

PLC 신호의 시간차이를 이용한 자동화 공정의 PLC 프로그램 모니터링

성길영*, 한관희**, 변재정***, 왕지남****, 박상철*****

PLC Program Monitoring for Manufacturing Systems Using PLC Signal Time Difference

KilYoung Scong*, KwanHee Han**, JaiJeong Pyun***, GiNam Wang**** and SangChul Park*****

ABSTRACT

Modern manufacturing systems consist of highly automated manufacturing devices, and they are controlled by complicated PLC programs. To make sure the achievement of the control objectives of a manufacturing system, it is very important to monitor the dynamic system behaviors of the manufacturing system. In this paper, we propose a monitoring methodology of a PLC program based on the Software In the Loop Simulation(SILS), which makes use of the time gap information between PLC signals. The errors relevant with PLC signals can be found using the proposed methodology, comparing a normal PLC signal trajectory with a target PLC signal trajectory. The proposed methodology has been implemented and tested with simple examples.

Key words : PLC(Programmable Logic Controller), PLC Monitoring, Automated manufacturing system

1. 서 론

PLC는 저 수준의 프로그램 언어의 불리언 연산을 통해 자동화 제조 시스템을 제어한다. IEC6113-3, 4의 표준에 따르면, Ladder Logic Diagram(LD)은 PLC에서 일반적으로 사용되는 프로그램 언어로 알려져 있다. Fig. 1은 로봇이 부품(판넬)을 잡기 위한 동작을 위한 신호를 출력하는 LD 언어의 예이다. 자동화 제조공정에서 설계된 PLC 프로그램은 검증과정을 거쳐서 프로그램의 정확성이 보장되어야 비로소 실제 시스템에 사용된다¹⁾. 오늘날의 자동화된 제조 과정은 매우 복잡하게 설계되고 있으며 PLC 프로그램도 매우 복잡하게 설계된다. 이에 따라 PLC 프로그램의

에러를 진단하는 방법은 점점 더 어려워지고 있으며, 에러를 디버깅하는데 걸리는 시간이 점점 증가하고 있다. Arbor은 “제조공정에서 에러를 규명하는데 걸리는 시간으로 인한 작업 중 지연은 전체 머신 고장시간의 80%를 넘는다.”라고 밝히고 있다²⁾.

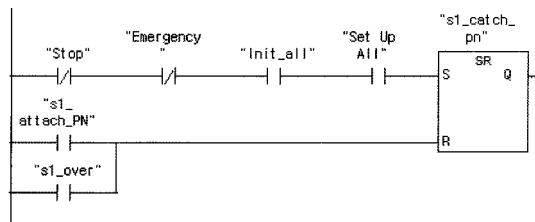


Fig. 1. A PLC program example(LD).

일반적인 자동화 시스템은 수많은 로봇과 자동화된 이송장치로 이루어져 있다. 로봇과 이송장치는 PLC 프로그램의 보직에 따라 용접이나 이송 등의 다양한 작업을 한다. 오늘날의 자동화 시스템은 대단위의 자동화 생산라인을 구축하고 있어 복잡도가 높아 장치 자체의 에러 또는 로봇 유통 범위 간섭과 같은 외부

*학생회원, 아주대학교 산업공학과
 **중신회원, 경상대학교 산업시스템공학부
 ***비회원, 국방과학연구소
 ****비회원, 아주대학교 산업정보시스템공학부
 *****중신회원, 아주대학교 산업정보시스템공학부
 - 논문투고일: 2008. 12. 08
 - 논문수정일: 2009. 03. 17
 - 심사완료일: 2009. 03. 20

요인에 의한 에러 등의 다양한 작업 실패 요소를 포함하고 있다. 작업이 실패하는 경우에는 공정의 오류를 발견하는 것이 어려울 뿐만 아니라, 장치를 셋업하기 위한 추가 시간이 필요하므로 경제적 손실이 매우 크다. 위의 문제를 해결하기 위해 제조 공정에서 사용되는 기존의 모니터링 방법은 몇 가지 한계점을 가지고 있다. 실제로 자동차 제조 공정에서는 작업 실패를 모니터링하기 위해서 에러의 가능성이 많은 작업 영역을 예상하여 해당하는 PLC프로그램에 작업 실패를 진단하는 진단선용 코드를 추가로 작성한다. 추가로 Table 1과 같은 미리 예상한 이상표시 모듈을 작성하여 PLC프로그램을 모니터링 한다^[1]. 이와 같은 방법은 에러의 발생 가능성이 높은 작업을 예상하여 에러 진단 목적의 코드를 별도로 작성하는 것이기 때문에 모든 신호를 모니터링 하는데 한계가 있다. 즉, 모니터링의 대상이 되는 공정은 매우 제한적이며, 점진적으로 발생하는 에러를 사전에 발견하기 어렵다.

Table 1. Error Indicating Module

이상표시모듈	1. CPU이상
	2. 공정이상
	3. 로봇이상
	4. 압축공기 이상
	5. 냉각수 이상
	6. 차종 이상
	7. 파트 장착 이상

PLC신호를 이용한 기존의 연구를 살펴보면 다음과 같다. Holloway는 이산사건 기반의 time templates를 이용한 모니터링 방법을 제안하였다^[2]. 하지만 이 방법은 공정규모가 커질수록 정의해야 할 state 수가 매우 많아지므로 대규모 공정의 적용에 어려움이 있다. Ioannides는 induction motor가 사용되는 시스템을 제어하기 위해서 모터의 V/f output 개념을 사용하였다^[3]. Zhang은 Steel Tube Mill을 모니터링하기 위해 PLC 신호를 이용하였다^[4]. 상기의 PLC신호를 통한 모니터링 방법은 특정 device에 제한적이어서 일반화하기 어렵다. 본 논문에서는 이와 같은 문제를 보완하기 위해서 PLC신호의 시간차이를 이용한 모니터링 방법을 제안하며 이것을 테스트하기 위하여 PlcStudioTM를 이용하였다^[5]. PlcStudioTM는 공정의 PLC를 검증하기 위한 상용 시뮬레이션 S/W로, Logical model과 Physical model로 구성된다. Logical model은 현장의 device를 PLC 신호의 입출력관점에서 Automata 기반의 I/O model로 플랫폼을 모델링하는데 사용된다. 다시 말해,

I/O model은 device의 제어를 위해 기 작성된 PLC 프로그램을 검증하는데 쓰인다. Fig. 2는 Finite State Automata(FSA)를 이용한 간단한 I/O model의 예를 보여준다. 입력신호 X1이 1의 값을 가질 때 State는 q0이다. Y1신호가 1의 값을 갖게 되면 state는 q0에서 q1으로 변화하며 X1의 값은 0이 된다. 마찬가지로 q1 state에서 Y2신호의 값이 1이 되면 q2로 변화한다. Physical model은 plant를 3차원 그래픽으로 모델링한 것으로 모델링 대상의 형상정보와 보전정보를 가지고 있다. Fig. 3은 실제 I/W PLC, Generic model 그리고 Graphic model이 연동되는 SII/S(Software In the Loop Simulation) 구조를 보여준다. 실제 플랫폼을 대신하는 Physical model의 device를 제어하기 위해서 PLC프로그램의 각 입출력신호를 모델링한 Logical model은 PLC프로그램의 신호에 따라 정의된 State의 변화에 따라 device의 보전을 제어하게 된다^[6]. 즉, PlcStudioTM는 제조 라인을 가시적으로 확인할 수 있는 physical model, 그리고 PLC 프로그램과 정보를 주고받고 physical model을 제어하는 logical model로 구성된 상용 S/W 도구로서, 이를 이용하여 제조라인을 구축하기 전에 PLC 프로그램을 검증하여 제조라인을 구축하는데 걸리는 시간을 줄이고 실제 라인의 운영단계에서 에러로 인한 비용을 줄일 수 있다. 본 논문에서는 실제로 제조공정에서 발생 가능한 에러유형을 실제 제조 라인에 적용하는 것은 비용, 시간적 측면에서 매우 비효율적이므로 실제로 발생 가능한 에러를 반영하여 시뮬레이션 할 수 PlcStudioTM를 이용하여 간단한 예제공정을 모델링하고 모니터링 방법을 설명하였다.

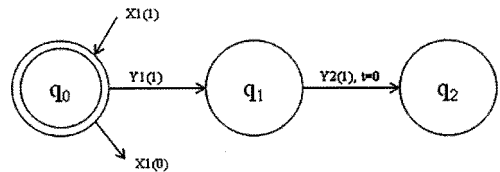


Fig. 2. An I/O model based on FSA.

본 논문의 구성은 다음과 같다. 먼저, PLC신호를 모니터링 하는 방법을 설명하기 위하여 진단되어야 할 작업 실패 상황의 정리가 필요하므로 2장에서는 이를 위해 자동화 생산 라인으로 이루어진 시스템에서 발생할 수 있는 에러를 정리한다. 3장과 4장에서는 기술한 에러를 진단 및 확인하기 위해 방법을 PlcStudioTM를 이용한 예제를 통해 제안한다. 마지막으로 5장에서 제안하는 방법의 효과 및 결론을 기술하고 향후 연구 방향을 기술하였다.

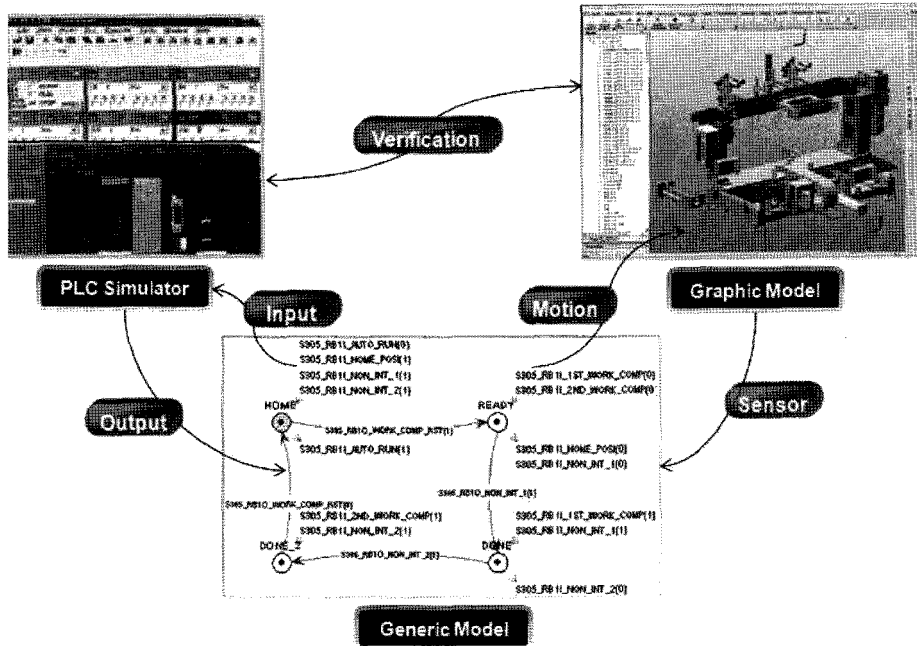


Fig. 3. SILS: A Logical(generic) & physical(graphic) simulator (PlcStudio™).

2. 에러 및 원인의 유형

자동화 시스템에서의 에러 유형을 나누어 보면 크게 제품을 운반하는 device에 대한 것과 용접 등의 기능을 하는 device에 관련된 오류로 나눌 수 있다.

2.1 운반과정에서의 에러 유형

일반적으로 자동화 시스템은 파드의 운반을 담당하는 컨베이어, 시분, 핸들링 로봇으로 이루어진다. 이러한 장치에서 발생가능한 작업 실패 상황들을 정리해보면 다음과 같다. 이송장치의 전진/후진 이상상황이 발생할 수 있고 인버터 제어, 혹은 서보 제어 드라이브 등에 이상이 발생할 수 있을 수 있다. 또한 드라이브 자체 이상으로 작업 실패 현상이 발생할 수 있다. 그리고 기계적 간섭으로 인한 과전류의 발생으로 스위치의 전원차단 현상이 발생할 수 있다. 이러한 에러 상황들을 컨베이어 등의 이송장치가 사용되는 자동화된 물류 라인에서 주로 발생하는 에러 요인이다.

2.2 조립공정의 에러 유형

제품을 가공하기 위해 사용되는 대표적인 device로서 다양한 산업용 로봇들이 사용된다. 이러한 로봇의 경우 발생할 수 있는 에러 요소는 로봇 자체의 에러와 제품을 고정하는 지그 또는 전에 대한 오류로 나눌 수

있다. 로봇에 대한 오류로는 로봇 하드웨어 기판오류로 인한 시스템 이상, CPU의 셋업 실패 등으로 발생하는 컨트롤러의 이상 등이 있으며 모두 PLC 프로그램 버그가 아닌 설비 자체에서 일어나는 오류이다. 주변 설비인 로봇 건이나 지그의 오류는 예를 들어 제품을 집기위해 클램프를 열고 닫는 운동을 유발하는 출력 신호가 하드웨어에 주어졌는데 그에 대한 피드백, 즉 완료신호가 PLC로 들어오지 않는 경우가 있다. 이외에도 다른 주변 설비인 냉각수상치나 압축공기장치 이상 등의 오류가 있을 수 있다.

2.3 에러의 원인

앞서 말한 물류 장치와 가공 장치들의 다양한 오류들은 하드웨어 자체가 가진 오류 즉, 하드웨어에 포함된 기판의 오류 혹은 하드웨어의 컨트롤러 오류이다. 이러한 오류는 부품의 파손 혹은 마모 등의 이유로 발생하며 특정 보선의 생각 또는 지연으로 인하여 작업 간 시간차이를 변화시킨다. 이 외에도 프로그램 작성자의 실수로 발생하는 오류가 있다. 이러한 에러는 주로 PLC 신호들의 종속관계로 인해 발생하는 것인데, 이것은 PLC 프로그램의 자체 버그라 할 수 있으며, 제조 라인의 테스트 및 안정화를 위한 시운전 과정에서 검증되었다고 가정한다. 본 논문은 PLC 프로그램이 아닌 실제 작업 중 외부 요소에 의한 에러로 인한

작업 실패를 진단하는 것이 목적이므로 이러한 PLC 프로그램상의 에러는 제한한다.

3. PLC 신호의 시차를 이용한 모니터링

3.1 시차기반의 모니터링 방법의 개요

PLC 프로그램 내에는 수많은 입출력 신호들이 사용되는데, 여기서 제안하는 방법은 다양한 입출력 신호들이 정상적으로 ON/OFF 되는 것을 모니터링하기 위하여 각각의 신호들 간의 시간차를 이용한다. Fig. 4는 본 논문에서 제안하는 모니터링 방법의 순서도를 나타내고 있다. 제안하는 모니터링 방법을 간단하게 설명하면 먼저 공정의 시운전 과정에서 정상 상태의 PLC 신호 시퀀스와 각 신호간의 시간차를 알 수 있다. 시운전을 통해 라인이 안정된 후 에러가 생길 수 있는 양산 과정에서도 PLC 신호의 시퀀스에 따른 시간차를 알 수 있다. 제안하는 방법을 통해 신호의 시간차이에 대한 데이터를 비교하여 양산라인이 가질 수 있는 장치나 부품의 마모에 따른 에러를 사전에 발견하고 경고할 수 있다.

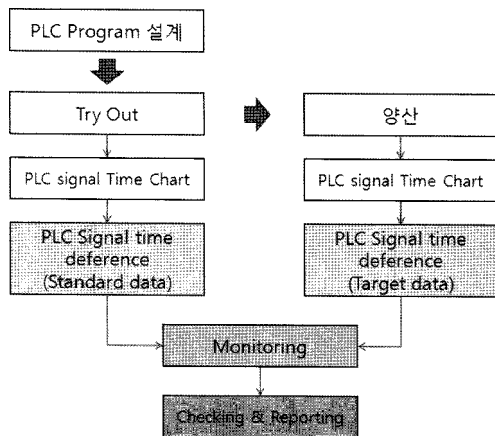


Fig. 4. A Flow chart of PLC signal monitoring.

3.2 시운전 과정(Try-Out Run)의 PLC 신호 데이터

자동화 제조 공정에서는 일반적으로 제조 시스템이 설치되었을 때, 라인의 테스트 운영을 통해 정상 제품을 생산하기 위한 표준적인 작업 절차가 수립된다. 라인의 테스트와 정상 제품을 생산하기 위한 안정화 과정을 거치면서 자동화 제조 시스템은 안정되며, 이에 따라 시스템을 제어하는 PLC 프로그램의 검증도 완료된다. 여기서 우리는 정상적인 제품을 만들기 위한 각 신호의 ON/OFF의 시점을 얻을 수 있으며 이 시점

을 통해 신호들 간의 표준적인 시간차이를 얻을 수 있다. Fig. 5는 PLC Signal Time Chart이다. 이것은 시스템의 제어에 관여하는 PLC 프로그램에서 사용되는 신호들의 목록과 각각의 신호들의 값이 시각에 따라 변하는 것을 나타내기 위한 그래프이다^[10]. 이 그래프를 통해서 PLC 신호들의 입출력 시와 시간의 차이는 쉽게 알 수 있다. Signal Time Chart는 PlcStudioTM에서 입출력 관계를 갖는 PLC 신호의 상태변화를 모니터링 하는데 사용되며, 신호의 값이 변하는 시점의 시간 값을 가지고 있으므로 PLC 신호 간 시간차이를 알기 위해 본 논문에서 유용하게 사용된다.

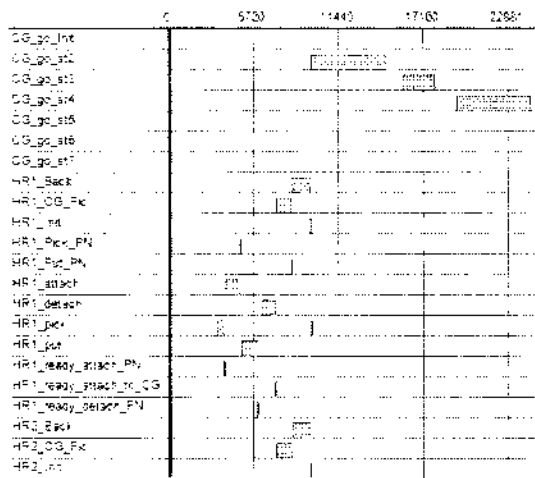


Fig. 5. A signal time chart.

3.3 양산 과정(Actual Run)의 PLC 신호 데이터

위에서 제조 시스템이 정상적으로 동작할 때의 PLC 신호의 값과 시간차이를 제조 시스템의 Try-Out Run 과정에서 얻을 수 있다. 여기서 얻은 시간차이에 대한 데이터는 실제 제조 시스템의 운영 과정에서 얻은 시간 차이와 비교를 위한 표준이라고 가정한다.

제조라인의 테스트 및 안정화를 위한 시운전 과정이 끝난 후, 제품의 양산에 들어가게 되는데 양산과정에서는 공정의 수많은 사이클이 진행되므로 외부환경으로 인한 다양한 에러가 발생할 수 있다. 본 논문에서는 외부환경에 의한 장치의 파손이나 마모와 같은 점진적인 에러를 발견하고 해결하는 것에 초점을 두고 있다. 그리고 이러한 에러를 해결하기 위해 양산라인에서의 PLC 신호들 간의 시간차이는 앞서 기술한 제조 시스템의 Test Run 과정과 동일한 방법으로 얻을 수 있으며 Try-Out Run에서 얻은 표준 시간 data와 비교함으로써 모니터링을 진행한다.

3.4 Try-Out Run과 Actual Run 데이터의 비교

앞에서 기술한 제조 시스템의 안정화를 위한 시운전 과정과 양산을 위한 실제 운영 과정에서 정상 제품을 만들어내는 제조시스템이 가지는 PLC 신호들의 표준 시간차이를 알 수 있으며, 또한 작업 실패를 유발하는 제조시스템의 PLC 신호들의 시간차를 알 수 있다. 선후관계를 갖는 PLC 신호의 시간차를 계산하여 정상 시간차와 비교함으로써 만약 시간차가 급격히 증가 또는 감소하는 것과 같은 비정상적인 상황이 발생할 경우에는 해당 장치나 해당 동작에 문제가 있을 가능성을 파악할 수 있다.

4. 에러 진단 방법의 적용과 실험

4.1 예제 공정의 설계

위에서 기술한 PLC 신호의 모니터링 방법을 PlcStudio™에서 모델링한 간단한 예제를 이용하여 설명한다. Fig. 6은 3차원 그래픽으로 모델링한 제조 공정이다. Fig. 7은 예제 공정에 사용된 Handling robot과 Welding robot의 모션이다. 각 모션은 PLC프로그램의 출력신호에 의해서 일어난다. 예제를 간단히 설명하면 가공될 제품이 station에 들어오면 모든 로봇들의 초기상태여부를 확인한다. 확인 후 Handling robot(HR1, HR2)이 동시에 라인에 들어온 제품에 부품을 파트의 좌측과 우측에 장착하고 Welding robot(WR1_1, WR1_2, WR2_1, WR2_2)이 장착된 부품에 대하여 용접을 한다. 완료 후 위와 마찬가지로 Handling robot(HR3, HR4)이 동시에 제품의 전면과 후면에 부품을 장착한 후, Welding robot(WR3_1, WR3_2, WR4_1, WR4_2)이 용접을 하고 공정이 종료된다. Fig. 8은 PLC 프로그램 작성 도구

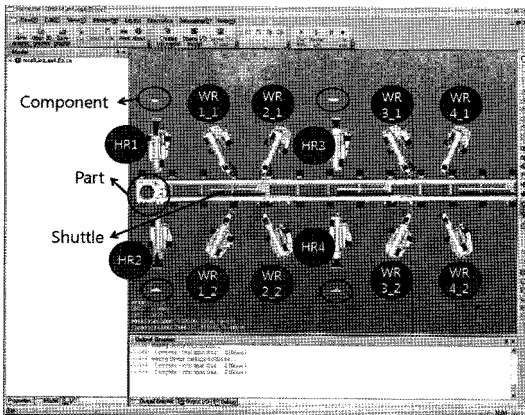
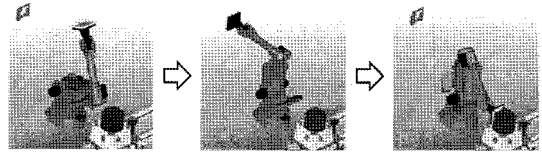
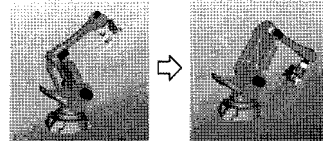


Fig. 6. The 3D automated manufacturing cell.



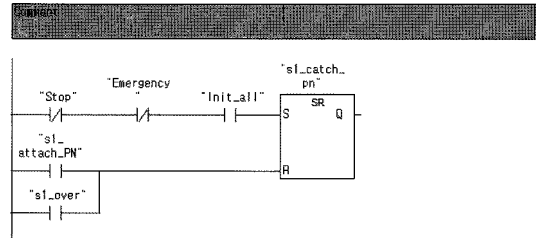
(1) Handling robot (Initial pose, Attaching Panel, Putting panel into part)



(2) Motions of welding robot (Initial pose, welding done)

Fig. 7. Motions of robots.

Network 2: catch panel in stage 1



Network 3: get pn to gun

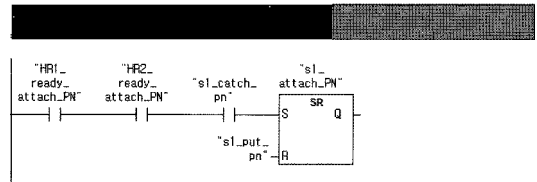


Fig. 8. The example model's PLC program.

인 Siemens의 SIMATIC Step 7™을 이용하여, 앞서 설명한 그래픽 예제를 제어할 PLC 프로그램의 일부를 보여준 것이다. Stop, Emergency 신호가 Off 되고 Init_all(로봇, 셔틀의 초기위치) 신호가 On 되면 로봇 1의 건을 부품위치로 움직이는 Output신호가 출력된다. 이후, HR1_ready_attach_PN, HR2_ready_attach_PN, sl_attach_PN 신호가 On되면 sl_attach_PN이 On 되면서 로봇1의 건에 부품을 장착하게 되는 제어 프로그램의 일부이다. 이것은 Fig. 7(1)에서 Handling robot이 Initial pose에서 Panel을 잡는 행동을 하도록 제어한다.

4.1.1 PLC 신호 간 표준 시차값

PlcStudio™의 3차원 그래픽 모듈에서 정한 애니메이션 속도를 절대 수치 30으로 주고, 이를 각 로봇들의

표준 동작 속도라고 할 때, 정상 제품을 만들어 내는, 신호관계를 갖는 PLC 신호들의 시간차이 값을 표준 시차 값이라고 가정한다. 그리고 이 공정을 Try-Out Run 공정의 결과 얻을 수 있는 표준 공정이라고 가정한다. 즉, Try-Out Run이 끝난 후 안정화된 제조 공정으로 오류가 없는 공정임을 의미한다. 예제 공정의 시뮬레이션을 수행하여 PLC 신호의 선후관제와 신호 값이 변하는 순간의 시차 값을 얻을 수 있다. 사용된 예제에서는 하나의 Cycle을 제어하기 위해서 사용된 PLC 신호들이 167번의 상태변화를 보였다. Table 2는 위 예제의 Try-Out Run 과정에서 얻은 표준 PLC 신호의 상태변화 시각과 값들의 공정의 시작으로부터 일부분을 보여준다. Signal Name은 상태변화를 보인 PLC 신호의 이름이며 Abs. Time(Absolute time)은 상태변화가 일어난 컴퓨터상의 시각(ms)이다. Val.(Value)은 신호 값의 상태변화 후의 PLC 신호 값을 의미한다.

Table 2. Signal time & value in try-out

Signal Name	Abs. Time(ms)	Val.
HR1_pick	12031	1
HR1_Init	12047	0
HR2_pick	12047	1
HR2_Init	12047	0
HR1_ready_attach_PN	12453	1
HR2_ready_attach_PN	12453	1
HR1_attach	12484	1

4.1.2 Actual Run에서 PLC 신호 간 시차값

제조 라인의 Try-Out Run이 끝나고 공정이 안정화된 후 양산을 위한 실제 운영에 들어가게 된다. 양산 과정에서는 공정에서 특징 로봇의 모션 속도가 느려지는 오류를 가정하고 이것을 예제에 적용하였다.

첫 번째 실험에서 적용한 예제는 HR 1의 특성 부품의 파손이나 파손으로 인하여 로봇이 가진 움직임 중에서 작업의 완료 후, 초기 상태로 복귀하는 부분이 느려지도록 설정하였다. 즉 해당 움직임의 속도를 표준 30에서 25, 20, 15 순차적으로 변경하여 실험을 진행하였다. 두 번째 실험에서 적용한 예제는 WR2_1과 HR3의 전체의 움직임을 15로 적용하여 실험을 진행하였다. 여기서 보트 모션의 애니메이션 속도는 로봇의 움직임의 속도를 반영한 것으로 비율의 특성을 갖는다. 예를 들면 30과 15는 2:1의 속도 비율을 갖는다. 실험에서는 속도의 관계에 따른 PLC신호의 변화 시간을 비교하여 에러를 발견하는 것으로 실제 로봇의 정확한 속도를 고려할 필요는 없다. 실험에서 기

대하는 것은 해당 로봇의 모션이 느려지는 에러로 인해 해당 움직임을 유발하는 입출력 PLC 신호 간 시간차가 길어지는 것을 발견하는 것이다. 역으로 말하면 PLC 신호 간 시간차가 길어지는 신호를 발견하면 해당 부분의 문제, 즉 자동화 장치의 작업속도가 표준에 비하여 지연되거나 또는 이른 시간에 끝나는 에러가 있는 동작을 발견할 수 있다.

4.2 Data의 비교 및 분석 결과

Try-Out Run에서 얻은 데이터와 실제 운영에서 얻은 데이터를 비교하여 에러를 명시적으로 나타내기 위해서는 아래와 같은 간단한 계산과정이 필요하다. Table 3은 비교를 위한 계산결과를 나타낸 것이다. 각 항목에 대해서 설명하면 다음과 같다.

(1) Rel. Time(Relative Time)

Table 3에서 새롭게 추가된 항목인 Reb. Time은 Table 2에서 볼 수 있듯이 각각의 운영 사이클에서 Try-Out Run의 시작 시각과 Actual Run의 시작시각이 다르기 때문에 각 운영 Cycle 의 시작 시각을 0으로 맞춰 주어야 한다. 따라서 Table 3에 이를 위하여 Rel. Time을 추가 하였다. 예를 들어 Try-Out Run에서 가장 먼저 상태변화가 일어나는 "HR1_pick" 신호의 시가 값인 12031을 Try-Out Run 사이클의 모든 신호의 시각(Abs. Time) 값에서 감산하여 얻을 수 있다(식 (1)).

$$\text{Rel. T}(i) = \text{Abs. T}(i) - \text{Abs. T}(1) \quad (1)$$

(2) Comp. Time(Compare Time)

Comp. Time 표준 공정과 실제 양산 공정의 각 사이클을 비교하기 위한 항목이다. 이것은 각각의 실제 운영 라인의 Rel. Time으로부터 표준이 되는 Try-Out Run의 Rel. Time 값을 감산하여 얻을 수 있다(식 (2)).

$$\text{Comp. T}(i) = \text{Abs. T}(i, \text{TO}) - \text{Abs. T}(i, \text{AR}) \quad (2)$$

TO = Try-Out Run, AR = Actual Run

(3) Time Gap

Time Gap은 각 실제 운영(Actual Run)에서 신호들이 변화할 때의 시간이 표준 공정과 얼마나 다른지 확인하는 것으로 이 값의 크기를 통해 에러가 있는 신호를 발견할 수 있다. 이것은 각 Comp. Time간의 차이를 절대 값으로 계산한 것이다. 즉, 실제 운영 사이클의 특정 신호의 Comp. Time을 바로 이전의 Comp. Time과 비교하여 시간차이의 폭을 계산한 것

Table 3. Comparison between try-out run data and actual run data (1)

Signal Name	Abs. Time (ms)	Val.	Rel. Time (ms)	Com. Time (ms)	Gap (절대값)
Standard Time(Try-Out Run)					
HR1_Put_PN	16844	0	4813		
HR2_Back	16844	1	4813		
IIR2_CG_Pic	16844	0	4813		
HR2_Put_PN	16844	0	4813		
HR1_CG_Pic	16859	0	4828		
IIR2_Init	18047	1	6016		
HR1_Init	18047	1	6016		
CG_go_st2	18078	1	6047		
carraige_init	18078	0	6047		
HR1_pick	18078	1	6047		
IIR1_Init	18078	0	6047		

Signal Name	Abs. Time (ms)	Val.	Rel. Time (ms)	Com. Time (ms)	Gap (절대값)
Actual Run 1 (IIR Robot 1 : pose speed 25)					
HR1_Put_PN	7640	0	4781	-32	
HR2_Back	7640	1	4781	-32	0
HR2_Put_PN	7640	0	4781	-32	0
HR1_CG_Pic	7656	0	4797	-31	1
HR2_CG_Pic	7656	0	4797	-16	15
IIR2_Init	8843	1	5984	-32	16
HR1_Init	9437	1	6578	562	594
CG_go_st2	9468	1	6609	562	0
carraige_init	9468	0	6609	562	0
HR1_pick	9468	1	6609	562	0
IIR1_Init	9468	0	6609	562	0

Actual Run 2 (IIR Robot 1 : pose speed 20)					
IIR1_Put_PN	9593	0	4796	-17	
HR2_Back	9593	1	4796	-17	0
HR2_Put_PN	9593	0	4796	-17	0
IIR2_CG_Pic	9593	0	4796	-17	0
HR1_CG_Pic	9609	0	4812	-16	1
HR2_Init	10797	1	6000	-16	0
IIR1_Init	11984	1	7187	1171	1187
CG_go st2	12031	1	7234	1187	16
carraige_init	12031	0	7234	1187	0
IIR1_pick	12031	1	7234	1187	0
IIR1_Init	12031	0	7234	1187	0

Actual Run 3 (HR Robot 1 : pose speed 15)					
IIR1_Put_PN	24750	0	4797	-16	
HR1_CG_Pic	24750	0	4797	-16	0
HR2_Back	24750	1	4797	-16	0
IIR2_CG_Pic	24750	0	4797	-16	0
HR2_Put_PN	24750	0	4797	-31	5
HR2_Init	25953	1	6000	-16	5
IIR1_Init	27953	1	8000	1984	2000
CG_go st2	27985	1	8032	1985	1
carraige_init	27985	0	8032	1985	0
IIR1_pick	27985	1	8032	1985	0
HR1_Init	27985	0	8032	1985	0

이다(식 (3)).

$$Gap(i) = |Comp. T(i) - Comp. T(i-1)| \quad (3)$$

Actual Run의 첫번째 실험의 결과에서 HR1_Init의 Time Gap 값이 이전 값들에 비해 갑자기 매우 커진 것을 볼 수 있다. 실험 1의 경우 HR1_Init 이전에 상태변화를 보인 HR2_Init2 신호의 경우 Comp. Time 값이 -32에 불과 하지만 HR1_Init의 Comp. Time 값은 이전에 비해 594의 차이를 보이고 있다. 로봇의 해당 움직임의 속도가 20으로 더욱 느려진 실험 2와 3에서는 HR1_Init의 Comp. Time 값은 각각 1187, 2000으로 이전신호의 값과 더욱 뚜렷하게 구분되고 있다. 즉 이것은 HR1_Init 신호가 1의 값으로 변하는 시간이 표준 공정에 비해 지연되었다는 것을 의미하며 장치의 움직임 입장에서 해석하면 IIR1_Init 신호가 On 되기 전까지, 그 이전 순서의 신호가 상태

변화 이후 동안의 해당 장치의 움직임이 평소와 매우 다름을 알 수 있다.

Table 4는 두 번째 실험결과에서 Time Gab이 유효한 것들을 모아 정리한 표이다. 두 번째 실험은 WR2_1과 HR3의 전체의 움직임을 15로 적용한 것이다. 모션의 속도를 느리게 적용한 WR2_1, HR3에 관련된 신호들인 WR2_1Init, IIR3_pick, HR3_ready_detach_PN, HR3_put, HR3_ready_attach_to_CG, IIR3_Init의 Time Gap이 비정상적으로 크며, 이것은 해당 device에 관련된 모션에 분체가 있음을 의미한다. 예러가 적용된 두 로봇과 관련이 없지만 Time Gap이 높은 HR4_ready_detach_PN신호(신호 값: 1)의 경우는 판넬을 로봇에서 탈착할 준비가 되었다는 신호인데, 이것은 단지 로봇의 상태를 확인하는 신호로 동작과 관계가 없는 신호이다. 이와 같이 PLC 신호 간 시간차가 Try-Out Run과 달라지는 시점을 발견함으

Table 4. Comparison between try-out run data and actual run data (2)

Signal Name	Abs. Time (ms)	Val.	Rel. Time (ms)	Signal Name	Abs. Time (ms)	Val.	Rel. Time (ms)	Com. Time (ms)	Gap (절대값)
Standard Time(Try-Out Run)				Actual Run					
All Motion Speed : 30				HR3 Robot, WR2 Robot Speed : 15					
WR2_2Init	29078	1	15922	WR2_2Init	17016	1	16812	-890	31
WR2_1Init	29078	1	15922	WR2_1Init	17204	1	17000	-1078	188
CG go st4	29172	1	16016	CG go st4	17344	1	17140	-1124	46
carraige st 4	34172	1	21016	carraige st 4	22360	1	22156	-1140	0
HR3_pick	34234	1	21078	HR3_pick	22532	1	22328	-1250	110
HR3_Init	34234	0	21078	HR3_Init	22532	0	22328	-1250	0
HR3_attach	35891	0	22735	HR3_attach	24188	0	23984	-1249	16
HR3_ready_detach_PN	37484	1	24328	HR3_ready_detach_PN	25563	1	25359	-1031	218
HR4_ready_detach_PN	37484	1	24328	HR4_ready_detach_PN	25766	1	25562	-1234	203
HR3_detach	37578	1	24422	HR3_detach	25875	1	25671	-1249	15
HR4_put	37594	0	24438	HR4_ready_detach_PN	25891	0	25687	-1249	0
HR3_put	37719	0	24563	HR3_put	25891	0	25687	-1124	125
HR3_ready_attach_to_CG	38609	1	25453	HR3_ready_attach_to_CG	26891	1	26687	-1234	110
HR4_ready_attach_to_CG	40609	1	27453	HR4_ready_attach_to_CG	28891	1	28687	-1234	0
HR4_Init	43016	1	29860	HR4_Init	31360	1	31156	-1296	16
HR3_Init	43219	1	30063	HR3_Init	32985	1	32781	-2718	1422
HR4_Back	43313	0	30157	HR4_Back	33063	1	32859	-2702	16

로써 문제발생 가능성을 발견할 수 있다.

본 논문에서는 모니터링을 위해 사용된 signal time chart는 현재 진행 중인 PLC 신호의 ON/OFF 정보와 변화 시점 정보를 실시간으로 문서 파일 형태에 저장하여 모니터링을 위한 자료를 ms 범위내에서 수집한다. 제안하는 모니터링 방법을 실제적으로 적용하기 위해서는 여러 가능성이 있는 Time Gap의 크기는 상황에 따라 강형적으로 된다면 필요가 있다. 제안하는 방법은 발생가능한 에러를 예측하는 기존의 모니터링 방법이 가진 한계점을 보완하기 위한 것으로 자동화 시스템을 직접적으로 제어하는 모든 PLC 신호를 모니터링 할 수 있다는 장점이 있다.

5. 결 론

본 논문은 자동화된 공정을 모니터링하기 위해 기존에 연구되었거나 사용되는 방법을 개선하기 위하여

PLC 신호의 시차 간격을 이용하여 자동화 공정에서 쓰이는 장치의 이상으로 인하여 발생하는 작업 실패를 미연에 방지하는 방법을 제안하였다. PLC 신호는 공정을 제어하기 위해 일반적으로 입력과 출력이라는 신호로 나누어지 있으며, 시차에 따른 선후관계를 갖는다. 제품을 만들기 위한 과정 중, Try-Out Run과정이 마무리 되고 안정화가 이루어진 과정, 즉 에러가 없다고 판단되는 공정의 사이클에서 이러한 선후관계의 시간차이를 계산하고 실제 생산과정에서의 시간차이와 비교함으로써 문제를 발견할 수 있다. 이것을 간단한 예제를 들어 설명하였고, 기존의 방법에 비해 개선된 점을 설명하였다.

본 논문은 자동화 장치의 움직임에 접근식으로 문제가 있는 경우에 초점을 맞추어 실험을 진행하였다. 하지만 이 외에도 PLC 프로그램에 내재된 비크 등 더욱 다양한 에러가 존재할 수 있으며 이것을 처리할 수 있는 방법을 개발하는 것은 현재 PLC 프로그램에

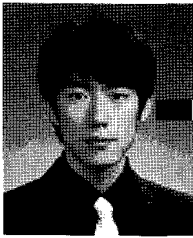
사용되는 모니터링 목적의 PLC 코드의 대체 효과를 얻을 수 있으므로 제조 라인의 생산성을 높이기 위해 유용할 것이다.

감사의 말

본 연구는 국방부 방위사업청과 국방과학연구소의 지원으로 수행되었습니다(UD080042AD). 관계자 여러분께 감사드립니다.

참고문헌

1. Jack, H., "Automating Manufacturing Systems with PLCs", version 4.7, 2005.
2. 박원일, "비호환 기종 PLC들의 통합제어를 위한 PLC에너지 설계", 한양대학교 석사학위 논문, 1993.
3. Arbor, A., "Operational Diagnostics, The Holy Grail of Control Automation, A White Paper" Entivity, Inc., 2008.
4. 홍상현, "표준화된 PLC Ladder 프로그램과 HMI 장비를 이용한 자체 조립라인의 운용에 관한 연구", 울산대학교 석사학위 논문, 2002.
5. Holloway, L. E. and Chand, S., "Time Templates for Discrete Event Fault Monitoring in Manufacturing Systems", *American Control Conference*, pp. 701-706, 1994.
6. Zhang, H., "On-line PLC Monitoring and Network Administering System for Steel Tube Mill", *Proceeding of the IEEE International Conference on Industrial Technology*, pp. 720-723, 1996.
7. Ioannides, M. G., "Design and Implementation of PLC-Based Monitoring Control System for Induction Motor", *IEEE Transactions on energy conversion*, Vol. 19, No. 3, pp. 469-476, 2004.
8. UDMTEK, "PLC Studio 제품 소개", <http://www.udmtek.com>, 2008.
9. Park, C. M., "Development of Virtual Simulator for Visual Validation of PLC Program", *International Conference on Computational Intelligence for Modeling Control and Automation*, pp. 32-36, 2006.
10. Park, C. M., "Simulation based Control Program Verification in an Automobile Industry". *The 27th ISATED International Conference on Modelling, Identification, and Control, Austria*, 2008.
11. SIEMENS, "SIMATIC Working with STEP 7 V5.3" http://www.fer.hr/_download/repository/S7gsV53_e.pdf, 2004.
12. 한관희, 박준우, "UML 기반 PLC 래더 로직 설계와 코드 자동 생성", 한국CAD/CAM학회 논문집, Vol. 14, No. 6, pp. 50-59, 2009.



성길영

2001년~2006년 아주대학교 산업정보시스템공학부 학사
 2006년~현재 아주대학교 산업공학과 석사과정
 연구분야: Simulation, 이산사건모델링, PLC



한관희

1982년 아주대학교 산업공학과 학사
 1984년 한국과학기술원 산업공학과 석사
 1996년 한국과학기술원 자동화 및 실체공학과 박사
 1984년~1992년 대우전자(주) MIS실 (생산관리팀장)
 1996년~1998년 대우정보시스템(주) CIM사업부(사업부장)
 1998년~2000년 대우정보시스템(주) 기술연구소(연구부장)
 2000년~현재 경상대학교 산업시스템공학부 조교수
 관심분야: PDM, Workflow, CSCW, sw개발관리



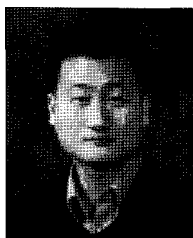
변재정

1978년 중부대학교 통계학과 학사
 1984년 숭실대학교 전산학과 석사
 1996년 미국 일리노이공과대학교(IIT) 전산학과 박사
 1984년~1991년 한국수방연구원 선임연구원
 1992년~1998년 국방정보체계연구소 연구위원
 1999년~현재 국방과학연구소 책임연구원
 관심분야: Simulation Based Acquisition (SBA), Wargame Model, Synthetic Environment Simulation, Real-Time Operating System(RTOS)



왕지남

1979년 3월~1983년 2월 아주대학교 공과대학 산업공학과
 1983년 3월~1985년 2월 한국과학기술원 산업공학과 석사
 1984년 12월~1987년 8월 현대전자(주) 연구원(Dept. of Production Informaion Control Systems)
 1987년 3월~1992년 12월 미 Texas A&M 대학 산업공학과
 1992년 12월~1993년 5월 미 Texas A&M University 연구원
 2000년 9월~2004년 8월 University of Texas at Austin Visiting Professor
 1993년 9월~현재 아주대학교 산업정보시스템공학부 교수
 연구분야: 자동감시 제어, 신경망, PLC, 시뮬레이션, 이산사건 모델링



박상철

B.S. (1994) in Industrial Engineering, Dept of I.E., KAIST, Korea
 M.S. (1996) in Industrial Engineering, Dept of I.E., KAIST, Korea
 Ph.D. (2000) in Industrial Engineering, Dept of I.E., KAIST, Korea
 2000년 9월~2001년 12월, 큐빅테크, 선임연구원
 2002년 1월~2004년 2월, DaimlerChrysler ITM Dept. Research Engineer
 2008년 2월~현재 아주대학교 산업정보시스템공학부, 부교수
 2005년 1월~현재 Computer-Aided Design and Applications
 2006년 9월~현재 E.I.T.(Entic Journal of Information Technology)
 2007년 2월~2009년 1월 31일 한국 CAD/CAM 학회 편집위원
 연구분야: CAD/CAM, 시뮬레이션, PLC, 이산사건 모델링