

제어시뮬레이션을 위한 생산시스템 로그데이터 기반 플랜트 모델 생성 방법

고민석¹ · 천상욱² · 박상철^{1†}

¹아주대학교 산업공학과, ²대우조선해양 중앙연구소 정보기술팀

A Method for Generating a Plant Model Based on Log Data for Control Level Simulation

Minsuk Ko¹, Sang Uk Cheon², and Sang Chul Park^{1†}

¹Department of Industrial Engineering, Ajou University

²Daewoo Shipbuilding & Marine Engineering

Received 9 October 2012; received in revised form 5 December 2012; accepted 21 December 2012

ABSTRACT

Presented in the paper is a log data based modeling method for effective construction of a virtual plant model which can be used for the virtual PLC (Programmable Logic Controller) simulation. For the PLC simulation, the corresponding virtual plant, consisting of virtual devices, is required to interact with the input and output symbols of a PLC. In other words, the behavior of a virtual device should be the same as that of the real device. Conventionally, the DEVS (Discrete Event Systems Specifications) formalism has been used to represent the behavior a virtual device. The modeling using DEVS formalism, however, requires in-depth knowledge in the simulation area, as well as the significant amount of time and efforts. One of the key ideas of the proposed method is to generate a plant model based on the log data obtained from the production system. The proposed method is very intuitive, and it can be used to generate the full behavior model of a virtual device. The proposed approach was applied to an AGV (Automated Guided Vehicle).

Key Words: DEVS, PLC simulation, Virtual commissioning, Virtual plant model

1. 서 론

현대 사회에서 기업의 성공 요건은 급변하는 소비자의 요구를 빠른 시간에 맞춰주는 데 있다. 이는 짧아진 제품 수명 주기(Product Life-cycle)에 대응 가능한 생산 시스템을 구축하는 문제로 귀결

되며, 새로운 제품을 위한 생산 시스템 구축 시, 발생하는 Down Time과 Ramp up Time을 효과적으로 감소시킴으로써 해결할 수 있다. 따라서 가장 많은 시간과 비용을 필요로 하는 생산 시스템 구축 초기 단계에서 공정 및 생산 라인의 설계, 검증을 통해, 시운전 및 현장 보정 기간을 단축할 수 있다^{1,2)}.

이 같은 목적 달성을 위해 제어 프로그램 검증을 통한 생산시스템 조기 안정화에 관한 연구가

[†]Corresponding Author, scpark@ajou.ac.kr
©2013 Society of CAD/CAM Engineers

활발히 수행되고 있다^[3-8]. 보통의 자동화 생산 공정에서 제어 프로그램은 PLC(Programmable Logical Controller) 프로그램을 말하는데, 이는 입력 신호와 출력 신호를 통해 대상 시스템을 논리적으로 운용하는 역할을 한다. 즉, 생산시스템의 구성요소들의 작업 수행과 선후 연결관계를 논리적으로 규정하는 시스템의 중추로써 역할을 한다. 하지만, PLC 프로그램은 센서(sensor), 액추에이터(actuator) 단위 신호 조합으로 표현되는 저 수준 언어(Low level language)이기 때문에, 엔지니어의 직관적 이해가 매우 어렵다. 또한, 프로그램 테스트 및 디버깅은 Off-line(생산시스템 현장)에서 수행되기 때문에, 프로그램 오류는 생산 일정에 치명적 영향을 끼친다.

이러한 PLC 프로그램 작성 및 테스트의 문제를 해결하고자, 자동화 산업 전반에 걸쳐 플랜트 모델(Plant model) 기반 PLC 시뮬레이션이 활발히 적용되고 있다^[4,8]. 가상환경의 플랜트 모델이 실제 설비의 행위를 대신함으로써, 엔지니어는 쉽고 빠르게 프로그램의 테스트 및 디버깅을 수행할 수 있게 되었다. Fig. 1은 가상환경 기반 PLC 시뮬레이션의 개념 및 구성 요소로써, 1) PLC(S/W, H/W), 2) PLC 프로그램, 3) 플랜트 모델(Plant Model)로 구성된다. 첫째, PLC는 생산시스템에 설치되는 S/W 또는 H/W 형식의 CPU이며, 둘째, PLC 프로그램은 PLC에 입력되는 제어 프로그램을 말한다. 그리고, 플랜트 모델은 PLC의 출력 값을 토대로 설비 행위를 대신하고, 행위 결과를 입력 신호로 산출하는 이산사건 모델(Discrete Event Model)

을 말한다. 특히, 본 연구에서 말하는 플랜트 모델은 PLC 시뮬레이션에서 가장 널리 사용되는 DEVS(Discrete Event Specification)의 원자(Atomic) 모델을 말한다^[9]. 과거 플랜트 모델이 없는 환경에서 PLC 프로그램 검증은, 현장 설비 대상의 시운전(commissioning)을 통해서만 가능하였다. 하지만, 가상환경 기반의 PLC 시뮬레이션을 통해 플랜트 모델과 PLC 프로그램이 Off-line상에 연결됨으로써, 사용자는 프로그램 검증 및 시운전을 보다 수월하게 진행할 수 있게 되었다.

이러한 PLC 시뮬레이션 도입으로 생산시스템 구축 및 생산시스템 안정화 시간을 크게 단축시킬 수 있게 되었다. 반면, PLC 시뮬레이션에 필요한 플랜트 모델을 작성에 소모되는 노력과 비용은 큰 낭비요소로 지적되고 있다. 왜냐하면, 대상 시스템을 이해하고 이를 플랜트 모델로 작성하는 과정에서 모델링 지식이 요구되고, 모델 오류가 발생하면 오히려 On-line 적용 시점이 늦어지기 때문이다. 이 같은 플랜트 모델 작성의 어려움을 해결하고자 PLC 심볼 이름을 기반으로 플랜트 모델을 생성하는 방법론을 제안되었으며^[10], 템플릿 기반의 플랜트 모델 생성 연구가 수행되었다^[10]. 하지만, 이러한 기존 연구들은 생산시스템 설계 단계(기계적 설계, 전기적 설계)에서 제안 방법론을 적용해야 하는 한계를 갖는다. 따라서, 기 구축된 생산시스템의 개조, 확장에 제안 방법론을 적용할 경우, 오히려 불필요한 낭비요소를 동반하게 된다.

본 연구에서는 이러한 기존 방법론의 한계를 효과적으로 해결하기 위해, 생산시스템의 운용정보인 LOG 정보를 기반으로 플랜트 모델을 자동 생성하는 방법론을 제안하고자 한다. 즉, PLC 기반으로 기 운용중인 시스템의 LOG 정보를 사용해 플랜트 모델을 생성함으로써 보다 빠르고 정확한 모델 및 시뮬레이션 환경 구축할 수 있는 효과를 얻을 수 있다. 디지털 제조 영역이 확대되고, PLC 시뮬레이션 연구가 실제 적용으로 이어지기 위해 플랜트 모델 작성 효율화를 위한 본 연구는 반드시 필요하다.

이를 위해 본 연구에서는 시스템 운용 시 발생하는 LOG 정보를 기반으로 플랜트 모델을 구성하는 세 개 집합 $\{I, O, S\}$ 과 네 개 함수 $\{\sigma_{int}, \sigma_{ext}, \lambda, t\}$ 를 자동으로 도출하는 방법을 제안한다. 여기서 σ_{int} 는 DEVS의 내부 상태 천이 함수(internal transition function), σ_{ext} 는 외부 상태 천이 함수

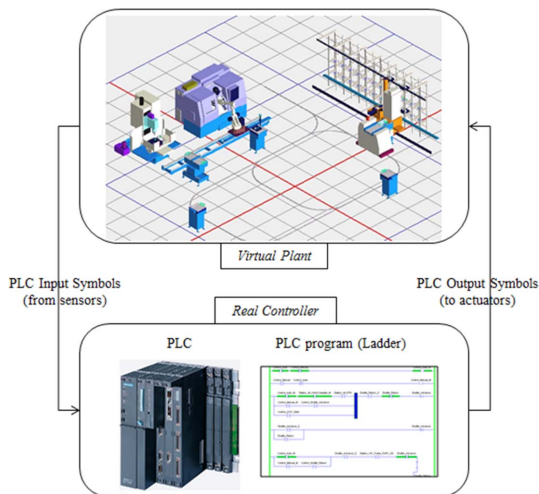


Fig. 1 PLC simulation environment

(external transition function), λ 는 출력함수(output function), t 는 시간 전진 함수(time advanced Function)을 말한다⁹⁾. LOG 정보 기반의 플랜트 모델 생성 방법론은 역공학(reverse engineering) 관점으로 모델을 생성하기 때문에, 모델의 사실성이 높고, 앞에서 언급한 낭비 요소가 제거된다는 큰 의미를 갖는다.

본 논문의 제 2장에서는 LOG 정보 구성 및 제어 시뮬레이션 접근 방법론에 관하여, 제 3장에서는 LOG 정보 기반의 플랜트 모델 생성 방법을 말하며, 제 4장에서는 예제를 대상으로 시뮬레이션 모델 구축 방법 및 결과를, 제 5장에서 결론을 말할 것이다.

2. 문제 접근 방법

본 연구에서 제안하는 LOG 정보 기반 플랜트 모델 생성 방법론은 기 구축된 생산시스템의 PLC 시뮬레이션 적용 시에 유용하게 사용될 수 있다. 여기서 LOG는 시간에 따른 제어 신호의 집합으로써, 일반적 생산시스템에서 쉽게 얻을 수 있는 LOG {시간, 신호} 형태를 말한다. 본 방법론의 데이터 흐름 및 적용 방안을 도식화하면 Fig. 2와 같다.

먼저, 본 방법론을 적용하기 위한 기본 제약 사항을 정리하면 다음과 같다.

- 1) 생산 시스템을 구성하는 각 device는 유한 개 (N)의 TASK를 가지고 있다.
- 2) 각 device는 한번에 하나의 TASK만 수행한다.
- 3) 각 TASK(T)는 한 개의 PLC 출력신호(Output

signal: X)와 이를 검증하기 위한 한 개의 PLC 입력신호(Input signal: Y)를 갖는다. 즉, $\{X, Y, T\}$ 간에 1:1:1 연결관계가 성립한다.

- 4) 각 device는 계획된 TASK를 수행한 뒤, 초기 상태로 회귀하는 Cycle 형식 작업을 한다.
- 5) LOG 정보로부터 각 신호가 발생한 시점을 획득할 수 있다.

이상의 다섯 가지 제약사항을 만족하는 device로 구성되는 생산시스템에 대하여 본 방법론을 적용하고자 한다. 제안한 Fig. 2의 첫 단계는 현재 구동중인 생산시스템으로부터 LOG data를 획득하는 것으로 시작한다. 이때 형태는 아래 LOG_{*i*}와 같으며, 이에 대한 예는 Table 1과 같다.

$LOG_i = \{Time (t), Signal\}$

Time (t): Marked time ' t '

Signal: PLC Signal Symbol or Address

// k is the number of log signals (symbols).

LOG_{*i*} (LOG data)는 일반적인 생산시스템에서 쉽게 획득할 수 있는 형태이다. 하지만, 본 정보는 Signal 시스템 상의 사용 명세를 확인할 수 없다. 따라서 이 정보를 device 및 Task(작업) 단위로 분리하는 것이 필요한데, 이때 PLC I/O Signal Table을 이용한다. 이 정보는 제어 엔지니어가 생산시스템 신호를 라인, 공정, 설비, 작업 단위로 관리하기 위해 신호를 명시하는 정보로써, 서로의 신호 주소(address)를 침범하지 않기 위해서 반드시 정의되어야 하는 사항이다. 따라서, 최초 Signal을 할당할 때 사용하는 PLC I/O Signal table에 대입하여 각 Signal의 명세 정보를 획득한다. 이때 형태는 아래 Signal_{*j*}와 같으며, 이에 대한 예는 Table 2와 같다.

$Signal_j = \{Device, Task, I/O Address, Symbol\}$

DEVICE: A device included in the production system.

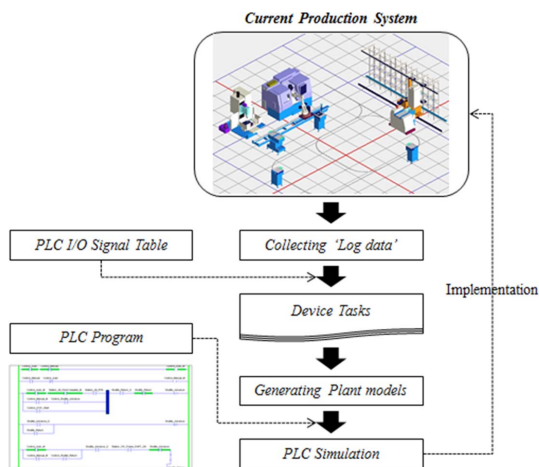


Fig. 2 Procedure of the proposed method

Table 1 LOG data table

LOG(i)	Time(t)	Signal
1	$Time_1$	I_signal_1
2	$Time_2$	O_signal_2
...
k	$Time_k$	I_signal_k

Table 2 PLC I/O signal table

#	DEVICE	TASK(t)	I/O Address	Symbol
1	Device ₁	Task1	I.0001	Symbol_1
2	Device ₁	Task1	O.0001	Symbol_2
...	
j	Device _i	Task _T	I.XXXX	Symbol_j

// i is the number of devices.

TASK: A device task.

I/O Address: PLC memory address.

Symbol: PLC Signal Symbol.

// j is the number of I/O signals (symbols).

이상의 두 단계를 거쳐 얻어진 LOG data와 PLC I/O Signal table를 기반으로 생산시스템을 구성하는 단위 설비 및 해당 작업으로 구성되는 LOG 모델(M)을 획득할 수 있다.

$$M_{Factory} = \{M_1, M_2, \dots, M_N\}$$

// N is the number of device models

$$M_n = \langle X_i, Y_i, T, Time \rangle$$

X_i : PLC output signal set;

Y_i : PLC input signal set;

T : Task set;

Time: (X or Y) \rightarrow Real: marked time;

N 개의 device LOG 모델(M)로 구성되는 공정 모델($M_{Factory}$)에 대하여, 각 device는 유한개의 Task를 가지며, 각 Task는 출력을 위한 X 와 결과 확인을 위한 Y 신호를 가지고 있다. 그리고 각 신호 (X or Y)는 Time 함수를 통해 LOG Data로 출력된 시간 값을 획득할 수 있다(Time: \rightarrow Real).

위 과정을 통해 LOG 모델 M_n 을 얻으면, 본 연구에서 목표로 하는 플랜트 모델 자동 생성 방법을 적용할 수 있다. 본 연구에서 말하는 플랜트 모델은 DEVS의 원자(Atomic) 모델로써, 다음 7가지 요소로 구성된다.

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

X : input events set;

S : sequential states set;

Y : output events set;

δ_{int} : $S \rightarrow S$: internal transition function;

δ_{ext} : $Q * X \rightarrow S$: external transition function

$Q = \{(s, e) | s \in S, 0 \leq e \leq t_a(s)\}$: total state of M ;

λ : $S \rightarrow Y$: output function;

t_a : $S \rightarrow Real$: time advance function.

이 같은 7가지 요소로 구성된 플랜트 모델을 이용하여 생산시스템을 구성하는 각 device 행위를 논리적으로 표현할 수 있으며, 이는 PLC 프로그램과 연동되어 PLC 시뮬레이션에 사용될 수 있다.

3. 플랜트 모델 생성

본 장에서는 2장에서 언급하였듯이, LOG 모델로부터 플랜트 모델을 생성하는 방법을 말하고자 한다. 다음은 LOG 모델에서 획득한 입력, 출력 정보와, 이들과 관계되는 Task로부터 플랜트 모델을 생성 방법을 순차적 단계로 표현한 것이다.

■ Generation of an atomic model from a device LOG model

// Input: device LOG model: $M_n = \langle X_i, Y_i, T, Time \rangle$

// Output: device plant model (an atomic model):

$$M_n = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

Step.1) X, Y 집합 정의

: LOG에 표기된 X, Y, T 를 통해 플랜트 모델의 X, Y 를 결정한다. 이때, 각 집합의 개수는 Task 개수(t)와 같다.

$$X = X_i;$$

// $X_i = \{O_T1, \dots, O_Tn\}$, n is the number of tasks

$$Y = Y_i;$$

// $Y_i = \{I_T1, \dots, I_Tn\}$, n is the number of tasks

$$T = T_i$$

// $T = \{T1, \dots, Tn\}$, n is the number of tasks

Step.2) 집합 S (State set) 정의

: 각 Task 는 세 개의 순차적 State $\{Start_T1, Do_T1, Done_T1\}$ 를 생성한다.

$$S = \{Start_T1, Do_T1, Done_T1, \dots, Start_Tn, Do_Tn, Done_Tn\}; // SizeOf(S) = 3 \times n$$

Step.3) Internal transition(δ_{int}) 정의

: Task 내부와 Task 간 Internal transition을 규정 짓는다.

$$Task \text{ 내부 상태 천이: } \delta_{int}(Do_T_i) = Done_T_i;$$

$$Task \text{ 간 상태 천이: } \delta_{int}(Done_T_i) = Start_T_{i+1};$$

Step.4) External transition(δ_{ext}) 정의
 : Start T_i 상태에서 PLC 출력신호 X_i 가 발생하면, External transition이 발생하여, Do T_i 로 전이한다.

$$\delta_{ext}(Start_{T_i}, O_{T_i}) = Do_{T_i};$$

Step.5) Output function (λ) 정의
 : 각 Task에 대하여 다음 Output functions가 정의된다.

$$\lambda(Do_{T_i}) = I_{T_i};$$

Step.6) Time advanced function(t_a) 정의
 : 각 Task에 대하여 다음 time advanced functions가 정의된다.

$$t_a(Start_{T_i}) = \infty;$$

$$t_a(Do_{T_i}) = T(I_{T_i}) - T(O_{T_i})$$

= T_i 를 수행하는데 소요되는 시간;

$$t_a(Done_{T_i}) = T(O_{T_{i+1}}) - T(I_{T_i})$$

= T_i 와 T_{i+1} 사이의 간격 시간

이상의 여섯 단계를 통해서 생산시스템으로부터 얻어진 LOG data 정보로부터 PLC 시뮬레이션을 위한 플랜트 모델을 생성할 수 있게 되었다. 다음 장에서는 간단한 AGV를 대상으로 본 방법론을 적용 예제를 설명할 것이다.

4. 적용 예제

본 연구에서는 AGV(Automatic Guided Vehicle)를 대상으로 LOG data 기반 플랜트 모델 생성 방법론의 적용 과정을 설명하고자 한다.

예제에 사용된 AGV는 두 개의 Task {T1, T2}를 가지며, I_T1_Done, I_T2_Done는 각 작업 완료에

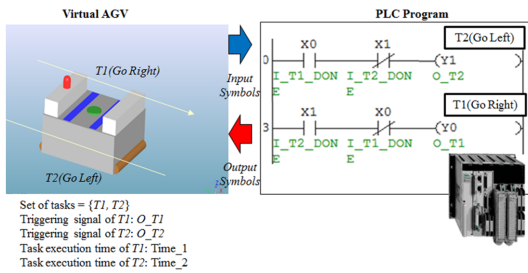


Fig. 3 Virtual device model of an AGV

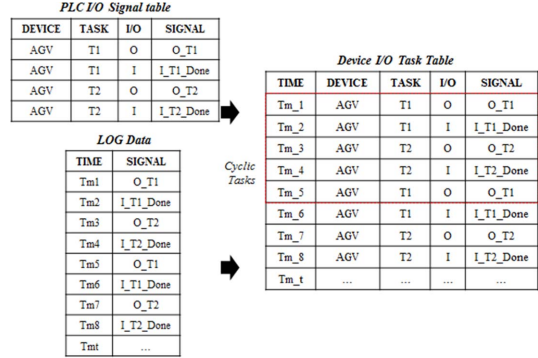


Fig. 4 Device I/O task table

대한 입력 신호로, O_T1, O_T2는 출력 신호로 사용된다. 본 AGV예제에 대한 LOG Data, PLC I/O Signal table와 이 둘을 입력으로 만들어진 device I/O task table은 Fig. 4와 같다.

Device I/O task table로부터 획득한 LOG Model (M_{AGV})은 아래와 같이 정의될 수 있다.

$$M_{AGV} = \langle X, Y, T, Time \rangle$$

$$X_i = \{O_{T1}, O_{T2}\}$$

$$Y_i = \{I_{T1_Done}, I_{T2_Done}\}$$

$$T_i = \{Task1, Task2\}$$

이 LOG Model로부터 본 연구에서 제안한 방법론을 순차적으로 적용하면 아래와 같다.

Step.1) X, Y 집합 정의

$$X: \{O_{T1}, O_{T2}\};$$

$$Y: \{I_{T1_Done}, I_{T2_Done}\};$$

Step.2) S (State set) 집합 정의

$$S: \{Start_{T1}, Do_{T1}, Done_{T1}, Start_{T2}, Do_{T2}, Done_{T2}\};$$

Step.3) Internal transition(δ_{int}) 정의

$$\delta_{int}(Do_{T1}) = Done_{T1};$$

$$\delta_{int}(Do_{T2}) = Done_{T2};$$

$$\delta_{int}(Done_{T1}) = Start_{T2};$$

$$\delta_{int}(Done_{T2}) = Start_{T1};$$

Step.4) External transition(δ_{ext}) 정의

$$\delta_{ext}(Start_{T1}, O_{T1}) = Do_{T1};$$

$$\delta_{ext}(Start_{T2}, O_{T2}) = Do_{T2};$$

Step.5) Output function (λ) 정의

$$\lambda(Do_T1) = I_T1_Done;$$

$$\lambda(Do_T2) = I_T2_Done;$$

Step.6) Time advanced function(t_d) 정의

$$t_d(Do_T1) = T(I_T1_Done) - T(O_T1)$$

$$t_d(Do_T2) = T(I_T1_Done) - T(O_T2)$$

$$t_d(Done_T1) = T(O_T2) - T(I_T1_Done)$$

$$t_d(Done_T2) = T(O_T1) - T(I_T2_Done)$$

$$t_d(Start_T1) = \infty;$$

$$t_d(Start_T2) = \infty;$$

이상의 과정을 거쳐 LOG data로부터 생성된 AGV 플랜트 모델은 Fig. 5와 같다.

이상의 과정을 통해 생성된 플랜트 모델은 Fig. 1에 제시된 구성요소들을 이용한 소프트웨어 PLC(GX Developer[®][11]) 및 PLC 프로그램과 연동되어 Off-line 상에서 시뮬레이션 될 수 있다(Fig. 6). 본 연구에서 제안한 방법을 통해 생성한 플랜트 모델은 PLCStudio[®]을 이용해 모델의 정상 운용되는지를 평가하였다. 적용실험에서 사용한 PLCStudio[®]은 DEVS 개념의 논리 모델(Logical model)과 물리모델(3D physical model)을 사용하여 제어 시

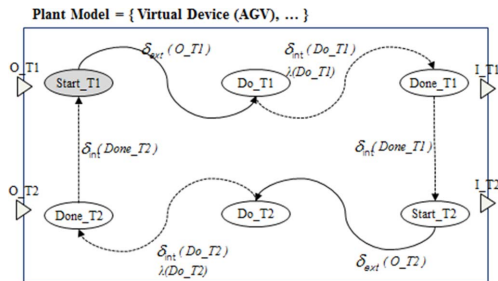


Fig. 5 Generated Virtual device model of an AGV

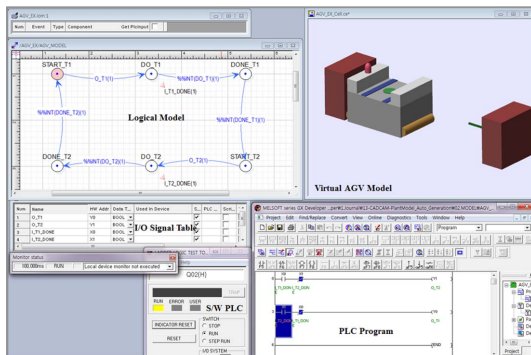


Fig. 6 PLC simulation environment for an AGV

뮬레이션을 수행하는 시뮬레이션 소프트웨어이다[3,5,10,12]. 제안한 방법론의 쉬운 이해를 위해 간단한 AGV를 예제로 사용하지만, 제 2장에서 언급한 제약사항 및 LOG 정보 획득 조건을 만족시킨다면 다양한 종류의 공정 설비들 (로봇, 지그, CNC 등)에 적용할 수 있다.

3. 결 론

다품종 소량생산이 보편화 됨에 따라 생산시스템 및 이를 제어하는 PLC 프로그램의 복잡도 또한 높아지고 있다. 따라서, 이러한 시스템적, 제어적 이슈를 Off-line 상에서 빠르게 테스트, 검증하기 위한 연구가 활발히 진행되고 있다. 특히, 가상환경 기반 PLC 시뮬레이션을 위한 연구는 자동화 생산 시스템을 중심으로 활발히 수행되고 있다. 이러한 Off-line 상의 PLC 시뮬레이션이 생산라인의 안정성 확보 및 시운전 효과의 증대를 가지고 오는 반면, 시뮬레이션에 필요한 플랜트 모델 작성에 대한 노력이 걸림돌로 작용하고 있다. 여기서 플랜트 모델은 PLC 시뮬레이션 환경에서 실제 모델의 행위를 대신하는 가상 모델로써, DEVS의 원자 모델을 말한다.

본 연구에서는 이러한 문제를 해결하고자 기 구축된 생산시스템으로부터 획득한 LOG 정보에 기반을 둔 플랜트 모델 자동생성 방법론을 제안하였다. 특히, 생산 시스템 운용 결과인 LOG 정보를 기반으로 제안한 LOG 모델 및 플랜트 모델을 생성하여, 모델작성에 필요한 낭비를 줄일 수 있도록 하였다. 기존의 플랜트 모델 작성 및 생성 방법론의 접근법과 다르게, 역공학 관점으로 접근하여 문제를 해결하였다. 제품의 설계부터 이를 만들기 위한 생산 그리고 시스템 검증 등의 이슈에서 다양한 가상환경 기반 시뮬레이션 연구가 수행되고 있다. 이러한 연구 및 적용 흐름에 맞춰 시뮬레이션 수행 효율화를 위한 방법 및 적합한 모델 작성 방법에 대한 연구가 필요하다.

감사의 글

본 연구는 방위사업청(UD110006MD), 국방과학연구소(UD100009DD), 한국연구재단(2012-0007496), 중소기업청 산학연 협력사업(C0003579)의 지원으로 수행 되었습니다.

참고문헌

1. Tempelmeier, H. and Kuhn, H., 1993, Flexible Manufacturing Systems, John Wiley and Sons.
2. Zeo, M. and Twiss, E., 1998, Design of Industrial Automated Systems Via Relay Ladder Logic Programming and Petri Nets, *Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, 28(1), pp. 137-150.
3. Koo, L. J., Park, S. C. and Wang, G. N., 2009, A Study and Application of Methodology for Applying Simulation to Car Body Assembly Line Using Logical Model, *Transactions of the Society of CAD/CAM Engineers*, 14(4), pp. 225-233.
4. Park, H. T., Wang, G. N. and Park, S. C., 2008, PLC Symbol Naming Rule for Auto Generation of Plant Model in PLC Simulation, *Journal of the Korea Society for Simulation*, 17(4), pp. 1-9.
5. Ko, M. S., Koo, L. J., Kwak, J. G., Hong, S. H., Wang, G. N. and Park, S. C., 2009, A Study of PLC Simulation for Automobile Panel AS/RS, *Journal of the Korea Society for Simulation*, 18(3), pp. 1-11.
6. Bae, S. H. and Kiam, Y. M., 2010, A Case Study of Virtual PLC Validation System's Implementation: In Case of An Automobile Trim Line, *Journal of the Korea Society for Simulation*, 19(2), pp. 9-16.
7. Park, S. C., Park, C. M. and Wang, G. N., 2008, A PLC Programming Environment Based on a Virtual Plant, *Int. J. Adv. Manufac. Technol.*, 39, pp. 1262-1270.
8. Park, C. M., Seong, G. Y., Park, S. C., Wang, G. N. and Han, K. H., 2008, Simulation Based Control Program Verification in Automobile Industry, *The International Conference on Modeling Identification and Control*, Innsbruck, Austria.
9. Kim, T. G., 1994, DEVSIM++user's Manual. Korea: Department of Electrical Engineering, KAIST.
10. Shin, H. S., Ko, M. S., Hong, S. H., Park, S. C. and Wang, G. N., 2011, A Template Based Process Modeling Methodology for Control Simulation, *Transactions of the Society of CAD/CAM Engineers*, 16(5), pp. 351-360.
11. GX-Developer User Manual, <http://www.mitsubishicorp.com>.
12. PLCStudio User Manual, <http://www.udmtek.com>.



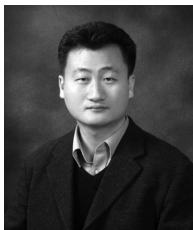
고민석

2007년 아주대학교 산업정보시스템 공학부 학사
 2013년 아주대학교 산업공학과 대학원 박사
 관심분야: Digital Manufacturing System, CAD/CAM, CAPP, Manufacturing System Modeling & Simulation



천상욱

2008년 KAIST 기계공학 박사
 2008년~현재 대우조선해양 중앙연구소 정보기술팀
 관심분야: Ship CAD, Freehand sketch-based modeling, Geometric modeling



박상철

1994년 한국과학기술원 산업공학과 학사
 1996년 한국과학기술원 산업공학과 석사
 2000년 한국과학기술원 산업공학과 박사
 2004년~현재 아주대학교 산업정보시스템공학부 부교수
 관심분야: Digital Manufacturing System, CAD/CAM, CAPP, Manufacturing System Modeling & Simulation