



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위 논문

Deep Learning Algorithm을 활용한
유압 매니퓰레이터의 고장 진단 및 복구에 관한 연구

A Study on Fault Diagnosis and Recovery
of Hydraulic Manipulator Using Deep Learning Algorithm

울산대학교 대학원
건설기계공학과
윤진수

Deep Learning Algorithm를 활용한
유압 매니플레이터의 고장 진단 및 복구에 관한 연구

지도교수 안경관

이 논문을 공학석사 학위 논문으로 제출함

2021년 2월

울산대학교 대학원

건설기계공학과

윤진수

윤진수의 공학석사 학위 논문을 인준함

심사위원 하 철 근



심사위원 이 병 룡



심사위원 안 경 관



울산대학교 대학원

2021년 2월

국문요약

매니플레이터는 사람을 대신해 국내외 산업에서 많이 쓰이고 있으며 최근에는 4차 산업혁명의 핵심 기술인 인공지능, 빅데이터, 사물인터넷과 같은 기술들이 제조 산업 전반으로 적용되고 있다. 또한 자동화 및 무인화 기술을 적용하여 생산 효율을 최대화하는 연구 역시 활발히 진행되고 있다.

매니플레이터에 부착된 다양한 센서에 고장이 생길 경우 매니플레이터의 오작동으로 공정 운영에 경제적 손실이 발생하는 것은 물론, 인명 피해도 발생할 수 있다. 이를 해결하기 위해 전문가가 일정 주기마다 이상 여부를 확인하여 관리하지만 고장 점검 시기를 놓치거나 발견하지 못하는 경우도 있으며, 데이터 수집을 위한 센서의 개수가 많아 이를 모두 확인하는 것은 어렵다. 따라서 자체적으로 임베디드 시스템에서 센서의 고장 유무를 실시간으로 파악하고 빠른 대응을 위해 고장 진단 및 유형을 판별하는 것이 필요하다.

고장 진단에 대한 접근방법 중 모델 기반 접근방법은 정확도가 높고 적은 양의 데이터로도 고장 진단이 가능하다. 또한 모델의 변수를 바꿔줌으로써 다양한 운영 환경에서도 적용이 가능한 장점이 있다. 그러나 모델 기반 방법은 시스템의 복잡도가 크게 증가하면서 모델을 완전히 구현하기 힘들고, 모델의 변수 또한 추정하기 쉽지 않다.

두 번째 방법인 데이터 주도 접근방법은 복잡한 시스템에 적용가능한 장점이 있지만, 예측 모델의 훈련을 위해서는 많은 데이터가 필요하다는 단점이 있다.

본 논문에서는 모델 기반 접근방법의 예측 불가능한 요인들을 보완하기 위해 데이터 주도 접근방법을 기반으로 한 고장 진단 및 복구 알고리즘을 제안하였으며, Deep Learning Algorithm을 적용하여 다음과 같은 결과를 도출하였다.

1. 3축 유압 매니플레이터에 대한 Forward Kinematics, Inverse Kinematics를 적용하여 매니플레이터의 기구학 특성을 분석하였으며, 유압 회로를 분석하였다.
2. 기구학 모델과 유압 모델을 연결하여 매니플레이터의 가상 시뮬레이션 모델을 구축하였다.
3. 일시적 고장에 대해 Kalman Filter를 적용하여 고장 신호를 복구하였다.
4. 영구적 고장에 대해 NN, RNN, LSTM, GRU를 활용한 알고리즘을 설계하여 적용하고 고장 신호를 복구하였다.
5. 가상 시뮬레이션 모델에서 검증된 결과를 바탕으로 실차에 적용하여 고장 신호를 복구하였으며, 고장 상황 발생 시 Position tracking이 잘 수행됨을 검증하였다.

목 차

목 차	i
표 및 그림 목차	iii
1. 서 론	1
1.1. 연구 배경 및 기존 연구 동향	1
1.2. 연구 목적 및 내용	5
2. 이론적 배경	6
2.1. 유압 매니퓰레이터	6
2.1.1. 정기구학	7
2.1.2. 역기구학	10
2.2. 유압 시스템	13
2.2.1. 유압 펌프	13
2.2.2. 유압 모터	16
2.2.3. 유압 실린더	19
2.2.4. 서보 밸브	20
2.3. 제어기	23
2.3.1. PID 제어	23
2.3.2. Sliding Mode 제어	24
2.4. 고장 진단 및 복구	26
2.4.1. 일시적 고장	27
2.4.2. 영구적 고장	29
2.4.2.1 NN (Neural Network)	30
2.4.2.2 RNN (Recurrent Neural Network)	31
2.4.2.3 LSTM (Long Short Term Memory)	32
2.4.2.4 GRU (Gated Recurrent Unit)	34
3. 시뮬레이션 및 결과	36
3.1. 시뮬레이션 모델 구축	36
3.1.1. 3D 모델링	36
3.1.2. Multi-Body 모델링	37
3.1.3. 유압 시스템 모델링	41
3.1.3.1 유압 펌프 모델링	41
3.1.3.2 유압 모터 모델링	42
3.1.3.3 유압 실린더 모델링	43

3.1.3.4 서보 밸브 모델링	44
3.1.3.5 유압 회로 모델링	45
3.1.4. 통합 모델링	47
3.2. 제어기 설계	49
3.2.1. PID 제어기 설계	49
3.2.2. Sliding Mode 제어기 설계	52
3.3. 고장 진단 및 복구 알고리즘 설계	55
3.3.1. 일시적 고장 복구 알고리즘 설계	55
3.3.2. 영구적 고장 복구 알고리즘 설계	57
4. 실차 실험 및 결과	66
4.1. 시스템 개요	66
4.2. 제어기 구축	68
4.2.1. PID 제어기 구축	69
4.2.2. Sliding Mode 제어기 구축	71
4.3. 고장 진단 및 복구 알고리즘 구축	73
5. 결론 및 향후 계획	81
참고문헌	82
Abstract	85

표 및 그림 목차

1. 표 목차

Table 2-1 Denavit - Hartenburg Table	8
Table 2-2 끝단 위치 계산 결과 (MATLAB 활용)	9
Table 2-3 끝단 위치 계산 결과 (벡터 기법 활용)	9
Table 2-4 3D Plotting 결과	10
Table 2-5 관절 제한 각도	10
Table 2-6 관절 제한 각도에 따른 매니플레이터 형상	10
Table 2-7 유압 펌프 제원	16
Table 2-8 유압 모터 제원	18
Table 2-9 유압 실린더 제원	20
Table 2-10 센서 고장 분류	26
Table 2-11 Kalman Filter 변수	27
Table 3-1 유압 매니플레이터 부품 형상	36
Table 3-2 관절 각도 값 비교 결과 (Calibration 수행 전)	38
Table 3-3 Calibration 수행 결과	38
Table 3-4 관절 각도 값 비교 결과 (Calibration 수행 후)	38
Table 3-5 Multi-Body 모델 동작 형상	39
Table 3-6 시뮬레이션 결과 (1→6 순서로 동작)	40
Table 3-7 유압 펌프 모델링 파라미터	41
Table 3-8 유압 모터 모델링 파라미터	42
Table 3-9 유압 실린더 모델링 파라미터	43
Table 3-10 밸브 모델링 파라미터	44
Table 3-11 시뮬레이션 결과 (Joint 1)	47
Table 3-12 시뮬레이션 결과 (Joint 2)	48
Table 3-13 시뮬레이션 결과 (Joint 3)	48
Table 3-14 기준 경로 설정 결과	49
Table 3-15 PID 제어기 파라미터	50
Table 3-16 기준 경로 추적 결과 (PID 제어기 적용 전)	50
Table 3-17 기준 경로 추적 결과 (PID 제어기 적용 후)	51
Table 3-18 Sliding Mode 제어기 파라미터	52
Table 3-19 기준 경로 추적 결과 (Sliding Mode 제어기 적용 전)	53
Table 3-20 기준 경로 추적 결과 (Sliding Mode 제어기 적용 후)	54
Table 3-21 Kalman Filter 초기값 및 파라미터	55
Table 3-22 노이즈 파라미터	56
Table 3-23 시뮬레이션 결과 (노이즈 상황)	56

Table 3-24 시뮬레이션 결과 (칼만 필터 적용 후)	56
Table 3-25 시뮬레이션 결과 (기준 경로 vs 칼만 필터 적용 결과)	56
Table 3-26 Case별 학습 데이터	57
Table 3-27 검증 데이터	57
Table 3-28 시뮬레이션 결과 (PID, Joint 1)	59
Table 3-29 시뮬레이션 결과 (PID, Joint 2)	60
Table 3-30 시뮬레이션 결과 (PID, Joint 3)	61
Table 3-31 시뮬레이션 결과 (SMC, Joint 1)	62
Table 3-32 시뮬레이션 결과 (SMC, Joint 2)	63
Table 3-33 시뮬레이션 결과 (SMC, Joint 3)	64
Table 4-1 매니플레이터 부품 제원	67
Table 4-2 기준 경로 그래프	68
Table 4-3 PID 제어기 파라미터	69
Table 4-4 기준 경로 추적 결과 (PID 제어기 적용 전)	69
Table 4-5 기준 경로 추적 결과 (PID 제어기 적용 후)	70
Table 4-6 Sliding Mode 제어기 파라미터	71
Table 4-7 기준 경로 추적 결과 (Sliding Mode 제어기 적용 전)	71
Table 4-8 기준 경로 추적 결과 (Sliding Mode 제어기 적용 후)	72
Table 4-9 Case별 학습 데이터	73
Table 4-10 검증 데이터	73
Table 4-11 실차 실험 결과 (PID, Joint 1)	74
Table 4-12 실차 실험 결과 (PID, Joint 2)	75
Table 4-13 실차 실험 결과 (PID, Joint 3)	76
Table 4-14 실차 실험 결과 (SMC, Joint 1)	77
Table 4-15 실차 실험 결과 (SMC, Joint 2)	78
Table 4-16 실차 실험 결과 (SMC, Joint 3)	79

2. 그림 목차

Fig. 1-1 국내외 매니플레이터 판매량 (2015년 ~ 2020년)	1
Fig. 1-2 국내외 고장 진단 관련 논문 편수 (2011년 ~ 2020년)	2
Fig. 1-3 국내외 매니플레이터 고장 진단 관련 논문 편수 (2011년 ~ 2020년)	2
Fig. 2-1 매니플레이터 구조	6
Fig. 2-2 표준 링크 도식도	7
Fig. 2-3 매니플레이터 기준 좌표계	8
Fig. 2-4 역기구학 해 검증 결과	12
Fig. 2-5 매니플레이터 유압 회로도	13
Fig. 2-6 유압 펌프 분류도	14

Fig. 2-7 유압 펌프 구조도	15
Fig. 2-8 유압 펌프 형상	15
Fig. 2-9 유압 액츄에이터 분류도	16
Fig. 2-10 베인 모터 구조도	17
Fig. 2-11 베인 모터 회로도	17
Fig. 2-12 유압 모터 형상	18
Fig. 2-13 유압 실린더 구조도	19
Fig. 2-14 유압 실린더 모델링 도식도	19
Fig. 2-15 유압 실린더 형상	20
Fig. 2-16 스펙 형상 (동적 상황)	20
Fig. 2-17 스펙 형상 (정적 상황)	21
Fig. 2-18 서보 밸브 형상 (유압 실린더 제어)	22
Fig. 2-19 서보 밸브 형상 (유압 모터 제어)	22
Fig. 2-20 PID 제어 블록 다이어그램	23
Fig. 2-21 센서 고장 분류에 따른 출력 그래프	26
Fig. 2-22 Kalman Filter 알고리즘	27
Fig. 2-23 고장 진단 및 복구 알고리즘	29
Fig. 2-24 Neural Network 구조	30
Fig. 2-25 표준 RNN 구조	31
Fig. 2-26 RNN 구조 (전개 형상)	31
Fig. 2-27 표준 LSTM 구조	32
Fig. 2-28 표준 GRU 구조	34
Fig. 3-1 시뮬레이션 모델링 진행 과정	36
Fig. 3-2 유압 매니플레이터 조립 형상	37
Fig. 3-3 Multi-Body 모델링 형상	37
Fig. 3-4 Multi-Body 모델링 형상 (Calibration 작업 후)	39
Fig. 3-5 유압 펌프 모델링 형상	41
Fig. 3-6 시뮬레이션 결과 (유량 비교)	42
Fig. 3-7 유압 모터 모델링 형상	42
Fig. 3-8 시뮬레이션 결과 (토크 비교)	43
Fig. 3-9 유압 실린더 모델링 형상	43
Fig. 3-10 시뮬레이션 결과 (힘 비교)	44
Fig. 3-11 밸브 모델링 형상	44
Fig. 3-12 시뮬레이션 결과 (유량 비교)	45
Fig. 3-13 유압 회로 모델링 형상	45
Fig. 3-14 시뮬레이션 결과 (유압 모터 1)	46
Fig. 3-15 시뮬레이션 결과 (유압 모터 2)	46
Fig. 3-16 시뮬레이션 결과 (유압 실린더)	46
Fig. 3-17 최종 모델링 형상	47
Fig. 3-18 제어기 블록선도	49

Fig. 3-19 Sliding Mode 제어기 블록선도	52
Fig. 3-20 일시적 고장 복구 알고리즘 구성	55
Fig. 3-21 영구적 복구 알고리즘 구성	58
Fig. 4-1 시스템 구성	66
Fig. 4-2 시스템 개략도	67
Fig. 4-3 유압 매니플레이터	68
Fig. 4-4 고장 진단 및 복구 알고리즘 스케치	73

1. 서론

1.1. 연구 배경 및 기존 연구 동향

매니플레이터란 인간의 상지¹⁾와 유사한 기능을 보유하고 끝단 부위에 해당하는 기계의 손으로 물체를 파지하여 공간적으로 이동시키는 작업 또는 그 끝단에 부착된 도구로 작업을 할 수 있는 장치를 의미한다. 매니플레이터는 사람을 대신해 국내외 산업에서 많이 쓰이고 있으며 2015년부터 2020년까지 국내 및 국외의 매니플레이터 판매량을 정량적으로 조사한 결과는 Fig. 1-1과 같다. [1]

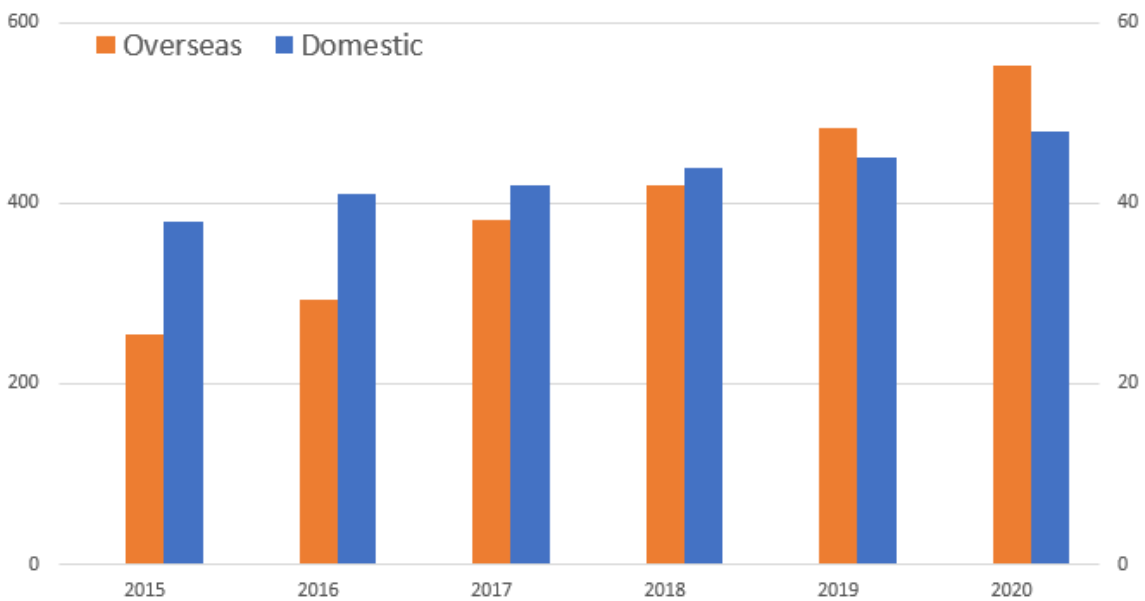


Fig. 1-1 국내외 매니플레이터 판매량 (2015년 ~ 2020년)

최근에는 4차 산업혁명의 핵심 기술인 인공지능, 빅데이터, 사물인터넷과 같은 기술들이 제조 산업 전반으로 적용되고 있으며 매니플레이터의 경우 자동화 및 무인화 기술을 적용하여 생산 효율을 최대화하는 연구가 활발히 진행되고 있다.

매니플레이터에 부착된 다양한 센서들은 실시간으로 데이터를 수집하고 분석하여 구동 상태 및 제어량을 파악하고 전체 공정의 작업 속도와 양을 조정해 생산을 관리한다. 하지만 센서에 결함이 생길 경우, 데이터 오류로 인한 매니플레이터의 오작동으로 공정 운영에 경제적 손실이 발생하며, 심한 경우에는 인명피해로 이어질 수 있다.

일반적으로 전문가가 일정 주기를 가지고 센서의 고장 여부를 확인하고 관리하지만 점검 시기를 놓치거나 고장을 발견하지 못하는 경우도 있으며, 데이터를 수집하기 위한 센서의 개수가 많아 이를 모두 확인하는 것은 어렵다. 따라서 자체적으로 임베디드 시스템에서 센서의 고

1) 신체의 상부 즉 손, 팔, 어깨를 지칭한다.

장 유무를 실시간으로 파악하고 빠른 대응을 위해 센서의 고장 진단 및 유형을 판별하는 것이 필요하다. [2]

상기 문제점을 타개하기 위해, 고장 진단, 예지, 복구와 관련된 연구들은 지속적으로 수행되어 왔으며 최근 10년간 관련 논문에 대한 정량적인 분석을 수행한 결과는 Fig. 1-2와 같고, 매니플레이터에 대해 고장 진단 관련 연구를 수행하였던 결과만 한정하면 Fig. 1-3과 같다.

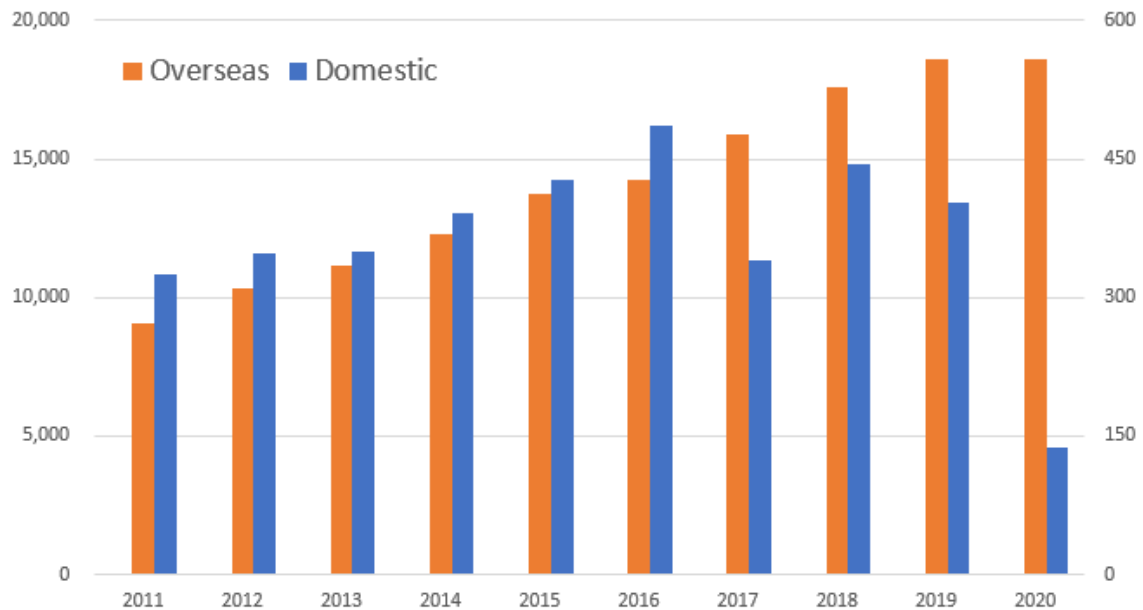


Fig. 1-2 국내외 고장 진단 관련 논문 편수 (2011년 ~ 2020년)

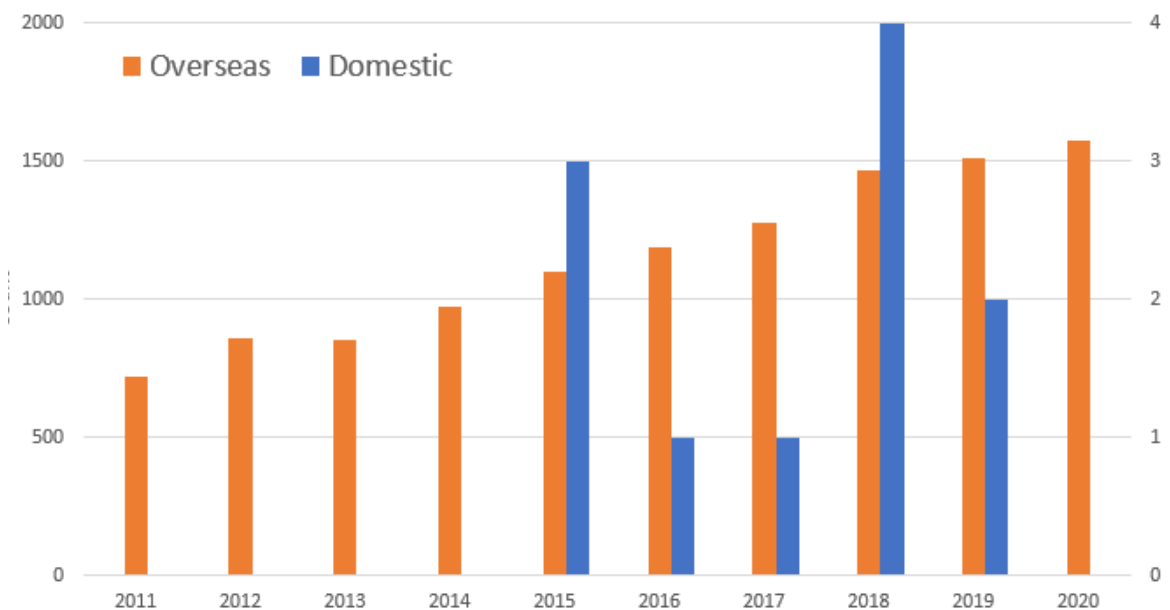


Fig. 1-3 국내외 매니플레이터 고장 진단 관련 논문 편수 (2011년 ~ 2020년)

선행 연구 분석 결과, 고장 진단과 관련된 연구는 2011년부터 현재까지 꾸준히 증가하고 있는 것을 확인할 수 있고 국외의 경우, 매니플레이터 고장 진단에 관한 연구 역시도 활발히 이루어지고 있는 것을 볼 수 있다. 하지만 국내의 경우, 국외와 비교하여 그 수가 적으며 연구가 많이 이루어지고 있지 않음을 확인할 수 있다.

고장 진단에 대한 접근방법은 크게 모델 기반 접근방법과 데이터 주도 접근방법으로 나뉜다. 첫 번째, 모델 기반 접근방법은 물리적 고장 모델을 기반으로 고장을 진단 및 예지하는 방법으로, 정확도가 높으며 적은 양의 데이터로도 고장 진단이 가능하고, 모델의 변수를 바꿔줌으로써 다양한 운영 환경에서도 적용이 가능한 장점이 있다. 그러나 모델 기반 접근방법의 단점은 시스템의 복잡도가 크게 증가하면서 모델을 완전히 구현하기 힘들고, 모델의 변수 또한 추정하기 쉽지 않다는 것이다. 또한 모델링 과정에 해당 분야 전문가의 도움이 필요해 모델을 구성하고 유지하는데 많은 비용이 든다. 이러한 요인들로 인해 현재 모델 기반 접근방법은 적용 분야가 매우 한정적이다. [3] ~ [7]

두 번째, 데이터 주도 접근방법은 데이터에서 고장과 관련된 중요한 인자를 찾아내어 통계적 방법, Machine Learning, Deep Learning 기법으로 시스템의 신뢰성, 건전성 정보를 추정하는 방법이다. Machine Learning과 Deep Learning 기법은 건전성 인자와 고장 여부의 관계를 학습을 통해 스스로 배움으로써 미래고장을 예측할 수 있다. 이 접근방법은 물리적 손상 모델을 구현하기 어려운 다변량·비선형 시스템에 적용이 가능한 장점이 있지만, 예측 모델의 훈련을 위해서는 많은 데이터가 필요하다는 단점이 있다. [8] ~ [9]

본 논문에서는 모델 기반 접근방법의 예측 불가능한 요인들을 보완하기 위해 데이터 주도 접근방법을 기반으로 한 고장 진단 및 복구 알고리즘을 제안한다.

기존 데이터 주도 접근방법의 경우, 고장 신호를 검출 및 분류하기 위해 주어진 모델을 기반으로 신호 처리 및 분석, 규칙기반 전문가시스템, 퍼지이론들을 적용한 연구가 활발히 이루어져 왔다. 하지만 이러한 방법들은 새로운 고장 유형이 발생할 경우 판별을 위한 임계값 설정과 같은 대처 알고리즘에 대한 추가 및 수정이 어렵고 작업 환경의 복잡성으로 인해 일부의 방법들만 적용이 가능하다. 최근에는 이러한 단점을 보완하기 위해 RNN(Recurrent Neural Network), LSTM(Long Short Term Memory), GRU(Gated Recurrent Unit), DBN(Deep Belief Network), RBM(Restricted Boltzmann Machine)과 같은 Deep Learning 기법이 고장 진단시스템에 적용되고 있으며, 성능 또한 우수하여 여러 분야에서 응용되고 있다. [10] ~ [15]

Ergin Kilic은 펌프의 회전속도, 유압 액추에이터의 위치 및 평균속도를 인자로 사용하는 RNN을 이용하여 가변 펌프 유압시스템의 실린더 챔버 압력을 예측하였다. 제안된 접근 방법은 소프트 센서 역할을 하여 전기 유압 시스템의 총 압력 센서 수를 절반으로 줄이며 산업 응용 분야에서 기술적 및 경제적 이점을 얻을 수 있음을 검증하였다. [16]

Samira Zare는 신호 이미지를 입력으로 사용하는 MC-CNN구조를 제안하고 윈드터빈에 대한 자율 데이터 기반 결함 진단 알고리즘을 개발하였다. 제안된 접근방법은 자동 진단 문제를

해결하기 위한 신호 이미지의 결함 특징에 대한 학습 능력을 검증하였다. [17]

Shuhui Wang은 축 피스톤 펌프의 여러 결함을 감지하기 위해 DBN 알고리즘을 제안하였다. 제안된 접근 방법은 축 피스톤 펌프의 4가지 결함 감지에 대해 SVM(Support Vector Machine), ANN(Artificial Neural Network)과 비교하여 DBN의 효율성을 검증하였다. [18]

Guangzheng Hu는 빅 데이터를 기반으로 한 산업 결함 진단을 효과적으로 처리하기 위해 DBM(Deep Boltzmann Machine)과 multi-grained scannig forest ensemble을 결합한 새로운 알고리즘을 제안하였다. 제안된 접근 방법은 오류 분류 정확도에서 DBN보다 경쟁력이 있고 gcForest보다 시간이 적게 소요되는 것을 검증하였다. [19]

1.2. 연구 목적 및 내용

본 논문에서는 3축 유압 매니플레이터의 시뮬레이션 모델을 구축하고 실제 제품과 비교하여 모델의 신뢰성을 확보한다. 그리고 신뢰성 있는 모델을 바탕으로 Deep Learning을 활용한 고장 진단 및 복구 알고리즘을 설계 후 적용하여 알고리즘의 타당성을 검토한다. 최종적으로 Labview 및 Embedded PC 등을 활용하여 실차 실험 환경을 구축하고 제안한 고장 진단 및 복구 알고리즘을 검증한다.

이를 위해서 다음과 같은 연구를 진행하였다. 3축 유압 매니플레이터의 부품을 3D Tool인 CATIA를 활용해 모델링하고 Simulink와의 연동성을 위해 SolidWorks를 활용하여 매니플레이터 부품을 조립하였다. 이후 SimMechanics를 활용하여 Multi-Body 모델링 작업을 수행하였다. 유압 회로를 분석하고 SimHydraulics를 활용하여 시뮬레이션 모델의 유압 회로를 구축하였다. 완성된 기구부와 유압부를 결합해 통합 시뮬레이션 모델을 구축하고 시뮬레이션 모델과 실제 시스템을 비교하여 모델의 신뢰성을 확보하였다. 고장 진단 및 복구 알고리즘을 설계하기 위해 고장을 일시적 고장, 영구적 고장으로 분류하였으며, 일시적인 고장에 대해 Kalman Filter를 설계하여 신호가 복구됨을 확인하였다. 영구적 고장에 대해 NN, RNN, LSTM, GRU를 활용하여 알고리즘을 설계하였고 시뮬레이션 모델과 실제 시스템에 설계된 알고리즘을 적용하여 센서 고장 발생 시, 고장을 진단하고 센서의 신호를 복구할 수 있음을 검증하였다. 연구 목적을 달성하기 위한 구체적인 연구 내용은 다음과 같다.

2장은 유압 매니플레이터에 대한 이론적 배경 및 고장 진단 및 복구 알고리즘에 대해 기술하였다. 3축 유압 매니플레이터의 기준 좌표계를 설정하고 Denavit - Hartenberg Table을 생성하였다. Forward Kinematics, Inverse Kinematics 이론을 바탕으로 매니플레이터 시스템의 기준 경로를 생성하였다. Motor, Pump, Hydraulic Motor, Servo-valve, Cylinder로 이루어진 유압 시스템을 분석하였다. Kalman Filter, NN, RNN, LSTM, GRU 기법을 분석하였다.

3장은 본 논문에 사용되는 3축 유압 매니플레이터의 시뮬레이션 모델 구축 및 시뮬레이션 결과에 대해 기술하였다. CATIA를 이용하여 3축 유압 매니플레이터의 각 부품을 모델링하고 무게 중심, 관성 모멘트에 관한 정보를 획득하였다. SolidWorks를 이용하여 부품을 결합하고 시뮬레이션을 위해 SimMechanics를 활용하여 Multi-Body 모델링을 진행하였고, 시스템의 기구부를 구성하였다. SimHydraulics를 활용하여 시스템의 유압부를 구성하고 기구부와 연결하여 통합 시뮬레이션 모델을 구축하였다. 시뮬레이션 모델의 신뢰성 확보를 위해 시뮬레이션 모델과 실차 모델에 동일 제어 입력을 인가하여 시뮬레이션 모델의 출력과 실차 모델의 출력을 비교하였다. 일시적 고장에 대해 Kalman Filter를, 영구적 고장에 대해 NN, RNN, LSTM, GRU를 활용한 알고리즘을 설계하고 시뮬레이션 모델에 적용하여 알고리즘을 검증하였다.

4장은 실험 결과 및 결론에 대해 기술하였다. 3장에서 구축된 시뮬레이션 모델을 통하여 검증된 알고리즘을 실제 매니플레이터 시스템에 적용하였다. 제안된 알고리즘은 매니플레이터에 장착된 센서 고장 발생 시 고장을 진단하고 센서의 신호를 복구할 수 있음을 검증하였다.

2. 이론적 배경

2.1. 유압 매니플레이터

본 논문에 사용된 유압 매니플레이터는 전기 모터, 유압 펌프, 유압 모터, 유압 실린더, 서보 밸브로 이루어져 있으며 구조는 Fig. 2-1과 같다.

매니플레이터의 에너지 흐름은 다음과 같다. 유압 모터에서 전기 에너지를 기계 에너지로 변환하고 유압 펌프에서 기계 에너지를 유체 에너지로 변환한다. 변환된 유체 에너지는 서보 밸브를 거쳐 각 액추에이터에서 사용된다.

첫 번째 서보 밸브는 유압 실린더를 제어하며 실린더는 직선 운동을 하고 세 번째 관절의 회전 운동을 만들어낸다. 두 번째, 세 번째 서보 밸브는 유압 모터를 제어하며 모터는 회전 운동을 하고 첫 번째, 두 번째 관절을 움직인다. 세 개 관절의 복합 동작으로 매니플레이터는 원하는 작업을 수행한다.

본 논문에서는 유압 매니플레이터를 대상으로 기구 및 유압 시스템에 관하여 모델링하고, 고장 진단 및 복구에 관한 시뮬레이션 및 실차 테스트를 진행하였다.

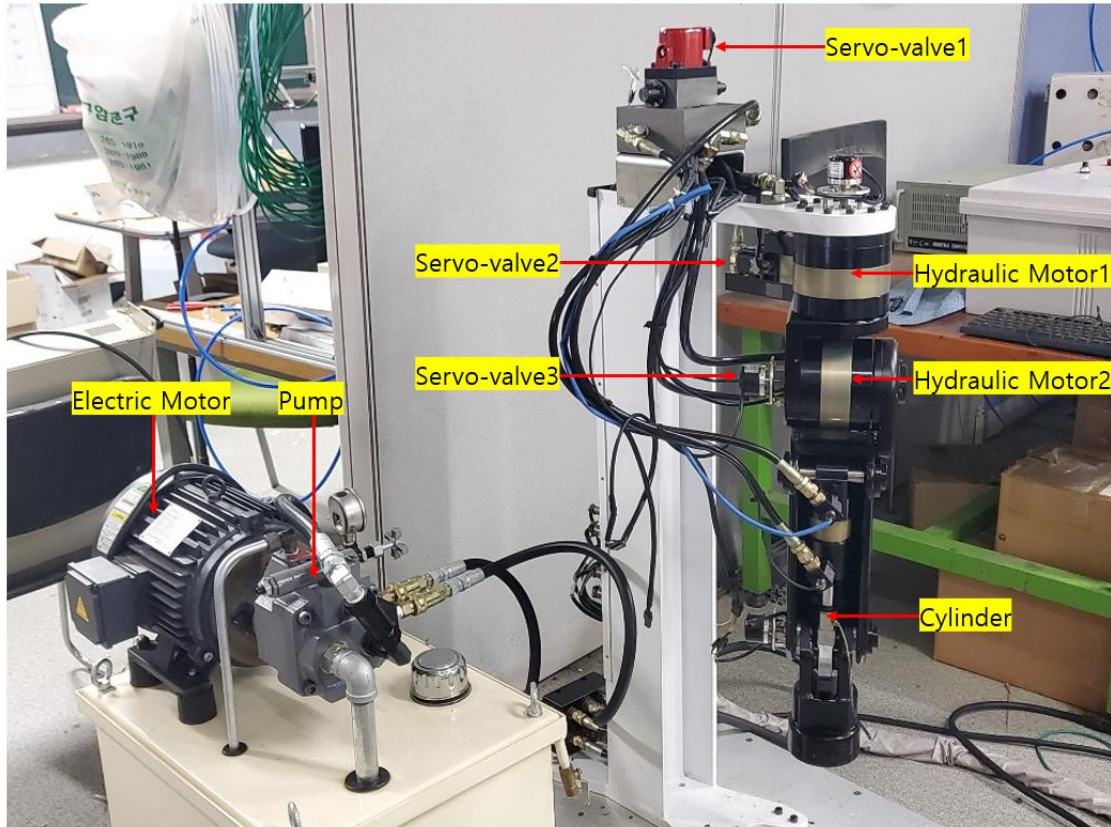


Fig. 2-1 매니플레이터 구조

2.1.1. 정기구학

정기구학은 주어진 각 관절의 각도 값으로 끝단의 위치와 방향을 계산하는 방법이다. 기준 좌표계에서 바라본 끝단의 위치를 구하기 위해 식 (1)에 나타낸 변환 행렬을 사용하였다.

$$T_{i-1}^i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

여기서 a_i 는 Z축 사이의 거리, α_i 는 Z축 간의 각도, d_i 는 X축 사이의 거리, θ_i 는 X축 간의 각도를 의미하며 일반적인 링크의 도식도는 Fig. 2-2와 같다.

최종적으로 기준 좌표계에서 바라본 끝단의 위치는 각 좌표계의 변환 행렬을 모두 곱함으로써 얻을 수 있으며 식 (2)와 같다.

$$T_0^n = T_0^1 \times T_1^2 \times T_2^3 \times \dots \times T_{n-1}^n \quad (2)$$

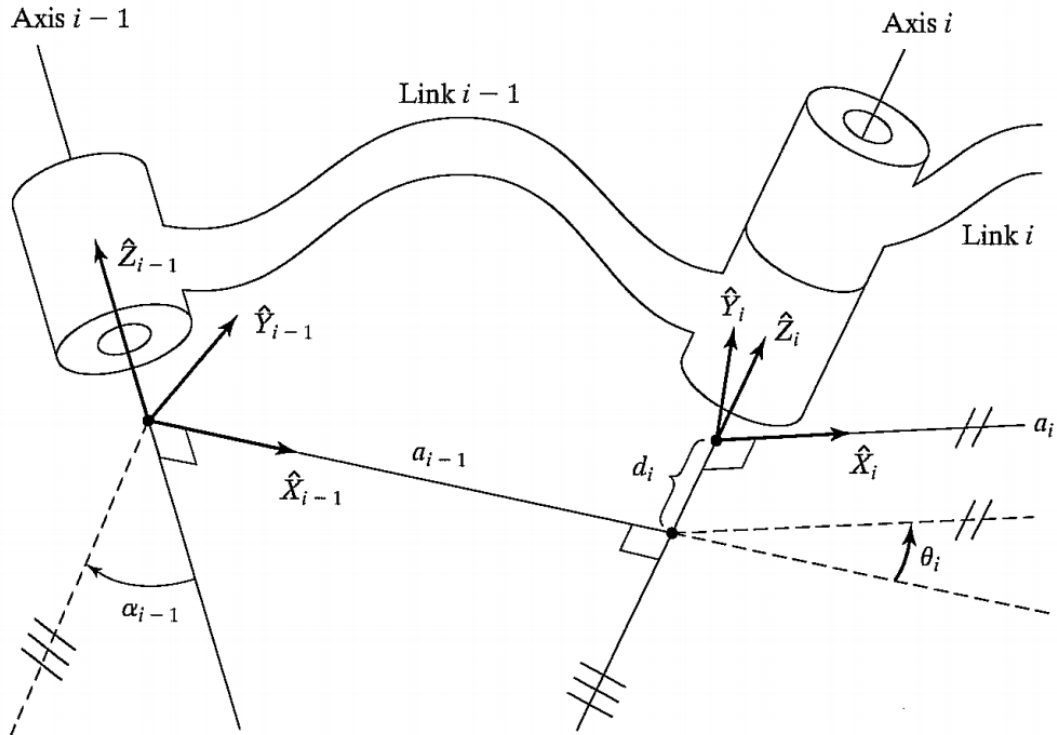


Fig. 2-2 표준 링크 도식도

본 논문에 사용된 유압 매니플레이터의 기준 좌표계를 설정한 모습은 Fig. 2-3과 같다.

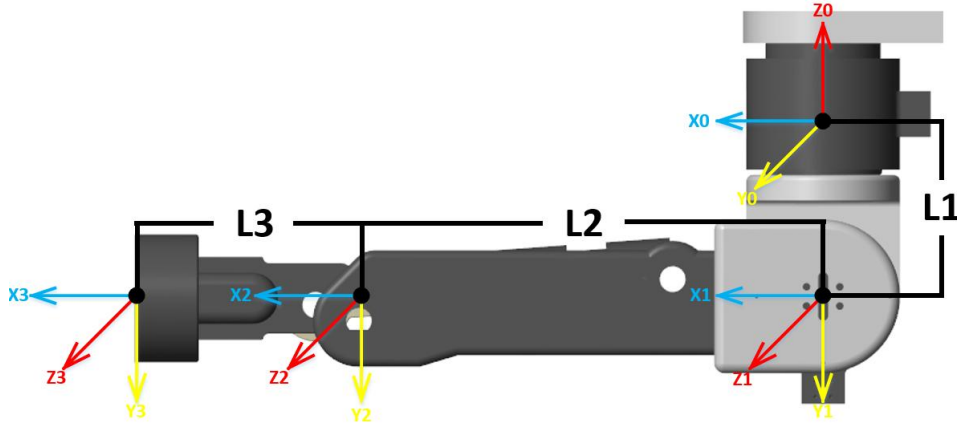


Fig. 2-3 매니플레이터 기준 좌표계

기준 좌표계에서 바라본 끝단의 좌표를 알아내기 위해서 각 좌표계 간의 변환 행렬을 구해야 한다. 이를 위해 Fig. 2-3을 기반으로 한 Denavit - Hartenberg Table을 작성하였으며, 결과는 Table 2-1과 같다.

Table 2-1 Denavit - Hartenberg Table

i	a_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	L_1	θ_1
2	L_2	0	0	θ_2
3	L_3	0	0	θ_3

Table 2-1을 활용하여 각 좌표계의 변환 행렬을 구하면 식 (3), (4), (5)와 같고 기준 좌표계에서 바라본 끝단의 방향과 위치는 식 (6)과 같다.

$$T_0^1 = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & -L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_1^2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_2^3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$P = T_0^1 T_1^2 T_2^3 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_{23}C_1 & -S_{23}C_1 - S_1 & C_1(L_3C_{23} + L_2C_2) \\ S_1C_{23} & -S_{23}S_1 & C_1(L_3C_{23} + L_2C_2) \\ -S_{23} & -C_{23} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

위의 결과를 기반으로 Home Position($\theta_1 = 0^\circ$, $\theta_2 = 0^\circ$, $\theta_3 = 0^\circ$, $\theta_4 = 0^\circ$)에서 유압 매니퓰레이터 끝단의 위치를 MATLAB을 이용하여 계산하였고, 그 결과는 Table 2-2와 같다. MATLAB 계산 결과와 비교하기 위해 3차원 공간 좌표 상에서 벡터 기법으로 끝단의 위치를 계산하였으며 결과는 Table 2-3과 같다.

Table 2-2 끝단 위치 계산 결과 (MATLAB 활용)

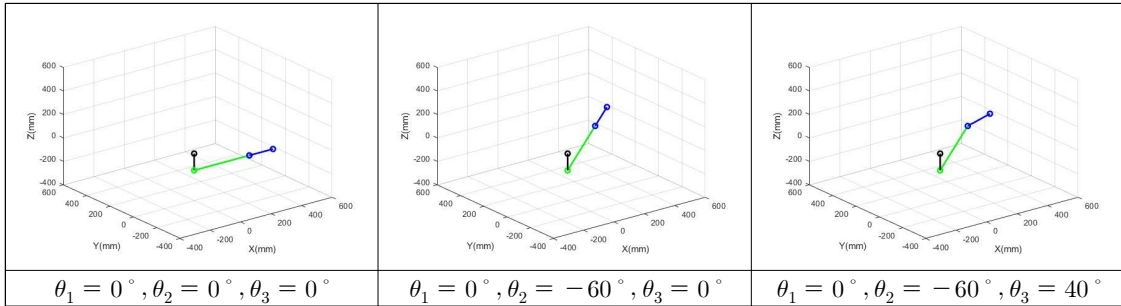
	x	y	z
1st Joint(x,y,z)	0,	0,	-l1
2nd Joint(x,y,z)	l2,	0,	-l1
End Effector(x,y,z)	l2 + l3,	0,	-l1

Table 2-3 끝단 위치 계산 결과 (벡터 기법 활용)

1st Joint(x,y,z)	(0, 0, -L ₁)
2nd Joint(x,y,z)	(L ₂ , 0, -L ₁)
End Effector(x,y,z)	(L ₂ + L ₃ , 0, -L ₁)

Home Position을 제외한 다른 각도에서도 앞서 구한 변환행렬의 타당성을 검증하였다. 관절 각도를 $(0^\circ, 0^\circ, 0^\circ)$, $(0^\circ, -60^\circ, 0^\circ)$, $(0^\circ, -60^\circ, 40^\circ)$ 3가지 Case로 분류하고 3D Plotting을 통해 확인하였으며, 그 결과는 Table 2-4와 같다.

Table 2-4 3D Plotting 결과



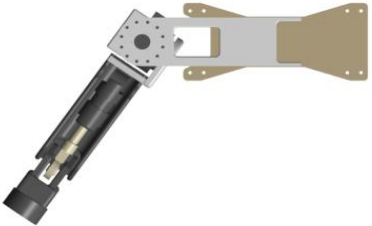
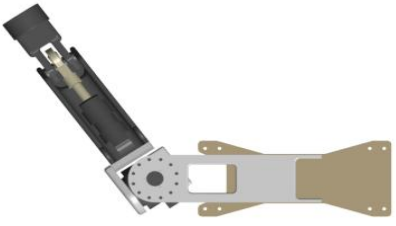
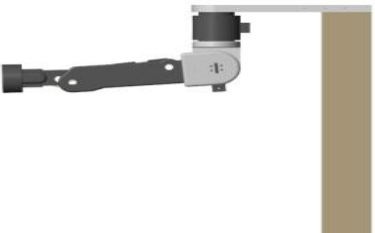

2.1.2. 역기구학

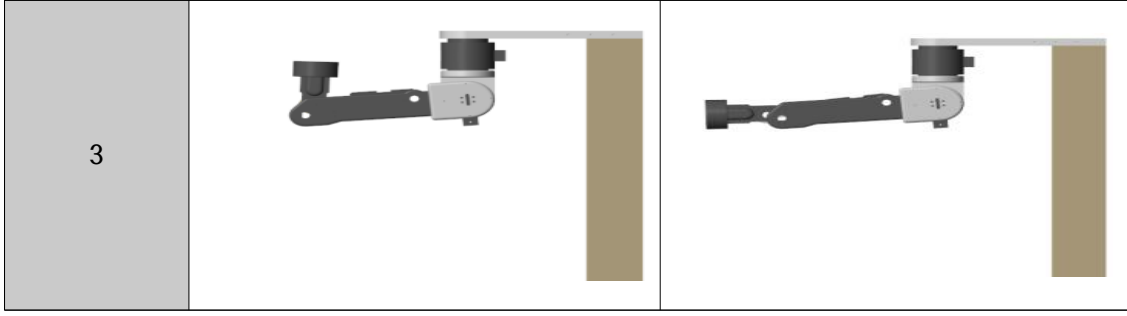
역기구학은 끝단의 방향과 위치가 주어졌을 때 각각의 관절 각도를 구하는 방법을 의미한다. 역기구학을 통해 관절 각도를 구하면 무수히 많은 해가 존재한다. 본 논문에서는 해를 줄이기 위해 3D Modeling Tool을 이용하여 각 관절의 제한 각도를 측정하였으며 그 결과는 Table 2-5, 2-6과 같다.

Table 2-5 관절 제한 각도

Joint	Min	Max
1	+55(CCW)	-55(CW)
2	5(up)	107(down)
3	-97(up)	-3(down)

Table 2-6 관절 제한 각도에 따른 매니플레이터 형상

Joint	min	max
1		
2		



역기구학의 해를 구하는 방법은 크게 기하학적 접근 방법과 변환 행렬 접근 방법 두 가지가 있다. 본 논문에서는 변환 행렬의 특성을 사용하여 해를 구하였고, 임의로 설정한 각도와 역기구학 연산을 통해 나온 각도가 같은 것만을 해로 인정하였다.

기준 좌표계에서 바라본 끝단의 위치는 각 관절의 변환 행렬을 곱한 것과 같다. 식 (6)에서 T_0^1 의 역행렬을 양변에 곱하면 식 (7)과 같고 전개하면 식 (8)과 같다.

$$[T_0^1]^{-1} \times P = [T_0^1]^{-1} \times T_0^3 = T_1^2 \times T_2^3 \quad (7)$$

$$\begin{bmatrix} n_x C_1 + n_y S_1 & o_x C_1 + o_y S_1 & a_x C_1 + a_y S_1 & p_x C_1 + p_y S_1 \\ -n_z & -o_z & -a_z & -L_1 - p_z \\ n_y C_1 - n_x S_1 & o_y C_1 - o_x S_1 & a_y C_1 - a_x S_1 & p_y C_1 - p_x S_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_{23} & -S_{23} & 0 & L_3 C_{23} + L_2 C_2 \\ S_{23} & C_{23} & 0 & L_3 S_{23} + L_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

θ_1 을 구하기 위해 두 행렬의 (3,1) 성분을 비교해 θ_1 을 구한 결과는 식 (9)와 같다.

$$n_y C_1 = n_x S_1 \rightarrow \frac{S_1}{C_1} = \frac{n_y}{n_x} = \tan \theta_1 \rightarrow \therefore \theta_1 = \text{atan2}(n_y, n_x) \quad (9)$$

S_{23} , C_{23} 를 구하기 위해 두 행렬의 (2,1) 성분 및 (2,2) 성분을 비교하면 식 (10), (11)과 같다.

$$S_{23} = -n_z \quad (10)$$

$$C_{23} = n_x C_1 + n_y S_1 \quad (11)$$

θ_2 를 구하기 위해 두 행렬의 (1,4) 성분 및 (2,4) 성분을 비교해 식 (12), (13)과 같이 S_2 , C_2 를 계산한다.

$$S_2 = \frac{(-L_1 - p_z - L_3 S_{23})}{L_2} \quad (12)$$

$$C_2 = \frac{(p_x C_1 + p_y S_1 - L_3 C_{23})}{L_2} \quad (13)$$

θ_2 를 구하면 식 (14)와 같다.

$$\therefore \theta_2 = \text{atan2}(S_2, C_2) \quad (14)$$

θ_3 를 구하기 위해 삼각함수 덧셈법칙을 이용하여 식 (15), (16)과 같이 S_3 , C_3 를 계산한다.

$$S_3 = \sin((\theta_2 + \theta_3) - \theta_2) = \sin(\theta_2 + \theta_3)\cos\theta_2 - \sin\theta_2\cos(\theta_2 + \theta_3) \quad (15)$$

$$C_3 = \cos((\theta_2 + \theta_3) - \theta_2) = \sin(\theta_2 + \theta_3)\sin\theta_2 + \cos\theta_2\cos(\theta_2 + \theta_3) \quad (16)$$

최종적으로 θ_3 를 구하면 식 (17)과 같다.

$$\therefore \theta_3 = \text{atan2}(S_3, C_3) \quad (17)$$

MATLAB/Simulink을 사용하여 제한 각도 범위 내에서 정기구학 연산을 통해 얻어진 12개의 파라미터($n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z, p_x, p_y, p_z$)를 이용해 역기구학 계산을 수행하였다. 정기구학 변환행렬로 입력된 임의의 각도와 역기구학 해를 통해 도출된 각도를 비교함으로써 구해진 해의 타당성을 검증하였다. 비교 결과는 Fig. 2-4와 같다.

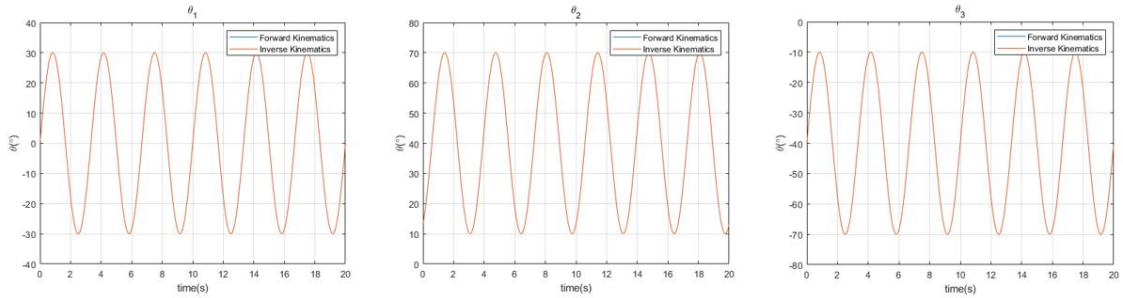


Fig. 2-4 역기구학 해 검증 결과

비교 결과 제한 각도 범위 내에서 각도가 변할 때 정기구학 변환행렬로 입력된 각도와 최종적으로 역기구학의 해를 통해 얻어진 각도가 일치함을 볼 수 있다. 따라서 앞서 구한 역기구학의 해가 타당하다 할 수 있다. [20]

2.2. 유압 시스템

매니플레이터의 유압 회로는 피스톤 펌프, 유압 모터, 유압 실린더, 서보 밸브로 구성되어 있으며 이를 도식화하면 Fig. 2-5와 같다. 서보 밸브는 방향 제어 밸브로 도식화하였다. 유압 매니플레이터의 시뮬레이션 모델 개발, 고장 진단 및 복구 알고리즘 개발 등의 연구를 진행하기 위해 유압 회로를 분석하였다.

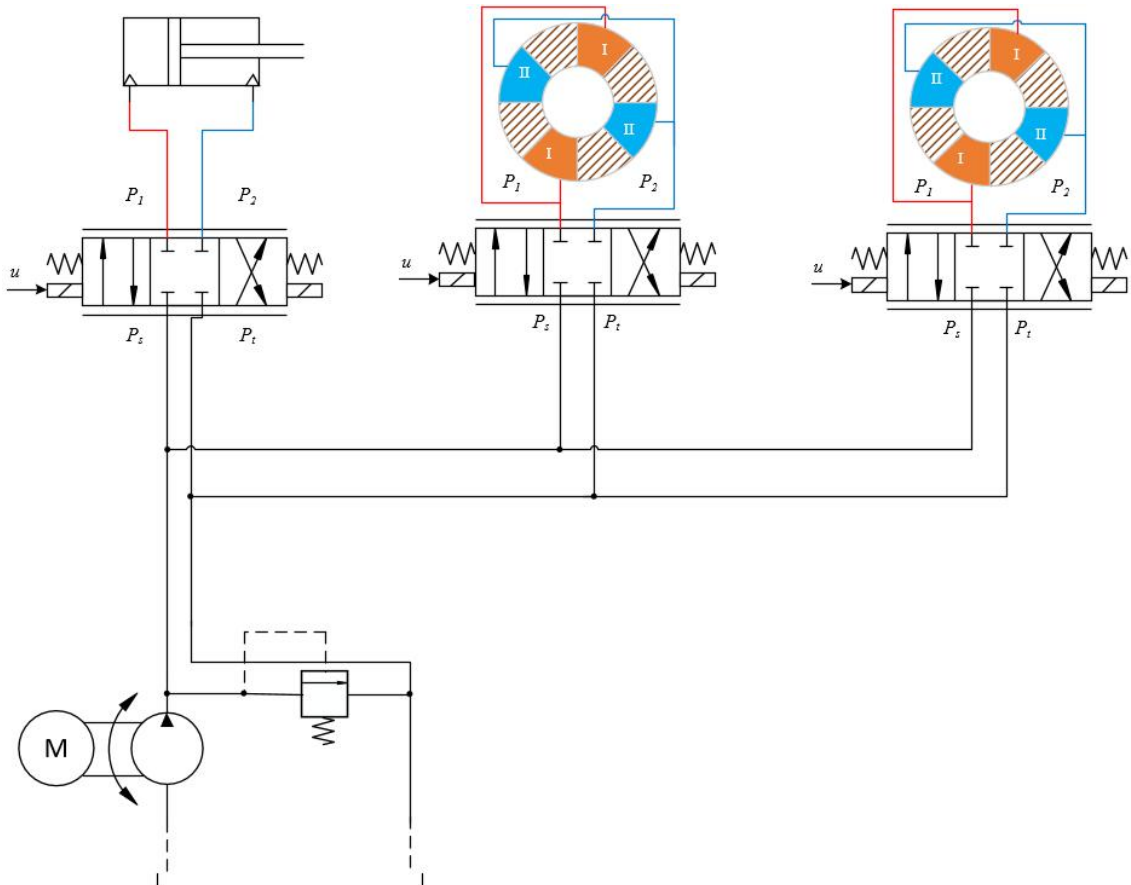


Fig. 2-5 매니플레이터 유압 회로도

2.2.1. 유압 펌프

유압 펌프는 전동기나 엔진 등에 의해 얻어진 기계적 에너지를 이용하여 유체 에너지를 생성하는 유압 장치이다. 유압 펌프는 크게 1회전 토출 유량을 조절할 수 없는 정용량형 펌프와 1회전 토출 유량을 조절할 수 있는 가변용량 펌프로 분류할 수 있다.

유압 펌프의 기구적 구조에 따른 분류는 Fig. 2-6과 같으며, 기어 펌프, 베인 펌프, 피스톤 펌프로 분류할 수 있다. 피스톤 펌프의 경우 피스톤의 배치에 따라 축 방향으로 배치된 여러 개의 피스톤이 왕복운동하며 기름을 토출하는 액시얼 피스톤 펌프와 반경 방향으로 배치된 여러 개의 피스톤이 왕복운동하며 기름을 토출하는 래디얼 피스톤 펌프로 분류된다. [21]

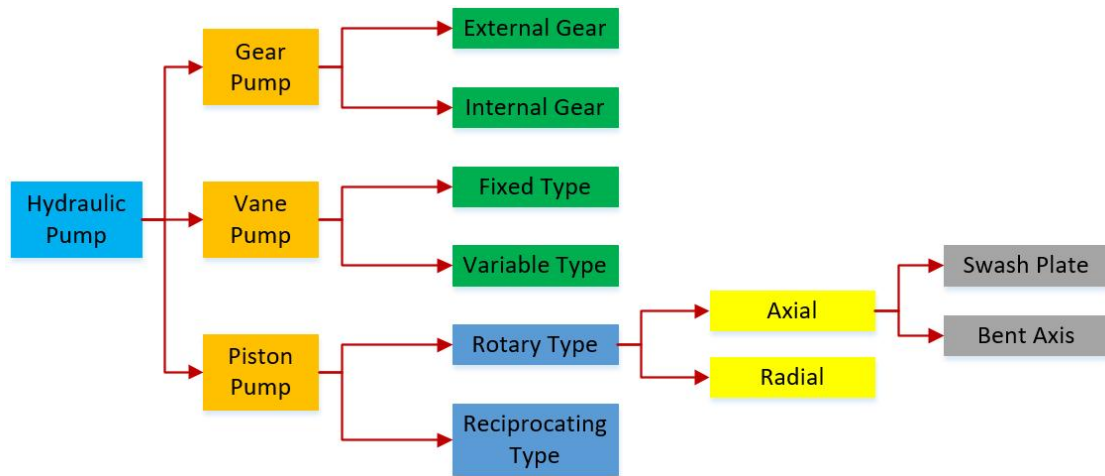


Fig. 2-6 유압 펌프 분류도

본 논문의 3축 유압 매니플레이터에는 사판식 가변용량형 피스톤 펌프가 장착되어 있으며 최대 용적은 16cc/rev이다.

피스톤 펌프는 기능에 따라 회전체, 제어, 하우징 3가지의 그룹으로 분류될 수 있다. 회전체 그룹은 엔진이나 모터의 동력에 의해 회전하면서 유량을 흡입하고 토출하는 역할을 수행하며 구동축, 실린더 블록, 피스톤, 슈, 리테이너 등으로 구성되어 있다. 제어 그룹은 유압 펌프에 인가되는 입력(파일럿 압력, 토출 압력, 전류 등)에 따라 사판을 적절하게 제어하여 시스템에서 요구되는 유량만큼만 토출하게 하며 레귤레이터, 사판, 서보 피스톤 등으로 구성되어 있다. 하우징 그룹은 메인 펌프의 외관을 제공하고 소음과 누유를 효과적으로 차단해야 하며 케이싱, 커버 등으로 구성되어 있다.

유량 토출 원리는 다음과 같다. 구동축은 모터에 의해서 구동되며 스플라인 결합에 의해 실린더 블록도 동시에 회전한다. 사판이 기울어진 만큼 실린더 블록에 장착된 피스톤은 실린더 블록과 함께 회전하면서 실린더 블록의 실린더에 대해 상대적으로 왕복운동을 하게 된다. 따라서 1개의 피스톤만 고려한다면 실린더 블록이 1회전 할 때 피스톤이 밸브 플레이트에서 이탈하는 방향(작동유를 흡입하는 행정)으로 180° 회전함으로써 밸브 플레이트 후단과 연결된 흡입, 토출 포트를 통해 작동유가 흡입, 송출된다. 펌프의 구조와 실제 사진은 Fig. 2-7, 2-8과 같다. [22], [23]

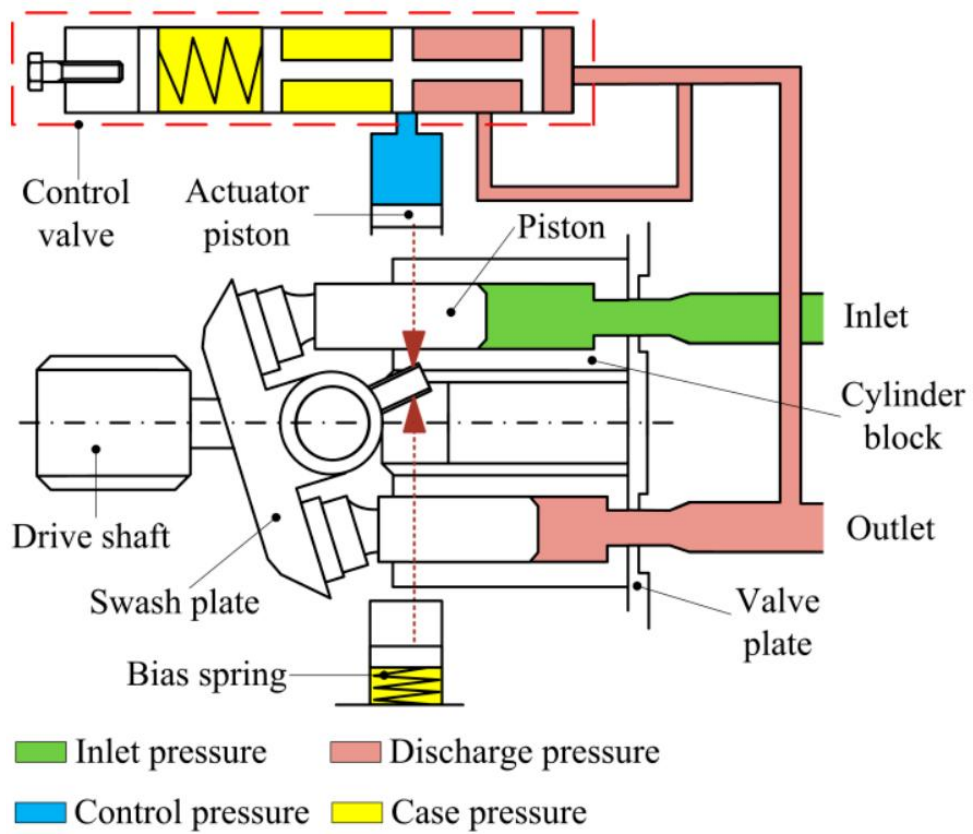


Fig. 2-7 유압 펌프 구조도



Fig. 2-8 유압 펌프 형상

본 논문에 사용된 매니퓰레이터에 장착되어 있는 펌프는 가변식이지만 컨트롤 밸브를 고정하고 사용을 하였기 때문에 송출량이 일정하므로 유압 펌프에서 송출되는 유량은 식 (18)과 같으며 장착된 펌프의 세부 제원은 Table 2-7과 같다.

$$Q = Dw\eta_v \quad (18)$$

Table 2-7 유압 펌프 제원

Max Displacement [cc/rev]	16
Max Pressure [bar]	210
Max Speed [rpm]	1800
Min Speed [rpm]	600
Weight [kg]	15

2.2.2. 유압 모터

유압 액추에이터는 크게 실린더 액추에이터와 회전형 액추에이터가 있다. 실린더는 다시 피스톤과 램형으로 나누어지며, 회전형은 기어, 베인, 피스톤 액추에이터로 나뉘며, 유압 액추에이터의 구조에 따른 분류는 Fig. 2-9와 같다. 회전형 액추에이터는 유압 모터라고도 불린다.

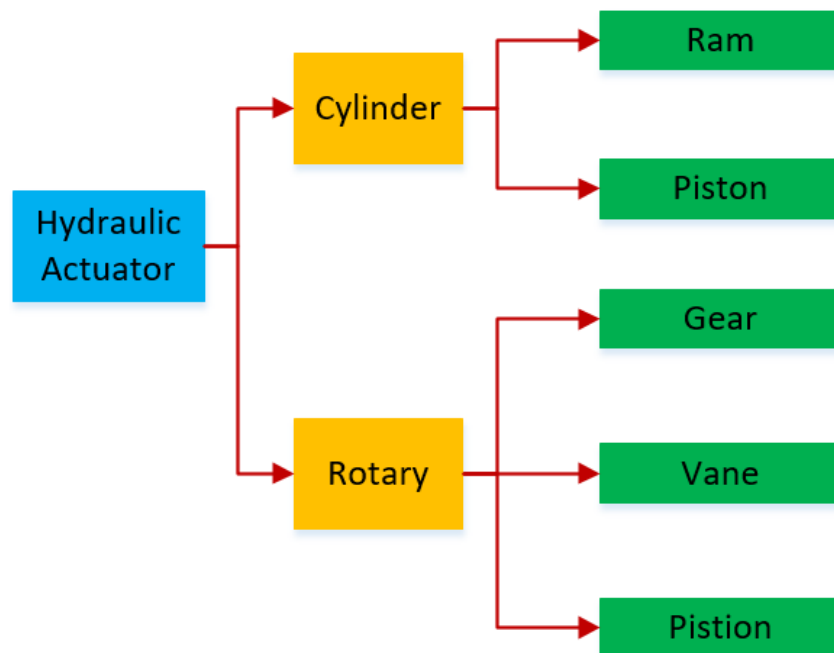


Fig. 2-9 유압 액추에이터 분류도

유압 모터는 유체 에너지를 회전시켜 기계에너지로 변환시키는 장치이며, 구조나 크기가 유압 펌프와 유사하다. 유압 펌프는 작동 중에 유체를 송출구로 밀어내지만 그에 비해 유압 모터는 작동 유체가 유압 모터 내의 회전 요소를 밀어 유압 모터 축을 회전시킨다는 점에서 차이가 있다.

유압 모터는 유압 펌프와 유사하며 기어 모터, 베인 모터, 피스톤 모터 3가지로 분류할 수 있다. 각 모터의 작동 원리는 다음과 같다. 기어 모터는 공급된 유압이 기어 톱니면에 작용하여 토크를 발생하고 출력축을 회전시키며 작동한다. 베인 모터는 편심이 있는 하우징으로 구성되며 하우징에서 베인이 들어오고 나가는 로터가 작동한다. 베인에 가해지는 유압의 불균형으로 생성된 힘 차이로 인해 로터가 한 방향으로 회전하면서 작동한다. 마지막으로 피스톤 모터는 유압이 크랭크 샤프트의 유체 경로를 통해 각 피스톤으로 공급되고 결합된 구동축이 피스톤에 의해 압력을 받으면서 회전하면서 작동한다. [24], [25]

본 논문의 3축 유압 매니플레이터에는 베인 모터가 장착되어 있으며 최대 토크는 400Nm이다. Fig. 2-10은 베인 모터의 구조를 보여준다. 유압 펌프에서 토출된 유체는 2개의 입구 포트로부터 내부로 유입하고, 그 압력은 로터의 베인에 작용하여 토크가 발생하고 로터를 화살표 방향으로 회전시킨다. 고압 유체가 유입하는 입구 포트는 로터 중심축에 대하여 대칭 위치에 있기 때문에 로터에 작용하는 반지름 방향의 힘은 상쇄되어 평형을 이룬다. [26]

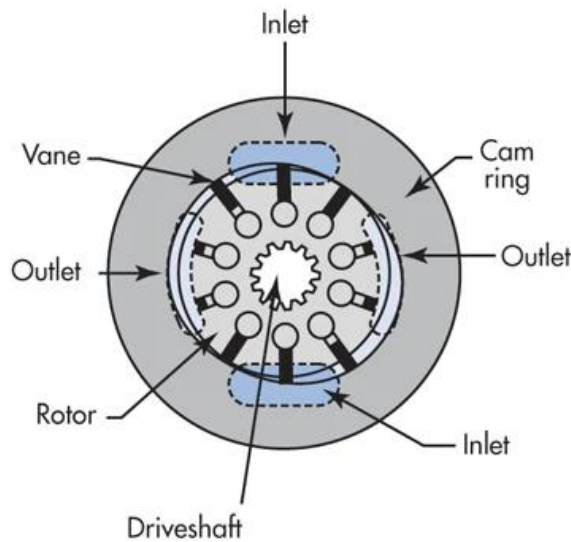


Fig. 2-10 베인 모터 구조도

베인 모터의 일반적인 회로는 Fig. 2-11과 같다.

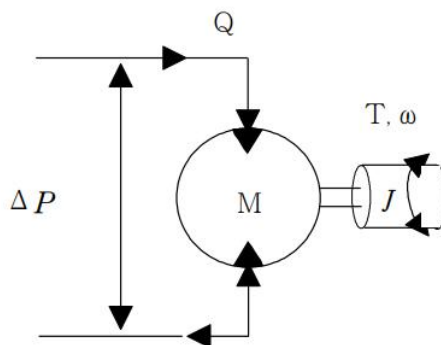


Fig. 2-11 베인 모터 회로도

유압 모터의 체적은 식 (19)와 같다.

$$V = D_m \theta \quad (19)$$

여기서 D_m 는 1회전당 배제 용적, θ 는 회전각이다. 모터의 유량은 배제 용적과 회전각으로 표현되며 식 (20)과 같다.

$$Q = \frac{d}{dt}(D_m \theta) = D_m w \quad (20)$$

모터의 토크를 구하면 식 (21)과 같다.

$$T = \frac{Q}{w} \Delta P = D_m \Delta P \quad (21)$$

모터의 토크 손실은 식 (22)와 같다.

$$T_L = D_m (P_1 - P_2) - T_v - (T_c + T_f) \quad (22)$$

T_v 는 속도에 비례하는 유체 점성에 의한 토크 손실이고 T_c 는 쿨롱 마찰에 의한 토크 손실, 마지막으로 T_f 는 마찰 토크 손실이며 기계의 속도와 관로 압력이 포함된 함수이다. [27]

본 논문에 사용된 매니플레이터에 장착된 유압 모터의 제원은 Table 2-8과 같고, 형상은 Fig 2-12와 같다.

Table 2-8 유압 모터 제원

Max Displacement [cc/rev]	0.082
Max Pressure [bar]	210
Max Torque [Nm]	400
Rotate limit [degree]	110
Weight [kg]	3



Fig. 2-12 유압 모터 형상

2.2.3. 유압 실린더

유압 실린더와 유압 모터는 유체의 운동에너지를 기계적인 에너지로 변환시킨다는 공통점을 가지고 있지만, 유압 모터는 회전 운동으로 변환하는데 비해 유압 실린더는 직선 운동으로 변환한다. 그래서 유압 실린더는 선형 유압모터라고도 불린다.

일반적인 유압 실린더는 실린더 튜브, 피스톤, 피스톤 로드, 커버, 실 등으로 이루어져 있으며 구조는 Fig. 2-13과 같다.

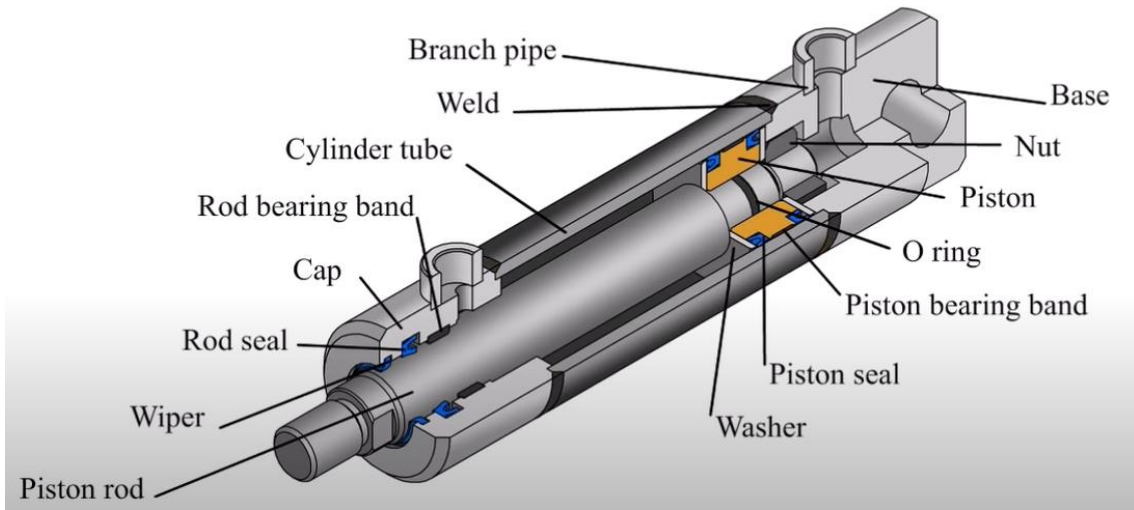


Fig. 2-13 유압 실린더 구조도

유압 실린더를 모델링하면 Fig. 2-14와 같으며 운동방정식은 식 (23)과 같다.

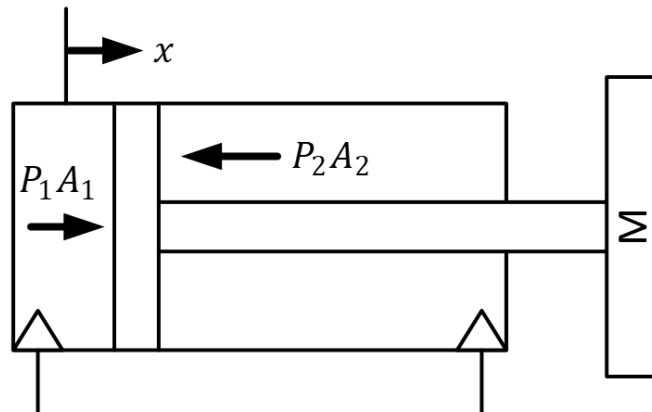


Fig. 2-14 유압 실린더 모델링 도식도

$$F = P_1 A_1 - P_2 A_2 = M \ddot{x} \quad (23)$$

본 논문에 사용된 매니플레이터에 장착된 유압 실린더는 Fig. 2-15와 같고 제원은 Table 2-9와 같다.

Table 2-9 유압 실린더 제원

Stroke [mm]	55.5
Max Pressure [bar]	210
Max Force [kN]	32
Rod Diameter [mm]	25
Head Diameter [mm]	44.2

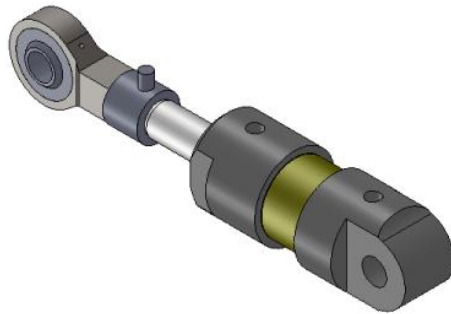


Fig. 2-15 유압 실린더 형상

2.2.4. 서보 밸브

토크 모터와 노즐-플래퍼 기구, 토크 모터와 제트 파이프 기구와 같은 전기-유압 변환기를 사용하는 고속 응답형 유압 제어밸브를 일반적으로 전기-유압 서보 밸브라 하며, 간단히 서보 밸브라 한다.

본 논문의 3축 유압 매니플레이터에 장착된 서보 밸브는 노즐-플래퍼형 방식이며 그 구조와 동작 원리를 그림으로 간략히 나타내면 Fig. 2-16, 2-17과 같다.

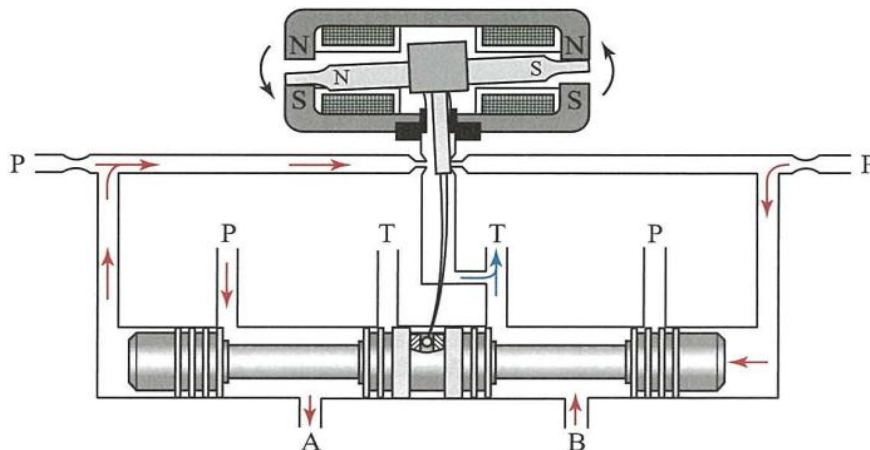


Fig. 2-16 스푼 형상 (동적 상황)

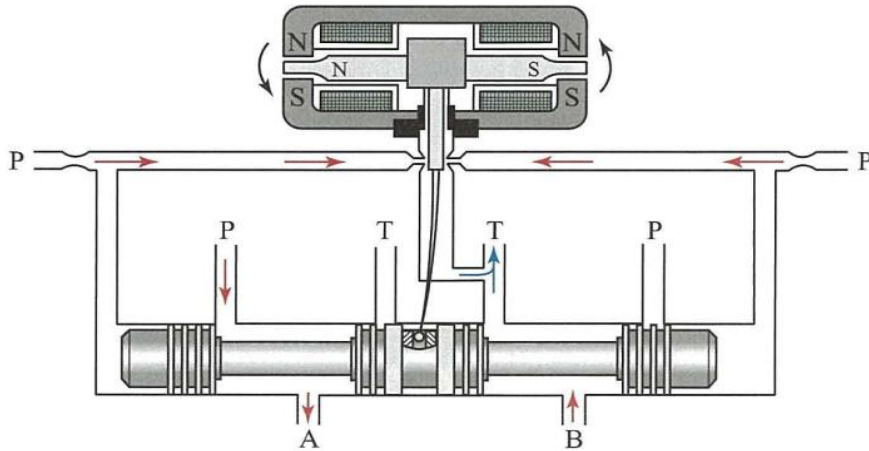


Fig. 2-17 스펴 형상 (정적 상황)

입력 전류는 코일을 통하여 흐르고, 아마추어는 자기적 극성(N, S)을 띤다. 입력 전류의 크기와 극성에 따라 발생하는 자기력으로 인하여 아마추어에 토크가 발생하면 아마추어는 기울어지게 된다.

좌·우 노즐 사이에 위치한 플래퍼는 토크 모터의 아마추어와 일체로 되어있기 때문에 아마추어의 경사각 변화에 따라 플래퍼와 노즐 사이의 틈새가 달라지며, 좌·우 노즐의 배압을 변화시킨다. 그러므로 플래퍼에 가까운 쪽의 노즐 배압은 높아지고 먼 쪽의 노즐 배압은 낮아진다. 좌·우 노즐 배압은 스펴의 양 끝으로 전달되고, 이 때 스펴 양 끝에 작용하는 압력의 차에 의하여 스펴이 움직인다. 노즐-플래퍼형 서보 밸브는 이러한 원리로 입력 전류에 비례하는 스펴의 변위 및 밸브 열림을 유지할 수 있으며, 밸브 열림에 따라 유량이 제어된다.

Fig. 2-16은 토크 모터의 코일에 전류가 가해져 스펴이 움직이고 있을 때의 밸브 작동을 보여준다. 스펴의 변위는 스펴과 플래퍼를 연결하는 금속 봉 스프링인 기계식 피드백 기구에 의해 힘으로 변환되어, 아마추어에 발생한 토크와 정반대의 토크를 발생시켜, 플래퍼를 중립 위치 부근으로 되돌려놓는다.

Fig. 2-17은 스펴이 목표로 하는 위치에 정지하면서 플래퍼도 중립 위치 부근에 정지했을 때를 보여준다. [26]

밸브를 통과하는 유량은 식 (24)로 계산된다.

$$Q = C_d A \sqrt{\frac{2\Delta P}{\rho}} \quad (24)$$

본 논문의 3축 유압 매니플레이터에 장착된 서보 밸브는 2종류이며 Fig. 2-18은 유압 실린더를 제어하는 서보 밸브이며, Fig. 2-19는 유압 모터를 제어하는 서보 밸브이다.



Fig. 2-18 서보 밸브 형상 (유압 실린더 제어)



Fig. 2-19 서보 밸브 형상 (유압 모터 제어)

2.3. 제어기

제어기는 센서로부터 필요한 입력들을 받아 매니퓰레이터의 실제 동작과 원하는 동작이 일치하도록 제어 모터 또는 액추에이터에 구동 신호를 주는 역할을 한다. 특히 용접, 제조와 같은 정밀작업을 하는 매니퓰레이터에 제어기와 센서는 필수적이다. 따라서 제어 성능을 높이기 위해서는 제어 입력인 센서의 신호가 정확해야하며, 센서 고장 발생 시 고장을 진단하고 센서 신호를 복구하는 기술이 필요하다.

매니퓰레이터 제어 기법은 PID Control, Sliding Mode Control, Backstepping Control, Impedance Control, Fuzzy Control 등 무수히 많다. 본 논문에서는 하중이 많이 걸리지 않는 Position Tracking 상황만을 고려하였으며 PID Control, Sliding Mode Control 기법을 적용하였다.

2.3.1. PID 제어

비례-적분-미분 제어는 실제 응용분야에서 가장 많이 사용되는 대표적인 형태의 제어 기법이다. PID 제어기는 기본적으로 피드백 제어기의 형태를 가지고 있으며, 제어하고자 하는 대상의 출력 값을 측정하여 이를 원하고자 하는 참조 값 또는 설정 값과 비교하여 오차를 계산한다. 이 오차 값을 이용하여 제어에 필요한 제어 값을 계산하는 구조로 되어 있다.

표준적인 형태의 PID 제어기는 아래의 식 (25)와 같이 세 개의 항을 더하여 제어 값을 계산하도록 구성이 되어 있다. 일반적인 PID 제어기의 블록 다이어그램은 Fig. 2-20과 같다.

$$MV(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt} \quad (25)$$

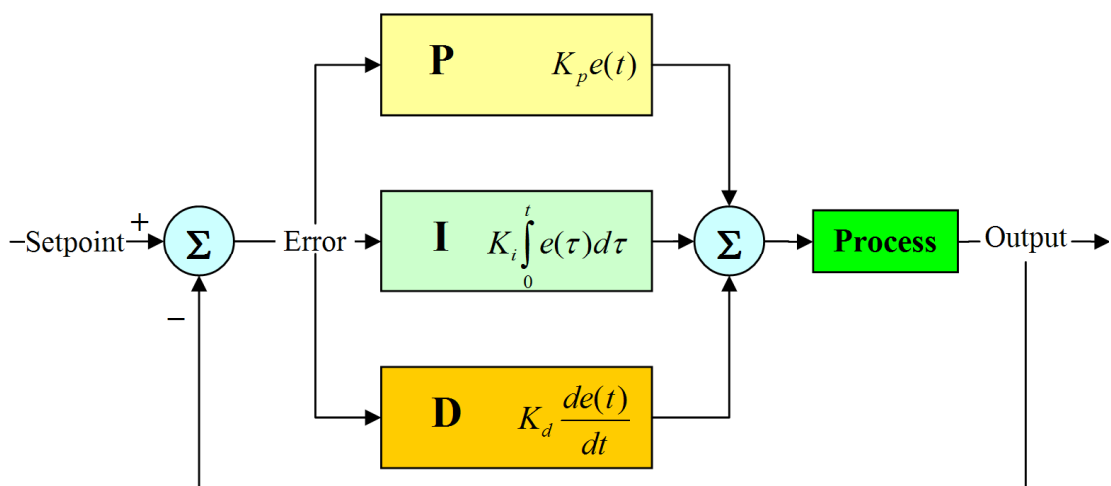


Fig. 2-20 PID 제어 블록 다이어그램

이 항들은 각각 오차 값, 오차 값의 적분, 오차 값의 미분에 비례하기 때문에 비례-적분-미분 제어기라는 명칭을 가진다. 이 세 개의 항들의 직관적인 의미는 다음과 같다. 비례항은 현재 상태에서의 오차 값의 크기에 비례한 제어 작용을 한다. 적분항은 정상상태의 오차를 없애는 작용을 한다. 마지막으로 미분항은 출력 값의 급격한 변화에 제동을 걸어 오버슈트를 줄이고 안정성을 향상시킨다.

PID 제어기는 위와 같은 표준식의 형태로 사용하기도 하지만, 경우에 따라서는 약간 변형된 형태로 사용하는 경우도 많다. 예를 들어, 비례항만을 가지거나, 비례-적분, 비례-미분항만을 가진 제어기의 형태로 단순화하여 사용하기도 하는데, 이 때 각각 P, PI, PD 제어기라고 불린다.

2.3.2. Sliding Mode 제어

슬라이딩 모드 제어는 슬라이딩 평면을 이용하여 외부 외란 및 파라미터의 불확실성에 대해 강인함을 가지는 제어 기법이다. 비선형 방정식의 일반적인 형태는 식 (26)과 같다.

$$\dot{x} = f(x,t) + g(x,t)u \quad (26)$$

여기서, $x \in R^n$ (상태변수 x 는 n 차원 벡터)는 시스템의 상태 벡터이고, $f \in R^n$, $g \in R^n \times R$ 이며 $u \in R$ 은 시스템에 인가되는 제어 입력이다.

가변 구조 제어기는 일반적으로 다음과 같은 두 단계에 의해서 설계된다. 첫 번째로 식 (27)과 같이 원하는 특성을 갖는 슬라이딩 평면을 설계한다.

$$S = \{x \in R^n | s(x) = 0\} \quad (27)$$

두 번째로 식 (28)과 같이 시스템의 상태 벡터들을 슬라이딩 평면상에 위치하도록 하는 제어 입력을 구성한다.

$$u = \begin{cases} u^+(x) & \text{when } s(x) > 0 \\ u^-(x) & \text{when } s(x) < 0 \end{cases} \quad (28)$$

시스템의 상태 벡터들이 슬라이딩 평면에 도달하게 되면 시스템은 불확실성과 외란에 대해 강인함을 가지게 된다. 그러므로 상태 벡터를 슬라이딩 평면으로 도달하게 하는 조건을 결정하는 것이 중요하다. 이러한 조건을 도달 조건이라 부른다. 도달 조건을 구하기 위해 식 (29)와 같은 Lyapunov function candidate를 정의한다.

$$V(x,t) = \frac{1}{2} s^2(x) \quad (29)$$

여기서 $s(x)$ 는 스위칭 함수를 의미하며 $x \in R^n$ 은 앞서 언급한 시스템의 상태 벡터이다. 슬라이딩 모드를 존재하게 하는 도달 조건은 다음 식 (30)과 같다.

$$\dot{V}(x,t) = s(x)\dot{s}(x) < 0 \quad \text{for} \quad x \in R^n - S \quad (30)$$

x 가 슬라이딩 평면($s(x)$)을 제외한 R^n 공간상의 어떤 위치라면 $\dot{V}(x,t) < 0$ 이므로 $s(x) = 0$ 인 슬라이딩 평면으로 간다. 식 (29)는 양수이므로 V 는 양수이다. 식 (31)과 같이 V 는 시간이 흐르면 감소하게 되고 결국 소멸된다.

$$\dot{V}(x,t) = V(x,t+\Delta t) - V(x,t) < 0 \rightarrow V(x,t+\Delta t) < V(x,t) \quad (31)$$

일반적으로 가변 구조 제어 시스템의 동특성은 도달 모드와 슬라이딩 모드로 이루어진다. 원하는 슬라이딩 모드 동특성은 적절한 스위칭 함수 $s(x)$ 를 설계함으로써 결정된다. 또한 도달 모드 설계 시 원하는 동특성뿐만 아니라 슬라이딩 모드를 얻을 수 있는 제어 입력을 결정하는 것도 중요하다.

다음 식 (32)와 같은 도달 법칙을 많이 사용한다.

$$\dot{s}(x) = -K \text{sign}(s), \quad K > 0 \quad (32)$$

식 (32)와 같은 도달 법칙을 사용하면 $s > 0: s(+)\cdot\dot{s}(-) < 0$, $s < 0: s(-)\cdot\dot{s}(+) < 0$ 이므로 항상 $s \cdot \dot{s} < 0$ 이다. 그러므로 \dot{s} 의 선정은 적절하다. 여기서 수렴속도는 스위칭 이득 K 에 의해 결정된다. 스위칭 함수 $s(x)$ 를 시간에 대해 미분하면 다음과 같은 식을 얻을 수 있다.

$$\dot{s}(x) = \frac{\partial s}{\partial x}f(x) + \frac{\partial s}{\partial x}g(x)u = -K \text{sign}(s) \quad (33)$$

식 (33)을 제어 입력 u 에 대해 정리하면 식 (34)와 같다. [28], [29]

$$u = - \left[\frac{\partial s}{\partial x}g(x) \right]^{-1} \left[\frac{\partial s}{\partial x}f(x) + K \text{sign}(s) \right] = u_{eq} + u_s \quad (34)$$

2.4. 고장 진단 및 복구

본 논문에서는 모델 기반 접근방법의 예측 불가능한 요인들을 보완하기 위해 데이터 주도 접근방법을 기반으로 고장 진단 및 복구 알고리즘을 설계하였다.

고장의 유형은 크게 일시적 고장과 영구적 고장으로 분류할 수 있다. 이를 도식화 및 간략하게 정리한 결과는 Table 2-10, Fig. 2-21과 같다. [30] 본 논문에서는 노이즈로 인한 센서의 신호가 불규칙적인 오차를 출력할 때 일시적 고장이라고 정의하였고, 센서의 신호가 하드웨어의 문제로 영구적이고 규칙적인 오차를 출력할 때 영구적 고장이라고 정의하였다. 일시적인 고장이 발생할 경우 Kalman Filter를 이용하여 센서의 신호를 복구하였다. 영구적인 고장이 발생할 경우 NN, RNN, LSTM, GRU를 이용한 알고리즘을 설계하여 센서의 신호를 복구하였다.

Table 2-10 센서 고장 분류

정의	원인	결과
일시적 고장	Noise	불연속적인 오차 값 출력
영구적 고장	단선/단락	신호 값 0 출력
	고착	일정한 신호 값 출력(기울기 0)
	Scale 변동	신호 값 기울기 변동
	Zero-Point 변동	offset 된 신호 값 출력(기울기 일정)

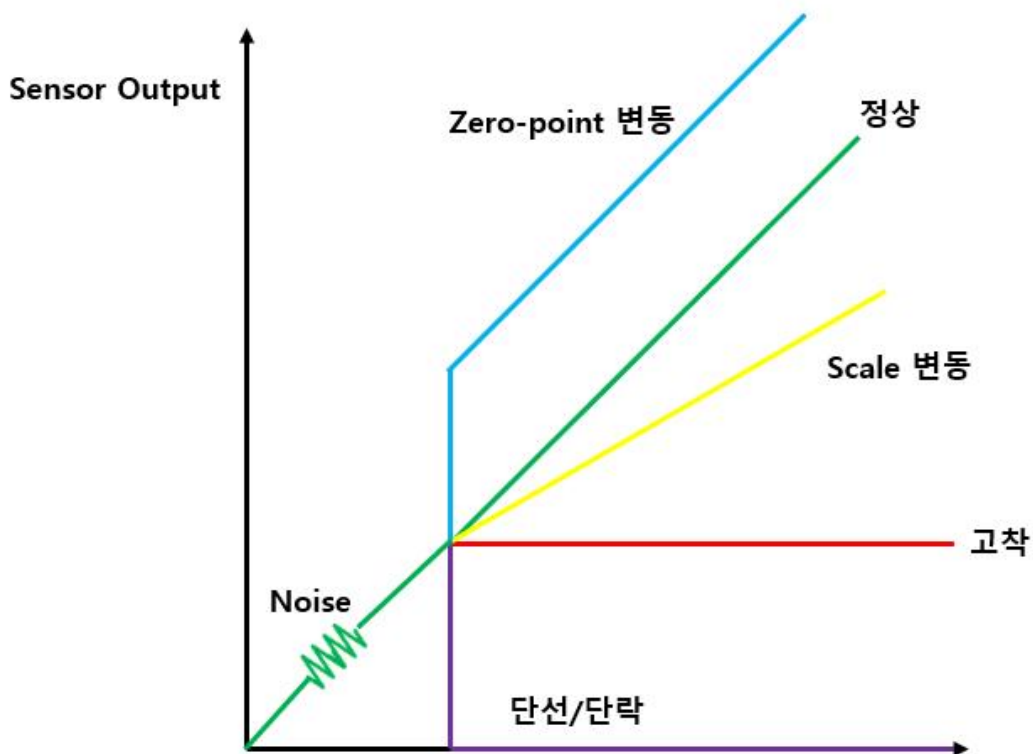


Fig. 2-21 센서 고장 분류에 따른 출력 그래프

2.4.1. 일시적 고장

본 논문에서는 센서에 외란으로 인한 일시적인 고장 발생 시 Kalman Filter를 이용하여 노이즈를 저감시키고 센서의 신호를 복구한다.

Kalman Filter는 잡음이 포함되어 있는 측정치를 바탕으로 선형 역학계의 상태를 추정하는 재귀 필터이다. Kalman Filter 알고리즘에 사용되는 변수는 Table 2-11과 같고 알고리즘을 도식화하면 Fig. 2-22와 같다.

Table 2-11 Kalman Filter 변수

Classification	Variable
External Input	z_k
Final Output	\hat{x}_k
System Model	A, H, Q, R
Internal Calculation	$\hat{x}_k^-, P_k^-, P_k, K_k$

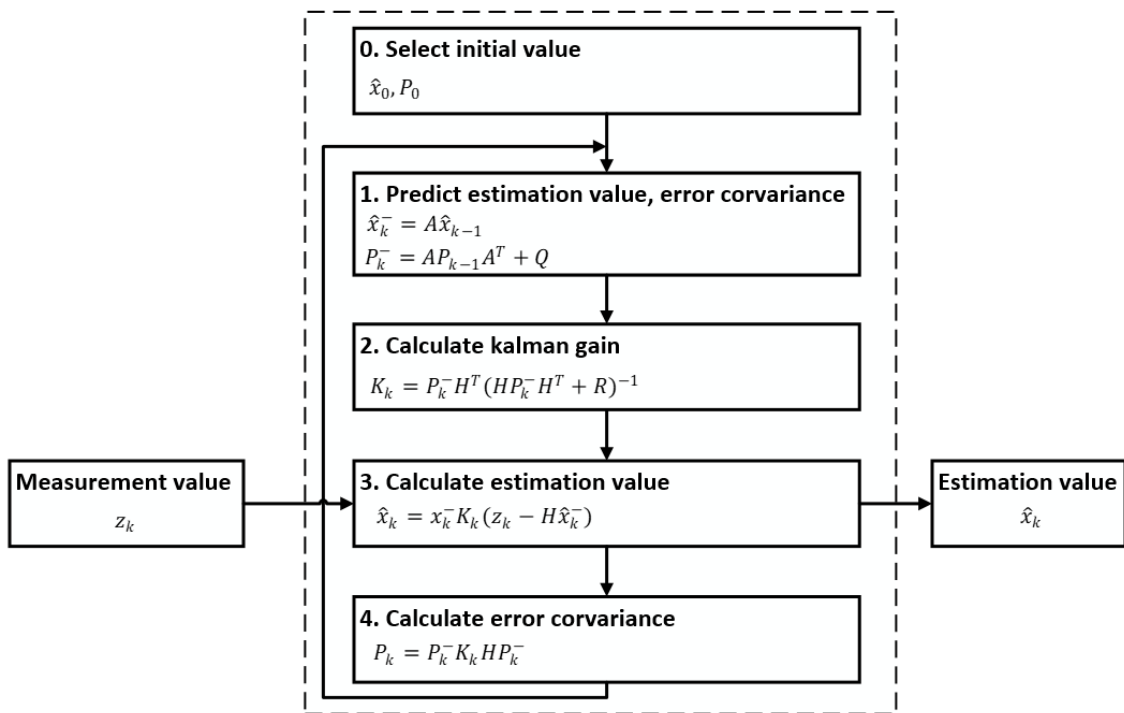


Fig. 2-22 Kalman Filter 알고리즘

Kalman Filter 알고리즘은 초깃값 설정 과정인 0단계와 예측 과정인 1단계, 추정 과정인 2~4단계로 분류할 수 있다.

0단계에서는 측정값과 예측방정식을 활용하여 초깃값인 \hat{x}_0 , P_0 를 선택한다.

예측 과정인 1단계에서는 시스템 모델 변수인 A , Q 를 활용하여 추정 값과 오차 공분산을 예측할 수 있다. A 는 시스템 행렬이며, 시스템의 운동방정식으로 구할 수 있다. Q 는 시스템 잡음인 ω_k 의 공분산 행렬이며, $(n \times n)$ 의 대각 행렬로 표현된다. 예측방정식은 식 (35), (36)과 같다.

$$\hat{x}_k^- = A\hat{x}_{k-1} \quad (35)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (36)$$

\hat{x}_k^- 는 예측된 추정 값이며, \hat{x}_{k-1} 는 현재 예측된 추정 값 바로 직전의 Kalman Filter의 출력 값이다. P_k^- 는 예측된 오차 공분산, P_{k-1} 는 현재 직전의 오차 공분산을 나타낸다.

추정 과정 중 2단계에서는 시스템 모델 변수인 H , R 를 활용하여 Kalman Gain을 계산할 수 있다. H 는 출력 행렬이며, 측정값과 상태변수로 구할 수 있다. R 은 측정 잡음인 v_k 의 공분산 행렬이며, $(m \times m)$ 의 대각행렬로 표현된다. Kalman Gain을 구하는 방정식은 식 (37)과 같다.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (37)$$

K_k 는 Kalman Gain으로 시스템 모델 변수 H , R 과 예측된 오차 공분산 P_k^- 를 이용해 계산한다. Kalman Gain은 Kalman Filter의 최종 출력인 추정 값 \hat{x}_k 계산에 필요한 가중치로 작용한다. 3단계에서는 추정 값을 계산하며 추정방정식은 식 (38)과 같다.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (38)$$

z_k 는 측정값이며 이 과정에서 추정 값을 계산한다. 4단계에서는 오차 공분산을 계산하며 방정식은 식 (39)와 같다.

$$P_k = P_k^- - K_k H P_k^- \quad (39)$$

P_k 는 오차 공분산으로 칼만 이득 K_k 와 예측 오차 공분산 P_k^- 를 계산하는데 사용된다. 오차 공분산은 추정방정식에서 추정된 값과 측정된 값의 차이로써 추정 값의 정확도를 판단하는데 사용되며, 오차 공분산에 따라 시스템 모델과의 추정 오차를 알 수 있다.

Kalman Filter 알고리즘은 0~4단계로 이루어진 이 과정을 반복하면서 추정된 센서 출력 값과 실제 센서 출력 값의 차이에 따라 기준 모델링한 값과 실제 센서 값의 오차를 실시간으로 수신하여 시스템 모델 변수인 Q , R 값을 변화시키고 Kalman Gain의 값을 변화시킨다. 최종적으로 추정 값을 출력하면서 센서의 신호를 복구시킨다.

2.4.2. 영구적 고장

앞서 일시적인 고장 상황에서는 Kalman Filter를 활용하여 센서 신호를 복구시킨다고 기술하였다. 그러나 외부에서 큰 충격을 받거나, 센서 자체에서 영구적인 고장 발생 시에는 Kalman Filter만으로는 해결하기 어렵다. 따라서 본 논문에서는 센서에 영구적인 고장 발생 시 Deep Learning 기법을 활용하여 고장을 감지하고 센서의 신호를 복구하며, 전체적인 고장 진단 및 복구 알고리즘은 Fig. 2-23과 같다.

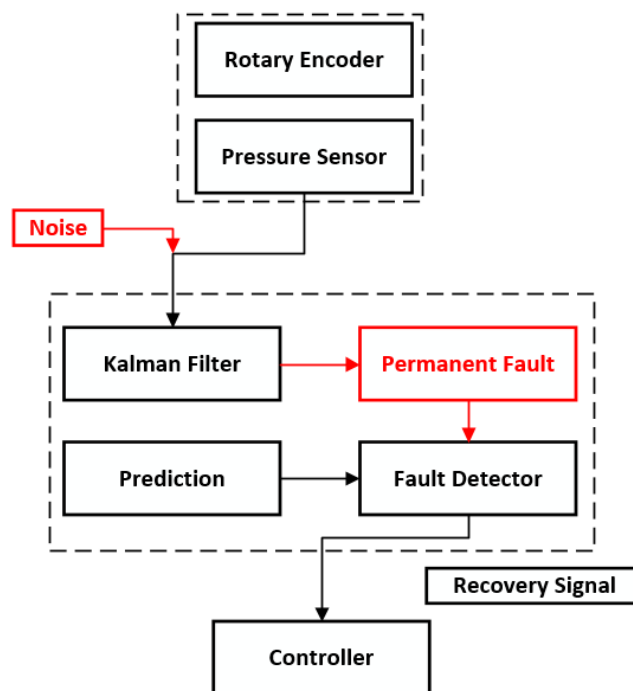


Fig. 2-23 고장 진단 및 복구 알고리즘

센서 출력에 영구적인 이상이 생길 경우 피드백 제어가 불가능해지며, 이는 큰 사고로 이어질 수 있다. 따라서 불안정한 센서 값을 대신해 예측 값을 출력하여 고장 상황을 극복해야 한다. 본 논문에서 제안한 알고리즘에서는 센서 값과 예측 값을 비교하여 고장 상황을 감지하였으며, 관련식은 식 (40), (41)과 같다. [33]

$$\theta_{pred} \quad \text{if} \quad \left| \frac{\theta_{pred} - \theta_{real}}{\theta_{pred}} \right| \geq 0.1 \quad (40)$$

$$\theta_{real} \quad \text{if} \quad \left| \frac{\theta_{pred} - \theta_{real}}{\theta_{pred}} \right| < 0.1 \quad (41)$$

θ_{pred} 는 예측 값, θ_{real} 은 센서 값을 나타낸다. 예측 값과 센서 값의 오차율이 10% 미만이면 센서의 신호를 출력하고, 오차율이 10% 이상일 경우 센서 값을 대체하여 예측 값을 출력하여 고장 상황을 극복한다. 출력 값은 제어기의 입력이 되어 매니플레이터의 동작을 제어한다.

2.4.2.1 NN (Neural Network)

NN은 신경망을 뜻하며 단순히 '함수'라 지칭할 수 있다. 함수는 입력에 대한 출력을 내보내는 변환기이며, 신경망도 함수와 같이 입력을 받아 무엇인가의 데이터를 출력한다.

2차원 데이터를 입력하여 3차원 데이터를 출력하는 기본적인 신경망의 그림은 Fig. 2-24와 같다.

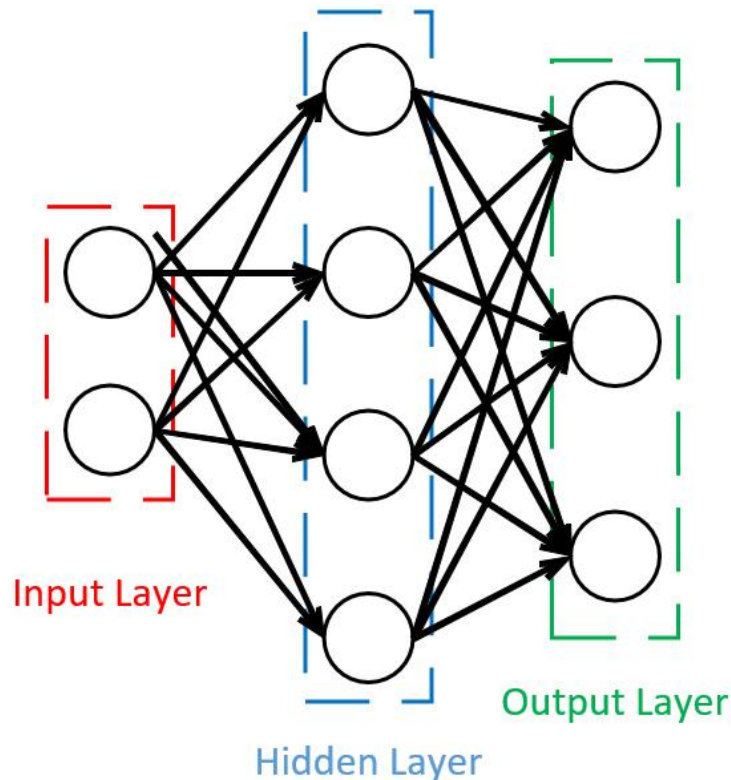


Fig. 2-24 Neural Network 구조

입력층에 뉴런 2개, 출력층에는 뉴런 3개, 그리고 은닉층에 4개의 뉴런을 배치한다. 0는 뉴런을 뜻하며, 뉴런 사이의 관계는 화살표로 표시된다. 화살표에는 가중치가 존재하여 가중치와 뉴런의 값을 곱하고 그 합이 다음 뉴런의 입력이 된다. 또한 각 층에는 이전 뉴런의 값에 영향을 받지 않는 정수도 더해지며 이 정수를 편향이라고 한다. 은닉층 중 첫 번째 뉴런의 값을 계산하는 식은 식 (42)와 같다.

$$h_1 = x_1w_{11} + x_2w_{21} + b_1 \tag{42}$$

x_1, x_2 는 입력층의 데이터, w_{11}, w_{21} 은 가중치, b_1 은 편향을 나타낸다. 식 (42)에서 가중치와 편향의 값을 바꿔가며 은닉층의 모든 뉴런의 값을 계산할 수 있다. 은닉층의 모든 뉴런을 계산하는 행렬은 식 (43)과 같다.

$$(h_1, h_2, h_3, h_4) = (x_1, x_2) \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{pmatrix} + (b_1, b_2, b_3, b_4) \quad (43)$$

최종 출력 층의 결과 역시 연결된 뉴런의 관계로 표현될 수 있으며 계산 방식은 앞서 계산한 식 (42)와 같이 계산할 수 있다.

2.4.2.2 RNN (Recurrent Neural Network)

RNN은 인공신경망의 한 종류로, 유닛간의 연결이 순환적 구조를 갖는 특징을 가지고 있다. RNN 계층은 순환하는 경로를 포함하며, 순환 경로를 따라 데이터를 계층 안에서 순환시킬 수 있으며, 이를 도식화하면 Fig. 2-25와 같다. x_t 는 시계열 데이터로 RNN 계층에 입력되는 데이터, h_t 는 입력에 대응하여 출력되는 데이터이다. RNN 계층을 순환구조로 펼친 도식도는 Fig. 2-26과 같다.

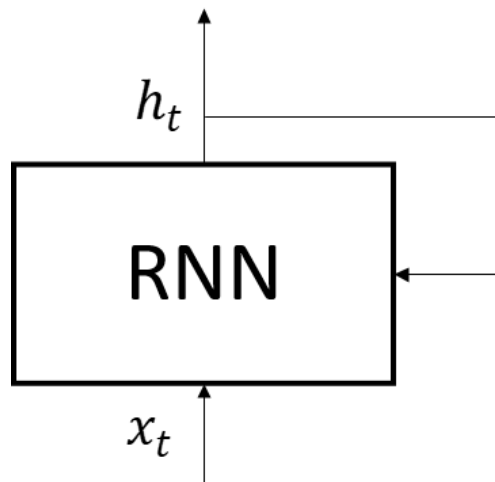


Fig. 2-25 표준 RNN 구조

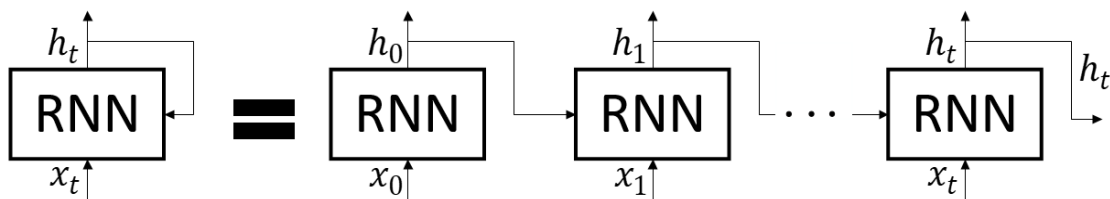


Fig. 2-26 RNN 구조 (전개 형상)

각 RNN 계층은 각 계층의 입력과 1개 전의 RNN 계층의 출력을 받는다. 2개의 정보를 바탕으로 현 시각의 출력을 계산하며 계산식은 식 (44)와 같다.

$$h_t = \tanh(h_{t-1}W_h + x_tW_x + b) \quad (44)$$

W_x 는 입력을 출력으로 변환하기 위한 가중치, W_h 는 이전 RNN 출력을 다음 시각의 출력으로 변환하기 위한 가중치, b 는 편향을 나타낸다. 식 (44)를 거친 결과인 h_t 는 다른 계층으로 출력되는 동시에 오른쪽의 RNN 계층으로도 출력이 된다. h_t 는 h_{t-1} 에 기초하여 계산되므로 RNN 계층은 h 라는 상태를 가지고 있으며, 기억력을 가지고 있는 계층이다.

2.4.2.3 LSTM (Long Short Term Memory)

LSTM은 Deep Learning 분야에서 사용되는 인공 신경망의 한 종류인 RNN에서 Memory cell이 추가된 구조이다.

RNN은 모델 학습변수에 대한 출력 결과와 라벨 데이터 사이 오차 Gradient를 계산하고 이를 오차에서부터 거꾸로 계산하는 역전파 과정을 거쳐 학습 변수에 오차를 반영하게 된다. 그러나 입력 데이터의 길이가 길어질수록 현재 시점(t) 결과를 출력하기 위해 의존해야 할 이전시점($t-n$)이 증가하게 되고 이로 인해 기울기가 사라지는 기울기 소실이 발생하여 학습 변수에 오차를 반영하지 못하게 된다.

그러므로 현재 시점(t)부터 이전 시점($t-n$) 길이가 제한되는 장기 의존성 문제가 발생하게 된다. 이를 해결하기 위해 본 논문에서는 Memory Cell이 추가된 LSTM을 고장 진단 및 복구 알고리즘에 적용한다. 기본적인 LSTM 구조를 도식화하면 Fig. 2-27과 같다.

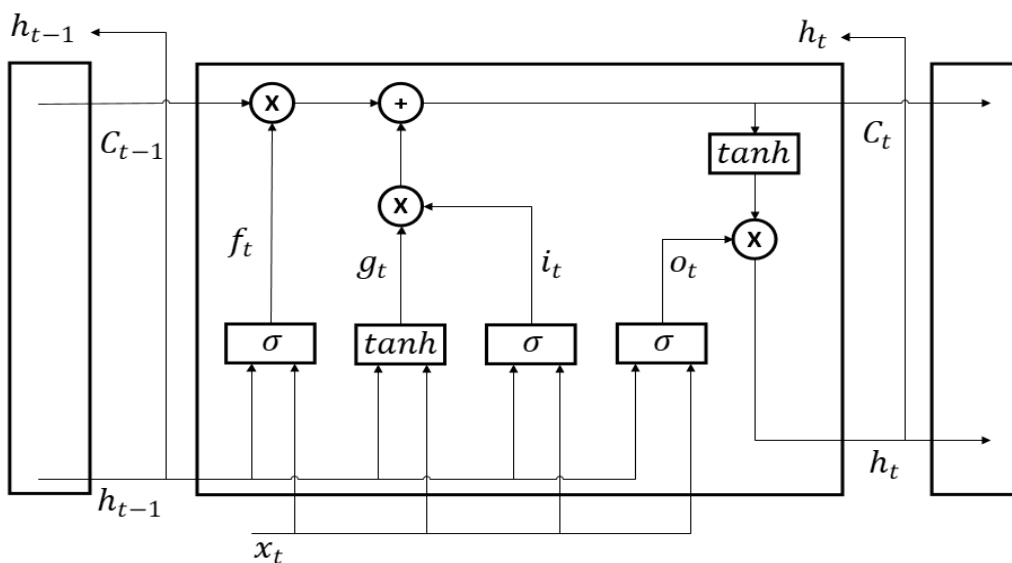


Fig. 2-27 표준 LSTM 구조

RNN은 재귀를 통한 정보전이 및 전파가 하나의 레이어로 제어되는 반면 LSTM은 Fig. 2-27과 같이 소실, 입력, 갱신, 출력 등 4가지 레이어가 재귀를 통한 정보 전파를 제어한다.

셀 상태 C_t 는 이전 시점($t-n$)부터 전달되는 정보를 전달하는 컨베이어 벨트와 같은 역할을 수행하며, 이를 통해 이전 시점($t-n$)이 증가하게 되더라도 Gradient가 사라지는 문제를 최소화하여 오차의 전파가 원활하게 이루어진다. Fig. 2-27에서 t 시점 입력 X_t 와 이전시점($t-1$) 은닉 상태 h_{t-1} 은 시그모이드(σ) 함수로 구성된 소실, 입력, 출력 레이어와 하이퍼볼릭탄젠트(\tanh) 함수로 구성된 갱신 레이어를 거쳐 현재 시점(t) 은닉 상태 h_t 와 셀 상태 C_t 를 생성하여 다음 시점($t+1$)로 전파한다.

소실은 Fig. 2-27에서 첫 번째 시그모이드 함수로 구성된 레이어로 이전 시점 은닉 상태 정보를 유지하기 위한 정도를 결정한다. 시그모이드 함수에 의해 소실 레이어 값은 0에서 1사이 값으로 결정된다. 0은 이전 시점($t-1$) 은닉 상태 h_t 정보를 유지할 필요가 없다는 의미이며, 1은 정보 전체를 유지해야 한다는 의미이다. 소실 레이어는 식 (45)와 같다.

$$F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (45)$$

입력은 Fig. 2-27에서 두 번째 시그모이드 함수로 구성된 레이어로 새로 입력받은 데이터 정보를 셀 상태에 반영하기 위한 정도를 결정한다. 시그모이드 함수에 의해 입력 레이어 값은 0에서 1사이 값으로 결정된다. 0은 새로운 입력 값 정보를 셀 상태에 반영할 필요가 없다는 의미이며, 1은 새로운 입력 값 정보 전체를 셀 상태에 반영해야 한다는 의미이다. 입력 레이어는 식 (46)과 같다.

$$I_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (46)$$

갱신은 Fig. 2-27에서 세 번째 위치한 하이퍼볼릭 탄젠트 함수로 구성된 레이어로 갱신 레이어는 새로 입력받은 데이터 정보를 셀 상태에 반영하기 위한 값을 생성한다. 하이퍼볼릭 탄젠트 함수에 의해 갱신 레이어 값은 -1에서 1사이 값으로 결정된다. 입력 레이어가 0에서 1사이 값으로 새로 입력받은 데이터 정보를 셀 상태에 반영하기 위한 정보를 나타냈다면, 갱신 레이어는 -1사이에서 1사이 값으로 새로 입력받은 데이터 정보를 셀 상태에 반영하기 위한 방향을 결정한다. 갱신 레이어는 식 (47)과 같다.

$$U_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (47)$$

현재 시점(t) 셀 상태는 앞선 소실, 입력, 갱신 레이어 값을 바탕으로 식 (48)과 같다. 소실 레이어에서 결정된 이전 시점($t-1$) 유지 정도와 이전 시점($t-1$) 셀 상태를 곱하고 현재 시점 입력 레이어 값과 갱신 레이어 값을 곱하여 합한 값으로 현재 시점(t)의 셀 상태를 생성한다.

$$C_t = F_{t-1} \times C_{t-1} + I_t \times U_t \quad (48)$$

출력은 Fig. 2-28에서 네 번째 위치한 시그모이드 함수로 구성된 레이어로 셀 상태 정보를 다음 시점($t+1$)에 전파하기 위한 정도를 결정한다. 시그모이드 함수에 의해 출력 레이어 값은 0에서 1사이 값으로 결정된다. 0은 셀 상태의 정보를 전파할 필요가 없다는 의미이며, 1은 정보 전체를 전파해야 한다는 의미이다. 출력 레이어는 식 (49)와 같다.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (49)$$

h_t 는 현재 시점(t)에서 다음 시점($t+1$)에 전파하는 정보를 나타내며, 식 (50)과 같다. 셀 상태를 하이퍼볼릭탄젠트 함수를 이용하여 -1부터 1사이 값으로 변환하고 출력 레이어 값과 곱하여 생성한다.

$$h_t = o_t \times \tanh(C_t) \quad (50)$$

2.4.2.4 GRU(Gated Recurrent Unit)

GRU는 검증된 게이트가 추가된 RNN을 나타낸다. 기본적인 GRU 계층은 Fig. 2-28과 같다.

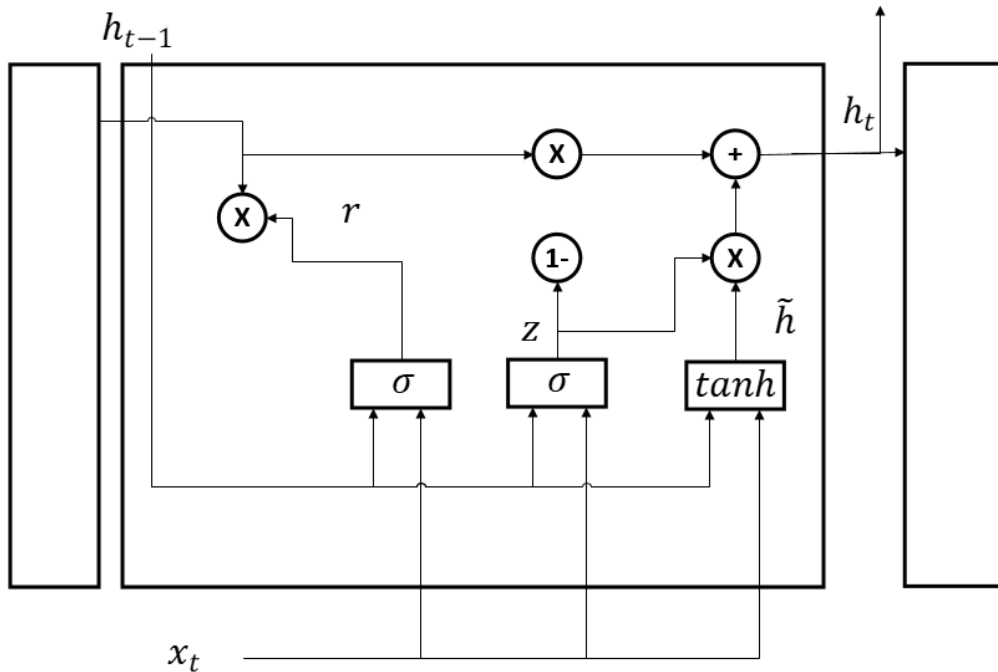


Fig. 2-28 표준 GRU 구조

GRU계층은 LSTM 계층과 다르게 기억셀이 없고, 시간 방향 전파는 은닉 상태의 h 만 존재한다. 그리고 r, z 의 2개의 게이트를 사용한다. r 은 reset 게이트, z 는 update 게이트를 의미한다. reset 게이트는 과거의 은닉 상태를 얼마나 무시할지를 결정한다. update 게이트는 은닉 상태를 갱신하는 게이트이며, LSTM의 forget 게이트와 input 게이트를 합친 역할을 담당

한다. 내부 계산식은 식 (51) ~ (54)와 같다.

$$z = \sigma(x_t W_x^{(z)} + h_{t-1} W_h^{(z)} + b^{(z)}) \quad (51)$$

$$r = \sigma(x_t W_x^{(r)} + h_{t-1} W_h^{(r)} + b^{(r)}) \quad (52)$$

$$\tilde{h} = \tanh(x_t W_x + (r \odot h_{t-1}) W_h + b) \quad (53)$$

$$h_t = (1 - z) \odot h_{t-1} + z \odot \tilde{h} \quad (54)$$

3. 시뮬레이션 및 결과

고장 진단 및 복구 알고리즘을 실차 실험에 적용하기 위해 시뮬레이션 모델을 구축한다. 실제 매니플레이터 시스템을 대상으로 알고리즘을 바로 적용하기에는 어려움이 있다. 고장이 발생한 센서를 확보하기가 쉽지 않고, 고장 센서를 장착하고 실험을 진행하면 시스템에 문제가 발생할 수 있다. 또한 안전에 대한 문제도 발생할 가능성이 있다. 이와 같은 문제들을 해결하기 위해 시뮬레이션 모델을 구축한다. 시뮬레이션 모델은 실제 시스템과 비교하여 반복성이 우수하고 안전성 문제를 해결할 수 있다는 장점이 있다. 따라서 본 논문에서는 실제 매니플레이터 시스템에 고장 진단 및 복구 알고리즘을 적용하기 위해 우선적으로 시뮬레이션 모델을 구축하고 고장 진단 및 복구 알고리즘을 적용하여 결과를 도출해 알고리즘의 타당성을 검증하였다.

3.1. 시뮬레이션 모델 구축

시뮬레이션 모델 구축을 위해 3D 모델링, 기구학 모델링, 유압 회로 모델링 과정을 수행하였으며, 이를 도식화하면 Fig. 3-1과 같다. 최종적으로 실제 매니플레이터 시스템과 비교하여 시뮬레이션 모델의 신뢰성을 확보하였다.

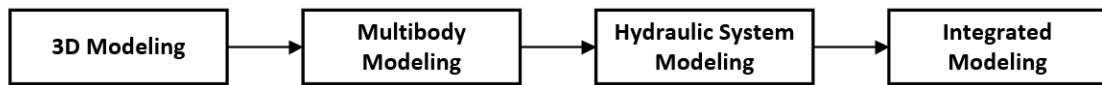
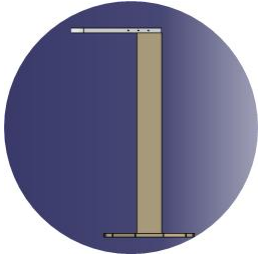


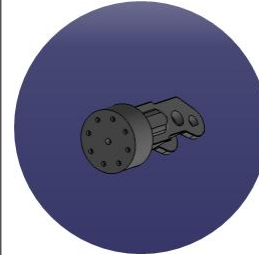


Fig. 3-1 시뮬레이션 모델링 진행 과정

3.1.1. 3D 모델링

3D 모델링 소프트웨어인 CATIA를 활용하여 유압 매니플레이터 부품에 대하여 모델링 작업을 수행하였다. 이후 SolidWorks를 활용하여 각 부품들을 조립하였고, 유압매니플레이터 부품 및 전체 조립 형상은 Table 3-1, Fig. 3-2와 같다.

Table 3-1 유압 매니플레이터 부품 형상

			
Base	Cylinder	Hydraulic Motor	End-Effector

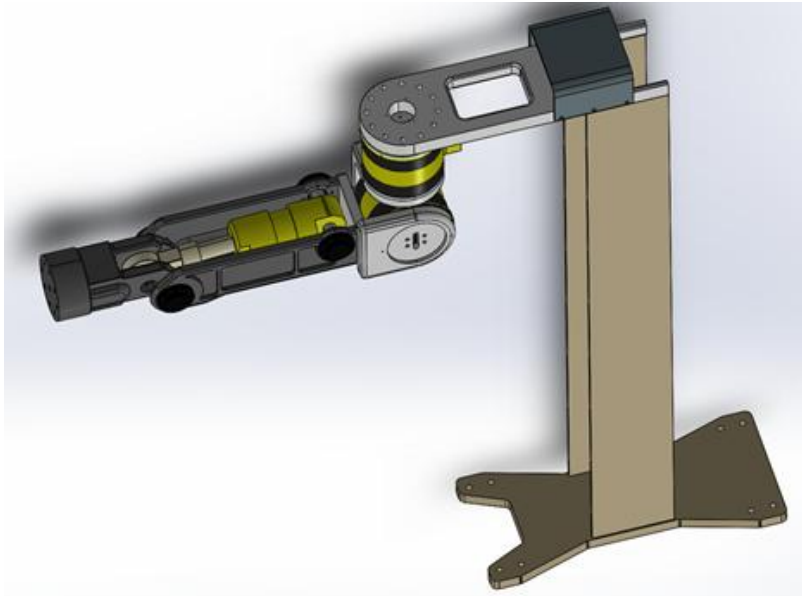


Fig. 3-2 유압 매니플레이터 조립 형상

3.1.2. Multi-Body 모델링

Simscape Multibody는 로봇, 차량 서스펜션, 건설 장비, 항공기 랜딩 기어 등 3차원 기계 시스템을 위한 멀티바디 시뮬레이션 환경을 제공하는 Simscape의 툴 박스이다. 질량, 관성, 관절, 제약 조건, 3차원 기하 구조 등 CAD 어셈블리를 모델로 불러올 수 있으며, 자동으로 생성된 3D 애니메이션을 통해 시스템 Dynamics를 시각화할 수 있다.

3.1.1절에서 SolidWorks를 활용해 조립된 모델을 SolidWorks에서 Simscape Multi-body 모델을 생성하였으며, 그 결과는 Fig. 3-3과 같다.

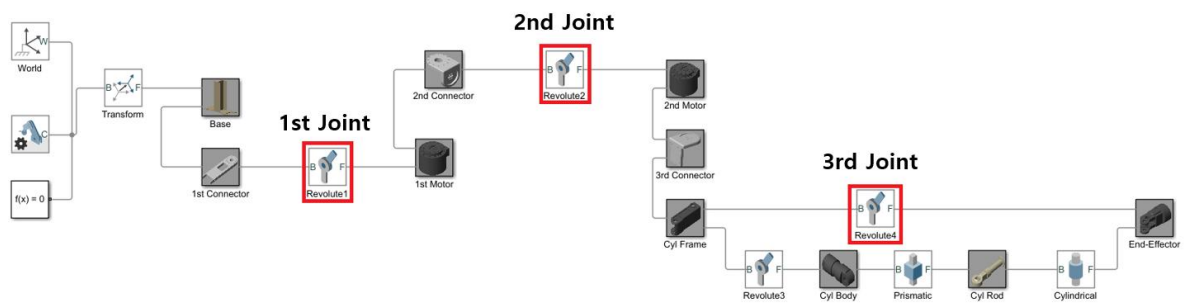


Fig. 3-3 Multi-Body 모델링 형상

첫 번째, 두 번째 관절은 유압 모터가 회전하면서 움직이는 관절이며, 세 번째 관절은 유압 실린더가 왕복운동하며 끝단이 회전운동을 하는 관절이다. Multi-body model을 생성할 때 각 블록들의 좌표계는 자동적으로 계산되어 변환된다. 하지만 계산된 좌표계가 앞서 2.1.1절

에서 정의한 좌표계와 일치하지 않는 경우가 존재할 수 있으므로 각 관절에 대해 센서를 부착하여 Calibration을 진행하였다. Calibration 작업은 각 관절에 +, -의 힘, 토크를 인가하여 관절 각도의 초기 값과 관절의 회전 방향에 따른 관절 각도의 부호를 확인하여 실제 좌표계와의 오차를 줄이는 것을 의미한다. Calibration을 수행하기 전 각 Mode의 이론값과 Simulation 상에서 도출된 값을 비교한 결과는 Table 3-2와 같다.

Table 3-2 관절 각도 값 비교 결과 (Calibration 수행 전)

Joint	Mode	Theory	Simulation
1	Home	0°	0°
	CCW	+55°	+55°
	CW	-55°	-55°
2	Home	0°	-85°
	Up	5°	-90°
	Down	107°	-192°
3	Home	0°	0°
	Up	-3°	-3°
	Down	-97°	-97°

비교 결과, 첫 번째, 세 번째 관절은 Calibration이 필요 없음을 알 수 있으며 두 번째 관절은 Calibration이 필요하다. Table 3-2의 결과를 바탕으로 각 관절에 대한 Calibration 결과는 Table 3-3과 같다.

Table 3-3 Calibration 수행 결과

Joint	Calibration
1	θ_1
2	$-\theta_2 - 85$
3	θ_3

Multi-Body 모델에서 Calibration 작업을 수행한 후의 결과는 Table 3-4와 같고, Simulink에서 구성한 형상은 Fig 3-4와 같다.

Table 3-4 관절 각도 값 비교 결과 (Calibration 수행 후)

Joint	Mode	Theory	Simulation
1	Home	0°	0°
	CCW	+55°	+55°
	CW	-55°	-55°
2	Home	0°	0°
	Up	5°	5°
	Down	107°	107°
3	Home	0°	0°
	Up	-3°	-3°
	Down	-97°	-97°

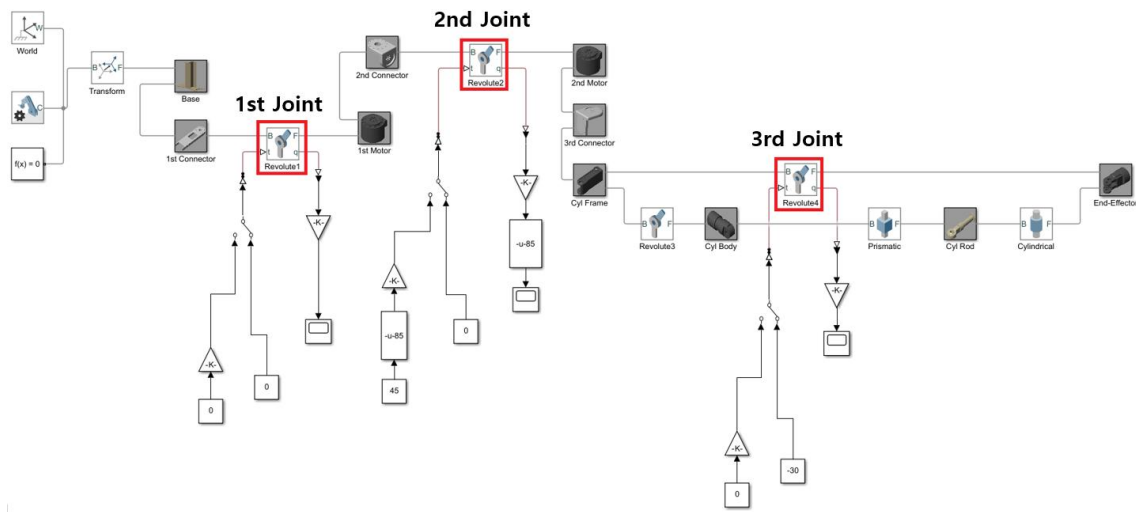
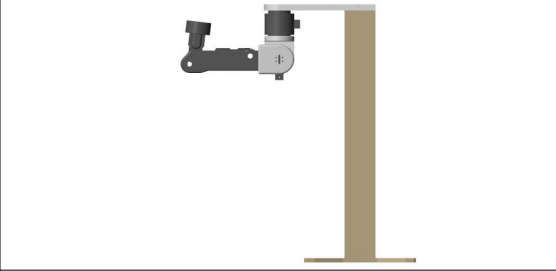
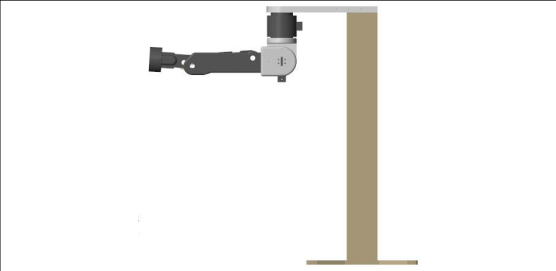


Fig. 3-4 Multi-Body 모델링 형상 (Calibration 작업 후)

Calibration 작업 후 Multi-Body 모델이 실제 시스템과 같이 동작하는지 검증하기 위해 각 관절에 토크와 힘을 인가하여 모델의 동작을 확인하였으며, 결과는 Table 3-5와 같다.

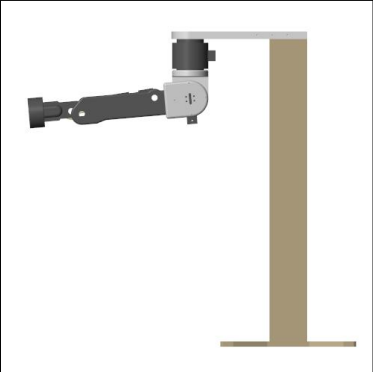
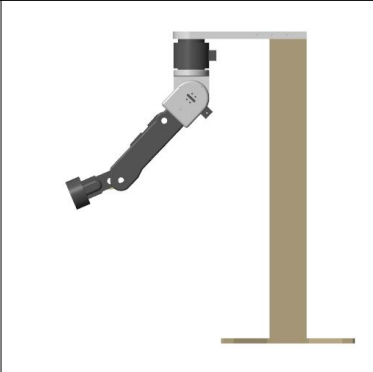
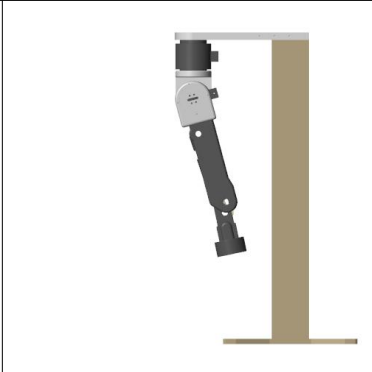
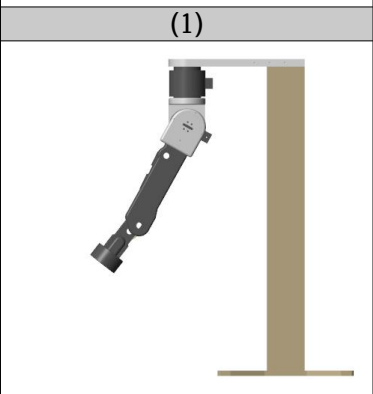
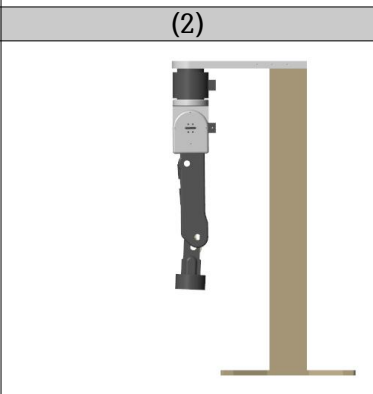
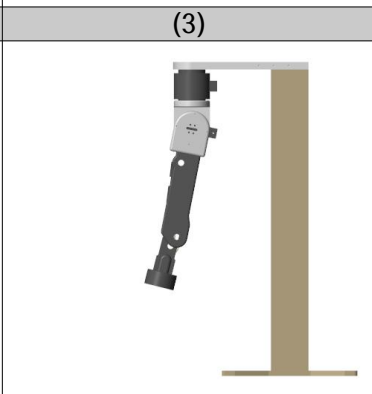
Table 3-5 Multi-Body 모델 동작 형상

Joint	Force/Torque	Motion
1	+30Nm	
	-30Nm	
2	+30Nm	
	-30Nm	

3	+10N	
	-10N	

최종적으로 Multi-Body 모델에 중력을 적용하여 시뮬레이션을 수행하였고 결과는 Table 3-6과 같다.

Table 3-6 시뮬레이션 결과 (1→6 순서로 동작)

		
(1)	(2)	(3)
		
(4)	(5)	(6)

시뮬레이션 결과 중력의 영향으로 매니플레이터가 하강하며 각 관절의 제한각도로 인해 작업 반경 내에서 동작하는 것을 확인할 수 있으며, 팔의 움직임이 자연스럽게 나오므로 Multi-Body 모델링 작업이 잘 이루어졌다 할 수 있다.

3.1.3. 유압 시스템 모델링

SimHydraulics는 유압 펌프, 모터, 액추에이터, 밸브, 파이프라인 등이 포함된 라이브러리를 제공하고 유체 및 유압 시스템에 대한 시뮬레이션 환경을 제공하는 Simscape의 Tool box이다.

2.2절의 유압 회로를 기반으로 유압 펌프, 유압 모터, 유압 실린더, 서보 밸브 각각의 구성 요소에 대해 SimHydraulics를 활용하여 모델링을 수행하였다. 최종적으로 이론적인 계산식의 결과와 SimHydraulic를 활용한 결과를 비교하여 타당성을 검증하였다.

3.1.3.1 유압 펌프 모델링

3축 유압 매니플레이터에 장착된 펌프는 가변 용량형 피스톤 펌프이며 최대 용적은 16cc/rev이다. 하지만 컨트롤 밸브를 고정하여 사용을 하였으므로 16cc/rev의 고정 용량형 펌프로 모델링해도 무방하다. 모터의 회전수는 1440rpm이며, 펌프의 체적 효율은 92%라 가정하였다. 모델링된 펌프의 모습과 파라미터는 Table 3-7, Fig. 3-5와 같다.

Table 3-7 유압 펌프 모델링 파라미터

Displacement	16cc/rev
Volumetric efficiency	0.92

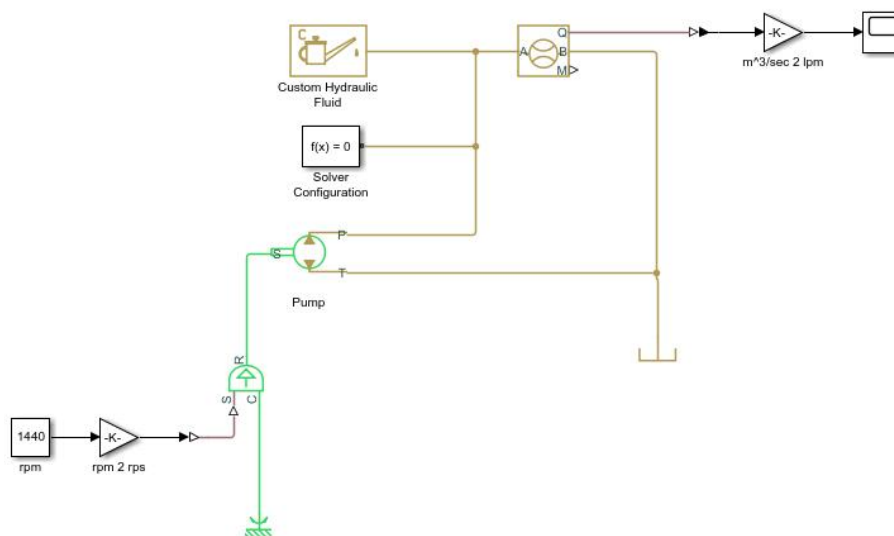


Fig. 3-5 유압 펌프 모델링 형상

시뮬레이션 모델의 유량과 이론적인 계산식을 통한 유량을 비교하기 위해 펌프가 10초 동안 작동한다고 가정하고 두 유량을 비교하였다. 결과는 Fig. 3-6과 같다.

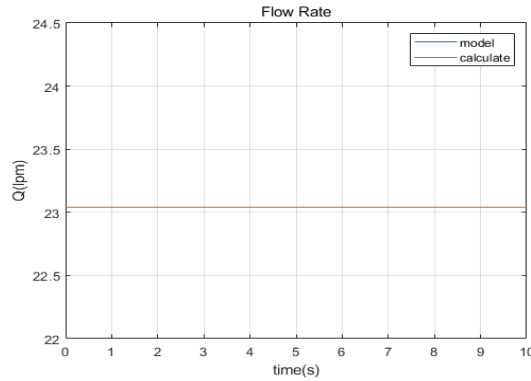


Fig. 3-6 시뮬레이션 결과 (유량 비교)

비교 결과 시뮬레이션 모델의 유량과 이론적인 계산식을 통한 유량이 일치함을 확인하였다.

3.1.3.2 유압 모터 모델링

3축 유압 매니퓰레이터에 장착된 유압 모터는 더블 베인 모터이며 용적은 13ml/rad, 최대 토크는 400Nm, 최대 압력은 210bar, 제한 각도는 110° 이다. 모터 체적 효율은 92%라 가정하고 모델링된 모터의 모습과 파라미터는 Table 3-8, Fig. 3-7과 같다.

Table 3-8 유압 모터 모델링 파라미터

Displacement	1.3m ³ /rad
Volumetric efficiency	0.92

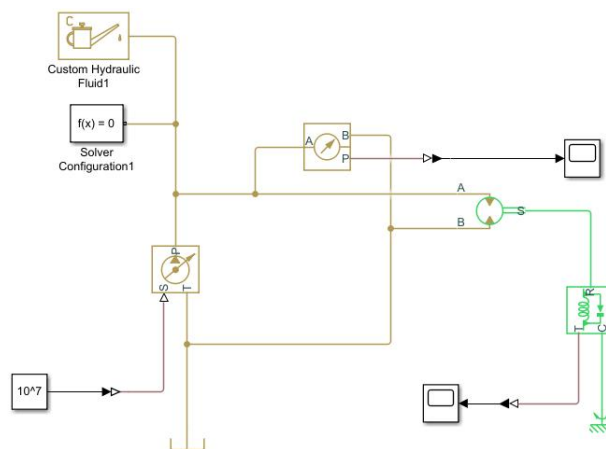


Fig. 3-7 유압 모터 모델링 형상

시뮬레이션 모델의 토크와 이론적인 계산식을 통한 토크를 비교하기 위해 모터가 10초 동안 회전한다고 가정하고 두 토크를 비교하였다. 결과는 Fig. 3-8과 같다.

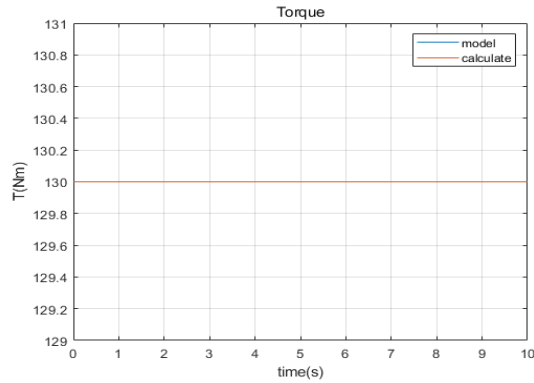


Fig. 3-8 시뮬레이션 결과 (토크 비교)

비교 결과 시뮬레이션 모델의 토크와 이론적인 계산식을 통한 토크가 일치함을 확인하였다.

3.1.3.3 유압 실린더 모델링

3축 유압 매니플레이터에 장착된 실린더는 싱글 로드 실린더이며 최대 힘 32kN, 최대 압력은 210bar, Rod Diameter는 25mm, Head Diameter는 44.2mm Stroke는 55.5mm이다. 모델링된 실린더의 모습과 파라미터는 Table 3-9, Fig. 3-9와 같다.

Table 3-9 유압 실린더 모델링 파라미터

Piston area A	$\frac{\pi}{4} \times 44.2^2 \text{ mm}^2$
Piston area B	$\frac{\pi}{4} \times (44.2^2 - 25^2) \text{ mm}^2$
Stroke	55.5mm

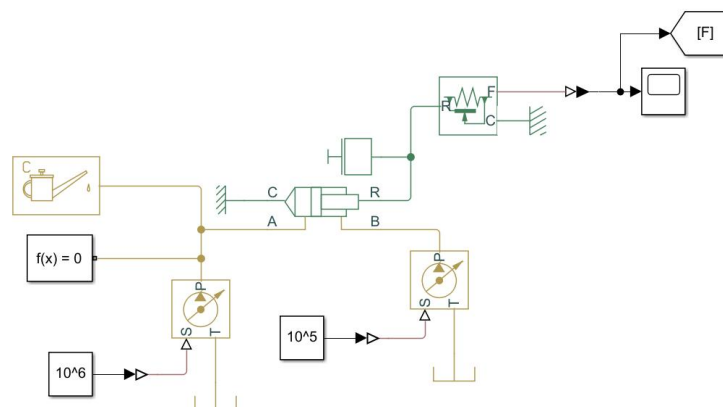


Fig. 3-9 유압 실린더 모델링 형상

시뮬레이션 모델의 힘 이론적인 계산식을 통한 힘을 비교하기 위해 실린더가 10초 동안 직선 운동을 한다고 가정하고 두 힘을 비교하였다. 결과는 Fig. 3-10과 같다.

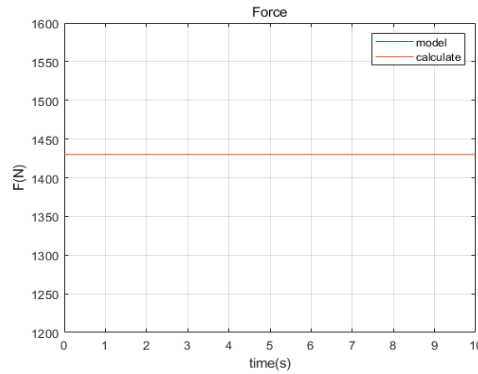


Fig. 3-10 시뮬레이션 결과 (힘 비교)

비교 결과 시뮬레이션 모델의 힘과 이론적인 계산식을 통한 힘이 일치함을 확인하였다.

3.1.3.4 서보 밸브 모델링

본 논문의 유압 매니플레이터에는 실린더를 제어하는 서보 밸브와 유압 모터를 제어하는 서보 밸브 두 가지가 장착되어 있다. 시뮬레이션 모델에서는 서보 밸브를 방향 제어 밸브로 대체하여 시뮬레이션을 수행하였다. 모델링된 밸브와 파라미터는 Table 3-10, Fig. 3-11과 같다.

Table 3-10 밸브 모델링 파라미터

Flow coefficient	0.7
Maximum opening area	300mm ²

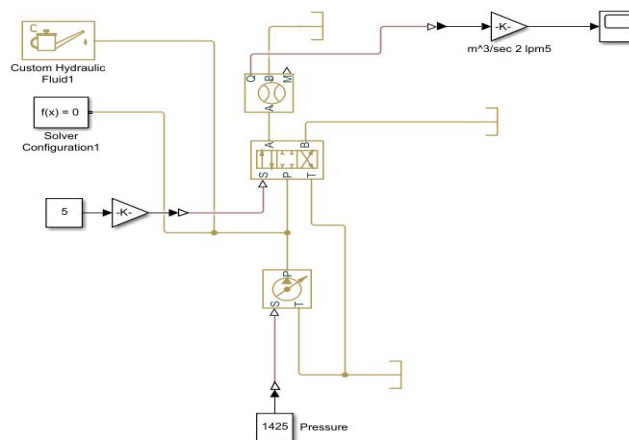


Fig. 3-11 밸브 모델링 형상

시뮬레이션 모델의 유량과 이론적인 계산식을 통한 유량을 비교하기 위해 밸브에 10초 동안 압력이 가해진다고 가정하고 두 유량을 비교하였다. 결과는 Fig. 3-12와 같다.

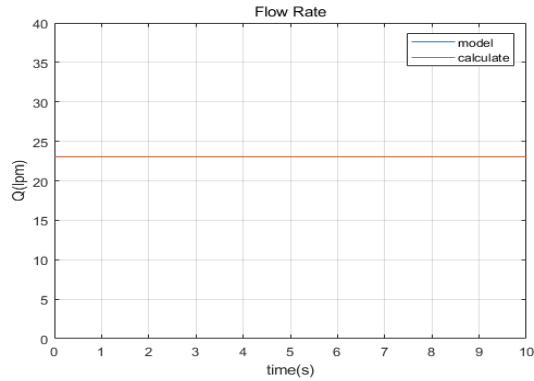


Fig. 3-12 시뮬레이션 결과 (유량 비교)

비교 결과 시뮬레이션 모델의 유량과 이론적인 계산식을 통한 유량이 일치함을 확인하였다.

3.1.3.5 유압 회로 모델링

SymHydraulics를 활용하여 모델링한 펌프, 유압 모터, 실린더, 밸브를 결합해 만든 가상 매니플레이터의 전체 유압 회로 모델링 형상은 Fig. 3-13과 같다.

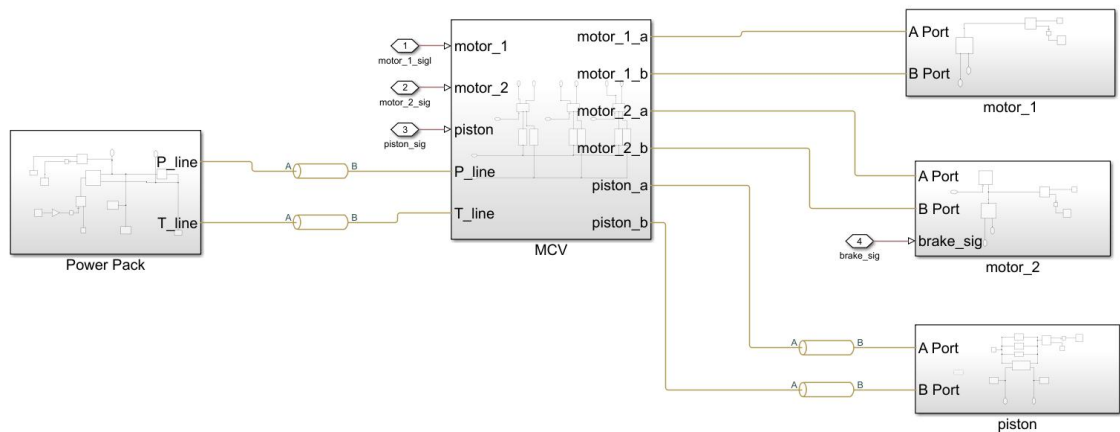


Fig. 3-13 유압 회로 모델링 형상

모델링된 유압 회로가 실제 시스템과 비슷하게 동작하는지 검증하기 위해 파워팩에서 유량을 토출하여 최종적으로 유압 모터와 실린더가 작동하는지 확인하였으며, 그 결과는 Fig. 3-14, 3-15, 3-16과 같다.

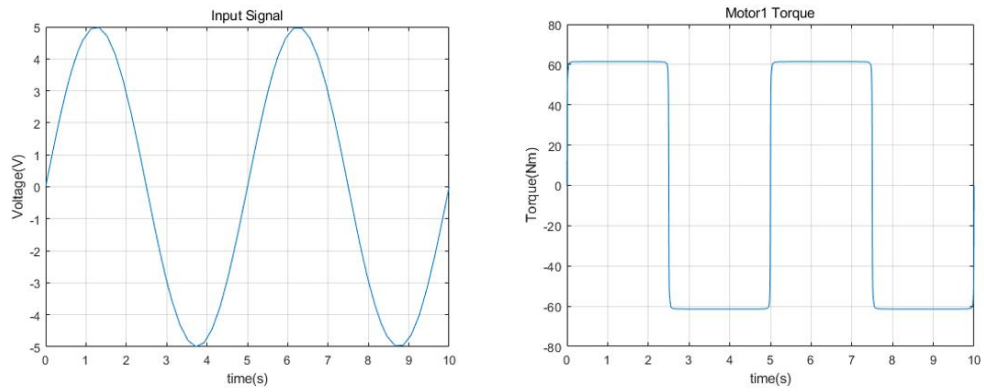


Fig. 3-14 시뮬레이션 결과 (유압 모터 1)

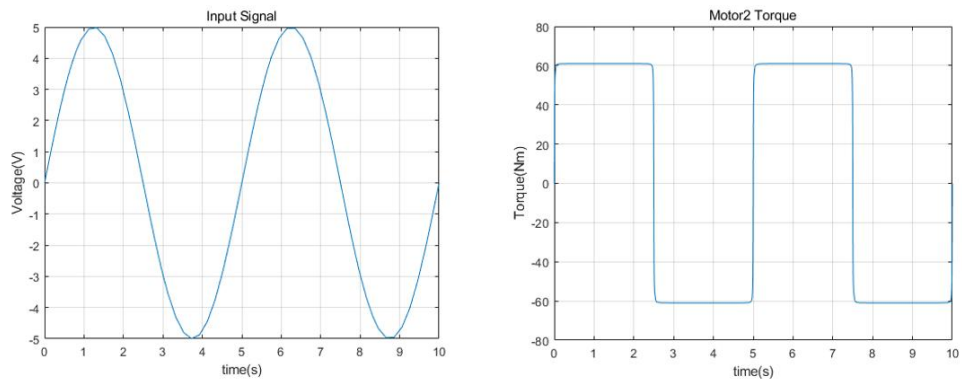


Fig. 3-15 시뮬레이션 결과 (유압 모터 2)

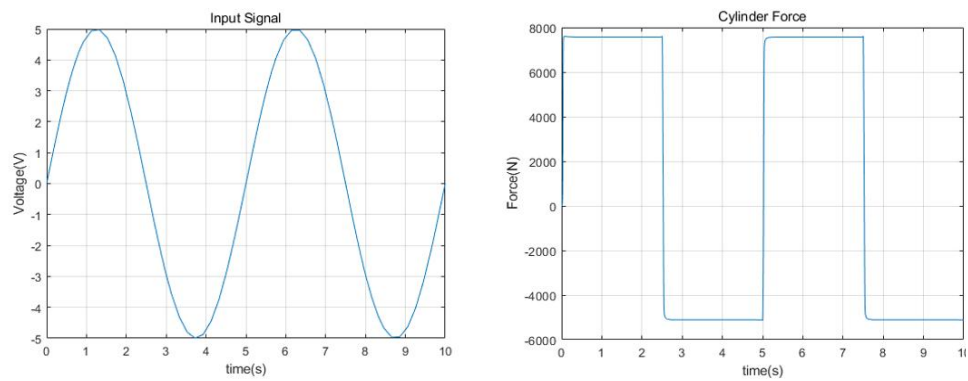


Fig. 3-16 시뮬레이션 결과 (유압 실린더)

시뮬레이션 결과 유압 모터와 실린더에 유량을 공급하여 토크와 힘을 만들어낼 수 있음을 확인하였다. 따라서 SimHydraulics를 활용한 전체 유압 회로 모델링은 타당하다 할 수 있다.

3.1.4 통합 모델링

3.1.1절부터 3.1.3절까지의 내용을 기반으로 한 3축 유압 매니플레이터에 대한 최종 모델의 형상은 Fig. 3-17과 같다.

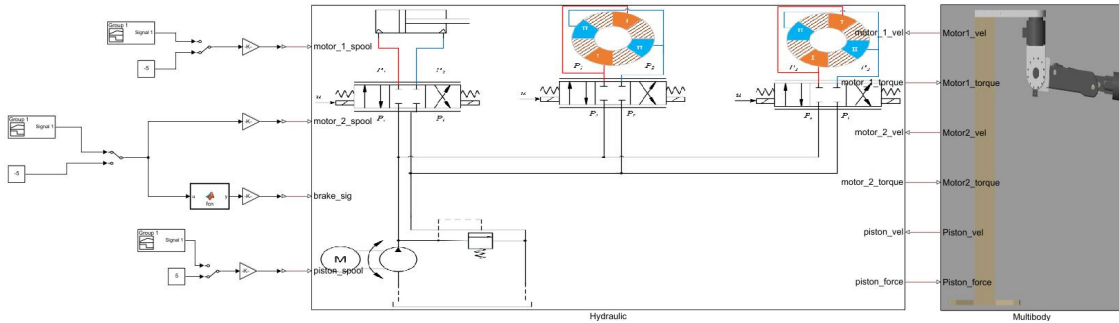


Fig. 3-17 최종 모델링 형상

본 논문에 사용된 시뮬레이션 모델에서는 유압 액추에이터에서 발생한 힘, 토크를 기구부로 전달하며, 기구부에서 발생한 속도, 각속도를 다시 유압부의 부하로 전달할 수 있게 유압 회로와 기구부를 결합하였다. 이와 관련된 에너지 방정식은 식 (55), (56)과 같다.

$$F = M\ddot{x} + C\dot{x} + kx \quad (55)$$

$$\tau = I\ddot{\theta} + B\dot{\theta} + K_s\theta \quad (56)$$

시뮬레이션 모델의 신뢰성을 확보하기 위해 실제 매니플레이터와 시뮬레이션 모델에 동일한 제어 신호를 인가하여 각 관절별로 압력과 각도를 비교하였으며, 그 결과는 Table 3-11, 3-12, 3-13과 같다.

Table 3-11 시뮬레이션 결과 (Joint 1)

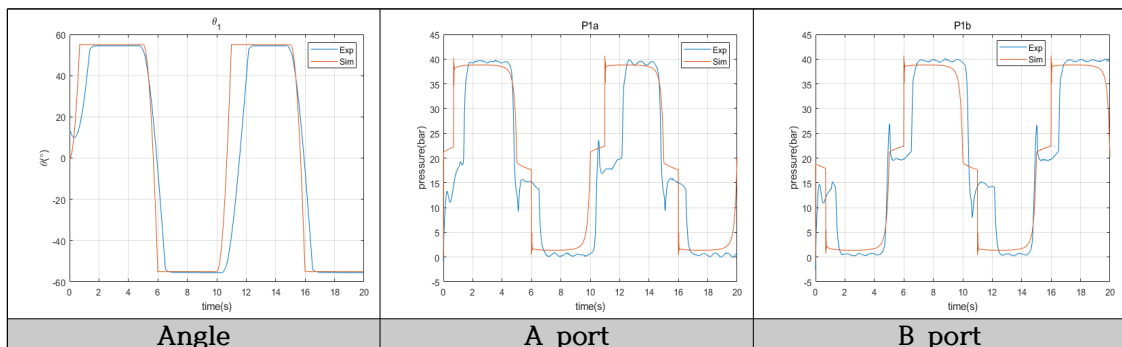


Table 3-12 시뮬레이션 결과 (Joint 2)

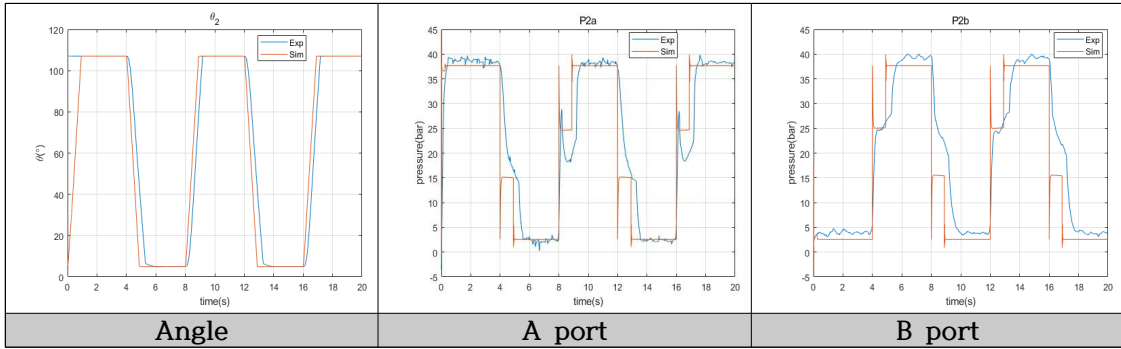
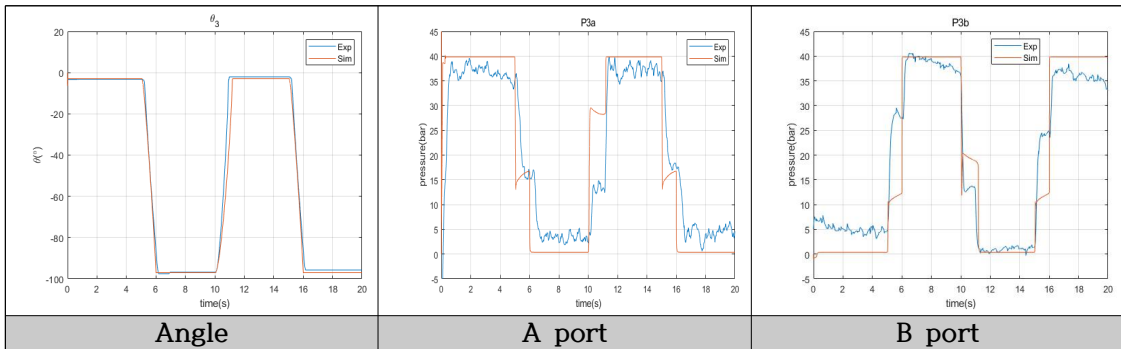


Table 3-13 시뮬레이션 결과 (Joint 3)

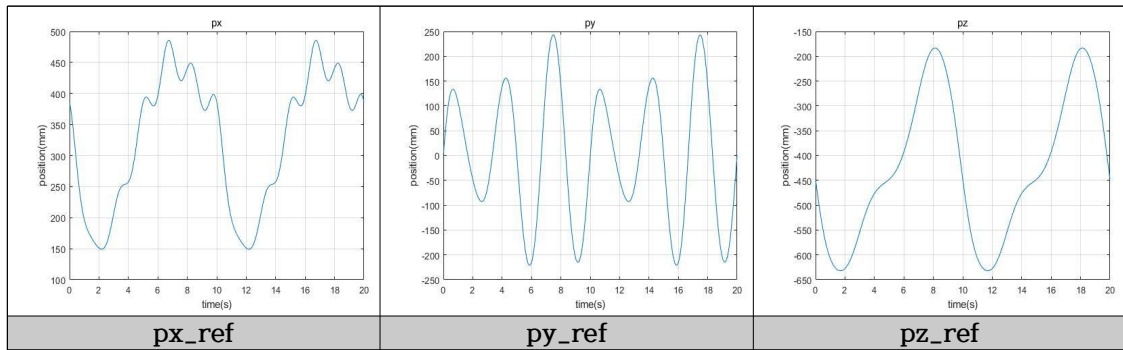


실제 매니플레이터와 비교 결과, 동일한 시간에 같은 제어입력을 인가하였을 때 출력되는 각도, 압력 그래프의 개형이 비슷한 것을 확인할 수 있으며, 가상 시뮬레이션 모델과 실제 시스템을 비교함으로써 구축한 시뮬레이션 모델의 신뢰성을 확보하였다.

3.2. 제어기 설계

본 논문에서는 매니플레이터 시스템에서 Position Tracking을 수행할 때 센서 고장 발생 시 고장을 진단하고 센서 신호를 복구하는 알고리즘을 개발한다. 따라서 본 절에서는 고장 진단 및 복구 알고리즘을 적용하기 전 제어기를 설계하여 Position Tracking을 수행한다. 3.1절에서 구축한 시뮬레이션 모델에 설계한 제어기를 적용하여 경로 추적 성능을 검증한다. 기준 경로는 Table 3-14와 같다.

Table 3-14 기준 경로 설정 결과



3.2.1 PID 제어기 설계

본 논문에서 사용된 PID 제어기의 블록선도는 Fig. 3-18과 같다.

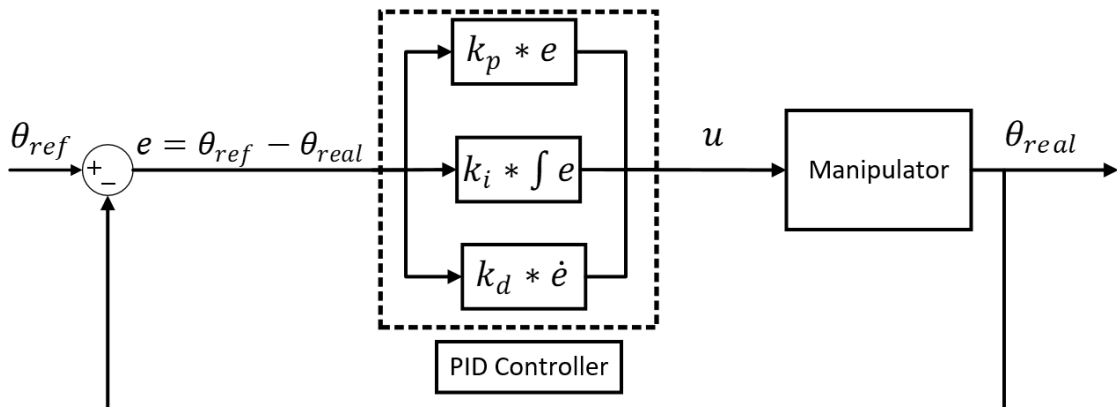


Fig. 3-18 PID 제어기 블록선도

θ_{ref} 는 각 관절의 원하는 각도를 나타내며 θ_{real} 은 매니플레이터의 센서를 통해 얻을 수 있는 실제 각도를 나타낸다. e 는 각 관절의 원하는 각도와 실제 각도의 차이를 나타낸다. u 는 PID 제어기에서 계산된 매니플레이터에 인가되는 제어 신호를 나타내며 계산된 제어 신호는 매니플레이터의 입력이 되어 원하는 각도인 θ_{ref} 를 추적한다. 본 논문에서 사용된 각 관절의 PID 제어기의 파라미터는 Table 3-15와 같다.

Table 3-15 PID 제어기 파라미터

Joint	k_p	k_i	k_d
1	5	1	1
2	5	1	1
3	2	0.1	0.01

제어기의 성능을 검증하기 위해 Table 3-14와 같은 기준 경로를 설정하고 시뮬레이션을 수행하였다. 제어기 적용 전 결과는 Table 3-16과 같고, 적용 후 결과는 Table 3-17과 같다.

Table 3-16 기준 경로 추적 결과 (PID 제어기 적용 전)

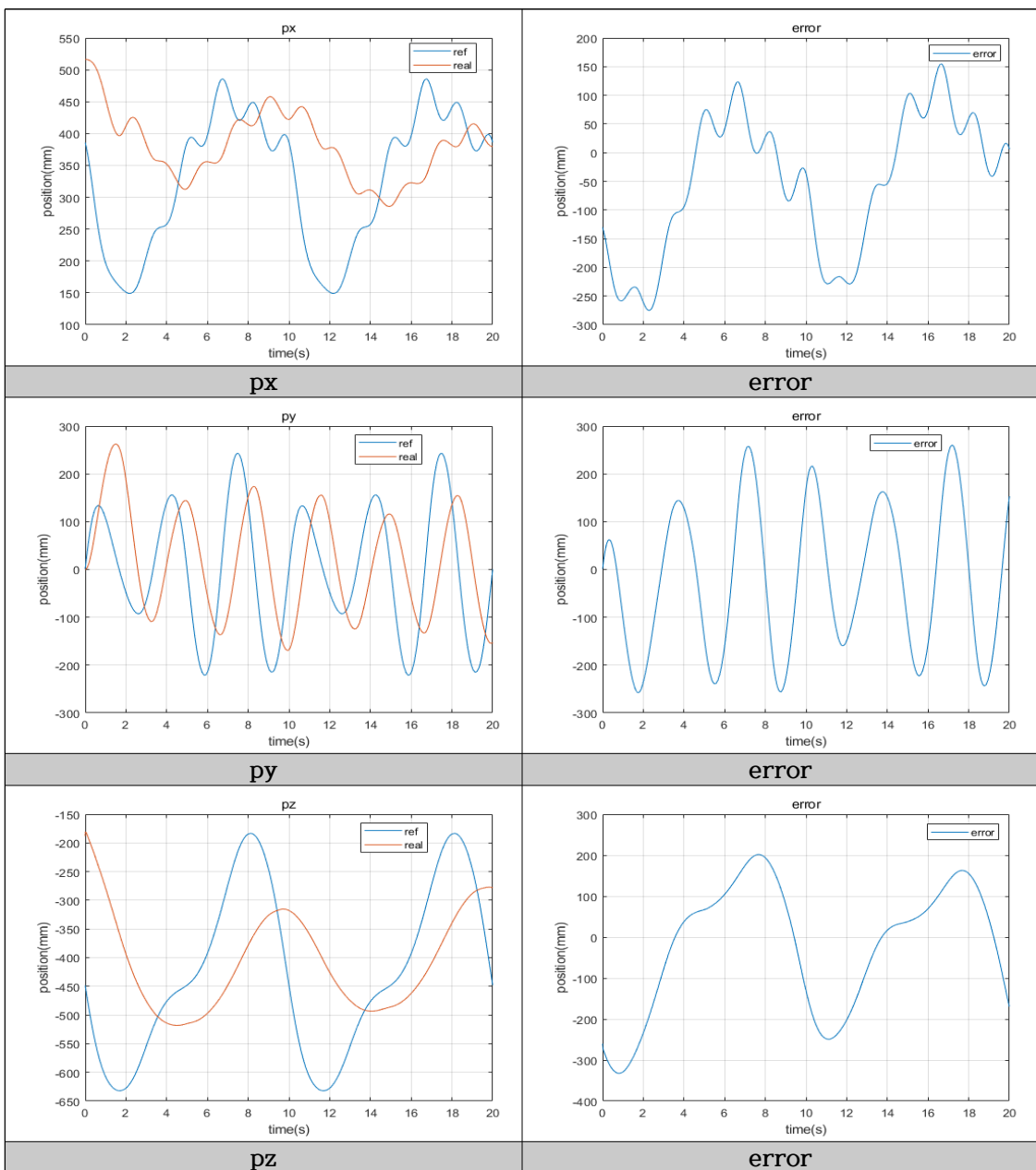
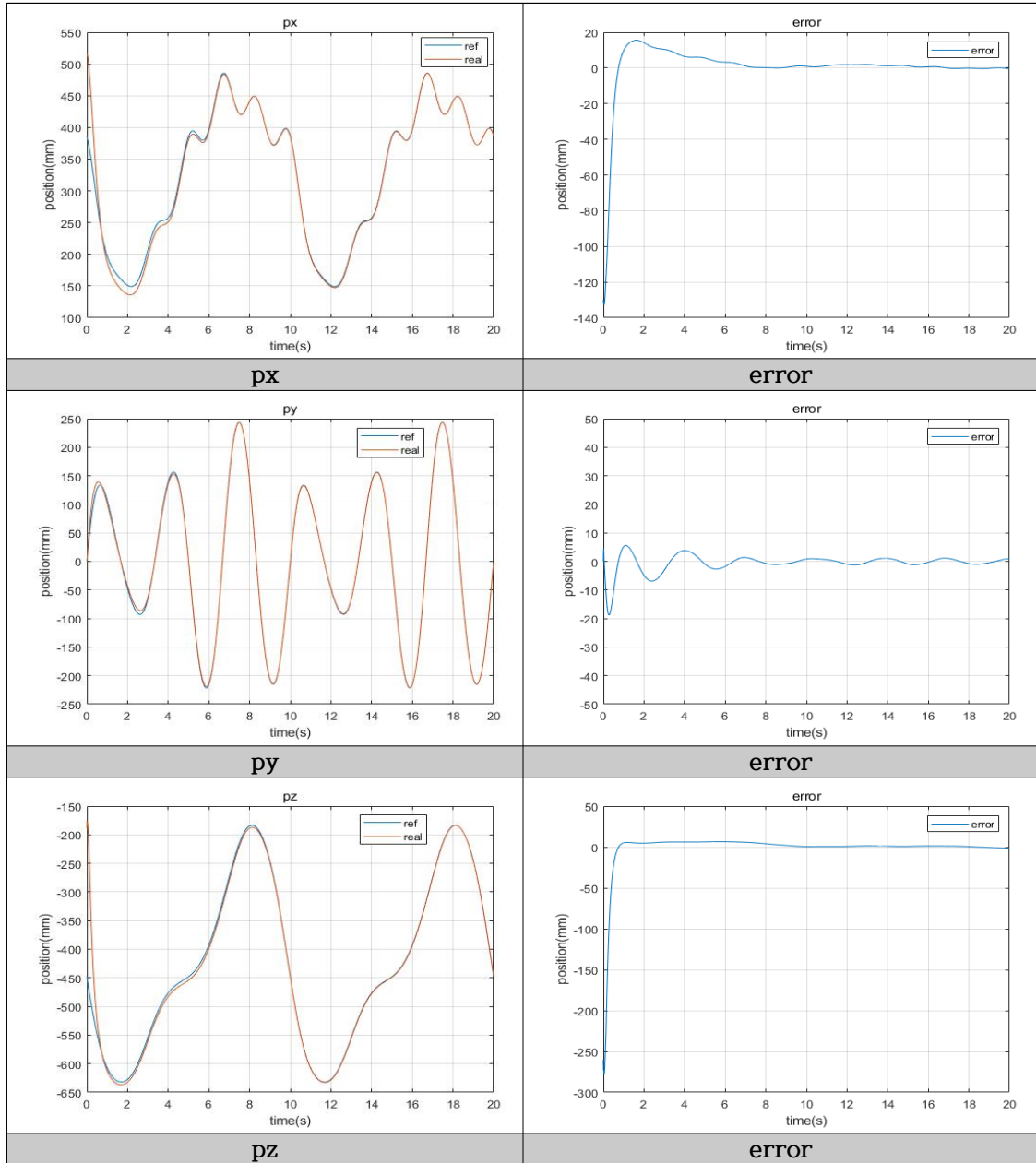


Table 3-17 기준 경로 추적 결과 (PID 제어기 적용 후)



시뮬레이션 결과, 제어기를 탑재하지 않은 상황과 비교하여 PID 제어기를 탑재하였을 때 오차가 5mm 이내로 발생하며, 기준 경로 추적을 원활하게 수행함을 알 수 있다. 따라서 본 논문에서 사용한 PID 제어기로 매니플레이터의 자세를 제어할 수 있으며 원하는 기준 경로를 추적할 수 있음을 확인하였다.

3.2.2 Sliding Mode 제어기 설계

본 논문에서 사용된 슬라이딩 모드 제어기의 블록선도는 Fig. 3-19와 같다.

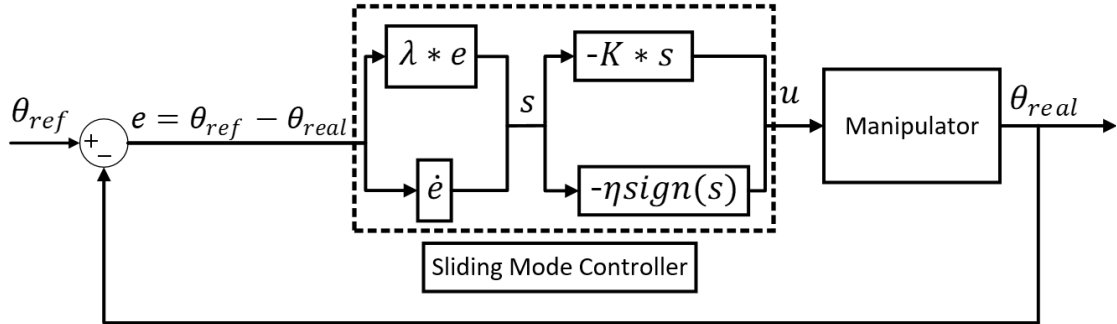


Fig. 3-19 Sliding Mode 제어기 블록선도

슬라이딩 모드 제어기의 작동 원리는 3.2.1절의 PID 제어기와 동일하다. 본 논문에서 사용된 각 관절의 슬라이딩 모드 제어기의 파라미터는 Table 3-18과 같다.

Table 3-18 Sliding Mode 제어기 파라미터

Joint	K	λ	η
1	0.01	50	0.001
2	0.01	50	0.001
3	0.01	50	0.001

제어기의 성능을 검증하기 위해 Table 3-14와 같은 기준 경로를 설정하고 시뮬레이션을 수행하였다. 제어기 적용 전 결과는 Table 3-19와 같고, 적용 후 결과는 Table 3-20과 같다.

Table 3-19 기준 경로 추적 결과 (Sliding Mode 제어기 적용 전)

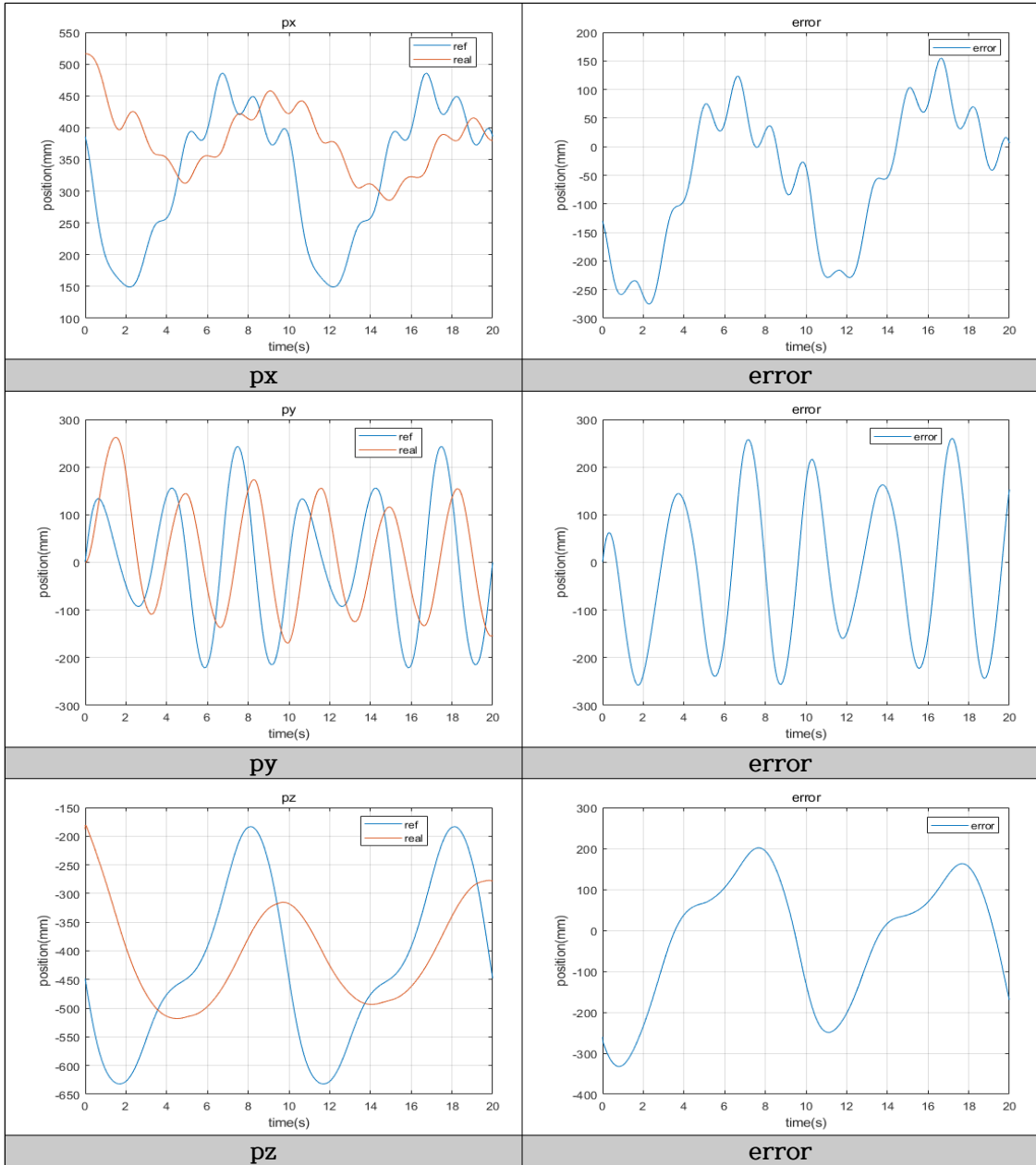
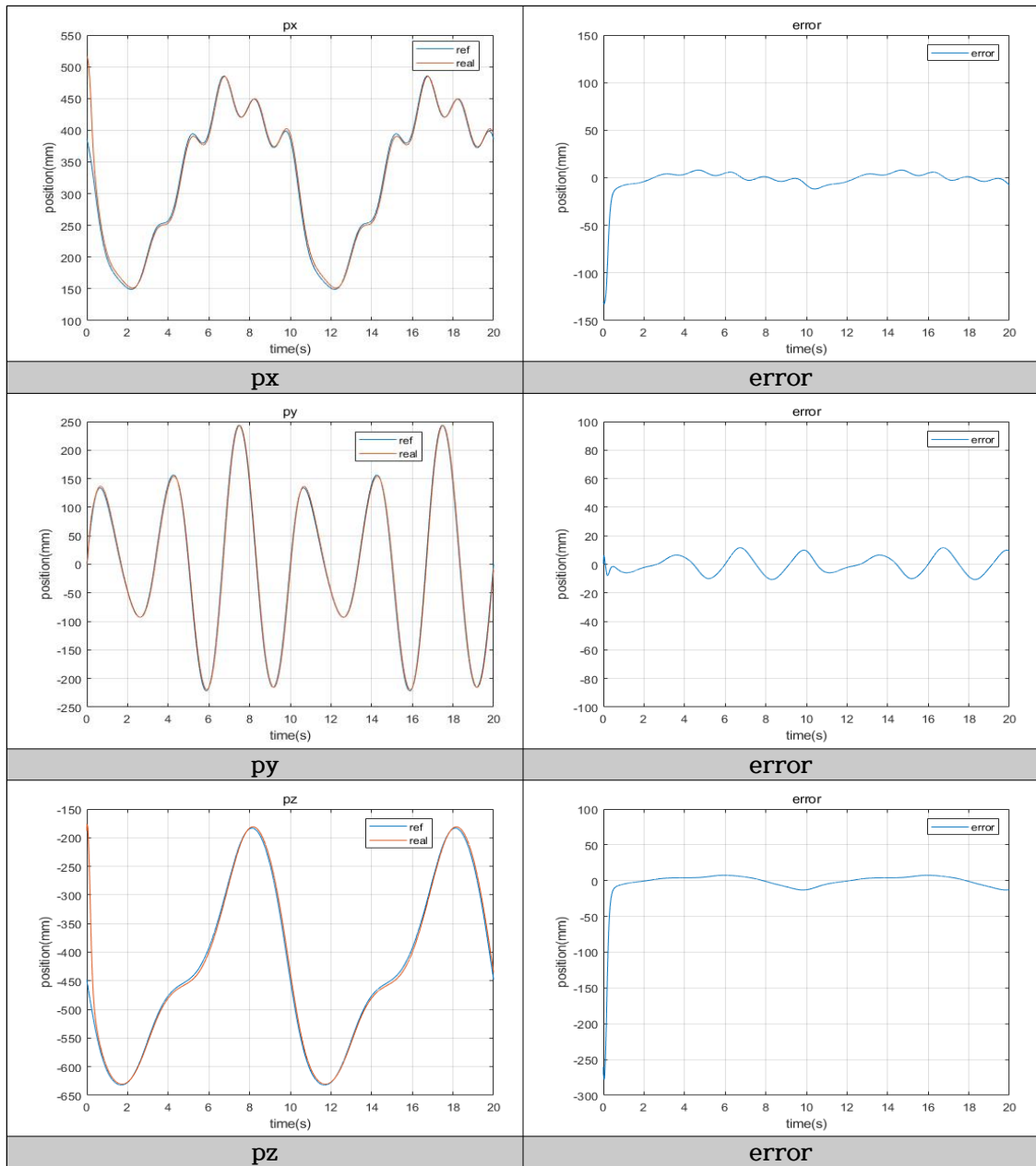


Table 3-20 기준 경로 추적 결과 (Sliding Mode 제어기 적용 후)



시뮬레이션 결과, PID 제어를 탑재한 경우와 마찬가지로 제어를 탑재하지 않은 상황과 비교하여 Sliding Mode 제어를 탑재하였을 때 기준 경로 추적을 원활하게 수행함을 알 수 있다. 따라서 앞서 설계한 PID 제어기, Sliding Mode 제어기 모두 기준 경로 추적을 잘 하고 있는 것을 확인하였으며, 원하는 매니플레이터의 동작을 수행할 수 있음을 확인하였다.

3.3. 고장 진단 및 복구 알고리즘 설계

3.3.1 일시적 고장 복구 알고리즘 설계

본 논문에서는 센서의 노이즈 저감을 위해 Kalman Filter를 활용하여 일시적인 고장 문제를 해결하고 센서 신호를 복구하였다. 노이즈가 발생하였을 때 복구된 신호는 기존의 신호보다 지연되어 출력되기 때문에 지연시간에 의한 오진단이 발생할 수 있다. 따라서 적절한 허용 지연시간을 설정하여 오진단을 방지해야 하며 본 논문에서는 0.005초를 허용 지연시간으로 설정하고 Kalman Filter를 적용하였다. 시스템 모델에 사용된 초기값 및 파라미터는 Table 3-21과 같으며, 적용된 수식은 식 (57) ~ (61)과 같다.

Table 3-21 Kalman Filter 초기값 및 파라미터

\hat{x}_0	P_0	A	H	Q	R
0	0.05	1	1	0.01	10

$$\hat{x}_k^- = \hat{x}_{k-1} \quad (57)$$

$$P_k^- = P_{k-1} + 0.01 \quad (58)$$

$$K_k = P_k^- (P_k^- + 10) \quad (59)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - \hat{x}_k^-) \quad (60)$$

$$P_k = P_k^- - K_k P_k^- \quad (61)$$

첫 번째 관절의 노이즈 발생 시 Kalman Filter를 적용하지 않은 값과 적용한 후의 값, 노이즈가 존재하지 않을 때의 센서 값을 측정하여 비교하였으며, Kalman Filter 부분의 Simulink 구성은 Fig. 3-20, 노이즈는 Table 3-22와 같다. 시뮬레이션 결과는 Table 3-23, 3-24, 3-25와 같다.

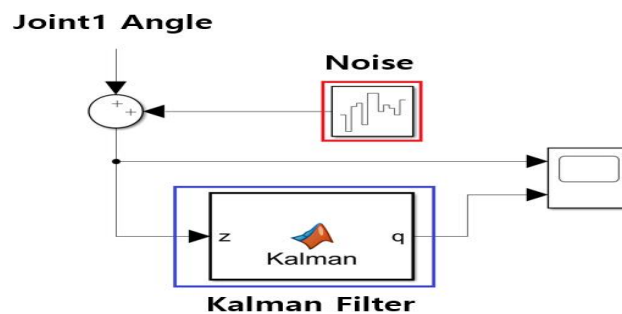


Fig. 3-20 일시적 고장 복구 알고리즘 구성

Table 3-22 노이즈 파라미터

Noise Power	Sample time	Seed
0.00001	0.0001	30000

Table 3-23 시뮬레이션 결과 (노이즈 상황)

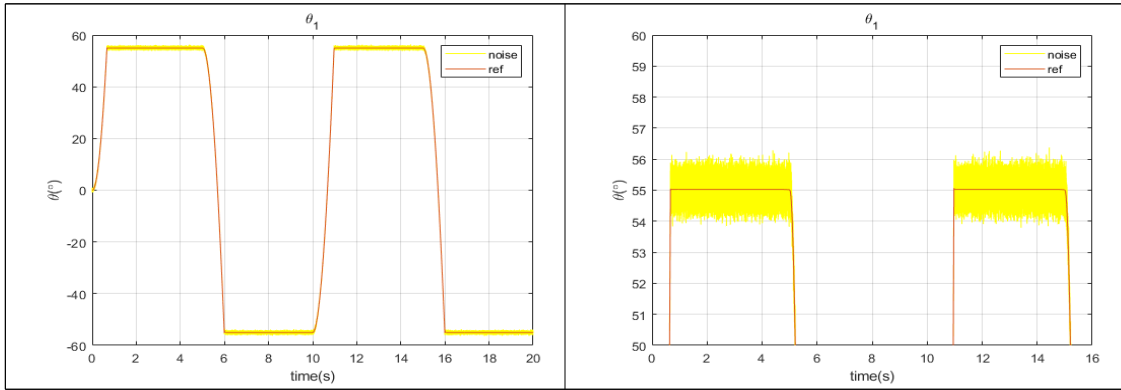


Table 3-24 시뮬레이션 결과 (칼만 필터 적용 후)

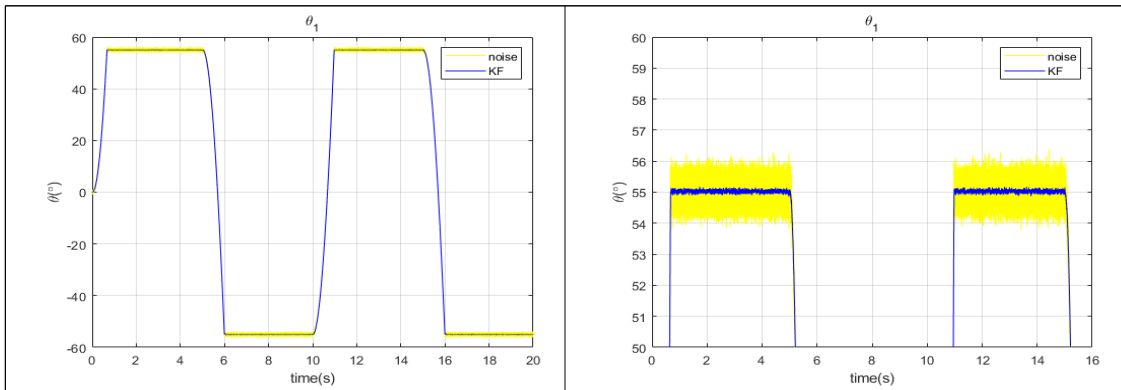


Table 3-25 시뮬레이션 결과 (기준 경로 vs 칼만 필터 적용 결과)

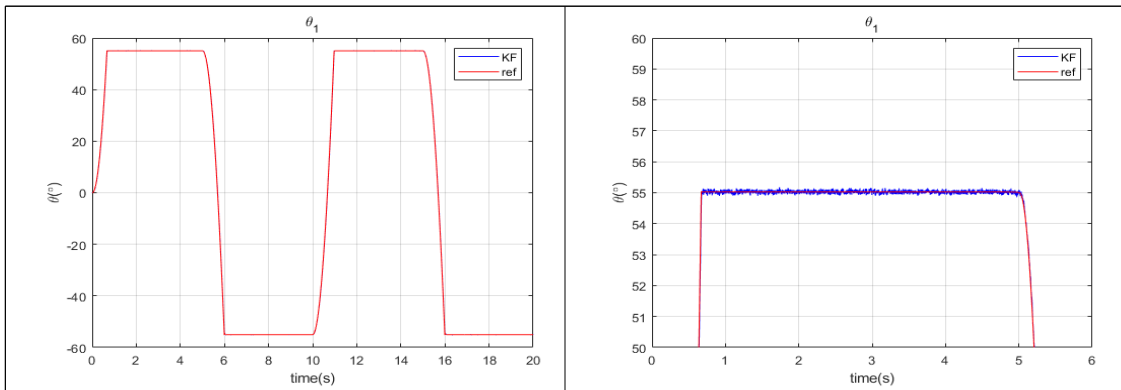


Table 3-23, 24, 25의 오른쪽 그래프는 왼쪽의 그래프는 확대한 그림이다. 인가된 노이즈는 Kalman Filter로 복구할 수 있음을 그래프로 확인할 수 있다. 0.003초의 지연시간이 발생하였지만 허용가능한 지연시간이기 때문에 Kalman Filter로 일시적인 고장을 복구할 수 있음을 확인하였다.

3.3.2 영구적 고장 복구 알고리즘 설계

본 논문에서는 센서에 영구적 고장 발생 시 Deep Learning 알고리즘을 활용하여 영구적인 고장 문제를 해결하고 센서 신호를 복구하였다. 매니퓰레이터의 제어가 작동할 때 Simulink의 블록을 활용하여 단선/단락, 고착 등 고장 상황을 구성하였으며, 학습된 데이터를 바탕으로 예측 값을 출력해 고장 상황을 극복하였다.

학습 데이터는 매니퓰레이터가 PID 제어기, Sliding Mode 제어기로 제어될 때 각 관절별 Input Signal, A Port Pressure, B Port Pressure, Joint Angle 총 12개의 값을 사용하였다. 각 관절은 사인파로 동작하며 Training case는 Table 3-26과 같다.

Table 3-26 Case별 학습 데이터

Case	Joint 1	Joint 2	Joint 3
1	$30\sin(0.2\pi)$	$30\sin(0.2\pi)$	$30\sin(0.2\pi)$
2	$30\sin(0.3\pi)$	$30\sin(0.3\pi)$	$30\sin(0.3\pi)$
3	$30\sin(0.4\pi)$	$30\sin(0.4\pi)$	$30\sin(0.4\pi)$
4	$30\sin(0.5\pi)$	$30\sin(0.5\pi)$	$30\sin(0.5\pi)$

학습 데이터 중 Joint1의 각도를 최종 예측 값으로 설정하였다. Joint1의 각도를 제외한 나머지 11개의 데이터를 학습 데이터로 구성하고 정답은 Joint1의 각도로 설정하고 학습을 진행하였다. 학습 성능을 검증하기 위한 Test Case는 Table 3-27과 같다.

Table 3-27 검증 데이터

Case	Joint1	Joint2	Joint3
1	$30\sin(0.6\pi)$	$30\sin(0.6\pi)$	$30\sin(0.6\pi)$

실제 값과 예측 값을 비교하기 위해 Simulink 상에서 고장 진단 및 복구 알고리즘을 적용한 구성은 Fig. 3-21과 같다.

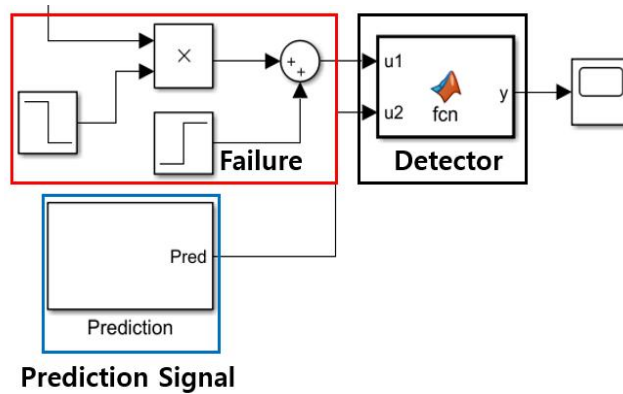


Fig. 3-21 영구적 복구 알고리즘 구성

각 관절별로 단선/단락, 고착 상황을 10초에 인위적으로 인가하여 고장을 발생시킨다. 검출기에서는 실제 신호와 예측 신호를 비교하며 오차율이 10%를 넘어섰을 때 예측 신호를 검출기에서 출력하게 되고 출력된 값은 제어기의 입력이 되어 매니퓰레이터의 Position Tracking 을 수행한다. 결과 값은 각 관절별, 제어기별로 비교하여 나타내었으며, 기준 경로를 얼마나 잘 추적하는지에 정도를 나타낸다. 결과는 Table 3-28 ~ 33과 같다.

Table 3-28 시뮬레이션 결과 (PID, Joint 1)

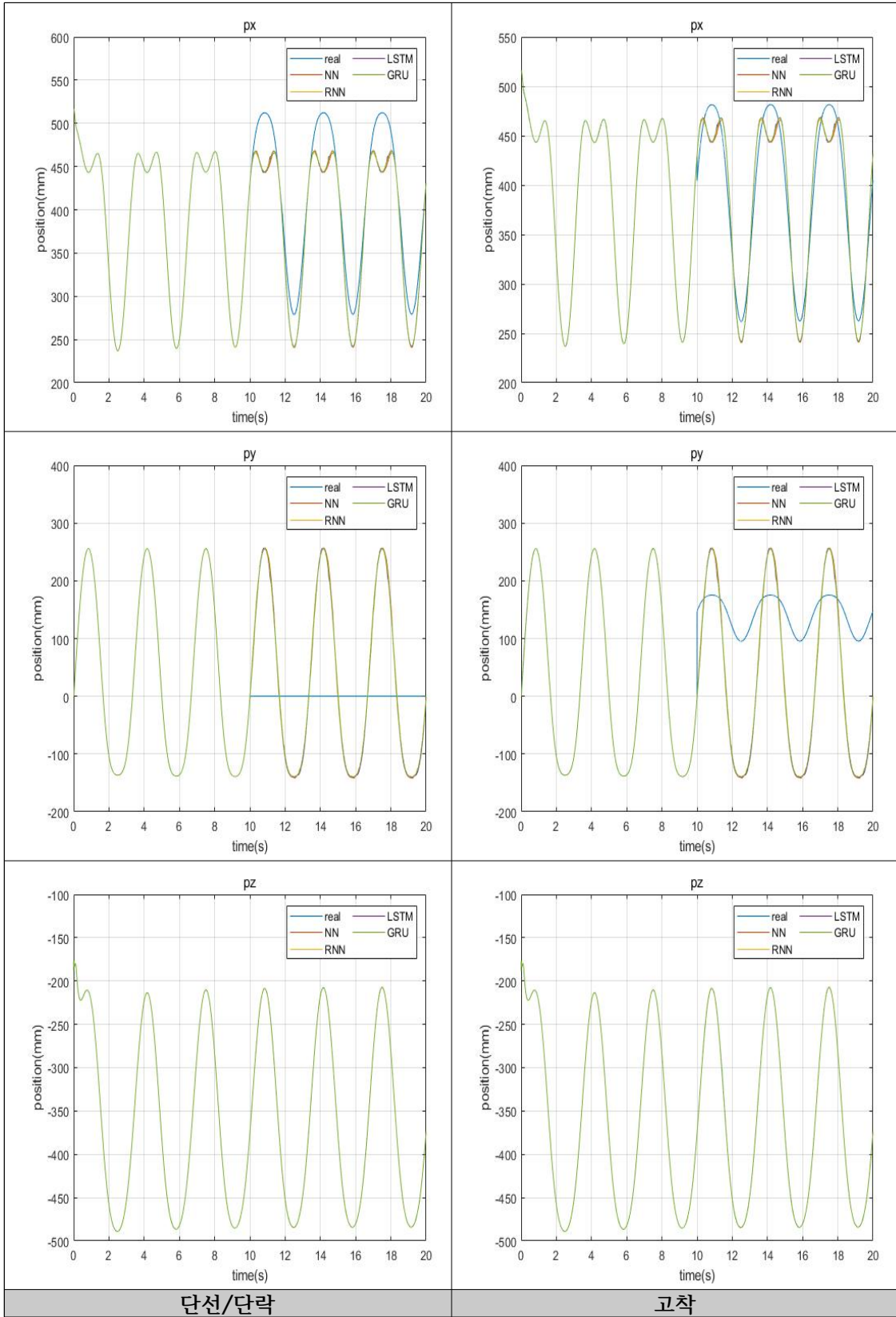


Table 3-29 시뮬레이션 결과 (PID, Joint 2)

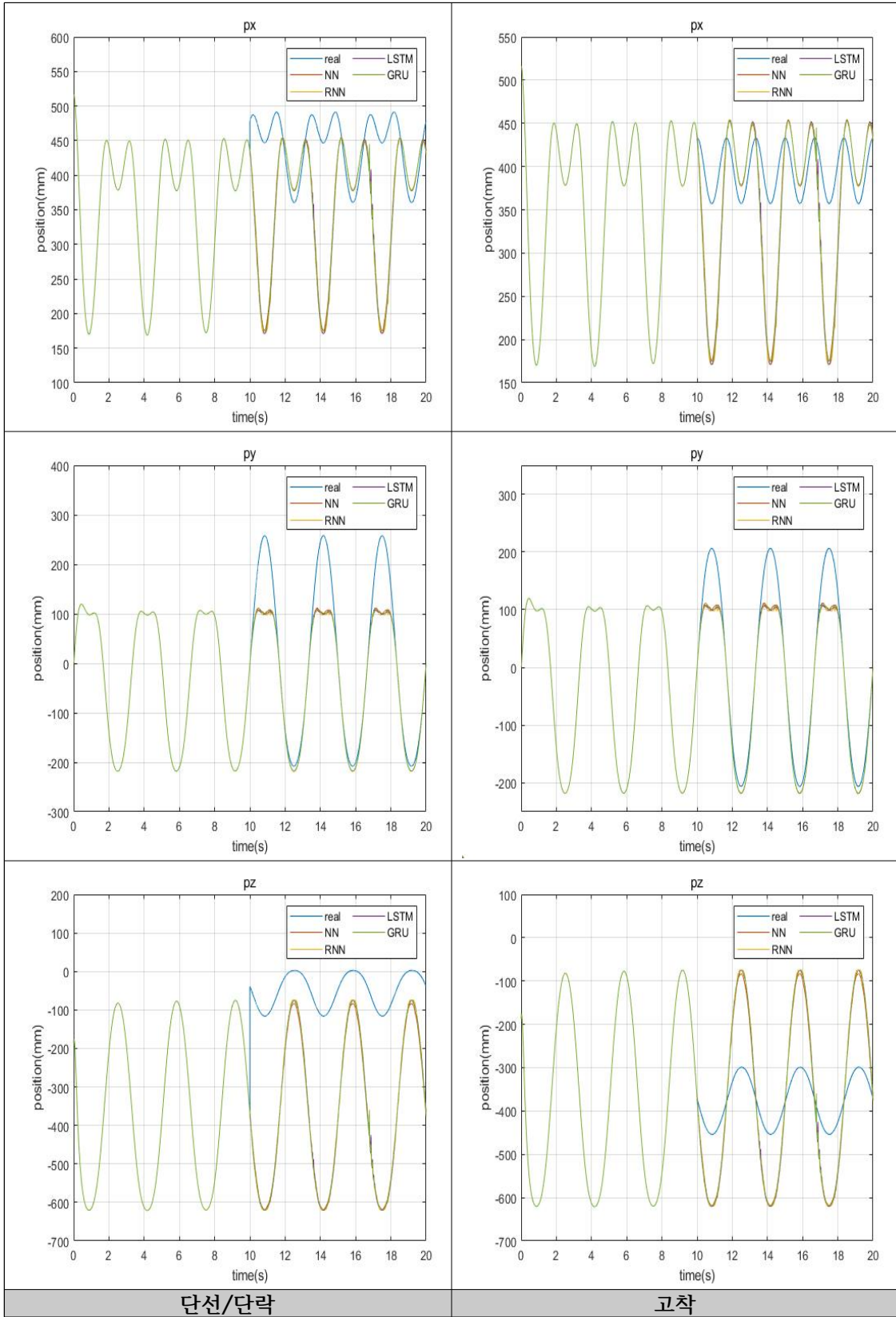


Table 3-30 시뮬레이션 결과 (PID, Joint 3)

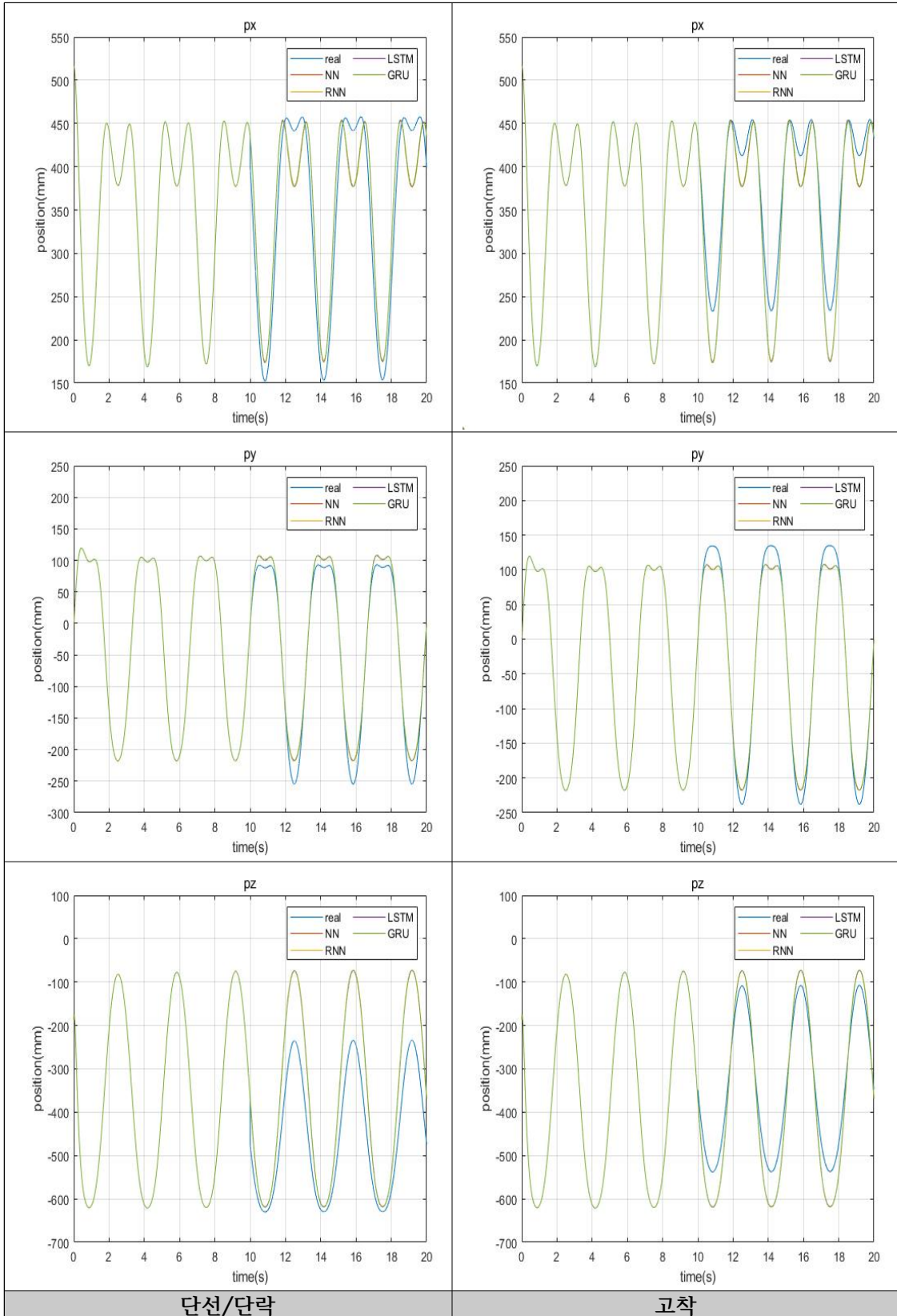


Table 3-31 시뮬레이션 결과 (SMC, Joint 1)

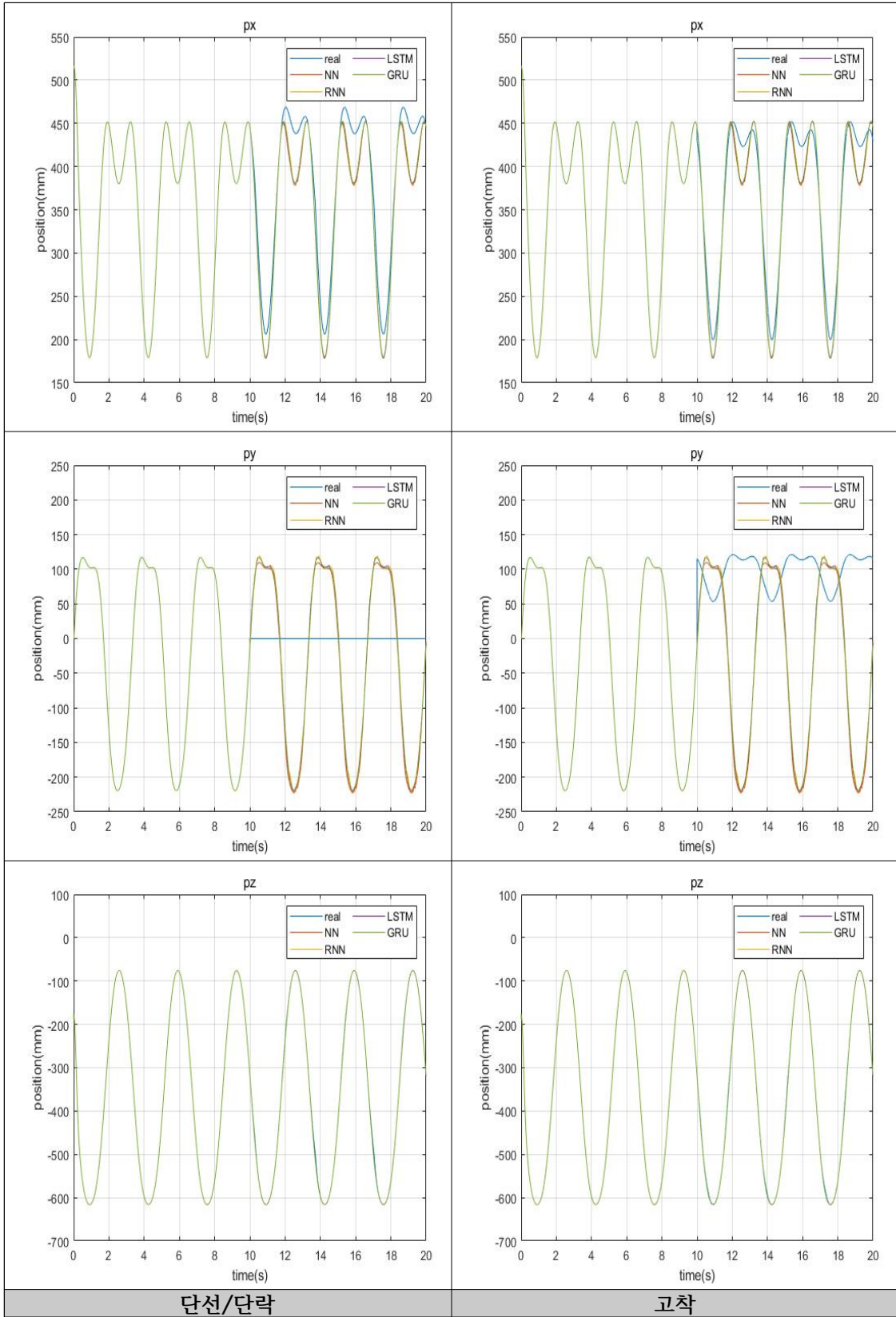


Table 3-32 시뮬레이션 결과 (SMC, Joint 2)

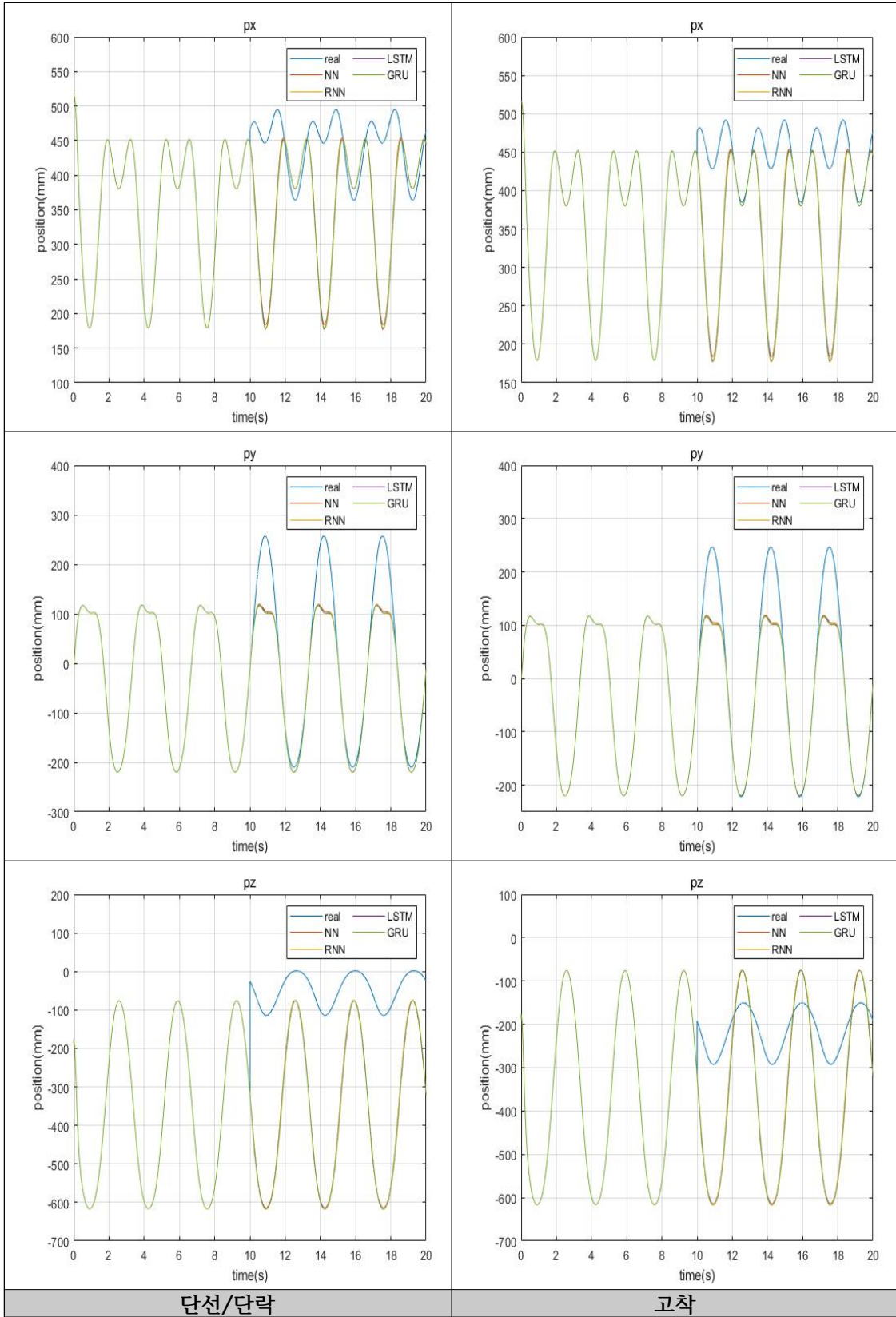
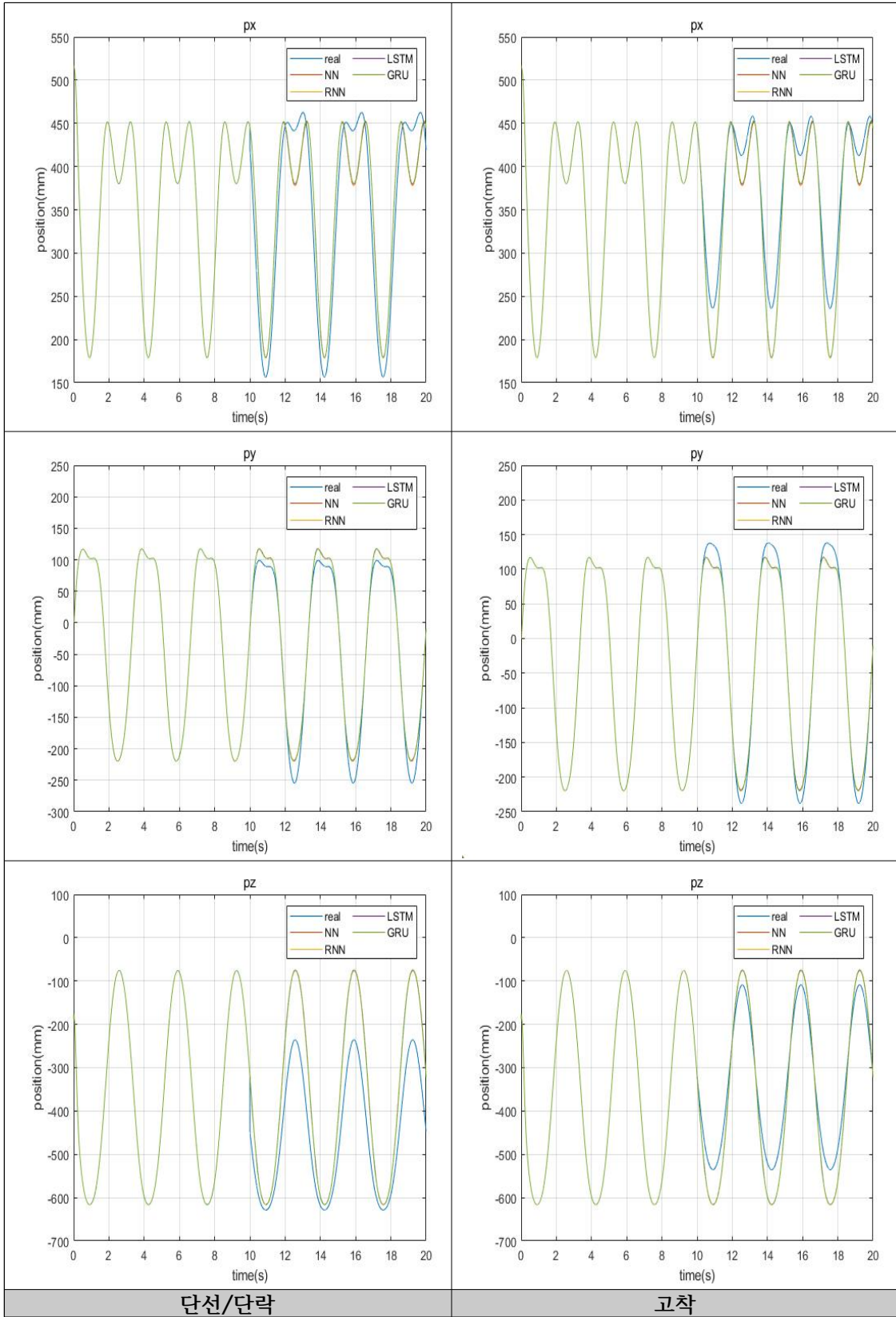


Table 3-33 시뮬레이션 결과 (SMC, Joint 3)



좌측의 결과는 단선/단락 상황에서, 우측의 결과는 고착 상황에서 실제 고장이 발생한 센서의 신호와 예측 신호를 비교한 그래프이다. 검출기에서는 실제 신호와 예측 신호를 실시간으로 비교하며 고장 상황 발생 시 실제 신호를 대신하여 예측 신호가 출력된다. 10초에 단선/단락, 고착 2개의 고장 상황이 발생하였으며, 오차율이 10%가 넘어가는 순간, 예측 신호로 매니플레이터의 Position Tracking이 수행된다. 고장 진단 및 복구 알고리즘이 적용되지 않은 경우에는 각 기준 경로별로 최대 200mm의 오차가 발생하였다. 고장 진단 및 복구 알고리즘을 적용한 경우에는 최대 10mm 이내의 오차가 발생하였으며, 기준 경로를 잘 추적함을 확인할 수 있다. 예측 신호는 NN, RNN, LSTM, GRU 네 개의 알고리즘으로 학습된 결과이며, 각 알고리즘 별로 차이는 존재하였으나 Deep Learning 알고리즘으로 예측한 결과 모두 고장 발생 상황에서 고장 복구가 성공적으로 이루어짐을 확인하였다.

따라서 고장 진단 및 복구 알고리즘을 사용하여 고장 상황이 발생하였을 때에도 기준 경로를 잘 추적함을 확인하였으며, 본 논문에서 제안한 알고리즘이 센서의 고장 신호를 복구할 수 있음을 검증하였다.

4. 실차 실험 및 결과

3장에서 검증된 고장 진단 및 복구 알고리즘을 실제 유압 매니플레이터에 적용한다. PID 제어기, 슬라이딩 모드 제어기를 적용하여 경로 추적을 진행할 때 각 관절에 고장을 인가하고 알고리즘을 적용하여 복구 성능을 확인한다. 실제 매니플레이터 시스템에 고장 진단 및 복구 알고리즘을 적용하여 복구 결과를 도출하였다.

4.1. 시스템 개요

본 논문에서 사용된 3축 유압 매니플레이터는 유압 시스템, PC, 파워 서플라이, 컨트롤 박스로 구성되어 있으며 전체 구성은 Fig. 4-1, 개략도는 Fig. 4-2와 같다.

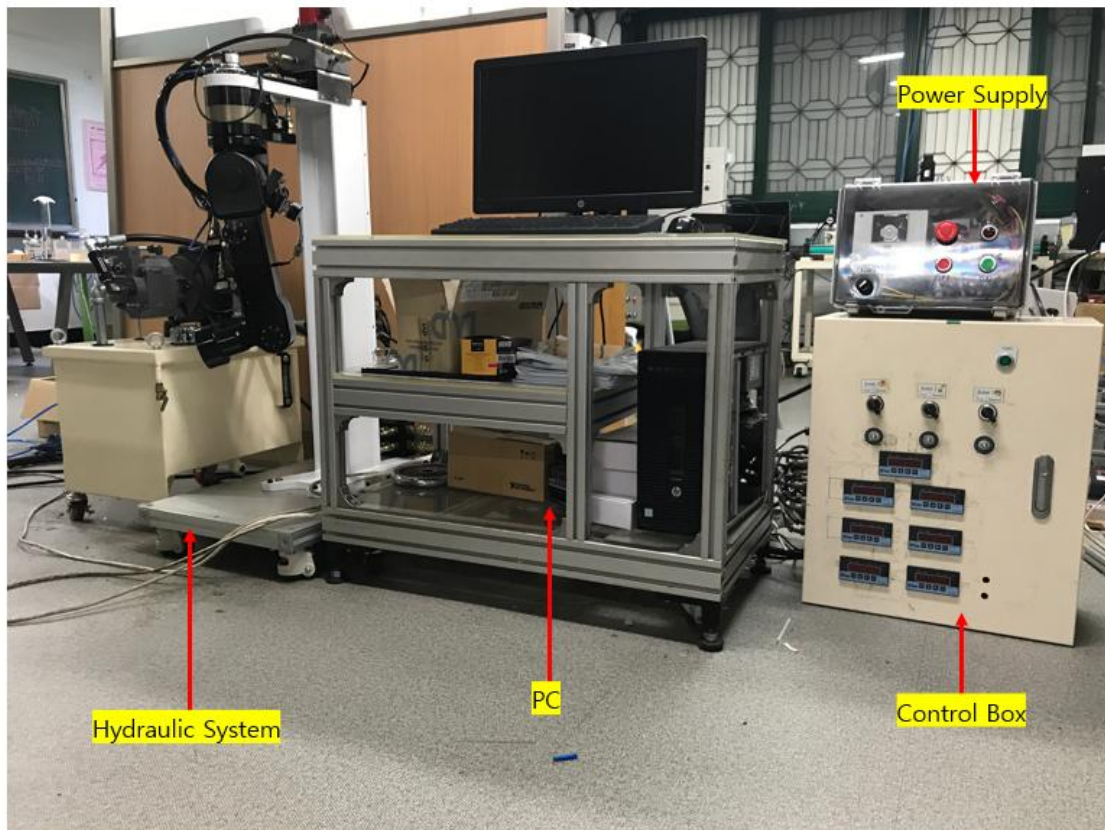


Fig. 4-1 시스템 구성

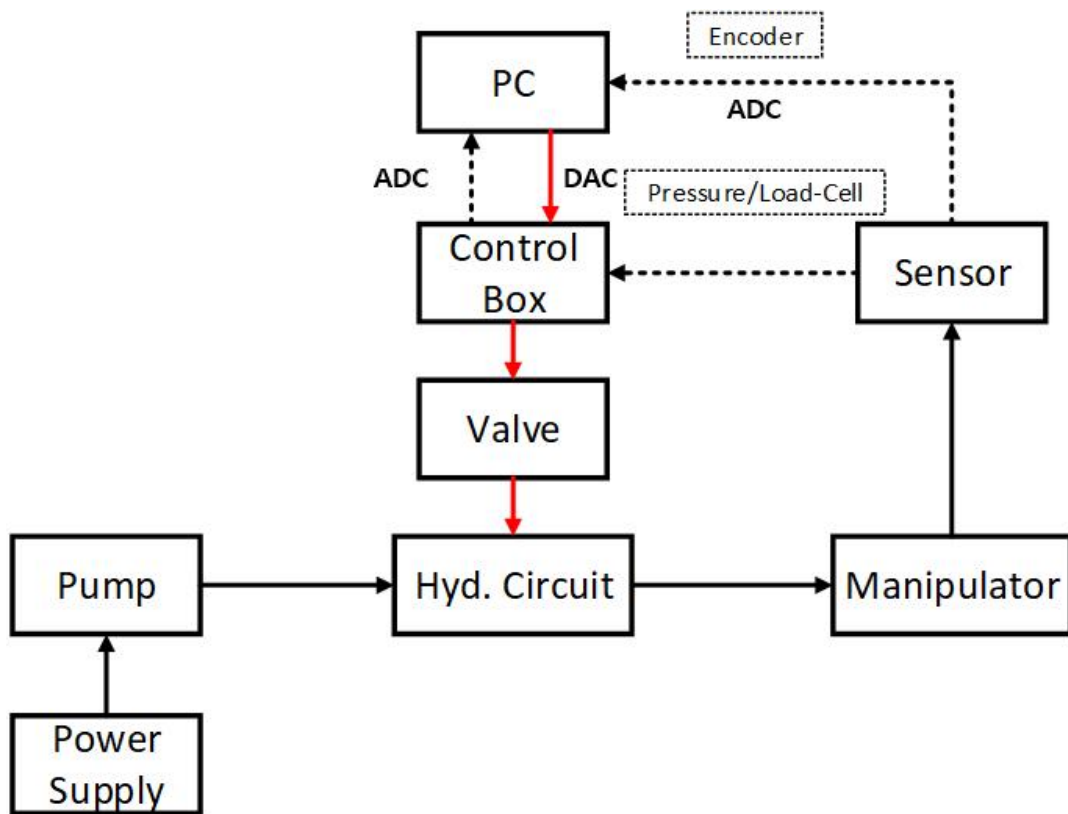


Fig. 4-2 시스템 개략도

파워 서플라이는 펌프에 신호를 인가하여 펌프가 회전하여 유량을 토출하도록 작동시킨다. PC에서 나온 디지털 신호는 컨버터를 통해 아날로그 신호로 변환되고 컨트롤 박스에 신호를 인가한다. 컨트롤 박스는 각 액추에이터에 연결된 서보 밸브를 제어하며 최종적으로 매니플레이터가 동작한다. 매니플레이터 장착된 압력 센서와 로드셀의 신호는 컨트롤 박스를 거쳐 PC로 입력되고 엔코더의 신호는 직접적으로 PC로 입력된다. 본 논문에서 사용한 매니플레이터의 부품 제원은 Table 4-1과 같고, 구성은 Fig. 4-3과 같다.

Table 4-1 매니플레이터 부품 제원

Component	Model
Motor	3.7kW-4P-(50)-60Hz-220/380V-M208008
Pump	P16V-RS-30-CC-21-K
Servo-valve 1	304F-40L-20-21101
Servo-valve 2, 3	KSV2.M070
Rotary Actuator 1, 2	Double-Vane-Motor-400Nm
Cylinder	Single-Rod-Cylinder-22-32kN

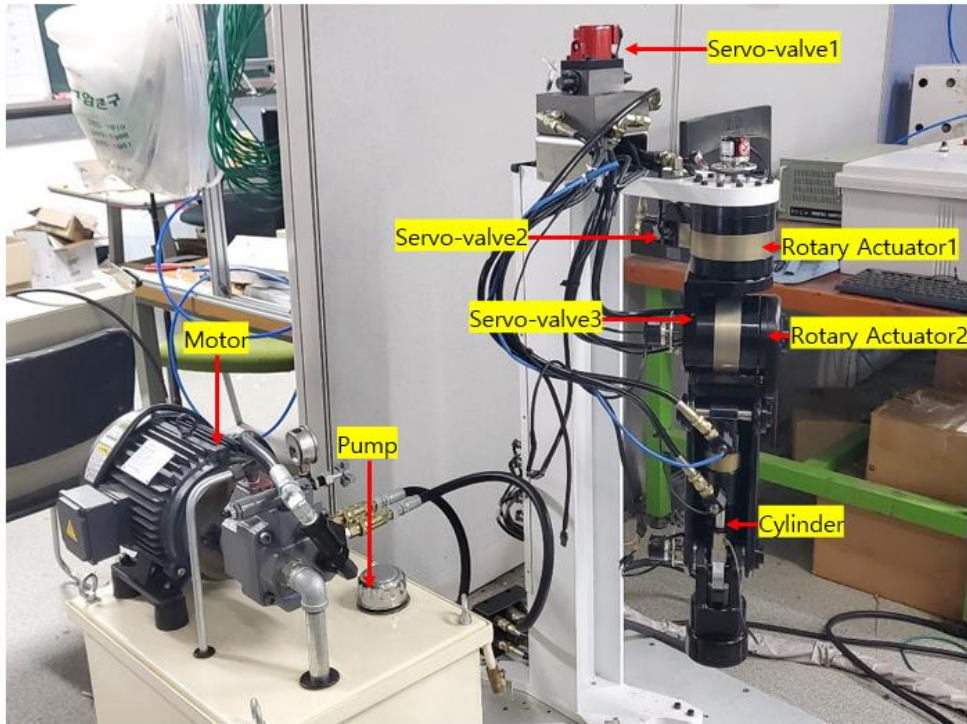
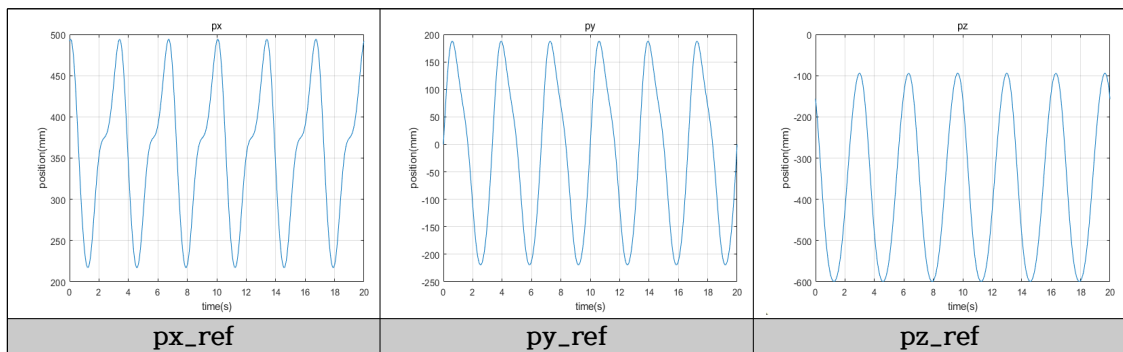


Fig. 4-3 유압 매니플레이터

4.2. 제어기 구축

본 절에서는 3장에서 수행한 방법과 동일하게 고장 진단 및 복구 알고리즘을 적용하기 전 제어기를 탑재하여 position tracking을 수행한다. 4.1절에서 소개한 3축 유압 매니플레이터에 앞서 시뮬레이션 상에서 검증한 PID 제어기, 슬라이딩 모드 제어기를 적용하여 경로 추적 성능을 확인한다. 기준 경로는 Table 4-2와 같다.

Table 4-2 기준 경로 그래프



4.2.1 PID 제어기 구축

본 논문의 실차 실험에서 사용된 각 관절의 PID 제어기의 파라미터는 Table 4-3과 같다.

Table 4-3 PID 제어기 파라미터

Joint1	$k_p = 2$	$k_i = 0$	$k_d = 0.1$
Joint2	$k_p = 2$	$k_i = 0.01$	$k_d = 0.001$
Joint3	$k_p = 0.02$	$k_i = 0.5$	$k_d = 0.005$

제어기의 성능을 검증하기 위해 Table 4-2와 같은 기준 경로를 설정하고 실험을 진행하였다. 제어기 적용 전 결과는 Table 4-4와 같고, 적용 후 결과는 Table 4-5와 같다.

Table 4-4 기준 경로 추적 결과 (PID 제어기 적용 전)

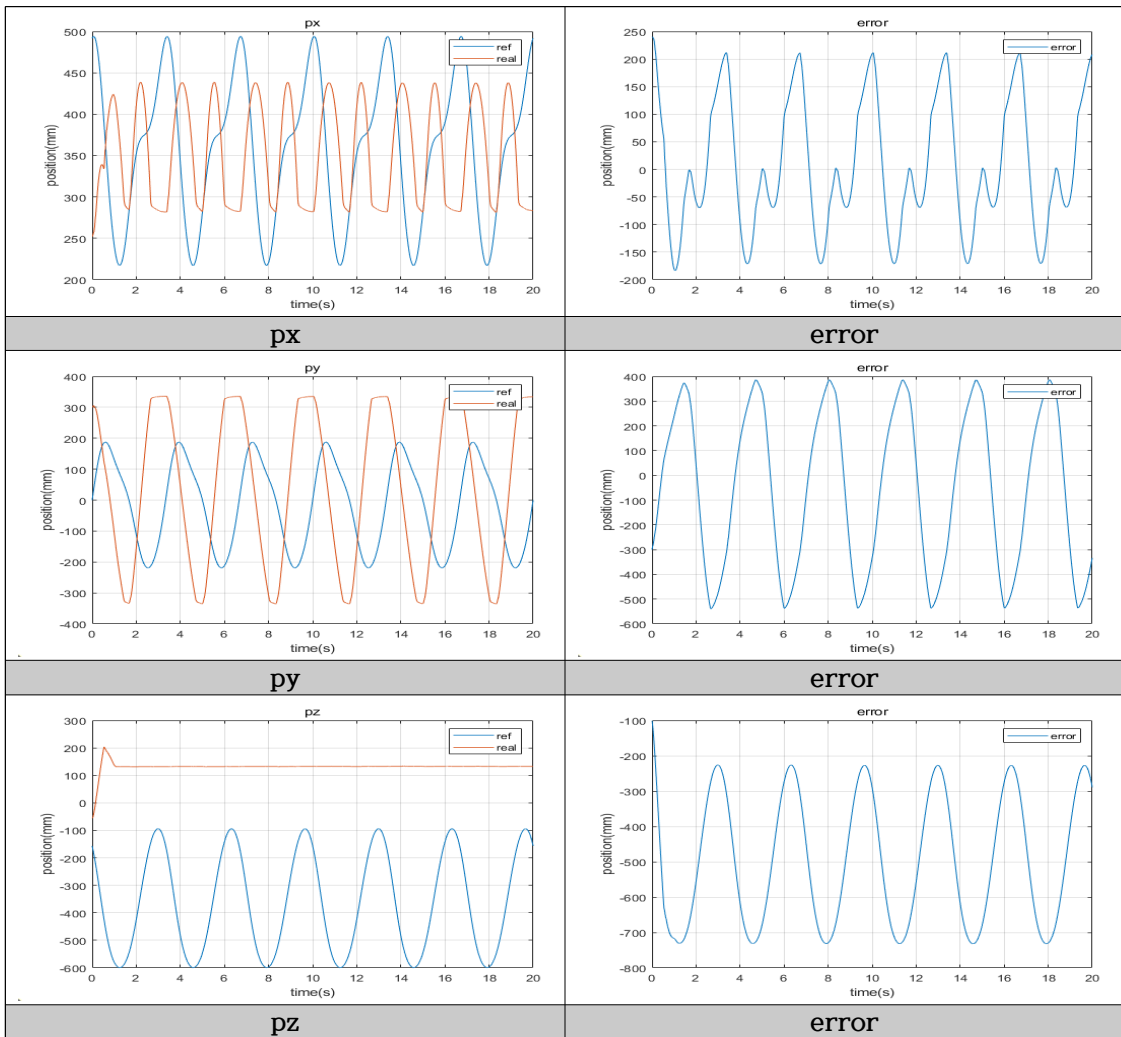
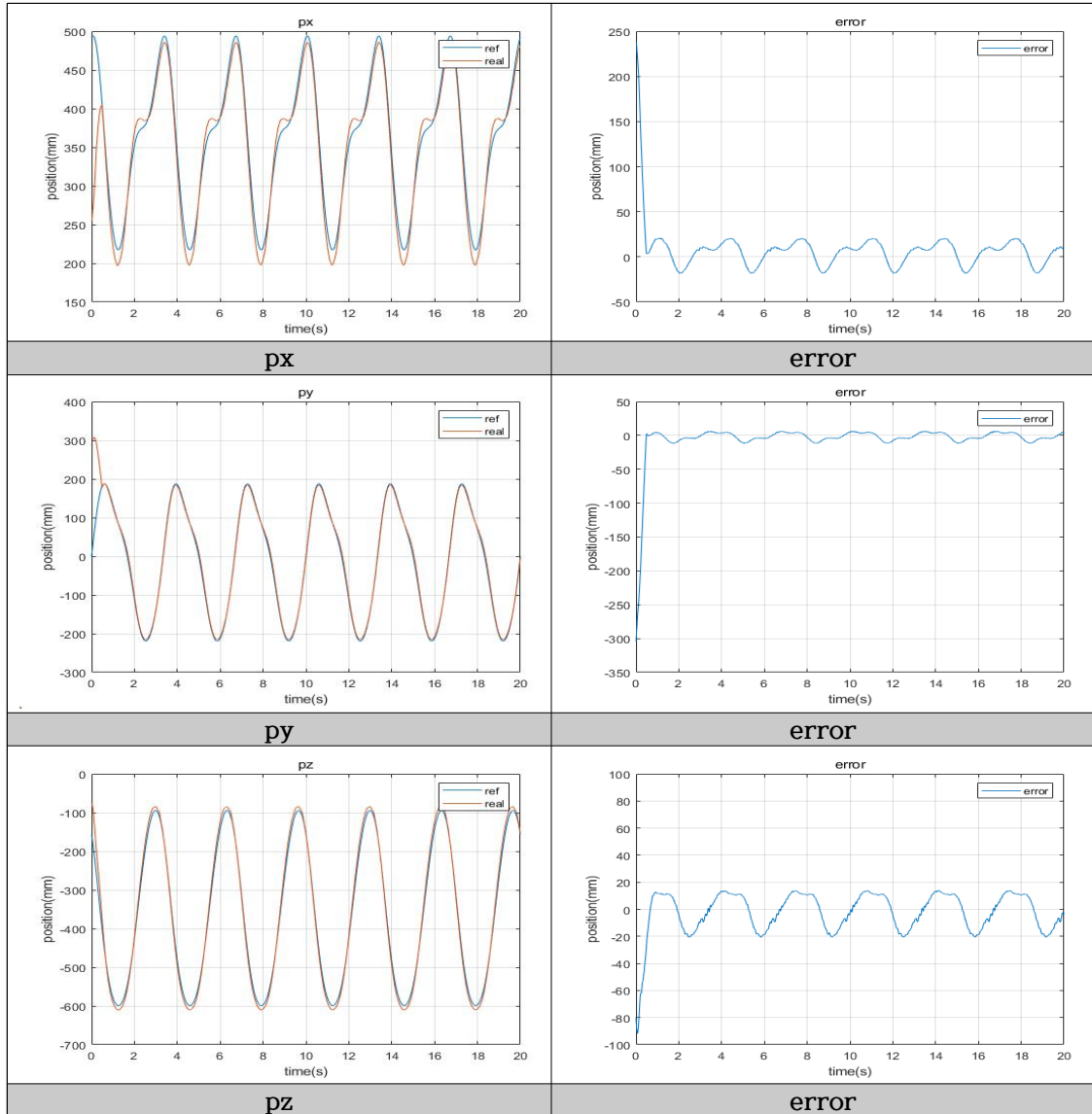


Table 4-5 기준 경로 추적 결과 (PID 제어기 적용 후)



실험 결과, 제어를 탑재하지 않은 상황과 비교하여 PID 제어를 탑재하였을 때 오차가 15mm 이내로 발생하며, 기준 경로 추적을 원활하게 수행함을 알 수 있다. 따라서 앞서 설계한 PID 제어기로 매니플레이터의 자세를 제어할 수 있으며 원하는 기준 경로를 추적할 수 있음을 확인하였다.

4.2.2 Sliding Mode 제어기 구축

본 논문에서 사용된 각 관절의 슬라이딩 모드 제어기의 파라미터는 Table 4-6과 같다.

Table 4-6 Sliding Mode 제어기 파라미터

Joint1	$K = 60$	$\lambda = 52.5$	$\eta = 2.5$
Joint2	$K = 50$	$\lambda = 45$	$\eta = 2.5$
Joint3	$K = 40$	$\lambda = 37.5$	$\eta = 1.5$

제어기의 성능을 검증하기 위해 Table 4-2와 같은 기준 경로를 설정하고 실험을 진행하였다. 제어기 적용 전 결과는 Table 4-7과 같고, 적용 후 결과는 Table 4-8과 같다.

Table 4-7 기준 경로 추적 결과 (Sliding Mode 제어기 적용 전)

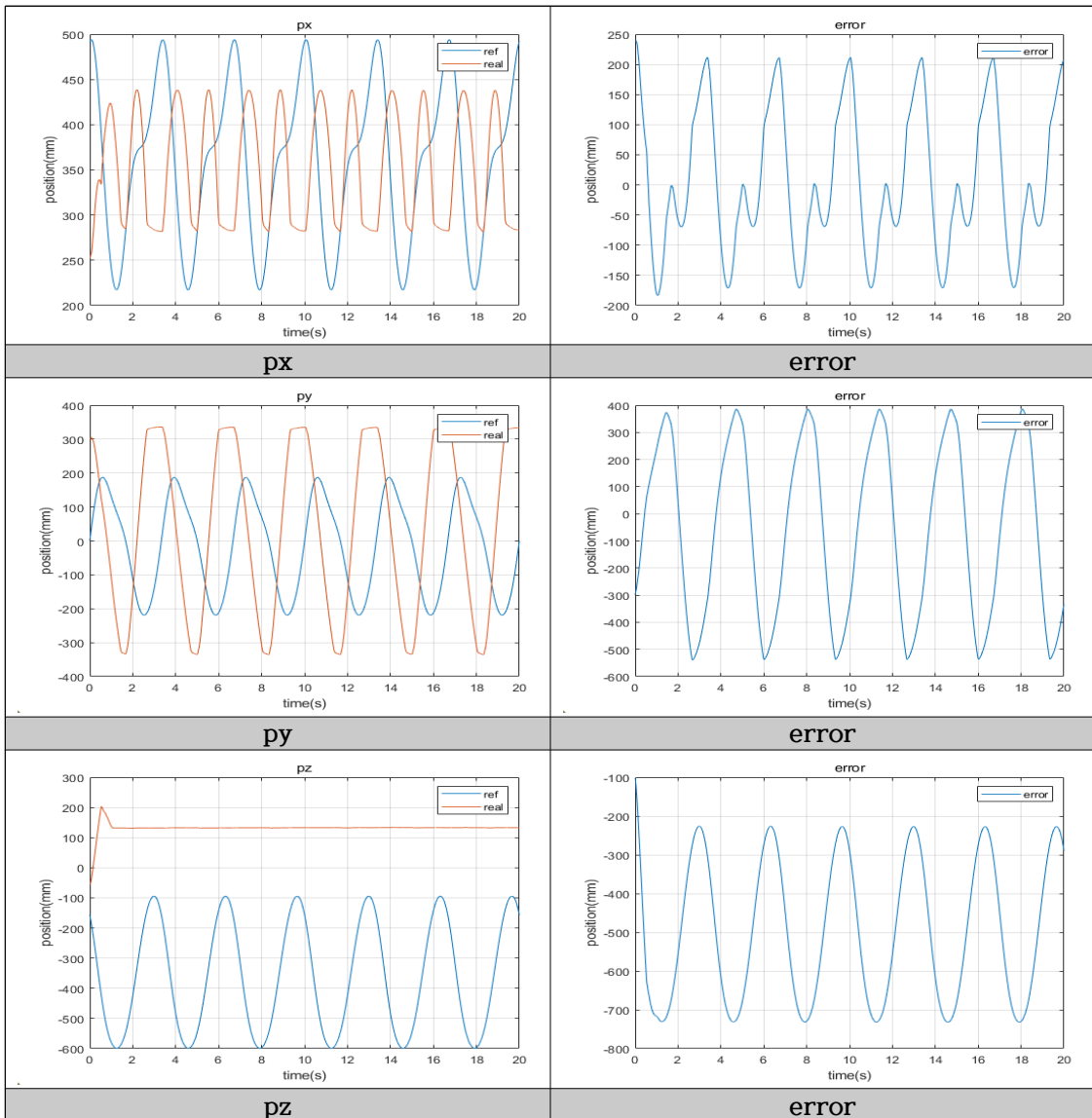
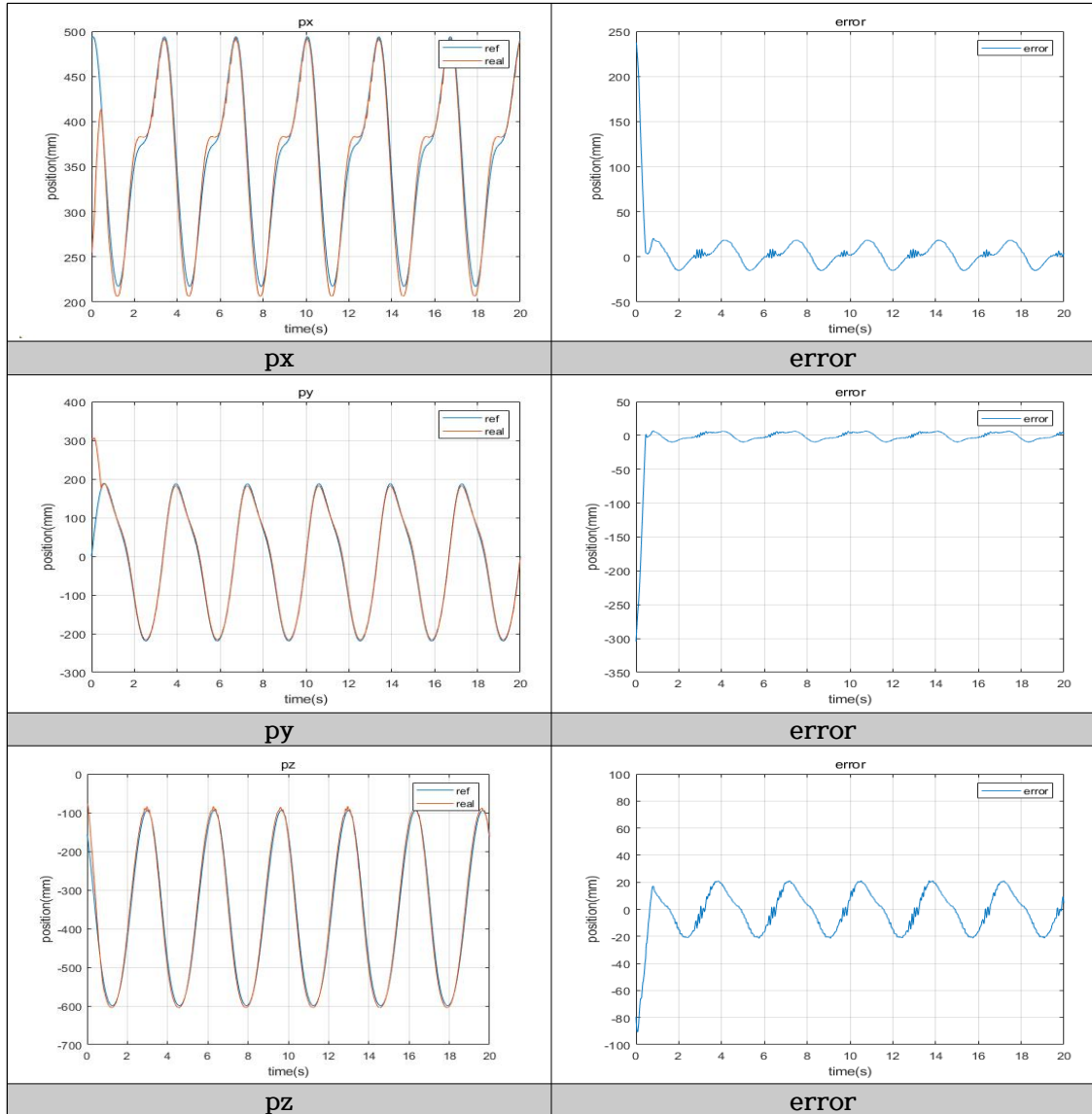


Table 4-8 기준 경로 추적 결과 (Sliding Mode 제어기 적용 후)



실차 실험 결과, 앞서 설계한 Sliding Mode 제어기가 실차 조건에서도 최대 20mm 내외의 오차를 보이며, 경로 추적 성능이 좋음을 확인하였다. 따라서 PID 제어기, Sliding Mode 제어기로 매니플레이터의 자세를 제어할 수 있으며 원하는 동작을 수행할 수 있음을 확인하였다.

4.3. 고장 진단 및 복구 알고리즘 구축

본 절에서는 시뮬레이션에서 검증된 알고리즘을 실제 유압 매니퓰레이터 시스템에 적용한다. PID 제어기, 슬라이딩 모드 제어기로 기준 경로를 추적할 때 고장 상황을 발생시켜 각 관절 센서의 신호를 복구한다.

Training Case는 Table 4-9, Test Case는 Table 4-10과 같다.

Table 4-9 Case별 학습 데이터

Case	Joint1	Joint2	Joint3
Case1	$30\sin(0.2\pi)$	$30\sin(0.2\pi)$	$30\sin(0.2\pi)$
Case2	$30\sin(0.3\pi)$	$30\sin(0.3\pi)$	$30\sin(0.3\pi)$
Case3	$30\sin(0.4\pi)$	$30\sin(0.4\pi)$	$30\sin(0.4\pi)$
Case4	$30\sin(0.5\pi)$	$30\sin(0.5\pi)$	$30\sin(0.5\pi)$

Table 4-10 검증 데이터

Case	Joint1	Joint2	Joint3
1	$30\sin(0.6\pi)$	$30\sin(0.6\pi)$	$30\sin(0.6\pi)$

실제 값과 예측 값을 비교하기 위해 실험 PC의 Simulink 내에서 고장 진단 및 복구 알고리즘을 적용한 모습은 Fig. 4-4와 같다.

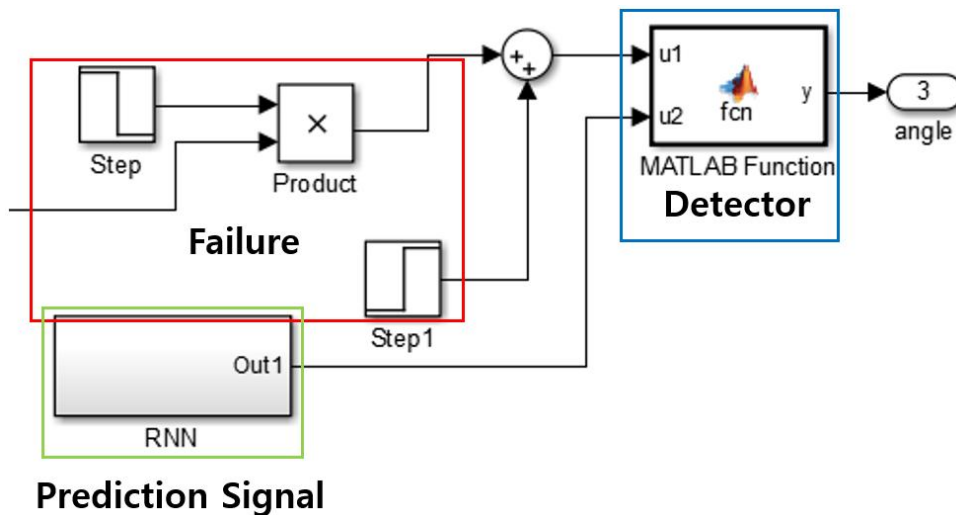


Fig. 4-4 고장 진단 및 복구 알고리즘 스케치

3장에서 고장 진단 및 복구 알고리즘을 적용한 것과 같이 실차 실험에서도 같은 방식으로 진행하였으며, 그 결과는 Table 4-11 ~ 16과 같다.

Table 4-11 실차 실험 결과 (PID, Joint 1)

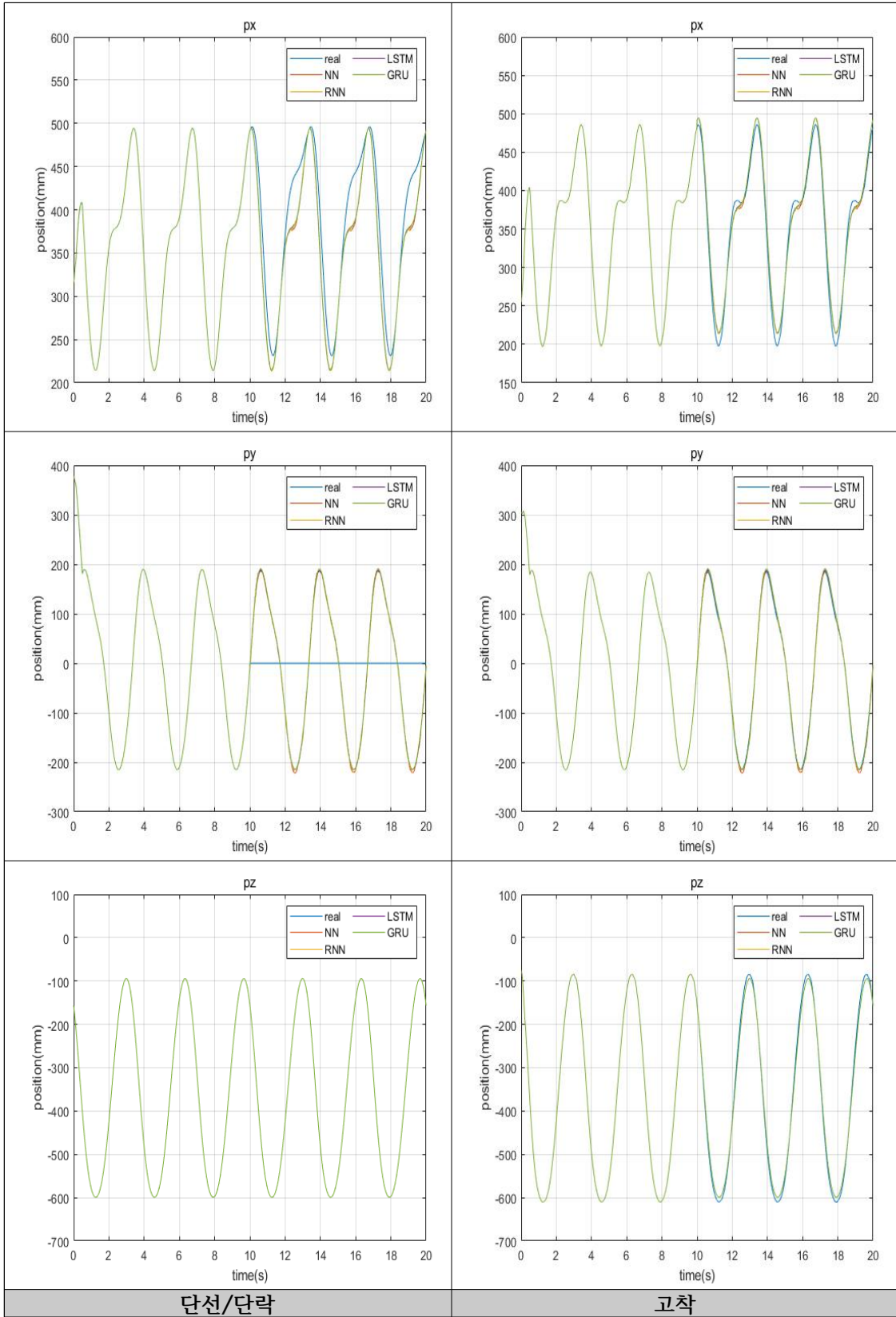


Table 4-12 실차 실험 결과 (PID, Joint 2)

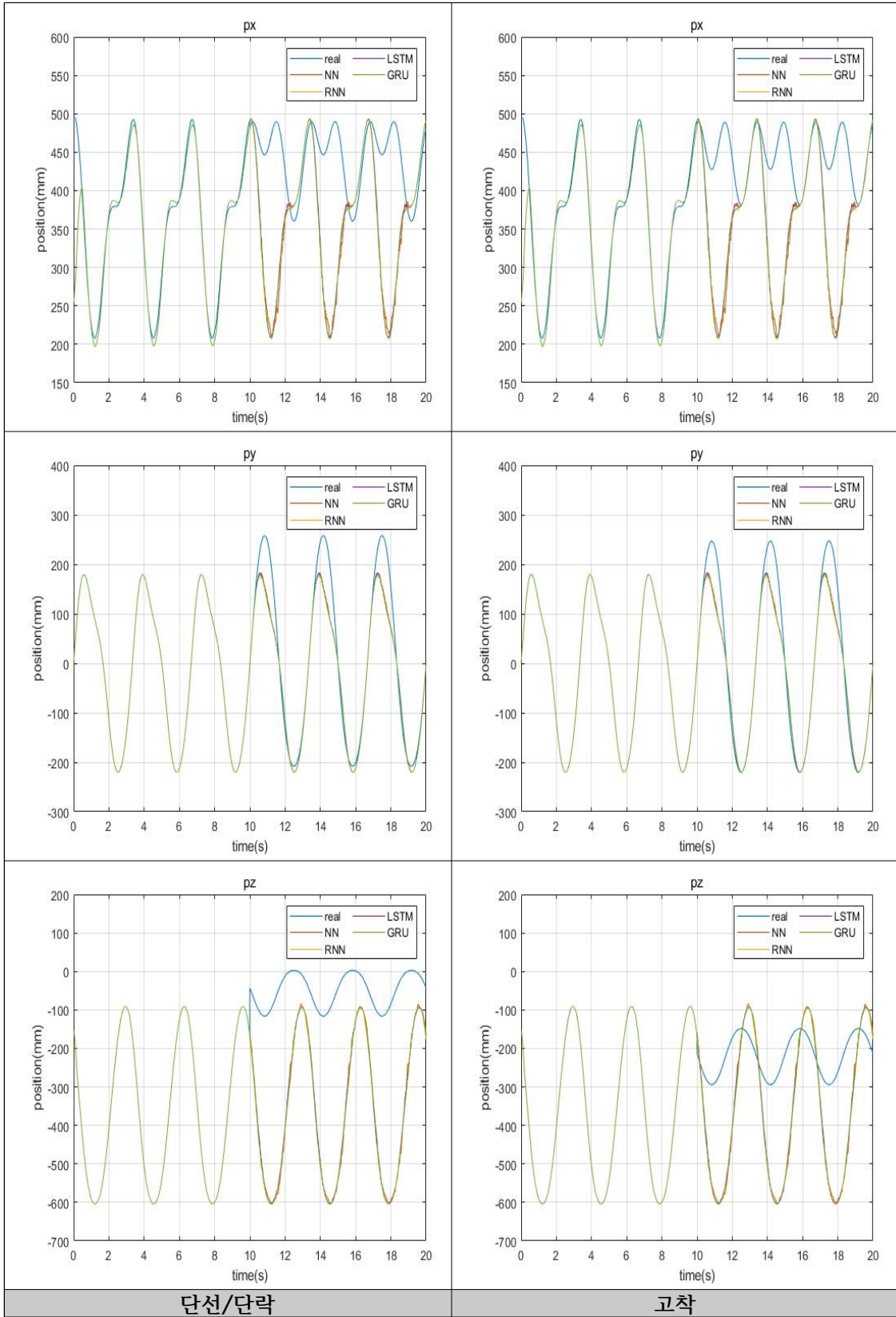


Table 4-13 실차 실험 결과 (PID, Joint 3)

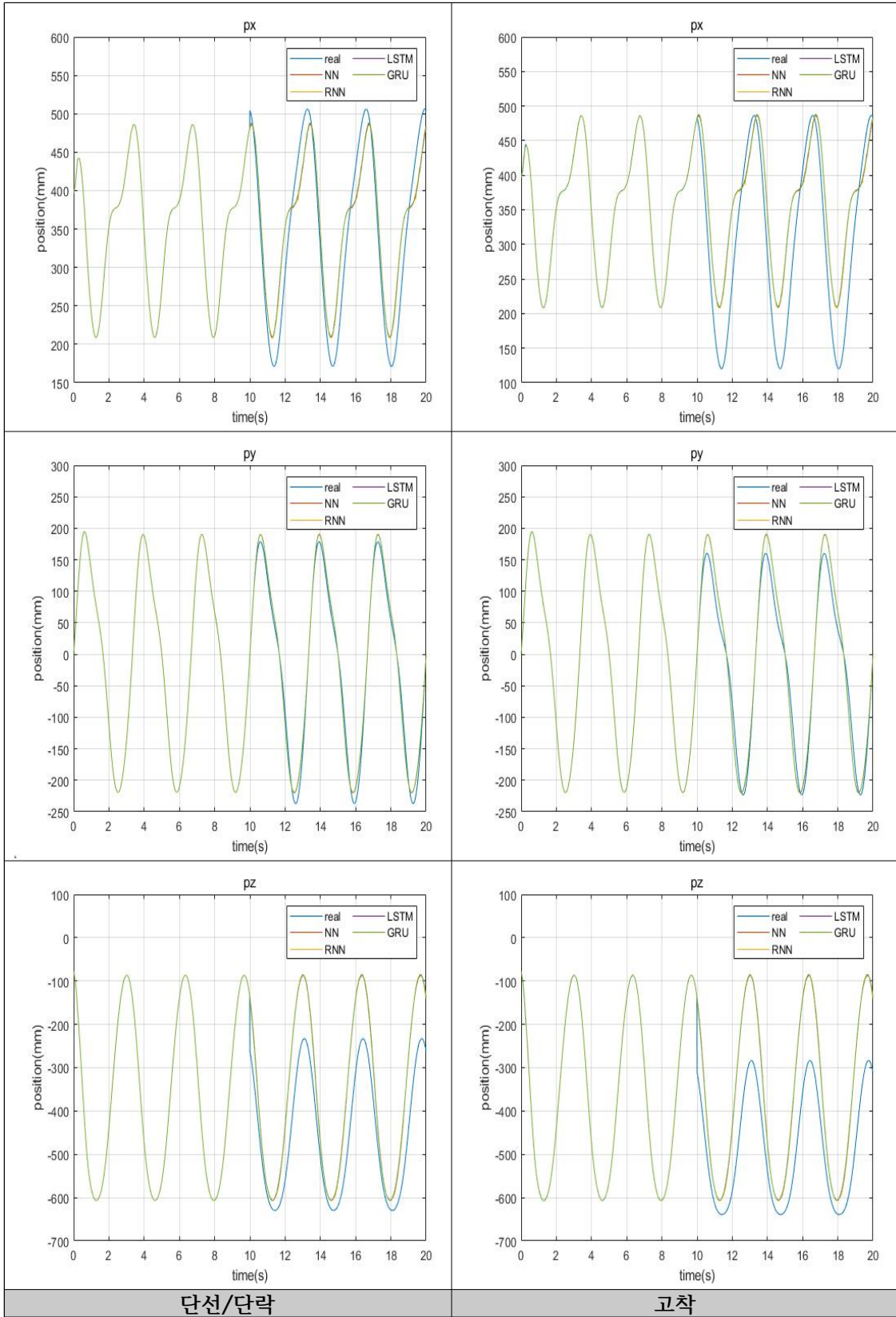


Table 4-14 실차 실험 결과 (SMC, Joint 1)

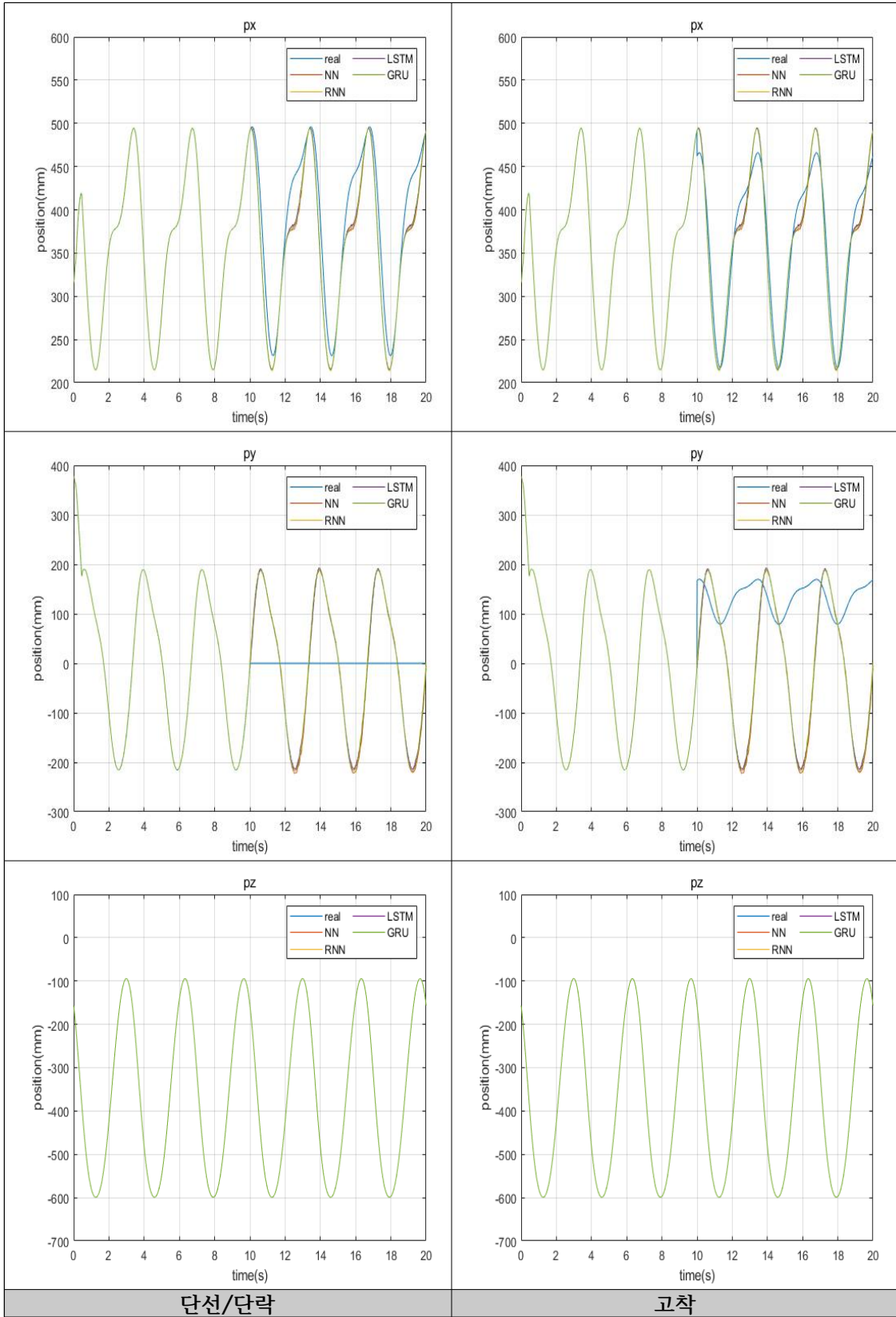


Table 4-15 실차 실험 결과 (SMC, Joint 2)

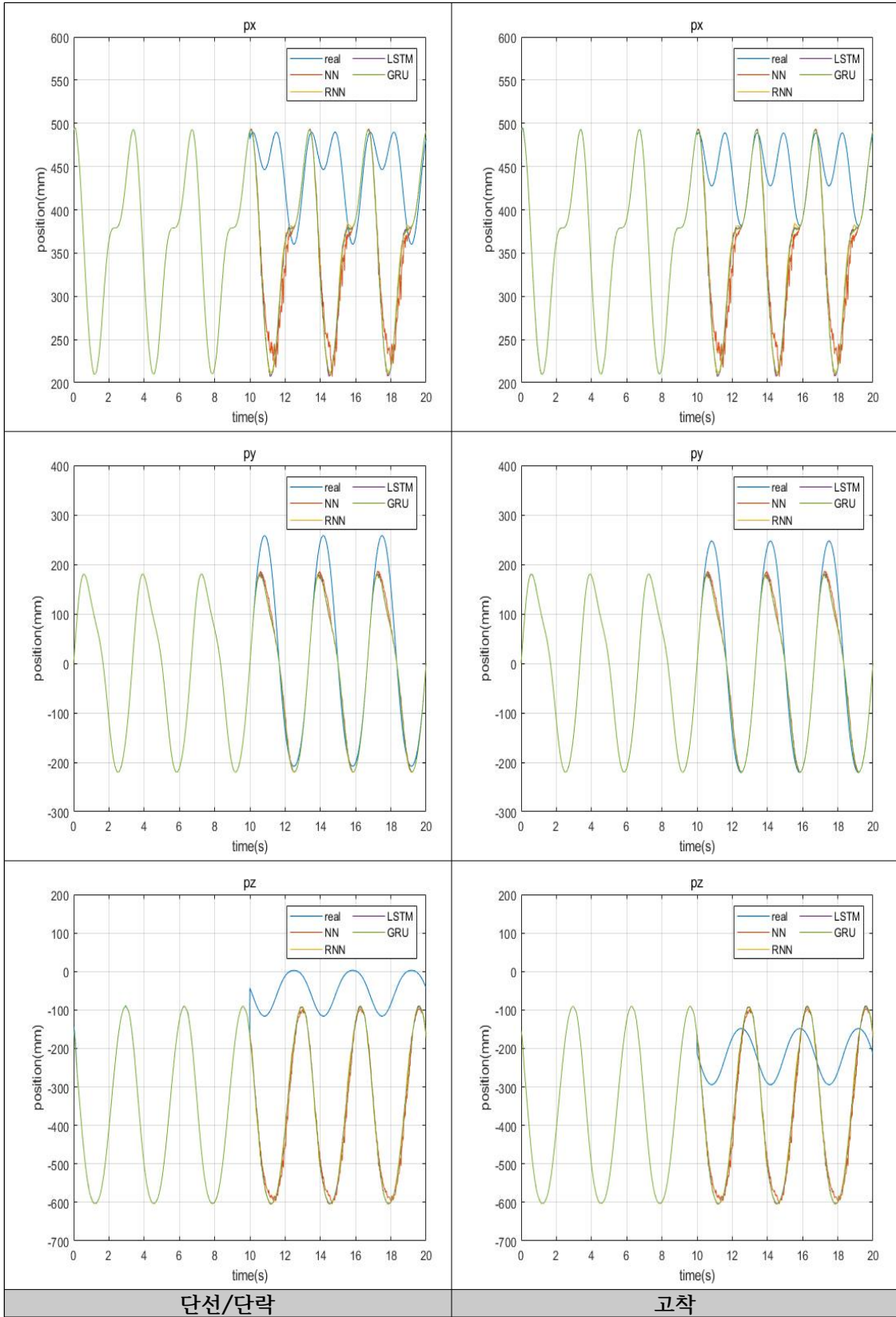
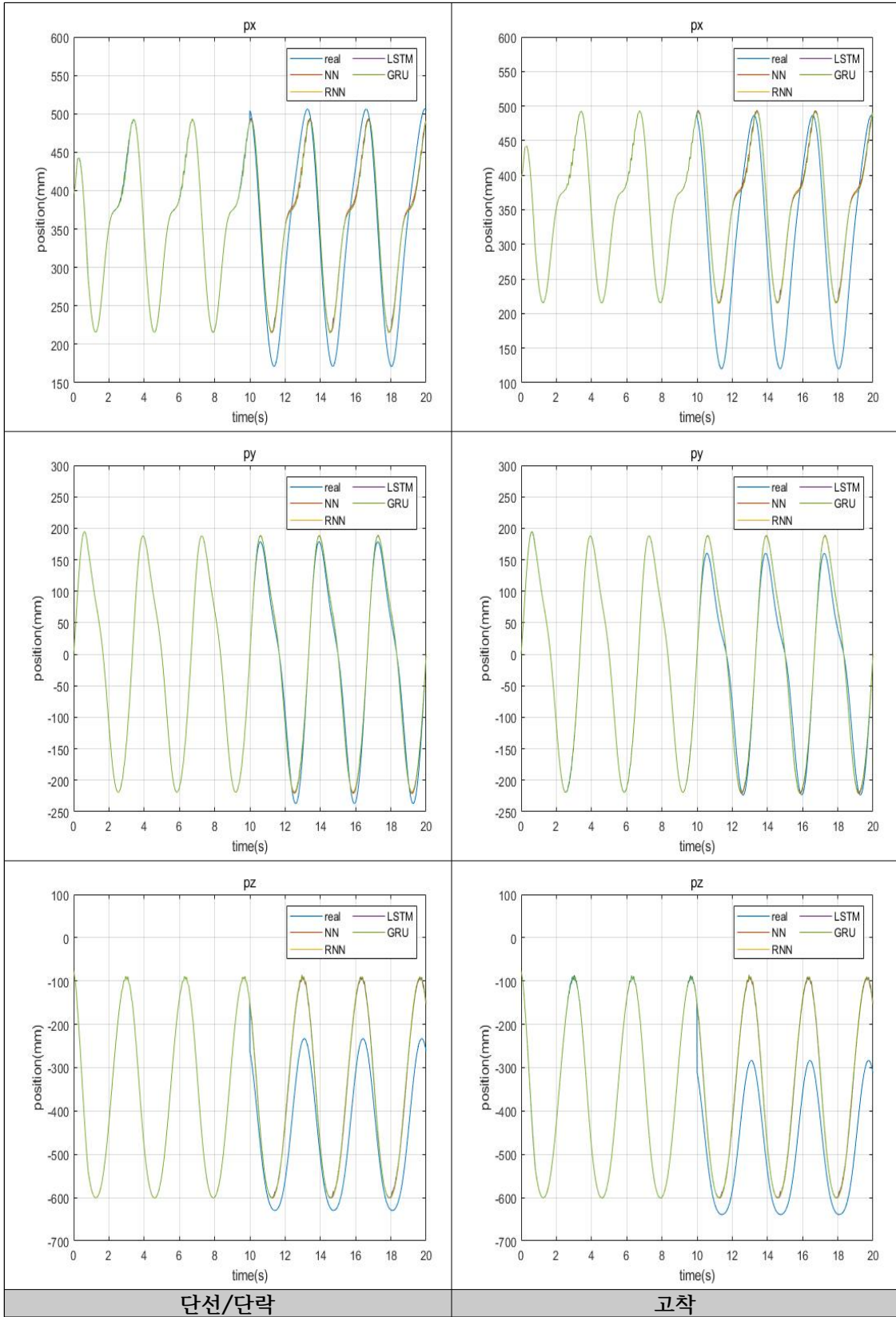


Table 4-16 실차 실험 결과 (SMC, Joint 3)



실차 실험 결과, 고장 진단 및 복구 알고리즘을 적용하지 않은 경우, 기준 경로 추적을 수행하지 못하고, 최대 300mm의 오차가 나오는 것을 확인하였다. 반면, 알고리즘을 적용한 경우에는 센서 고장 상황에서도 예측 신호를 출력하여 최대 오차 약 20mm정도로 기준 경로 추적을 잘 수행하고 있음을 확인하였다.

따라서 본 논문에서 제안한 고장 진단 및 복구 알고리즘은 실차 상황에서도 센서의 고장 신호를 복구할 수 있음을 검증하였다. 다만 오차가 많이 발생한 NN 알고리즘 학습의 경우는, 하이퍼 파라미터 튜닝을 최적화한다면 실차 상황에서도 오차를 많이 줄일 수 있을 것으로 판단된다.

5. 결론 및 향후 계획

본 논문에서는 3축 유압 매니퓰레이터의 고장 진단 및 복구 알고리즘을 설계하였다.

CATIA를 활용하여 유압 매니퓰레이터의 부품에 대해 모델링하였다. Forward Kinematics, Inverse Kinematics 해석을 통해 매니퓰레이터 기구학 모델링 분석을 진행하였으며 SolidWorks를 활용하여 각 부품을 조립하였다. 유압 회로 분석을 통해 유압부 모델링을 진행하였고 완성된 기구부와 유압부를 결합하여 가상 시뮬레이션 모델을 구축하였다.

실제 매니퓰레이터 시스템과 시뮬레이션 모델의 정합성을 확보하기 위해 각 Case 별 실험과 시뮬레이션을 진행하였다. CATIA, Simulink를 활용하여 내부 파라미터를 조정하였고, 최종적으로 실제 시스템과 같은 입력 인가 시 동일한 출력이 나오는 시뮬레이션 모델을 완성하였다.

고장 진단 및 복구 알고리즘을 적용하기 전 PID 제어기, 슬라이딩 모드 제어기를 적용하여 미리 생성한 Path Plan을 잘 추적하는지 검증하였다.

일시적인 고장 발생 시 Kalman Filter를 설계하여 노이즈를 저감시키고 센서의 고장 신호를 복구하였다.

영구적인 고장 발생 시 고장 진단 및 복구 알고리즘을 설계하기 위해 Python Keras 환경에서 학습 모델을 생성하였다. 학습 모델에서 훈련된 데이터를 기반으로 테스트 데이터에 대한 예측 값을 생성하였다. 예측 값과 센서 값을 비교하여 고장 발생 시 예측 값을 출력할 수 있는 고장 진단 및 복구 시스템을 구축하였다.

단선/단락, 고착 상황 발생 시 검출기에서 센서 값과 예측 값을 비교하여 기준 오차율을 초과하였을 때 예측 값을 출력하고 그 결과를 비교하였다. 비교 결과 예측 값은 센서의 신호를 정확히 복구하여 고장 발생 상황에서도 Position Tracking이 유지됨을 확인하였다.

본 연구를 통해 개발된 고장 진단 및 복구 알고리즘을 적용하면 센서 고장 발생 시 신호를 복구 할 수 있으며, 특히 매니퓰레이터의 위치 제어 시 센서 값이 불안정하더라도 피드백 제어를 수행할 수 있을 것으로 판단된다.

향후, 본 논문에 나오지 않은 여러 종류의 고장 상황을 인가하여 본 연구에서는 수행하지 못했던 여러 Case를 검증하고 적절한 하이퍼 파라미터 최적화 작업을 통해 알고리즘의 정밀도를 높이는 것이 요구된다.

참 고 문 헌

- (1) A. Freddi, S. Longhi, A. Monteriu, D. Orteni, D. Proietti Pagnotta, "Fault Tolerant Control Scheme for Robotic Manipulators Affected by Torque Faults", IFAC PapersOnLine, Vol. 51, No. 24, pp.886-893, 2018.
- (2) Jae-Wan Yang, Young-Doo Lee, In-Soo Koo, "Timely Sensor Fault Detection Scheme based on Deep Learning", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 20, No. 1, pp.163-169, 2020.
- (3) Jaemann Park, Jongho Shin, and Hyoun Jin Kim, "Formation Flight Control of Unmanned Aerial Vehicles Using Model Predictive Control", Journal of Institute of Control, Robotics, and Systems, Vol. 14, No. 12, pp.1212-1217, 2008.
- (4) 최병훈, 이병현, 김종암, 김현진, "모델 기반 관측기를 이용한 페루프 박리 제어 시스템 설계", 한국항공우주공학회, pp.1394-1399, 2011.
- (5) 박재현, 김성완, 김현진, "2자유도 로봇팔의 모델 비의존적 추정기 및 제어기 설계", 제어 로봇시스템학회, pp.191-192, 2016.
- (6) Dohyun Jang, Taewan Kim, and H. Jin Kim, "Quadrotor Cooperative Flight and Intruding Aircraft Capturing Algorithm using Model Predictive Control", Journal of Institute of Control, Robotics, and Systems, Vol. 23, No. 12, pp.1049-1054, 2017.
- (7) Dohyun Jang, H. Jin Kim, "Position Estimation of UAV Using Model Predictive Control in a Network Environment with Time Delay", Korea Institute Of Communication Sciences, pp.489-490, 2018.
- (8) Jun-Yeong Jang, Kang-Woon Lee, and Won-Tae Kim, "S-FDS: a Smart Fire Detection System based on the Integration of Fuzzy Logic and Deep Learning", The Institute of Electronics and Information Engineers, Vol. 54, No. 4, pp.50-58, 2017.
- (9) 최근호, 김건우, "머신러닝 기법을 이용한 납축전지 열화 예측 모델 개발", 한국지능정보 시스템학회, Vol. 24, No. 2, pp.243-264, 2018.
- (10) Jae-Wan Yang, Young-Doo Lee, In-Soo Koo "Sensor Fault Detection Scheme based on Deep Learning and Support Vector Machine", Journal of The Institute of Internet, Broadcasting and Communication, Vol. 18, No. 2, pp.185-195, 2018.
- (11) SANA ULLAN JAN, YOUNG-DOO LEE, JUNGPIIL SHIN, AND INSOO KOO, "Sensor Fault Classification Based on Support Vector Machine and Statistical Time-Domain

Features”, IEEE Trans, Ind. Electron, Vol. 5, pp.8682-8690, 2017.

(12) Thomas W. Rauber, Francisco de Assis Boldt, and Flavio Miguel Varejao, “Heterogeneous Feature Models and Feature Selection Applied to Bearing Fault Diagnosis”, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol. 62, No. 1, pp.637-646, 2015.

(13) Seung-Jae Kim, Jung-Jae Lee, “A Study on Face Recognition using Support Vector Machine”, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 16, No. 6, pp.183-190, 2016.

(14) Seok-Woo Jang, Gyungju Lee, Myunghee Jang, “Effective Detection of Target Region Using a Machine Learning Algorithm”, Korea Academy Industrial Cooperation Society, Vol. 19, No. 5, pp.697-704, 2018.

(15) Beomsang Kim, Ik Hyun Lee, “Retinal Blood Vessel Segmentation using Deep Learning”, Journal of KIIT, Vol. 17, No. 5, pp.77-82, 2019.

(16) Ergin Kilic, Melik Dolen, Hakan Caliskan, Ahmet Bugra Koku, Tuna Balkan, “Pressure prediction on a variable-speed pump controlled hydraulic system using structured recurrent neural networks”, Control Engineering Practice, Vol. 26, pp.51-71, 2014.

(17) Samira Zare, Moosa Ayati, “Simultaneous fault diagnosis of wind turbine using multichannel convolutional neural networks”, ISA Transactions, 2020.

(18) Shuhui Wang, Jiawei Xiang, Yongteng Zhong, Hesheng Tang, “A data indicator-based deep belief networks to detect multiple faults in axial piston pumps”, Mechanical Systems and Signal Processing, Vol. 112, pp.154-170, 2018.

(19) Guangzheng Hu, Huifang Li, Yuanqing Xia, Lixuan Luo, “A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis”, Computers in Industry, Vol. 100, pp.287-296, 2018.

(20) 나선준, “IoT를 활용한 소형 굴착기용 원격 시스템 구축에 관한 연구”, 울산대학교 석사 논문, 2020.

(21) 김봉재, “사판식 피스톤 펌프를 이용한 차량용 자동운할 시스템 개발”, 부산대학교 석사 논문, 2013.

- (22) D. H. Jang, S. K. Lee, Kwon and S. H. Park, "A Study on Pressure, Flow Fluctuation and Noise in the Cylinder of Swash Plate Type Axial Piston Pump", The Korea Fluid Power Systems Society, Vol. 6, No. 3, pp.38-46, 2009.
- (23) Xu Fang, Xiaoping Ouyang and Huayong Yang, "Investigation into the Effects of the Variable Displacement Mechanism on Swash Plate Oscillation in High-Speed Piston Pumps", applied sciences, Vol. 8, pp.658-673, 2018.
- (24) 김선민, "더블베인 로터리 유압액츄에이터 시스템의 임피던스 제어를 위한 토크서보디자인", 포항공과대학교 석사 논문, 2010.
- (25) 이일영, BRKR 교육사업부, "유압공학", 문운당, 2012.
- (26) Hydraulics & Pneumatics, "Fundamentals of Hydraulic Motors", Technologies, HYDRAULIC PUMPS & MOTORS, 2014.
- (27) 조한정, "마찰부하 상태에서의 유압모터 운동제어", 홍익대학교 석사 논문, 2007.
- (28) 심형보, "Sliding Mode Control의 기초", 제어이론학교/튜토리얼 워크샵, 2011.
- (29) Piazza d'Armi snc, "A QUICK INTRODUCTION TO SLIDING MODE CONTROL AND ITS APPLICATIONS", Dipartimento di ingegneria elettrica ed elettronica.
- (30) 오인영, "중대형 상용차용 100Nm 이상급 전기-유압 조향시스템을 위한 LSTM 기반 센서 고장복구 알고리즘", 성균관대학교 석사 논문, 2020.

Abstract

Manipulators are widely used in domestic and overseas industries instead of people, and in recent years, technologies such as artificial intelligence, big data, and IoT, which are the core technologies of the 4th industrial revolution, have been applied to the overall manufacturing industry. In addition, researches on maximizing production efficiency by applying automation and unmanned technologies are also actively progressing.

In the event of fault of various sensors attached to the manipulator, the malfunction of the manipulator may result in economic loss to the process operation as well as personal injury. In order to solve this problem, experts check and manage abnormalities at regular intervals, but there are cases where fault check timing is missed or not found, and it is difficult to check all of them due to the large number of sensors for data collection. Therefore, it is necessary to identify the presence or absence of a sensor fault in the embedded system in real time and to diagnose the fault and determine the type for quick response.

Among the approaches to fault diagnosis, the model-based approach has high accuracy and enables fault diagnosis with a small amount of data. In addition, by changing the parameters of the model, there is an advantage that it can be applied in various driving environments. However, in the model-based method, as the complexity of the system increases significantly, it is difficult to implement the model completely, and it is difficult to estimate the parameters of the model.

The second method, the data-driven approach, has an advantage that can be applied to a complex system, but has a disadvantage in that it requires a lot of data to train a predictive model.

In this paper, to compensate for the unpredictable factors of the model-based approach, a fault diagnosis and recovery algorithm based on the data-driven approach was proposed, and the following results were derived by applying the Deep Learning Algorithm.

1. The kinematic characteristics of the manipulator were analyzed by applying Forward Kinematics and Inverse Kinematics for the 3-axis hydraulic manipulator, and the hydraulic circuit was analyzed.
2. A virtual simulation model of the manipulator was built by connecting the kinematic model and the hydraulic model.

3. The fault signal was restored by applying Kalman Filter for temporary fault.
4. For permanent fault, an algorithm using NN, RNN, LSTM, and GRU was designed and applied, and the fault signal was recovered.
5. Based on the verified result in the virtual simulation model, the fault signal was restored by applying it to a real vehicle, and it was verified that position tracking was performed well when a fault condition occurred.