

PLC 래더다이아그램의 체계적인 분석을 위한 이산사건모델 변환 알고리즘

Discrete Event Model Conversion Algorithm for Systematic Analysis of Ladder Diagrams in PLCs

강 봉 석, 조 광 현

(Bong-Suk Kang and Kwang-Hyun Cho)

Abstract : As product lifecycles become shorter, factories are pushed to develop small batches of many different products. The highly flexible control systems has become a necessity. The majority of existing automated industrial systems are controlled by programmable logic controllers(PLCs). In most cases, the control programs for PLCs are developed based on ladder diagrams(LDs). However, it is difficult to debug and maintain those LDs because the synthesis of LD itself mainly depends on the experience of the industrial engineer via trial-and-error methods. Hence, in this paper, we propose a discrete event model conversion algorithm for systematic analysis of LDs. The proposed discrete event model conversion algorithm is illustrated by an example of a conveyor system.

Keywords : PLC, ladder diagram, discrete event model, state diagram

I. 서론

최근 PLC(Programmable Logic Controller)는 사용자의 프로그래밍에 의해 운전되기 때문에 릴레이 로직에 비해 유연성, 보수성, 신뢰성이 월등히 높아 대다수의 공정 및 생산시스템의 제어장치로서 그 중요성이 더욱 높아지고 수요도 계속 확대되고 있다. 이러한 PLC의 제어 언어로는 LD(Ladder Diagram), IL(Instruction List), SFC(Sequential Function Chart), FBD(Function Block Diagram), ST(Structured Text) 등이 주로 사용되고 있다[1]-[3]. 이 가운데 릴레이 논리에 기반한 그래픽 제어언어인 래더다이아그램(LD)이 보편적으로 사용되고 있다. 그러나 LD는 숙련된 프로그래머의 축적된 현장 경험으로 제안된 알고리즘에 크게 의존하여 작성되기 때문에 LD의 수정 및 변형을 위한 유연성 등이 결여되어 있다[4]. 또한, LD는 현장의 작업상황을 순차적으로 작성한 심볼들의 집합으로서 단지 순차적 표현으로는 적합하지만 시스템의 모델링이나 분석 측면에서는 한계를 지니고 있다. 따라서 본 논문에서는 대상 시스템을 직관적으로 파악하며 체계적으로 해석하고 분석하기 위하여 LD를 이산사건모델로 변환하는 알고리즘을 제안하고자 한다. 이산사건모델의 논리적인 동작을 다루는 모델 중 가장 대표적인 방법 중 하나는 상태선도(SD: State Diagram)에 기반한 방법이다. 상태선도는 시스템의 상태가 변화하는 순서, 각 상태들의 조건 그리고 상태천이의 조건 등을 도식화한 것이다. 상태선도에서 각 상태는 원으로 표시되고 상태사이의 천이는 원을 연결하는 화살표로 표시된다. 이러한 상태선도는 직관적이고 체계적으로 시스템을 파악하고 분석할 수 있을 뿐 아니라 모델링 측면에

서도 우수한 성능을 지니고 있다. 따라서 LD를 이산사건모델로 변환함으로써 체계적으로 LD를 해석하고 분석하여 수정 및 변형을 용이하게 할 수 있다. 특히, [5]에서 시스템의 동작사양을 이용하여 이산사건모델로 모델링한 후 이를 LD로 자동합성하는 알고리즘이 제안되었다. 이 알고리즘은 시스템의 동작사양을 이산사건모델로 모델링함으로써 시스템의 동작사양 추가 및 변경을 용이하게 할 수 있다. 따라서 기존의 작성된 LD를 이산사건모델로 변환하여 시스템의 변경 및 추가 후 [5]의 알고리즘을 이용하여 다시 LD로 변환함으로써 체계적인 방법에 의하여 LD를 생성할 수 있다. 또한, LD를 이산사건모델로 변환하여 이산사건모델에 기반한 고장진단 알고리즘을 적용함으로써 PLC의 고장진단을 보다 조직적이고 체계적인 방법으로 접근할 수 있다[6]-[10]. 단 본 논문에서 제안하는 알고리즘은 PLC의 기본기능인 내부 출력과 외부출력에 대한 이산사건모델로의 변환을 목적으로 한다.

본 논문의 구성은 다음과 같다. 먼저 2절에서는 LD로부터 상태선도로 변환하기 위한 사건(event) 및 상태(state)를 정의하고 이를 이용하여 상태선도로 변환하는 알고리즘을 기술한다. 3절에서는 컨베이어 시스템의 동작사양을 만족시키는 LD를 제안한 알고리즘에 적용하여 상태선도로 변환하는 과정을 기술한다. 마지막으로 4절에서는 결론을 맺고 향후 연구과제를 논의한다.

II. 이산사건모델 변환 알고리즘

본 절에서는 LD를 SD로 변환하는 알고리즘을 소개한다. 그림 1은 LD를 SD로 변환하는 과정을 개략적으로 표현한 것이다. 그림 1에서 보는 바와 같이 이산사건모델 변환 알고리즘은 3개의 주요 단계로 구성된다. 먼저 LD를 이용하여

접수일자 : 2001. 9. 17., 수정완료 : 2002. 1. 24.

강봉석, 조광현 : 울산대학교 전기전자정보시스템공학부

(kbs@sys.ulsan.ac.kr/ckh@mail.ulsan.ac.kr)

※ 이 논문은 2001년도 한국학술진흥재단의 지원에 의하여 연구가 되었음.(KRF-2001-041-E00279)

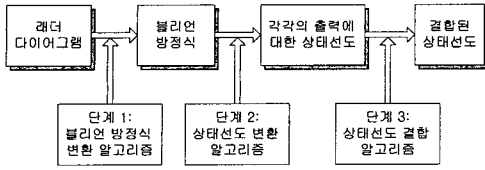


그림 1. 이산사건모델 변환 알고리즘의 블록 다이어그램.
Fig. 1. Block diagram of the proposed discrete event model conversion algorithm.

불리언 방정식을 유도하고 이를 이용하여 각각의 불리언 방정식에 의한 출력을 구한 후 SD로 변환하며 마지막으로 개별적으로 구성된 SD를 결합하는 과정으로 이루어져 있다.

1. 불리언 방정식 변환 알고리즘

본 논문에서는 LD를 SD로 변환하는데 있어서 PLC의 동작특성을 고려하여 사건 및 상태를 다음과 같이 정의한다:

- Y_i : PLC의 내부출력(Mi) 또는 외부출력(Pi),
- $Y_{i,s}$: 출력상태(output state) Y_i 를 동작시키기 위한 셋(set) 상태,
- $Y_{i,r}$: 출력상태 Y_i 를 정지시키기 위한 리셋(reset) 상태,
- P_j : 푸시버튼, 리미트스위치 등의 j번째 입력접점,
- $Y_{j,s}$: j번째 셋(set) 입력상태(input state)접점,
- $Y_{j,r}$: j번째 리셋(reset) 입력상태(input state)접점,
- ζ : 출력상태의 자기유지를 위한 사건, 입력상태접점과 출력상태가 같을 경우 ζ 에 의하여 케환루프 형성,
- ϵ : 시스템의 내부적인 변화를 나타내는 사건, ϵ 은 입력사건에 대하여 우선순위를 갖는다.

여기서, 첨자 s는 셋 상태를 나타내고, 첨자 r은 리셋 상태를 나타낸다. 또한, j는 입력접점을 나타내고 i는 출력상태를 나타낸다.

위에서 정의한 상태 및 사건을 이용하여 불리언 방정식으로 유도하는 과정은 다음과 같다:

- 단계 1: LD로부터 불리언 방정식의 변환.
- 단계 1-1: 출력상태 Y_i 를 선택한다.
- 단계 1-2: 라인의 교점을 찾고 이를 n_i 라 한다. 단, 교점이 존재하지 않을 때 단계 1-5로 이동한다.
- 단계 1-3: 교점을 기준으로 왼쪽라인의 입력은 Y_i 를 동작시키기 위한 $Y_{i,s}$ 로의 입력상태 및 사건으로 정의되고, 오른쪽라인에서 입력의 보수(complement)는 Y_i 를 정지시키기 위한 $Y_{i,r}$ 로의 입력상태 및 사건으로 정의된다.
- 단계 1-4: 입력상태접점 중에서 상시개로접점 Y_j 는 $Y_{j,s}$ 로 치환하고, 상시폐로접점 Y_j 는 $Y_{j,r}$ 로 치환한다. 또한, 입력접점 중에서 상시개로접점 P_j 는 그대로 유지하고, 상시폐로접점 P_j 는 \bar{P}_j 로 치환한다.

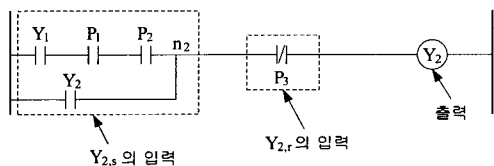


그림 2. 불리언 방정식 변환 알고리즘의 예.
Fig. 2. Example of the Boolean equation conversion algorithm.

단계 1-5 : 교점이 존재하지 않을 경우 Y_i 의 모든 입력이 $Y_{i,s}$ 로의 입력상태 및 사건으로 정의되고, 또한, Y_i 의 입력사건의 보수가 $Y_{i,r}$ 로의 입력사건으로 정의된다.

단계 1-6 : 출력의 동작을 위한 $Y_{i,s}$ 로의 입력 및 출력의 정지를 위한 $Y_{i,r}$ 로의 입력에 대하여 동일 라인의 입력은 AND, 서로 다른 라인의 입력은 OR 연산을 수행한다.

그림 2는 LD가 주어졌을 때 불리언 방정식을 유도하기 위한 입력 및 출력의 관계를 나타낸 것이다. 그림 2와 같이 LD가 주어졌을 때 n_2 를 기준으로 왼쪽라인은 $Y_{2,s}$ 의 입력이 되어 $Y_{2,s} = Y_{1,s} \cdot P_1 \cdot P_2 + Y_{2,s}$ 가 되고, 오른쪽 라인은 $Y_{2,r}$ 의 입력이 되어 $Y_{2,r} = \bar{P}_3 = P_3$ 가 된다.

이상은 LD를 이용하여 SD로 변환하기 위한 상태 및 사건을 정의하고 이를 불리언 방정식으로 나타내는 과정을 설명하였다. 위의 알고리즘을 이용하면 LD에서의 출력과 입력은 SD에서의 출력상태에 대한 입력상태 및 사건으로 분리되고 이를 이용하여 불리언 방정식으로 변환하게 된다.

2. 상태선도 변환 알고리즘

위에서 유도된 불리언 방정식을 이용하여 SD로 변환하기 위한 알고리즘은 다음과 같다. 첫번째 단계는 출력상태의 동작을 위한 셋(set) 상태 $Y_{i,s}$ 로의 입력을 이용하여 SD로 변환하는 과정이고, 두 번째 단계는 출력상태의 정지를 위한 리셋(reset) 상태 $Y_{i,r}$ 로의 입력을 이용하여 SD로 변환하는 과정을 나타낸다. 첫번째 단계는 출력상태 $Y_{i,s}$ 의 입력이 입력상태접점 $Y_{j,s}$ 가 존재할 경우 만약 출력상태와 입력상태접점이 같다면 $Y_{i,s}$ 의 자기케환루프를 생성하고 ζ 에 의하여 자기사건으로 천이되고, 출력상태와 입력상태접점이 같지 않다면 입력사건에 의하여 입력상태접점인 $Y_{j,s}$ 에서 $Y_{i,s}$ 로 천이된다. 그 외의 경우는 입력사건에 의하여 $Y_{i,r}$ 에서 $Y_{i,s}$ 로 천이된다. 두 번째 단계는 출력상태 $Y_{i,r}$ 의 입력이 입력상태접점 Y_j 가 존재할 경우 입력사건에 의하여 입력상태접점인 Y_j 에서 $Y_{i,r}$ 로 천이되고, 그 외의 경우는 입력사건에 의하여 $Y_{i,s}$ 에서 $Y_{i,r}$ 로 천이된다. 단, 입력사건이 없을 경우 ϵ 에 의하여 천이가 발생된다. 이상의 과정을 수도코드(pseudo code)로 나타내면 다음과 같다. 여기서, $E(Y_j, Y_i)$ 는 상태 Y_j 에서 상태 Y_i 로의 천이 사건을 나타낸다.

단계 2 : 불리언 방정식으로부터 상태선도로의 변환

```

begin step 2
for(i=1;i=N;i++)
if(output state=  $Y_{i,s}$ )
for(j=1;j=N;j++)
if(input state=  $Y_{j,s}$  and  $i \neq j$ )
E( $Y_{j,s}, Y_{i,s}$ )= $P_j$ 
else if(input state=  $Y_{j,s}$  and  $i = j$ )
E( $Y_{j,s}, Y_{i,s}$ )= $\zeta$ 
else
E( $Y_{i,r}, Y_{i,s}$ )= $P_j$ 
endif
end
end
else if(output state=  $Y_{i,r}$ )
for(j=1;j=N;j++)
if(input state=  $Y_j$ )
    
```

```

E(Yj, Yi,r)=Pj
else
E(Yi,s, Yi,r)=Pj
endif
end
endif
end
    
```

위의 수도코드는 PLC를 제어하기 위한 동작사양을 이용하여 작성된 LD로부터 유도된 블리언 방정식을 SD로 변환하는 과정을 나타낸 것이다. 이와 같은 과정을 거치면 LD 각각의 출력으로부터 SD가 생성된다.

3. 상태선도 결합 알고리즘

다음 단계 3은 개별적으로 구해진 SD를 서로 결합하는 과정을 나타낸 것이다. 단계 3에서는 PLC의 동작사양에 맞게 각 상태 및 사건 간의 우선순위를 부여하여 결합한다. 이러한 단계 3을 정리하면 다음과 같다:

- 단계3:** 각각의 출력에 대한 상태선도의 결합
 각 상태와 사건의 결합시 상호간의 우선순위는 다음과 같다.
- 1) ε 이 외부입력사건 P_j보다 우선순위가 높다.
 - 2) 셋(set) 상태가 리셋(reset) 상태보다 우선순위가 높다.
 - 3) 내부출력 M이 외부출력 P보다 우선순위가 높다.

그림 3은 앞에서 기술한 이산사건모델 변환 알고리즘의 각 단계를 플로우차트로 표현한 것이다. 그림 3과 같이 이산사건 모델 변환 알고리즘은 LD를 이용하여 블리언 방정식

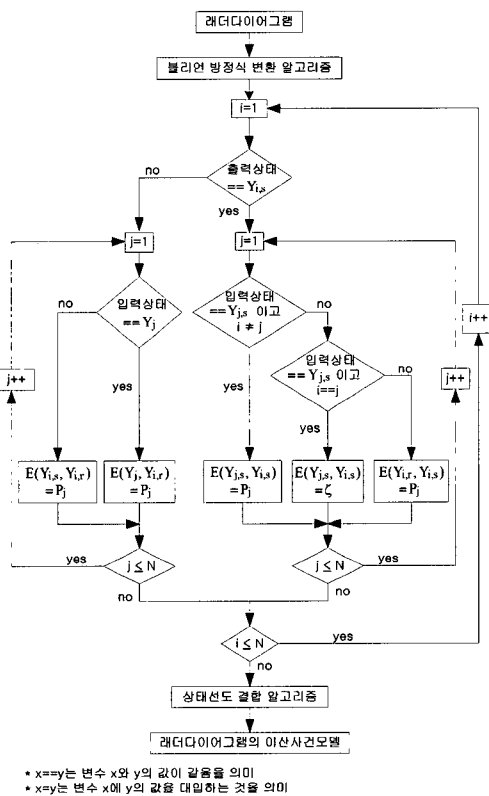


그림 3. 이산사건모델 변환과정의 플로우차트.
 Fig. 3. Flowchart of the discrete event model conversion procedures.

으로 변환하고 이를 상태선도 변환 알고리즘을 이용하여 각각의 출력에 대한 SD를 구한 후 상태선도 결합 알고리즘을 이용하여 최종적으로 SD로 변환하게 된다.

III. 적용사례

본 절에서는 위에서 제시한 이산사건모델 변환 알고리즘을 컨베이어 시스템에 적용하여 각 단계를 예제를 통해 기술한다.

그림 4는 제1컨베이어로부터 이송되어 온 부품상자를 높이가 다른 제2컨베이어로 옮겨주는 컨베이어간 이송장치의 구성도이다. 이송장치의 동작순서는 부품상자가 제1컨베이어에 의해 이송되어 상승위치까지 도착되면 P₇ 검출 스위치가 온(on)이 되고 실린더 A가 상승하여 부품상자를 실린더 B 앞까지 들어올린다. A실린더가 부품상자를 상승완료시켜 P₄신호가 온(on)이 되면 실린더 B가 전진하여 부품상자를 제2컨베이어로 밀어 이송한다. 그 다음에 A실린더는 다시 하강하고 하강완료 후 P₃ 리미트 스위치를 온(on) 시키면 실린더 B가 복귀하여 비로소 1사이클이 종료된다[11].

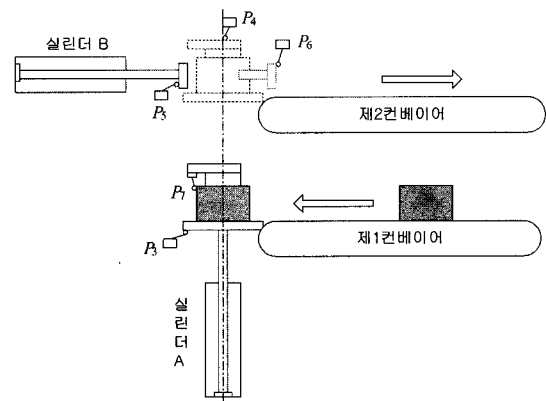


그림 4. 컨베이어 시스템의 구성도.
 Fig. 4. Structure of the example of the conveyor system.

표 1. 컨베이어 시스템의 변수 정의.
 Table 1. Definition of the variables for the conveyor system.

| 구분 | PLC I/O | 입출력에 대한 설명 |
|------|-----------------|--------------------|
| 입력 | P ₁ | 시동 스위치 |
| | P ₂ | 정지 스위치 |
| | P ₃ | 실린더 A의 후진검출 리미트스위치 |
| | P ₄ | 실린더 A의 전진검출 리미트스위치 |
| | P ₅ | 실린더 B의 후진검출 리미트스위치 |
| | P ₆ | 실린더 B의 전진검출 리미트스위치 |
| 내부출력 | P ₇ | 제1컨베이어의 상자 검출 스위치 |
| | M ₅ | 실린더 B 후진 내부출력 |
| 외부출력 | P ₅₀ | 컨베이어 시동 외부출력 |
| | P ₅₁ | 실린더 A 상승 외부출력 |
| | P ₅₂ | 실린더 B 전진 외부출력 |
| | P ₅₃ | 실린더 A 하강 외부출력 |

그림 5는 컨베이어 시스템의 동작사양을 참조하여 작성된 LD이다. 그림 5를 이산사건모델 변환 알고리즘을 이용하여 주어진 LD를 이산사건모델로 변환하는 과정을 설명하면 다음과 같다.

첫번째, 그림 5의 LD로부터 시동 외부출력 P₅₀의 셋 상태 P_{50,s}의 불리언 방정식은 P_{50,s}=P₁+P_{50,s}가 된다. 따라서 단계 2에 의해 E(P_{50,r}, P_{50,s})=P₁이 되고 E(P_{50,s}, P_{50,s})=ζ가 된다. 이를 SD로 변환하면 그림 6(a)와 같은 상태전도를 얻을 수 있다. 또한, P₅₀의 리셋 상태 P_{50,r}의 불리언 방정식은 P_{50,r}= $\overline{P_2}$ =P₂가 되고 이를 단계 2에 따라 전개하면 E(P_{50,s}, P_{50,r})=P₂가 되므로 SD로 변환하면 그림 6(b)와 같이 된다.

두번째, 실린더 A 상승 외부출력 P₅₁의 셋 상태 P_{51,s}는 P_{51,s}=P_{50,s} · P₃ · P₅ · P₇+P_{51,s}이므로 E(P_{50,s}, P_{51,s})=P₃ · P₅ · P₇이 되고 E(P_{51,s}, P_{51,s})=ζ가 된다. 따라서 P_{51,s}에 대한 SD는 그림 6(c)와 같이 된다. 또한, P₅₁의 리셋 상태 P_{51,r}의 불리언 방정식은 P_{51,r}= $\overline{P_{33,r}}$ =P_{53,s}이므로 E(P_{53,s}, P_{51,r})=ε이 되고 이를 SD로 변환하면 그림 6(d)가 된다. 이와 같은 방법으로 P₅₂와 P₅₃도 구할 수 있다.

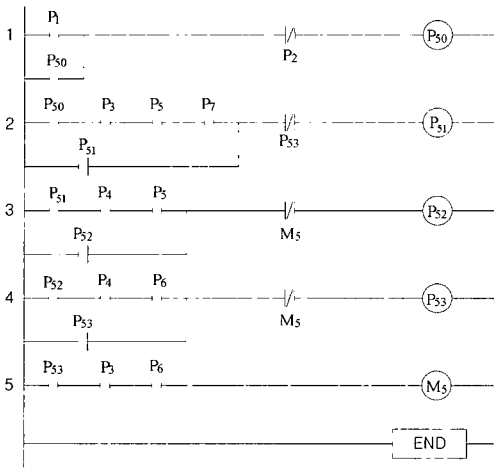


그림 5. 컨베이어 시스템의 래더 다이어그램.
Fig. 5. Ladder diagram of the conveyor system.

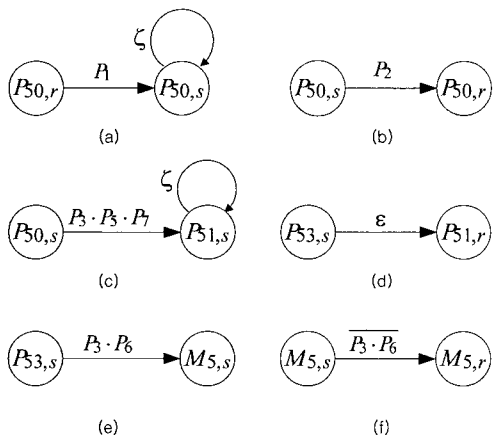


그림 6. 컨베이어 시스템의 상태전도.
Fig. 6. State diagram of the conveyor system.

또한, 실린더 B 후진 내부출력 M₅의 셋 상태 M_{5,s}는 M_{5,s}=P_{53,s} · P₃ · P₆이 되므로 단계 2에 의하여 E(P_{53,s}, M_{5,s})=P₃ · P₆이 되어 SD로 변환하면 그림 6(e)와 같은 상태전도가 된다. 또한, M₅의 리셋 상태 M_{5,r}는 M_{5,r}= $\overline{P_3 \cdot P_6}$ 가 되어 E(M_{5,s}, M_{5,r})= $\overline{P_3 \cdot P_6}$ 이 되므로 M_{5,r}의 SD는 그림 6(f)가 된다.

그림 6과 같이 LD의 각각의 출력에 대하여 SD를 구한 후 이를 단계 3에 따라 우선순위를 참조하여 결합하면 그림 7과 같은 결과를 얻을 수 있다. 예를 들면 그림 6(e)는 P₃ · P₆에 의해 P_{53,s}에서 M_{5,s}로 천이되고 그림 6(d)는 ε에 의하여 P_{53,s}에서 P_{51,r}로 천이된다. 여기서, 입력사건보다 ε이 우선순위가 높기 때문에 그림 7과 같이 ε에 의하여 P_{53,s}에서 P_{51,r}로 천이한 후 P₃ · P₆에 의하여 P_{51,r}에서 M_{5,s}로 천이함을 알 수 있다.

그림 7의 SD를 [5]에서 제안된 이산사건모델로부터 LD로 변환하는 알고리즘을 적용하여 LD로 변환시키면 그림 8과 같은 결과를 얻을 수 있다. [5]에 따르면 이산사건모델을 LD로 변환하는 알고리즘은 시스템의 동작사양을 만족시키는 이산사건모델을 불리언 방정식으로 변환한 후 LD로 변환하게 된다. 본 예제에서는 PLC의 자기유지 기능을 고려하여 출력상태의 자기유지를 위한 사건 ζ를 추가하여 입력상태 점점과 출력상태가 같을 경우 ζ에 의하여 궤환루프가 형성된다고 가정한다. 추가된 기능과 [5]에서 제안된 알고리즘을 기반으로 하여 그림 7의 SD를 LD로 변환시키는 과정을 설명하면 다음과 같다. 여기서, 상태전도로부터 불리언 방정식을 유도하기 위한 상태(Q)와 사건(E)에 대하여 다음과 같이 정의한다 : α₁, α₂ ..., α_{i-1}, α_i, α_{i+1}와 같은 상태들(α_i≡Q)이 사건의 천이에 의하여 순서대로 정의되어 있다고 가정하면 d(α_i, α_{i+1})=α_i이다. 여기서, α_i는 입력상태이고 α_{i+1}은 출력상태를 나타낸다. 그리고 E=e(α_i, α_{i+1})는 입력상태 α_i에서 출력상태 α_{i+1}로 천이하는 사건을 나타낸다. 또한, g : Q → {0, 1}과 f : Q → {0, 1}는 다음과 같이 정의된다: f(α_i)는 δ(α_j, E)=α_i일 때 '1'이고, 그 외의 경우는 '0'이다. 여기서, δ는 천이함수로서 상태 α_j로부터 사건 E에 의하여 상태 α_i로의 천이를 나타낸다. g(α_i)는 α_i∈Y_{j,s}이고 α_i∈Y_{i,s}이거나, i ≠ j인 경우 α_i∈Y_{j,s}이면서 α_i∈Y_{i,r}이면 g(α_i)=f(α_i)이고 그 외의 경우는 1이다. 이때 첨자 j는 입력상태(단, 1 ≤ j ≤ N), 첨자 i는 출력상태(단, 1 ≤ i ≤ N), 첨자 s는 셋 상태, 첨자 r은 리셋 상태를 나타낸다. 또한, 변수 N은 상태의 수, Y는 내부 출력 또는 외부출력 상태를 각각 나타낸다.

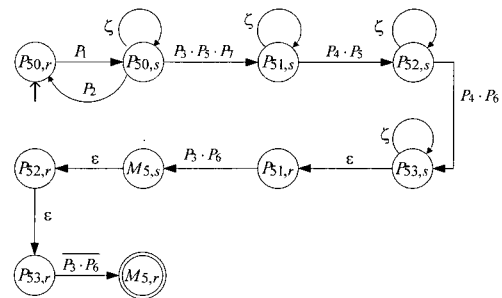


그림 7. 컨베이어 시스템의 결합된 상태전도.
Fig. 7. Combined state diagram of the conveyor system.

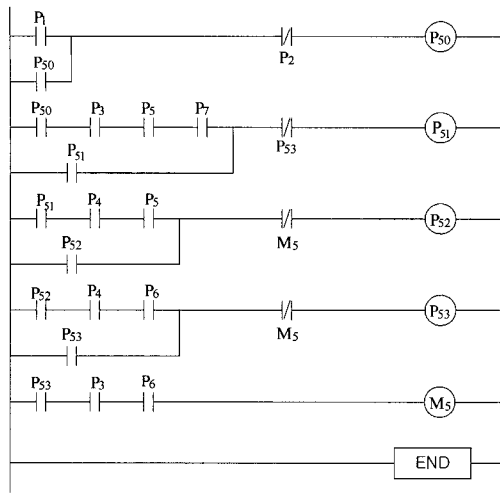


그림 8. 이산사건모델에 기반한 컨베이어 시스템의 래더다이아그램.

Fig. 8. Ladder diagram of the conveyor system based on discrete event model.

첫번째, 그림 7의 상태선도로부터 시동 외부출력 P50의 셋 상태는 $g(d(P_{50,r}, P_{50,s}))=1$, $g(d(P_{50,s}, P_{50,s}))=f(P_{50,s})$ 이므로 $P_{50,s}=P_1 \cdot g(P_{50,r})+g(P_{50,s})=P_1+f(P_{50,s})$ 이 되고 P50의 리셋 상태는 $g(d(P_{50,s}, P_{50,r}))=1$ 이므로 $P_{50,r}=P_2 \cdot g(P_{50,s})=P_2$ 이 되므로 $P_{50} = P_{50,s} \cdot \bar{P}_{50,r} = (P_1+f(P_{50,s})) \cdot \bar{P}_2$ 로 표현된다.

두번째, 실린더 A 상승 외부출력 P51의 셋 상태 P51,s는 $g(d(P_{50,s}, P_{51,s}))=f(P_{50,s})$, $g(d(P_{51,s}, P_{51,s}))=f(P_{51,s})$ 이므로 $P_{51,s}=P_3 \cdot P_5 \cdot P_7 \cdot g(P_{50,s})+g(P_{51,s})=P_3 \cdot P_5 \cdot P_7 \cdot f(P_{50,s})+f(P_{51,s})$ 이 되고, P51의 리셋 상태 P51,r은 $g(d(P_{53,s}, P_{51,r}))=f(P_{53,s})$ 이므로 $P_{51,r}=g(P_{53,s})=f(P_{53,s})$ 이 되어 $P_{51} = P_{51,s} \cdot \bar{P}_{51,r} = (P_3 \cdot P_5 \cdot P_7 \cdot f(P_{50,s})+f(P_{51,s})) \cdot \bar{f}(P_{53,s}) = (P_3 \cdot P_5 \cdot P_7 \cdot f(P_{50,s})+f(P_{51,s})) \cdot \bar{f}(P_{53,r})$ 이 된다.

세번째, 실린더 B 전진 외부출력 P52의 셋 상태 P52,s는 $g(d(P_{51,s}, P_{52,s}))=f(P_{51,s})$, $g(d(P_{52,s}, P_{52,s}))=f(P_{52,s})$ 이므로 $P_{52,s}=P_4 \cdot P_5 \cdot f(P_{51,s})+f(P_{52,s})$ 이 되고, P52의 리셋 상태 P52,r은 $g(d(M_{5,r}, P_{52,r}))=f(M_{5,s})$ 이므로 $P_{52,r}=g(M_{5,s})=f(M_{5,s})$ 이 되어 $P_{52} = P_{52,s} \cdot \bar{P}_{52,r} = (P_4 \cdot P_5 \cdot f(P_{51,s})+f(P_{52,s})) \cdot \bar{f}(M_{5,s}) = (P_4 \cdot P_5 \cdot f(P_{51,s})+f(P_{52,s})) \cdot \bar{f}(M_{5,r})$ 이 된다.

네번째, 실린더 A 하강 외부출력 P53의 셋 상태 P53,s는 $g(d(P_{52,s}, P_{53,s}))=f(P_{52,s})$, $g(d(P_{53,s}, P_{53,s}))=f(P_{53,s})$ 이므로 $P_{53,s}=P_4 \cdot P_6 \cdot f(P_{52,s})+f(P_{53,s})$ 이 되고, $E(M_{5,s}, P_{52,r})=E(P_{52,r}, P_{53,r})=\epsilon$ 이므로 P53의 리셋 상태는 $g(d(M_{5,s}, P_{53,r}))=f(M_{5,s})$ 이고 $P_{52,r}=g(M_{5,s})=f(M_{5,s})$ 이 되어 $P_{53} = P_{53,s} \cdot \bar{P}_{53,r} = (P_4 \cdot P_6 \cdot f(P_{52,s})+f(P_{53,s})) \cdot \bar{f}(M_{5,s}) = (P_4 \cdot P_6 \cdot f(P_{52,s})+f(P_{53,s})) \cdot \bar{f}(M_{5,r})$ 이 된다.

마지막으로 실린더 B 후진 내부출력 M5의 셋 상태 M5,s는 $g(d(P_{53,s}, M_{5,s}))=f(P_{53,s})$ 이므로 $M_{5,s}=P_3 \cdot P_6 \cdot f(P_{53,s})$ 이 되고, M5의 리셋 상태 M5,r은 $g(d(M_{5,s}, M_{5,r}))=1$ 이므로 $M_{5,r} = \bar{P}_3 \cdot \bar{P}_6 \cdot \bar{f}(M_{5,s}) = \bar{P}_3 \cdot \bar{P}_6$ 가 되어 $M_5 = M_{5,s} \cdot \bar{M}_{5,r} = (P_3 \cdot P_6 \cdot f(P_{53,s})) \cdot \bar{P}_3 \cdot \bar{P}_6 = P_3 \cdot P_6 \cdot f(P_{53,s})$ 이 된다.

이와 같은 과정을 통하여 유도된 블리언 방정식으로부터 논리곱은 LD에서 직렬로 연결하고 논리합은 병렬로 연결하며, 논리 '1'은 상시개로접점인 a접점으로 변환하고 논리 '0'

은 상시폐로접점인 b접점으로 변환한 후 각각의 출력상태를 LD에서의 내부출력 또는 외부출력으로 변환하면 그림 8과 같은 LD를 얻을 수 있다.

그림 8의 결과는 그림 5의 결과와 일치함을 알 수 있다. 이와 같은 결과는 본 논문에서 제안한 알고리즘과 [5]의 알고리즘이 서로 상호변환이 가능함을 나타내며, 그림 7의 이산사건모델이 그림 5에 주어진 LD에 대해 논리적으로 정확한 모델링의 결과임을 보여준다.

IV. 결론

생산시스템이 점점 더 복잡해짐으로써 PLC의 사용 또한 확대되어 그에 따른 LD가 복잡해지고 프로그램의 용량도 증가되고 있다. 이로 인해 PLC의 제어 알고리즘인 LD를 직관적이고 체계적으로 분석할 수 있는 방법의 필요성이 대두되고 있다. 본 논문에서는 LD를 이산사건모델로 변환하는 알고리즘을 제안하였다. 먼저 LD로부터 블리언 방정식을 유도한 후 이를 이산사건모델로 변환하는 과정을 제안하였다. 또한, 제안한 알고리즘을 컨베이어 시스템에 적용하여 이산사건모델로 변환하는 과정을 설명하였다. 향후 연구과제로는 타이머, 카운터 등 고급기능의 명령어들을 포함시킬 수 있도록 제안한 알고리즘을 확장하는 연구 등이 있다.

참고문헌

- [1] E. A. Parr, *Programmable Controllers : An engineers guide*, 3rd ed., Oxford, U.K. : Newnes, 1999.
- [2] A. Falcione and B. H. Krogh, "Design recovery for relay ladder logic," *IEEE Control Systems Magazine*, vol. 13, no. 2, pp. 90-98, 1993.
- [3] R. W. Lewis, *Programming industrial control systems using IEC 1131-3*, 2nd ed., London, U. K. : The Institution of Electrical Engineers, 1995.
- [4] K. Venkatesh, M. C. Zhou, and R. Caudill, "Comparing ladder diagrams and petri nets for sequence controller design through a discrete manufacturing system," *IEEE Trans. on Industrial Electronics*, vol. 41, no. 6, pp. 611-619, 1994.
- [5] 강봉석, 조광현, "이산사건모델에 기반한 PLC 래더다이아그램 자동합성," *제어자동화시스템공학 논문지*, vol. 7, no. 11, pp. 939-943, 2001.
- [6] M. Sampath, R. Sengupta, S. Laforutne, K. Sinnamo-hideen, and D. Teneketzis, "Failure diagnosis using discrete-event models," *IEEE Trans. Contr. Syst. Technol.*, vol. 4, no. 2, pp. 105-124, 1996.
- [7] M. Sampath, S. Laforutne, and D. Teneketzis, "Active diagnosis of discrete-event systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 7, pp. 908-929, 1998.
- [8] K.-H. Cho and J.-T. Lim, "Synthesis of fault-tolerant supervisor for automated manufacturing systems : A case study on photolithographic process," *IEEE Trans. Robot. Automat.*, vol. 14, no. 2, pp. 348-351, 1998.
- [9] G. Provan and Y. L. Chen, "Model-based fault tolerant control reconfiguration for discrete event systems," *Proc. IEEE Int. Conf. Control Applications*, pp. 473-478, 2000.

- [10] K.-H. Cho and J.-T. Lim, "Multiagent supervisory control for anti-fault-propagation in serial production systems," *IEEE Trans. on Industrial Electronics*, vol. 48, no. 2, pp.

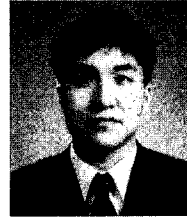
460-466, 2001.

- [11] 김원희, PLC를 중심으로 한 종합시퀀스 제어, 성안당, 1999.



강 봉 석

1975년생. 2000년 울산대학교 전기 공학과 졸업. 2000년~현재 울산대학교 전기전자정보시스템공학부 석사 과정 재학중. 관심분야는 PLC 응용 기술 및 생산시스템 자동화, 광통신 망. 이산사건시스템의 관리제어.



조 광 현

1993 한국과학기술원 전기 및 전자 공학과 졸업. 동대학원 석사(1995). 동대학원 박사(1998). 한국과학기술원 위촉연구원(1998-1999). 1999-현재 울산대학교 전기전자정보시스템 공학부 조교수. 관심분야는 이산사건시스템의 관리제어, 생산시스템 자동화, 통신망 분석 및 제어, 광통신망, 지능형 교통관제시스템, 바이오정보공학 등.