The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 20, No. 2, pp.261-269, Apr. 30, 2020. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2020.20.2.261
JIIBC 2020-2-34

제조설비 상태 진단 알고리즘 기반의 공장 모니터링 시스템에 대한 연구

A Study on Factory Monitoring System based on Manufacturing Facility Condition Diagnosis Algorithm

송은주*, 송교진*, 고동범**, 박정민***

Enjoo Song*, Kyogin Song*, Dongbeom Ko**, Jeongmin Park***

요 약 본 논문은 스마트 팩토리의 효율적인 오류 탐지를 위한 설비 시뮬레이션 시스템을 소개한다. 공장 설비들의 관계를 분석하여 설비오류 탐지시 오류를 자율적으로 추론하고 해결할 수 있는 설비 시뮬레이션 시스템은 높은 생산성을 가진 스마트 팩토리를 구성하는 데 중요한 기술 중 하나이다. 이러한 자율 제어 시스템 구현을 위해서는 공장 설비의 데이터를 통해 설비의 상태를 파악하고 설비 간 관계를 분석할 수 있어야 한다. 따라서 본 논문은 정의된 설비 상태를 이용하여 공정 시나리오를 기반으로 오류 발생시 공정 오류의 원인이 되는 설비를 쉽게 탐지할 수 있는 프로그램을 설계하고 개발한다. 시뮬레이션을 통해 공정 Map과 설비 상태 기반의 오류 추론 과정이 일반적인 오류 추론 과정보다 효율적임을 보였다. 본 시뮬레이션 프로그램을 통해 설비 오류 발생에 따른 추론 및 해결 과정을 직관적으로 볼 수 있도록 한다.

Abstract This paper introduces a facility simulation system for efficient error detection of smart factories. The facility simulation system, which can infer and solve errors autonomously when analyzing the relationship between plant facilities, is one of the important technologies for constructing a smart factory with high productivity. In order to implement this autonomic control system, it is necessary to be able to identify the status of facilities and analyze the relationship between facilities through the data of factory facilities. Therefore, in this paper, we design and develop a simulation program that can detect the equipment that causes the process error when an error occurs based on the process scenario using the defined equipment status. The simulation shows that the error inference process based on the process map and facility status is more efficient than the general error inference process. This simulation program provides an intuitive view of the reasoning and resolution of facility failures.

Key Words : MES, Digital Twin, CPS, Simulation

*준회원, 한국산업기술대학교 컴퓨터공학화
**준회원, 한국산업기술대학교 스마트팩토리융합학과
***정회원, 한국산업기술대학교 컴퓨터공학과(교신저자)
접수일자 2020년 1월 15일, 수정완료 2020년 3월 5일
게재확정일자 2020년 4월 3일

Received: 15 January, 2020 / Revised: 5 March, 2020 / Accepted: 3 April, 2020
*Corresponding Author: jmpark@kpu.ac.kr
Dept. of Computer Engineering, Korea Polytechnic University, korea

Ⅰ. 서 론

최근 인공지능, 빅데이터를 활용한 첨단 기술이 등장하면서 4차 산업혁명에 대한 관심이 높아지고 제조 산업에 큰 변화를 가져오게 되었다. 제조 산업을 기반으로 한 중국, 일본, 미국 등의 국가에서는 새로운 정책을 수립하기 시작했고, 국내의 경우 제조업 3.0 정책을 발표하였다. 특히 통신 기술이 발달하면서 기존보다 고성능의 센서와 인공 지능 기반의 시스템, 사물 인터넷(Internet of Things, IOT)과 같은 기술들이 결합된 'Industry 4.0'이 등장하게 되었고, 기존의 '소품종 대량생산'의 제조업 패러다임을 다양한 종류의 상품을 유연하게 생산하는 '다품종 소량생산'으로 변화하였다. 이러한 제조업 패러다임에 발맞춰 제조 공장의 설비들의 유기적이고 정교한 제어가 가능한 디지털 트윈과 자율제어 시스템에 대한 연구가 활발하게 진행되고 있다[1,2].

디지털 트윈은 발전된 센서와 IOT 기술을 활용하여 실제 세계를 모델링하여 가상 환경에 구현하는 기술이다. 디지털 트윈 기술은 제조업 분야에서 제조 공장 내 설비 정보들을 바탕으로 가상 환경에서 공장 시뮬레이션을 진행하여 효율적인 공장 관리를 가능하게 한다. 실제 공장 환경은 여러 제조 설비들 간의 상호 작용을 바탕으로 가동되기 때문에 이를 가상 환경에서 시뮬레이션하기 위해서는 다양한 상황에 대처하고 판단할 수 있는 자율제어 시스템이 필요하다.

그러나 자율제어 시스템을 활용하여 규모가 큰 공장 환경을 시뮬레이션 할 경우 설비 오류 발생 시 오류 지점을 빠르게 파악할 수 없다는 단점이 존재한다. 이러한 단점을 해결하기 위해서는 기존연구에서 제안한 제조설비의 자율제어를 위한 4가지 상태를 기반으로 오류 진단 프로세스 및 알고리즘 정의가 필요하다. 따라서 본 논문에서는 제조 공장에서 설비들의 상태와 공정 과정 분석을 바탕으로 도출된 상태 진단 알고리즘을 활용하여 설비들의 상태를 모니터링하여 오류가 발생했을 때 오류 지점을 빠르게 찾아낼 수 있는 오류 진단 프로세스를 정의한다. 그리고 실제 전기 모터 생산 테스트 베드에 해당 프로세스를 적용한 공장 시뮬레이션 프로그램을 구현하고 테스트를 진행하여 정의된 프로세스와 알고리즘이 오류 지점 파악의 효율성을 검증한다.

본 논문의 구성은 다음과 같다. 2장에서는 제조 공장 설비를 기반으로 한 가상 공간 구성과 시뮬레이션 수행에 필요한 관련 기술을 소개하고, 3장에서는 오류 진단 프로세스와 시스템 구조를 설명한다. 4장에서는 시뮬레이션 시스템 소개 및 평가를 진행하고 5장에서 결론을 맺는다.

Ⅱ. 관련연구

본 장에서는 효과적인 자율제어가 가능한 스마트 팩토리와 이에 필요한 개념 및 기술인 디지털 트윈, 자율제어 시스템에 대해 알아본다.

1. 스마트 팩토리

스마트 팩토리는 공장 내 설비와 기계에 센서가 설치되어 데이터를 실시간으로 수집하고 분석되어 공장 내 모든 상황들을 확인할 수 있으며 이를 분석하여 사용자의 요구에 맞게 스스로 제어가 가능하다[3]. 스마트 팩토리의 목표는 각 공장에서 수집된 많은 양의 데이터를 분석하고, 상황에 따라 설비 스스로 의사결정을 할 수 있는 데이터 기반의 공장 운영체계를 갖추는 것이다. 구축된 데이터기반 운영체계는 원인을 알 수 없었던 갑작스러운 장애나 품질 불량과 같이 생산 현장에서 발생되는 오류의 인과관계를 쉽게 알아낼 있으며 자체적인 해결을 가능하게 한다[4]. 또한 스마트 팩토리로부터 수집된 데이터들은 MES, ERP와 같은 생산, 경영 분야의 시스템과 연동되어 효율적인 판단을 가능하게 하고, 공장은 최적의 생산체제 하에서 운영 될 수 있다.[5] 본 논문에서는 전기 모터 공장 환경에서 부품들을 옮기는 시뮬레이션을 진행하여 공장 상황을 모니터링하고 오류를 자율적으로 해결할 수 있는 프로그램을 제안한다.

2. 디지털 트윈

제조 기술의 혁신을 이끌어내는 스마트 팩토리를 만들기 위해서 독일의 Industry 4.0 전략은 사이버 물리 시스템(Cyber-Physical Systems, CPS)과 클라우드 컴퓨팅 및 사물 인터넷을 발전시켜 적용하고 있다[6]. 이 중 CPS는 실 세계의 물체와 가상 세계의 물체가 서로 상호작용할 수 있도록 맵핑하고, 연결된 시스템을 관리할 수 있는 기술이다[7]. Industry 4.0 전략의 주요한 구성 요소인 CPS를 정보 통신 기술을 이용해 실제 존재하는 생산 요소와 가상 생산 요소를 연결하여 의사 결정을 효율적으로 지원하는 디지털 트윈(Digital Twin)개념을 이용해서 구현할 수 있다[8,9].

성공적인 스마트 팩토리를 구축하기 위해서는 각 제조

설비의 상태와 상황을 실시간으로 확인할 수 있어야 한다. 또한 설비에 오류가 생겼을 때 오류가 난 설비를 확인하고 그 오류의 원인까지 파악하여 스스로 오류를 해결할 수 있도록 자율적이고 효율적인 의사결정을 할 수 있는 알고리즘이 필요하다. 본 논문에서는 실제 공장 환경과 가상의 시뮬레이션 환경을 통합 공장 맵을 통해 진행할 수 있는 프로그램을 제안한다. 이를 통해 실제 공장 환경에서 발생하는 오류들을 미리 확인하고 해결할 수 있는 전략 수립 및 실행에 기여한다.

3. 자율제어 시스템

자율제어 시스템은 사용자가 요구하는 목표에 따라 스스로를 관리하고 변화에 동적으로 적응할 수 있는 컴퓨팅 환경이다[10]. 자율제어 시스템은 기본적으로 1) 목표 시스템 모니터링, 2) 분석, 3) 계획, 4)실행의 4가지 단계를 가지고 있다. 1) 목표 시스템 모니터링 단계는 타깃 시스템의 목표를 추상화한 목표 모델을 바탕으로 전체 시스템의 요구되는 달성도를 계산하여 현재 상태를 관찰한다. 2) 분석 단계에서는 시스템의 만족도가 요구 만족도에 미치지 못할 경우 모니터링 단계에서 전달 받은 데이터를 기반으로 현재 발생한 오류를 분석하여 원인을 추론한다. 3) 계획, 4) 실행 단계는 2) 분석 단계에서 추론한 오류 원인을 토대로 오류 해결전략을 수립하고 이를 수행한다[11]. 이전 연구에서는 컴포넌트 기반의 자율제어 시스템을 제안하고 1)Ready, 2)Working 3) Finish, 4) Failure의 4가지 상태를 정의하여 원활한 자율제어 모니터링이 가능할 수 있게 하였다. 본 논문에서는 위 4가지 상태를 기반으로 자율제어 프로세스 및 알고리즘을 이용하여 모니터링 단계를 효율적으로 수행하며 오류발생 시 오류 지점을 빠르게 찾아낼 수 있는 시뮬레이션 프로그램을 설계하고 구현한다.

Ⅲ. 시스템 프로세스 및 구조

본 장에서는 설비를 분석하고 상태를 진단할 수 있는 알고리즘을 도입한 설비 시뮬레이션 프로그램의 시스템 프로세스 및 구조를 설명한다.

1. 시스템 프로세스

본 절에서는 제조 설비를 분석하여 이를 관리하기 위한 설비 상태와 통합 공장 Map을 정의하고, 시뮬레이션 도중 발생하는 오류 이벤트 감시, 진단 및 해결전략 수립을 위한 과정을 1)설비 레이아웃 분석, 2)공장 시뮬레이션, 3)오류 탐지 및 분석, 4)해결 전략 실행의 4단계 프로세스로 설명한다. 그림 1은 이러한 4단계의 프로세스를 나타낸다.

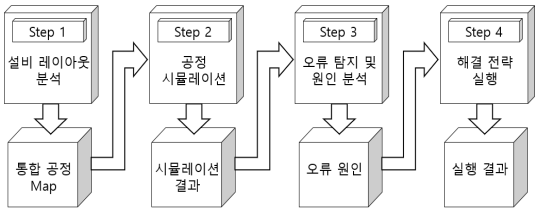

그림 1. 시스템 프로세스
Fig. 1. System Process

- Step1: 설비 레이아웃 분석

여러 제조 설비들을 시뮬레이션하기 위해서는 전체적인 공장 흐름 파악과 설비 작동에 관한 정보가 필요하다. 설비 레이아웃 분석 단계에서는 공장의 레이아웃과 공장에 필요한 부품, 공장별 이동경로 등을 포함한 설비 정보 분석을 진행한다. 또한 설비 정보를 바탕으로 설비 간의 관계를 파악하여 오류 탐지 및 원인 분석에서 사용되는 통합 공장 Map을 작성한다.

- Step2: 공장 시뮬레이션

공장 시뮬레이션 단계에서는 설비 레이아웃 분석을 통해 얻은 설비 정보들과 통합 공장 Map을 이용하여 시뮬레이션을 진행한다. 시뮬레이션 도중 변화하는 설비들의 상태를 통합 공장 Map에 갱신하여 시뮬레이터 결과를 도출한다. 도출된 시뮬레이터 결과는 추후 오류를 탐지하고 그 원인을 파악하는 데 사용된다.

- Step3: 오류 탐지 및 원인 분석

통합 공장 Map에 갱신된 설비들의 상태를 바탕으로 어느 설비에서 오류가 발생하였는지 탐지하고 분석하는 단계이다. 통합 공장 Map에 변화한 설비들의 상태를 통해 오류 발생 여부를 확인하고 오류가 발생했다면 오류의 종류(내부, 외부)를 파악하여 해당 오류의 원인을 파악한다.

- Step4: 해결 전략 실행

해결 전략 실행 단계에서는 이전 단계인 오류 탐지와

원인 분석 단계에서 도출된 오류 원인을 바탕으로 해당 오류 상황에 적합한 해결 전략을 세우고 실행한다. 이를 통해 오류가 해결된 시뮬레이션의 결과를 시각화한다.

2. 시스템 구조

공장 시뮬레이션 프로그램의 시스템 구조는 제조 설비를 기반으로 각 단계별 작업 수행 결과로 도출된 설비 데이터 및 통합 공장 Map, 시뮬레이션 결과, 오류 원인을 바탕으로 동작하는 Simulator의 'Machine Analysing Module', 'Simulation Executing Module'과 Simulator의 작업 진행 상황과 결과를 시각화하는 Visualizer의 'States Visualizing Module'로 구성된다. 그림 2는 해당 시스템의 전체적인 구조를 그림으로 나타낸다.

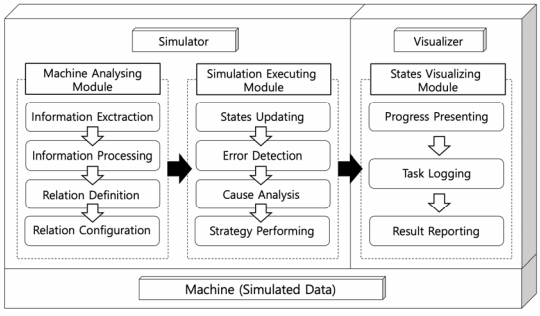

그림 2. 전체 시스템 구조
Fig. 2. System Architecture

가. Machine (Simulated Data)

전기 모터 부품을 생산하는 공장의 설비들을 기반으로 시뮬레이션을 진행할 설비의 정보를 가지고 있는 모듈이다. 여기에서 발생되는 값들은 Simulator가 효율적인 의사결정을 할 수 있는 데 사용된다.

나. Simulator

• Machine Analysing Module

Machine Analysing Module은 Machine으로부터 시뮬레이션에 필요한 정보를 추출하고 분석하는 작업을 진행한다. 그리고 추출된 정보를 기반으로 설비들 간의 관계를 정의하여 통합 공장 Map을 구성한다.

• Information Extraction

Machine으로부터 이전 설비, 다음 설비, 작업 시간 작업 내용 등을 포함한 시뮬레이션에 필요한 설비 정보를 추출하는 과정이다.

• Information Processing

이전 작업을 통해 얻은 raw data들을 Simulator에서 사용 가능한 데이터 형태로 가공한다.

• Relation Definition

Processing 과정을 통해 얻은 데이터들을 바탕으로 제조 설비 간의 관계를 정의한다.

• Relation Configuration

정의된 설비들 간의 관계를 바탕으로 통합 공장 Map을 구성한다.

• Simulation Executing Module

Simulation Executing Module은 Machine Analysing Module의 작업 수행 결과인 설비 데이터와 통합 공장 Map를 통해 시뮬레이션을 진행하며 설비의 동작에 맞춰 설비 상태를 Ready, Working, Finish, Failure의 4단계로 변경한다. 동작하는 설비들의 상태와 단계를 바탕으로 오류를 감지하고 오류가 발생한다면 이를 해결하기 위한 전략을 수행한다.

• States Updating

시뮬레이션 진행 도중 설비들의 동작에 따라 상태를 4가지로 분류한다. 1)재료나 부품을 기다리는 상태인 Ready, 2)작업을 진행하는 Working, 3)해당 작업이 완료된 Finish 상태로 분류하고 4)설비에 오류 발생 시 Failure 상태로 분류한다. 이 때 설비 내부 오류인 경우 Internal Failure, 주변 설비의 오류로 인해 정상적인 동작을 진행하지 못하는 경우 External Failure 상태로 분류한다.

• Error Detection

작동 중인 설비들의 상태를 확인하고, Failure 상태의 설비를 발견하면 해당 Failure 상태(Internal, External)에 따라 오류의 원인을 정의한다.

• Cause Analysis

Failure 상태로 변경돼 작동을 멈춘 설비들의 상태를 확인하고 다른 설비들의 External Failure 상태의 원인이 되는 Internal Failure 상태의 설비를 확인하고 오류 원인을 분석한다.

• Strategy Performing

분석한 오류 원인을 바탕으로 해당 오류를 처리하기 위한 적절한 해결전략을 확인하고 이를 수행한다.

다. Visualizer

• States Visualizing Module

시뮬레이션을 진행하면서 변화하는 설비들의 상태와 통합 공장 Map, 재료나 부품의 이동 경로와 각 설비들의 작업 내용을 시각화하고 사용자에게 보여준다.

• Progress Presenting

설비들을 통해 가공되는 재료나 부품의 이동 경로, 설비들의 4가지 상태(Ready, Working, Finish, Failure)를 사용자가 확인할 수 있도록 시각화한다.

• Task Logging

시뮬레이션 진행 도중 설비들이 재료나 부품을 가공하고 이동시키는 작업의 내용을 저장하고 표현한다.

• Result Reporting

사용자가 지정한 단위의 시뮬레이션 수행이 완료되었을 때 시뮬레이션 결과를 저장한다.

Ⅳ. 실험 및 결과

본 장에서는 제조 설비에서 생성되는 데이터들을 분석하고 상태 진단이 가능한 설비 시뮬레이션 프로그램을 구현하고 1) 공장 모니터링 실시, 2) 오류가 없는 정상 시나리오, 3) 설비에 오류가 발생한 이상 시나리오의 3단계로 데모 및 실험을 진행한다. 실험 수행을 위해 전기 모터 공장의 개별적인 설비 상태를 추상화할 수 있는 제조 설비의 4가지 상태를 정의한다. 또한 공장의 레이아웃 및 부품 별 공장 이동 경로를 추출함으로써 전체 공장의 흐름과 시나리오를 분석하고 분석한 결과를 토대로 모니터링을 수행한다. 모니터링 중 설비에 오류가 발생하면 오류가 발생한 설비를 추론하고 오류의 원인을 파악하여 자율적으로 오류의 원인을 해결할 수 있게 한다.

• 프로그램 구조 및 UI

설비 시뮬레이션 프로그램은 설비들의 작업 시간과 부품 전달 시간 등 정의된 정보들을 토대로 공장의 흐름과 시나리오를 분석하고 분석 값을 이용해 모니터링을 수행한다. 그림 3은 설비 시뮬레이션 프로그램의 구조와 프로세스를 도식화한 것이다.

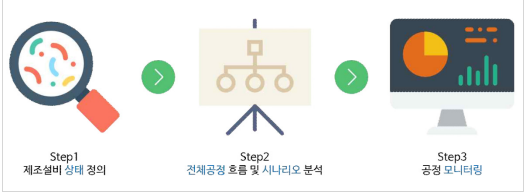

그림 3. 시뮬레이션 프로그램의 전체 설비 모니터링 과정
Fig. 3. Entire plant monitoring process of the simulation program

프로그램을 통해 공장의 각 설비의 상태를 간단히 확인할 수 있다. 또한 특정 설비에 오류가 발생하면 오류의 원인이 되는 장비를 쉽게 추론할 수 있다. 프로그램의 왼쪽 위에는 설비 정보를 나타내고, 왼쪽 아래에는 통합 공장 Map을 나타낸다. 마지막으로 오른쪽에는 시스템 로그가 실시간으로 출력된다. 'Simulation Start' 버튼을 클릭하면 시뮬레이션이 시작된다. 그림 4는 설비 시뮬레이션 프로그램의 UI를 보여 준다.

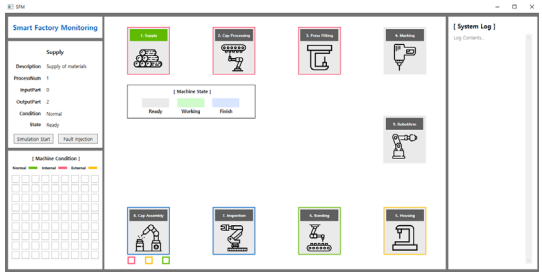

그림 4. 설비 시뮬레이션 프로그램의 UI
Fig. 4. Program UI of simulation program

• 1단계: 공장 모니터링 실시

먼저 정의한 제조 설비의 1) Ready, 2) Working, 3) Finish, 4) Failure의 4가지 상태와 공장 모니터링 알고리즘을 도입한 설비 시뮬레이션 프로그램을 이용하여 설비 모니터링을 실시한다. 모니터링 시 전기모터 공장의 설비들과 부품이 움직이는 이동 경로를 추출하고 이를 토대로 설비 데이터들을 분석하며 설비 오류를 진단한다.

공장의 레이아웃을 분석한 시뮬레이션 프로그램에는 9개의 설비가 있으며 시나리오를 진행하기 위해 각 설비에 번호를 붙인다. 설비 1은 Supply, 설비 2는 Cap

Processing, 설비 3은 Press Fitting을 수행하며 설비 4는 Marking, 설비 5는 Housing, 설비 6은 Bonding을 수행한다. 또한 설비 7은 Inspection을 수행하고 8은 Cap Assembly를 수행한다. 마지막으로 부품들이 조립되고 검사를 위해 이동하거나, 검사를 마친 후 마킹을 위해 부품이 이동할 때 사용되는 설비 9인 6축 로봇 팔 'RV-7EL'이 있다. 모터 캡, 모터 케이스, 모터 코일의 3가지 부품이 번호가 붙여진 설비에 따라 이동하며 조립된다.

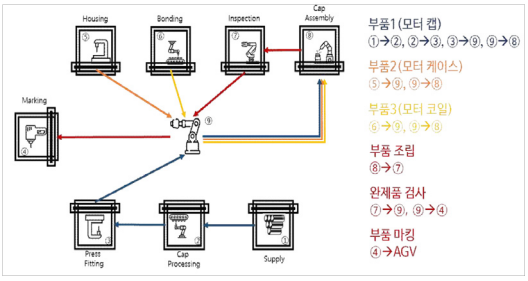

그림 5. 부품들의 이동 경로
Fig. 5. Path of components

모터 캡은 설비 1에서 2, 2에서 3, 3에서 9, 9에서 8번으로 이동하며, 모터 케이스는 설비 5에서 9, 9에서 8으로 이동한다. 마지막으로 모터 코일은 설비 6에서 9, 9에서 8으로 이동한다. 3가지의 부품이 모두 설비 8에 모이게 되면 부품들을 조립한 후 검사를 위해 설비 7으로 이동한다. 이후 검사가 완료된 부품들은 설비 7에서 9로 이동한 후, 9에서 4로 이동한다. 마지막으로 부품에 마킹하기 위해 설비 4에서 AGV(Autonomic Guided Vehicle)로 이동하게 된다. 그림 5는 시나리오에서 부품들의 이동 경로를 가시화한 것이다.

- 2단계: 오류가 없는 정상 시나리오

정상 시나리오에서는 설비들이 정의했던 상태 중 Ready, Working, Finish의 3가지 상태를 반복하며 부품들을 조립한다. 오류가 없는 정상적인 시나리오이므로 Failure 상태는 나타나지 않는다. 부품들이 조립되고 마킹을 위해 AGV로 이동하면 시뮬레이션이 종료된다.

- 3단계: 설비 오류 발생 시나리오

이상이 발생한 시나리오는 설비들의 동작이 정상적으로 수행되다가 설비 7에 오류가 나서 부품들이 제대로 이동할 수 없기 때문에 원인을 파악하여 오류가 해결되는 상황을 가정한 시나리오이다. 그림 6은 정상적으로 모터 캡이 설비 1에서 2로 이동하고 있는 모습을 보여 준다.

시나리오가 정상적으로 진행되던 상황에서 설비 7에 오류가 발생하는 상황을 가정한다. 설비 7에 오류가 발생하면 프로그램은 설비 7의 상태를 Internal Failure로 변경한다. 또한 내부 오류로 인해 설비 7에서 9로 이동할 수 없고, 설비 7은 설비 8에서 오는 부품을 받을 수 없으므로 왼쪽 아래의 통합 공장 Map에도 Internal Failure로 나타낸다. 이에 따라 설비 9도 다른 설비들에게 부품을 전달해 줄 수 없으므로 설비 9는 External Failure로 상태를 변경한다. 그림 7은 설비 7의 내부 오류가 나타난 직후의 상황을 보여 주는 화면이다.

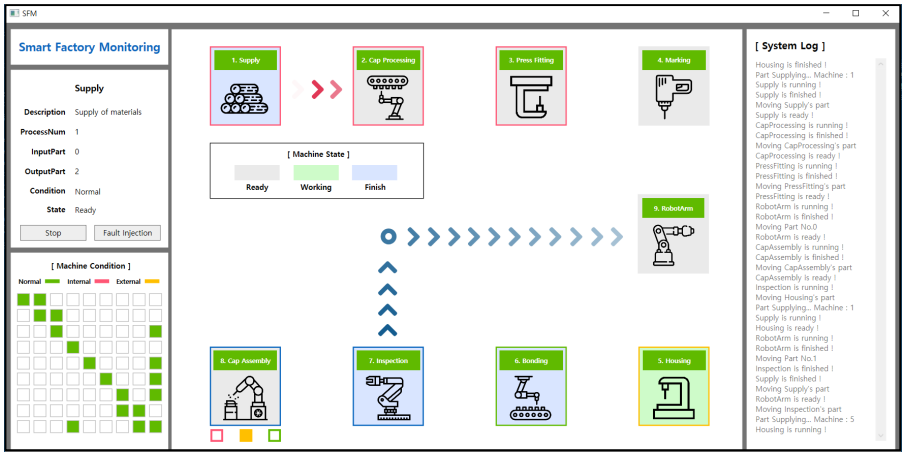

그림 6. 정상 시나리오의 시뮬레이터 프로그램
Fig. 6. Simulator program in normal scenario

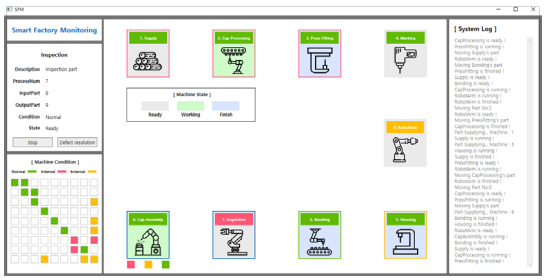

그림 7. 설비 7의 내부 오류가 나타난 직후의 상황
Fig. 7. Situation after internal error of facility 7.

설비 7에 오류가 발생함에 따라 전체적인 설비들이 부품들을 조립하거나 이동할 수 없으므로 순차적으로 다른 설비들도 External Failure 상태로 변경된다. 이상 시나리오가 진행되면 통합 공장 Map의 모든 설비들이 Failure 상태로 변경된다. 그림 8은 설비 7 외의 나머지 설비들이 외부 오류를 나타낸 모습을 나타낸 것이다.

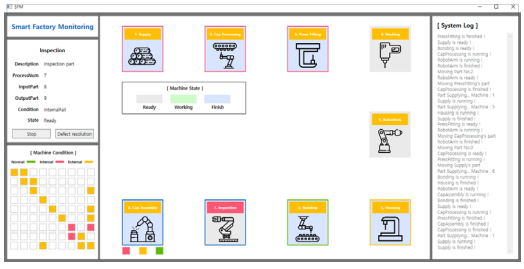

그림 8. 설비 7과 관련된 설비들이 외부 오류를 나타낸 모습
Fig. 8. External failure in facilities related to facility 7

설비 시뮬레이션 프로그램은 오류가 난 것을 확인하고 설비 7의 오류의 원인에 대해 파악하여 자율적으로 오류를 해결한다. 그림 9는 설비 7의 오류가 해결된 것을 보여준다.

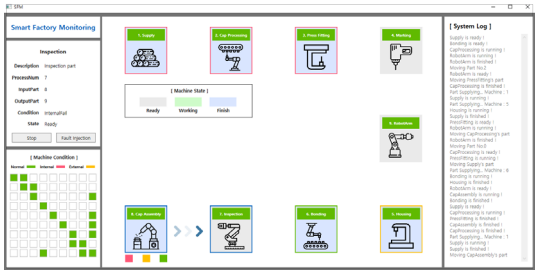

그림 9. 설비 7의 오류가 해결되고 정상 가동되는 모습
Fig. 9. Normal operation after fault in facility 7 has been cleared

- 성능 평가

설비 시뮬레이션을 진행하여 부품이 공급되는 단계들부터 조립되어 완성되는 단계까지의 시간을 측정한다. 부품이 완성되는 데 걸리는 시간은 부품들이 조립되는 시간과 마킹 설비에 도달되는 시간에 따라 큰 차이를 나타내게 된다. 그림 10은 100개의 공장 부품을 생산할 때 각각의 공장 부품을 생산하는데 걸리는 시간을 나타낸다.

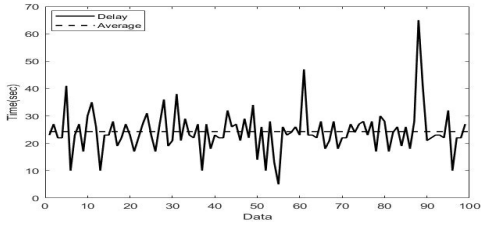

그림 10. 부품 생산 시간
Fig. 10. Time of component production

통합 공장맵의 오류 해결 방법의 성능을 평가하기 위해 모든 설비의 오류 상태를 직접 확인하는 방법(Brute force)과 오류 해결 시간을 비교 측정한다.

그림 11은 통합 공장맵을 이용한 오류 해결방법과 모든 설비의 오류 상태를 확인하는 방법의 각 설비별 오류 발생 시 오류 해결 시간을 나타낸다. 총 9개의 설비에서 각각 100번의 오류를 발생시키고 이를 두 가지의 방법으로 오류를 해결하였을 때 걸린 시간의 평균을 나타낸다.

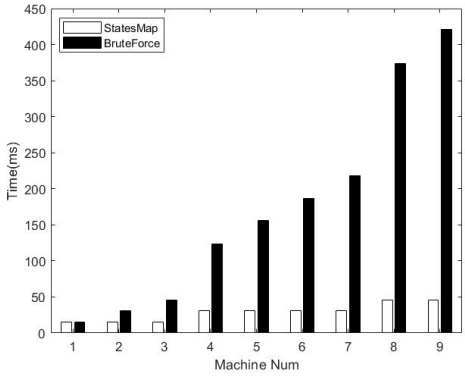

그림 11. 공정 맵과 전체 탐색을 이용한 오류 탐지 시간 비교
Fig. 11. Error detection time comparison using

통합 공정맵을 활용한 오류 해결 방법은 다른 설비들의 External Failure 상태의 원인이 되는 Internal Failure가 발생되는 설비를 통합 공장맵을 통해 확인하

게 된다. 이 방법은 어떤 설비에서 오류가 발생하더라도 오류 해결에 걸리는 시간이 큰 차이가 나지 않는다. 하지만, 모든 설비의 오류 상태를 확인하는 방법의 경우 Internal Failure가 발생한 설비를 먼저 확인할 때와 나중에 확인할 때의 시간이 큰 차이를 나타내게 된다.

V. 결 론

본 논문에서는 효과적인 의사결정을 할 수 있는 스마트 팩토리를 구축하기 위해 공장 설비들의 관계를 정의하여 Mapping하고 동작을 1) Ready, 2) Working, 3) Finish, 4) Failure 상태로 분류했다. 또한 설비들의 상황을 파악하고 오류가 발생하면 자율적으로 해결할 수 있는 방법에 대해 제안하였다. 제안 사항을 통해 제조설비 모니터링 시뮬레이터를 개발하고 시뮬레이션을 진행하면서 특정 설비에 오류가 발생하면 오류가 난 설비를 쉽게 추론할 수 있었다. 그리고 오류 원인을 분석하고 자율적인 의사결정으로 오류를 해결할 수 있었다.

스마트 팩토리 관점에서 설비에 오류가 발생했을 때 오류가 난 설비를 찾아내고 오류를 자율적으로 해결하는 것이 중요하다. 그러나 설비에 오류가 발생해 생산이 중단되면 고장 난 설비를 발견하기 위해 관리자가 확인하고 해결하는 기존의 방법은 스마트 팩토리의 생산성 저하를 일으킬 수 있다. 이는 설비의 상태를 정의하고 설비 간의 관계를 분석하여 통합 공장 Map을 작성함으로써 해결된다.

본 논문에서는 설비 관계 분석을 위해 전기 모터 공장의 레이아웃을 본 딴 가상의 설비를 사용했기 때문에 실제 공장 설비와 데이터를 통한 검증이 필요하다. 따라서 차후 실제 스마트 팩토리에서 사용하는 설비들의 관계를 분석하여 공장 Map을 작성하고 실제 설비에서 발생되는 데이터들을 활용하여 오류를 자율적으로 찾는 방안에 대해 연구할 예정이다.

References

[1] H.Y. Park, “Trend of Smart Factory and Manufacturing Technology”, Journal of Korean Communication Society, Vol. 33, No. 1, pp. 24-29, 2015.

[2] S.B Lee and J.K Park. “A Study on the Effects of Management Strategy, Collaboration Process and SCM Integration on Corporate Performance in Small and Midsize Parts Manufacturers”, IJournal of the Korea Academia-Industrial cooperation Society, Vol. 20, No. 7, pp.448-466, 2019.
DOI: https://doi.org/10.5762/KAIS.2019.20.7.448

[3] Osterrieder, P., Budde, L., & Friedli, T. “The smart factory as a key construct of industry 4.0: A systematic literature review”, International Journal of Production Economics, In Press, 2019.
DOI: https://doi.org/10.1016/j.ijpe.2019.08.011

[4] Longo, F., Nicoletti, L., & Padovano, “A. Smart operators in industry 4.0: A human-centered approach to enhance operators’ capabilities and competencies within the new smart factory context”. Computers & industrial engineering, Vol 113, pp. 144-159, 2017.
DOI: https://doi.org/10.1016/j.cie.2017.09.016

[5] Liu, W., Kong, C., Niu, Q., Jiang, J., & Zhou, X.. “A method of NC machine tools intelligent monitoring system in smart factories”. Robotics and Computer-Integrated Manufacturing, Vol 61, 2020.
DOI: https://doi.org/10.1016/j.rcim.2019.101842

[6] Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. “Intelligent manufacturing in the context of industry 4.0: a review”. Engineering, Vol. 3(5), pp. 616-630. 2017.
DOI: https://doi.org/10.1016/J.ENG.2017.05.015

[7] Lee, J., Bagheri, B., & Kao, H. A, “A cyber-physical systems architecture for industry 4.0-based manufacturing systems”. Manufacturing letters, Vol 3, pp. 18-23. 2015.
DOI: https://doi.org/10.1016/j.mfglet.2014.12.001

[8] N.C Do and Y.H Lee, “Development of Manufacturing Facility-Centered Digital Twin Architecture : Case of Building Process-BOM”, Korean Journal of Computational Design and Engineering, Vol. 24, No. 1, pp. 92-102. 2019.
DOI: https://doi.org/10.7315/CDE.2019.092

[9] K.S Kim, B.M. Lim, S.S Choe, S.Y. Ji and K.H Ryu. “Design and Implementation of Web-based Virtual Twin Model for Synchronization between Physical and Cyber Space in Manufacture Industry”, The Journal of Korean Institute of Information Technology, Vol. 14, No. 2, pp.115-124, 2016.
DOI: https://doi.org/10.14801/jkiit.2016.14.2.115

[10] Gronau, N. “Determining the appropriate degree of autonomy in cyber-physical production systems”. CIRP Journal of Manufacturing Science and Technology. Vol 26, pp. 70-80, 2019.
DOI: https://doi.org/10.1016/j.cirpj.2019.05.001

[11] Fakhir, M. I., & Kazmi, S. A. R. “Formal specification and verification of self-adaptive concurrent systems”, IEEE Access, Vol. 6, pp. 34790-34803, 2018.
DOI: https://doi.org/10.1109/ACCESS.2018.2849821

저 자 소 개

송 은 주(준회원)

- 2020년 : 한국산업기술대학교 컴퓨터공학과 학사

송 교 진(준회원)

- 2020년 : 한국산업기술대학교 컴퓨터공학과 학사

고 동 범(준회원)

- 2018년 ~ 현재 : 한국산업기술대학교 스마트팩토리융합학과 박사과정
- 2018년 : 한국산업기술대학교 컴퓨터공학과 석사
- 2016년 : 한국산업기술대학교 컴퓨터공학과 학사
- 주관심분야 : CPS, Autonomic Computing, Artificial Intelligence

박 정 민(정회원)

- 2014년 3월 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 부교수
- 2012년 7월 ~ 2014년 2월 : 한국전자통신연구원 선임연구원
- 2008년 ~ 2011년 : 동양미래대 조교수
- 2009년 : 성균관대학교 컴퓨터공학과 박사
- 2005년 : 성균관대학교 컴퓨터공학과 석사
- 2003년 : 한국산업기술대학교 컴퓨터공학과 학사
- 주관심분야 : CPS, Autonomic Computing, Software Enginnering

※ 이 성과는 2018년 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2017R1A2B4011243).
※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음"
(IITP-2019-2018-0-01426)