003.



شکل ۱۰- کد LD متناظر با کنترل ناظر توسعه یافته

مراجع

- [1] J. G. Ramadge, W.M. Wonham, "The control of discrete-event systems," Proc. IEEE, vol. 77 no. 1, pp. 81-98, 1989.
- [2] A. Afzalian, W.M. Wonham, "Discrete Event System Supervisory Controller Design for an Electrical Power Transmission Network," International Conference on Electrical Engineering ICEE, 2006.
- [3] R.J. Leduc and W.M. Wonham, "DISCRETE EVENT SYSTEMS MODELIN AND CONTROL OF A MANUFACTURING TESTBED," 1995.
- [4] S. Balemi, "Control of Discrete Event Systems: Theory and Application, Ph.D thesis, Swiss Federal Institute of Technology, Zurich, Switzerland,," 1992.
- [5] J. Liu and H. Darabi, "Ladder Logic Implementation of Ramadge-Wonham Supervisory Controller," Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02), 2002.
- [6] M. Fabian, A. Hellgren, "PLC-based Implementation of Supervisory Control for Discrete Event Systems ", Proceedings of the 37th IEEE conference on Decision & Control, Tampa, Florida, USA, 1998.
- [7] C. Music, D. Matko, "Discrete Event Control Theory Applied to PLC Programming" University of Ljubljana, Slovenia.
- [8] K. Latha, B. Umamaheswari, "Supervisory control of an automated system with ladder logic programming using Petri Net" Anna University, IEEE SMC TAICI, 2002.
- [9] S. S. Peng, M.C. Zhou, "Ladder Diagram and Petri-Net-Based Discrete-Events Control Design Methods," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, vol. 34, NO. 4, NOVEMBER 2004.
- [10] M.H. Queiroz, E.R. Cury, "Synthesis and Implementation of Local Modular Supervisory Control for a Manufacturing Cell," Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02), 2002.
- [11] I.T. Hasdemir, S. Kurtulan L. Geren, "IMPLEMENTATION OF LOCAL MODULAR SUPERVISORY CONTROL FOR A PNEUMATIC SYSTEM USING PLC," Copyright © 2004 IFAC.