

PLC기반 차체조립라인의 안전감시를 위한 진단프로그램 생성에 관한 연구

박 창 목*

*인덕대학 테크노경영과

Auto-Generation of Diagnosis Program of PLC-based Automobile Body Assembly Line for Safety Monitoring

Chang-Mok Park*

*Dept. of Technology & Systems Management, Induk University

Abstract

In an automated industry PLC plays a central role to control the manufacturing system. Therefore, fault free operation of PLC controlled manufacturing system is essential in order to maximize a firm's productivity. On the contrary, distributed nature of manufacturing system and growing complexity of the PLC programs presented a challenging task of designing a rapid fault finding system for an uninterrupted process operation. Hence, designing an intelligent monitoring, and diagnosis system is needed for smooth functioning of the operation process. In this paper, we propose a method to continuously acquire a stream of PLC signal data from the normal operational PLC-based manufacturing system and to generate diagnosis model from the observed PLC signal data. Consequently, the generated diagnosis model is used for distinguish the possible abnormalities of manufacturing system. To verify the proposed method, we provided a suitable case study of an assembly line.

Keywords: PLC, Diagnosis, Monitoring, Operational Fault

1. 서 론

PLC(Programmable Logic Controller)는 생산라인의 두뇌역할을 하는 소형 컴퓨터로써, 다중 입출력 함수와 로직 제어 함수로 이루어지는 프로그램을 구동한다. PLC 프로그램의 유연성과 하드웨어 견고성 때문에 대부분의 자동화된 생산라인의 제어에 사용되고 있으며, 특히 자동차 차체 조립라인에서 PLC는 세부 단위 설비로부터 전체 생산라인까지의 모든 요소를 포함하는 계층적인 형태로 수십대가 서로 연결되어, 모든 제어를 담당하는 아주 핵심적인 제어역할을 하고 있다. 이러한 차체 조립라인의 제어 프로그램은 입출력(I/O), 시퀀스(Sequence) 제어, 안전(Safety)에 해당하는 3가지 주요 기능을 포함하고 있

다. 이중 안전을 담당하는 프로그램이 전체 프로그램의 약 60% 이상을 차지할 정도로 매우 중요하다.

실제 공장의 동작 오류는 이러한 안전에 관련된 부분을 PLC에서 충분히 고려하지 않아 발생된다. 또한 이러한 동작 오류 이후에 라인 정지 시간의 약 80%를 프로그램상의 오류 위치를 찾기 위해 소비되고 있다고 조사 되고 있다[1]. 이러한 동작 오류는 고가 장비 파손, 라인 정지로 인한 생산량 저하로 이어져 막대한 손실을 야기하고 있다. 또한 새로운 라인을 구축하는데 있어, 이러한 안전에 관련된 제어 프로그램을 설계하고 개발하는데 많은 시간이 필요하고, 더욱 완벽한 구현을 위하여 시운전 시간을 거쳐 프로그램을 수정하고 있다.

PLC기반 시스템의 안전진단에 관한 기존 연구는, 오

† 본 연구는 인덕대학 학술연구비 일부 지원에 의해서 수행되었음.

† 교신저자: 박창목, 서울특별시 노원구 월계동 초안산실 14 인덕대학 테크노경영과

M · P: 010-4928-6634, E-mail: cmpark@induk.ac.kr

2010년 4월 15일 접수; 2010년 5월 25일 수정본 접수; 2010년 6월 1일 게재확정

류를 찾기 위한 방법론, 효율적인 신호 데이터 수집, 효율적인 진단프로그램 구조에 관한 내용으로 분류할 수 있다. Hao et al. [2]은 제어 시스템 모니터링이 가능하도록 신호 수집 및 분석을 위한 보조 도구를 개발하였다. 이러한 방법은 대부분의 PLC 모니터링 소프트웨어 제품들이 채택하고 있는 방법이다.

이러한 신호분석 도구들은 일정한 시간영역에서 주요한 신호들에서 변화를 표시하는 차트를 제공하고 있다.

W. Hu et al.[3]은 PLC에 의해 제어되는 생산시스템을 진단하기 위한 두 가지 모델을 제안하였다. 로직에 대한 진단을 위하여 LFC(Logical Function Chart), 동작 순서를 진단하기 위한 SCP(Sequential Control Process) 모델을 사용하였다. 여기서 제안된 방법은 시스템에 이상이 감지되면, PLC 관련 신호를 수집한 후 진단모델을 이용하여 예러가 발생된 프로그램 위치를 알아내는 방법이다. W. Hu et al.[4]은 또한 효과적인 예러 위치 탐색을 위하여 Fault tree analysis (FTA)를 제안하기도 했다. 신경망이 제어프로그램을 진단하기 위해 사용되기도 하였는데, Abdelhameed 와 Drabi [5]는 레더 로직으로 만들어진 제어 프로그램에 대한 정상적인 상태의 신호와 비정상적인 신호를 신경망으로 학습한 후, 실제 구동 중에 시스템의 이상 상황을 감지하였다. Theiss et al. [6]은 제어 시스템에 영향을 최소화할 수 있는 데이터 수집 에이전트를 이용하여 실시간 데이터 수집을 가능하게 할 수 있는 방법론을 제안하였다. Kaghazchi et al.[7]은 OPC를 이용하여 네트워크상에 존재하는 PLC 관련 신호들을 수집하고 진단하기 위한 시스템 구조를 제안하였다. 유정봉[8]은 SFC(Sequential Function Chart)를 이용한 제어 프로그램에 자기진단기능을 효율적으로 처리하는 방법을 제시하였다. 이는 전체 시스템의 진단이 아니라 제어코드 자체에 대한 진단을 다루고 있다. 성길영[9]은 PLC 생산공정에서 단순한 두 신호간의 시간차이를 모니터링에 사용하였다.

지금까지 기존 연구에서 사용된 진단 모델은 제어엔지니어가 직접 진단 프로그램을 설계해야 하기 때문에 설계 능력에 매우 의존적이며, 제어 프로그램 작성의외 추가적인 작업을 해야 하는 부담을 가지고 있다. 반면 본 연구에서는 제어 프로그램을 테스트 및 수정하는 시운전 기간 동안 축적된 데이터를 활용하여, 관측된 신호들의 변화를 분석하여 제어 프로그램 명세서에서 추출하기 어려운 신호레벨의 정상적인 상태 변화 패턴을 추출하여 안전 진단 시스템을 자동 생성하는 방법론을 제안하고자 한다. 생성된 진단 시스템은 생산시스템의 각 디바이스를 실제 연동하여 얻어진 패턴이기 때문에 더욱 정밀한 모니터링을 가능하게 한다.

2. 차체 조립 라인 의 제어 시스템

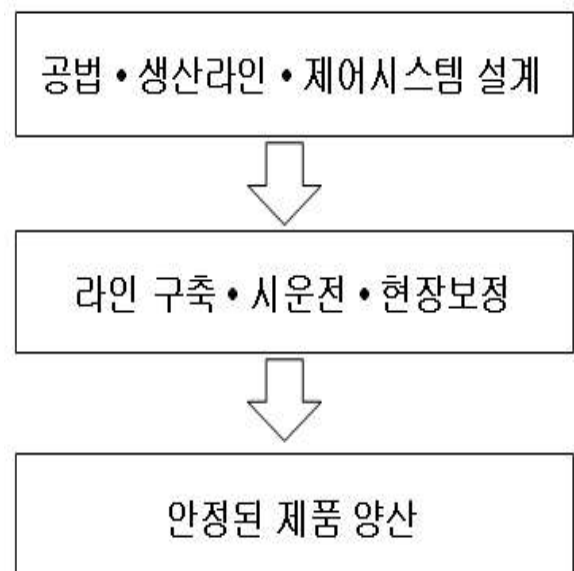
2.1 제어 프로그램에서 안전 고려 사항

차체 조립라인에서 고려되고 있는 안전에 대한 프로그램은 인터록(Interlock) 체크 라고 하는 정적 상태 체크와 시퀀스(Sequence) 체크라는 동적 변화 체크로 이루어져 있다. 인터록체크는 동시에 ON되지 말아야할 디바이스 출력에 대한 체크로서 실제 이러한 일이 발생할 경우 예러 신호를 발생시켜 시스템이 더 이상 진행하지 않도록 하는 것이다. 시퀀스 체크는 작업의 선후행 관계를 체크하는 것으로 선행 디바이스들의 작업 진행여부를 메모리에 기억시킨 후에 해당 작업의 진행 여부를 결정하는 프로그램이다. 차체 조립라인은 많은 설비들이 함께 작업하기 때문에 안전을 고려한 프로그램 설계가 매우 까다롭고, 실제 시운전을 통해 주변 설비와 연동하고 나서야 발견되는 위험요소들이 많다. 특히 이러한 안전 관련 프로그램들은 주로 논리 회로를 이용하여 구현되므로, 시스템 상태의 시간적 변화에 대한 모니터링이 취약하다.

2.2 PLC 연동 시운전

차체 조립 라인 구축은 <그림 1>과 같이 설계, 라인 구축 및 제품 양산이라는 3단계의 주요 업무를 거치게 된다. 제어 프로그램은 라인 설계의 마지막 단계에 설계가 되며, 초기에는 주요 시퀀스 제어로직을 구현한다.

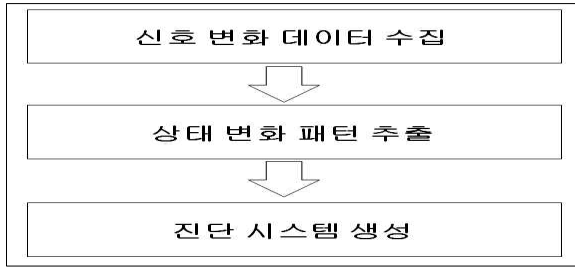
중간 단계인 라인구축단계에서는 시운전 및 현장보정을 통해 오류감지 및 모니터링에 관한 부분을 수정보완하게 된다. 이때 시운전은 <그림 2>와 같이 PLC를 실제 공장 혹은 가상 공장에 연결하여 테스트하게 된다.



<그림 1> 차체조립 라인 개발 절차



<그림 2> PLC 연동 시운전 모습



<그림 3> 제안된 방법론

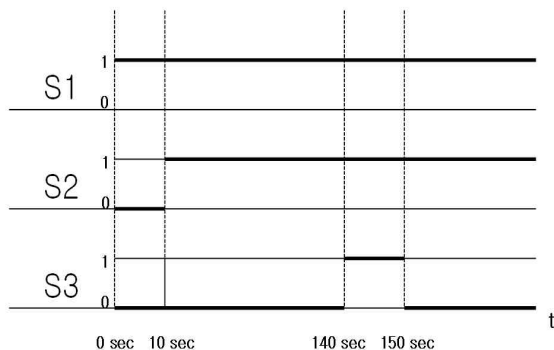
3. 제안된 방법론

실제 PLC를 연동한 시운전은 매우 정상적인 상태에서 진행되므로 초기에 개발된 제어로직이 안전에 대한 고려가 미흡하더라도 생산시스템은 잘 작동하기 마련이다. 이러한 시운전에서 관측된 정보는 생산시스템의 이상 징후를 판단하기 위해서 활용될 수 있다.

제안된 방법은 <그림 3>과 같이 첫째 시운전을 통한 신호변화 데이터 수집, 둘째 신호변화 데이터로부터 상태 변화 패턴 추출하기, 셋째 상태 변화 패턴으로부터 모니터링 및 진단용 프로그램 생성으로 요약할 수 있다.

3.1 시운전을 통한 신호변화 데이터 수집

다수의 시운전 기간 동안 PLC 신호들의 변화는 신호 시간 차트(Signal time chart)형식으로 표현하면 <그림 4>와 같은 형태를 가진다.



<그림 4> 신호시간차트(Signal time chart)

<표 1> 신호변화 데이터 예

i	t_i	e_i
0	0	-
1	10	2
2	140	3
3	150	-3

<그림 4>는 OPC 서버에 연결된 각 작업셀의 PLC로부터 3 개의 신호(S1, S2, S3)에 대한 변화를 나타낸 것으로, 그림에서 보는 바와 같이 3개의 시간지점에서 발생하는 상승에지(Raising edge)와 하강에지(Falling edge) 이벤트로 표현될 수 있다. 개수가 N 개인 신호이름에 고유한 번호 s ($s=1,2,\dots,N$)를 할당하고 상승에지일 경우 $+s$, 하강에지일 경우 $-s$ 로 표현하면 i 번째 시간지점 t_i 에서의 이벤트 e_i 는 $-N \sim N$ 사이의 정수로 표현될 수 있다. 예를 들면 <그림 4>의 예는 아래와 같은 <표 1>로 저장되게 된다.

3.2 상태 변화 패턴 추출

PLC에 의해 제어되는 시스템의 상태, q_i 는 <표 3>와 같이 각 신호값의 조합으로 표현될 수 있다.

시스템의 상태를 가능한 적은 정보로 표현하는 것이 진단프로그램의 사이즈를 줄일 수 있으므로, 신호의 상태가 변화가 없는 신호는 고려하지 않는다. 위 예에서 S1은 변화가 없으므로 고려하지 않고 S2와 S3로 시스템 상태를 <표 3>과 같이 정의할 수 있다. 이때 신호이름과 신호상태정보를 모두 표현하기 위하여 3.1절에서 언급한 e_i 와 같은 정의 방법을 사용할 수 있다.

<표 2> 시스템의 상태 정의

시간		$t_0=0$	$t_1=10$	$t_2=140$	$t_3=150$
신호 이름	고유 번호	q_0	q_1	q_2	q_3
S1	1	1	1	1	1
S2	2	0	1	1	1
S3	3	0	0	1	0

<표 3> 최소정보를 사용한 시스템 상태 정의

시간		$t_0=0$	$t_1=10$	$t_2=140$	$t_3=150$
신호 이름	고유 번호	q_0	q_1	q_2	q_3
S2	2	-2	2	2	2
S3	3	-3	-3	3	-3

이제 각 상태 q_i 는 i 상태에서 각 심볼의 상태를 “:” 로 구분하여 연결한 문자열로 표현하게 되면 위의 예의 4개 상태는 $q_0 = -2 : -3, q_1 = 2 : -3, q_2 = 2 : 3, q_3 = 2 : -3$ 와 같은 형태로 표현할 수 있다.

상태전이함수를 $\delta(A, E|\Delta_t) = B$, 즉 “현재 상태가 A 이고 Δ_t 시간 이후에 이벤트 E 가 발생하면 상태 B 로 변한다” 라고 정의하면, <표 4>로부터 아래와 같은 3가지 상태전이함수를 정의할 수 있다.

$$\delta(q_0, e_1|\Delta_1) = q_1, e_1 = 2, \Delta_1 = t_1 - t_0 = 10 \quad (1)$$

$$\delta(q_1, e_2|\Delta_2) = q_2, e_2 = 3, \Delta_2 = t_2 - t_1 = 130 \quad (2)$$

$$\delta(q_2, e_3|\Delta_3) = q_3, e_3 = -3, \Delta_3 = t_3 - t_2 = 10 \quad (3)$$

실제 이러한 상태천이를 추출하면 매우 다양한 Δ_t 를 가지는 천이함수를 수집하게 된다. 이러한 상태천이는 아래와 같이 두 가지 그룹으로 분류할 수 있다.

- 비관리 대상 천이함수: 비교적 작은 Δ_t 를 가지고 있으며, 제어프로그램 내에서 신호 값을 변경함으로 인해 관측되는 천이함수
- 관리대상 천이함수: 비교적 큰 Δ_t 를 가지고 있으며, 물리적인 기계동작의 완료 후에 발생하는 종료신호 및 센서 감지로 인해 관측되는 천이함수

위에서 설명한 관리대상 천이함수가 주요 모니터링 대상이므로, 일정 크기 이상의 Δ_t 를 가지는 상태천이함수를 사용하는 것이 합리적이라 볼 수 있다. 또한 너무 작은 Δ_t 를 가지는 상태천이함수를 이용하여 진단프로그램으로 사용할 경우 시간측정의 어려움으로 인해 잘못된 오류 알람(False alarm)을 만들 가능성이 커진다. 이러한 Δ_t 의 최소값을 Δ_{Min} 이라 정의하자. Δ_{Min} 의 선택은 생산 라인의 특성, PLC 통신 성능, 관측된 천이들의 시간특성을 분석하여, 비관리대상 천이함수와 관리대상 천이함수를 분리할 수 있도록 적절히 결정하여야 할 것이다. 위 예제에서 $\Delta_{Min} = 100$ 로 설정한다면 $\delta(q_1, e_2|130) = q_2$ 만이 관리되어야 하는 상태천이함수가 된다.

상태천이함수의 선택에 있어서 또 다른 고려사항은 반복성이다. 즉 반복적인 시운전을 통해 동일하게 반복되는 상태천이함수를 선택하여야 시스템의 정확한 상태천이함수를 추출할 수 있다. 만일 10번의 시운전을 하였는데, 1번만 관측된 상태천이함수는 생산시스템의 올바른 상태천이함수로 보기 어렵다는 것이다. 이때 정확히 10번 추출된 상태천이함수만이 시스템을 정확히 표현하는 것으로 간주해야 하는 것이다.

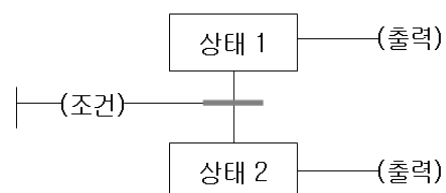
3.3 진단 프로그램 생성

앞 절에서 추출된 상태천이함수를 기반으로 실제 라인 구동 중에 시스템을 모니터링하고 진단할 프로그램을 생성하는 단계이다. 만일 $\delta(A, E|\Delta_t) = B$ 와 같이 상태천이함수가 존재하고, Δ_t 를 기준으로 시간에 대한 허용값 τ 를 정의할 수 있다면, 주어진 상태천이함수를 이용하여 생산 시스템이 상태 A 가 되고나서 θ 만큼의 시간이 흐른 후 아래와 같은 3가지 경우를 정의할 수 있다.

- Case 1: 이벤트 E 발생 시점이 θ 이고, 식 $\theta \leq (\Delta t - \tau)$ 을 만족하게 되면 시스템은 이상 징후를 나타내고 있다. (즉 정상변화 보다 빠른 상태변화 발생)
- Case 2: 이벤트 E 가 발생되지 않았고, 식 $\theta \geq (\Delta t + \tau)$ 을 만족하게 되면 시스템은 이상 징후를 나타내고 있다. (즉 허용된 시간이 후에도 상태변화 E 가 발생하지 않음)
- Case 3: 이벤트 E 발생 시점이 θ 이고, 식 $(\Delta t - \tau) < \theta < (\Delta t + \tau)$ 을 만족하게 되면 정상적인 상태변화가 관측된 것이다. (즉 허용된 시간 내에 상태변화 E 가 발생)

각 상태천이함수에 대한 진단프로그램은 위 3가지 경우를 반복적으로 점검하는 함수형태로 생성될 수 있으며, 생성된 프로그램은 실제 공장의 안전진단 시스템으로 활용될 수 있다.

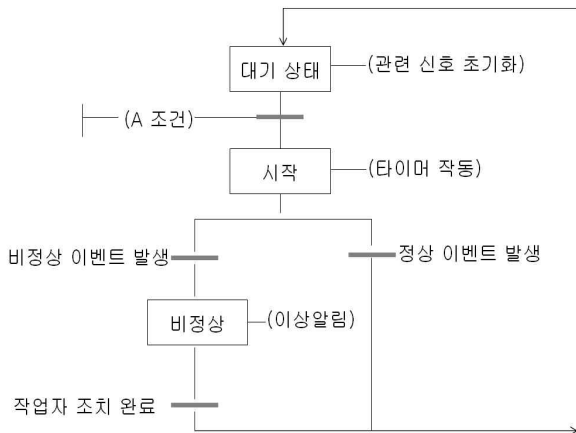
실제 자동차 차체 조립 라인의 PLC 제어 프로그램은 레더다이아그램(LD, Ladder Diagram)으로 작성되나, 본 연구에서는 생성된 제어프로그램은 이해를 쉽게 하기 위해 SFC(Sequential Function Chart)를 사용하였다. 본 연구에서 생성된 SFC 프로그램은 쉽게 LD형태로 변환이 가능하다. <그림 5>에서 보는바와 같이 SFC는 상태(State)와 천이(Transition)로 구성되어 있으며, 상태는 박스 형태로 표현되며, 박스 오른쪽에 각 상태에서 실행되어야 할 출력이 표시된다. 이때 R은 Reset (FALSE로 설정)을 나타내고, S는 Set(TRUE로 설정)을 나타낸다. 또한 D는 Delay를 나타내며 지정된 시간 후에 값을 TRUE로 설정해주는 것을 말한다. 천이는 굵은 선으로 표현되며, 천이를 일으키는 조건은 왼쪽에 LD형태로 존재한다.



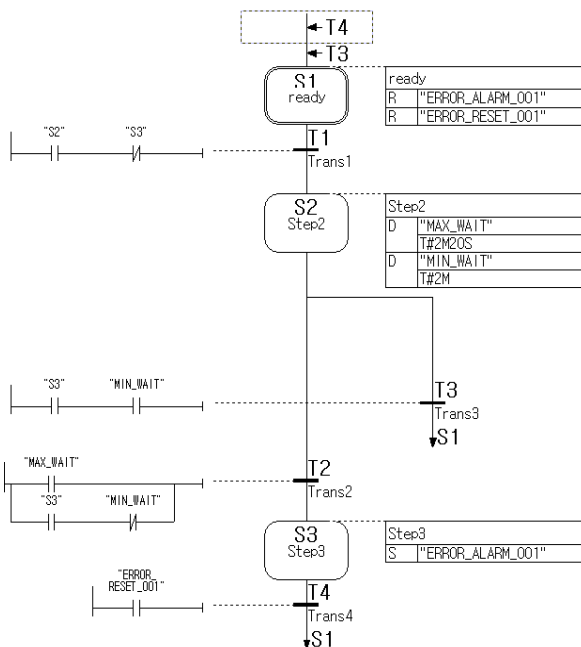
<그림 5> SFC 형태

SFC 로 상태전이 함수, $\delta(A, E|\Delta_t) = B$ 를 위 Case 1,2,3에 따라 체크하는 진단 프로그램 원형은 <그림 6>과 같은 기본 형태를 가지게 된다. <그림 6>에서 보는바와 같이 A조건은 진단함수에서 대기상태에서 시작상태로 변하는 조건이며, 추출된 상태전이 함수의 초기상태를 이용하여 생성하게 된다. 또한 정상 이벤트 발생과 비정상 이벤트 발생은 이벤트 E와 Δ_t 에 의해서 생성되게 되어 각 상태전이함수별 진단 함수를 자동 생성할 수 있게 된다.

앞절에서 생성된 상태전이 함수, $\delta(q_1, e_2|130) = q_2$ 를 이용하여 생성한 예는 <그림 7>과 같다. 여기서 시간에 대한 허용값 τ 은 10으로 설정하였으며, 진단프로그램에서 $\Delta_t - \tau$, $\Delta_t + \tau$ 는 각각 MIN_WAIT, MAX_WAIT 로 표시되었다.



<그림 6> SFC로 표현된 진단함수 원형



<그림 7> 예제에 대해 생성된 진단프로그램

앞 절의 예에서 생성된 상태전이 함수 $\delta(q_1, e_2|130) = q_2$ 를 이용하여 생성된 진단함수를 보면, 상태 S1은 초기상태를 나타내면 이때 Action은 다음 진단을 위하여 시스템의 이상 상태를 알려주는 ERROR_ALARM_001 출력값을 RESET 해주고, 이상 상태를 해제하여 주는 ERROR_RESET_001 입력값을 마찬가지로 RESET 해주게 된다. 이때 001과 같은 번호는 진단함수 번호를 의미한다. 첫 번째 천이 T1은 시작 상태가 되었음을 의미하는 것으로 이 때 천이 조건은 상태전이함수의 q_1 에 해당되는 조건이 왼쪽에 표현된다. 두 번째 상태 S2에서는 시간 측정을 위한 타이머를 가동하는 Action이 수행된다. 이때 앞에서 설명한 $\Delta_t - \tau$, $\Delta_t + \tau$ 두 값이 사용된다. 이제 진단함수는 두 개의 분기를 따라 가게 되는데, T3는 정상적인 천이를 의미하고, 즉 3.3 절에서 제시한 "Case 3"을 의미한다 이 때 진단함수는 다음 진단을 위하여 자동적으로 상태 S1으로 돌아가게 된다. T2는 비정상적인 천이를 의미하고 이때 S3로 이동하면서 ERROR_ALARM_001을 TRUE로 설정하여 시스템에 문제가 발생한 것을 알려주게 된다. 이 때 작업자가 문제를 확인하고 시스템을 다시 진단모드로 바꾸기 위해 수동으로 ERROR_RESET_001을 TRUE 바꾸게 되면, 진단함수는 다시 진단모드로 바뀌어 상태 S1으로 돌아가게 된다.

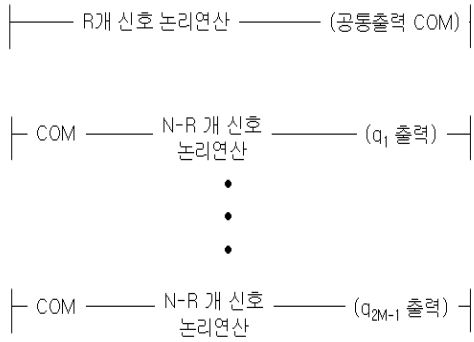
3.4 진단 프로그램 사이즈 최소화

진단 프로그램의 원형이 되는 상태전이 함수, $\delta(A, E|\Delta_t) = B$ 로부터 프로그램을 생성할 경우 모니터링을 시작하는 시작조건은 상태 A에 의해 표현되며, 만일 시스템이 N개의 신호들로 구성되어 있다면, 진단프로그램은 시작상태 A를 표현하기 위하여 N개의 신호를 논리 연산하는 프로그램이 필요하다. 또한 모니터링 대상이 되는 M개의 상태전이 함수를 추출하였다면, M개에 해당하는 논리함수(각 함수는 N개 신호를 논리 연산함)가 필요하게 된다. 본 연구에서는 이러한 프로그램 사이즈를 줄이기 위해 M개의 상태전이 함수의 시작상태를 종합적으로 관리하는 함수를 생성하여 사용하였다.

즉 M개의 상태전이 함수를 발생 시간 순서에 따라 아래와 같이 정의한다면

$$\delta(q_1, e_1|\Delta_1) = q_2, \delta(q_3, e_2|\Delta_2) = q_4, \dots, \delta(q_{2M-1}, e_M|\Delta_M) = q_{2M} \quad (4)$$

M개의 시작상태 $q_1, q_3, \dots, q_{2M-1}$ 를 표현하는데 있어서, 항상 동일한 상태를 유지하는 신호가 R개 있다면 이 신호들의 상태를 하나의 함수로 표현하여, 각 진단 프로그램에서 시작상태를 판단하기 위하여 공통적으



<그림 8> 시작 상태 모니터링 프로그램

로 쓰일 수 있다. 이 공통 함수를 활용하여 시작상태 $q_1, q_3, \dots, q_{2M-1}$ 을 판단하는 <그림 8>과 같은 프로그램이 생성 가능하다. 결과적으로 M개의 진단프로그램에서 각 시작을 알리는 상태 A의 연산프로그램을 사용하지 않고, <그림 8>과 같이 생성된 프로그램에서 각 시작 상태를 알리는 출력값을 사용하면 프로그램 사이즈를 줄일 수 있다.

4. 사례 연구

본 연구에서는 제안된 방법론을 적용하기 위하여 자동차 차체 조립 라인의 시운전 방식을 그대로 사용하였다. 현재 국내 자동차회사에서 사용하는 PLCSTUDIO라는 시뮬레이션 소프트웨어를 사용하여 <그림 9>와 같은 가상생산라인을 꾸미고, 이를 제어하는 Siemens 기반 PLC 제어 환경을 구축하였다. 실제 시운전은 컴

퓨터상에서 모두 이루어진다.

본 연구를 위해 개발된 데이터 수집 및 진단모델 생성 시스템을 사용하여 신호들의 변화를 저장하였다. 시운전은 4번 반복하였다.

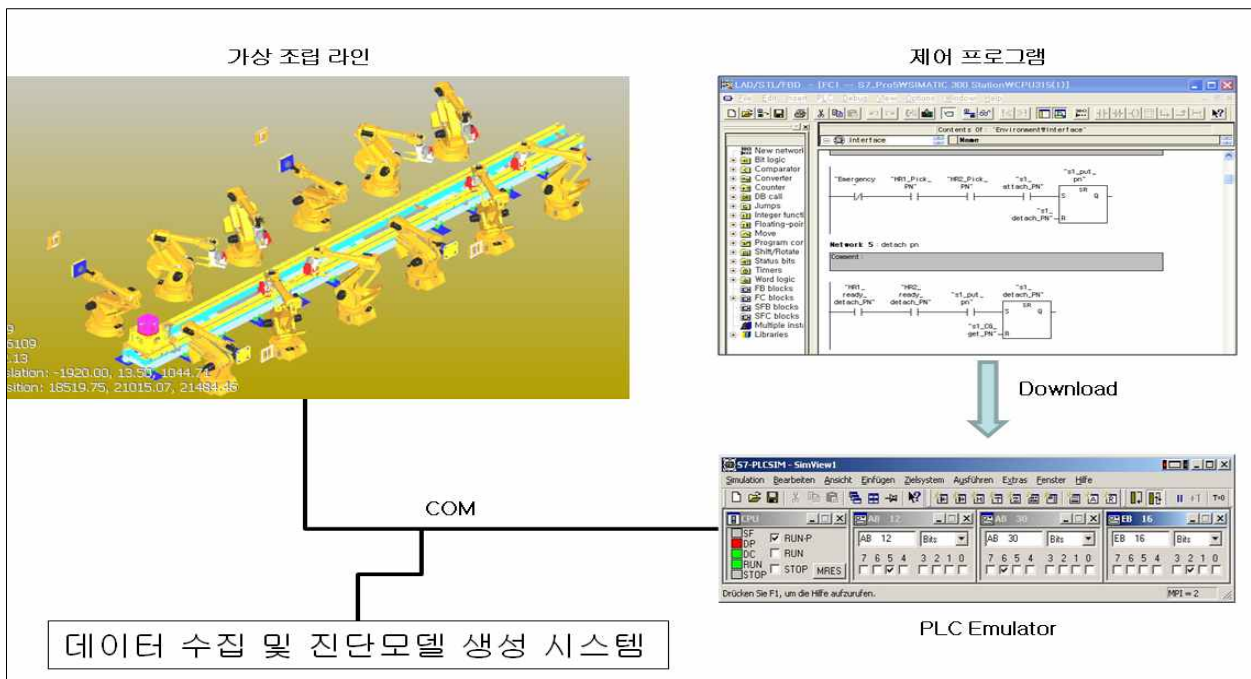
가상 생산라인은 4개의 운송로봇, 8개의 용접로봇 그리고 단일 컨베이어로 구성되어 있다. 작업물이 4개의 Station으로 이동하면서, 필요한 작업물 부착 및 로봇 용접을 수행하는 조립작업을 수행한다. 생산라인을 제어하는 PLC 프로그램은 94개의 제어 신호를 사용하며, 94개 신호의 값에 따라 시스템의 상태가 결정된다.

앞서 제시한 신호들의 변화를 숫자로 표현하기 위하여 각 신호들에 유일한 번호를 부여하고 값이 TRUE가 되면 양수, FALSE가 되면 음수로 표현하여, 각 시간별로 이벤트를 수집한 결과는 <그림 10>과 같다.

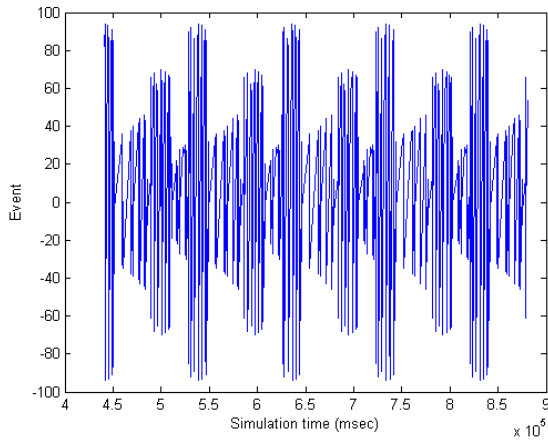
그림에서 보는 바와 같이 4번 반복되는 시스템의 변화들을 관측할 수 있다.

또한 상태천이함수중 4번 반복된 것들의 평균 Δ_t 를 그래프로 표현한 것은 <그림 11>과 같다. 가상 생산설비의 동작시간의 최저값은 2000 msec이기 때문에 그림에서 보는 바와 같이 Δ_t 가 2000 msec 이상 되는 천이함수들을 추출하게 되면 16개의 관리대상 상태천이함수를 분리할 수 있으며, 이를 이용하여 16개의 진단함수를 생성할 수 있었다.

16개의 상태천이를 정의하기 위한 32개의 상태중 동일한 상태를 유지하는 신호(45개)를 제거하고 남은 49개 신호로 표현한 16개의 상태천이함수는 <그림 12>과 같다. <그림 12>에서는 94개 신호를 각각 고유한 번호로 표현하고, 신호의 상태를 1 혹은 0로 표현하였다.



<그림 9> 사례 연구를 위한 가상 시운전 환경



<그림 10> 시운전을 통한 신호변화 추출

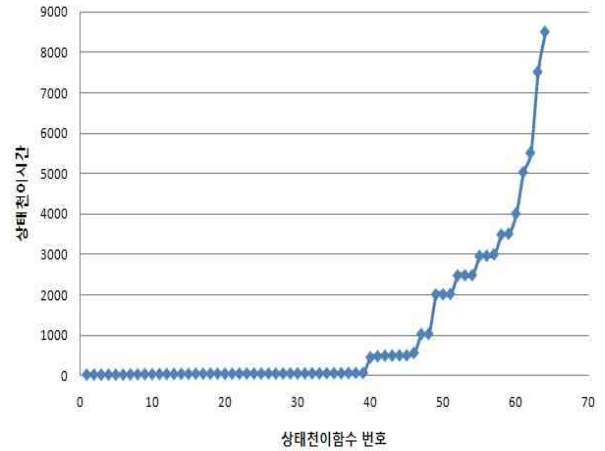


그림 11 관측된 상태전이함수의 Δ_t

s	#1 e1=74	#2 e2=93	#3 e3=79	#4 e4=82	#5 e5=54	#6 e6=41	#7 e7=45	#8 e8=33	#9 e9=37	#10 e10=29	#11 e11=25	#12 e12=14	#13 e13=13	#14 e14=12	#15 e15=11	#16 e16=10
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
10	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1
11	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
32	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
35	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
36	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
39	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
43	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
44	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
49	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
66	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
71	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
72	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
74	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
76	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
77	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
83	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
84	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
85	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
88	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
89	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<그림 12> 추출된 상태 천이 함수

<그림 12>에서 행은 천이 함수를 표현하기 위한 신호번호를 나타내며, 열은 각 상태천이함수의 시작과 종료상태를 나타낸다. 천이이벤트가 나타나는 신호 상태 영역에는 직사각형을 표시하였다.

상태중 동일한 상태를 유지하는 신호(45개)를 공통신호(COM)으로 표현하고, 각 상태천이함수의 시작상태, 즉 q_1, q_3, \dots, q_{31} 를 모니터링하는 프로그램이 자동 생성되며, 한 예로서 q_1 과 q_3 상태를 모니터링하는 로직은

<그림 13>와 같다.

생성된 진단함수의 한 예는 <그림 14>이다. 이 진단함수는 16개의 진단함수 중 1번 상태전이함수(천이 이벤트는 74번 신호가 '1' 상태가 되는것, Δ_t 는 2000 msec)를 이용하여 생성된 것이다. 이때 허용값 τ 는 100 msec으로 가정 하였다.

5. 결 론

본 연구에서는 자동차 차체 조립 라인에서 활용할 수 있는 생산시스템 진단프로그램의 자동생성 방법론을 제시하고 있다. 실제 라인에서는 진단프로그램의 중요성이 커져가고 있으며, 이를 설계하고 구현하기 위해

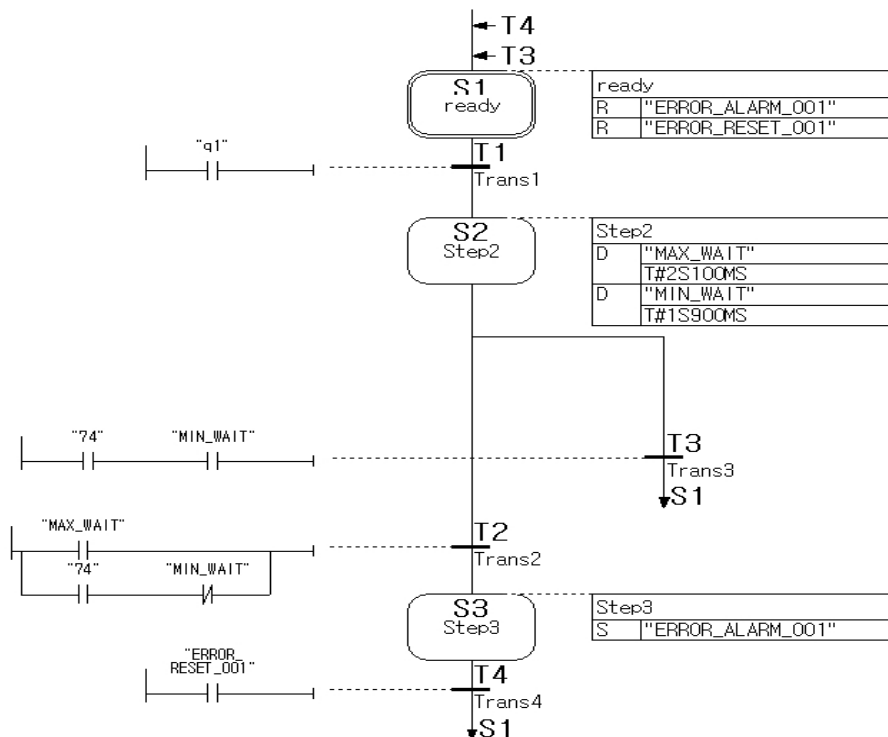
서 많은 시간과 노력이 필요하다. 이러한 환경을 개선하기 위하여 자동화된 방법론이 필요하다.

본 논문에서는 차체 조립라인에서 매번 수행하는 시운전기간동안 획득된 데이터를 활용하여 매우 정확한 진단프로그램을 생성 할 수 있음을 사례연구를 통해 확인하였다. 그러나 관리대상 상태전이 함수를 분리하는 적절한 Δ_{Min} 의 선택 및 허용값 τ 의 선택에 대한 더욱 심층적인 고찰이 필요하다.

본 연구가 실제 산업에 사용되기 위해서는 2가지가 고려되어야 한다. 첫 번째 생성된 진단 프로그램의 사이즈를 최소화 하도록 진단함수를 최적화 하는 것 과 두 번째는 현장에서 사용되는 PLC에 적합한 제어 언어로 변환되어야 한다는 것이다.

-1&-2&-8&15&-16&-17&-18&19&-20&-21&-22&-26&-30&-34&-38&-42&-46&-47&-50&-51&-52&-53&-55&-56&-57&-58&-59&-60&-62&-63&-64&-65&-67&-68&-69&-70&-75&-78&-80&-81&-86&-87&-90&-91&-92&-94	=	COM		
COM	&	-3&-4&-5&-6&-7&-9&-10&-11&-12&-13&14&23&-24&-25&27&-28&-29&31&-32&-33&35&-36&-37&39&-40&-41&43&-44&-45&-48&49&-54&61&-66&-71&-72&-73&-74&-76&77&-79&-82&-83&-84&-85&-88&89&-93	=	q1
COM	&	-3&-4&-5&-6&-7&-9&-10&-11&-12&-13&14&23&-24&-25&27&-28&-29&31&-32&-33&35&-36&-37&39&-40&-41&43&-44&-45&-48&49&-54&61&-66&-71&-72&-73&-74&76&-77&-79&-82&-83&-84&-85&88&-89&-93	=	q3

<그림 13> 시작상태 q1 과 q3 모니터링 로직



<그림 14> 생성된 진단함수 1번

6. 참 고 문 헌

- [1] R.L.Kegg, On line machine and process diagnostics, Annals of the CIRP,32(2), 1984, pp. 469-473
- [2] Hao Zhang, Jianfeng Lu, Yunjun Mu, On-line PLC Monitoring and Network Administering System for Steel Tube Mill, International Conference on Industrial Technology, 1999, pp 720-723
- [3] W. Hu, A.G. Starr, A. Y. T. Leung, Two diagnostic models for PLC controlled flexible manufacturing systems, International Journal of Machine Tools & Manufacture 39, 1999, pp. 1979-1991
- [4] W. Hu, A.G. Starr, A. Y. T. Leung, Operational fault diagnosis of manufacturing systems, Journal of Materials Processing Technology 133, 2003, pp. 108-117
- [5] Magdy M. Abdelhameed, Houshang Darabi, Diagnosis and Debugging of Programmable Logic Controller Control Programs by Neural Networks, International Conference on Automation Science and Engineering, 2005, pp. 313-318
- [6] S. Theiss, J. Naake, H. Dibowski, K. Kabitzsch, PLC-integrated Process Monitoring and Prediction of the Resulting Real-Time Load, IEEE International Conference on Industrial Informatics, 2006, pp. 880-885
- [7] Kaghazchi, H., Hayes, J., Heffernan, D. Development of an OPC Server for a Fieldbus Diagnosis Tool. IEEE Proceedings on Industrial Informatics, INDIN '07. IEEE Electronics Society. Vienna . June 2007
- [8] 유정봉, SFC로 설계된 PLC 제어에서 효율적인 자기진단 기능, 조명·전기설비학회논문지 22(4), 2008, PP. 153-158
- [9] 성길영, 한관희, 변재정, 왕지남, 박상철, PLC 신호의 시간차이를 이용한 자동화 공정의 PLC 프로그램 모니터링, 한국 CAD/CAM 학회 논문집 14(3), 2009, PP. 176-185

저 자 소 개

박 창 목



아주대학교 산업공학과에서 학사, 석사, 박사학위를 취득하였고, 지식시스템 선임연구원, 유디엠텍 대표이사를 거쳐 현재는 인덕대학 테크노경영과 전임강사로 재직 중이다. 관심분야는 생산최적화, 생산자동감시제어, 시뮬레이션 등이다.

주소: 서울시 노원구 월계동 인덕대학 테크노경영과