



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](#)

공학박사 학위논문

스마트 필드로봇의 시뮬레이션 모델
구축 및 심층강화학습을 이용한
작업경로 학습에 관한 연구

A Study on Building a Simulation Model of a
Smart Field Robot and Learning the Work Path
using Deep Reinforcement Learning

울산대학교 대학원
건설기계공학과
최성응

스마트 필드로봇의 시뮬레이션 모델
구축 및 심층강화학습을 이용한
작업경로 학습에 관한 연구


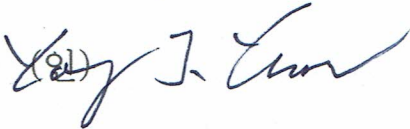
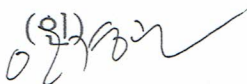


지 도 교 수 안 경 관
공 동 지 도 교 수 양 순 용

이 논문을 공학박사학위 논문으로 제출함

2022년 02월

울산대학교 대학원
건설기계공학과
최 성 응

최성웅의 공학박사학위 논문을 인준함

심사위원	이 병 룡	(인) 
심사위원	염 영 진	(인) 
심사위원	안 경 관	(인) 
심사위원	임 태 형	(인) 
심사위원	윤 영 환	(인) 

울산대학교 대학원

2022년 02월

감사의 글

7년의 시간 동안 부족한 저를 지도해주시고 많은 경험의 기회를 주신 양순용 교수님과 안경관 교수님께 고개 숙여 깊은 감사를 드립니다. 그리고 부족한 논문을 심사해주시고 지도해주신 이병룡 교수님, 염영진 교수님, 임태형 박사님, 운영환 박사님께 감사의 말씀을 드립니다. 아울러 대학원 과정 동안 많은 도움을 주신 기계공학부 교수님들 모두에게 감사의 말씀을 드립니다.

대학원 과정 동안 연구실에서 연구를 수행함에 있어 많은 조언과 도움을 주신 현대건설기계의 김영범 책임연구원님과 이상욱 연구원님, 생산기술연구원의 박상덕 박사님과 조정산 박사님, J&F Solution의 김동명 박사님, 모트롤의 이창돈 박사님과 박형규 박사님, 이튼모터스의 임구 사장님, CAE-CUBE의 이현규 부장님과 전재주 차장님, 디아이씨의 이성민 부장님께 감사드립니다.

오랜 시간동안 함께한 차량성능실험실의 구성원들에게도 감사드립니다. 부족했던 저를 지도해 준 김용석 박사님, 인호 형, 영만이 형, 7년 동안 동고동락을 같이 한 태운이, 연구실에 자주 찾아와서 조언해준 태형이 형, 세영이 형, 성희 형, 찬세 형께 감사를 드립니다. 그리고 실험실에서 말썽도 많았고 같이 재밌게 지낸 후배들 영재, 선준, 성원, 태곤, Le Quang Hoan, Nguyen Chi Tahn, Li Rui, Wang Shiliang, Zhang gaoqi 모두 감사드립니다. 그리고 건설기계공학과로 박사과정을 들어와 다사다난을 함께 나눈 건설기계공학과 학생들 유진, 효승, 보문, 은진, 양화, 진수, 용수, 범기, 서준, 성탁, 경신 모두 감사드립니다.

마지막으로 회사를 그만두고 대학원을 들어오면서 7년이란 세월동안 속도 많이 켜졌음에도 언제나 옆에서 아낌없는 지원과 응원해 주신 존경하는 부모님과 나를 지지해주고 스스로없는 친구처럼 대해준 여동생에게 진심으로 감사를 드립니다.

부족한 저를 졸업할 수 있게 많은 도움을 주신 모든 분들께 다시 한 번 고개 숙여 감사드립니다.

스마트 필드로봇의 시뮬레이션 모델 구축 및 심층강화학습을 이용한 작업경로 학습에 관한 연구

최성웅

울산대학교 대학원 건설기계공학과

국문요약

필드로봇은 건설업뿐만 아니라 농업, 임업, 제조업 등 다양한 산업에서 활용되고 있으며 해저영역으로까지 범위를 확장해가고 있다. 여기서 말하는 필드로봇은 공장이 아닌 필드에서 작업하는 로봇으로 그 중 건설작업에서 사용되는 로봇으로 건설기계를 자동화한 로봇을 의미한다. 대표적인 필드로봇은 굴착기, 휠로더, 지게차 등이 있으며 본 논문에서는 굴착기를 필드로봇으로 표현하였다.

필드로봇의 대표 작업으로는 굴착 작업, 평탄화 작업, 철거 작업 등이 있으며 본 논문에서 대상으로 하는 필드로봇은 소형 굴착기로 작업공간이 협소한 곳에서 작업을 많이 하게 된다. 그러나 협소한 작업 공간에서 작업을 하는데 있어 동작 범위의 한계로 인해 불필요한 동작으로 작업 공수가 늘어나고 작업 시간이 증가하는 등 작업 효율성이 떨어진다. 따라서, 작업 효율성 증가와 작업 편의성 증가를 위해 필드로봇의 자유도 증가가 필요하다. 마찬가지로 초소 선회를 대체할 버킷의 틸팅, 로테이션과 같은 새로운 메커니즘이 필요하다. 이렇게 작업 효율을 증가시키기 위한 새로운 메커니즘인 틸트로테이터를 적용함으로써 작업시간 감소 등으로 작업 효율성이 증가하게 된다.

최근 건설 분야에서는 스마트 건설을 실현하기 위한 핵심 기술들의 연

구가 활발히 진행되고 있다. 스마트 건설기계의 한 분야로 인공지능, AI 등과 건설기계를 접목한 지능형 건설기계 연구를 진행하고 있다. 이러한 연구들은 가상 공간의 필드로봇 모델을 이용하여 실제 필드로봇 작업을 수행하면서 발생하는 특성을 예측하는데 활용되고 있다.

따라서 본 논문에서는 필드로봇인 1.5톤 소형 굴차기를 대상으로 틸트로테이터가 적용된 6DOF 필드로봇에 대해 시뮬레이션 모델을 구축하고 동작 특성을 확인하여 분석하는 시뮬레이션과 필드로봇의 기구학적 모델과 강화학습 알고리즘을 이용한 심층강화학습 모델을 제안하고 실제 필드로봇에서 많이 수행하는 평탄화 작업에 대한 작업경로 시나리오를 모사한 경로 학습에 대한 기초연구를 수행하여 다음과 같은 결론을 도출하였다.

1. 다양한 필드로봇의 시뮬레이터 구축을 위한 시뮬레이션 모델 연구
 - 틸트로테이터가 적용된 6DOF 필드로봇 시스템 제시
 - 6DOF 필드로봇 시스템에 대한 기준 좌표계를 설정하고 순기구학, 역기구학의 수학적 모델을 제시하고 타당성을 검증
 - 6DOF 필드로봇 시스템에 대한 제원 및 설계사양을 확인하여 3D모델, 유압모델, 기구/동역학 모델인 Multibody 모델, 제어모델 구축
 - sine 파형, 굴착 동작에 대한 단독동작과 복합동작의 3가지 시뮬레이션을 통해 입력 값에 대해 출력 값이 유사하게 동작하는 것을 확인

2. 운전자 보조용 머신 가이드를 위한 심층강화학습 모델 기초 연구
 - 인공지능, AI 중 한 파트인 강화학습과 알고리즘에 대한 정의 및 이론 등 기본적인 개념 정립
 - 강화학습 엔진에 적용하기 위한 필드로봇의 URDF 모델, GYM 환경 모델, PPO 알고리즘으로 작업경로 학습 모델 구축
 - 평탄화 작업의 작업 경로를 학습한 결과 버켓 끝단의 위치가 설정한 오차 범위 0.1 m 이내에서 학습되어 동작하는 것을 확인
 - 학습된 결과를 시뮬레이션 모델에 적용하여 동작된 결과를 확인

목 차

국문요약	i
목 차	iii
표 목차	v
그림목차	vi
제 1 장 서론	1
1.1 연구배경 및 목적	1
1.2 연구현황	4
1.3 본문의 구성	6
제 2 장 필드로봇 시스템	7
2.1 기존 필드로봇 시스템	7
2.2 6DOF 필드로봇 시스템	8
2.3 6DOF 필드로봇 시스템 모델링 및 해석	9
2.3.1 순기구학(Forward kinematics)	9
2.3.2 역기구학(Inverse kinematics)	12
제 3 장 강화학습	22
3.1 강화학습 개요	22
3.1.1 강화학습의 정의	22
3.1.2 강화학습의 역사	23
3.1.3 강화학습의 구성요소	26
3.1.4 마르코프 결정 과정(Markov decision process)	28
3.1.5 가치 함수(Value function)	29
3.1.6 벨만 방정식(Bellman equation)	30
3.2 강화학습 알고리즘	31
3.2.1 강화학습 알고리즘 분류	31
3.2.2 동적 프로그래밍	33

3.2.3 시간차 학습	36
3.3 심층강화학습 알고리즘.....	41
3.3.1 정책 기울기(Policy gradient)	41
3.3.2 액터-크리틱(Actor-Critic)	44
3.3.3 근위 정책 최적화(Proximal policy optimization)	45
제 4 장 스마트 필드로봇 시스템 모델	47
4.1 스마트 필드로봇 시스템 모델 구성	47
4.2 스마트 필드로봇 기구 모델	48
4.3 스마트 필드로봇 유압 모델	50
4.4 스마트 필드로봇 제어 모델	68
제 5 장 시뮬레이션 및 결과	70
5.1 Matlab/Simulink Simscape 시뮬레이션	70
5.1.1 시뮬레이션 모델 및 동작 시뮬레이션 방법	70
5.1.2 시뮬레이션 결과	74
5.2 심층강화학습 시뮬레이션	85
5.2.1 제안 모델	85
5.2.2 학습 모델	86
5.2.3 학습 시뮬레이션	90
5.2.4 학습 시뮬레이션 결과	93
5.3 학습 결과 시뮬레이션	102
제 6 장 결론 및 향후 연구.....	104
참고문헌	108
Appendix	116
Abstract	124

표 목차

Table 1 6DOF Field robot Denavit-Hartenburg Table	10
Table 2 Limit angle for each joint of 6DOF field robot	18
Table 3 Range of PPO hyperparameters	46
Table 4 Opening area value of boom valve	55
Table 5 Opening area value of arm valve	56
Table 6 Opening area value of bucket and swing valve	57
Table 7 Cylinder specifications of 1.5 ton field robot	59
Table 8 Comparison of hydraulic models	62
Table 9 Simulation results of single operation	76
Table 10 Simulation results of compound operation	82
Table 11 Random target point at the end tip of the bucket	90
Table 12 Target point at the end tip of the bucket for straight path	92
Table 13 Result of random target point at the end tip of the bucket	93
Table 14 Error value of random target point at the end tip of the bucket	95
Table 15 Learning result of straight path at the end tip of the bucket	97
Table 16 Torque values of boom, arm and bucket for straight path	97
Table 17 Error value of straight path at the end tip of the bucket	100
Table 18 Result of angle values of boom, arm, bucket for straight path	102

그림 목차

Fig. 1 Type of field robot : excavator, forklift, wheel loader	1
Fig. 2 Excavation work using field robot	2
Fig. 3 Flattening work using a field robot	3
Fig. 4 Demolition work using field robot	4
Fig. 5 Field robot hydraulic oil circulation diagram	7
Fig. 6 Field robot with tiltrotator	8
Fig. 7 6DOF field robot reference coordinate system	9
Fig. 8 Simple reference coordinate system of 6DOF field robot	10
Fig. 9 Forward kinematics and inverse kinematics program (home position)	20
Fig. 10 Forward kinematics and inverse kinematics program (Max. angle)	21
Fig. 11 Reinforcement learning cycle flow chart	22
Fig. 12 Reinforcement learning algorithm map	31
Fig. 13 Bootstrapping update	36
Fig. 14 SARSA update	38
Fig. 15 Q-Learning update	40
Fig. 16 Configuration of smart field robot simulation model	47
Fig. 17 3D model of 1.5ton field robot using CATIA	48
Fig. 18 3D model of tiltrotator using CATIA	48
Fig. 19 Simscape multibody model of smart field robot	49
Fig. 20 Hydraulic circuit of 1.5 ton field robot	50
Fig. 21 Hydraulic circuit of Tiltrotator(SMP)	50
Fig. 22 Hydraulic pump model of 1.5 ton field robot	51
Fig. 23 Opening area diagram of boom valve	52
Fig. 24 Opening area diagram of arm valve	53
Fig. 25 Opening area diagram of bucket valve	54
Fig. 26 Opening area diagram of swing valve	54
Fig. 27 Boom valve	58
Fig. 28 Arm Valve	58
Fig. 29 Bucket valve	58
Fig. 30 Swing valve	58
Fig. 31 Tilt valve	58

Fig. 32 Rotator valve	58
Fig. 33 Boom Cylinder	60
Fig. 34 Arm Cylinder	60
Fig. 35 Bucket Cylinder	60
Fig. 36 Tilt Cylinder	60
Fig. 37 Swing Motor	61
Fig. 38 Rotation Motor	61
Fig. 39 Hydraulic model of field robot	63
Fig. 40 Simulation results of boom	64
Fig. 41 Simulation results of arm	65
Fig. 42 Simulation results of bucket	66
Fig. 43 Simulation results of tilt	67
Fig. 44 Block diagram of field robot	68
Fig. 45 PID controller model of field robot	69
Fig. 46 Simulation model of field robot using MATLAB/Simulink simscape	70
Fig. 47 Simulation GUI of field robot	71
Fig. 48 Setting angle of the excavation operation scenario of the boom	72
Fig. 49 Setting angle of the excavation operation scenario of the arm	72
Fig. 50 Setting angle of the excavation operation scenario of the bucket	72
Fig. 51 Setting angle of the excavation operation scenario of the Tilt	73
Fig. 52 Setting angle of the excavation operation scenario of the rotator	73
Fig. 53 Result of input and output angle using sine wave (boom)	74
Fig. 54 Result of input and output angle using sine wave (arm)	74
Fig. 55 Result of input and output angle using sine wave (bucket)	75
Fig. 56 Result of input and output angle using sine wave (tilt)	75
Fig. 57 Result of input and output angle using sine wave (rotator)	75
Fig. 58 Simulation results of single operation (boom)	77
Fig. 59 Simulation results of single operation (arm)	78
Fig. 60 Simulation results of single operation (bucket)	79
Fig. 61 Simulation results of single operation (tilt)	80
Fig. 62 Simulation results of single operation (rotator)	81
Fig. 63 Simulation results of compound operation (boom)	83
Fig. 64 Simulation results of compound operation (arm)	83

Fig. 65 Simulation results of compound operation (bucket)	83
Fig. 66 Proposed deep reinforcement learning model	85
Fig. 67 Field robot URDF model using Solidworks URDF expoter	86
Fig. 68 URDF model properties of field robots	86
Fig. 69 URDF model of field robot for reinforcement learning	87
Fig. 70 Open AI GYM environment model structure	88
Fig. 71 Simulation for random target points	91
Fig. 72 Working range at the end tip of the bucket for a straight path	92
Fig. 73 x-axis position of the end tip of the bucket for random point	94
Fig. 74 y-axis position of the end tip of the bucket for random point	94
Fig. 75 z-axis position of the end tip of the bucket for random point	94
Fig. 76 x-axis error value of the end tip of the bucket for random point	96
Fig. 77 y-axis error value of the end tip of the bucket for random point	96
Fig. 78 z-axis error value of the end tip of the bucket for random point	96
Fig. 79 x-axis position of the end tip of the bucket for straight path	98
Fig. 80 y-axis position of the end tip of the bucket for straight path	98
Fig. 81 z-axis position of the end tip of the bucket for straight path	98
Fig. 82 Torque values of boom for straight path	99
Fig. 83 Torque values of arm for straight path	99
Fig. 84 Torque values of bucket for straight path	99
Fig. 85 Results of boom joint torque value changes during path learning	100
Fig. 86 Results of arm joint torque value changes during path learning	100
Fig. 87 Results of bucket joint torque value changes during path learning	100
Fig. 88 x-axis error value of the end tip of the bucket for straight path	102
Fig. 89 y-axis error value of the end tip of the bucket for straight path	102
Fig. 90 z-axis error value of the end tip of the bucket for straight path	102
Fig. 91 Result of angle values of boom for straight path	104
Fig. 92 Result of angle values of arm for straight path	104
Fig. 93 Result of angle values of bucket for straight path	104

1. 서론

1.1 연구 배경 및 목적

필드로봇은 건설업뿐만 아니라 농업, 임업, 제조업 등 다양한 산업에서 활용되고 있으며 최근에는 해저영역으로 범위가 확장되고 있다[1][2][3]. 여기서 말하는 필드로봇은 공장이 아닌 필드에서 작업하는 로봇으로 본 논문에서는 건설작업에 사용되는 로봇으로 건설기계를 자동화한 로봇을 의미한다. 여러 산업에서 사용되는 만큼 건설기계의 메카트로닉스화, 로봇화가 필요하다. 대표적인 필드로봇은 Fig. 1과 같이 굴착기, 휠로더, 지게차 등이 있다. 본 논문에서는 1.5톤 소형 굴착기를 필드로봇이라 표현하였다[4].



Fig. 1 Type of field robot : excavator, forklift, wheel loader[5]

현장에서 필드로봇의 대표적인 작업은 굴착 작업, 평탄화 작업, 철거 작업 등이 있다. Fig. 2와 같이 굴착 작업은 논/밭 개량 작업, 관로 작업, 배수로 작업, 건물 철거 작업 등으로 나눌 수 있다. 논/밭 개량 작업의 경우 기존의 논과 밭을 용도 변경하는 작업으로 땅고르기 작업, 논둑 작업을 위한 굴착 작업과 용도에 맞는 평탄화 작업 등을 수행한다. 관로 작업의 경우 도로에 매립된 전기관로, 수도관로, 가스관로 등 매립된 관로 교체를 위한 굴착 작업을 수행한다. 배수로 작업 역시 용도에 맞게 사용하기 위해 토양을 걷어내는 굴착 작업을 수행한다. 또한 건물 철거 작업에서 발생하는 잔해들을 수거하기 위해 트럭에 옮겨 싣는 작업을 위한 굴착 작업을 수행한다. Fig. 3과 같이 평탄화 작업에는 건물 부지와 공원 부지 등 건설 현장에서의 기초 다지기 작업을 위해 평탄화 작업을 수행한다. 또한 상용차량이나 건설기계

등 장비들이 현장에 진입하기 위한 진입로를 만들기 위해 평탄화 작업을 수행한다. Fig. 4와 같이 철거 작업은 건물 신축 공사를 위한 구축 건물 철거, 실내 리모델링을 위한 실내 철거 작업, 건물의 내/외벽 철거를 위한 철거 작업을 수행한다.



Fig. 2 Excavation work using field robot



Fig. 3 Flattening work using a field robot



Fig. 4 Demolition work using field robot

본 논문에서 대상으로 하는 필드로봇인 1.5톤 소형 굴착기는 크기의 특성상 중/대형 굴착기가 진입하기 힘든 곳인 건물이 밀집한 좁은 골목, 도로 폭이 좁은 농로, 사람이 다니는 인도에서의 작업이 많다. 이러한 공간에서 아무리 작은 필드로봇이 작업 수행이 가능하더라도 작업 범위로 인해 자세의 한계점이 발생하여 불필요한 동작으로 작업 공수가 늘어나고 작업 시간이 증가하는 등 필드로봇의 작업 효율성

이 떨어진다. 이러한 한계점을 극복하기 위해 많은 필드로봇 작업자들이 버켓의 자유도를 증가시켜 주는 틸트로테이터(Tiltrotator)를 많이 장착해서 사용한다. 틸트로테이터는 필드로봇의 손목 역할을 해주므로 불필요한 동작이나 이동을 줄여줘 작업의 효율성을 올려주며 작업시간을 상당히 줄여준다. 틸트로테이터에 대한 수요가 증가함에 따라 여러 기업과 연구원 등에서 틸트로테이터를 이용하여 필드로봇의 자율/자동화를 위한 머신 컨트롤, 머신 가이드스에 다양한 연구들을 진행하고 있다. 최근 건설업에서는 스마트 건설(Smart construction)을 실현하기 위한 핵심기술 개발 연구를 진행하고 있다. 건설기계 분야에서는 지능형 건설기계로 AI, 인공지능, 강화학습 등의 기술을 건설기계와 접목하여 건설기계의 무인화, 자율화에 대한 연구를 추구하고 있다. 이러한 연구들을 필드로봇에 적용하기 전에 시뮬레이션 모델을 이용한 선행연구를 진행하여 가상공간의 필드로봇 모델에서 실제 필드로봇에 적용하였을 때 발생하는 특성들을 예측, 확인할 수 있다. 그래서 실제 필드로봇을 가상공간의 시뮬레이션 모델로 정확하게 구현할 수 있는 연구와 실제 필드로봇에서 수행하는 작업들을 시뮬레이션 모델에서 학습하여 실제 필드로봇 운전자들을 보조할 수 있는 연구들이 필요하다.

따라서 본 논문에서는 필드로봇인 1.5톤 소형 굴착기를 대상으로 틸트로테이터가 적용된 6DOF 필드로봇에 대해 연구하고 시뮬레이션 모델을 구축하여 필드로봇의 동작 특성을 확인하고 필드로봇의 기구학적 모델을 이용한 심층강화학습 모델을 제안하고 실제 필드로봇에서 많이 수행하는 평탄화 작업에 대한 작업경로를 시나리오로 모사하여 학습을 수행한다.

1.2 연구현황

현재 필드로봇의 운용자들은 기존의 필드로봇의 작업 한계점을 극복하기 위해 틸트로테이터를 장착하여 다양한 작업을 수행하고 있다. 건설기계와 관련된 기업들과 연구소, 대학에서는 이러한 트렌드에 맞춰 기존 연구에서 틸트로테이터가 장착된 필드로봇에 대한 연구가 여러 방면으로 이루어지고 있다. 기존 필드로봇의 작업성을 향상시키기 위해 많은 연구 개발이 있었지만 기존 필드로봇의 동작 범위 제한으로 작업의 자유도 측면에서 효율성에 대한 한계가 있었다. 따라서 기존 필드로봇 작업의 자유도를 높이기 위해서는 새로운 시스템에 대한 연구가 필요했다. 여기서 틸팅과 로테이팅이 동시 수행이 가능한 틸트로테이터의 핵심기술이 되었다. 현재 해외뿐만 아니라 국내에서도 틸트로테이터를 이용한 6자유도 필드로봇에 대한 많은 연구와 개발이 이루어지고 있다.

먼저 틸트로테이터에 대해 해외에서 많은 기술 개발이 이루어 졌으며 이미 전 세계적으로 상용화가 많이 되어있다. 대표적인 회사로는 Steelwrist(스웨덴)[6], Engcon(스웨덴)[7], Rototilt(스웨덴)[8] 등이 있으며 국내에서는 기존 해외 제품을 수입하여 필드로봇에 장착하여 사용했으나 최근에는 국내 개발되어 상용화 되고 있으며 대표적인 회사로는 제이케이[9], 틸트프로[10], 주현[11] 등 많은 기업에서 개발 상용화 하였다.

이렇게 틸트로테이터의 상용화로 인해 완성차 기업, 연구소, 대학 등에서는 틸트로테이터를 적용한 필드로봇에 대한 시뮬레이션 연구, 시뮬레이션 모델을 이용한 제어기법, 운전자 보조를 위해 학습을 통한 머신 가이드스, 머신 컨트롤 등 다방면으로 연구들이 이루어지고 있다.

틸트로테이터가 적용된 필드로봇의 시뮬레이션 모델 개발에 대한 연구로 Kim 등은 6축 굴착기 3D 시뮬레이션 모델에 대한 연구를 발표하였다[12][13]. H社의 30톤 굴착기를 RecurDyn을 환경 기반으로 하여 모델의 동역학 시스템을 구성하고 MATLAB/Simulink를 이용하여 유압 시스템을 구성하여 Co-Simulation을 위한 플랫폼 구축에 대해 연구하였다. Sami는 굴착기 시뮬레이션 모델을 구축에 대한 연구를 발표하였다[14]. K社의 굴착기를 이용하여 MATLAB/Simulink 기반으로 굴착기 시뮬레이션 모델을 구축하였으며 틸트로테이터를 활용한 굴착 및 덤프 시퀀스에 대한 궤적 제어와 굴착 작업에서 버킷의 엔드 이펙터의 위치와 힘 제어를 하는 하이브리드 위

치/힘 제어에 대해 연구하였다. Kim 등은 굴착기용 틸트로테이터에 대해 동역학 모델 개발과 작업모드를 고려한 틸트로테이터 시스템의 분석에 대한 연구를 발표하였다. 틸트로테이터의 작업에서 발생하는 특성과 여러 연구에 적용을 위한 다물체 동역학 모델을 개발하고 운용조건에 따른 동작 특성을 분석에 대해 연구하였다[15]. 그리고 굴착 작업 중에 발생하는 틸트로테이터의 거동 특성을 분석에 대해 연구하였다[16]. Kim 등은 6자유도 굴착기 고르기 작업을 위한 최적 위치 선정에 대한 연구를 발표하였다[17]. 이 연구에서는 6자유도 굴착기의 작업 시간과 에너지 소모를 최소화 하는 최적 위치 선정하기 위해 토크와 각속도를 이용하여 에너지 소모량을 해석하고 작업 공간을 이산화하여 최적 위치를 탐색하는 방법에 대해 연구하였다. Kurinov 등은 강화학습과 다물체 시스템 역학을 이용한 자동 굴착기에 대해 발표하였다[18]. 틸트로테이터가 적용된 6자유도 굴착기를 모델링하고 강화학습의 PPO(Proximal Policy Optimization) 알고리즘을 이용하여 토양을 굴착하여 호퍼에 적재하는 작업을 학습하여 학습된 동작을 보여준다.

1.3 본문의 구성

본 논문의 본문은 총 6개의 장으로 구성된다.

제 2장에서는 기존 필드로봇 시스템과 6DOF 필드로봇의 개요와 특징에 대해 알아보고 6DOF 필드로봇에 대한 기구학적 모델링과 해석하고 기구학적 모델을 이용하여 순기구학과 역기구학 해석을 검증한다.

제 3장에서는 강화학습의 개요와 강화학습, 심층강화학습의 알고리즘에 대해 알아보고 본 논문에서 제안하는 학습 모델링을 소개한다.

제 4장에서는 틸트로테이터가 적용된 스마트 필드로봇의 시스템 모델의 구성에 대해 소개하고 MATLAB/Simulink를 이용하여 기구 모델(Multibody model)과 유압 모델(Hydraulics model), 제어기 모델(Controller)에 대해 모델링하고 모델링한 유압 모델을 시뮬레이션 하여 필드로봇 제원과 비교하여 검증한다.

제 5장에서는 MATLAB/Simulink를 이용하여 필드로봇 모델의 굴착 동작에 대한 동작 시뮬레이션을 통해 스마트 필드로봇 시뮬레이션 모델의 동작 특성을 분석하고 심층강화학습 기법을 이용하여 스마트 필드로봇의 작업경로 학습을 위한 모델을 소개하고 시나리오에 대한 학습 시뮬레이션에 대한 결과를 분석한다.

마지막 6장에서는 본 논문에 대한 결론을 내리며 향후연구에 대해 설명하는 것으로 논문을 마무리한다.

2. 필드로봇 시스템

2.1 기존 필드로봇 시스템

기존 필드로봇은 유압장치를 이용하여 동작하고 버킷(Bucket)을 장착하여 굴착작업을 위해 제작된 건설기계의 한 종류이다. 굴착작업 외에도 다양한 어태치먼트를 장착해서 다양한 작업이 가능하여 건설현장 이외에 산업현장에서도 많이 사용되고 있다. 필드로봇의 유압장치는 유압유를 원활히 공급해주는 펌프(Pump), 유압유 방향을 결정하여 동작하고자 하는 곳으로 보내주는 메인 컨트롤밸브(Main control valve), 유압유를 공급받아 동작을 하는 액추에이터(Actuator), 유압유를 순환하여 보관하는 유압탱크(Hydraulic tank)로 Fig. 5와 같이 구성된다.

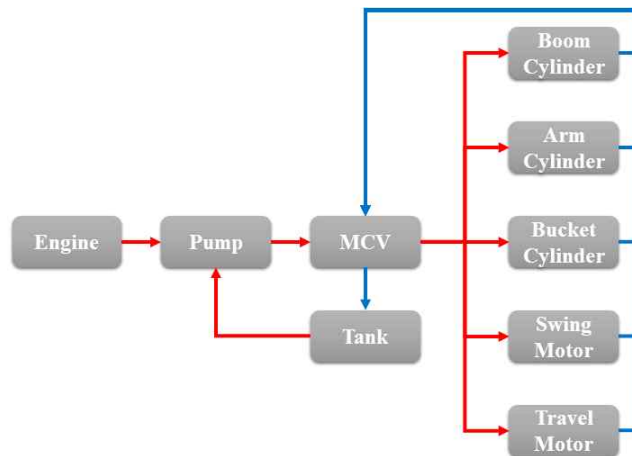


Fig. 5 Field robot hydraulic oil circulation diagram

Fig. 5와 같이 엔진을 구동하여 펌프를 동작시키면 펌프에서 유압유를 MCV(Main Control Valve)에 보내주고 MCV를 통해 각 실린더와 모터로 보내주는데 이는 RCV (Remote Control Valve)를 통해 선택된 동작에 따라 작동이 이루어진다. 작동은 실린더와 모터 2종류가 있는데 실린더는 붐, 암, 버킷을 작동하는데 사용되고, 모터는 필드로봇의 상부체 스윙과 필드로봇 주행을 위해 사용된다. 필드로봇은 주행 장치에 따라 크롤러 타입(Crawler type)과 휠 타입(Wheel type)으로 분류된다. 본 논문에서는 크롤러 타입의 필드로봇을 대상으로 하였다.

2.2 6DOF 필드로봇 시스템

기존의 필드로봇은 붐, 암, 버켓, 스윙으로 상부체의 4자유도를 가지는 시스템이다. 이러한 기존 필드로봇의 시스템은 협소한 공간에서의 작업을 수행 시 필드로봇의 제한적인 작업 반경으로 작업 시 불필요한 추가 동작을 하게 되어 작업시간이나 작업공정이 늘어나 작업 효율성이 떨어지게 된다. 기존 필드로봇의 시스템은 작업 환경에 따라 한계점을 가진다. 이를 보완하기 위해 Fig. 6과 같이 기존 필드로봇의 어태치먼트의 한 종류로 틸트로테이터를 장착하여 버켓을 좌우로 틸팅하거나 버켓을 360도 회전이 가능하도록 2자유도를 증가시켜 기존 필드로봇의 한계점을 극복할 수 있다.



Fig. 6 Field robot with tiltrotator[7]

틸트로테이터의 특징으로 필드로봇의 붐의 선형 이동 축을 따라 동작할 수 있도록 암과 버켓 사이에 장착되어 필드로봇의 버켓 자유도와 작동 범위가 증가한다. 필드로봇 붐 기준으로 버켓을 360도 회전이 가능하며 좌우로 40도 틸팅이 가능하다. 이로 인해 버켓의 자유도가 증가함에 따라 불필요한 동작을 제거함으로써 작업시간이 단축되고 필드로봇의 작업 효율성이 증가한다.

2.3 틸트로테이터 장착 6DOF 모델링 및 해석

2.3.1 순기구학(Forward kinematics)[19]

6DOF 필드로봇의 작업장치를 다관절 매니퓰레이터로 각 관절 각도가 주어졌을 때 직교 공간에서 End-effector의 좌표를 구하기 위해 6DOF 필드로봇의 기준 좌표계를 Denvit-Hartenburg법으로 좌표계를 설정하면 Fig. 7과 같다. 여기서 각 좌표계는 {0}번 좌표계는 원점 및 Swing 좌표계, {1}번 좌표계는 Boom 좌표계, {2}번 좌표계는 Arm 좌표계, {3}번 좌표계는 Bucket 좌표계, {4}번 좌표계는 Tilt 좌표계, {5}번 좌표계는 Rotator 좌표계, {6}번 좌표계는 End-effector 좌표계이다.

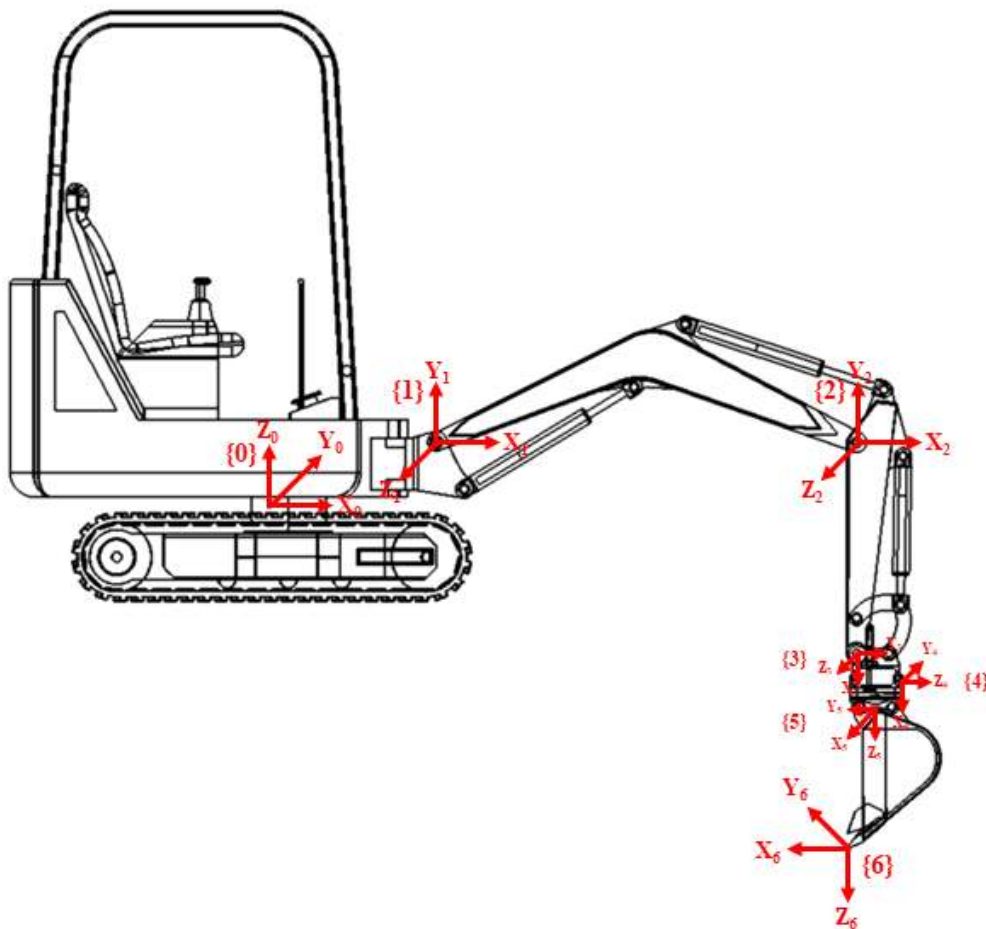


Fig. 7 6DOF field robot reference coordinate system

Fig. 7을 간략화한 기준 좌표계는 Fig. 8과 같으며 {4}번 좌표계는 실제 A 위치에 위치하며 {5}번 좌표계는 실제 B 위치에 위치한다. 기준 좌표계로부터 End-effector의 위치를 구하기 위해서는 각 관절의 좌표계 사이의 변환 행렬을 구해야 한다. 이에 Denavit-Hartenburg Table을 구하면 Table 1과 같다.

Table 1 6DOF field robot Denavit-Hartenburg Table

i	a_i	α_i	d_i	θ_i	status
1	l_2	$\frac{\pi}{2}$	l_1	$\theta_1 (0)$	0 \rightarrow 1
2	l_3	0	0	$\theta_2 (0)$	1 \rightarrow 2
3	l_4	0	0	$\theta_3 (-\frac{\pi}{2})$	2 \rightarrow 3
4	l_6	$-\frac{\pi}{2}$	0	$\theta_4 (0)$	3 \rightarrow 4
5	0	$-\frac{\pi}{2}$	l_5	$\theta_5 (-\frac{\pi}{2})$	4 \rightarrow 5
6	l_9	0	$l_7 + l_8$	$\theta_6 (\frac{\pi}{2})$	5 \rightarrow 6

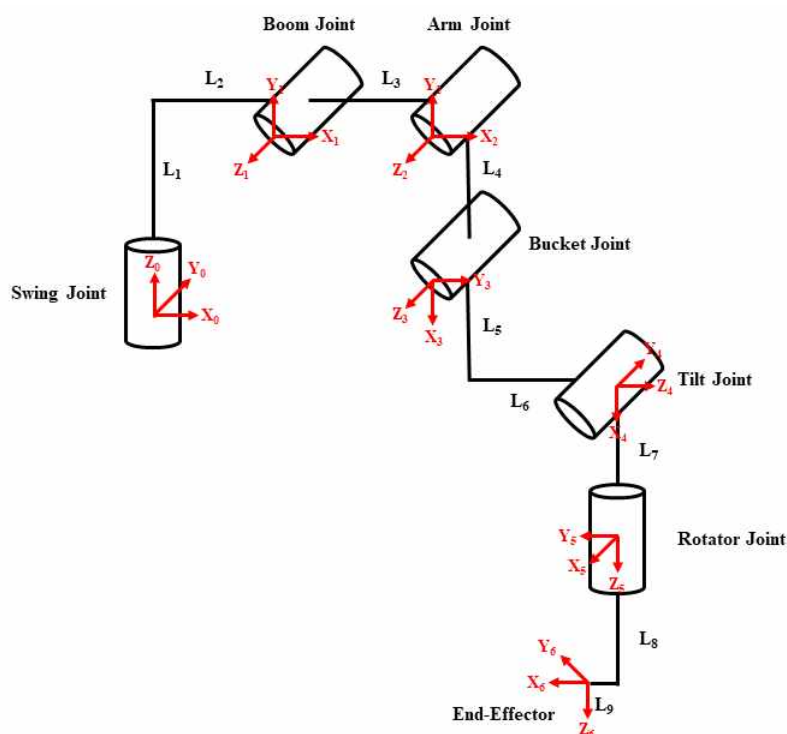


Fig. 8 Simple reference coordinate system of 6DOF field robot

각 관절의 변환 행렬을 표현하면 식 (1) ~ (2)과 같다.

$$T_{z,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, T_{z,d} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, T_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T = T_{z,d} \times T_{z,\theta} \times T_{x,\alpha} \times T_{x,a} \quad (2)$$

Forward kinematics rule에 따라 각 관절의 변환 행렬을 표현하면 식 (3)과 같다.

$$T_0^1 = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_1^2 = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_2^3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -l_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_3^4 = \begin{bmatrix} 1 & 0 & 0 & l_6 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_4^5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & l_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_5^6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & l_9 \\ 0 & 0 & 1 & l_7 + l_8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

최종적으로 Forward kinematics 전체 행렬은 식 (4)과 같다.

$$T = T_0^1 \times T_1^2 \times T_2^3 \times T_3^4 \times T_4^5 \times T_5^6 = \begin{bmatrix} -1 & 0 & 0 & l_2 + l_3 + l_5 - l_9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & l_1 - l_4 - l_6 - l_7 - l_8 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

2.3.2 역기구학(Inverse kinematics)[19]

6DOF 필드로봇의 Inverse kinematics는 End-effector의 위치가 주어졌을 때 각 관절의 각도를 구하는 것이다. Forward kinematics에서는 p_x, p_y, p_z 를 확인하였으며, Inverse kinematics에서는 Roll(Φ), Pitch(θ), Yaw(Ψ)를 확인한다. 따라서 Roll, Pitch, Yaw와 n, o, a 의 관계를 보면 식 (5) ~ (6)과 같다.

$$R_z(\Phi) = \begin{bmatrix} \cos\Phi & -\sin\Phi & 0 & 0 \\ \sin\Phi & \cos\Phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_x(\Psi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\Psi & -\sin\Psi & 0 \\ 0 & \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$R_t[\Phi, \theta, \Psi] = R_z(\Phi) \times R_y(\theta) \times R_x(\Psi) \quad (6)$$

본 논문에서는 변환 행렬을 이용하여 Inverse kinematics를 구한다. 각 관절의 변환 행렬을 모두 곱한 결과는 식 (7)과 같다.

$$T_0^6 = T_0^1 \times T_1^2 \times T_2^3 \times T_3^4 \times T_4^5 \times T_5^6 \quad (7)$$

식 (7)의 양변에 T_0^1 의 역행렬을 곱하면 식 (8)과 같다.

$$[T_0^1]^{-1} \times T_0^6 = T_1^2 \times T_2^3 \times T_3^4 \times T_4^5 \times T_5^6 \quad (8)$$

이 때, 식 (8)의 왼쪽 항을 A행렬, 오른쪽 항을 B행렬로 정의하고 A행렬과 B행렬의 성분을 비교한다. 비교한 성분은 식 (9) ~ (14)와 같다.

$$A(1,1)=n_x C_1 + n_y S_1, A(1,2)=o_x C_1 + o_y S_1 \quad (9)$$

$$A(1,3)=a_x C_1 + a_y S_1, A(1,4)=p_x C_1 - l_2 + p_y S_1$$

$$A(2,1)=n_z, A(2,2)=o_z, A(2,3)=a_z, A(2,4)=p_z - l_1 \quad (10)$$

$$A(3,1)=n_x S_1 - n_y C_1, A(3,2)=o_x S_1 - o_y C_1 \quad (11)$$

$$A(3,3)=a_x S_1 - a_y C_1, A(1,4)=p_x S_1 - p_y C_1$$

$$B(1,1)=S_{234} S_6 + C_{234} C_5 C_6, B(1,2)=S_{234} C_6 - C_{234} C_5 S_6, B(1,3)=-C_{234} C_5 \quad (12)$$

$$B(1,4)=l_4 C_{23} + l_3 C_2 - l_5 S_{234} + l_6 C_{23} C_4 - l_6 S_{23} S_4 + \frac{l_9 C_{234} C_{56}}{2} \\ - C_{234} S_5 (l_7 + l_8) + l_9 S_{234} S_6 + \frac{l_9 C_{56} C_{234}}{2}$$

$$B(2,1)=S_{234} C_5 C_6 - C_{234} S_6, B(2,2)=-C_{234} C_6 - S_{234} C_5 S_6, B(2,3)=-S_{234} S_5 \quad (13)$$

$$B(1,4)=l_4 S_{23} + l_3 S_2 + l_5 C_{234} + \frac{l_9 C_{56} C_{234}}{2} + l_6 C_{23} S_4 - l_6 S_{23} C_4 + \frac{l_9 S_{234} C_{56}}{2} \\ - S_{234} S_5 (l_7 + l_8) - l_9 C_{234} S_6$$

$$B(3,1)=-C_6 C_5, B(3,2)=S_5 S_6, B(3,3)=-C_5, B(1,4)=-C_5 (l_7 + l_8) - l_9 C_6 C_5 \quad (14)$$

A행렬과 B행렬의 성분을 비교하여 $\theta_1, \theta_5, \theta_6$ 을 구할 수 있다.

$$\textcircled{1} \theta_1 : A(3,4)=B(3,4)$$

$$p_x S_1 - p_y C_1 = (a_x S_1 - a_y C_1)(l_7 + l_8) + l_9 (n_x S_1 - n_y C_1) \\ (p_x - l_7 a_x - l_8 a_x - l_9 n_x) S_1 = (p_y - l_7 a_y - l_8 a_y - l_9 n_y) C_1 \\ \therefore \theta_1 = \text{Atan2}(p_y - l_7 a_y - l_8 a_y - l_9 n_y, p_x - l_7 a_x - l_8 a_x - l_9 n_x)$$

$$\textcircled{2} \theta_5 : A(3,3)=B(3,3)$$

$$a_x S_1 - a_y C_1 = -C_5 \\ \therefore \theta_5 = \text{acos}(a_y C_1 - a_x S_1)$$

$$\textcircled{3} \theta_6 : A(3,1)=B(3,1), A(3,2)=B(3,2)$$

$$\frac{B(3,2)}{-B(3,1)} = \frac{S_6 S_5}{C_6 S_5} = \tan(\theta_6) = \frac{o_x S_1 - o_y C_1}{n_y C_1 - n_x S_1}$$

$$\therefore \theta_6 = \text{Atan2}(o_x S_1 - o_y C_1, n_y C_1 - n_x S_1)$$

$\theta_2, \theta_3, \theta_4$ 를 구하기 위한 변환 행렬은 식 (15)와 같다.

$$[T_0^1]^{-1} \times T_0^6 \times [T_5^6]^{-1} \times [T_4^5]^{-1} = T_1^4 \quad (15)$$

이 때, 식 (15)의 왼쪽 항을 C행렬, 오른쪽 항을 D행렬로 정의하고 C행렬과 D행렬의 성분을 비교한다. 비교한 성분은 식 (16) ~ (21)과 같다.

$$C(1,1) = n_x C_1 C_5 C_6 - a_y S_1 S_5 - a_x C_1 S_5 + n_y C_5 C_6 S_1 - o_x C_1 C_5 S_6 - o_y C_5 S_1 S_6 \quad (16)$$

$$C(1,2) = a_x C_1 C_5 + a_y C_5 C_1 - n_x C_1 C_6 S_5 + n_y C_6 S_1 S_5 - o_x C_1 S_5 S_6 - o_y S_1 S_5 S_6$$

$$C(1,3) = -o_x C_1 C_6 - n_x C_1 S_6 - o_y C_6 S_1 - n_y S_1 S_6$$

$$C(1,4) = p_x C_1 - l_2 + p_y S_1 - a_x l_7 C_1 - a_x l_8 C_1 - l_9 n_x C_1 - a_y l_7 S_1 - a_y l_8 S_1 \\ - l_9 n_y S_1 + l_5 n_y S_1 S_6 + l_5 o_x C_1 C_6 + l_5 n_x C_1 S_6 + l_5 o_y C_6 S_1$$

$$C(2,1) = n_z C_5 C_6 - a_z S_5 - o_z C_5 S_6 \quad (17)$$

$$C(2,2) = a_z C_5 + n_z C_6 S_5 - o_z S_5 S_6$$

$$C(2,3) = -o_z C_6 - n_z S_6$$

$$C(2,4) = p_z - l_1 - a_z l_7 - a_z l_8 - l_9 n_z + l_5 o_z C_6 + l_5 n_z S_6$$

$$C(3,1) = a_y C_1 S_5 - a_x S_1 S_5 - n_y C_1 C_5 C_6 + n_x C_5 C_6 S_1 + o_y C_1 C_5 S_6 - o_x C_5 S_1 S_6 \quad (18)$$

$$C(3,2) = a_x C_5 S_1 - a_y C_1 C_5 - n_y C_1 C_6 S_5 + n_x C_6 S_1 S_5 + o_y C_1 S_5 S_6 - o_x S_1 S_5 S_6$$

$$C(3,3) = o_y C_1 C_6 + n_y C_1 S_6 - o_x C_6 S_1 - n_x S_1 S_6$$

$$C(3,4) = p_x S_1 - p_y C_1 - a_y l_7 C_1 + a_y l_8 C_1 + l_9 n_y C_1 - a_x l_7 S_1 - a_x l_8 S_1 \\ - l_9 n_x S_1 + l_5 n_x S_1 S_6 - l_5 o_y C_1 C_6 - l_5 n_y C_1 S_6 + l_5 o_x C_6 S_1$$

$$D(1,1) = C_{234} \quad (19)$$

$$D(1,2) = 0$$

$$D(1,3) = -S_{234}$$

$$D(1,4) = l_4 C_{23} + l_3 C_2 + l_6 C_{234}$$

$$D(2,1) = S_{234} \quad (20)$$

$$D(2,2) = 0$$

$$D(2,3) = C_{234}$$

$$D(2,4) = l_4 S_{23} + l_3 S_2 + l_6 S_{234}$$

$$D(3,1) = 0 \quad (21)$$

$$D(3,2) = -1$$

$$D(3,3) = 0$$

$$D(3,4) = 0$$

C행렬과 D행렬의 성분을 비교하여 $\theta_2, \theta_3, \theta_4$ 을 구할 수 있다.

$$\textcircled{1} \theta_2 + \theta_3 + \theta_4 : C(1,1) = D(1,1), C(2,1) = D(2,1)$$

$$\frac{D(2,1)}{D(1,1)} = \frac{S_{234}}{C_{234}} = \frac{n_z C_5 C_6 - a_z S_5 - o_z C_5 S_6}{n_x C_1 C_5 C_6 - a_y S_1 S_5 - a_x C_1 S_5 + n_y C_5 C_6 S_1 - o_x C_1 C_5 S_6 - o_y C_5 S_1 S_6}$$

$$\therefore \theta_2 + \theta_3 + \theta_4 = \text{Atan2} \left(\begin{array}{l} n_z C_5 C_6 - a_z S_5 - o_z C_5 S_6, \\ n_x C_1 C_5 C_6 - a_y S_1 S_5 - a_x C_1 S_5 + n_y C_5 C_6 S_1 - o_x C_1 C_5 S_6 - o_y C_5 S_1 S_6 \end{array} \right)$$

$$\textcircled{2} \quad \theta_2, \theta_3 : C(1,4)=D(1,4), C(2,4)=D(2,4)$$

$$p_x C_1 - l_2 + p_y S_1 = l_4 C_{23} + l_3 C_2 - l_5 S_{234} + l_6 C_{234} + \frac{l_9 C_{234} C_{56}}{2} - C_{234} S_5 (l_7 + l_8) + l_9 S_{234} S_6 + \frac{l_9 C_{5-6} C_{234}}{2}$$

$$p_x C_1 - l_2 + p_y S_1 + l_5 S_{234} - l_6 C_{234} - \frac{l_9 C_{234} C_{56}}{2} + C_{234} S_5 (l_7 + l_8) - l_9 S_{234} S_6 - \frac{l_9 C_{5-6} C_{234}}{2} = l_4 C_{23} + l_3 C_2$$

$$K_1 = l_4 C_{23} + l_3 C_2$$

$$p_z - l_1 = l_4 S_{23} + l_3 S_2 + l_5 C_{234} + \frac{l_9 C_{5-6} S_{234}}{2} + l_6 S_{234} + \frac{l_9 S_{234} C_{56}}{2} - S_{234} S_5 (l_7 + l_8) - l_9 C_{234} S_6$$

$$p_z - l_1 - l_5 C_{234} - \frac{l_9 C_{5-6} S_{234}}{2} - l_6 S_{234} - \frac{l_9 S_{234} C_{56}}{2} + S_{234} S_5 (l_7 + l_8) + l_9 C_{234} S_6 = l_4 S_{23} + l_3 S_2$$

$$K_2 = l_4 S_{23} + l_3 S_2$$

$$(l_4 S_{23} + l_3 S_2)^2 + (l_4 C_{23} + l_3 C_2)^2 = (K_2)^2 + (K_1)^2$$

$$(l_4)^2 + 2l_3 l_4 C_3 + (l_3)^2 = (K_2)^2 + (K_1)^2$$

$$\therefore \theta_3 = \arccos\left(\frac{(K_2)^2 + (K_1)^2 - (l_3)^2 - (l_4)^2}{2l_3 l_4}\right)$$

$$(l_4 C_3 + l_3) S_2 + l_4 S_3 C_2 = K_2$$

$$(l_4 C_3 + l_3) C_2 + l_4 S_3 S_2 = K_1$$

$$(l_4 C_3 + l_3) \times l_4 S_3 S_2 + (l_4)^2 (S_3)^2 C_2 = K_2 l_4 S_3$$

$$\rightarrow ((l_4 C_3 + l_3)^2 + (l_4)^2 (S_3)^2) C_2 - K_1 (l_4 C_3 + l_3) = K_2 l_4 S_3$$

$$\therefore C_2 = \frac{K_2 + \frac{K_1 (l_4 C_3 + l_3)}{l_4 S_3}}{(l_4 C_3 + l_3)^2 + \frac{(l_4 S_3)^2}{l_4 S_3}} = X_1$$

$$(l_4 C_3 + l_3) S_2 = K_2 - l_4 S_3 C_2$$

$$\therefore S_2 = \frac{K_2 - l_4 S_3 C_2}{(l_4 C_3 + l_3)}$$

$$\therefore \theta_2 = \text{Atan2}(Y_1, X_1)$$

$$\textcircled{3} \theta_4 : C(1,3)=D(1,3), C(2,3)=D(2,3)$$

$$C_{234} = \frac{a_x C_1 + a_y S_1}{-S_5} = X_2, S_{234} = -a_z = Y_2$$

$$C_{23} C_4 - S_{23} S_4 = X_2, S_{23} C_4 + C_{23} S_4 = Y_2$$

여기서 $C_{23} = x, S_{23} = y$ 로 치환하면 아래와 같다.

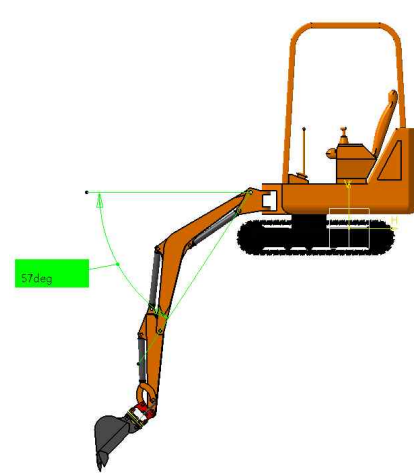
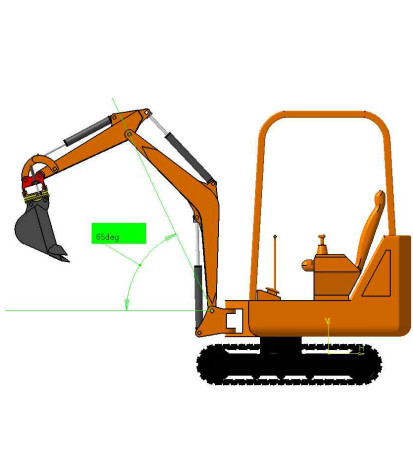
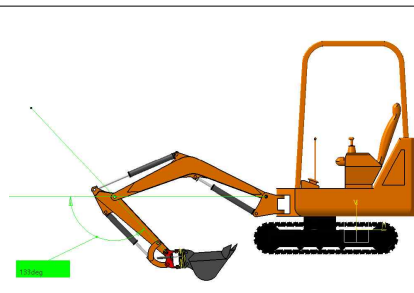
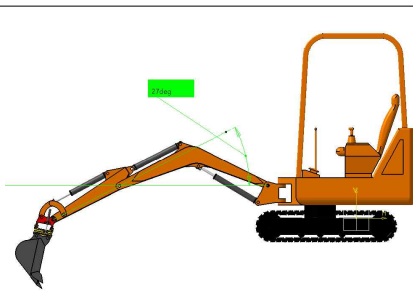
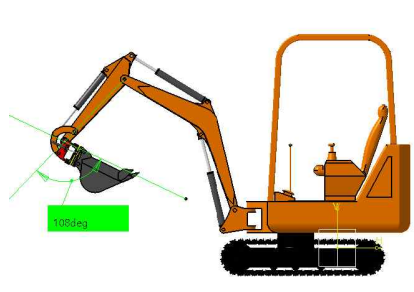
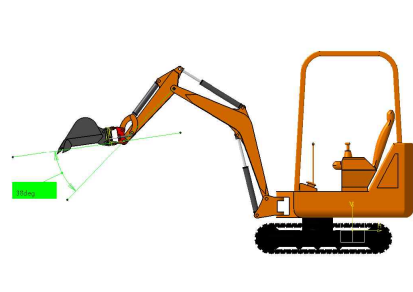
$$x C_4 - y S_4 = X_2, y C_4 + x S_4 = Y_2$$

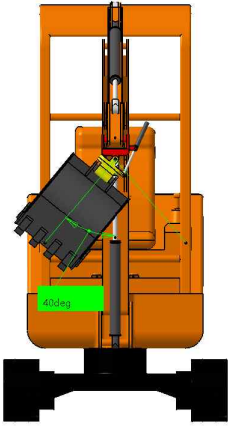
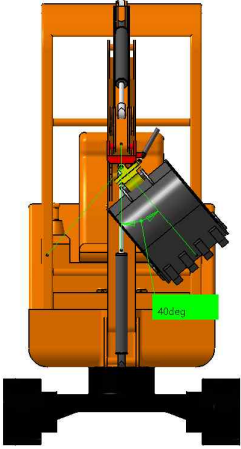
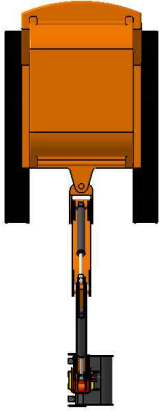

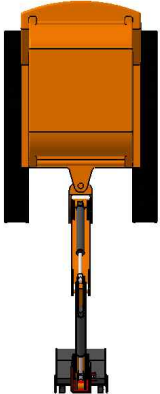

$$\therefore C_4 = \frac{x \text{ times } X_2 + y \text{ times } Y_2}{x^2 + y^2}, S_4 = \frac{x \text{ times } Y_2 - y \text{ times } X_2}{x^2 + y^2}$$

$$\therefore \theta_4 = \text{Atan2}(S_4, C_4)$$

Inverse kinematics의 많은 해를 줄이기 위해서 Table 2와 같이 6DOF 필드로봇의 각 관절의 각도를 제한한다.

Table 2 Limit angle for each joint of 6DOF field robot

	최소 각도 [deg]	최대 각도 [deg]
Swing (θ_1)	-180 (Left)	180 (Right)
Boom (θ_2)		
	-57 (Down)	65 (Up)
Arm (θ_3)		
	-133 (Down)	-27 (Up)
Bucket (θ_4)		
	-108 (Down)	38 (Up)

Tilt (θ_5)		
	-40 (Left)	40 (Right)
Rotator (θ_6)		
	-90 (Left)	90 (Right)
		
	180 (Left / Right)	

틸트로테이터 장착 6DOF 모델링의 Forward kinematics와 Inverse kinematics의 계산 결과를 확인하기 위해 MATLAB의 Guide를 이용하여 계산 결과를 Fig. 9 ~ 10과 같이 값과 그래프로 나타내었다. 먼저, Forward kinematics를 확인하기 위해 각 관절의 각도를 입력했을 때 홈 포지션일 때의 오리엔테이션과 끝점 위치와 붐, 암, 버킷이 최대각도일 때의 오리엔테이션과 끝점 위치를 확인하였다. 다음으로, Inverse kinematics를 확인하기 위해 Forward kinematics와 반대로 오리엔테이션과 끝점 위치를 입력했을 때 각 관절의 각도를 확인하였다. Forward kinematics를 통해 확인한 오리엔테이션과 끝점 위치를 Inverse kinematics 모델에 입력하여 계산된 각 관절의 각도 값을 비교한 결과 모델의 계산식이 타당한 것을 확인하였다.

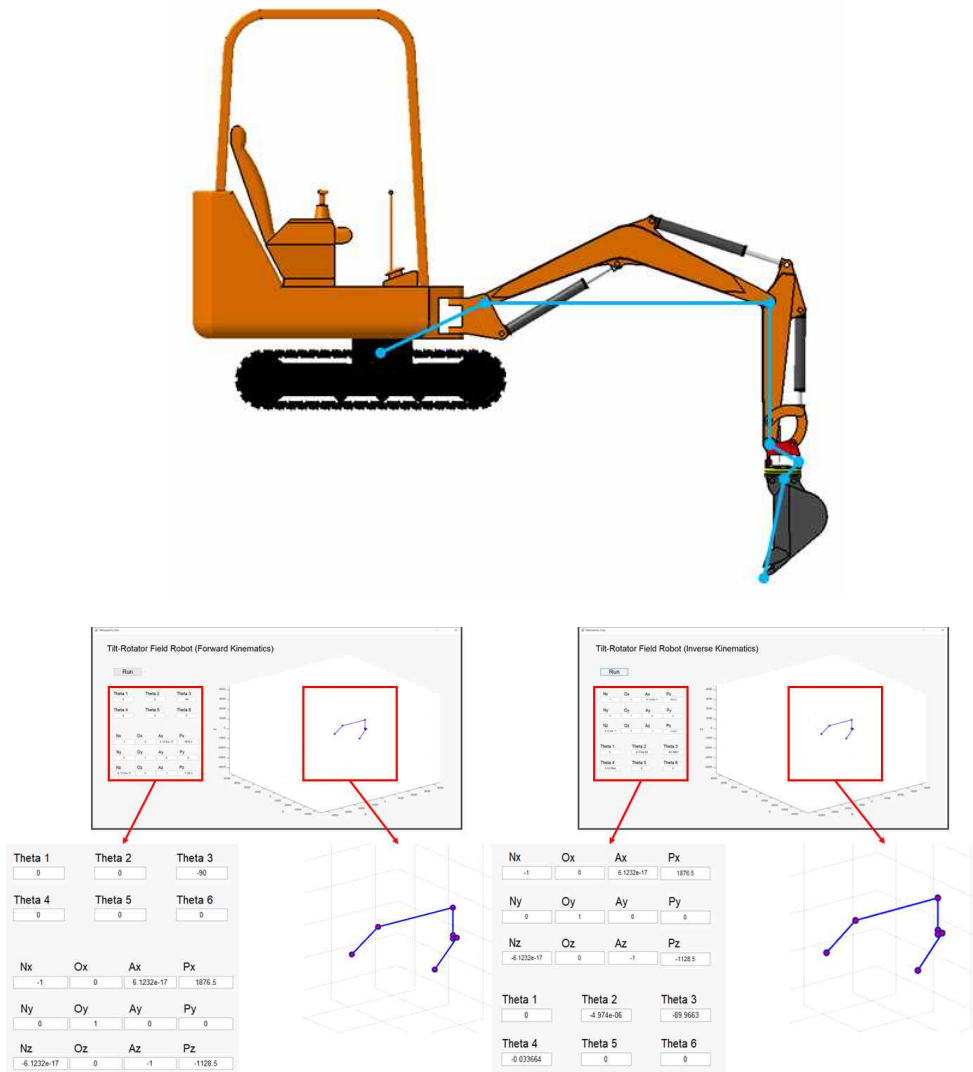


Fig. 9 Forward kinematics and inverse kinematics program (home position)

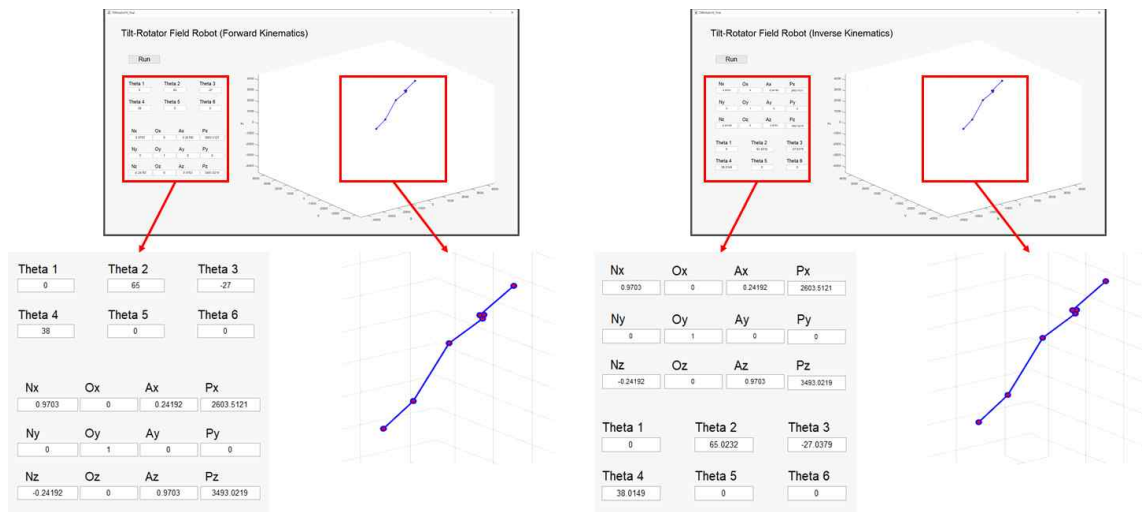
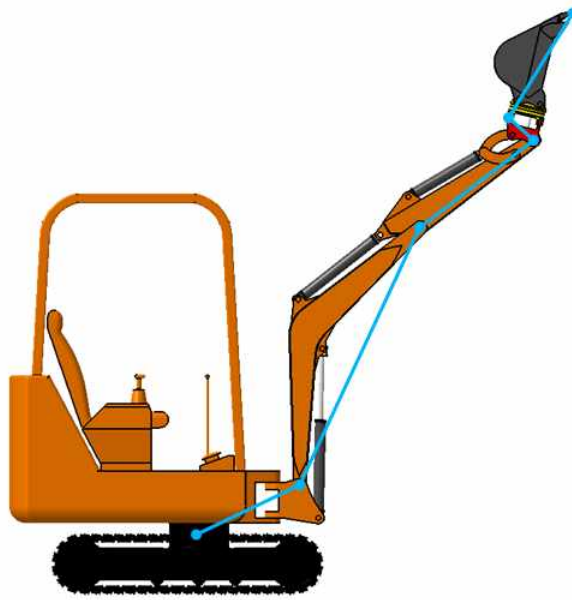


Fig. 10 Forward kinematics and inverse kinematics program (Max. angle)

3. 강화학습

심층강화학습(Deep reinforcement learning)을 수행하기 위해 강화학습(Reinforcement learning)의 개요와 이론에 대해 본 내용은 Sutton과 Barto의 ‘Reinforcement Learning 2nd edition’ [20]과 Lonza의 ‘Reinforcement Learning Algorithm with python’ [21]의 내용을 요약하여 설명한다.

3.1 강화학습 개요

3.1.1 강화학습의 정의

강화학습은 문제에 대해 순차적으로 의사결정을 생각하여 목표를 달성하는 기계 학습의 분야이다. 강화학습 문제는 의사결정자인 에이전트(Agent)에 끼치는 영향을 행동(Action, a_t)으로 가상으로 이루어진 환경과 상호작용을 하게 된다. 따라서 Fig. 11과 같이 가상으로 이루어진 환경은 에이전트가 수행한 행동에 대해 새로운 상태(State, s_{t+1})와 보상(Reward, r_t)을 피드백한다.

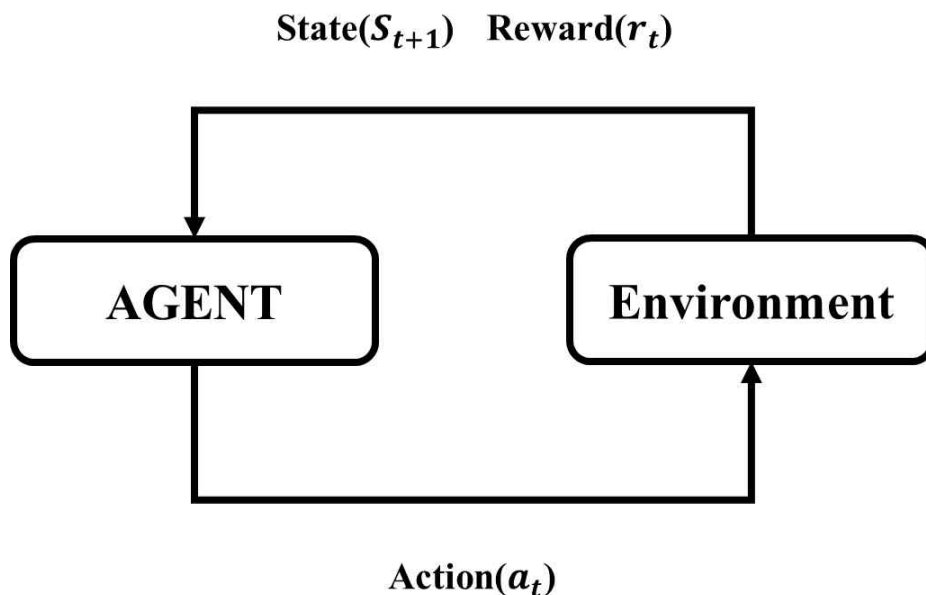


Fig. 11 Reinforcement learning cycle flow chart

여기서, 상태와 보상은 에이전트가 행동을 수행한 결과이다. 보상의 경우는 행동에 대해 좋은 결정인지 나쁜 결정인지에 대한 피드백이고 상태는 에이전트와 환경의 현재 상태를 나타낸다. 에이전트는 환경에 따라 어떤 행동을 해서 목표를 달성하기 위한 경로를 찾는 것과 같으며 강화학습 사이클은 목표하는 위치에 도달할 때까지 계속 학습과 실행을 반복 수행하게 된다. 그리고 에이전트는 목표에 도달하기 위해 경로를 찾으면서 피드백 받은 보상을 누적하면서 보상을 최대화한다. 에이전트는 누적된 보상의 총 합을 최대화하기 위해 행동 a_0, a_1, \dots, a_t 을 취한다.

강화학습의 중요한 특징은 동적이며 불확실하고 비결정론적인 환경을 다루는 특징이 있으며 현실에서 강화학습을 채택하는 중요한 항목이라고 할 수 있다.

3.1.2 강화학습의 역사

강화학습의 초기 역사는 현대 강화학습과 밀접하게 관련되기 전에 독립적으로 추구된 오래되고 풍부한 두 가지의 주요 방법이 있다. 첫 번째 방법은 시행착오를 통한 학습과 관련하여 동물 학습 심리학에서 시작되었다. 이는 인공지능의 초기 작업의 일부 실행하고 1980년대 초 강화학습이 다시 유행하기 시작했다. 두 번째 방법은 최적 제어와 동적 프로그래밍을 사용한 솔루션에 관한 것이다. 하지만 이 당시에 두 번째 방법은 학습에 포함되지 않았다. 이 두 가지 방법은 독립적이지만 시간차 방법과 관련하여 덜 구별되는 세 번째 방법을 중심으로 상호 연관이 되었다. 세 가지 방법은 모두 1980년대 후반 현대 강화학습 분야를 만들게 되었다.

시행착오 학습에 중점을 둔 방법은 짧은 역사에도 친숙하고 많은 이야기를 가진 방법이다. 하지만 이를 수행하기 전 최적 제어 방법에 대해 설명한다.

최적 제어는 1950년대 후반 시간 경과에 따른 동적 시스템의 동작 특징을 최소화 또는 최대화하는 제어기를 설계하는 문제를 설명하기 위해 사용되었다. 문제 접근 방식 중 1950년대 중반 Bellman이 Hamilton과 Jacobi의 19세기 이론을 확장하여 개발했으며 이 방식을 벨만 방정식(Bellman equation)이라 불린다. 방정식을 정의하기 위해 동적 시스템의 상태와 가치 함수와 함께 최적 반환 함수의 개념을 사용한다. 이 방정식을 풀어 최적 제어 문제를 해결하는 방법의 클래스는 동적 프로그래밍[22]으로 알려졌으며 Bellman도 마르코프 결정 프로세스(Markov decision process)로 알려진 최적 제어 문제의 이산 확률을 도입했다[23]. Howard는 마르코프 결정 프로세

스에 대한 정책 반복 방법을 제시했다[24]. 이는 현대 강화학습의 이론과 알고리즘의 기초가 된다.

시간차 학습의 기원은 부분적으로 동물 학습 심리학과 2차 강화물(Secondary reinforcers)의 개념이 있다. 2차 강화물은 1차 강화물과 짝을 이룬 자극을 결과적으로 유사한 강화 특성을 갖는다. Minsky는 이러한 심리학적 원리가 인공 학습 시스템에 중요하다는 것을 처음 제시했다[25]. Samuel은 체커 게임의 일부로 시간적 차이 아이디어를 포함한 학습방법을 제안하고 구현하였다[26]. Samuel은 Minsky의 연구나 동물 학습과 연결 가능성에 대해 언급하지 않았다. 그의 아이디어는 Shannon이 체스를 플레이 하기 위해 평가 기능을 사용하는 컴퓨터 프로그래밍 할 수 있고 온라인으로 수정하여 게임을 향상 시킬 수 있다[27]는 아이디어에서 시작되었다. Minsky는 Samuel의 논문에서 그의 연구를 광범위하게 논의하며 자연적이고 인공적인 이차 강화 이론과의 연결을 제안했다[28]. Sutton은 Klopf의 아이디어와 동물 학습 이론과의 연결을 발전시켜 시간적으로 연속적 예측 변화에 따른 학습 규칙을 소개했다[29][30][31]. Sutton과 Barto는 이런 아이디어를 시간차 학습에 기반한 고전 조건화의 심리학 모델을 개발했다[32][33].

위에서 언급한 선행 작업을 시간차 학습 및 시행착오 학습의 일부로 충분히 알고 있다. 이때 Sutton은 Actor-Critic 구조로 알려진 시행착오 학습과 시간차 학습을 사용하여 두 가지 학습을 결합한 방법을 개발했다[34]. Sutton은 시간차 학습을 통제로부터 분리하여 일반적인 예측 방법을 사용함으로써 중요한 단계를 수행했다. 이 연구는 TD(λ) 알고리즘을 소개하고 일부 수렴하는 속성을 증명했다[35].

시간차 학습과 최적 제어 방법은 Watkins의 Q-learning 개발로 인해 완전 통합되었다. 강화학습 연구의 세 방법에서 확장하고 통합했다[36]. Werbos는 1977년 이래 시행착오 학습과 동적 프로그래밍의 수렴을 소개했으며 통합에 기여하였다[37]. 왓킨스가 작업하던 당시 강화학습, 기계학습 분야에서 엄청난 성장을 이루었다. 인공지능, 인공신경망 등에서 더욱 광범위하게 사용되었다. 1992년 테사우로의 주사위 놀이 프로그램인 TD-Gammon의 성공이 많은 관심을 일으켰다. 2013년과 2015년 Mnih은 심층 신경망 훈련의 최근 발전을 사용하여 종단 강화 학습을 사용한 고차원 감각 입력에서 성공적인 정책을 학습하는 새로운 인공 에이전트인 심층 Q-네트워크(Deep Q-network, DQN)를 개발했다. 이 모델에 대한 검증은 Atari 2600 게임을 이용하여 이전 접근 방식들과 사람을 능가하는 실력을 보여주었다[38]. 2016년 인공

지능과 인간의 대결로 세간의 이목을 집중시킨 이세돌과 바둑 대결로 유명해진 알파고가 공개되었다. 알파고는 구글 딥마인드 팀에서 개발된 바둑프로그램으로 가치 네트워크와 정책 네트워크를 사용하여 컴퓨터 바둑에 대한 새로운 접근 방식을 소개하였다. Monte carlo 시뮬레이션과 가치 정책 네트워크를 결합한 알고리즘으로 유럽 바둑 챔피언은 5:0으로 이겼으며 2016년 바둑프로 이세돌과의 대결에서는 4:1로 이기며 알파고의 능력을 검증하였다[39]. 2015년 Lillicrap은 Deep Q-learning의 성공 기반이 되는 아이디어를 지속적인 행동 영역에 적용하여 연속 행동 공간에서 작동하는 결정론적 정책 기울기 기반 Actor-Critic, 모델 프리인 심층 결정론적 정책 기울기(Deep Deterministic Policy Gradient, DDPG) 알고리즘을 제시하였다[40]. 2015년 Schulman은 단조로운 개선을 보장하고 제어 정책을 최적화하며 이론적으로 정당화되는 몇 가지 근사치를 적용한 실용적인 알고리즘인 신뢰 지역 정책 최적화(Trust Region Policy Optimization, TRPO)를 개발한다. 이 알고리즘은 신경망과 같은 대규모 비선형 정책을 최적화하는데 효과적이다[41]. 2017년 Schulman은 환경과 상호작용을 통해 확률적 기울기 상승을 사용하고 목적 함수를 최적화하는 것을 번갈아가며 강화학습하는 새로운 정책 기울기 방법을 제안했다. 미니 배치 업데이트의 여러 에포크를 가능하게 하는 새로운 목적 함수로 근위 정책 최적화(Proximal Policy Optimization, PPO)라는 새로운 방법을 제시했다. TRPO보다 구현이 간단하고 일반적이며 샘플 경험성이라는 이점을 가진다[42].

3.1.3 강화학습의 구성요소

강화학습의 에이전트는 행동을 통해 환경과 상호작용을 한다. 에이전트는 행동의 좋음과 나쁨의 정도와 새로운 상태는 보상을 피드백 받는다. 에이전트는 학습을 하는 동안 상황에 따라 최적의 행동을 학습하며 최대의 누적 보상을 받는다. 정책(Policy)에 따라 특정 상태의 행동을 선택하여 얻을 수 있는 누적 보상을 가치함수(Value function)라고 한다.

정책(Policy)은 에이전트가 주어진 상태에서 행동을 선택하는 방법을 정의하며 즉각적인 보상이 아닌 해당 상태에서의 누적 보상을 최대화하는 행동을 선택한다. 정책에서는 확률적 정책(Stochastic policy)은 행동에 대한 확률 값을 제공하는 정책이며 결정론적 정책(Deterministic policy)은 하나의 결정론적 행동에만 제공하는 정책을 말하며 확률적 정책과 결정론적 정책 사이에 중요한 차이점이다. 강화학습 알고리즘을 분류하는 방법은 학습 중 정책이 개선되는 방식을 기반으로 한다. 간단한 알고리즘은 환경에 대해 행동하는 정책이 학습하면서 개선되는 정책과 유사한 경우이다. 즉, 정책에서 생성된 동일한 데이터를 이용하여 학습한다. 이런 알고리즘을 On-policy라고 한다. 반대로 Off-policy 알고리즘은 환경에 대해 행동하는 정책과 학습을 하지만 실제로 사용되지 않는 정책 두 가지 정책을 가진다. On-policy는 행동 정책(Behavior policy), Off-policy는 대상 정책(Target policy)이라 한다.

가치함수(Value function)는 상태의 장기적인 품질을 나타낸다. 에이전트가 주어진 상태에서 시작하면 미래에 기대되는 누적 보상이다. 보상이 즉각적인 성과를 측정하면 가치함수는 장기적으로 성과를 측정한다. 높은 보상이 높은 가치함수를 의미하지 않고 낮은 보상이 낮은 가치함수를 의미하지 않는다. 가치함수는 상태 함수 또는 상태-행동 쌍의 함수일 수 있다. 상태 함수를 상태-가치함수(State-Value function)라 하고 상태-행동 쌍의 함수를 행동-가치함수(Action-Value function)라고 한다. 행동-가치 방법 또는 가치-함수 방법은 강화학습 알고리즘의 또 다른 유형의 방법이다. 이 방법은 행동-가치함수를 학습하고 이를 사용하여 수행할 최적의 행동을 선택한다. 일부 정책 기울기 방법은 두 방법의 장점을 결합하여 가치 함수를 사용하고 적절한 정책을 학습할 수 있다. 이런 방법을 액터-크리틱 방법(Actor-Critic methods)이라고 한다.

보상(Reward)은 각 시간 단계에서 에이전트가 이동할 때마다 환경은 해당 행동이

에이전트에게 얼마나 좋은지 나타내는 값을 보낸다. 이것을 보상이라고 한다. 에이전트의 최종 목표는 환경과 상호 작용하는 동안 얻은 누적 보상을 최대화하는 것이다. 보상은 환경의 일부로 간주되지만 실제로는 그렇지 않다. 에이전트에서도 보상을 얻을 수 있지만 의사결정 단계에서는 보상을 절대 얻을 수 없다. 이런 이유로 보상은 항상 환경에서 전달된다. 보상은 강화학습 주기에 주입되는 슈퍼비전 신호이며 좋은 행동을 가진 에이전트를 얻기 위해 올바른 방식으로 보상을 설계하는 것이 필수적이다. 보상에 결함이 있는 경우 에이전트가 결함을 찾아 잘못된 행동을 따를 수 있다. 보상은 환경에 따라 다른 빈도로 나타난다. 빈도가 높은 보상을 조밀한 보상(Dense reward), 빈도가 낮은 보상을 희소 보상(Sparse reward)이라 한다. 빈도가 희소 보상의 경우 에이전트가 보상을 받고 최적의 행동을 찾는 것이 어려울 수 있다. 모방 학습(Imitation learning)과 역 강화학습(Inverse RL)은 환경에 보상이 없는 경우를 다루는 기술이다. 모방 학습은 상태를 행동에 매핑하기 위해 전문가 데모를 사용한다. 역 강화학습은 전문가의 최적 행동에서 보상 함수를 추론한다.

모델(Model)은 에이전트의 선택 구성 요소로 환경에 대한 정책을 찾는 데 필요하지 않다. 모델은 주어진 상태와 행동에 대한 다음 상태와 보상을 예측하며 환경이 어떻게 행동하는지 설명한다. 모델이 알려진 경우 계획 알고리즘을 사용하여 모델과 상호 작용하고 향후 행동을 권장할 수 있다. 환경 모델은 사전에 제공되거나 환경과의 상호 작용을 통해 학습할 수 있다. 환경이 복잡한 경우 심층 신경망(Deep neural networks)을 사용해서 근사화하는 것이 좋다. 이와 같이 이미 알려진 환경 모델을 사용하거나 학습하는 강화학습 알고리즘을 모델 기반 방법(Model-based methods)이라고 한다.

3.1.4 마르코프 결정 과정(Markov Decision Process, MDP)

Markov Decision Process는 행동이 다음 상태와 결과에 영향을 미치는 순차적 의사결정 문제를 나타낸다. MDP는 강화학습으로 해결되는 동일한 문제인 상호 작용을 통해 목표를 학습하는 문제를 공식화할 수 있을 만큼 충분히 일반적이고 유연하다. MDP 관점에서 강화학습 문제를 표현하고 추론할 수 있다.

MDP는 4-튜플(S, A, P, R)로 구성되어 있다. S 는 유한한 상태 집합이 있는 상태 공간, A 는 행동의 유한 집합이 있는 행동 공간, P 는 s 에서 행동 a 를 통해 상태 s' 에 도달할 확률을 정의하는 전이 함수이다. $P(s', s, a) = p(s'|s, a)$ 에서 전이 함수는 s 와 a 가 주어질 때 s' 의 조건부 확률과 같다. R 은 상태 s 에서 행동을 취한 후 상태 s' 로 전환하기 위해 받은 값을 결정하는 보상 함수이다.

MDP는 상태와 행동($S_0, A_0, S_1, A_1, \dots$)의 궤적을 생성하는 일련의 이산 시간 단계에 의해 제어되며 여기서 상태는 MDP의 역학, 상태 전이 함수를 따른다. 이런 방식으로 전환 함수는 환경의 역학을 완전히 특성화한다. 정의에 따르면 전이 함수와 보상 함수는 현재 상태에 의해서만 결정되며 이전에 방문한 상태의 순서가 결정되지 않는다. 이 속성을 Markov 속성이라 한다. 프로세스에 메모리가 없고 미래 상태는 현재 상태에만 의존하고 히스토리에 의존하지 않는다. 따라서 상태는 모든 정보를 보유하게 된다. 이런 속성을 가진 시스템을 완전히 관측이 가능하다.

실제로 많은 강화학습 사례에서 Markov 속성은 유지되지 않는다. 실용성을 위해 MDP라고 가정하고 유한한 수의 이전상태($S_t, S_{t-1}, S_{t-2}, \dots, S_{t-k}$)를 사용하여 문제를 해결할 수 있다. 이런 시스템은 부분적으로 관측 가능하며 상태를 관측이라고 한다.

MDP의 최종 목표는 식 (22)와 같이 누적보상을 최대화하는 정책 π 를 찾는 것이다. R_π 는 정책 π 를 따라 각 단계에서 얻은 보상이다. 정책이 MDP의 각 상태에서 가능한 최선의 행동을 했을 때 MDP의 해가 된다. 이 정책을 최적 정책이라 한다.

$$\sum_{t=0}^{\infty} R_\pi(s_t, s_{t+1}) \tag{22}$$

3.1.5 가치 함수(Value Function)

리턴 $G(\tau)$ 은 궤적의 값에 대한 좋은 통찰력을 제공하지만 여전히 방문한 단일 상태의 품질을 나타내지는 않는다. 이 품질지표는 정책에서 차선택을 선택하는데 사용할 수 있기 때문에 중요하다. 정책은 최고 품질의 다음 상태를 초래할 작업을 선택하기만 하면 된다. 가치 함수는 정확히 이를 수행한다. 정책을 따르는 상태에서 예상되는 수익의 관점에서 품질을 추정한다. 가치함수에는 상태가치함수(State value function)와 행동가치함수(Action value function)가 있다. 상태가치함수는 V 로 표기하고 행동가치함수는 Q 로 표기한다. 상태가치함수는 상태에 대해 가치를 출력하는 함수로 상태를 입력 받고 상태의 가치를 출력한다. 정책을 적용해 얻어낸 상태에서 예상 리턴 값은 식 (23)과 같이 정의된다.

$$V_{\pi}(s) = E_{\pi}[G|s_0 = s] = E_{\pi}\left[\sum_{t=0}^k \lambda^t r_t | s_0 = s\right] \quad (23)$$

행동가치함수는 상태에서 특정 행동에 대해 가치를 출력하는 함수로 상태와 행동을 입력으로 받고 해당 상태에서 행동의 가치를 출력한다. 정책을 통해 얻은 상태와 행동에서 예상 리턴 값은 식 (24)과 같이 정의된다.

$$Q_{\pi}(s, a) = E_{\pi}[G|s_0 = s, a_0 = a] = E_{\pi}\left[\sum_{t=0}^k \lambda^t r_t | s_0 = s, a_0 = a\right] \quad (24)$$

상태가치함수와 행동가치함수는 각각 V -함수와 Q -함수라 하며 상태가치함수는 행동가치함수로 식 (25)와 같이 정의할 수 있기 때문에 서로 상관관계를 가진다. 따라서, 최적 Q^* 를 알면 최적 가치함수는 식 (26)과 같다.

$$V_{\pi}(s) = E_{\pi}[Q_{\pi}(s, a)] \quad (25)$$

$$V^*(s) = \max_a Q^*(s, a) \quad (26)$$

최적 행동이 $a^*(s) = \operatorname{argmax}_a Q^*(s, a)$ 이기 때문에 식 (26)이 성립하게 된다.

3.1.6 벨만 방정식(Bellman Equation)

상태가치함수 V 와 행동가치함수 Q 는 정책 π 를 따르는 궤적을 실행한 다음 얻은 값을 평균화하여 추정할 수 있다. 이 방법은 효과적이며 많은 상황에서 사용되지만 리턴에는 전체 궤적의 보상이 필요하다는 점을 고려한다면 문제가 발생하게 된다.

벨만(Bellman) 방정식은 행동가치함수와 상태가치함수를 반복적으로 정의하여 다음 상태에서부터 추정할 수 있다. 벨만방정식은 벨만이 제안한 식으로 현재 상태에서 얻은 보상과 다음 상태의 값을 사용하여 수행한다. 식 (27)의 리턴 값의 재귀 공식을 이용해 상태가치함수에 적용하면 식 (28)과 같이 정의된다.

$$G_t = r_t + \lambda G_{t+1}(\tau) \quad (27)$$

$$\begin{aligned} V_\pi(s) &= E_\pi[G_t | s_0 = s] = E_\pi[r_t + \gamma G_{t+1} | s_0 = s] \\ &= E_\pi[r_t + \gamma V_\pi(s_{t+1}) | s_t = s, a_t \sim \pi(s_t)] \end{aligned} \quad (28)$$

행동가치함수에 대한 벨만방정식은 식 (29)와 같이 정의된다.

$$\begin{aligned} Q_\pi(s, a) &= E_\pi[G_t | s_t = s, a_t = a] = E_\pi[r_t + \gamma G_{t+1} | s_t = s, a_t = a] \\ &= E_\pi[r_t + \gamma Q_\pi(s_{t+1}, a_{t+1}) | s_t = s, a_t = a] \end{aligned} \quad (29)$$

식 (28) ~ (29)를 이용하여 상태가치함수와 행동가치함수는 궤도를 끝까지 풀 필요 없이 현재 시점에 획득한 보상과 다음 가치만으로 연속 상태의 값으로만 업데이트 된다.

3.2 강화학습 알고리즘

3.2.1 강화학습 알고리즘 분류

알고리즘 설계에는 많은 부분이 포함되어 사용자의 실제 요구에 가장 적합한 알고리즘을 결정하기 전에 많은 특성을 고려해야 한다. 따라서 Fig. 12와 같이 강화학습 알고리즘 맵[43]을 통해 알고리즘에 대한 이론과 목표에 따른 알고리즘에 대한 위치와 아이디어를 확인할 수 있다.

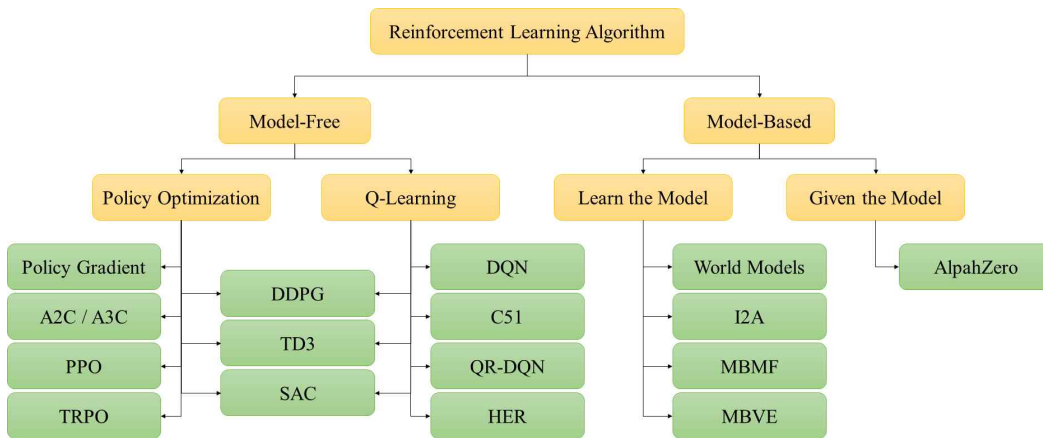


Fig. 12 Reinforcement learning algorithm map

첫 번째 차이점으로 모델-기반(Model-Based) 알고리즘과 모델-프리(Model-Free) 알고리즘 간의 차이이다. 첫 번째 모델-기반 알고리즘은 환경 모델이 필요하고 모델-프리 알고리즘은 환경 모델이 필요 없다. 환경 모델은 원하는 정책을 찾는데 사용할 수 있는 정보를 포함하고 있어 중요하지만 대부분의 경우 모델에서 얻는 것이 어렵다. 이에 모델-프리 알고리즘은 환경에 대한 가정 없이 정보를 학습할 수 있다.

강화학습 알고리즘은 모델-기반 알고리즘과 모델-프리 알고리즘으로 구분되며 모델-프리 알고리즘에서는 정책-기반(Policy-Based)과 가치-기반(Value-Based)으로 구분되며 정책-기반 알고리즘은 정책 기울기(Policy Gradient)와 가치-기반 알고리즘은 Q-학습(Q-Learning)으로 구분할 수 있다.

가치함수 알고리즘이라고 하는 가치-기반 알고리즘은 벨만 방정식을 사용하여 Q-함수(Q-Function)를 학습하고 차례대로 정책을 학습하는데 사용한다. 일반적으로 심

층 신경망을 함수 근사기로 사용하고 다른 트릭을 사용하여 높은 분산과 일반적인 불안정성을 처리한다. 따라서 가치-기반 알고리즘은 지도회귀 알고리즘에 가깝다. 일반적으로 이런 알고리즘은 정책에 맞지 않으므로 데이터를 생성하는데 사용된 것과 동일하게 정책을 최적화 할 필요가 없다. 이런 방법으로 샘플링 된 데이터를 재생 버퍼에 저장할 수 있기 때문에 이전 경험에서 학습을 할 수 있음을 의미한다. 이전 샘플을 사용하는 기능은 가치 함수를 다른 모델-프리 알고리즘보다 샘플 효율성이 높다.

모델-프리 알고리즘의 다른 유형은 정책 기울기(Policy Gradient) 또는 정책 최적화 방법(Policy optimization method)이다. 개선 방향으로 매개변수를 업데이트하여 매개변수 정책에서 직접 학습하기 때문에 강화학습 문제를 보다 직접적이고 명확하게 해석한다. 이는 나쁜 행동을 억제하면서 좋은 행동은 장려되어야 한다는 강화학습 원칙을 기반으로 한다.

가치 함수 알고리즘과 달리 정책 최적화는 주로 On-policy data를 필요로 하므로 알고리즘의 샘플 효율을 비효율적으로 만든다. 정책 최적화 방법은 곡률이 높은 표면이 있는 상태에서 가장 가파른 상승을 하면 쉽게 주어진 방향으로 너무 멀리 이동하면 나쁜 영역으로 떨어질 수 있다는 사실로 인해 매우 불안정할 수 있다. 이런 문제를 해결하기 위해 신뢰 영역 내에서만 정책을 최적화하는 방법(TRPO)과 정리된 서로게이트(Surrogate) 목적함수를 최적화하여 정책에 대한 변경을 제한하는 방법(PPO) 등 많은 알고리즘이 제안되었다. 정책 기울기 방법의 장점은 연속 작업 공간이 있는 환경을 쉽게 처리할 수 있다는 것이다. 이는 상태-행동 쌍에 대한 Q-값을 학습하기 때문에 가치 함수 알고리즘으로 접근하기 어렵다.

액터-크리틱(Actor-Critic, AC) 알고리즘은 정책에 대한 피드백을 제공하기 위해 크리틱(Critic)이라는 가치함수(Q-함수)를 학습하는 정책 기반 정책 기울기 알고리즘이다. 목적지까지 가는데 걸리는 시간을 추정하고 새로운 경로가 더 좋은지 여부를 계산할 수 있다. 이러한 평가는 크리틱(Critic)이 수행한다. 이런 방법으로 최종 목적지에 도달하지 못하더라도 액터(Actor)를 향상시킬 수 있다. 액터와 크리틱을 결합하는 것은 매우 효과적인 것으로 나타나며 정책 기울기 알고리즘에 사용된다. 이런 기술은 신뢰 영역 알고리즘과 같은 정책 최적화에 사용되는 다른 아이디어와 결합할 수도 있다.

3.2.2 동적 프로그래밍(Dynamic Programming, DP)

동적 프로그래밍(Dynamic Programming, DP)는 문제를 겹치는 하위 문제로 나누어 다음 하위 문제의 솔루션을 결합하여 원래 문제에 대한 솔루션을 찾는 일반적인 알고리즘 패러다임이다. DP는 강화학습에 사용할 수 있으며 가장 간단한 접근방식 중 하나이다. 완벽한 환경 모델을 제공하여 최적의 정책을 계산하는데 적합하다[44].

DP는 강화학습 알고리즘 역사에서 중요한 디딤돌이며 차세대 알고리즘의 기초를 제공하지만 계산적으로 많은 자원이 필요하다. DP는 가능한 모든 상태를 고려하여 각 상태가치 또는 행동 가치를 업데이트해야 하므로 제한된 수의 상태 및 행동을 가진 MDP(Markov Decision Process)와 함께 작동한다. 또한 DP 알고리즘은 가치 함수를 배열이나 테이블에 저장한다. 정보를 저장하는 방법은 정보의 손실이 없어 효과적이고 빠르지만 큰 테이블을 저장해야 한다. DP 알고리즘은 테이블을 사용하여 가치 함수를 저장하기 때문에 테이블 형식 학습이라고 한다. 이는 근사 값 함수를 사용하여 인공신경망과 같은 고정크기함수에 값을 저장하는 근사학습과 반대이다.

DP는 부트스트래핑(Bootstrapping)을 사용하는데 다음 상태의 기대 값을 이용하여 상태의 추정 값을 향상시킨다. 부트스트래핑은 벨만 방정식에서 사용한다. DP는 V^* 와 Q^* 를 추정하기 위해 벨만 방정식 식 (28)과 식 (29)를 적용하여 식 (30)과 Q-함수를 사용한 식 (31) 같이 표현한다.

$$V^*(s) = \max_a E[r_t + \gamma V^*(s_{t+1}) | s_t = s, a_t = a] \quad (30)$$

$$Q^*(s, a) = E[r_t + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) | s_t = s, a_t = a] \quad (31)$$

다음으로 최적의 상태-가치함수와 행동-가치함수가 발견되면 기대치를 최대화하는 행동을 수행함으로써 최적의 정책을 찾을 수 있다.

■ 정책 평가(Policy evaluation)

최적 정책을 찾으려면 먼저 최적의 가치함수를 찾아야 한다. 이를 수행하는 반복적인 과정을 정책 평가라고 한다. 모델의 상태 가치 전이(State value transition), 시간 t 에서 취한 행동의 보상 값, 감가율(Discount factor) γ 을 적용한 상태의 가치함

수 값을 이용하여 정책 π 에 대한 가치함수를 시간 순으로 반복적으로 개선한 $\{V_0, \dots, V_k\}$ 를 만든다. 정책 평가는 벨만 방정식을 이용해 가치함수를 개선하는 시퀀스는 식 (32)와 같다.

$$\begin{aligned} V_{k+1}(s) &= E_{\pi} [r_t + \gamma V_k(s_{t+1}) | s_t = s] \\ &= \sum_a \pi(s, a) \sum_{s', r} p(s' | s, a) [r + \gamma V_k(s')] \end{aligned} \quad (32)$$

가치함수 식 (32)은 모든 상태와 행동에 대한 상태 전이 함수 p 와 보상 함수 r 을 알고 있는 경우에만 업데이트를 할 수 있으므로 환경 모델을 완전히 알고 있는 경우에만 가능하다. 정책이 각 작업에 대한 확률을 출력하기 때문에 확률적 정책에는 식 (32)의 행동에 대한 합산이 필요하다. 단순성을 위해 결정적 정책만 고려하여 가치 함수가 개선되면 더 나은 정책을 찾는데 사용할 수 있다. 이 절차를 정책 개선(Policy improvement)라고 한다. 정책 개선은 식 (33)과 같이 정책 π 를 찾아낸다.

$$\pi' = \operatorname{argmax}_a Q_{\pi}(s, a) = \operatorname{argmax}_a \sum_{s', r} p(s' | s, a) [r + \gamma V_{\pi}(s')] \quad (33)$$

기존 정책의 가치 함수 V_{π} 에서 정책 π' 를 생성한다. 새로운 정책 π' 는 항상 π 보다 우수하며 V 가 최적인 경우에만 정책이 최적이다. 정책 평가(Policy evaluation)와 정책 개선(Policy improvement)의 조합은 최적의 정책을 계산하는 두 가지 알고리즘을 발생시킨다. 정책 반복(Policy iteration)과 가치 반복(Value iteration)이다. 둘 다 정책 평가를 사용하여 가치 함수를 단조롭게 개선하고 정책 개선을 사용하여 새 정책을 추정한다. 차이점은 정책 반복은 두 단계를 주기적으로 실행하는 반면에 가치 반복은 단일 업데이트로 결합하여 처리한다.

■ 정책 반복(Policy iteration)

식 (32)를 사용하여 현재 정책 π 에서 V_{π} 를 업데이트하는 정책 평가와 정책 개선 식 (33) 사이의 정책 반복 주기는 개선된 가치 함수 V_{π} 를 사용하여 π' 를 계산한다. 결국, n 사이클 후에 알고리즘은 최적의 정책 π^* 를 생성한다.

초기화 단계 후 외부 루프는 안정적인 정책을 찾을 때까지 정책 평가 및 정책 반복을 반복한다. 이러한 각 반복에서 정책 평가는 이전 정책 개선 단계에서 발견된 정책을 평가하고 차례로 추정 가치 함수를 사용한다.

■ 가치 반복(Value iteration)

가치 반복은 MDP에서 최적의 가치를 찾기 위한 동적 프로그래밍 알고리즘이지만 루프에서 정책 평가 및 정책 반복을 수행하는 정책 반복과 달리 가치 반복은 식 (34)와 같이 두 가지 방법을 단일 업데이트로 결합한다. 최상의 행동을 즉시 선택하여 상태 값을 업데이트한다.

$$V_{k+1}(s) = \max_a \sum_{s', r} p(s'|s, a) [r + \gamma V_k(s')] \quad (34)$$

유일한 차이점은 새로운 가치 추정 업데이트와 적절한 정책 반복 모듈이 없다는 것이다. 결과적으로 최적의 정책은 식 (35)과 같다.

$$\pi^* = \operatorname{argmax}_a \sum_{s', r} p(s'|s, a) [r + \gamma V^*(s)] \quad (35)$$

3.2.3 시간차 학습(Temporal Difference Learning, TD)

몬테카를로(Monte carlo) 방법은 환경에서 샘플링하여 직접 학습하는 방법이지만 전체적인 궤적에 의존하는 단점이 있다. 이는 에피소드가 끝날 때까지 기다려 상태 가치를 업데이트 할 수 있다. 따라서 중요한 요소는 궤적의 끝이 없는 경우 또는 매우 긴 경우 어떤 일이 발생하는지 알 수 있다. 이 문제에 대한 유사한 솔루션인 DP 알고리즘에서 이미 나타났다. 여기서 상태 가치는 끝날 때까지 기다리지 않고 각 단계에서 업데이트 된다. 궤적 동안 누적된 리턴 값을 사용하는 대신 즉각적인 보상과 다음 상태 가치의 추정을 사용한다. 단일 학습 단계와 관련된 이 기술을 부트스트래핑이라고 하며 Fig. 13과 같다. 잠재적으로 무한한 에피소드에 유용할 뿐만 아니라 모든 길이의 에피소드에 유용하다. 첫 번째 이유는 기대 리턴 값의 분산을 줄이는 데 도움이 되기 때문이다. 상태 가치는 궤적의 모든 보상이 아니라 바로 다음 보상에만 의존하기 때문에 분산이 감소한다. 두 번째 이유는 학습 프로세스가 모든 단계에서 발생하여 이런 알고리즘이 온라인으로 학습하도록 만들기 때문이다. 반면 몬테카를로 방법은 에피소드가 끝난 후에만 정보를 사용하기 때문에 오프라인 상태이다. 따라서 부트스트랩을 사용하여 온라인으로 학습하는 방법을 시간차 학습(Temporal Difference Learning TD)이라고 한다.

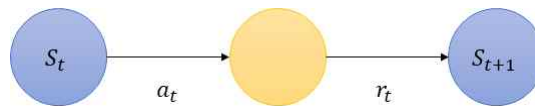


Fig. 13 Bootstrapping update

시간차 학습은 몬테카를로 방법(Monte carlo method)과 동적 프로그래밍(Dynamic programming)의 조합으로 볼 수 있다. 몬테카를로 방법의 샘플링 아이디어와 동적 프로그래밍의 부트스트랩 아이디어를 사용했기 때문이다. 시간차 학습은 강화학습 알고리즘 전반에 걸쳐 널리 사용되며 이러한 알고리즘의 핵심으로 구성한다. SARSA와 Q-Learning 모두 원-스텝, 테이블 구조, 모델-프리의 시간차 학습 방법이다.

동적 프로그래밍을 이용한 문제 해결을 통해 다음과 같이 식 (36)을 알고 있다.

$$V_{\pi}(s) = E_{\pi}[G_t | s_t = s] \quad (36)$$

경험적으로 몬테카를로 업데이트는 여러 궤적을 평균화하여 이 값을 추정한다. 식 (36)을 전개하면 식 (37)과 같다.

$$\begin{aligned} & E_{\pi}[G_t | s_t = s] \\ &= E_{\pi}[r_t + \gamma G_{t+1} | s_t = s] \\ &= E_{\pi}[r_t + \gamma V_{\pi}(s_{t+1}) | s_t = s] \end{aligned} \quad (37)$$

앞의 식은 DP 알고리즘에 의해 근사된다. 차이점으로는 TD 알고리즘이 계산하는 대신 예상 가치를 추정한다는 것이다. 추정은 식 (38)과 같이 몬테카를로 방법과 동일한 방식으로 평균화하여 추정한다.

$$E_{\pi}[r_t + \gamma V_{\pi}(s_{t+1}) | s_t = s] \approx \frac{1}{N} \sum_{i=0}^N \pi[r_t^i + \gamma V_{\pi}(s_{t+1}^i) | s_t = s] \quad (38)$$

실제로 평균을 계산하는 대신 상태 값을 최적의 값으로 조금씩 개선하여 식 (39)와 같이 TD 업데이트를 수행한다.

$$V(s_t) \leftarrow V(s_t) + \alpha [r + \gamma V(s_{t+1}) - V(s_t)] \quad (39)$$

α 는 업데이트에서 상태 가치가 얼마나 변경되어야 하는지를 설정하는 상수이다. $\alpha=0$ 이면 상태 가치는 전혀 변경되지 않는다. 대신에 $\alpha=1$ 이면 상태 가치는 TD 목표라고 하는 $r + \gamma V(s_{t+1})$ 와 같으며 이전 가치를 완전히 잊어버리게 된다. 실제로 이런 극단적인 경우를 원치 않으며 일반적으로 0.5~0.001 사이의 값을 사용한다.

■ SARSA(State-Action-Reward-State-Action)

주어진 정책에 대한 가치함수를 추정하는 일반적인 방법으로 시간차 학습을 제안했다. 실제로 시간차 학습은 정책을 개선하기 위한 요소가 없어 그대로 사용할 수

없다. SARSA와 Q-Learning은 가치 함수를 추정하고 정책을 최적화하는 원-스텝, 테이블 형식의 TD 알고리즘이며 실제로 다양한 강화학습 문제에서 사용할 수 있다. SARSA를 사용하여 주어진 MDP에 대한 최적의 정책을 학습한다.

TD 학습의 우려 사항은 상태 가치를 추정한다는 것이다. 주어진 상태에서 다음 상태 가치가 가장 높은 행동을 어떻게 선택할 수 있는지는 앞서 에이전트를 가장 높은 가치를 가진 상태로 이동시킬 행동을 선택해야 한다. 그러나 가능한 다음 상태 목록을 제공하는 환경 모델이 없으면 어떤 작업이 에이전트를 해당 상태로 이동하는지 알 수 없다. SARSA는 가치 함수를 학습하는 대신 상태-행동 함수 Q 를 학습하고 적용한다. $Q(s, a)$ 는 행동 a 가 취해지면 상태 s 의 가치를 알려준다.

기본적으로 TD 업데이트에 대해 수행한 모든 관측(Observation)은 SARSA에도 유효하다. Q-function 정의에 적용하면 식 (40)과 같이 SARSA 업데이트를 얻는다.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (40)$$

α 는 행동 가치가 얼마나 업데이트 되었는지 결정하는 계수이고, γ 는 감가율(Discount Factor)로 미래의 결정에서 오는 가치에 덜 중요하게 여기는데 사용되는 0과 1 사이의 계수이다. SARSA 업데이트의 시각적 해석은 Fig. 14와 같다.

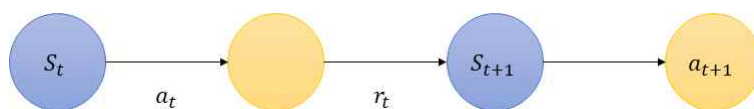


Fig. 14 SARSA update

Fig. 14와 같이 SARSA라는 이름은 상태 s_t , 행동 a_t , 보상 r_t , 다음 상태 s_{t+1} , 다음 행동 a_{t+1} 을 기반으로 하는 업데이트에서 가져왔다. 이 모든 요소를 합치면 s, a, r, s, a 가 된다.

SARSA는 On-policy 알고리즘이다. On-policy는 환경과 상호작용을 통해 경험을 수집하는데 사용되는 정책이 업데이트되는 동일한 정책을 의미한다. 이 방법의 On-policy 특성은 현재 정책을 사용하여 다음 행동 a_{t+1} 을 선택하고 $Q(s_{t+1}, a_{t+1})$ 를 추정하고 다음 행동에서 동일한 정책을 따를 것이라는 가정으로 인해 발생한다.

On-policy 알고리즘은 일반적으로 Off-policy 알고리즘보다 쉽지만 덜 강력하고 학습하는데 많은 데이터가 필요하다. 그럼에도 불구하고 TD 학습의 경우 SARSA는 모든 상태-행동(State-Action)을 무한 횟수 방문하여 시간이 지남에 따라 정책이 안정화되어 결정론적인 정책이 되면 최적의 정책으로 수렴하는 것이 보장된다. 실용적이 알고리즘은 0 또는 0에 가까운 값이 되는 경향이 있는 감쇠와 함께 ϵ -greedy 정책을 사용한다.

■ Q-Learning

Q-Learning은 SARSA의 유용하고 독특한 기능을 가지는 또 다른 TD 알고리즘이다. Q-Learning은 TD 학습에서 원-스텝 학습의 모든 특성인 TD 학습에서 각 단계에서 학습하는 능력과 적절한 환경 모델 없이 경험에서 배우는 특성을 상속한다.

SARSA에 비해 Q-Learning의 가장 큰 특징인 Off-policy 알고리즘이다. 정책 외에는 경험을 수집하는 정책과 독립적 업데이트를 수행할 수 있다. Off-policy 알고리즘은 이전 경험을 사용하여 정책을 개선할 수 있다. 환경과 상호작용하는 정책과 실제로 개선되는 정책을 구별하기 위해 전자를 행동 정책(Behavior policy), 후자를 대상 정책(Target policy)이라 한다.

Q-Learning은 현재 최적의 행동 가치를 사용하여 Q-function을 근사화하는 것이다. Q-Learning 업데이트는 식 (41)과 같이 최대 상태-행동 값을 취한다는 점을 제외하고 SARSA에서 수행된 업데이트와 유사하다.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (41)$$

α 는 일반적인 학습률(Learning rate)이고 γ 는 감가율(Discount factor)이다.

SARSA 업데이트는 Fig. 15와 같이 ϵ -greedy 정책과 같은 행동정책에서 수행되는 반면 Q 업데이트는 최대 행동 가치에서 발생하는 탐욕적 대상(Greedy target) 정책에서 수행된다. SARSA에서 행동 a_t 와 a_{t+1} 이 모두 동일한 정책에서 오는 반면 Q-Learning에서는 다음 최대 상태 행동 가치를 기반으로 행동 a_{t+1} 이 선택된다. Q-Learning의 업데이트는 행동 정책에 의존하지 않기 때문에 정책 외 알고리즘이 된다.

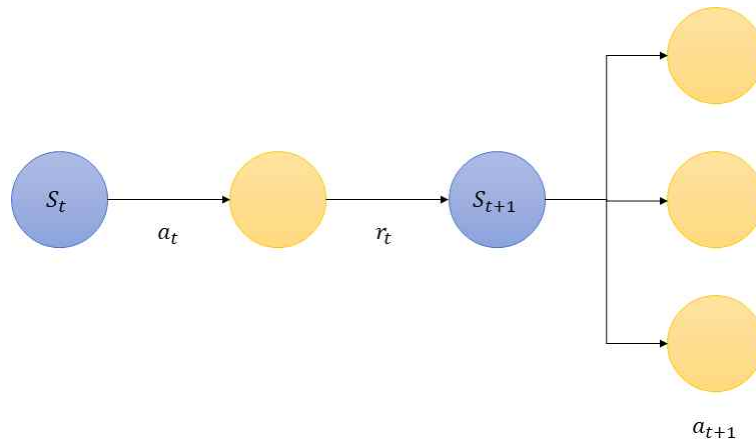


Fig. 15 Q-Learning update

Q-Learning은 TD 방법으로 시간이 지남에 따라 결정론적 정책으로 수렴되는 행동 정책이 필요하다. 좋은 전략은 선형 또는 지수 감쇠가 있는 SARSA의 경우처럼 ϵ -greedy 정책을 사용한다.

3.3 심층강화학습 알고리즘

3.3.1 정책 기울기(Policy Gradient, PG)

지금까지 학습하고 개발된 알고리즘은 핵심에서 가치 함수 $V(s)$ 또는 행동 가치 함수 $Q(s, a)$ 를 학습하는 가치 기반입니다. 가치 함수는 주어진 상태 또는 상태-행동 쌍에서 누적될 수 있는 총 보상을 정의하는 함수이다. 그런 다음 예상 행동 또는 예상 상태 가치를 기반으로 행동을 선택할 수 있다. 따라서 greedy 정책은 식 (42)와 같이 정의할 수 있다.

$$\pi(s) = \operatorname{argmax}_a Q(s, a) \quad (42)$$

가치 기반 방법은 심층 신경망과 결합하면 고차원 공간에서 작동하는 에이전트를 제어하여 정교한 정책을 학습할 수 있다. 이런 훌륭한 특성에도 불구하고 많은 수의 행동으로 문제를 처리하거나 행동 공간이 연속적일 경우 어려움이 있다. 이런 경우에는 최대 작동이 불가능하다. 정책 기울기(Policy Gradient, PG) 알고리즘은 연속 행동 공간에서 쉽게 적용 할 수 있다. PG 알고리즘의 특징은 정책의 기울기를 사용하므로 정책 기울기라는 이름을 사용한다.

강화학습의 목적은 궤적의 총 보상(Total reward), 할인(Discounted) 또는 할인 되지 않는(Undiscounted) 예상 보상(Expected return)을 최대화하는 것이다. 따라서 목적 함수는 식 (43)과 같이 표현할 수 있다.

$$\mathcal{J}(\theta) = E_{\tau \sim \pi_\theta} [R(\tau)] \quad (43)$$

여기서, θ 는 심층 신경망 학습이 가능한 변수와 같은 정책의 매개변수이다. PG방법에서 목적 함수의 최대화는 목적 함수 $\nabla_{\theta} \mathcal{J}(\theta)$ 의 기울기를 통해 수행된다. 기울기 상승을 사용하면 기울기는 함수가 증가하는 방향을 가리키며 기울기 방향으로 매개변수를 이동하여 $\mathcal{J}(\theta)$ 를 향상시킬 수 있다.

최대 값이 발견되면 정책 π_θ 는 가능한 높은 수익을 가진 궤적을 생성한다. 직관적인 수준에서 정책 기울기는 확률을 높여주고 나쁜 정책은 처벌하고 좋은 정책에는

인센티브를 제공한다. 식 (43)을 사용해서 목적 함수의 기울기를 식 (44)와 같이 정의한다.

$$\nabla_{\theta} \mathcal{J}(\theta) = \nabla_{\theta} E_{\tau \sim \pi_{\theta}} [R(\tau)] \quad (44)$$

정책 기울기 방법에서 정책 평가는 보상 R 의 추정이다. 대신 정책 개선은 매개변수 θ 의 최적화 단계이다. 따라서 정책 기울기 방법은 정책을 개선하기 위해 두 단계를 공생적으로 수행한다. 공식화에서 목적 함수 기울기가 정책 상태의 분포에 따라 달라져서 식 (44)를 볼 때 초기 문제가 발생한다.

$$\nabla_{\theta} \mathcal{J}(\theta) = \nabla_{\theta} E_{\tau \sim \pi_{\theta}} [R(\tau)] = \nabla_{\theta} \sum_s d(s) \sum_a \pi_{\theta}(a|s) R(s, a) \quad (45)$$

기대치의 확률적 근사치를 사용하지만 상태 분포 $d(s)$ 를 계산하기 위해서는 여전히 완전한 환경 모델이 필요하다. 따라서 식 (45)는 목적에 적합하지 않다. 정책 기울기 정리를 여기서 구할 수 있다. 목적은 상태 분포의 도함수를 포함하지 않고 정책의 매개변수와 관련하여 목적 함수의 기울기를 계산하는 분석 공식을 제공한다. 공식적으로 정책 기울기 정리를 통해 목적 함수의 기울기를 식 (46)과 같이 표현한다.

$$\nabla_{\theta} \mathcal{J}(\theta) = \nabla_{\theta} E_{\tau \sim \pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(\tau) R(\tau)] = E_{\pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(a|s) Q_{\pi_{\theta}}(s, a)] \quad (46)$$

목표 도함수가 상태 분포의 도함수를 포함하지 않으므로 정책에서 샘플링하여 기대치를 추정할 수 있다. 목적의 미분은 식 (47)과 같이 근사될 수 있다.

$$\nabla_{\theta} \mathcal{J}(\theta) = \frac{1}{N} \sum_{i=0}^N [\nabla_{\theta} \log \pi_{\theta}(a_i|s_i) Q_{\pi_{\theta}}(s_i, a_i)] \quad (47)$$

식 (48)은 기울기 상승으로 확률적 업데이트를 생성하는데 사용한다.

$$\theta = \theta + \alpha \nabla_{\theta} \mathcal{J}(\theta) \quad (48)$$

목적 함수를 최대화하는 것이 목표이므로 $\theta = \theta + \alpha \nabla_{\theta} J(\theta)$ 을 수행하는 기울기 하강과 반대로 기울기 상승은 기울기와 같은 방향으로 매개변수를 이동하는데 사용된다. 식 (47)의 배경은 나쁜 행동의 확률을 줄여주는 동시에 좋은 행동이 미래에 다시 제안될 확률을 높이는 것이다. 동작 품질은 상태-행동 쌍의 품질을 제공하는 $Q_{\pi_{\theta}}(s_i, a_i)$ 의 일반적인 스칼라 값에 의해 수행된다.

3.3.2 액터-크리틱(Actor-Critic, AC)

간단한 강화학습은 편향되지 않는 특징이 있지만 높은 분산을 보인다. 베이스라인(Baseline)을 추가하면 점근적으로 알고리즘은 로컬 최소 값으로 수렴되므로 분산이 감소하면서 편향되지 않는다. 베이스라인이 있는 강화학습의 주요 단점은 매우 느리게 수렴되어 환경과의 일관된 수의 상호작용이 필요하다는 것이다. 훈련 속도를 높이는 접근 방식을 부트스트래핑(Bootstrapping)이라고 한다. 이는 여러 번 접한 기술이며 후속 상태 가치에서 반환 가치를 추정할 수 있다. 이를 사용하는 정책 기울기 알고리즘을 액터-크리틱(Actor-Critic, AC)이라고 한다. AC 알고리즘에서 액터는 정책이고 크리틱은 액터의 행동을 평가하여 빨리 학습할 수 있도록 돕는 상태-가치 함수이다. AC 방법의 장점은 여러 가지이지만 가장 중요한 것은 비-에피소드(Non-episodic) 문제를 학습하는 능력이다. 강화학습으로 연속 작업을 해결하는 것은 불가능하다. 미래 보상을 계산하려면 궤적의 끝까지 모든 보상이 필요하다. 부트스트래핑 기술에 의존하는 AC 방법은 불완전한 궤적에서 행동 가치를 학습할 수도 있다. 원-스텝 부트스트래핑을 사용하는 행동-가치 함수는 식 (48)과 같이 정의된다.

$$Q(s, a) = r + \gamma V(s') \quad (48)$$

여기서, s' 는 다음 상태이며, 식 (49)와 같이 부트스트랩을 사용하는 액터 π_θ 와 크리틱 V_ω 을 사용하면 원-스텝 AC 스텝을 얻을 수 있다. 식 (49)는 식 (50)과 같이 강화학습 단계를 베이스라인으로 대체한다.

$$\theta = \theta + \alpha (r_t + \gamma V_\omega(s'_t) - V_\omega(s_t)) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \quad (49)$$

$$\theta = \theta + \alpha (G_t - V_\omega(s_t)) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \quad (50)$$

강화학습과 AC에서 상태-가치 함수를 사용하는 것의 차이점에 주목해야 한다. 강화학습에서는 현재상태의 상태 가치를 제공하기 위해 베이스라인으로만 사용된다. AC에서 상태-가치 함수는 다음 상태의 가치를 추정하는데 사용되어 현재 보상만 $Q(s, a)$ 를 추정하도록 한다. 따라서 원-스텝 AC 모델은 완전한 온라인 증분(Fully on line incremental) 알고리즘이라고 할 수 있다.

3.3.3 근위 정책 최적화(Proximal Policy Optimization, PPO)

Schulman은 실제로 방법의 복잡성을 줄이면서 TRPO(Trust Region Policy Optimization)와 유사한 아이디어를 사용한 연구에서 가능하다는 것을 보여준다. 이 방법을 Proximal Policy Optimization(PPO)라고 한다. TRPO에 비해 신뢰성을 떨어뜨리지 않고 1차 최적화만 사용하는 것이 장점이다. PPO는 또한 TRPO보다 더 일반적이고 샘플 효율성이 높으며 미니 배치로 다중 업데이트가 가능하다. PPO의 기본 아이디어는 TRPO에서와 같이 써로게이트 목적함수를 제한하는 대신 써로게이트 목적 함수를 이동할 때 클립하는 것이다. 이렇게 하면 너무 큰 업데이트를 수행하는 것을 방지할 수 있다. 목적함수는 식 (51)과 같다.

$$\mathcal{L}^{CLIP}(\theta) = E_{s \sim p_{old}, a \sim \pi_{old}} [\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)] \quad (51)$$

여기서, $r_t(\theta)$ 는 식 (52)와 같이 정의된다.

$$r_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \quad (52)$$

목적함수는 새로운 정책과 기존 정책 간의 확률 비율 $r_t(\theta)$ 가 상수 ϵ 보다 높거나 낮으면 최소 값을 취해야한다. 이것은 r_t 가 구간 $[1-\epsilon, 1+\epsilon]$ 밖으로 이동하는 것을 방지한다. 1의 값은 기준으로 사용되며 $r_t(\theta_{old})=1$ 이다. PPO 논문에서 소개된 실용적인 알고리즘은 논문[45]에서 처음 소개된 아이디어인 일반화된 이점 추정인 GAE(Generalized Advantage Estimation)의 정리된 버전(Truncated version)을 사용한다. GAE는 식 (53)과 같이 이점을 계산한다.

$$A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \dots + (\gamma\lambda)^{T-t+1}\delta_{t+1} \quad (53)$$

$$\text{where } \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

공통 이점 추정기 대신 식 (54)를 사용한다.

$$A_t = r_t + \gamma t_{t+1} + \dots + \gamma^{T-t+1} r_{T-1} V(s_T) \quad (54)$$

PPO 알고리즘은 계속 사용하면 반복에서 여러 병렬 액터의 N개 궤적이 시간 범위 T로 수집되고 정책은 미니 배치로 K번 업데이트 된다. 이러한 추세에 따라 크리틱은 미니 배치를 사용하여 여러 번 업데이트 될 수도 있다. Table 3은 모든 PPO 하이퍼 파라미터 및 계수의 표준 값이 포함되어 있다. 모든 문제에 임시 하이퍼 파라미터가 필요하다는 사실에도 불구하고 해당 범위에 대한 아이디어를 얻는 것에 유용할 것이다.

Table 3 Range of PPO hyperparameters

Hyperparameter	Symbol	Range
Policy learning rate	-	$[1e^{-5}, 1e^{-3}]$
Number of policy iterations	K	[3, 15]
Number of trajectories (equivalent to the number of parallel actors)	N	[1, 20]
Time horizon	T	[64, 5120]
Mini-batch size	-	[64, 5120]
Clipping coefficient	ϵ	0.1 or 0.2
Delta (for GAE)	δ	[0.9, 0.97]
Gamma (for GAE)	γ	[0.8, 0.995]

4. 스마트 필드로봇 시스템 모델

4.1 스마트 필드로봇 시스템 모델 구성

스마트 필드로봇 시스템 모델의 구성은 Fig. 16과 같이 3가지의 파트로 구성되어 있다. 첫 번째는 기구로 필드로봇의 상부체 캐빈, 하부체 크롤러, 붐, 암, 버킷, 액추에이터 3D 모델과 틸트로테이터의 틸팅부, 로테이터부, 필드로봇의 암과 틸트로테이터를 연결해주는 브라켓 3D 모델과 필드로봇과 틸트로테이터의 Simscape multibody 모델로 구성되어 있다. 두 번째는 유압부로 연구 대상인 1.5톤 필드로봇의 메인 펌프, Main Control Valve, 필드로봇 붐, 암, 버킷 실린더, 스윙 모터와 틸트로테이터의 틸팅 실린더, 틸트로테이터의 로테이터 모터의 유압 시스템으로 구성되어 있다. 세 번째는 필드로봇의 메인 컨트롤 밸브의 스톱을 제어하기 위한 제어기로 PID 제어기로 구성되어 있다.

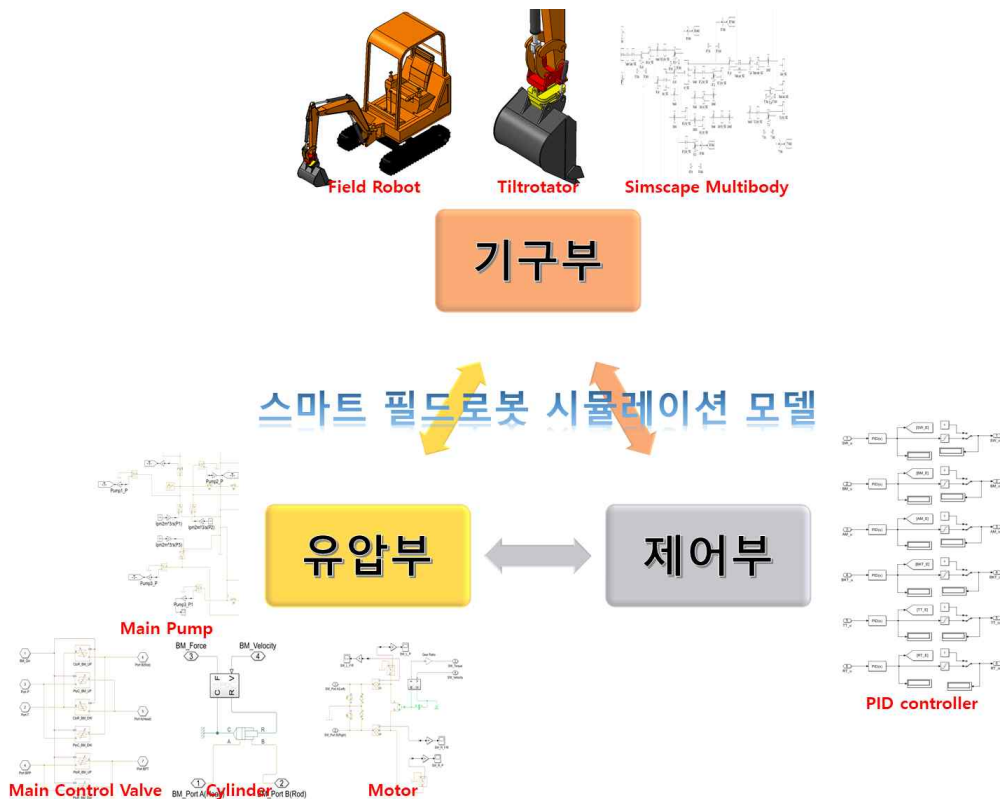


Fig. 16 Configuration of smart field robot simulation model

4.2 스마트 필드로봇 기구 모델

스마트 필드로봇의 기구 모델은 1.5톤 필드로봇과 틸트로테이터 두 가지로 구분할 수 있다. 그리고 기구 모델은 3D 모델과 Simscape multibody 모델로 구분할 수 있다. 먼저, 3D 모델은 CATIA를 이용하여 1.5톤 필드로봇의 상부체 캐빈, 하부체 크롤러, 붐, 암, 버킷, 각 관절의 실린더와 틸트로테이터의 틸팅 파트, 로테이터 파트, 필드로봇의 암과 틸트로테이터를 연결하는 브라켓을 3D 모델링하였다. 1.5톤 필드로봇과 틸트로테이터의 3D 모델은 Fig. 17, Fig. 18과 같다.

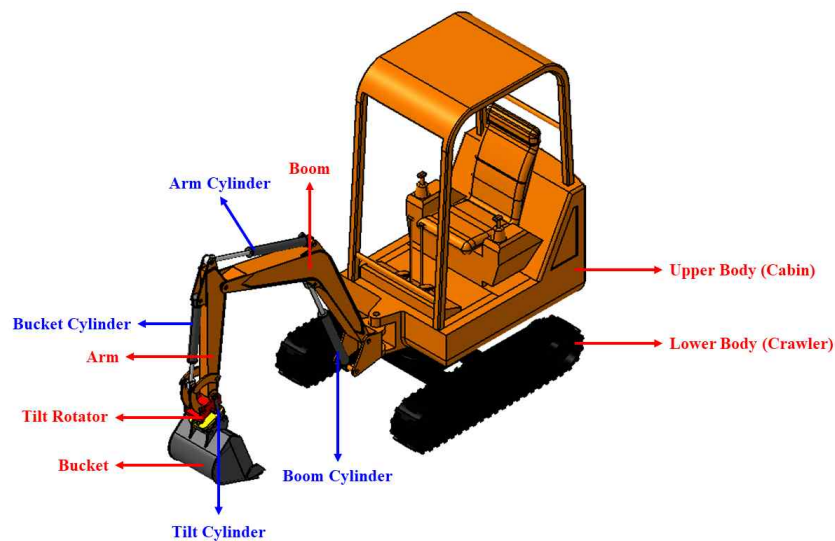


Fig. 17 3D model of 1.5ton field robot using CATIA

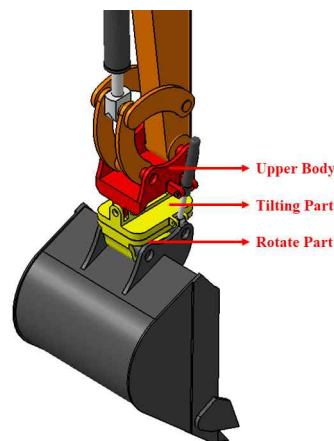


Fig. 18 3D model of tiltrotator using CATIA

다음으로, 필드로봇의 동작을 구현하기 위해 동력원인 유압모델과 각 관절에서 적용되는 힘과 토크 값을 나타낼 수 있는 기구/동역학 모델이 필요하다. 이에 대해 기구/동역학 모델인 Simscape multibody 모델은 MATLAB/Simulink의 Toolbox를 이용하여 모델링하였다. Simscape multibody 모델은 CATIA에서 모델링한 모델을 Solidworks에 import하여 STP 파일에서 XML 파일로 변환하는 과정이 필요하다. 여기서 export된 multibody 모델에서는 CATIA에서 모델링된 assembly에 대한 정보를 담은 data file이 같이 생성된다. 이렇게 생성된 파일을 MATLAB/Simscape에 import하면 Fig. 19와 같이 multibody 모델링하였다[46][47][48].

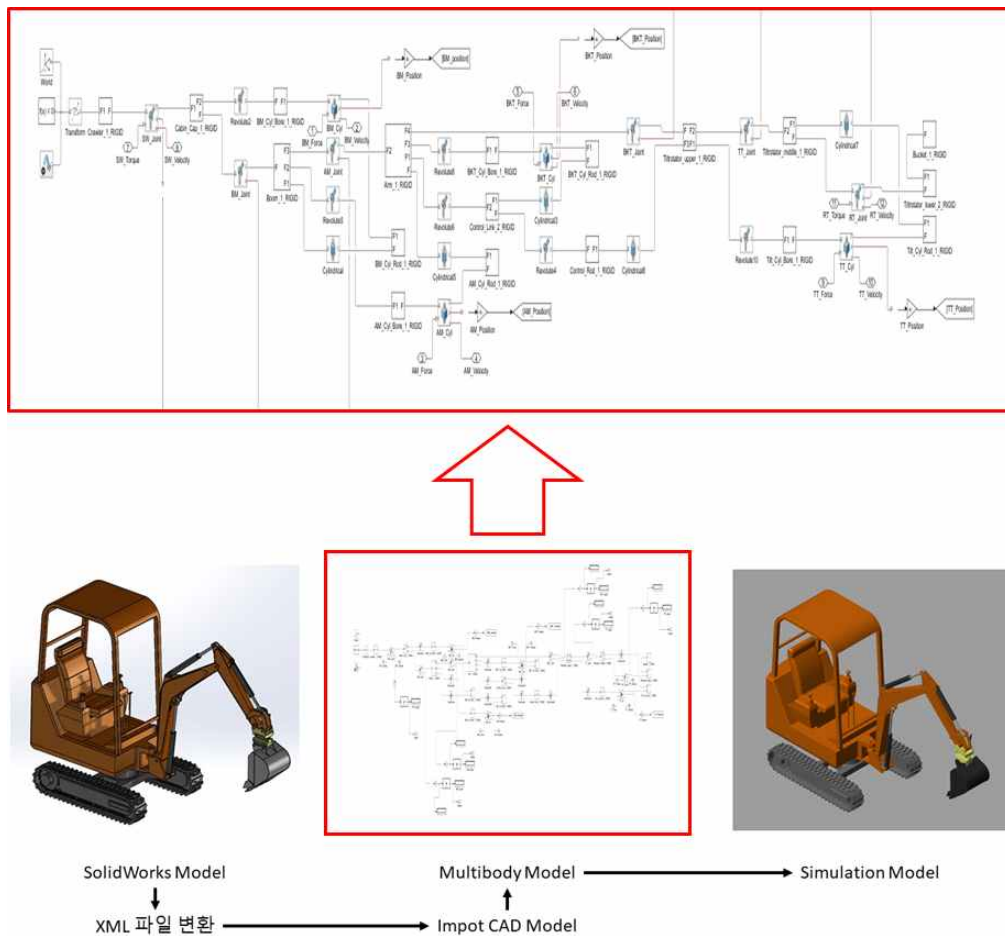


Fig. 19 Simscape multibody model of smart field robot

4.3 스마트 필드로봇 유압 모델

스마트 필드로봇의 유압 모델은 1.5톤 필드로봇의 유압회로도인 Fig. 20을 MATLAB/Simulink의 Toolbox인 Simscape fluids를 이용하여 Main pump, Main control valve, Cylinder, Motor의 제원을 활용하여 모델링하였다[49][50][51][52]. 틸트로테이터의 유압모델은 Fig. 21과 같이 SMP社의 틸트로테이터 유압회로도를 참고하여 1개의 틸트 실린더를 가지는 틸트로테이터의 Tilt cylinder, Rotation motor를 모델링하였다[53].

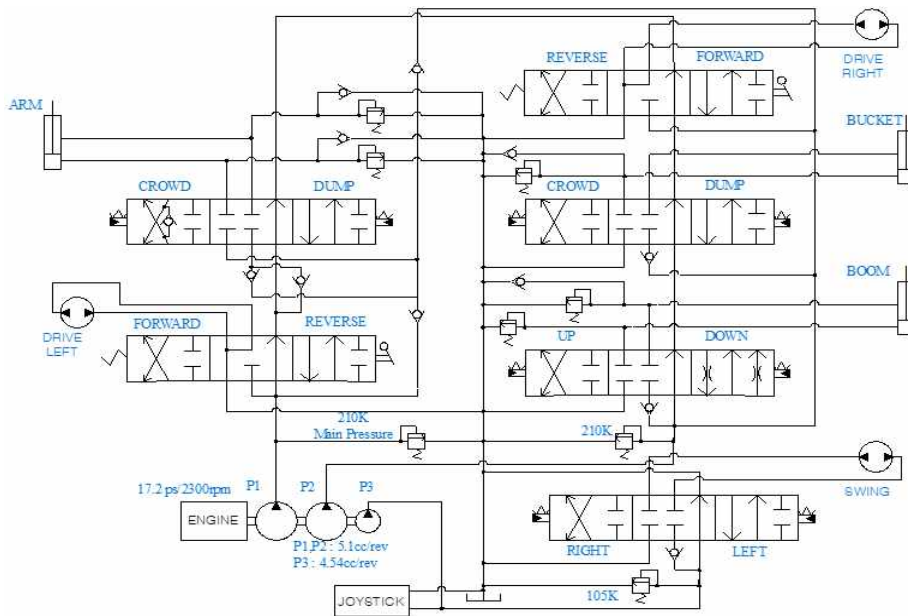


Fig. 20 Hydraulic circuit of 1.5 ton field robot

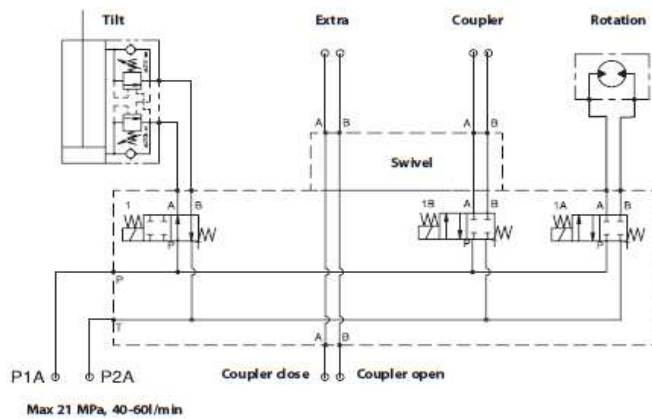


Fig. 21 Hydraulic circuit of Tiltrotator(SMP)

1.5톤 필드로봇의 유압모델에서는 Pump, Main control valve, cylinder, motor를 모델링하였다. 첫 번째 1.5톤 필드로봇의 Main pump는 Fig. 22에서 P1과 P2이고 고정형 액시얼 피스톤 펌프(Fixed axial piston pump)로 2 x 6.1 cc/rev 용적을 가지고 2 x 14.6 l/min 유량을 토출하는 펌프로 봄, 압 버킷으로 유량을 전달한다. 스윙으로 유량을 전달하는 펌프는 Fig. 21에서 P3이고 기어 펌프로 4.5 cc/rev 용적을 가지고 10.8 l/min 유량을 토출하는 펌프이다. Main pump와 swing pump는 고정형으로 일정 유량을 토출하도록 모델링하였다. 토출되는 유량에 따라 각 펌프에서 발생하는 압력을 센싱하여 값을 계측할 수 있도록 모델링하였다.

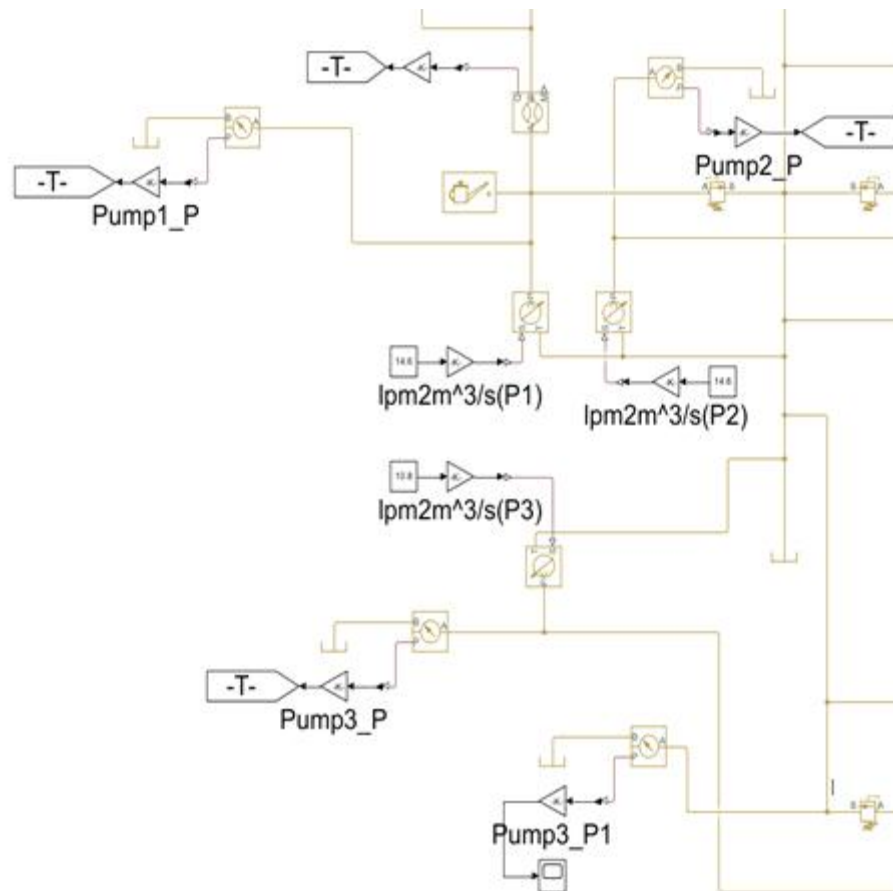


Fig. 22 Hydraulic pump model of 1.5 ton field robot

1.5톤 필드로봇의 Main control valve 모델은 6port 3way 밸브로 Bypass를 포함하고 있다. Simscape fluids library에는 해당되는 밸브 블록이 없어 오리피스를 이용하여 밸브를 모델링하였다. 오리피스 설정은 붐, 암, 버켓, 스윙의 동작에 따라 변화되는 스톱의 스트로크에 따른 개구면적을 이용하였다. 틸트로테이터에 대한 밸브 설정은 버켓의 밸브 설정 값과 개구면적 선도와 동일하게 적용하였다. 여기서, 밸브의 스톱 스트로크 값은 -7 ~ 7 mm, 밸브의 개구면적은 70 mm² 이다. 스톱 스트로크에 따른 개구면적은 붐(Up-Down) Fig. 23, 암(In-Out) Fig. 24, 버켓(In-Out) Fig. 25, 스윙(Left-Right) Fig. 26과 같다. 개구면적선도에서 P는 Pump, R은 Return, C는 Cylinder(액추에이터)를 나타낸다.

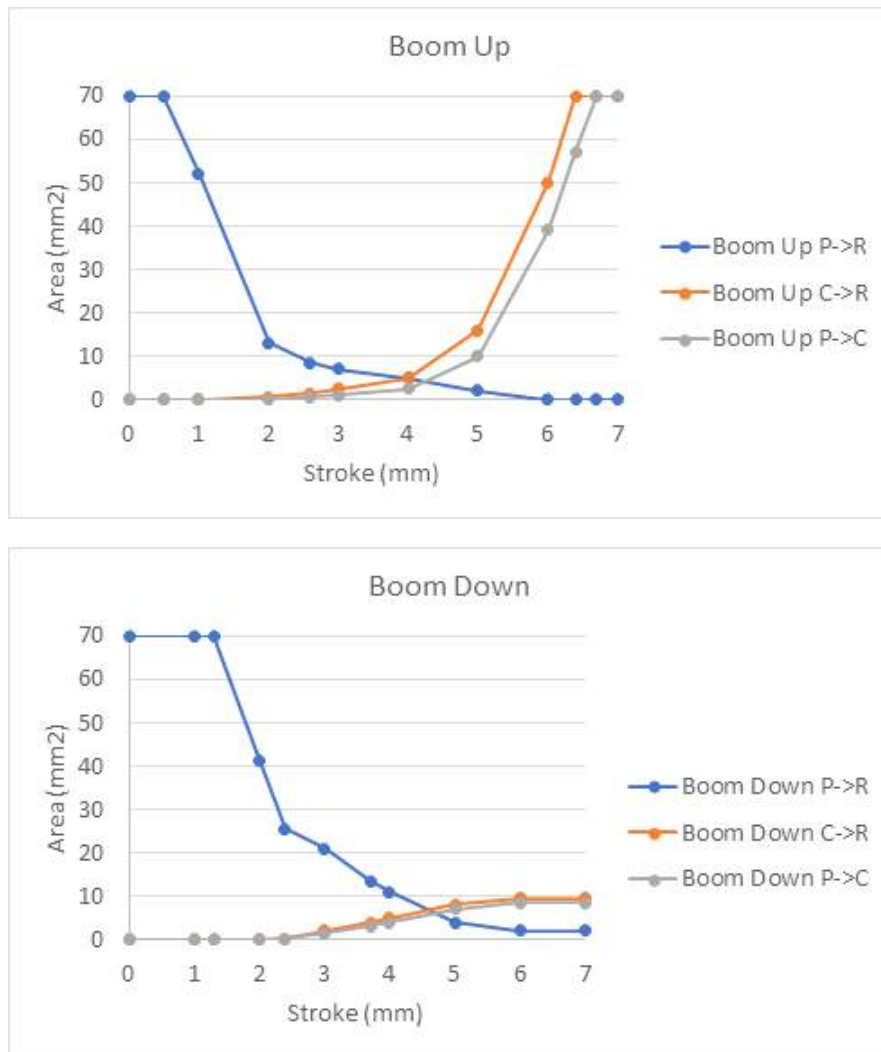


Fig. 23 Opening area diagram of boom valve

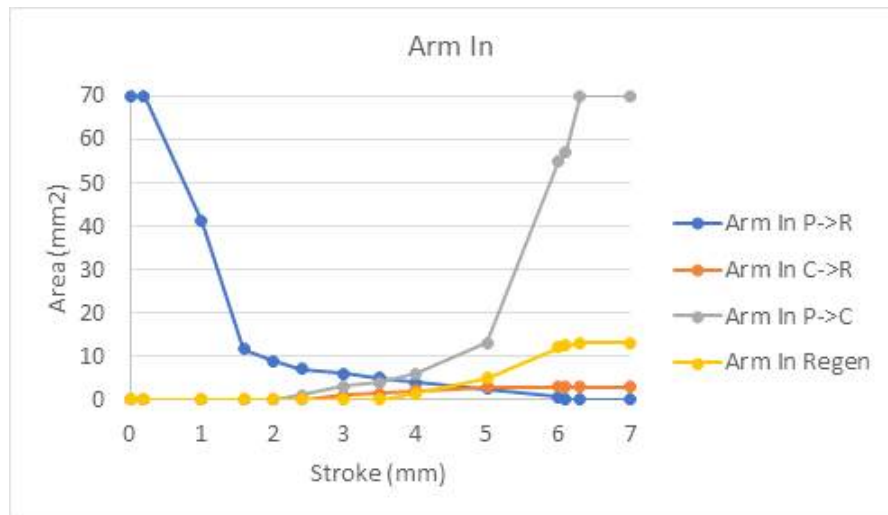


Fig. 24 Opening area diagram of arm valve

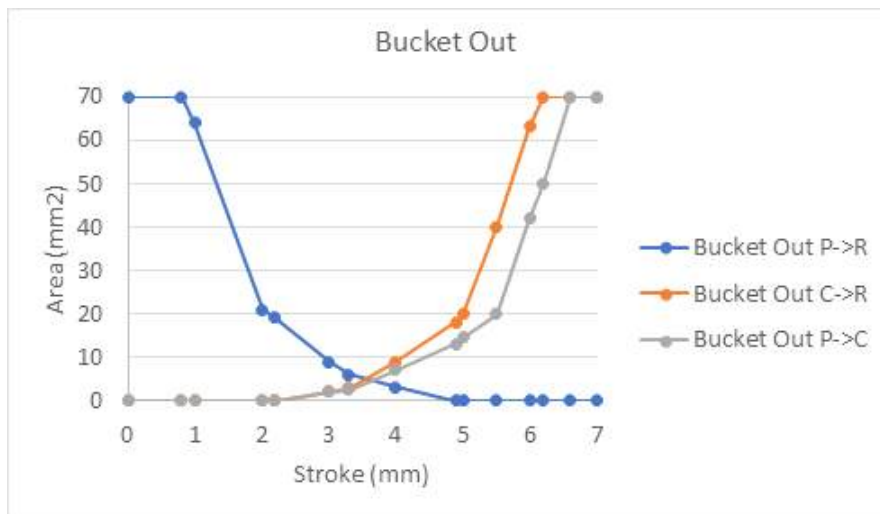
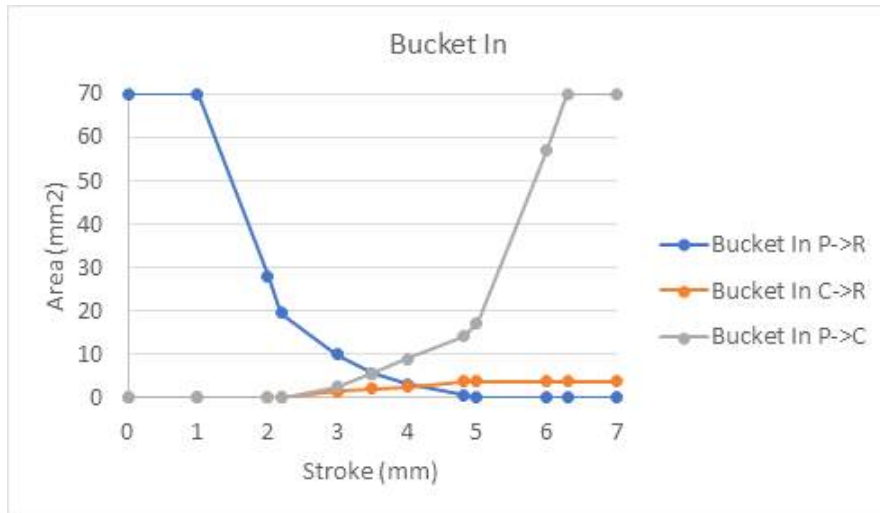


Fig. 25 Opening area diagram of bucket valve



Fig. 26 Opening area diagram of swing valve

Main control valve 모델은 스톱 스트로크에 따른 개구면적 값을 오리피스에 적용하여 모델링하였다. 붐, 암, 버킷, 스윙의 스톱 스트로크에 따른 개구면적 값은 Table 4 ~ 6과 같다. 여기서 ST는 스톱 스트로크 (mm), PR은 P → R 개구면적 (mm²), CR은 C → R 개구면적 (mm²), PC는 P → C 개구면적 (mm²), Re는 Regen 이다. 개구면적선도와 값을 이용하여 각 밸브의 모델과 설정된 값은 Fig. 27 ~ 32와 같다.

Table 4 Opening area value of boom valve

Boom Up				Boom Down			
ST	PR	CR	PC	ST	PR	CR	PC
0	70	0	0	0	70	0	0
0.5	70	0	0	1	70	0	0
1	52	0	0	1.3	70	0	0
2	13	0.5	0	2	41	0	0
2.6	8.5	1.5	0.5	2.4	25.5	0.2	0.2
3	7	2.5	1	3	21	2	1.5
4	5	5	2.5	3.7	13.5	4	3
5	2	16	10	4	11	5	4
6	0	50	39	5	4	8	7
6.4	0	70	57	6	2	9.5	8.5
6.7	0	70	70	7	2	9.5	8.5
7	0	70	70				

Table 5 Opening area value of arm valve

Arm In					Arm Out			
ST	PR	CR	PC	Re	ST	PR	CR	PC
0	70	0	0	0	0	70	0	0
0.2	70	0	0	0	0.5	70	0	0
1	41	0	0	0	1	52	0	0
1.6	11.5	0	0	0	2	11	0	0
2	9	0	0	0	2.2	10	0	0
2.4	7	0	1	0	2.4	9	1	0
3	6	1	3	0	2.8	5.8	4	2
3.5	5	1.5	4	0	3	5.5	5	3
4	4	2	6	1.5	4	3	13	7
5	2.5	2.8	13	5	4.9	2.1	23	12
6	0.5	2.8	55	12	5	2	25	13
6.1	0	2.8	57	12.5	5.5	0	45	34
6.3	0	2.8	70	13	6	0	66	54
7	0	2.8	70	13	6.1	0	70	58
					6.4	0	70	70
					7	0	70	70

Table 6 Opening area value of bucket and swing valve

Bucket In				Bucket Out				Swing			
ST	PR	CR	PC	ST	PR	CR	PC	ST	PR	CR	PC
0	70	0	0	0	70	0	0	0	70	0	0
1	70	0	0	0.8	70	0	0	0.2	70	0	0
2	28	0	0	1	64	0	0	1	38	0	0
2.2	19.5	0	0	2	21	0	0	1.6	13.5	0	0
3	10	1.5	2.5	2.2	19	0	0	1.8	12	0	0
3.5	5.5	2	5.5	3	9	2	2	2	10.5	1	1
4	3	2.5	9	3.3	6	2.8	2.5	3	5.5	4	4
4.8	0.5	3.8	14	4	3	9	7	4	3.5	7	8
5	0	3.8	17	4.9	0	18	13	4.6	2.5	10	13
6	0	3.8	57	5	0	20	14.5	5	2	14	17
6.3	0	3.8	70	5.5	0	40	20	5.8	0.5	21	46
7	0	3.8	70	6	0	63	42	6	0	33	57
				6.2	0	70	50	6.2	0	46.5	70
				6.6	0	70	70	6.9	0	70	70
				7	0	70	70	7	0	70	70

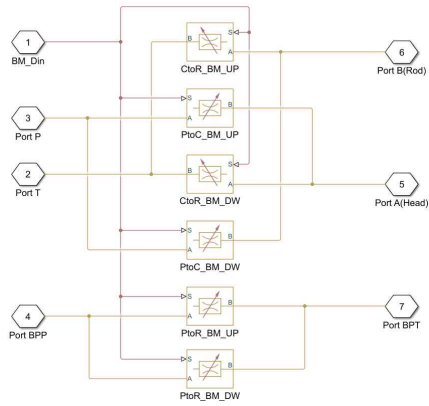


Fig. 27 Boom valve

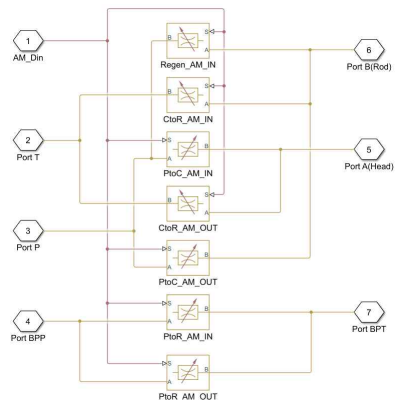


Fig. 28 Arm Valve

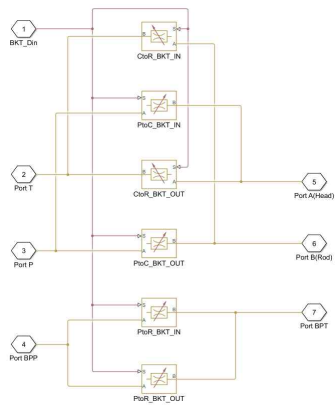


Fig. 29 Bucket valve

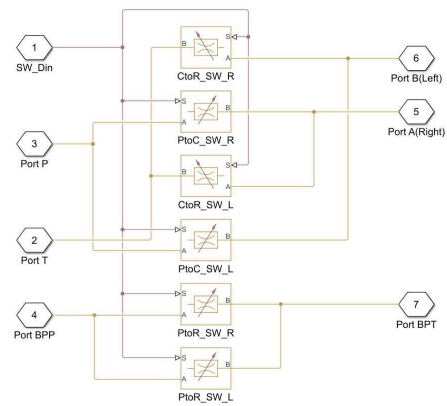


Fig. 30 Swing valve

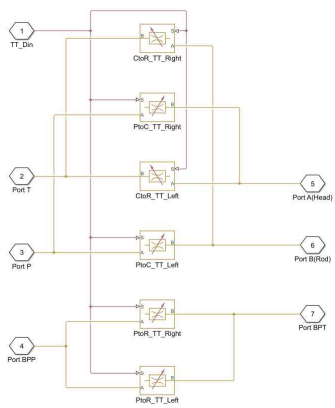


Fig. 31 Tilt valve

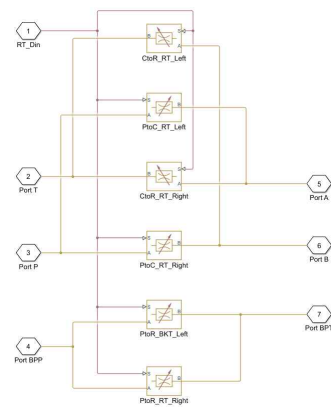


Fig. 32 Rotator valve

1.5톤 필드로봇의 액추에이터는 붐, 암, 버킷의 실린더와 스윙 모터, 틸트로테이터의 액추에이터는 틸트 실린더, 로테이션 모터로 6개의 액추에이터를 구동한다. 실린더는 1.5톤 필드로봇의 제원은 붐, 암, 버킷 실린더의 헤드측과 로드측 면적과 스트로크, 틸트로테이터의 틸트 실린더의 헤드측과 로드측 면적과 스트로크를 적용하였다. 1.5톤 필드로봇의 붐, 암, 버킷 실린더와 틸트로테이터의 틸트 실린더의 제원은 Table 7과 같다.

Table 7 Cylinder specifications of 1.5 ton field robot

		Boom	Arm	Bucket	Tilt
Head	Out Dia [mm]	65	65	65	42
	Inner Dia [mm]	55	55	55	32
Rod	Dia [mm]	30	30	30	14
Stroke [mm]		385	380	280	135
Pressure [bar]		210	210	210	210
Max Flow rate [l/min]		15.8	15.8	15.8	20

1.5톤 필드로봇의 각각 실린더 모델은 MATLAB/Simulink Simscape fluids에서 제공되는 복동실린더를 사용하였다. 실린더 모델은 헤드측, 로드측 면적과 실린더 스트로크, 실린더의 초기위치를 입력하여 모델링하였다. 실린더의 모델링은 각각의 실린더에 동일한 블록을 사용하였으며 실린더의 모델링은 Fig. 33 ~ 36과 같다. 실린더 모델에서 Force 센서와 Velocity 센서를 적용하여 실린더의 Force와 Velocity를 측정할 데이터를 그래프로 확인할 수 있다. 실린더에 적용한 센서는 각각의 실린더에 동일하게 적용하였으며 적용된 센서의 모델링은 Fig. 33 ~ 36과 같다.

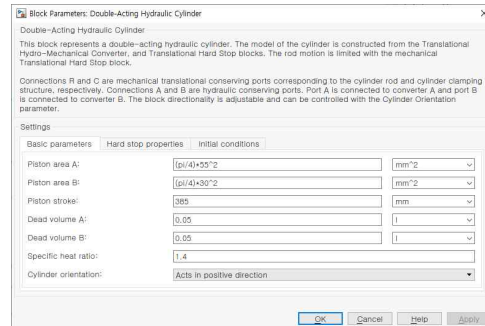
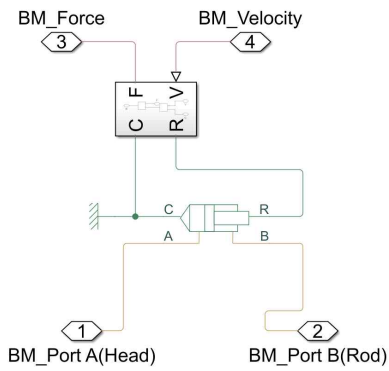


Fig. 33 Boom Cylinder

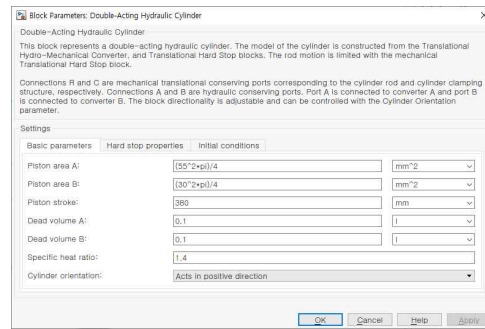
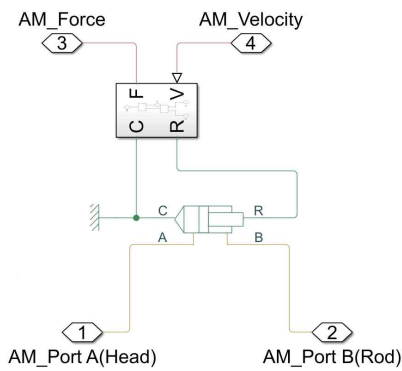


Fig. 34 Arm Cylinder

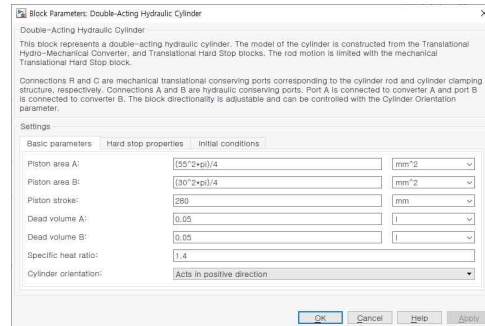
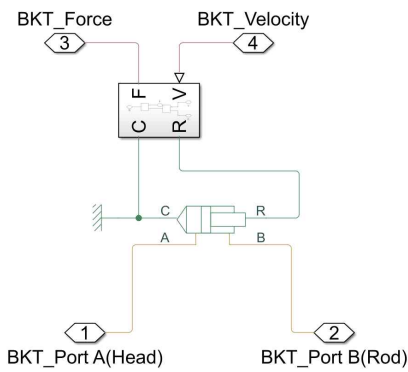


Fig. 35 Bucket Cylinder

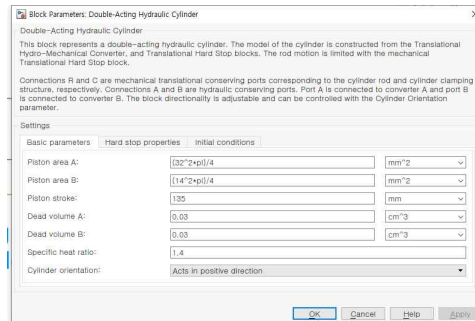
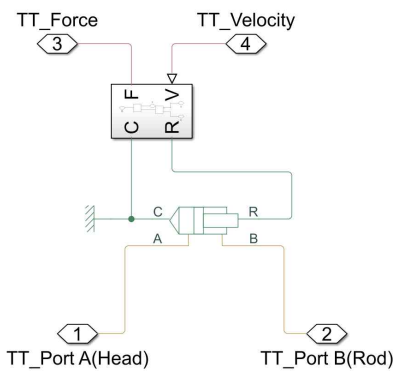


Fig. 36 Tilt Cylinder

1.5톤 틸트로테이터의 스윙모터와 틸트로테이터의 로테이션 모터는 MATLAB/Simulink Simscape fluids에서 제공되는 고정용량형 모터를 사용하였다. 스윙모터의 초기 모델은 유압회로도를 따라 모델링을 하였으나 시뮬레이션 시 스윙동작에 문제가 있음을 확인하였다. 초기모델의 문제점으로 스윙모터의 유압회로도에는 기계적 요소인 브레이크와 Safety valve가 표현되어 있지 않음을 확인하였다. 이에 대해 브레이크를 Friction으로 브레이크를 구현하였고 Safety valve를 구현하였다. 스윙 모터의 모델링은 Fig. 37이고 로테이션 모터의 모델링은 Fig. 38과 같다.

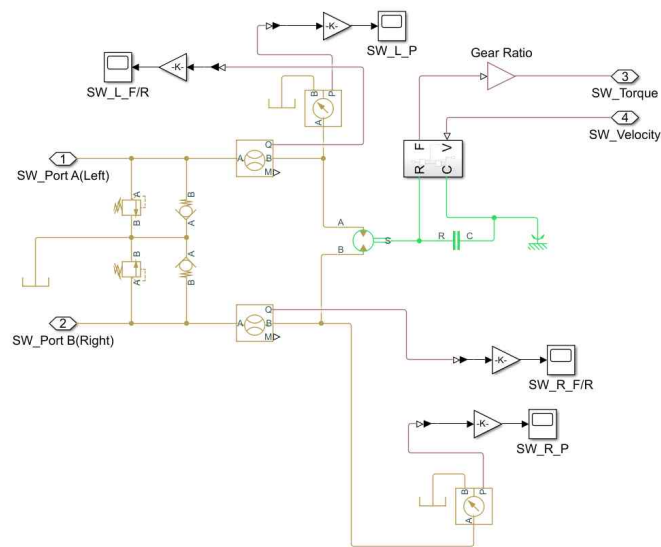


Fig. 37 Swing Motor

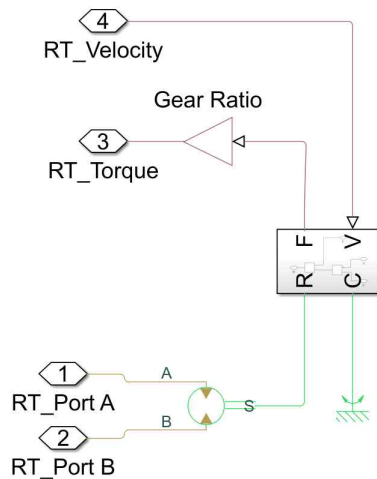


Fig. 38 Rotation Motor

유압모델은 Fig. 39와 같다. 유압모델의 시뮬레이션은 입력된 각도 값을 밸브 스톱 신호 값으로 변환하여 MCV로 신호를 전달한다. 전달받은 신호는 MCV의 오리피스를 작동하여 MCV 방향에 따라 액추에이터로 유량이 전달되어 동작하고 힘과 토크 값을 발생시켜 multibody 모델로 전달한다. 이러한 시뮬레이션을 통해 실린더를 동작하여 실린더의 특성을 확인할 수 있다. Table 2의 붐, 암, 버킷, 틸트 제원의 동작 각도를 입력해 실린더의 단동 동작 시뮬레이션을 수행하여 붐, 암, 버킷, 틸트 실린더의 스트로크와 동작 각도를 Table 2의 동작 제한 각도와 Table 7의 실린더 제원과 비교하였다. 실린더 스트로크 결과로 -3 ~ -5 mm 오차가 있었으나 동작각도 결과에서는 ± 1 deg의 오차가 나타나 동작 수행에는 미미한 영향을 끼칠 것으로 사료된다. 그리고 유압 성능은 유압 회로도에 설정된 릴리프 밸브에 의해 실린더에 적용된 압력 값은 최대 210 bar에서 작동하였다. 확인된 결과 값은 Table 8과 같고 결과 그래프는 Fig. 40 ~ 43과 같다.

Table 8 Comparison of hydraulic models

	Stroke [mm]		
	Spec.	Simulation	Error
Boom	385	380	-5
Arm	380	380	0
Bucket	280	277	-3
Tilt	135	130	-5

	Angle [deg]		
	Spec.	Simulation	Error
Boom	-57 ~ 65	-56 ~ 64	1 ~ -1
Arm	-133 ~ -27	-132 ~ -27	1 ~ 0
Bucket	-108 ~ 38	-109 ~ 38	-1 ~ 0
Tilt	-40 ~ 40	-40 ~ 40	0

	Max. Pressure [bar]			
	Spec.	Simulation		Error
		Head	Rod	
Boom	210	188 (up)	210 (down)	-22
Arm	210	210 (in)	210 (out)	0
Bucket	210	210 (in)	210 (out)	0
Tilt	210	210 (right)	201 (left)	-9

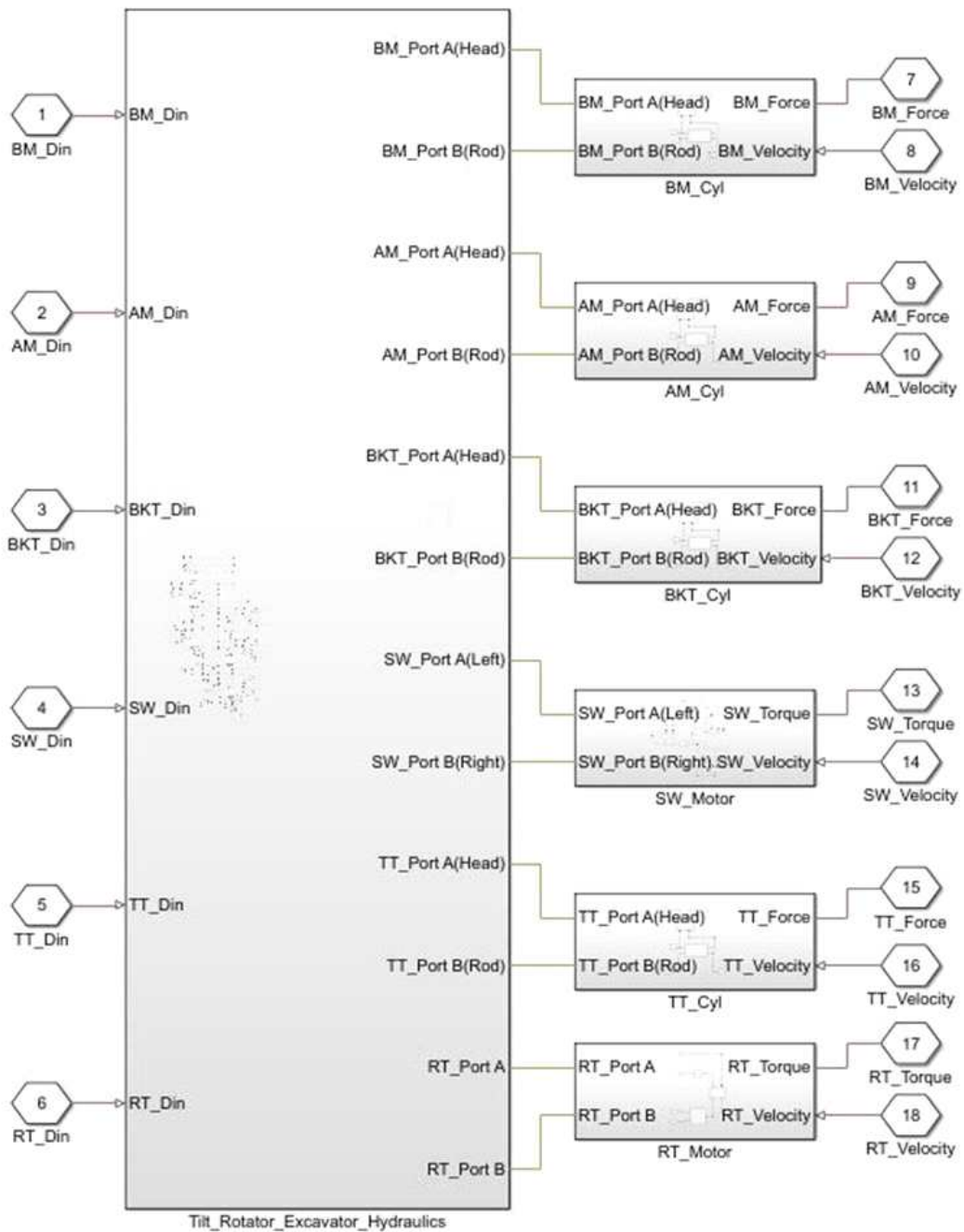


Fig. 39 Hydraulic model of field robot

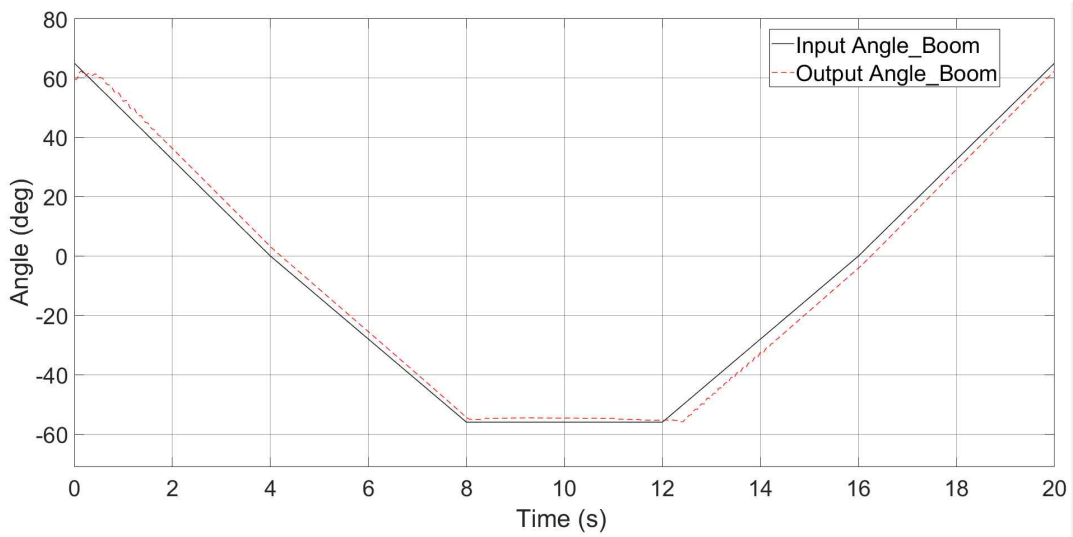
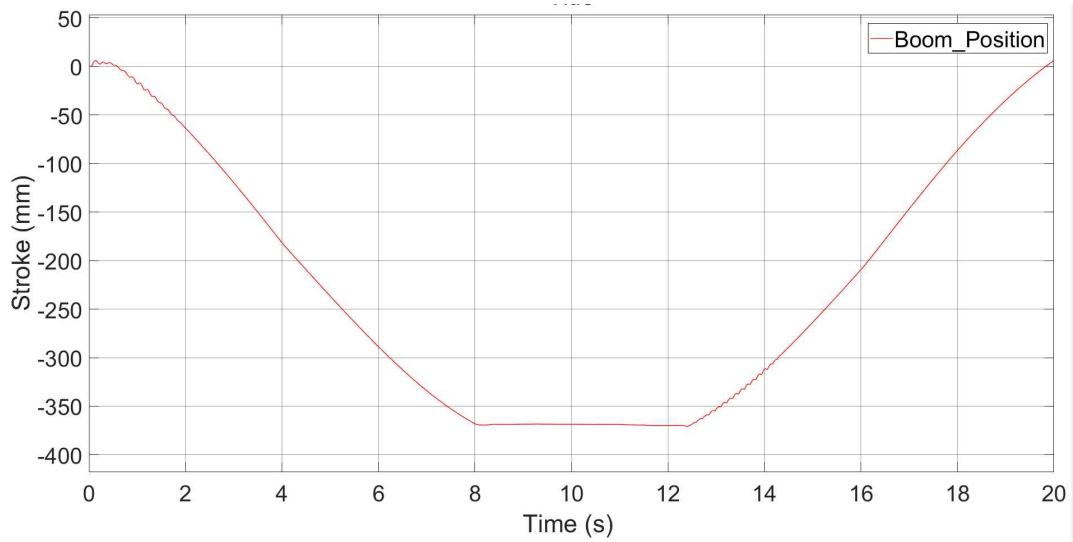


Fig. 40 Simulation results of boom

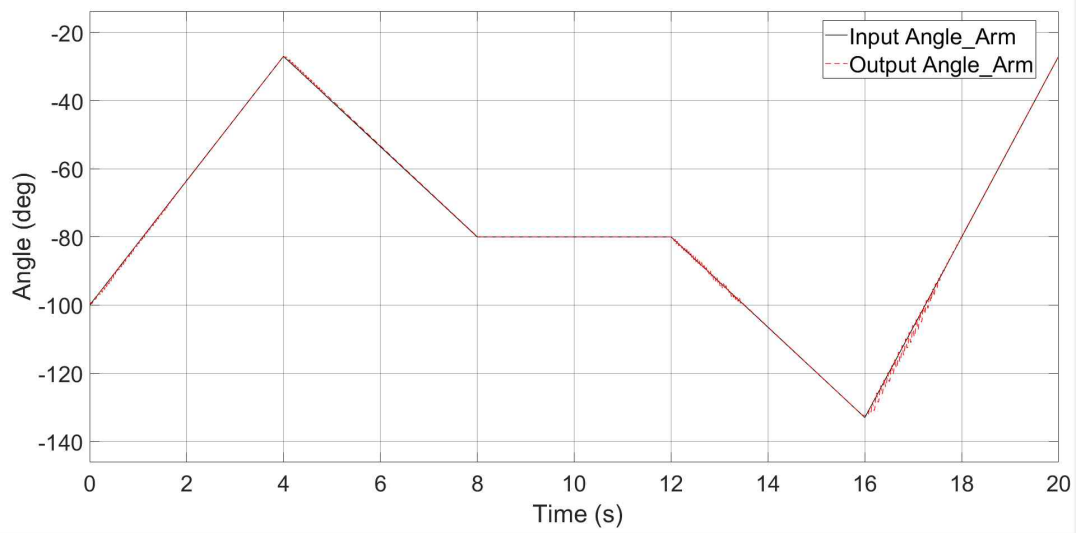
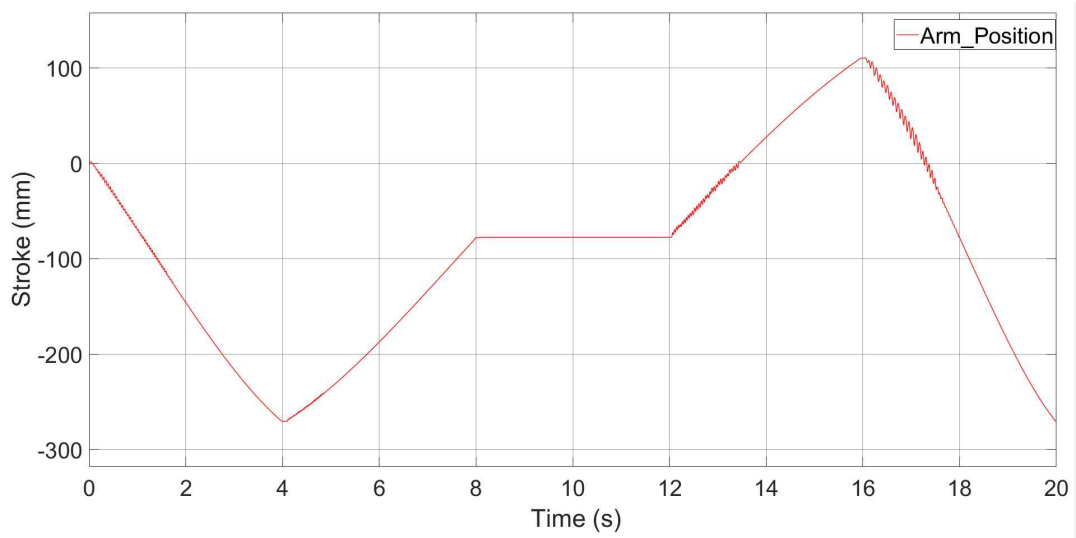


Fig. 41 Simulation results of arm

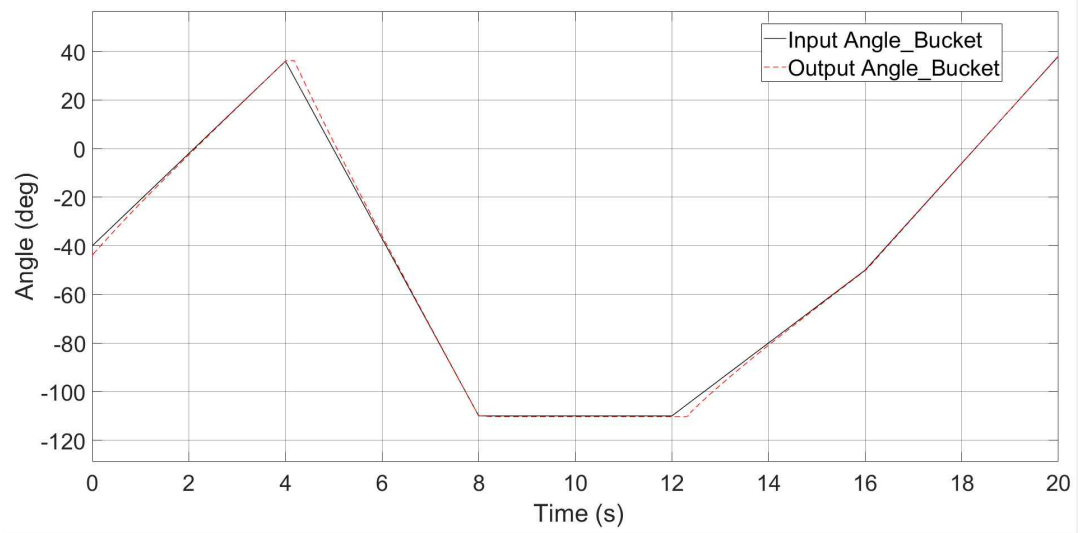
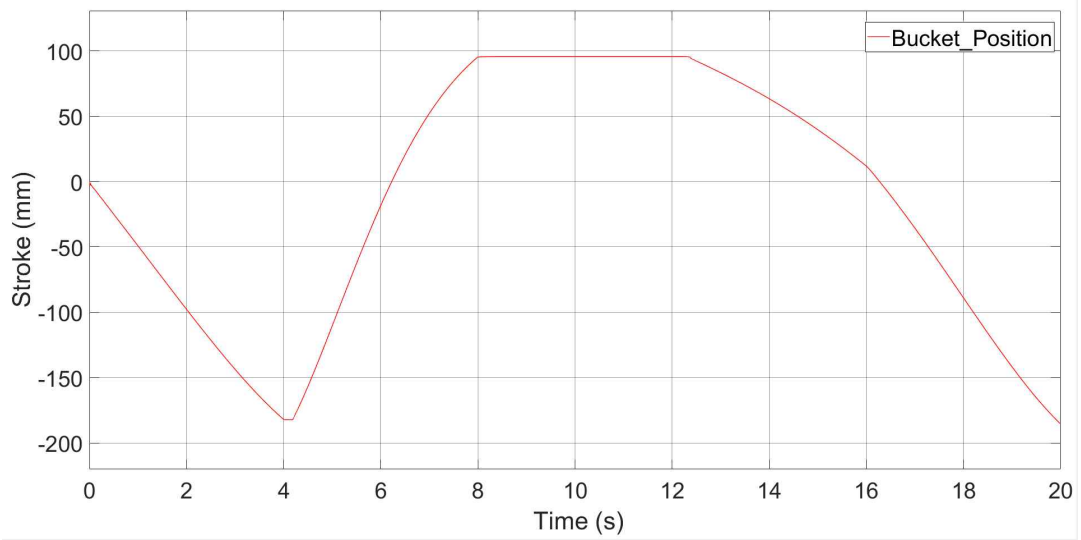


Fig. 42 Simulation results of bucket

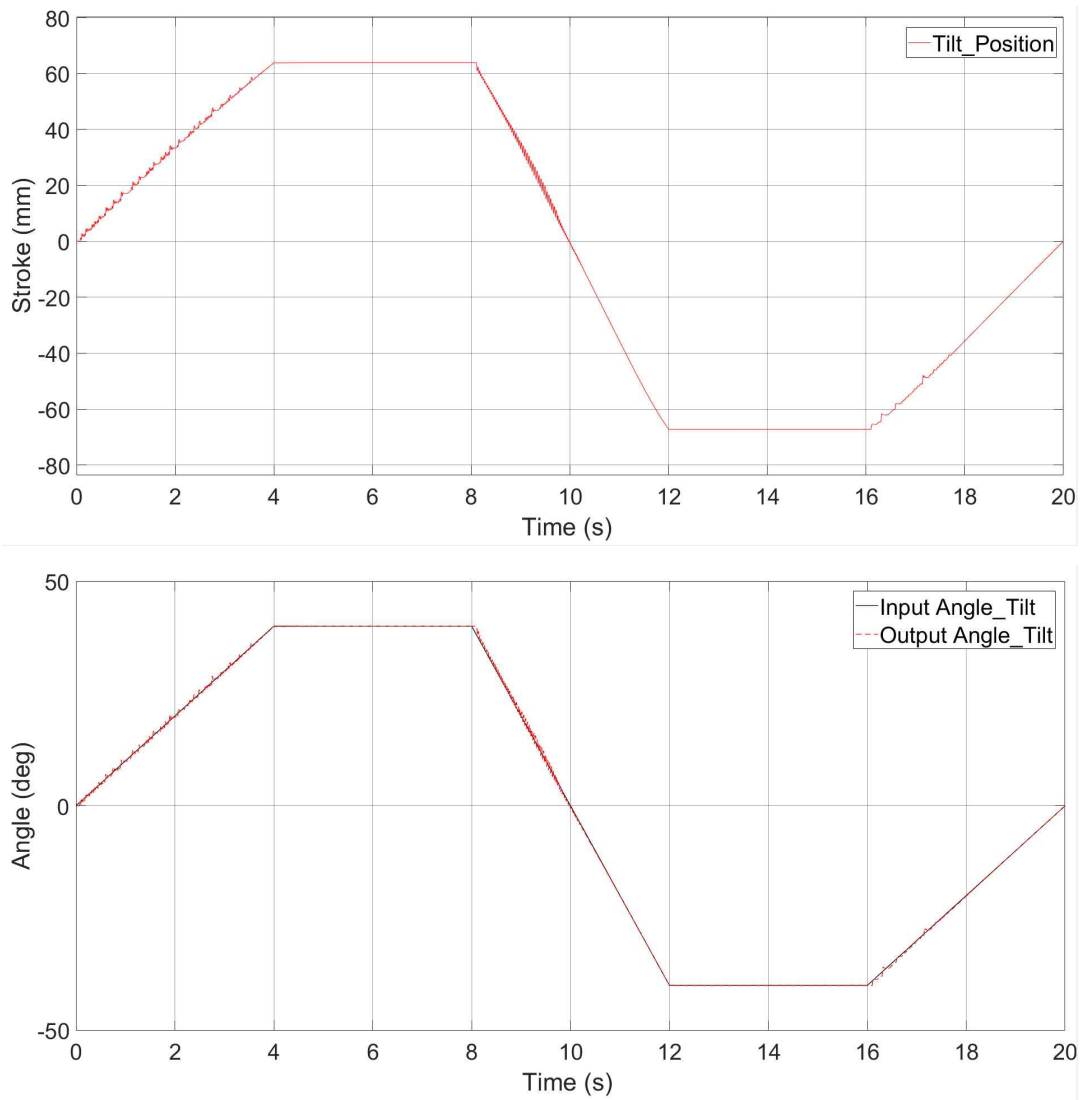


Fig. 43 Simulation results of tilt

4.4 스마트 필드로봇 제어 모델

필드로봇의 유압 시스템은 유압 공급, 전자 제어, 액추에이터 등으로 구분할 수 있다. 이런 유압 시스템은 부하에 따른 유량 압력 곡선의 비선형적인 요소를 포함하고 있어 유압 제어에 어려움이 있다. 이러한 유압 제어의 어려움을 해소하여 향상된 결과를 도출하기 위해 제어기를 사용한다. 유압 시스템의 비선형적 요소를 해결하기 위해 PID 제어기를 사용하였다. 필드로봇의 입력 각도를 이용한 PID 제어기를 MATLAB/Simulink를 이용하여 설계하였다.

PID 제어기는 제어성능이 우수하고, 제어이득 조정이 쉬워 산업현장에서 많이 사용되는 장점과 적용대상이 단입출력 시스템에 제한되어 있는 단점이 있다. PID 제어기는 피드백 구조를 가지는 제어기로 본 논문에서는 제어 대상인 밸브 스톱을 제어하여 필드로봇의 출력 각도를 출력 값으로 도출한다. 출력 각도를 피드백 받아 입력 각도와 비교하여 오차를 계산한다. 여기서 계산된 오차 값을 밸브 스톱 제어에 필요한 값을 계산하는 순서로 구성되어 있다. 필드로봇에서 유압밸브 스톱의 제어를 통해 필드로봇의 제어 동작 블록선도는 Fig. 44와 같다.

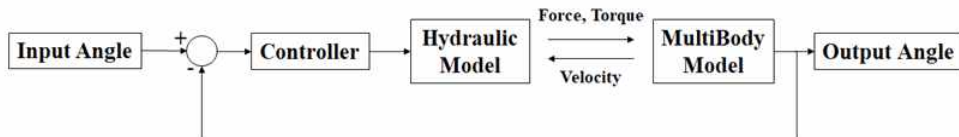


Fig. 44 Block diagram of field robot

PID 제어는 비례항, 적분항, 미분항을 모두 추가하여 k_p 는 비례 게인, k_I 는 적분 게인, k_D 는 미분 게인을 시간영역에서 비례적분미분(PID : proportional integral derivative) 제어를 표현하면 식 55과 같다.

$$u(t) = k_p e(t) + k_I \int_{t_0}^1 e(\tau) d\tau + k_D \dot{e}(t) \quad (55)$$

PID 제어기의 게인 값은 적절한 게인 값을 찾아 조절하여 PID 제어기의 응답시간, 안정성 등의 제어 성능을 만족시킨다. 본 논문에서의 PID 제어기 게인 튜닝은

시행착오(try and error) 방법으로 게인 값을 조정하였다. 시행착오 방법을 통해 조정된 게인 값을 적용하여 Fig. 45와 같이 밸브 스톱을 제어하기 위한 제어를 모델링하였다[54][55].

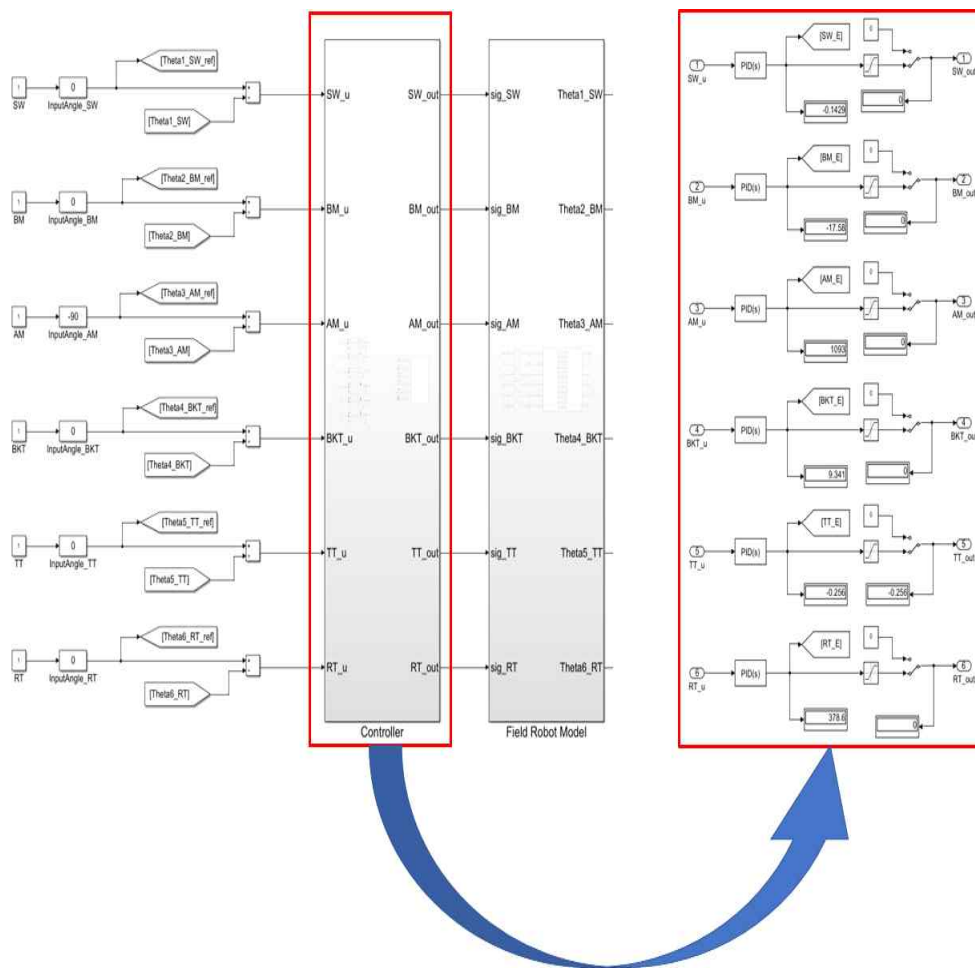


Fig. 45 PID controller model of field robot

5. 시뮬레이션 및 결과

5.1 MATLAB/Simulink Simscape 시뮬레이션

5.1.1 시뮬레이션 모델 및 동작 시뮬레이션 방법

스마트 필드로봇의 기구 모델인 multibody 모델과 유압부 모델인 fluid 모델을 연동하여 Fig. 46과 같이 최종 시뮬레이션 모델을 구축하였다. 최종 필드로봇 모델은 각 조인트를 조종하기 위한 입력신호는 PID 컨트롤러를 통해 밸브 스펴 변위를 제어하는 신호로 변환하여 유압 모델로 밸브 스펴 변위 신호를 전달한다. 입력된 밸브 스펴 변위 신호는 밸브 스펴 변위 제어에 따라 각 조인트의 실린더 또는 모터를 동작하게 된다. 필드로봇의 제어 흐름에 따라 동작된 결과는 Fig. 47과 같이 시뮬레이션 GUI로 동작을 확인할 수 있다.

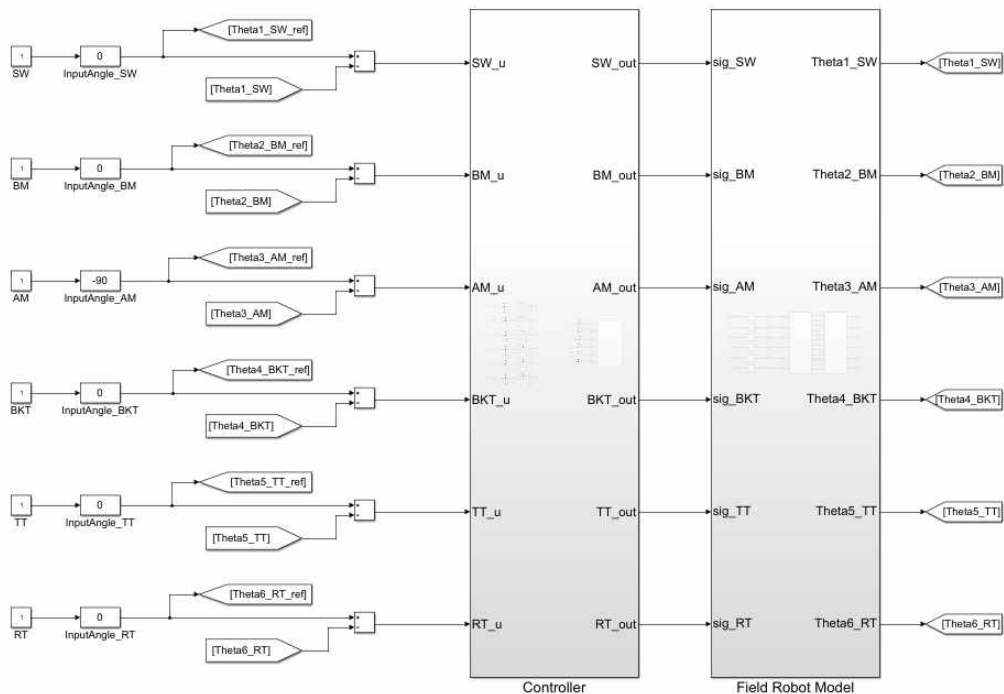


Fig. 46 Simulation model of field robot using MATLAB/Simulink simscape

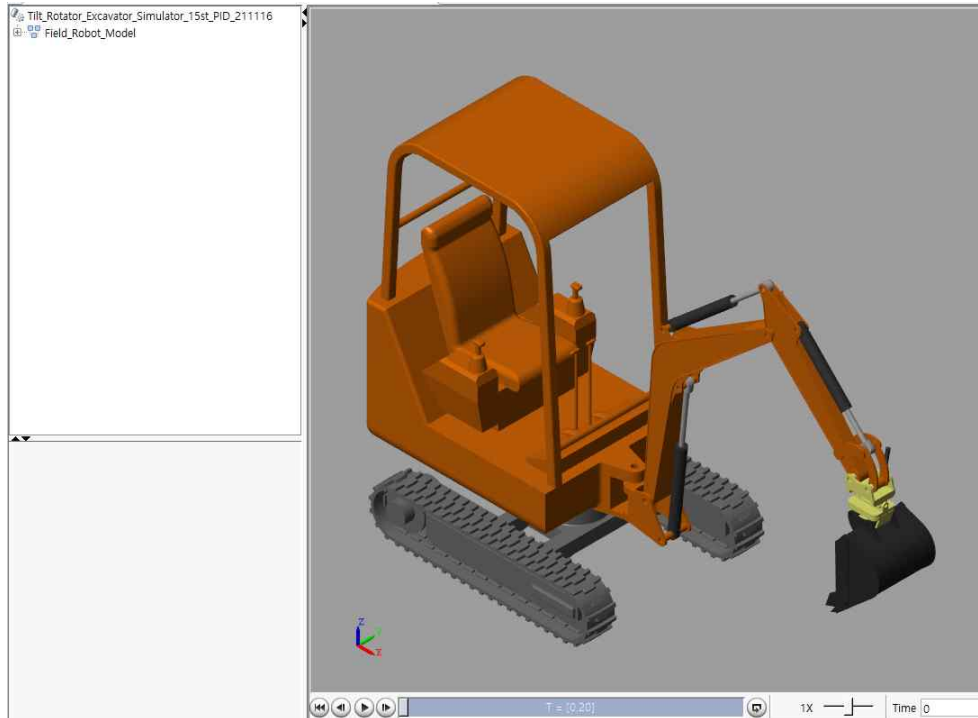


Fig. 47 Simulation GUI of field robot

구축된 스마트 필드로봇의 시뮬레이션 모델을 이용하여 동작 시뮬레이션을 진행하였다. 첫 번째로 구축된 모델의 입력신호 값을 sine 파형으로 입력하여 붐, 암, 버킷, 틸트로테이터의 단독 동작을 통해 시뮬레이션을 진행하였다. 두 번째로 굴착작업 시나리오에 대한 붐, 암, 버킷의 단독 동작과 틸트로테이터의 단독 동작 시뮬레이션을 진행하기 위해 Simulink의 Signal builder를 이용하여 입력 각도를 Fig. 48 ~ 52와 같이 설정하였다. 세 번째로 굴착작업 시나리오에 대해 붐, 암, 버킷의 복합 동작 시뮬레이션으로 굴착 작업을 모사하였다. 여기서, 굴착 작업 모사에 대한 단독 동작과 복합 동작 시뮬레이션을 수행하기 위해 설정한 붐, 암, 버킷, 틸트로테이터의 입력 각도는 Fig. 47 ~ 49와 같다.

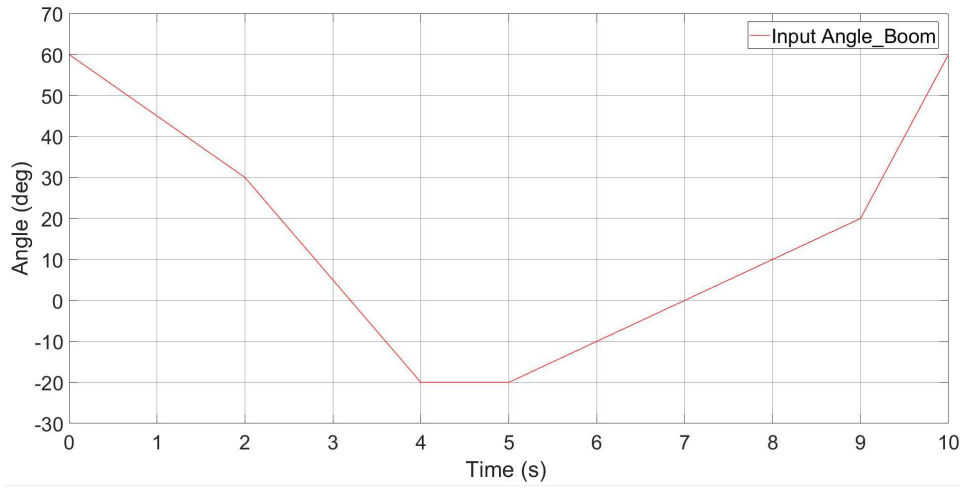


Fig. 48 Setting angle of the excavation operation scenario of the boom

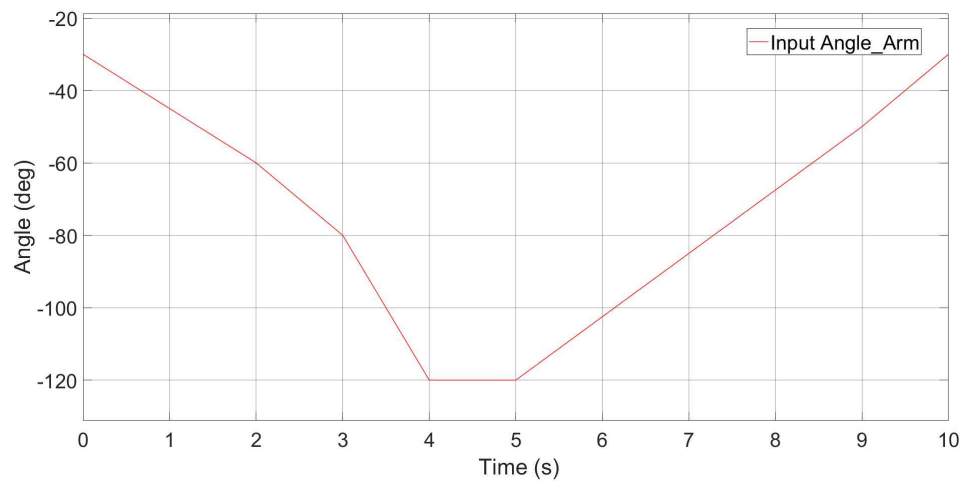


Fig. 49 Setting angle of the excavation operation scenario of the arm

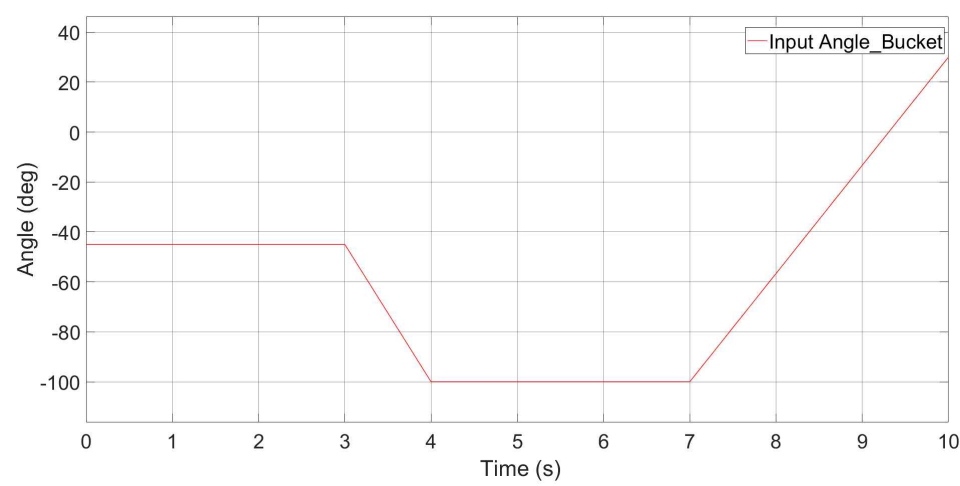


Fig. 50 Setting angle of the excavation operation scenario of the bucket

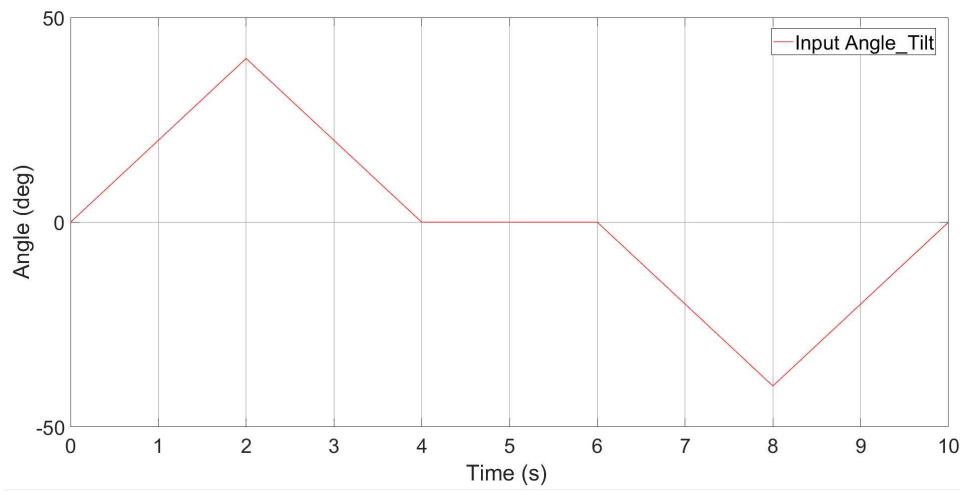


Fig. 51 Setting angle of the excavation operation scenario of the Tilt

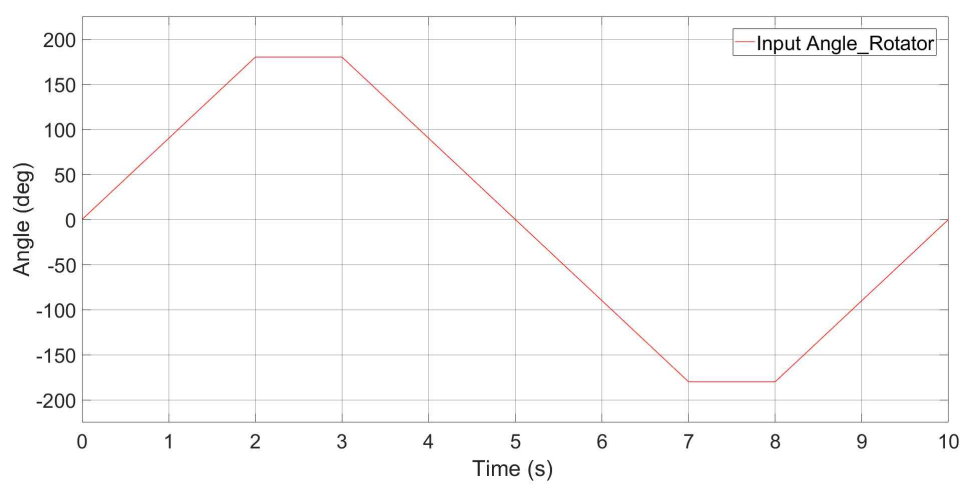


Fig. 52 Setting angle of the excavation operation scenario of the rotator

5.1.2 시뮬레이션 결과

첫 번째 시뮬레이션에 대한 결과는 필드로봇의 각 조인트의 동작 제한 각도 범위 내의 sine 파형의 입력을 통해 밸브 스펴 변위를 제어하여 동작하는 필드로봇의 붐, 암, 버켓, 틸트로테이터의 출력각도를 입력각도와 비교하였다. 입력각도와 출력각도를 비교한 결과는 Fig. 53 ~ 57과 같다. 결과 그래프에서 보이는 바와 같이 일부 구간에서 0.3 ~ 0.5 sec의 차이를 보였지만 입력된 각도에 대해 출력된 각도의 경향이 유사함을 확인할 수 있다.

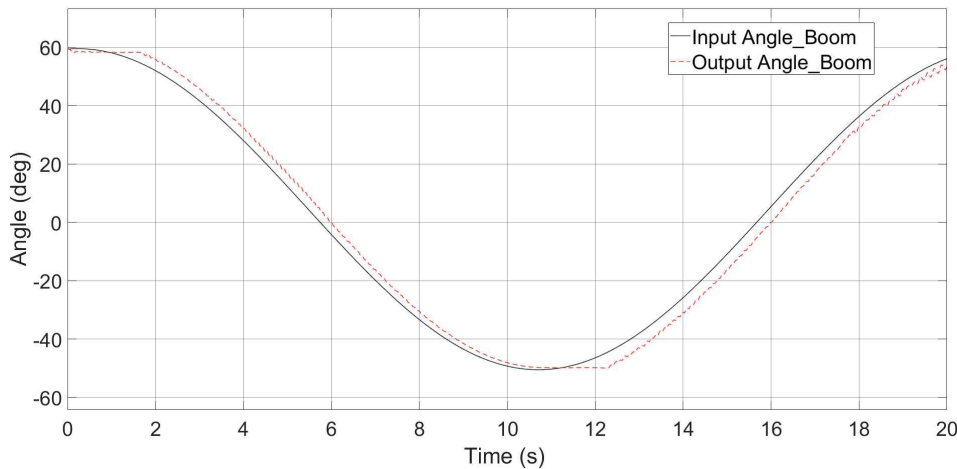


Fig. 53 Result of input and output angle using sine wave (boom)

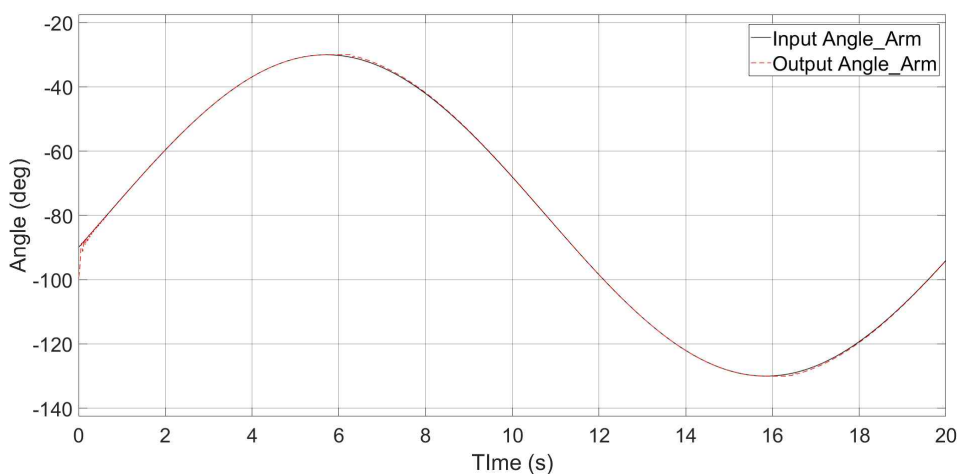


Fig. 54 Result of input and output angle using sine wave (arm)

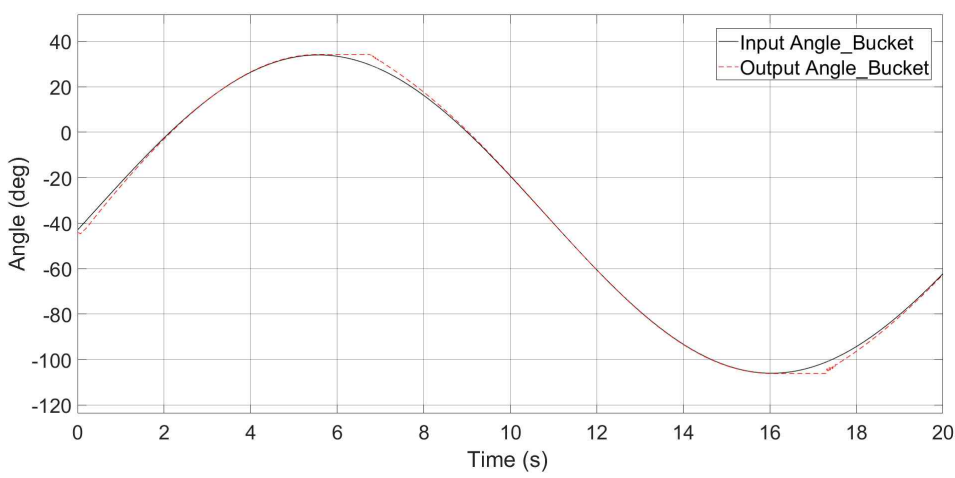


Fig. 55 Result of input and output angle using sine wave (bucket)

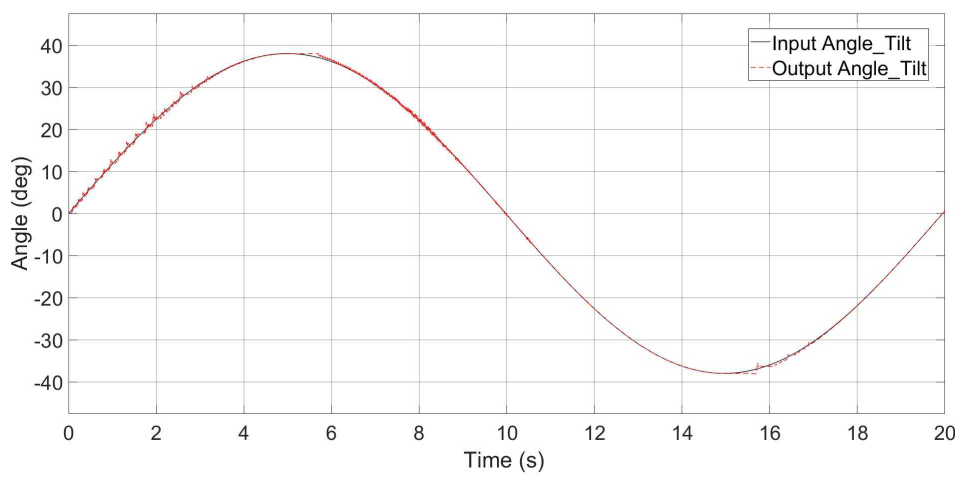


Fig. 56 Result of input and output angle using sine wave (tilt)

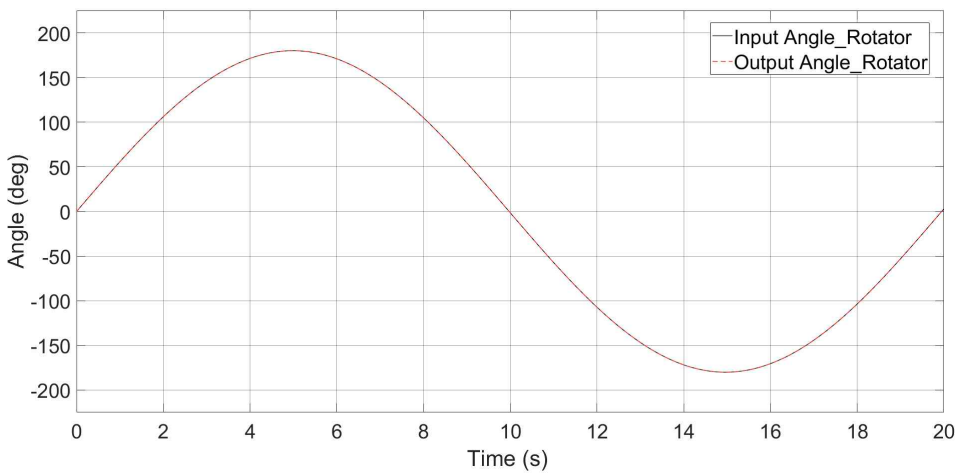


Fig. 57 Result of input and output angle using sine wave (rotator)

두 번째 시뮬레이션인 굴착 작업 시나리오에 따른 붐, 암, 버켓, 틸트로테이터의 단독 동작에 대한 시뮬레이션 결과이다. 앞서 설명한 Simulink의 Signal builder를 이용하여 필드로봇 동작 범위 내에서 굴착 동작을 모사하는 각 조인트의 각도를 입력하여 입력각도와 출력각도를 비교하였으며 설정된 릴리프 압력 210 bar내에서 작동을 하는지 확인하였다. 동작에 대해 입력각도와 출력각도를 비교한 결과와 설정된 릴리프 압력에 대한 동작 결과 값은 Table 9와 같으며 붐, 암, 버켓, 틸트로테이터의 동작 수행 모습과 결과 그래프는 Fig. 58 ~ 62와 같다. 결과 그래프에서 X축은 시뮬레이션 시간, Y축은 각 조인트의 각도이다. 결과 그래프와 같이 단독 동작에 대해서는 설정된 릴리프 압력 210 bar 내에서 동작하는 것을 확인할 수 있었으며 몇몇 부분에서 발생하는 차이를 제외하면 전반적으로 입력각도에 대해 출력각도의 경향이 유사하게 동작하는 것을 확인할 수 있다.

Table 9 Simulation results of single operation

	Angle [deg]		Max. Pressure [bar]	
	Input	Output	Head	Rod
Boom	-20 ~ 60	-18 ~ 59	158 (up)	210 (down)
Arm	-120 ~ -30	-119 ~ -30	210 (in)	210 (out)
Bucket	-100 ~ 30	-100 ~ 29	105 (in)	210 (out)
Tilt	-40 ~ 40	-40 ~ 40	200 (right)	210 (left)
Rotator	-180 ~ 180	-180 ~ 180	186 (right)	185 (left)

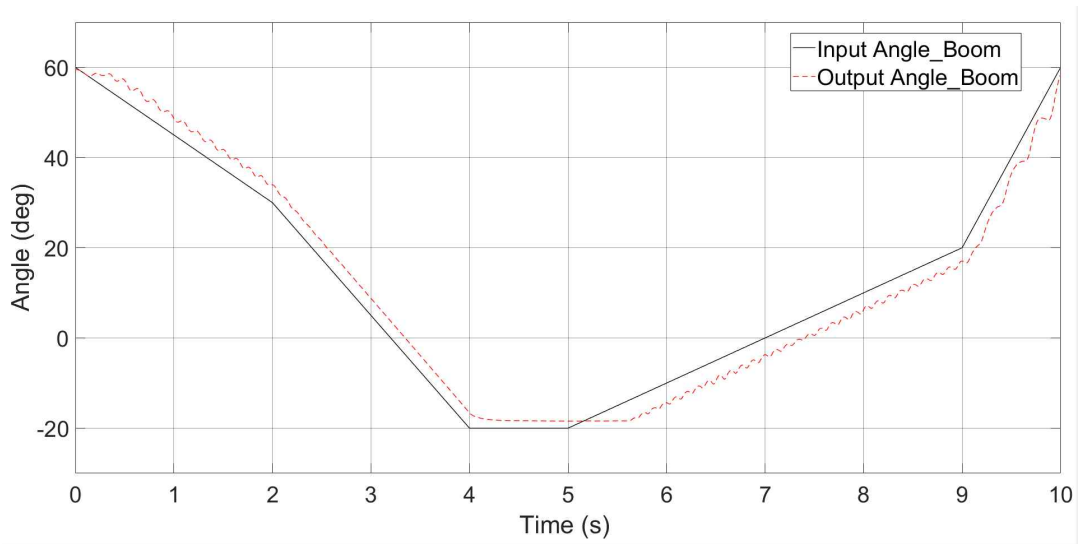
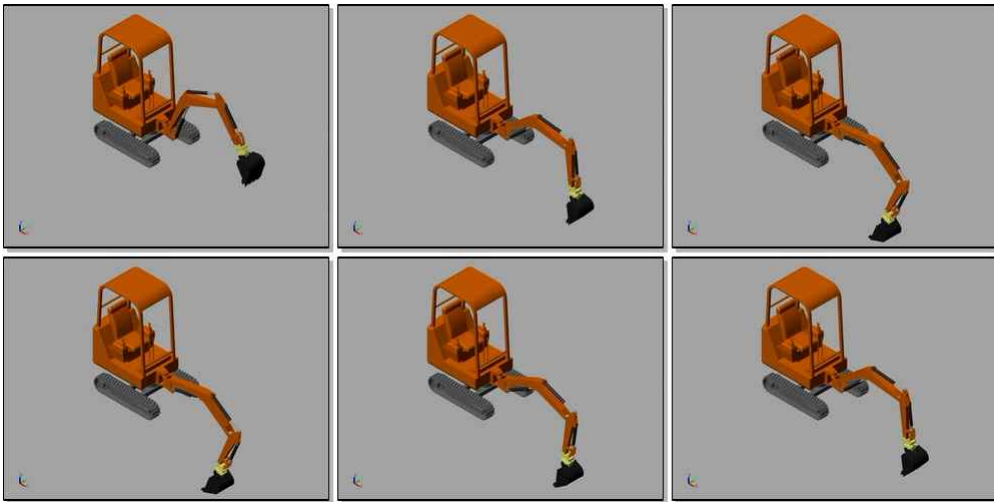


Fig. 58 Simulation results of single operation (boom)

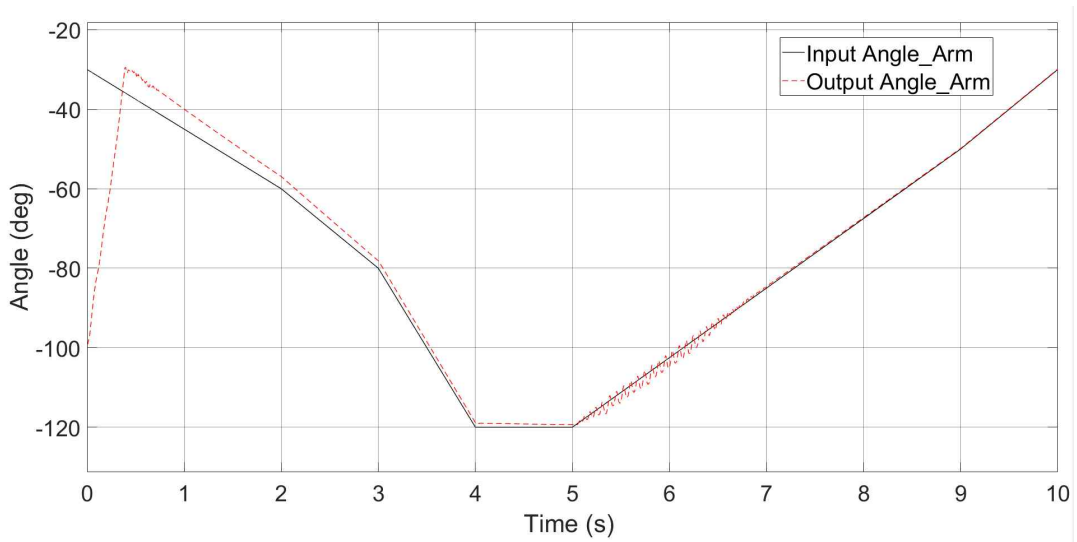
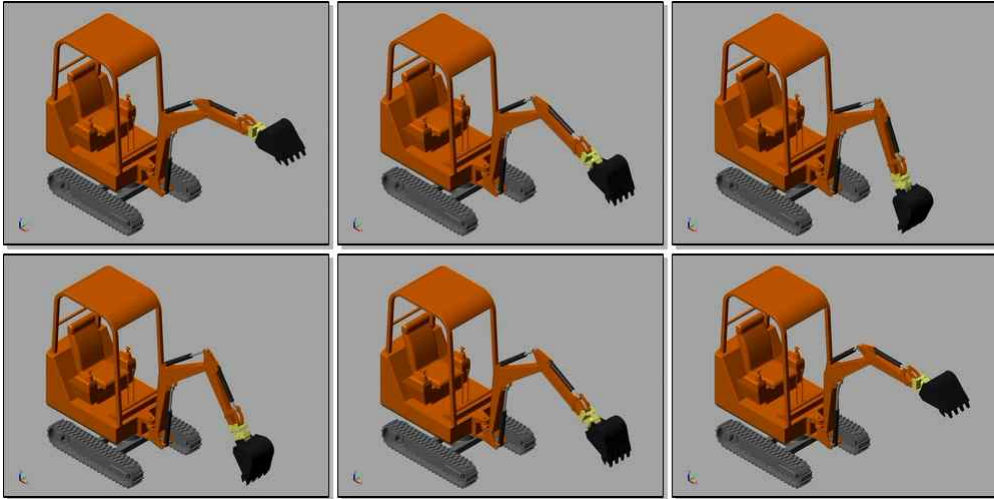


Fig. 59 Simulation results of single operation (arm)

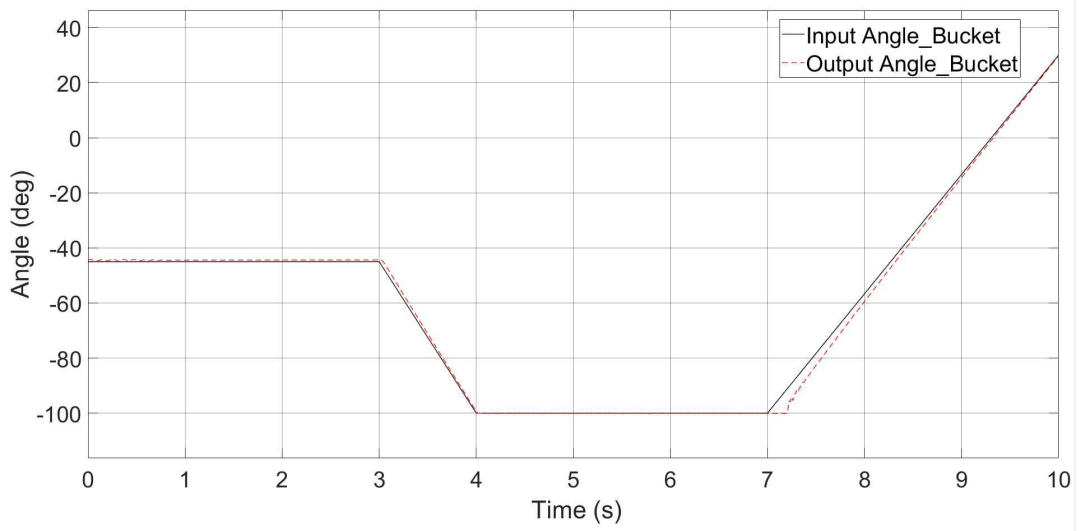
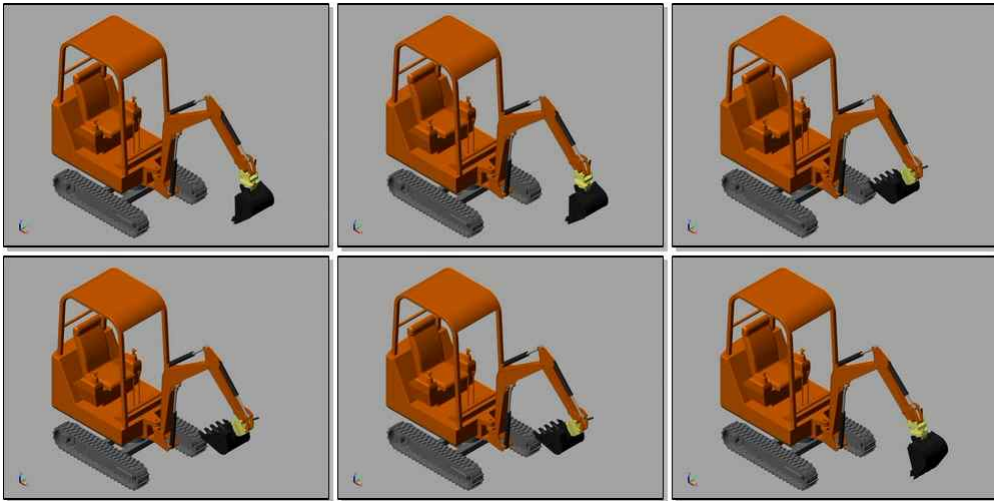


Fig. 60 Simulation results of single operation (bucket)

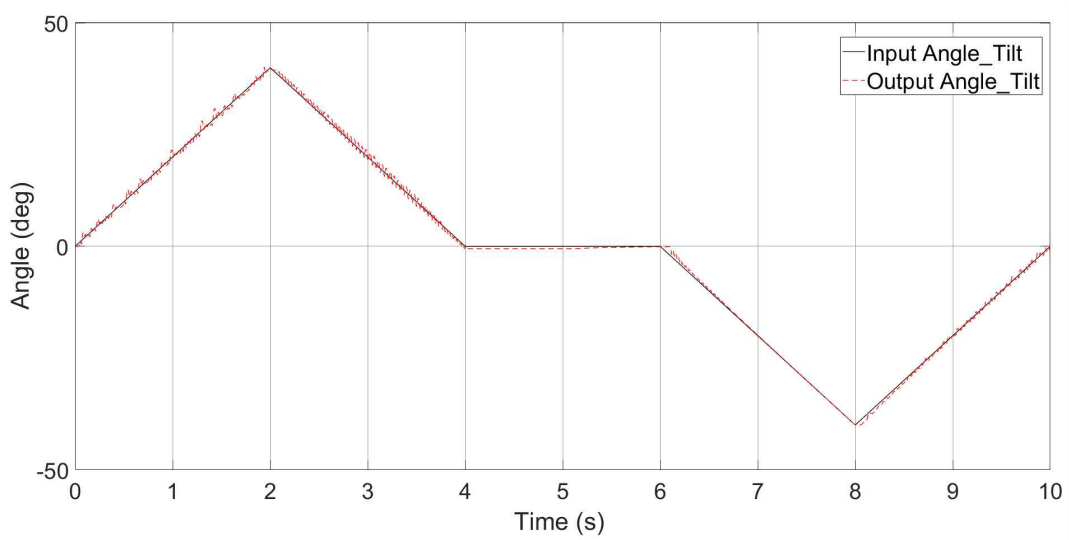
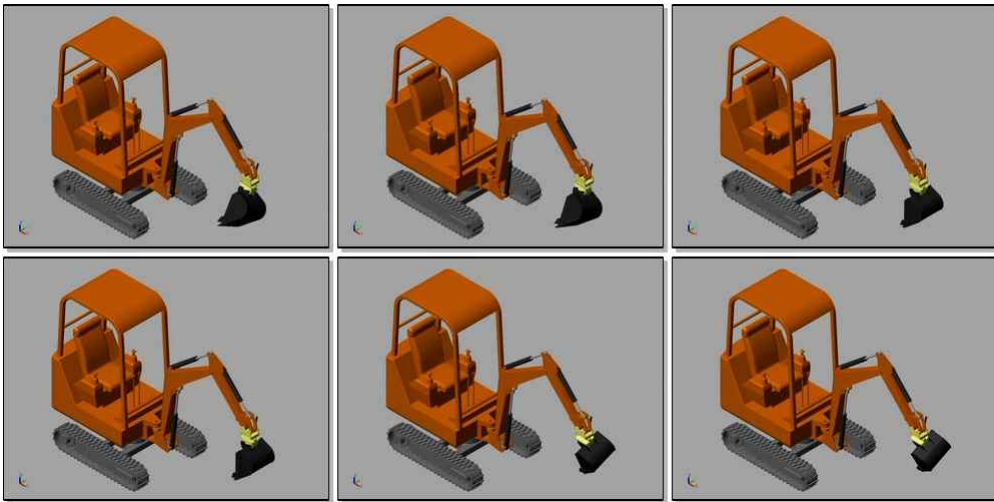


Fig. 61 Simulation results of single operation (tilt)

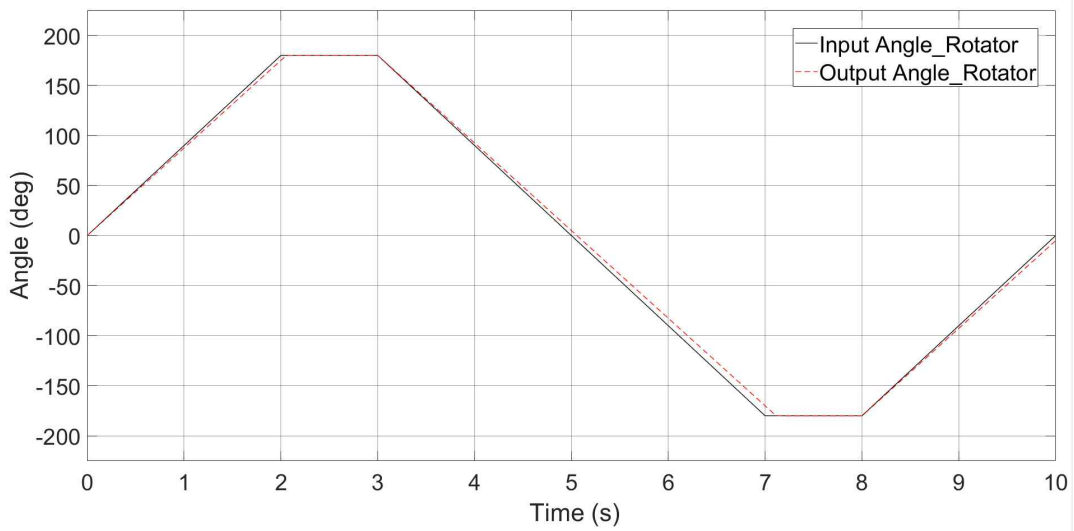
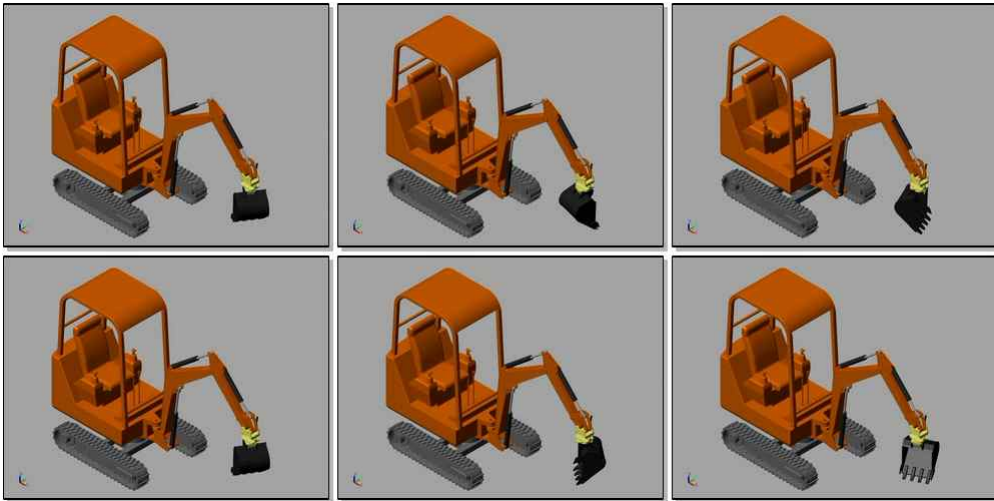


Fig. 62 Simulation results of single operation (rotator)

세 번째 굴착 작업 시나리오에 대한 붐, 암, 버킷이 복합 동작하는 시뮬레이션의 결과이다. 단독 동작과 앞서 설명한 Simulink의 Signal builder를 이용하여 필드로봇 동작 범위 내에서 굴착 동작을 모사하는 붐, 암, 버킷의 각도를 입력하여 입력각도와 출력각도를 비교하고 설정된 릴리프 압력인 210 bar 내에서 동작하는지 확인하였다. 동작에 대해 입력각도와 출력각도를 비교한 결과와 설정된 릴리프 압력에서 동작한 결과 값은 Table 10과 같으며 붐, 암, 버킷의 복합 동작 수행 모습과 결과 그래프는 Fig. 63 ~ 65과 같다. 단독 동작과 복합 동작에서도 전반적으로 필드로봇의 동작 제한 각도 범위와 설정된 릴리프 압력 210 bar 내에서 입력된 각도를 따라 최대 압력으로 각 조인트의 출력 각도 값을 보였으며 입력각도와 출력각도의 진행 경향이 유사함을 확인할 수 있다.

Table 10 Simulation results of compound operation

	Angle [deg]		Max. Pressure [bar]	
	Input	Output	Head	Rod
Boom	-20 ~ 60	-19 ~ 60	210 (up)	210 (down)
Arm	-120 ~ -30	-120 ~ -29	210 (in)	210 (out)
Bucket	-100 ~ 30	-110 ~ 30	191 (in)	210 (out)

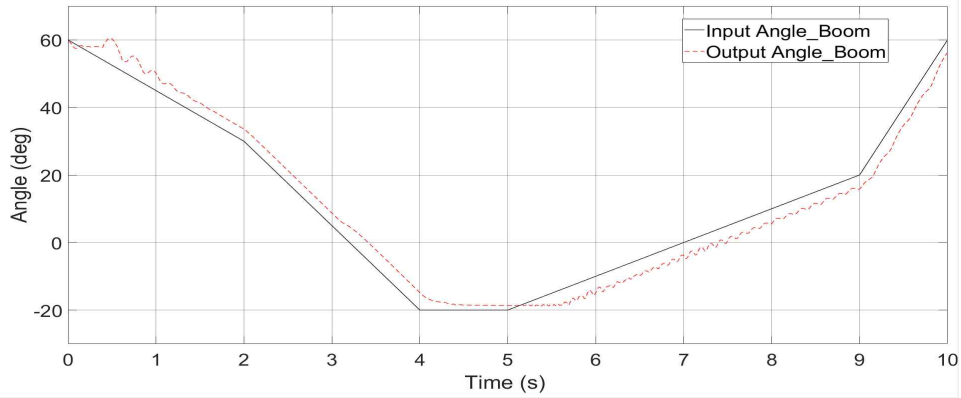
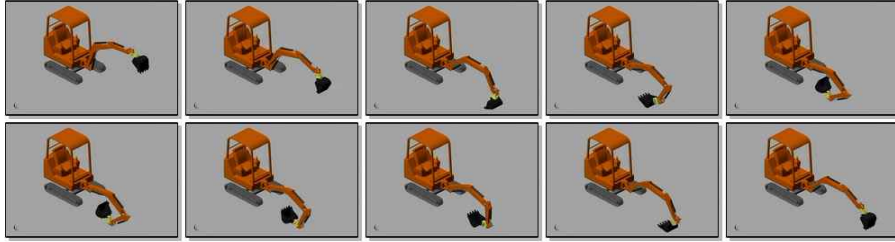


Fig. 63 Simulation results of compound operation (boom)

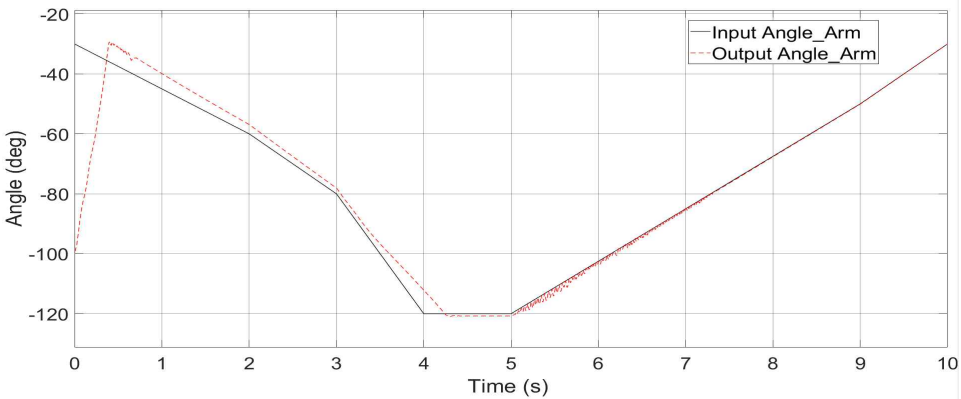


Fig. 64 Simulation results of compound operation (arm)

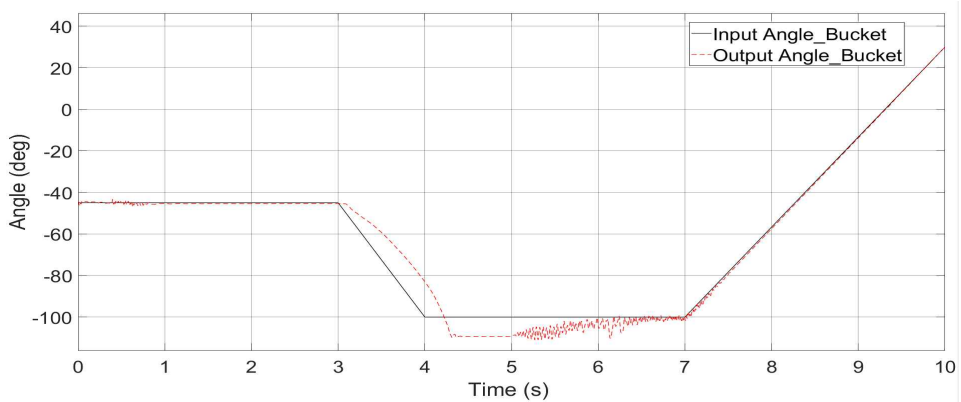


Fig. 65 Simulation results of compound operation (bucket)

시뮬레이션의 결과에 대해 전반적으로 시뮬레이션 모델의 동작이 입력 각도에 대해 출력 각도의 경향이 모두 유사하게 동작하는 것을 확인할 수 있다. 시뮬레이션 결과에서 발생한 몇 가지 차이는 단독 동작과 복합 동작에서 처음 0 ~ 0.5초 동안 압의 결과가 차이가 있음을 보이는데 초기 자세에서 입력된 각도로 동작하면서 발생한 차이이다. 결과 그래프에서 발생하는 오차의 경우 밸브의 방향 전환에 따라 최대 유량을 토출하는 고정형 펌프에서 발생하는 불안정한 영역에서의 속도변화에서 발생하는 충격으로 인해 생긴 진동과 오차인 것으로 사료된다.

5.2 심층강화학습 시뮬레이션

5.2.1 제안 모델

스마트 필드로봇의 심층강화학습 모델은 크게 학습 대상의 상태를 가상으로 이루어진 환경 모델과 학습하고자 하는 경로와 정책 알고리즘인 에이전트 모델과 에이전트 모델에서 학습을 통해 환경 모델을 동작하는 행동 값, 환경 모델의 학습된 동작에서 측정되는 관측 값, 학습된 행동에 대한 보상 값으로 5가지로 구성 된다.

본 논문에서 제안하는 스마트 필드로봇의 작업경로 학습 모델은 Fig. 66과 같다. 심층강화학습 모델에서 스마트 필드로봇 모델을 가상으로 이루어진 환경 모델을 구성하였으며 스마트 필드로봇의 버킷 끝단 위치에 대한 작업 경로와 작업 경로 학습 알고리즘은 학습 수행함에 있어 행동에 대한 보상 값을 피드백 받아 이전 정책에서 새로운 정책이 업데이트 되는 PPO 알고리즘으로 에이전트 모델을 구성하였다. 작업 경로 학습 수행을 통해 행동 값으로는 x, y, z 위치 값, 환경 모델을 통해 확인할 수 있는 관측 값은 버킷 끝단의 위치와 붐, 암, 버킷의 조인트 토크 값, 환경 모델에 입력으로 들어가는 행동 값과 환경 모델을 통해 피드백 되는 관측 값의 오차 값에 따라 에이전트 모델에 보상 값을 피드백해주는 구성으로 스마트 필드로봇 작업경로 학습 모델을 구성하였다. 여기서, 오차 값은 버킷 끝단의 x, y, z 축 위치의 오차이다.

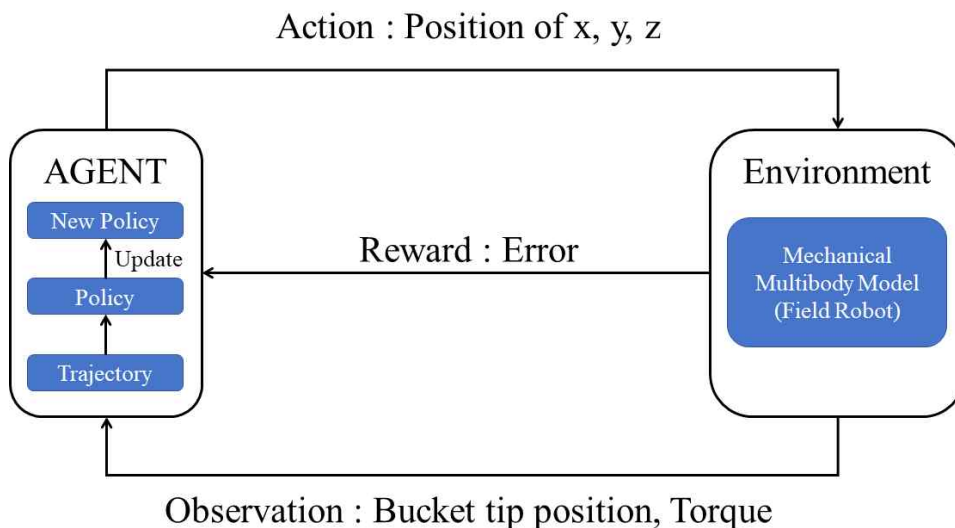


Fig. 66 Proposed deep reinforcement learning model

5.2.2 학습 모델

스마트 필드로봇의 작업 경로 학습을 위해 심층강화학습법을 이용하여 학습 모델링을 하였다. 본 논문에서의 스마트 필드로봇 시뮬레이션 모델의 3D 모델을 이용하여 스마트 필드로봇 작업 경로 학습 모델 중 기구학적 시스템을 작업경로 학습 모델로 이용하였다. 모델링된 3D 모델을 강화학습 모델에 적용하기 위한 모델을 URDF(Unified Robot Description Format) 모델이라 한다. URDF 모델은 필드로봇 모델을 Fig. 67과 같이 Solidworks의 URDF 변환 기능을 이용하였다. Fig. 67과 같이 Solidworks URDF Exporter는 필드로봇의 각 링크 시스템과 조인트의 좌표계를 구성하는데 사용된다. Fig. 68과 같이 플러그인을 통해 URDF 모델과 물리 매개변수, 외부 그래픽을 완성하는 메쉬 파일을 생성한다[56].

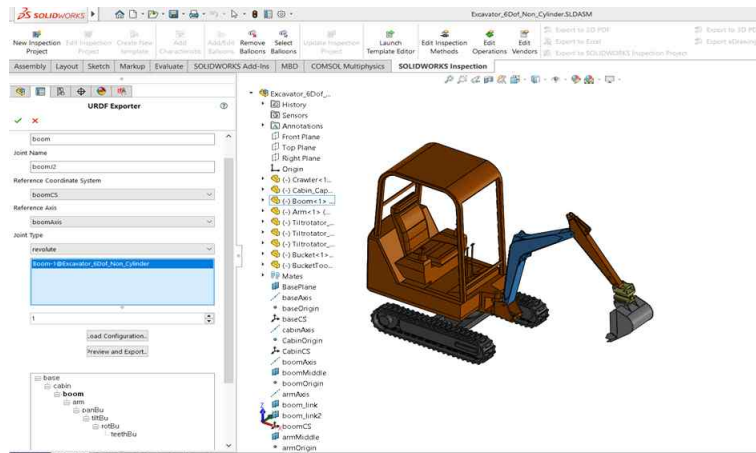


Fig. 67 Field robot URDF model using Solidworks URDF exporter

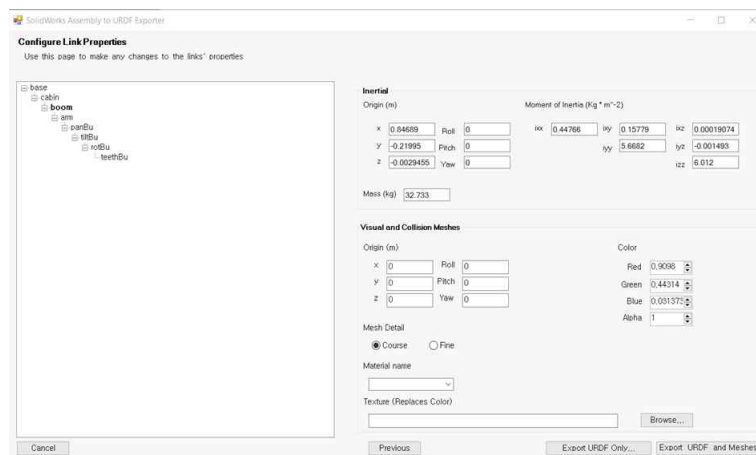


Fig. 68 URDF model properties of field robots

생성된 스마트 필드로봇의 URDF 모델을 강화학습 시뮬레이션 물리 엔진인 pybullet에 가져왔다. 강화학습에 사용한 시뮬레이션 물리 엔진인 PyBullet은 로봇 시뮬레이션과 기계학습을 위해 사용하기 쉬운 Python 모듈이다. PyBullet을 사용해서 URDF 관절 모델을 로드할 수 있다. PyBullet 엔진에서 URDF 모델을 가져오기 위해 ‘loadURDF(“필드로봇 모델”)’ 코드를 사용한다. 필드로봇의 각 조인트의 위치, 속도, 토크로 구성된 ‘setJointMotorControl2’ 기능을 통해 필드로봇의 URDF 모델을 작동한다. 필드로봇의 URDF 모델은 경로 학습을 하게 되면서 링크와 조인트의 상태가 업데이트가 되는데 이는 ‘p.getLinkState(modelID, link_index)’ 와 ‘getJointState(modelID, link_index)’ 코드를 통해 업데이트가 진행된다. 필드로봇을 제어하는 코드와 학습에 따라 상태 업데이트 코드를 통해 위치, 토크 등 필드로봇의 현재 상태가 출력된다[56]. PyBullet 엔진에서 작업 경로 학습을 위해 필드로봇의 URDF 모델은 Fig. 69와 같다.

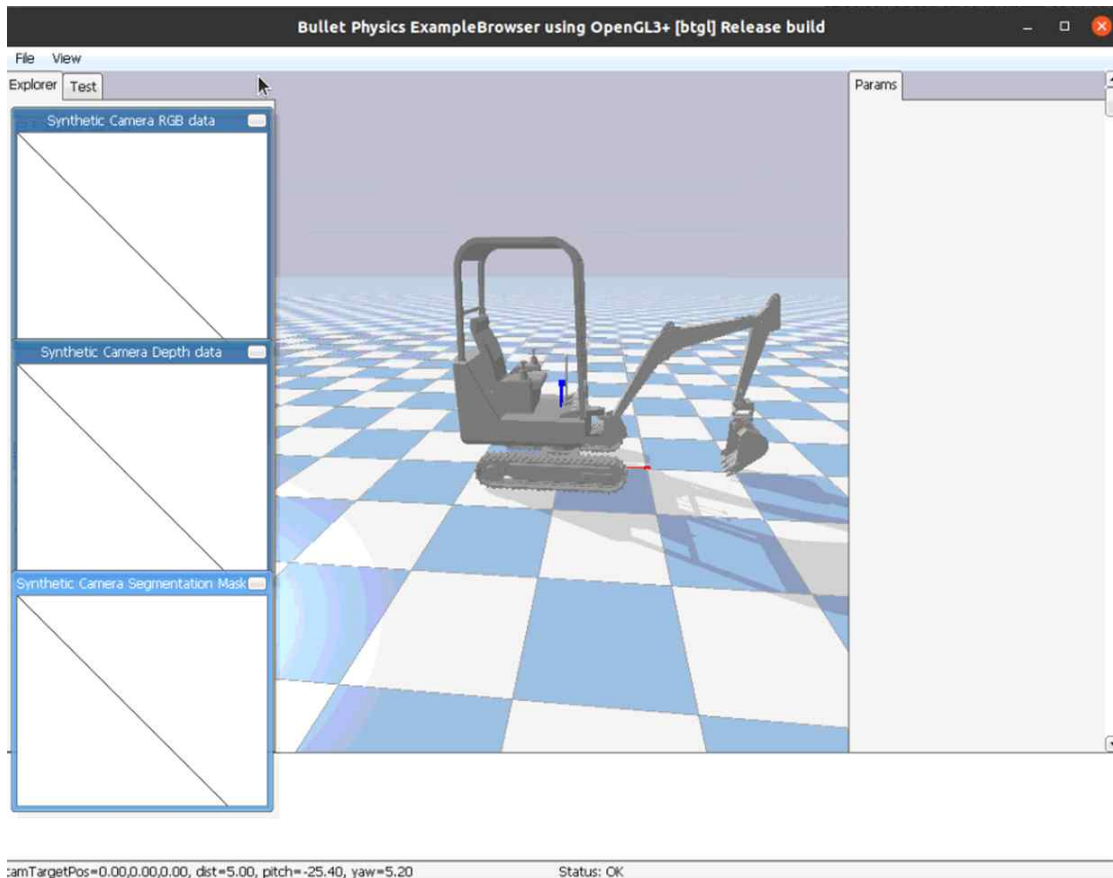


Fig. 69 URDF model of field robot for reinforcement learning

필드 로봇의 작업 경로를 학습하기 위한 URDF 모델을 구성하였다. 다음으로 학습을 하기 위해서는 강화학습 환경 모델을 만들어야 한다. 본 논문에서는 환경 모델은 GYM이라 부르는 OpenAI에서 공개한 인터페이스를 사용하였다. GYM은 강화학습 알고리즘을 개발하고 비교하기 위한 Toolkit으로 TensorFlow, Theano와 같은 수치계산 라이브러리와 호환된다. GYM 라이브러리를 이용하여 구성된 환경 모델은 일반적인 알고리즘을 작성하기 위한 공유 인터페이스가 있는데 Fig. 70과 같이 구성된다 [56]. 강화학습 알고리즘을 적용하고 테스트하기 위한 GYM 환경 모델을 생성하는 방식이다.

The figure illustrates the structure and setup of a custom GYM environment. It is divided into several sections:

- File Structure:** A tree view showing the directory layout:


```
gym-basic/
  README.md
  setup.py
  gym_basic/
    __init__.py
    envs/
      __init__.py
      basic_env.py
      basic_env_2.py
```
- How to Register your Environment:** A text box providing instructions:

Navigate up in your terminal to the folder that contains gym-basic, and use pip

```
pip install -e gym-basic
```
- gym-basic/setup.py:** A code snippet for the setup script:


```
from setuptools import setup

setup(name='gym_basic', version='0.0.1', install_requires=['gym'] )
```
- gym-basic/gym_basic/__init__.py:** A code snippet for the package's initialization:


```
from gym.envs.registration import register

register(id='basic-v0', entry_point='gym_basic.envs:BasicEnv',)

register(id='basic-v2', entry_point='gym_basic.envs:BasicEnv2',)
```
- gym-basic/gym_basic/envs/__init__.py:** A code snippet for the environment sub-package:


```
from gym_basic.envs.basic_env import BasicEnv
from gym_basic.envs.basic_env_2 import BasicEnv2
```

Fig. 70 Open AI GYM environment model structure

환경 모델에는 몇 가지 필수 기능이 필요하다. ‘__init__’에서는 고정된 이름과 유형을 가지는 두 가지 변수를 생성해야 한다. 여기서 두 가지 변수는 ‘self.action_space’와 Discrete라는 1차원과 Box라는 N 차원의 ‘self.observation_space’이다. Reset은 ‘self.observation_space’ 내에 있는 값을 반환해주는 함수로 환경 모델을 다시 시작해 주고 현재 상태를 추적하여 ‘Discrete observation space’에서의 상태 값은 정수를 ‘Box’에서의 상태 값은 ‘numpy.array’라는 변수를 호출하는 기능을 한다. Step은 하나의 입력 매개 변수인 행동 값으로 ‘self.action_space’를 가진다. 행동 값은 정수 또는 ‘numpy.array’일 수 있다. 이것은 응답 기능으로 에이전

트가 수행하려는 행동을 제공하고 환경은 에이전트에서 가져온 상태를 반환한다. 여기서 반환 값은 '4-tuple' 로 첫 번째 tuple인 상태는 'self.observation_space' 의 Reset 함수의 반환과 같은 유형의 변수, 두 번째 tuple인 보상은 에이전트가 행동에 대한 결과를 알려주는 값, 세 번째 tuple인 Done은 환경이 끝점에 도달했을 때는 'True' 이고 재설정되어야 하거나 그렇지 않은 경우 'False' 인 불리언 값, 네 번째 tuple인 Info는 버그 수정에 사용할 수 있는 사전 값으로 구성된다[56].

이러한 형식을 이용하여 필드로봇의 환경 모델은 행동 공간(action space)은 필드로봇의 버켓 끝단 위치와 버켓 끝단 방향으로 버켓 끝단의 x, y, z 인 3개의 값을 가지고 관찰 공간(observation space)은 필드로봇의 버켓 끝단의 오차 값으로 $e(x), e(y), e(z)$ 3개의 값을 가진다. 보상 값은 버켓 끝단의 x, y, z 위치와 초기 위치의 오차가 0.1 m 이내에 들어올 경우 좋은 행동에 대한 보상 값을 부여하고 현재 단계에서 다음 단계로 넘어가고 0.1 m 이내에 들어오지 못할 경우 나쁜 행동에 대한 보상 값을 부여하고 현재 단계를 계속 학습을 수행하도록 설계하였다[56].

5.2.3 학습 시뮬레이션

첫 번째 필드로봇의 작업 경로 학습을 위한 학습 결과를 확인하기 위해 Table 11과 같이 랜덤 포인트 19개를 설정하고 10,000,000회 학습을 수행하여 학습된 필드로봇의 버킷 끝단의 위치를 확인하고 설정한 랜덤 포인트 19개와 비교하여 학습 결과를 확인하였다. 시뮬레이션 화면은 Fig. 71과 같다.

Table 11 Random target point at the end tip of the bucket

Target Point [m]			
No.	x	y	z
1	2.568	0.000	0.069
2	2.620	0.000	0.056
3	2.400	0.000	0.060
4	2.079	0.000	0.061
5	2.413	0.000	0.064
6	2.091	0.000	0.073
7	2.842	0.000	0.057
8	2.933	0.000	0.071
9	2.378	0.000	0.057
10	2.359	0.000	0.056
11	2.841	0.000	0.065
12	2.589	0.000	0.062
13	2.825	0.000	0.068
14	2.135	0.000	0.066
15	2.914	0.000	0.054
16	2.677	0.000	0.074
17	1.926	0.000	0.061
18	2.881	0.000	0.067
19	2.802	0.000	0.066

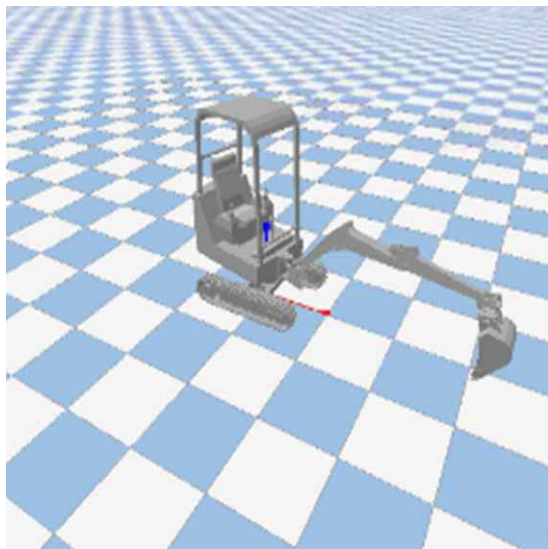
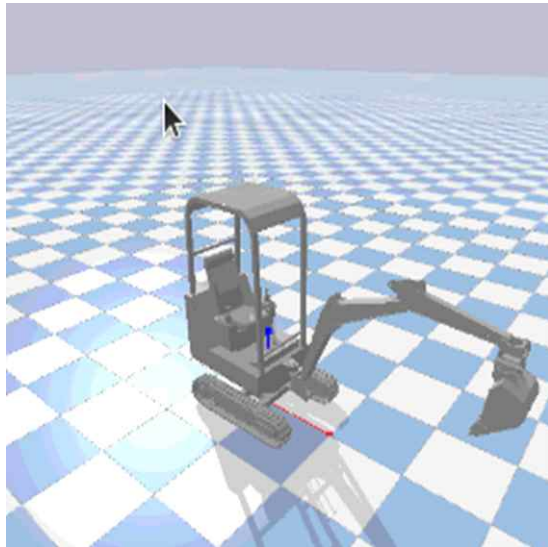
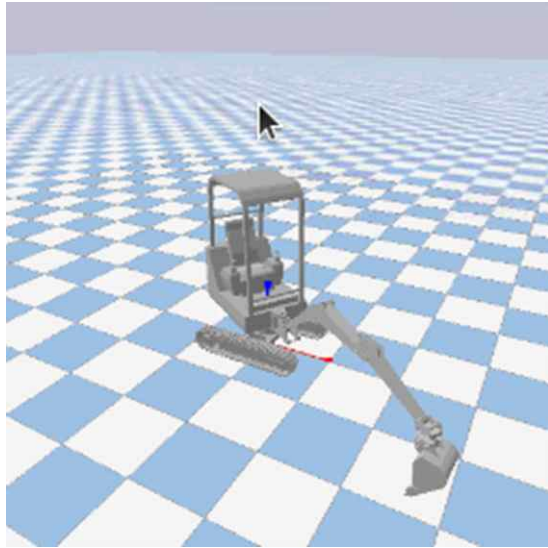


Fig. 71 Simulation for random target points

두 번째 필드로봇의 작업 경로 학습은 필드로봇 정면의 일직선 평탄화 작업 시나리오로 진행하였다. 필드로봇의 평탄화 작업 범위는 Fig. 72와 같이 약 2~3 m 구간에서 0.1 m 마다 작업 경로 포인트를 설정하여 작업경로를 생성하였다. 설정된 작업 경로 값은 Table 12와 같다. 작업 경로를 학습하기 위한 학습 횟수는 10,000,000 회로 설정하여 학습을 진행하였다.

Table 12 Target point at the end tip of the bucket for straight path

Target Point [m]			
No.	x	y	z
1	3.000	0.000	0.000
2	2.900	0.000	0.000
3	2.800	0.000	0.000
4	2.700	0.000	0.000
5	2.600	0.000	0.000
6	2.500	0.000	0.000
7	2.400	0.000	0.000
8	2.300	0.000	0.000
9	2.200	0.000	0.000
10	2.100	0.000	0.000
11	2.000	0.000	0.000

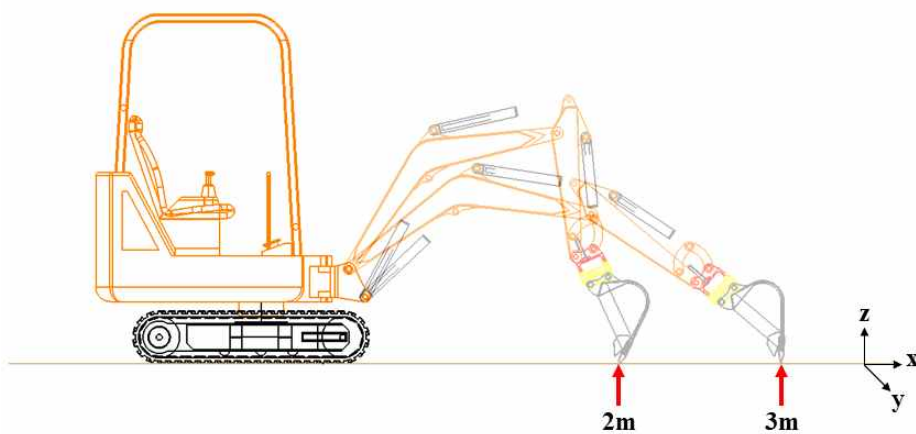


Fig. 72 Working range at the end tip of the bucket for a straight path

5.2.4 학습 시뮬레이션 결과

첫 번째 랜덤 포인트 19개를 학습하여 Table 13과 같이 학습 결과를 확인하였다. 오차 범위인 0.1 m 이내에 학습이 수행되어 동작하는 것을 확인할 수 있었다. 결과에 대해 비교한 버켓 x, y, z 축의 그래프는 Fig. 73 ~ 75와 같다.

Table 13 Result of random target point at the end tip of the bucket

No	Target Point [m]			Learning Point [m]		
	x	y	z	x	y	z
1	2.568	0.000	0.069	2.665	-0.010	0.087
2	2.620	0.000	0.056	2.718	0.008	-0.029
3	2.400	0.000	0.060	2.304	0.076	0.084
4	2.079	0.000	0.061	1.980	-0.031	0.080
5	2.413	0.000	0.064	2.508	-0.045	0.089
6	2.091	0.000	0.073	1.994	0.020	0.109
7	2.842	0.000	0.057	2.941	-0.027	0.073
8	2.933	0.000	0.071	3.026	-0.043	0.102
9	2.378	0.000	0.057	2.281	-0.032	0.074
10	2.359	0.000	0.056	2.259	0.007	0.073
11	2.841	0.000	0.065	2.938	0.002	0.090
12	2.589	0.000	0.062	2.559	0.005	-0.038
13	2.825	0.000	0.068	2.918	0.022	0.095
14	2.135	0.000	0.066	2.036	0.026	0.091
15	2.914	0.000	0.054	3.010	0.043	0.068
16	2.677	0.000	0.074	2.694	0.009	-0.026
17	1.926	0.000	0.061	1.827	0.027	0.081
18	2.881	0.000	0.067	2.978	-0.015	0.093
19	2.802	0.000	0.066	2.885	-0.019	-0.032

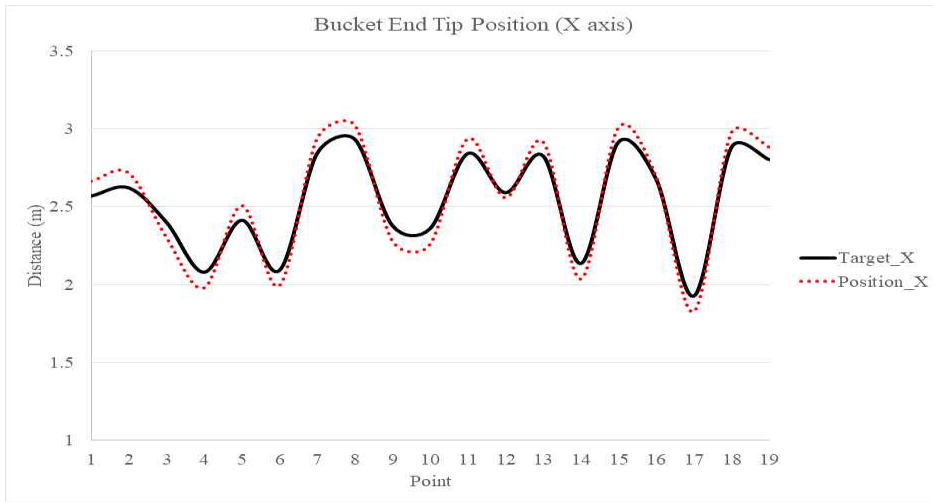


Fig. 73 x-axis position of the end tip of the bucket for random point

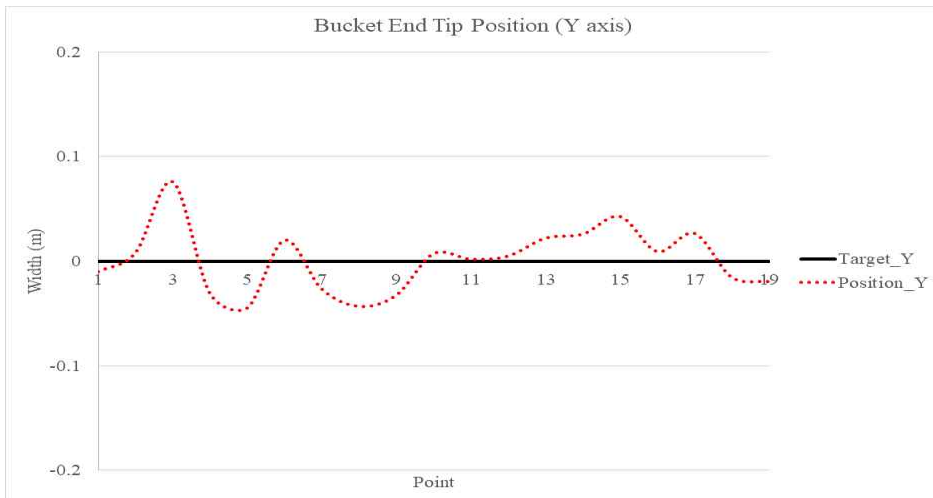


Fig. 74 y-axis position of the end tip of the bucket for random point

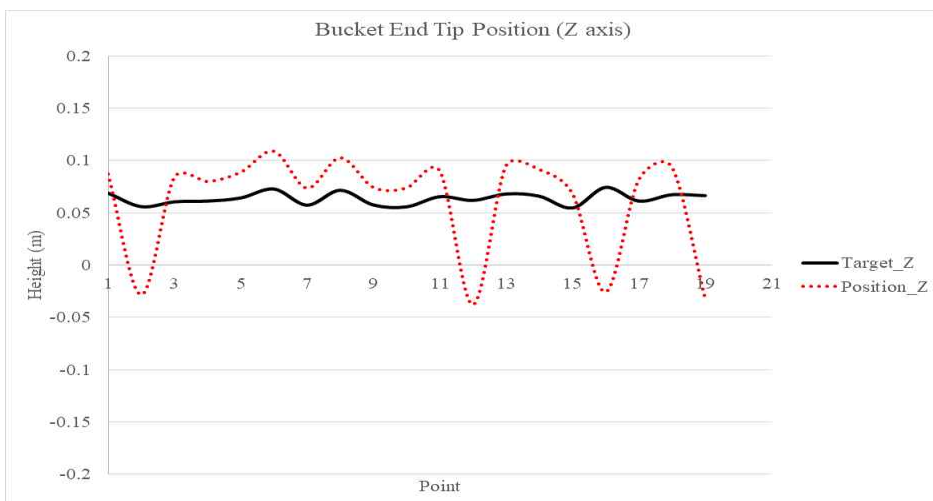


Fig. 75 z-axis position of the end tip of the bucket for random point

그리고 학습 결과에 대해 목표 포인트와 학습 포인트의 오차를 Table 14와 같이 확인하였다. 환경 모델에서 설정된 오차에 맞게 Fig. 76 ~ 78과 같이 버킷 끝단의 위치를 0.1 m 이내에서 작업할 수 있도록 학습이 된 것을 알 수 있다.

Table 14 Error value of random target point at the end tip of the bucket

Error [m]			
No.	x	y	z
1	-0.098	0.010	-0.018
2	-0.098	-0.008	0.084
3	0.096	-0.076	-0.023
4	0.099	0.031	-0.019
5	-0.095	0.045	-0.024
6	0.097	-0.020	-0.036
7	-0.099	0.027	-0.016
8	-0.093	0.043	-0.031
9	0.097	0.032	-0.017
10	0.100	-0.007	-0.018
11	-0.097	-0.002	-0.025
12	0.031	-0.005	0.100
13	-0.093	-0.022	-0.027
14	0.099	-0.026	-0.025
15	-0.096	-0.043	-0.014
16	-0.017	-0.009	0.100
17	0.099	-0.027	-0.020
18	-0.097	0.015	-0.026
19	-0.083	0.019	0.098

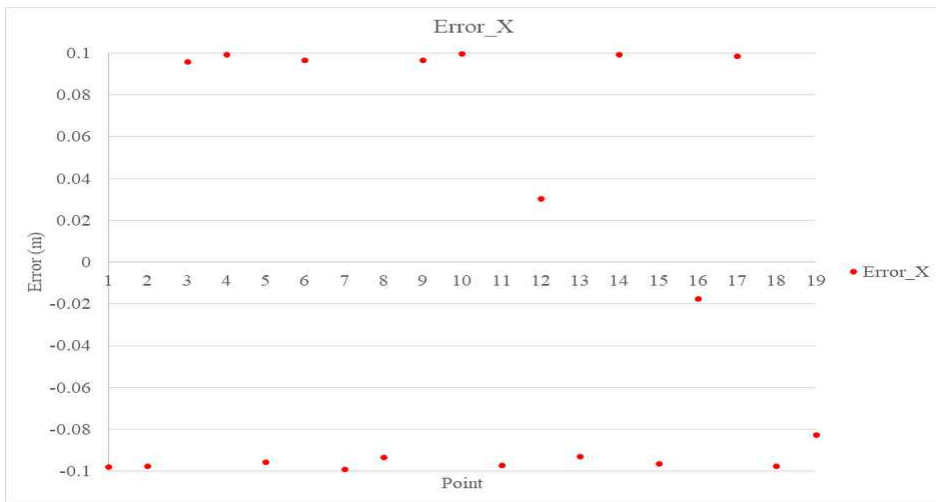


Fig. 76 x-axis error value of the end tip of the bucket for random point

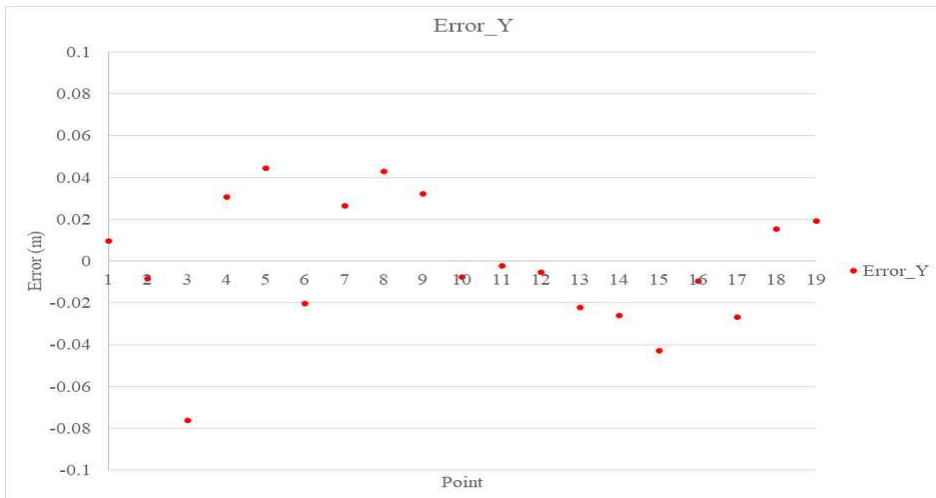


Fig. 77 y-axis error value of the end tip of the bucket for random point

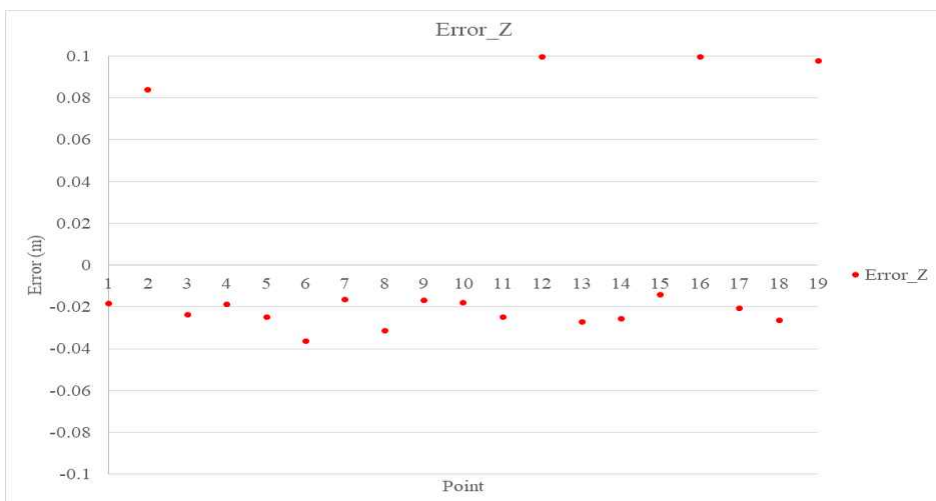


Fig. 78 z-axis error value of the end tip of the bucket for random point

두 번째 필드로봇 정면의 일직선 평탄화 작업 경로를 학습하여 Table 15와 같이 경로에 대한 목표 값과 경로 학습에 대한 결과와 Table 16과 같이 학습된 포인트에 대한 붐, 암, 버킷의 토크 값을 확인하였다. 오차 범위인 0.1 m이내에 학습이 수행되어 동작하는 것을 확인할 수 있었다. 학습된 경로 포인트의 버킷 끝단 x, y, z 축 결과 그래프는 Fig. 79 ~ 81과 같으며 붐, 암, 버킷의 토크 값 결과 그래프는 Fig. 82 ~ 84와 같다. 경로 학습 과정에서 붐, 암, 버킷의 조인트 토크 값 변화에 대한 그래프는 Fig. 85 ~ 87과 같다.

Table 15 Learning result of straight path at the end tip of the bucket

No	Target Point [m]			Learning Point [m]		
	x	y	z	x	y	z
1	3.000	0.000	0.000	3.041	0.041	-0.063
2	2.900	0.000	0.000	2.933	-0.042	-0.062
3	2.800	0.000	0.000	2.756	-0.093	-0.055
4	2.700	0.000	0.000	2.623	0.084	0.000
5	2.600	0.000	0.000	2.532	-0.018	-0.025
6	2.500	0.000	0.000	2.407	-0.067	-0.022
7	2.400	0.000	0.000	2.373	-0.092	-0.010
8	2.300	0.000	0.000	2.370	0.080	-0.077
9	2.200	0.000	0.000	2.135	-0.068	0.024
10	2.100	0.000	0.000	2.030	0.063	-0.066
11	2.000	0.000	0.000	1.966	-0.006	-0.087

Table 16 Torque values of boom, arm and bucket for straight path

No	Torque [Nm]		
	Boom	Arm	Bucket
1	77673.075	-37691.850	11414.249
2	-100000.000	-100000.000	33045.086
3	100000.000	-100000.000	14456.230
4	100000.000	-61184.574	-16833.014
5	100000.000	100000.000	-32504.828
6	-100000.000	-34867.761	16603.407
7	-100000.000	14172.622	-3794.785
8	100000.000	44470.111	-12368.967
9	100000.000	100000.000	-54538.272
10	-49853.746	83307.823	-22895.921
11	-100000.000	-64781.229	15621.469

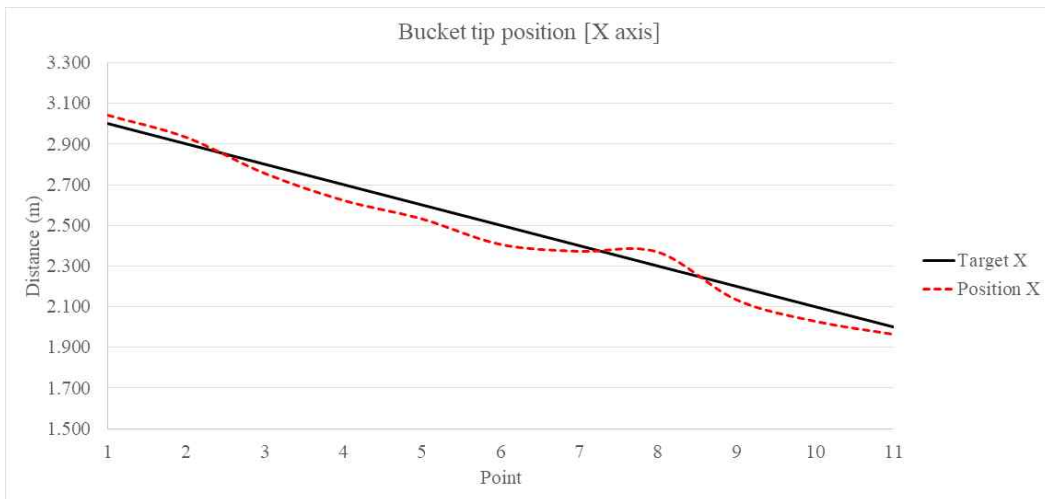


Fig. 79 x-axis position of the end tip of the bucket for straight path

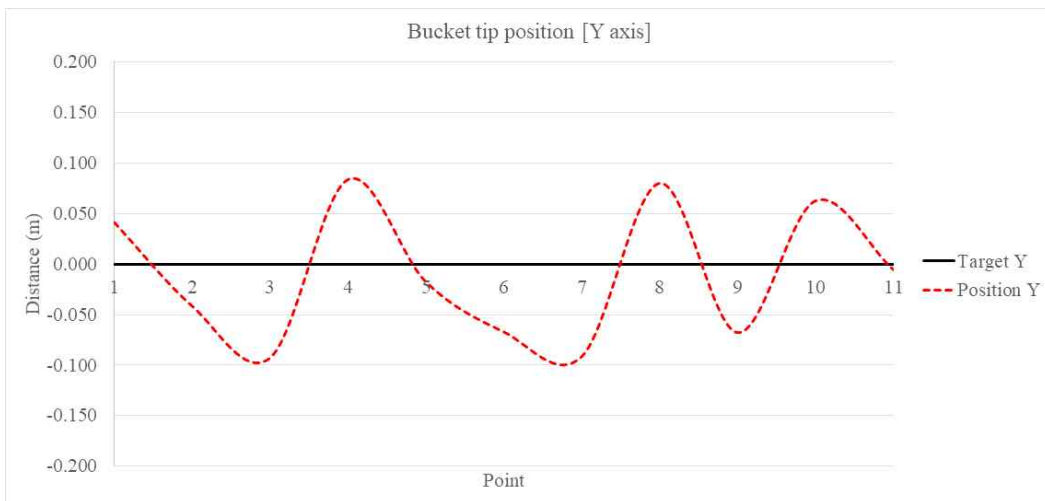


Fig. 80 y-axis position of the end tip of the bucket for straight path

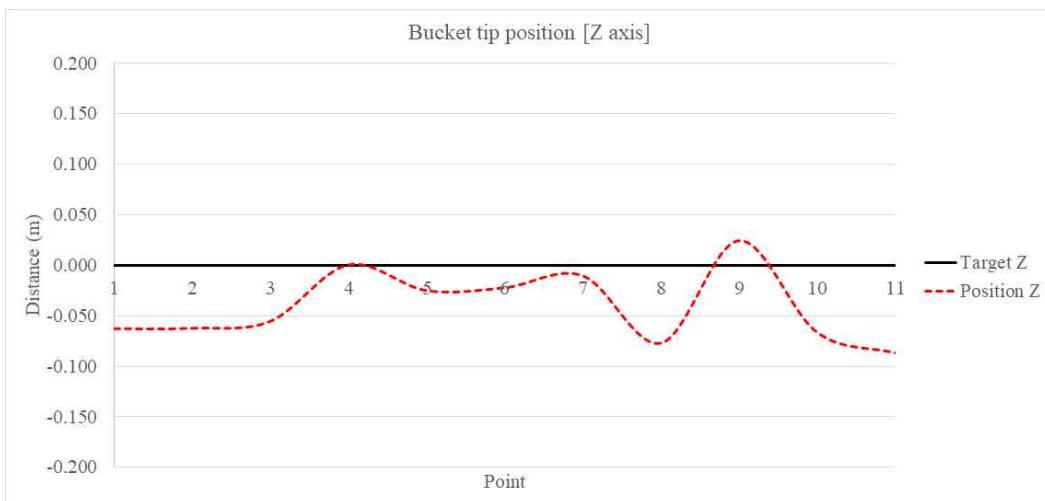


Fig. 81 z-axis position of the end tip of the bucket for straight path

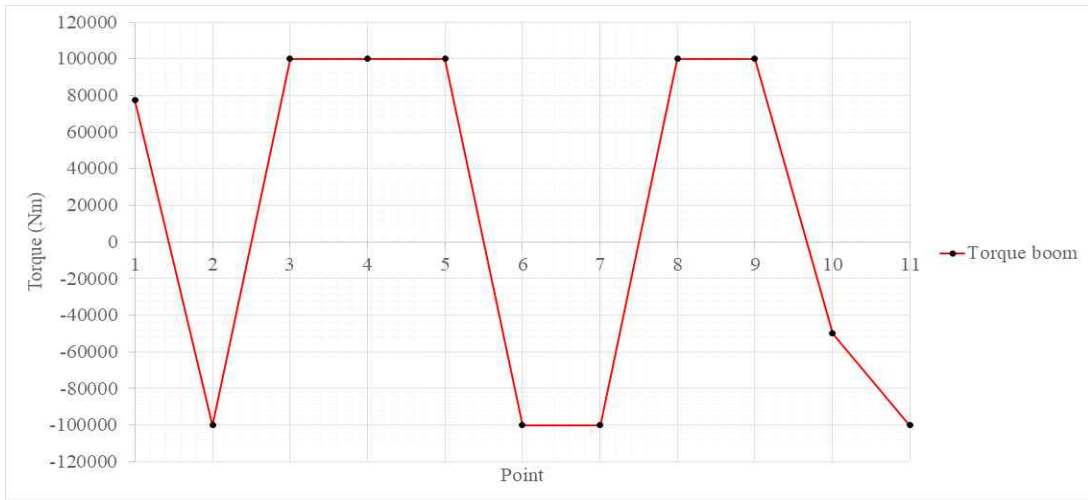


Fig. 82 Torque values of boom for straight path

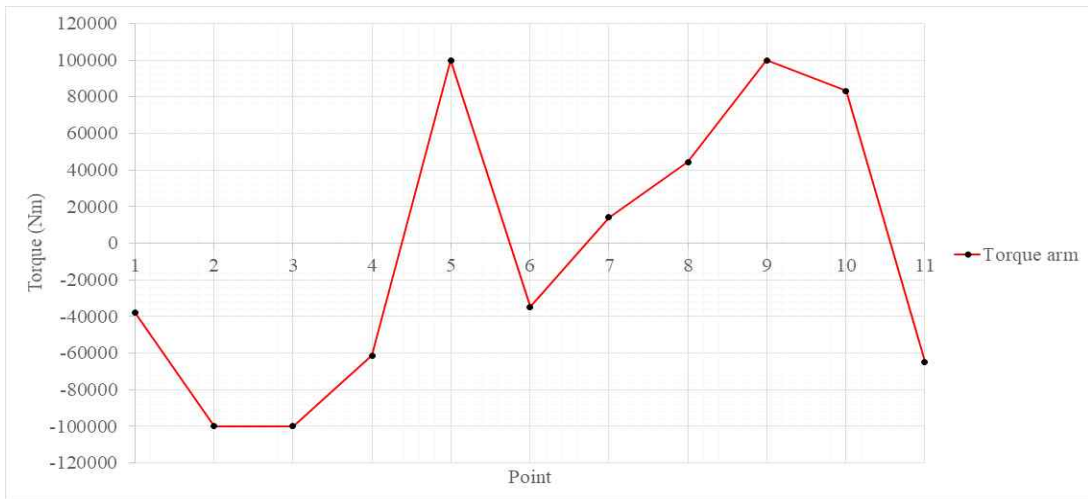


Fig. 83 Torque values of arm for straight path

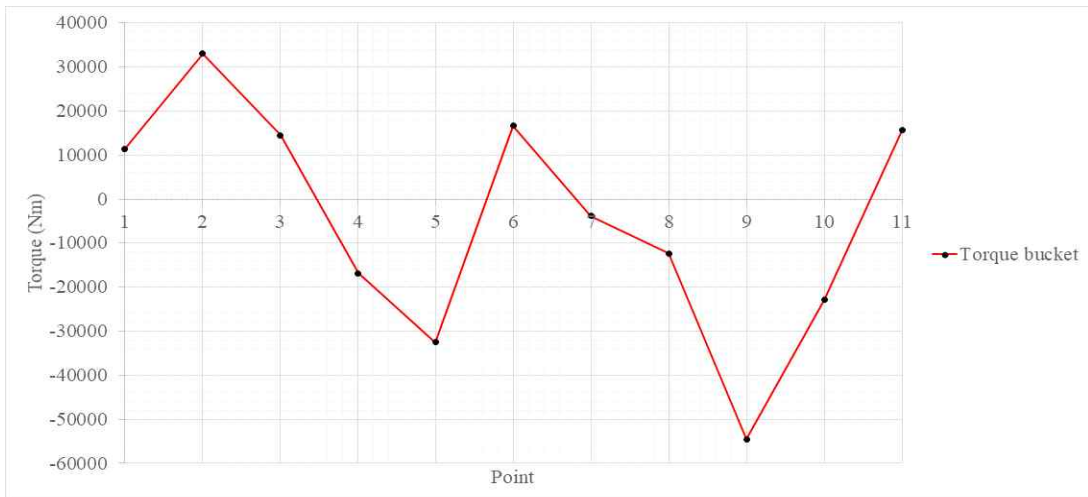


Fig. 84 Torque values of bucket for straight path

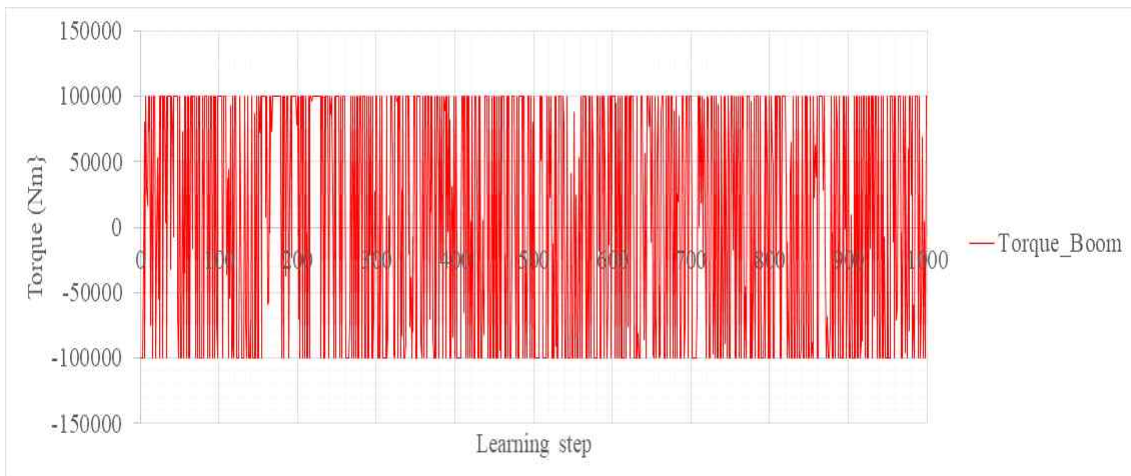


Fig. 85 Results of boom joint torque value changes during path learning

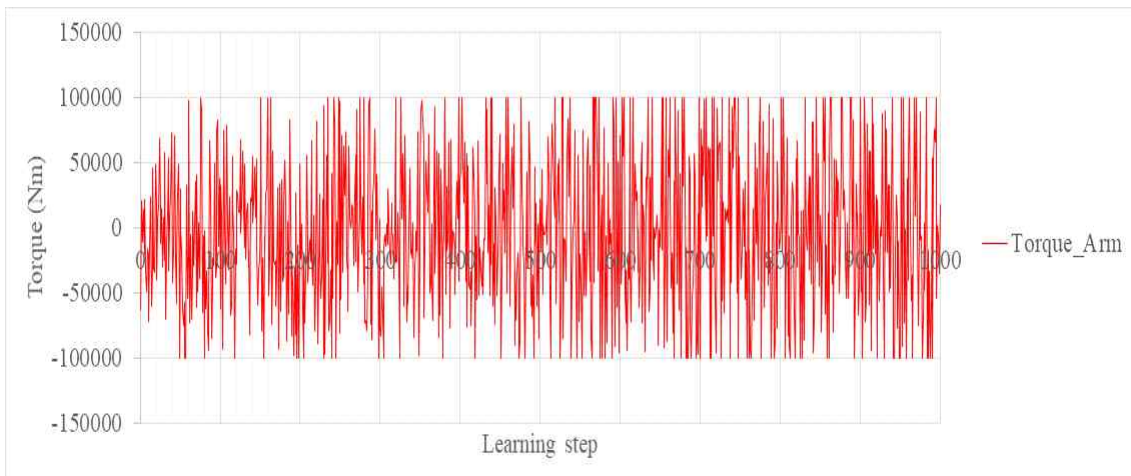


Fig. 86 Results of arm joint torque value changes during path learning

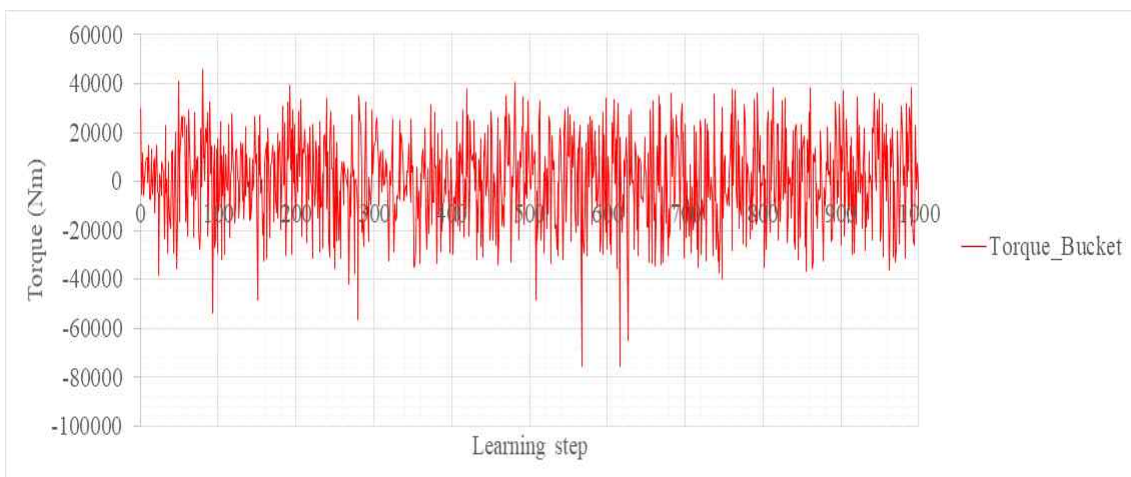


Fig. 87 Results of bucket joint torque value changes during path learning

그리고 학습 결과에 대해 목표 포인트와 학습 포인트의 오차를 Table 17과 같이 확인하였다. 환경 모델에서 설정된 오차에 맞게 Fig. 88 ~ 90과 같이 버켓 끝단의 위치를 0.1 m 이내에서 작업할 수 있도록 학습이 된 것을 알 수 있다.

Table 17 Error value of straight path at the end tip of the bucket

Error [m]			
No.	x	y	z
1	-0.041	-0.041	0.063
2	-0.033	0.042	0.062
3	0.044	0.093	0.055
4	0.077	-0.084	0.000
5	0.068	0.018	0.025
6	0.093	0.067	0.022
7	0.027	0.092	0.010
8	-0.070	-0.080	0.077
9	0.065	0.068	-0.024
10	0.070	-0.063	0.066
11	0.034	0.006	0.087

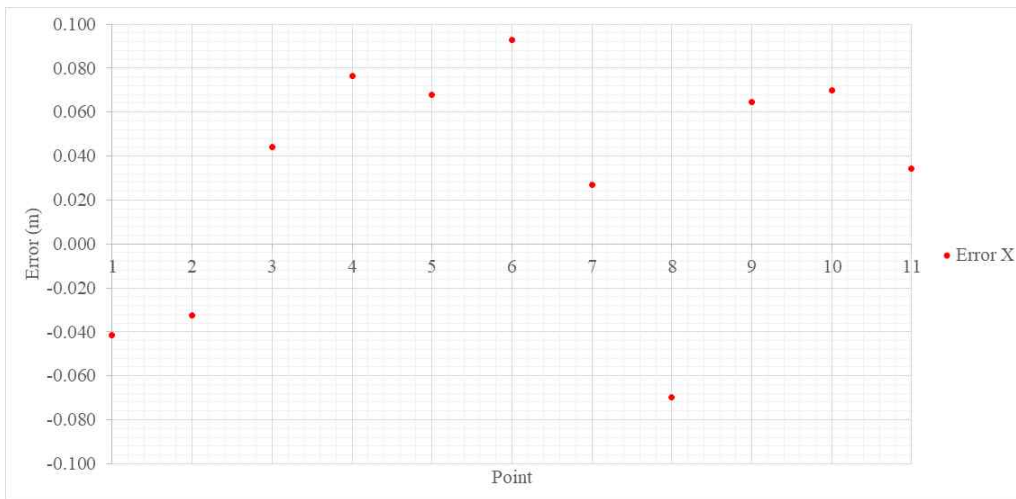


Fig. 88 x-axis error value of the end tip of the bucket for straight path

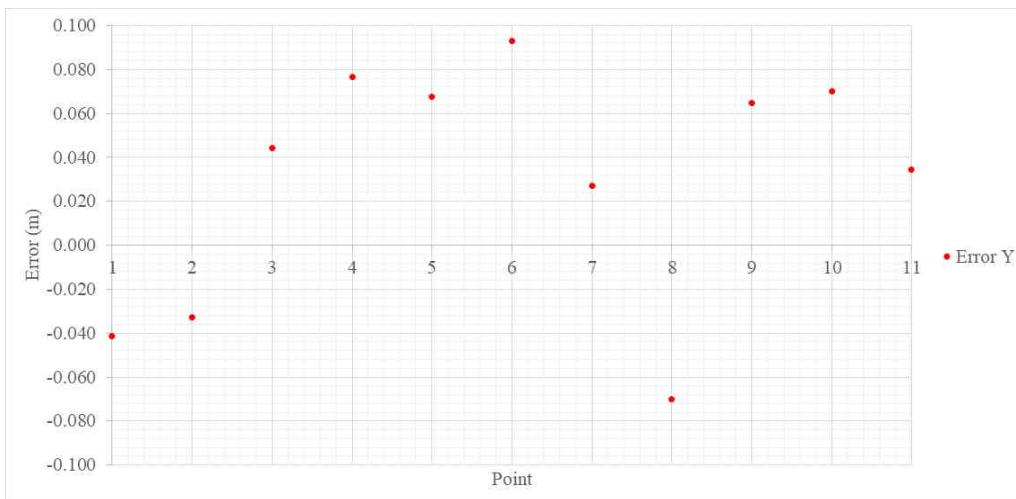


Fig. 89 y-axis error value of the end tip of the bucket for straight path

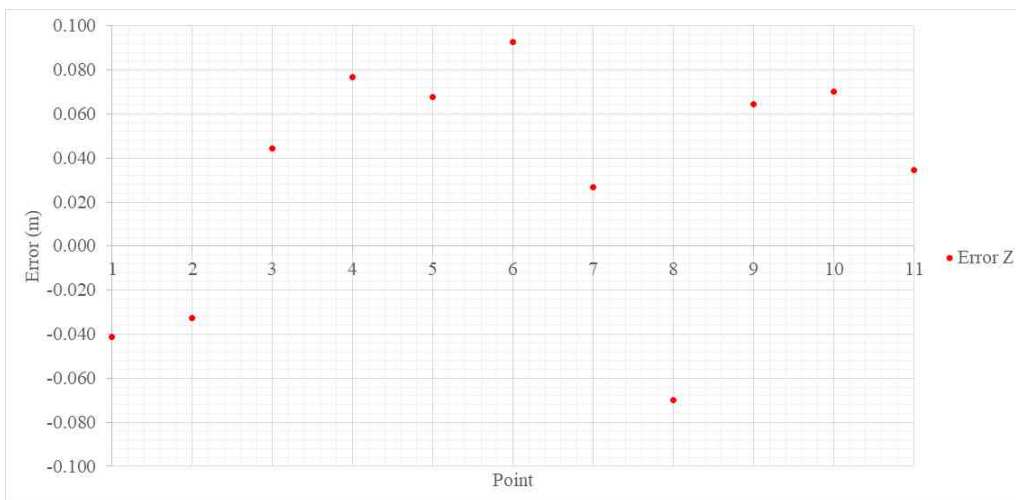


Fig. 90 z-axis error value of the end tip of the bucket for straight path

5.3 학습 결과 시뮬레이션

필드로봇 정면의 일직선 평탄화 작업 경로 학습 결과로 도출된 붐, 암, 버킷의 조인트 각도를 스마트 필드로봇 시뮬레이션 모델에 입력하여 학습된 결과에 대한 시뮬레이션을 진행하여 결과를 비교하였다. 작업 경로 학습된 결과에 대한 붐, 암, 버킷의 조인트 각도와 시뮬레이션 모델에서 도출된 붐, 암, 버킷의 조인트 각도는 Table 18과 같다. 붐, 암, 버킷의 결과 그래프는 Fig. 91 ~ 93과 같다. 학습 모델은 유압 요소가 고려되지 않은 기구학적 모델이며 시뮬레이션 모델은 유압 요소가 고려된 모델이다. Table 18과 같이 유압 요소의 여부에 따라 동작 각도의 차이가 있다는 것을 확인할 수 있다. 암의 결과에서 Point 8, 10, 11과 같이 시뮬레이션 모델은 제한된 동작각도 내에서 모델이 동작하였으나 학습 모델은 Table 2에서 정의한 제한 동작각도를 넘어선 결과가 나타나는 것을 확인할 수 있다.

Table 18 Result of angle values of boom, arm, bucket for straight path

No	Learning Angle [deg]			Simulation Angle [deg]		
	Boom	Arm	Bucket	Boom	Arm	Bucket
1	33.41	-65.21	-24.26	37.61	-64.98	-25.24
2	30.56	-65.65	-4.21	33.43	-65.13	-4.70
3	29.29	-52.79	9.12	31.73	-53.05	9.05
4	31.12	-58.99	-12.62	31.72	-58.62	-9.48
5	31.10	-64.03	-3.65	31.47	-63.81	-6.50
6	25.98	-39.49	-25.85	28.18	-39.77	-23.73
7	23.42	-29.87	-32.16	25.28	-30.01	-31.54
8	21.36	-11.81	-60.45	23.03	-27.30	-60.09
9	25.15	-27.88	-30.44	23.00	-27.30	-33.62
10	21.20	-20.30	-36.84	22.94	-27.30	-33.39
11	19.73	-0.46	-58.99	21.27	-27.30	-57.58

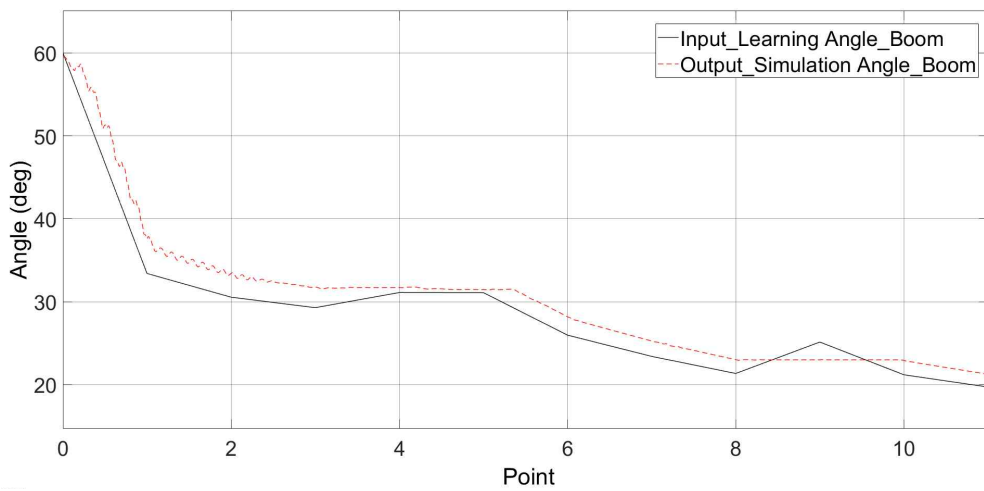


Fig. 91 Result of angle values of boom for straight path

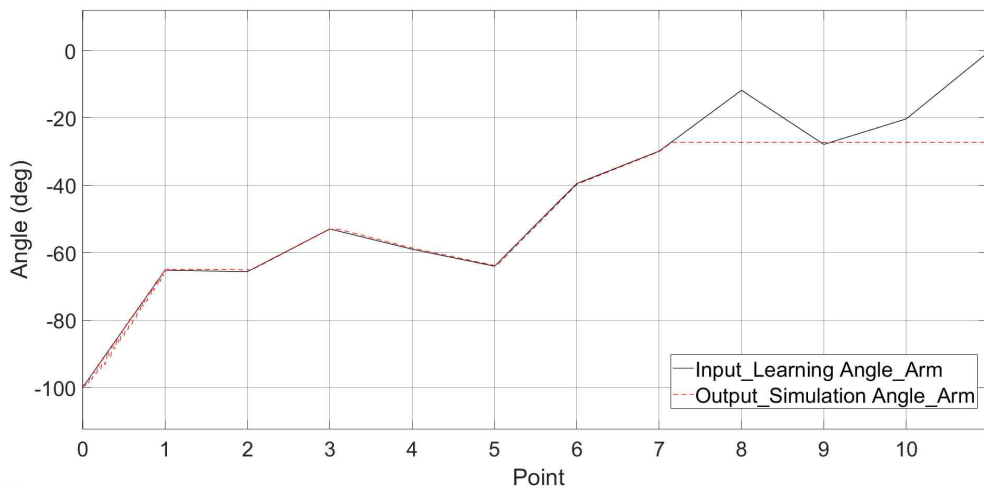


Fig. 92 Result of angle values of arm for straight path

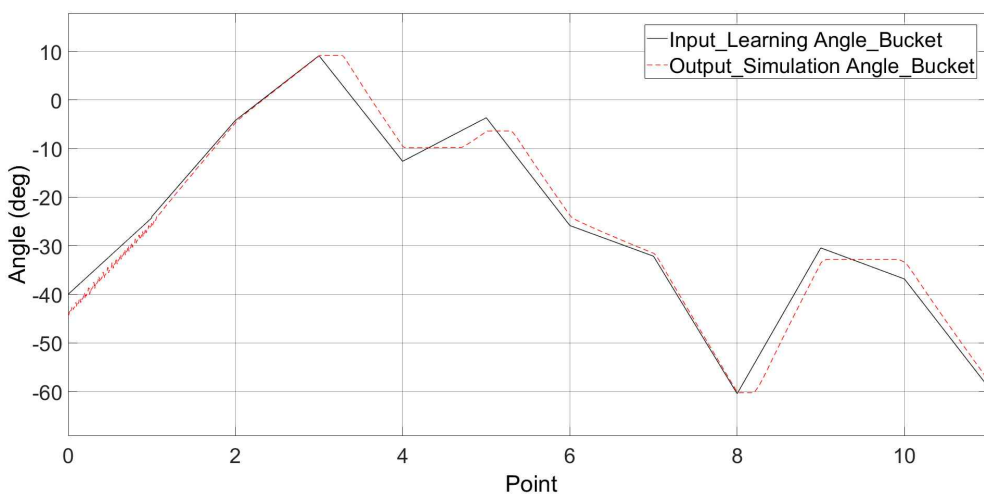


Fig. 93 Result of angle values of bucket for straight path

6. 결론 및 향후 연구

본 논문에서는 틸트로테이터가 적용된 6DOF 스마트 필드로봇 시스템에 대해 MATLAB/Simulink의 Simscape를 사용한 시뮬레이션 모델 구축과 Python, Pybullet을 사용한 강화학습 플랫폼 모델 구축 두 가지의 연구를 진행하였다. 첫 번째로 시뮬레이션 모델 구축에 대한 연구로 필드로봇의 유압 시스템, 기구 모델, 제어기 모델을 MATLAB/Simulink의 Simscape를 이용하여 시뮬레이션 모델을 구축했다. 6DOF 스마트 필드로봇 시스템의 시뮬레이션 모델에 대한 연구를 통해 다음과 같은 결론을 도출하였다.

1. 기존 필드로봇 시스템의 제한된 동작 범위로 인해 작업의 범위가 제한되어 작업의 효율성을 떨어뜨리는 단점을 가지고 있다. 이러한 단점을 보완하기 위해 틸트로테이터 시스템을 기존 필드로봇 시스템과 결합하여 필드로봇의 손목 역할을 하는 틸트로테이터 시스템으로 인해 작업의 효율성이 증가하는 6DOF 필드로봇 시스템 모델을 제시함.
2. 기존 필드로봇 시스템과 틸트로테이터 시스템을 결합한 6DOF 필드로봇 시스템에 대해 기준 좌표계를 설정하고 D-H 파라미터 데이터를 도출하고 순기구학과 역기구학에 대한 수학적 모델을 정립하고 MATLAB의 GUIDE 기능을 이용해 순기구학 모델과 역기구학 모델의 수학적 모델을 검증하는 프로그램을 설계하고 순기구학의 수학적 모델 요소인 조인트 각도와 역기구학 수학적 모델 요소인 오리엔테이션과 버켓의 끝단 위치를 이용하여 정립한 수학적 모델에 대해 오차 없이 동일한 결과 값을 얻음으로 수학적 모델의 타당성을 검증함.
3. 필드로봇 시스템의 시뮬레이션 모델을 구축하기 위해 실험실에서 보유하고 있는 1.5톤 필드로봇의 제원에 대해 설계사양을 확인하여 CATIA를 이용해서 필드로봇의 3D 모델을 설계함.
4. 필드로봇 시스템의 시뮬레이션 모델을 구축하기 위해 설계된 필드로봇 3D 필드로봇을 이용하여 시뮬레이션에 필요한 기구/동역학 요소인 Multibody 모델을 설계

하였다. Multibody 모델은 Solidworks의 XML 변환 기능을 이용하여 3D 모델의 Assembly 정보를 포함한 데이터 파일과 Multibody 모델을 생성함.

5. 필드로봇 시스템의 시뮬레이션 모델 중 유압 시스템을 모델링하기 위해 1.5톤 필드로봇의 유압 시스템의 제원, 밸브 개구면적 선도, 유압 회로도를 분석하여 메인 펌프가 포함된 파워팩 시스템, 유압 방향 제어를 위한 메인 컨트롤 밸브, 필드로봇 시스템의 구동을 위한 액추에이터들을 모델링하였다. 유압 모델 검증에 위해 유압 실린더의 유압 성능과 동작 성능을 제원과 비교하였으며 유압 성능은 최대 210bar 내에서 유압 시스템이 작동하였으며, 동작 성능에서는 각 조인트 각도의 오차가 1 ~ 2 deg 정도의 미소한 차이를 보였으나 출력 값이 입력 값과 유사한 경향을 보이는 것을 확인함.

6. 필드로봇 시스템의 제어 시스템은 피드백 구조를 가지는 PID 제어기를 사용하였으며 시행착오 방법을 사용하여 각 조인트에 해당되는 PID gain의 최적 값을 찾아냄.

7. 최종 필드로봇 시뮬레이션 모델에서 3가지의 시뮬레이션을 진행하였다. 첫 번째 시뮬레이션으로 sine 파형을 이용한 필드로봇의 붐, 암, 버켓, 틸트로테이터의 단독 동작에 대한 시뮬레이션을 진행하였다. 시뮬레이션 결과는 일부 구간에서 0.3 ~ 0.5 sec의 차이를 보였지만 전반적으로 입력 값에 대해 출력 값은 유사한 경향을 보이는 것을 확인함.

8. 두 번째 시뮬레이션으로 굴착 작업에 대한 시나리오를 모사한 입력 각도 값을 이용한 붐, 암, 버켓, 틸트로테이터의 단독동작에 대한 시뮬레이션을 진행하였다. sine 파형 시뮬레이션과 유사하게 일부 구간에서 0.3 ~ 0.5 sec의 차이를 보였으며 암의 경우 0 ~ 0.5 sec에서 많은 차이를 보였으나 이는 필드로봇의 초기 위치에서 입력 값으로 동작하면서 발생한 오차이다. 그러나 전반적으로 입력 값에 대한 출력 값은 유사한 경향을 보이는 것을 확인함.

9. 세 번째 시뮬레이션으로 굴착 작업에 대한 시나리오를 모사한 입력 값을 이용한

뿔, 압, 버켓의 복합동작에 대한 시뮬레이션을 진행하였다. 단독동작과 유사한 경향을 보였으나 일부 구간에서 진동이 발생하는 것을 확인하였다. 이는 필드로봇의 유압 동력원인 메인 펌프가 고정형 타입으로 방향 전환 할 때나 방향 전환 후 일정 시간 동안 최대 압력을 토출함에 있어 불완전한 영역에서 발생하는 속도변화에 따라 발생한 것으로 추정된다. 이를 보완해줄 수 있는 제어 방식이나 게인 값 조정에 대한 추가적인 연구가 필요할 것으로 사료됨.

두 번째로 설계된 6DOF 스마트 필드로봇 모델에 대한 작업 경로 학습을 위해 Python과 Python Toolkit인 PyBullet을 이용하여 6DOF 스마트 필드로봇 강화학습 플랫폼을 구축하였다. 작업 경로 학습을 위해 구축한 6DOF 스마트 필드로봇 강화학습 플랫폼에 대한 연구를 통해 다음과 같은 결론을 도출하였다.

1. 강화학습에 대한 학습을 통해 현재 유행하고 있는 분야인 AI, 인공지능 중 한 파트인 강화학습의 알고리즘에 대한 정의 및 이론 등 기본적인 개념을 정립함.
2. CATIA를 사용하여 설계된 필드로봇의 3D 모델을 강화학습의 플랫폼에 적용하기 위해서는 URDF 모델이 필요하다. 이 URDF 모델은 Solidworks의 URDF 변환 기능을 사용하여 강화학습에 필요한 URDF 모델과 강화학습 알고리즘을 적용하고 테스트하기 위한 GYM 환경 모델을 생성함.
3. 스마트 필드로봇 강화학습 플랫폼에서 사용한 강화학습 알고리즘은 근위 정책 최적화 알고리즘(Proximal Policy Optimization Algorithm)을 적용하여 버켓 끝단의 위치에 대한 작업 경로를 학습함.
4. 강화학습을 통해 필드로봇의 버켓 끝단 경로를 두 가지 작업경로로 랜덤 포인트에 대한 경로와 일직선 평탄화 작업에 대한 경로를 목표 값으로 설정하여 학습을 진행하였으며 버켓 끝단의 위치의 x, y, z 축 오차 범위를 ± 0.1 m로 설정하여 10,000회 학습을 진행함.
5. 설정된 작업경로에 대해 학습된 필드로봇의 강화학습 모델은 목표로 한 x, y, z 축

의 오차 범위인 ± 0.1 m 이내에서 동작되며 시뮬레이션에서 중점적으로 확인한 x 축의 경로에 대해서는 랜덤 포인트에 대한 경로 학습과 일직선 평탄화 작업에 대한 경로 모두 설정된 경로를 추정하여 유사한 경로로 동작하는 것을 확인함.

6. 학습 결과 값 중 붐, 암, 버킷의 각도 값을 스마트 필드로봇 시뮬레이션 모델의 조인트 각도 값으로 입력하여 시뮬레이션 한 결과 입력 값에 대해 출력 값이 유사한 각도로 동작하는 것을 확인하였으며 학습 모델은 유압 요소가 고려되지 않은 기구학적 모델이며 시뮬레이션 모델은 유압 요소가 고려된 모델이다. 따라서 유압 요소 여부에 따라 동작 각도의 차이가 있는 것을 확인하였다. 그리고 학습 모델에서 제한 동작각도 범위를 벗어나는 값을 시뮬레이션 모델에서는 제한 동작각도 범위 내에서 작동하려는 경향을 확인함.

필드로봇 정면의 일직선 평탄화 작업 경로 학습 결과로 도출된 붐, 암, 버킷의 조인트 각도를 스마트 필드로봇 시뮬레이션 모델에 입력하여 학습된 결과에 대한 시뮬레이션을 진행하여 결과를 비교하였다. 작업 경로 학습된 결과에 대한 붐, 암, 버킷의 조인트 각도와 시뮬레이션 모델에서 도출된 붐, 암, 버킷의 조인트 각도는 Table 18과 같다. 붐, 암, 버킷의 결과 그래프는 Fig. 88 ~ 90과 같다. 학습 모델은 유압 요소가 고려되지 않은 기구학적 모델이며 시뮬레이션 모델은 유압 요소가 고려된 모델이다. Table 18과 같이 유압 요소의 여부에 따라 동작 각도의 차이가 있다는 것을 확인할 수 있다. 암의 결과에서 Point 8, 10, 11과 같이 시뮬레이션 모델은 제한된 동작각도 내에서 모델이 동작하였으나 학습 모델은 Table 2에서 정의한 제한 동작각도를 넘어선 결과가 나타나는 것을 확인할 수 있다.

본 논문에서 연구한 필드로봇 시뮬레이션 모델에 대해서는 향후 MATLAB/Simulink를 이용하여 구축한 시뮬레이션 모델을 이용하여 본 논문에서 적용한 필드로봇인 굴착기 이외에 다양한 필드로봇들을 적용하고 동작에 대한 최적화한 추가적인 제어 알고리즘을 통해 하나의 시뮬레이터를 구축한다면 다양한 필드로봇 개발하기 전 검증이 가능한 통합 필드로봇 시뮬레이터로 활용할 수 있을 것으로 사료된다. 그리고 필드로봇의 작업 경로 학습을 위한 강화학습 플랫폼을 활용하여 다양한 작업들에 대해 전문 운전자들의 데이터를 저장하여 작업경로들을 미리 생성하고 강화학습 플랫폼과 학습 연계를 시켜 실차에 적용한다면 조작성 미숙한 운전자들을 보조할 수 있는 머신 가이드스 연구에 기여할 수 있을 것으로 사료된다.

참고문헌

- [1] T. H. Lim, H. S. Lee and S. Y. Yang, “Development of Simulator of Hydraulic Excavator” , Journal of Drive and Control, Vol. 2, No. 1, pp. 63-68, 2005.
- [2] T. H. Lim and S. Y. Yang, “Development and Application of Simulator for Hydraulic Excavator” , Journal of the Korean Society for Precision Engineering, Vol. 23, No. 9, pp. 142-148, 2006.
- [3] S. W. Choi, Q. H. Le, T. G. Son and S. Y. Yang, “A Study on Construction of Control System for Wireless Remote Control of Small Field Robot” , Journal of Drive and Control, Vol. 17, No. 4, pp. 103-112, 2020.
- [4] S. Y. Yang, T. H. Lim, M. H. Lee and J. S. Yang, “Trend of Research and Development on Mechatronics for Construction Equipment” , Journal of Drive and Control, Vol. 3, No. 1, pp. 9-15, 2006.
- [5] HYUNDAI Construction Equipment, [http://http://www.hyundai-mh.com/ko](http://www.hyundai-mh.com/ko)
- [6] Steelwrist, <https://steelwrist.com/>
- [7] Engcon, <https://engcon.com/>
- [8] Rototilt, <https://www.rototilt.com/>
- [9] (주)제이케이, <http://www.jkattach.co.kr/>
- [10] Tiltpro, <https://www.tiltpro.co.kr>
- [11] (주)주현, <http://www.jhtr.co.kr/>
- [12] Y. J. Kim, T. G. Son, Y. S. Kim, Y. S. Jun and S. Y. Yang, “A Study on 3D Simulation Platform of Excavator with Tilt Rotator” , 2019 Spring Conference on Driv and Control, Jeju, Korea, pp.248-249, 2019.

- [13] Y. J. Kim, T. S. Kim, T. G. Son and S. Y. Yang, “A Study on 3D Simulator Development of 6-axis Excavator” , 2019 Autumn Conference on Drive and Control, Cheonan, Korea, pp.87-88, 2019.
- [14] Päckilä, S., “Modeling and Simulation of a six degrees of Freedom Excavator” , Tampere University of technology Master of Science Thesis, 2017.
- [15] D. J. Kim, J. S. Jang, H. I. Won, J. W. Cho, J. Y. Oh and C. H. Song, “Development of a Tilt-rotator Multi-body Dynamics Model for an Excavator” , 2020 Autumn Conference on Drive and Control, Online, pp.109-110, 2020.
- [16] D. J. Kim, K. B. Kwon, S. S. Kweon, C. H. Song, “Operational Deflection Shape Analysis of Tilt-rotator System considering Excavator Working Mode” , 2020 Autumn Conference on Drive and Control, Online, pp.114-115, 2020.
- [17] Y. B. Kim, J. H. Kim, “Optimal Positioning for Grading Work with a 6 DOF Excavator” , KSAE 2018 Annual Autumn Conference & Exhibition, Jeongseon, Korea, pp.1275-1275, 2018.
- [18] Kurinov, I., Orzechowski, G., Hämäläinen, P., Mikkola, A., “Automated Excavator Based on Reinforcement Learning and Multibody System Dynamics” , Journal of IEEE Access, Vol. 8, pp.213998-214006, 2020.
- [19] S. W. Choi et al, A Study on Construction of Control System for Wireless Remote Control of Small Field Robot, Journal of Drive and Control, Vol.17, No.4, pp.103-112, 2020.
- [20] Richard S. Sutton, Andrew G. Barto, Reinforcement Learning 2nd edition, The MIT Press, 2018.
- [21] Andrea Lanza, Reinforcement Learning Algorithms with Python, Packt Publishing, 2019.
- [22] Bellman, R. E., “Dynamic Programming” , Princeton University Press, Princeton. 1957.

- [23] Bellman, R. E., “A Markovian Decision Process” , Journal of Mathematics and Mechanics, Vol. 6, No. 5, pp. 679-684, 1957.
- [24] Howard, R. A., “Dynamic Programming and Markov Processes” , MIT Press, Cambridge, MA, 1960.
- [25] Minsky, M. L., “Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain-Model Problem” , Ph.D. thesis, Princeton University, 1954.
- [26] Samuel, A. L., “Some studies in machine learning using the game of checkers” , IBM Journal on Research and Development, Vol.3, No.3, pp.210-229, 1959.
- [27] Shannon, C. E., “Programming a computer for playing chess” , Philosophical Magazine and Journal of Science, Vol.41, No.314, pp.256-275, 1950.
- [28] Minsky, M. L., “Steps toward artificial intelligence” , Proceedings of the Institute of Radio Engineers, Vol.49, pp.8-30, 1961. (Reprinted in E. A. Feigenbaum and J. Feldman (Eds.), “Computers and Thought” , McGraw-Hill, New York, pp. 406-450, 1963.)
- [29] Sutton, R. S., “Learning theory support for a single channel theory of the brain” , Unpublished report, 1978.
- [30] Sutton, R. S., “Single channel theory: A neuronal theory of learning” , Brain Theory Newsletter, Vol.4, pp.72-75, 1978.
- [31] Sutton, R. S., “A unified theory of expectation in classical and instrumental conditioning” , Bachelors thesis, Stanford University, 1978.
- [32] Sutton, R. S., Barto, A. G., “Toward a modern theory of adaptive networks: Expectation and prediction” , Psychological Review, Vol.88 No.2, pp.135-170. 1981.

- [33] Barto, A. G., Sutton, R. S., “Simulation of anticipatory responses in classical conditioning by a neuron-like adaptive element” , Behavioural Brain Research, Vol.4, No.3, pp.221-235. 1982.
- [34] Sutton, R. S., “Temporal Credit Assignment in Reinforcement Learning” , Ph.D. thesis, University of Massachusetts, Amherst, 1984.
- [35] Sutton, R. S., “Learning to predict by the method of temporal differences” , Machine Learning, Vol. 3, No.1, pp.9-44, 1988.
- [36] Watkins, C. J. C. H., Dayan, P., “Q-learning” , Machine Learning, Vol.8, No.3-4, pp.279-292. 1992
- [37] Werbos, P. J., “Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research” , IEEE Transactions on Systems, Man and Cybernetics “, Vol.17, No.1, pp.7-20, 1987.
- [38] Mnih, V., et al., “Human-level control through deep reinforcement learning. nature” , Vol.518, No.7540, pp.529-533, 2015.
- [39] Silver, D., et al., “Mastering the game of Go with deep neural networks and tree search” , nature, Vol.529, No.7587, pp.484-489, 2016.
- [40] LILLICRAP, Timothy P., et al., “Continuous control with deep reinforcement learning” , arXiv preprint arXiv:1509.02971, 2015.
- [41] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P., “Trust region policy optimization” , In International conference on machine learning, pp. 1889-1897, PMLR, 2015.
- [42] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O., “Proximal policy optimization algorithms” , arXiv preprint arXiv:1707.06347, 2017.
- [43] OpenAI Spinning Up, <https://spinningup.openai.com>

- [44] Bertsekas, D. P., “Dynamic Programming and Optimal Control,” Athena Scientific, 2nd edition, 2000.
- [45] Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P., “High-Dimensional Continuous Control Using Generalized Advantage Estimation” , 4th International Conference on Learning Representations, ICLR, 2016.
- [46] D. S. Ahn, “Integrated SolidWorks & Simscape Platform for the Model-Based Control Algorithms of Robot Manipulators” , Journal of the Korean Society for Power System Engineering, Vol.18, No.4, pp.91-96, 2014.
- [47] D. S. Ahn, I. Y. Lee, H. Y. Kim, “Integrated SolidWorks & Simscape Platform for the Model-Based Control Algorithms of Hydraulic Manipulators” , 2015 Autumn Conference on Drive and Control, Busan, Korea, pp.81-86, 2015.
- [48] Dong Sang Yoo, “3D Modeling and Balancing Control of Two-link Underactuated Robots using Matlab/Simulink” , Journal of information and communication convergence engineering, Vol.17, No.4, pp.255-260, 2019.
- [49] Y. M. Jeong, C. S. Jeong, H. S. Kim, C. D. Lee, S. Y. Yang, “A Study on Hydraulic Simulation for Excavator using MATLAB/Simscape” , 2009 Autumn Conference on Drive and Control, pp.95-100, 2009.
- [50] Q. H. Le, S. Y. Yang, “Study on the Architecture of the Remote Control System for Hydraulic Excavator” , 2011 11th International Conference on Control, Automation and Systems, Ilsan, Korea pp.939-943, 2011.
- [51] Q. H. Le, S. Y. Yang, “Hydraulic System Simulation for Manipulator of Virtual Excavator using Matlab/SimHydraulic” , 2012 Spring Conference on Drive and Control, pp.165-169, 2012.
- [52] Q. H. Le, Y. M. Jeong, C. T. Nguyen, S. Y. Yang, “Control of Virtual Excavating System Base on Real-time Simulation” , 2012 12th International Conference on Control, Automation and Systems, Jeju, Korea, pp.703-707, 2012.

[53] SMP, SMP Tiltrotator Installation Manual, <https://smpparts.com/>

[54] S. W. Choi, Y. S. Kim, S. Y. Yang, “A study on the simulation model of a small excavator with tiltrotator using MATLAB/Simscape” , 2021 Spring Conference on Drive and Control, Busan, Korea, pp.96-96, 2021.

[55] S. W. Choi, K. S. Kwak, Y. S. Kim, K. K Ahn and S. Y. Yang, “A Study on the Virtual Simulation Model of an Excavator Equipped with a Tiltrotator Based on Simscape” , The 11th JFPS International Symposium on Fluid Power HAKODATE 2020 , pp. 76-80(online paper), 2021.

[56] Coumans. E., “PyBullet Quickstart Guide” , <https://pybullet.org>, 2016-2020.

APPENDIX

APPENDIX I. 논문실적

1. 학술저널

- 1) 최성웅, 김용석, 양순용, “차세대 건설기계 센서시스템에 대한 고찰”, 유공압건설기계학회 드라이브·컨트롤, Vol.15, No.3, pp.80-88, 2018 (KCI)
- 2) 최성웅, 이창돈, 양순용, “차륜형 장갑차용 액시얼 피스톤 펌프 개발을 위한 피스톤 수에 대한 시뮬레이션에 관한 연구”, 유공압건설기계학회 드라이브·컨트롤, Vol.16, No.1, pp.14-21, 2019 (KCI)
- 3) 천세영, 최성웅, 양순용, “차량 전자 제동 시스템을 위한 실시간 시뮬레이터 개발”, 유공압건설기계학회 드라이브·컨트롤, Vol.16, No.1, pp.22-28, 2019 (KCI)
- 4) 최성웅, 김용석, 양순용, “차륜형 장갑차용 가변형 사판식 피스톤 펌프 케이스의 구조해석 및 설계검증에 관한 연구”, 유공압건설기계학회 드라이브·컨트롤, Vol.16, No.2, pp.43-50, 2019 (KCI)
- 5) 천세영, 최성웅, 양순용, “전자식 차체 자세제어 장치 실시간 시뮬레이션을 위한 유압 모델 개발”, 유공압건설기계학회 드라이브·컨트롤, Vol.16, No.2, pp.36-42, 2019 (KCI)
- 6) 김용석, 최성웅, 양순용, “금속 3D 프린팅 적층제조(AM) 공정 시뮬레이션 기술에 관한 고찰(I)”, 유공압건설기계학회 드라이브·컨트롤, Vol.16, No.3, pp.42-50, 2019 (KCI)
- 7) 김용석, 최성웅, 양순용, “금속 3D 프린팅 적층제조(AM) 공정 시뮬레이션 기술에 관한 고찰(II)”, 유공압건설기계학회 드라이브·컨트롤, Vol.16, No.3, pp.51-58, 2019 (KCI)
- 8) 최성웅, 레광호안, 손태곤, 양순용, “소형 필드로봇의 무선 원격 제어를 위한 조종시스템 구축에 관한 연구”, 유공압건설기계학회 드라이브·컨트롤, Vol.17, No.4, pp.103-112, 2020 (KCI)
- 9) Seong-Woong Choi, Yong-Seok Kim, Young-Jin Yum and Soon-Yong Yang, “A Study on Strengthening Mechanical Properties of a Punch Mold for Cutting by Using an HWS Powder Material and a DED Semi-AM Method of Metal 3D Printing”, Journal of Manufacturing and Materials Processing, Vol.4, No.4, pp.1-16, 2020 (SCOPUS)
- 10) Gaoqi Zhang, Shiliang Wang, Yong-Seok Kim, Seong-Woong Choi, Young-Jin Yum and Soon-Yong Yang, “The design of a semi-additive manufacturing shape using metal 3D printing for a partially strengthened mold based on a high-alloy tool steel powder”, Journal of Mechanical Science and Technology, Vol.34, No.10, pp.4149-4159, 2020 (SCI(E))

2. 학술대회

- 1) **최성웅**, 김영재, 주동욱, 정찬세, 이상훈, 양순용, “선박기관실용 1.5톤급 소형 유압크레인의 유압시스템 설계와 시뮬레이션에 관한 연구”, 유공압건설기계학회 2018년도 춘계학술대회 논문집, pp. 98-104, **2018**
- 2) **최성웅**, 김영재, 양순용, “재난 현장의 협소 공간에서 양팔 필드로봇 개발의 타당성 조사에 관한 연구”, 유공압건설기계학회 2018년도 추계학술대회 논문집, pp. 91-93, **2018**
- 3) **최성웅**, 레광환, 전용수, 양순용, “재난대응 특수목적기계 양팔 작업기의 관절별 유압 시뮬레이션에 관한 연구”, 유공압건설기계학회 2019년도 춘계학술대회 논문집, pp. 218-219, **2019**
- 4) 김태성, 김영재, 나선준, 손태곤, **최성웅**, 양순용, “장애인 도우미 차량용 휠체어 수납 시스템에 관한 연구”, 유공압건설기계학회 2019년도 추계학술대회, pp. 123-124, **2019**
- 5) **최성웅**, 레광호안, 김용석, 전용수, 양순용, “소형 굴착기 원격 시스템 개발을 위한 사용자 인터페이스에 관한 연구”, 유공압건설기계학회 2020년도 춘계학술대회, pp. 33-34, **2020**
- 6) 손태곤, **최성웅**, 레광호안, 김경동, 장고기, 양순용, “원격 거리 측정 시스템 모듈 개발에 대한 연구”, 유공압건설기계학회 2020년도 추계학술대회, pp. 165-165, **2020**
- 7) **최성웅**, 김용석, 양순용, “MATLAB/Simscape를 이용한 틸트로테이터 부착 소형 굴착기 시뮬레이션 모델에 관한 연구”, 유공압건설기계학회 2021년도 춘계학술대회, pp. 96-96, **2021**
- 8) **Seongwoong Choi**, Dongwook Joo, Chanse Jeong, Sanghoon Lee and Soonyong Yang, “A Study on Hydraulic System Simulation of Small Hydraulic Crane for Marine Engine Room”, 22nd International Conference on Mechatronics Technology(ICMT2018), pp. 37-37, **2018**
- 9) Quang Hoan Le, **Seongwoong Choi**, Youngjae Kim, Sungwon Choi, Dakarimov Sayat and Soonyong Yang, “Design and Simulation of Hydraulic System for Dual Arm Excavator in Disaster Environment”, 22nd International Conference on Mechatronics Technology(ICMT2018), pp. 28-28, **2018**
- 10) **Seongwoong Choi**, Quang Hoan Le and Soonyong Yang, “A Study on Hydraulic Simulation Analysis of a 7 DOF Dual Arm Machinery”, 23rd International Conference on Mechatronics Technology(ICMT2019), pp. 1-1, **2019**
- 11) **Seongwoong Choi**, Kyungsin Kwak, Yongseok Kim, Kyoungkwon Ahn and Soonyong Yang, “A Study on the Virtual Simulation Model of an Excavator Equipped with a Tiltrotator Based on Simscape”, The 11th JFPS International Symposium on Fluid Power HAKODATE 2020 , pp. 76-80(online paper), **2021**

- 12) **최성웅**, 김태운, 정영만, 김용석, 양순용, 장재홍, “차량용 자전거 캐리어 개발에 관한 연구”, 한국자동차공학회 부산·울산·경남지부 추계학술대회 논문집, pp. 85~85, 2015
- 13) 김인호, **최성웅**, 양순용, “Diesel Plug-In 하이브리드 버스 시스템에 관한 연구”, 한국자동차공학회 부산·울산·경남지부 추계학술대회 논문집, pp. 45~45, 2015
- 14) Quang Hoan Le, **Seong Woong Choi**, Tae Un Kim, Chi Thanh Nguyen, Jae Woo Lee, Soon Yong Yang, “Design of Soil-Pile Scanning System Using the Laser Rangefinder”, 유공압건설기계학회 학술대회논문집, pp. 67~70, 2016
- 15) **최성웅**, 김용석, 장재홍, 양순용, “자동 업로드 차량용 자전거 루프 캐리어 시스템에 대한 연구”, 유공압건설기계학회 학술대회논문집, pp. 27~32, 2016
- 16) 김인호, **최성웅**, 오지우, 양순용, “플러그인 하이브리드 버스 HCU 개발”, 유공압건설기계학회 학술대회논문집, pp. 43~52, 2016
- 17) Quang Hoan Le, **Seong Woong Choi**, Tae Un Kim, Chi Thanh Nguyen, Jae Woo Lee, Soon Yong Yang, “A Study on Trajectory Tracking Control of Field Robot”, 유공압건설기계학회 학술대회논문집, pp. 119~123, 2016
- 18) Quang Hoan Le, **Seong Woong Choi**, Tae Un Kim, Soon Yong Yang, “A Study on Intuitive Configuration of Joystick for Operator in Flattening task of Excavator”, The 20th International Conference on Mechatronics Technology, pp. 168, 2016
- 19) 염영진, 양순용, 김용석, 황반토, 김진영, **최성웅**, 금종원, “3DP 기법을 이용한 핫스탬핑에 의한 초고강도부품 후공정을 위한 피어싱 및 트림용 적층편치 제조에 관한 연구”, 대한기계학회 춘추학술대회, pp. 307~308, 2017
- 20) Quang Hoan Le, **Seong Woong Choi**, Chi Thanh Nguyen, Soon Yong Yang, “Development of Intuitive Configuration of Joystick in Grading Task of Excavator”, 유공압건설기계학회 학술대회논문집, pp. 101~105, 2017

21) Quang Hoan Le, Chi Thanh Nguyen, **Seong Woong Choi**, Tae Un Kim, Soon Yong Yang, “Remote Control of Excavator Using Smart Observation System” , The 21th International Conference on Mechatronics Technology, pp. 372~377, 2017

APPENDIX II. 연구과제 수행실적

1	과제명	재난·재해 대응특수목적기계용 핵심요소부품 및 시스템 개발
	사업명	한국산업기술평가관리원 산업핵심기술개발사업
	연구기간	2015.07.01. ~ 2020.09.30.
	참여기관	한국생산기술연구원, 건설기계부품연구원, 수산중공업, 울산대학교
2	과제명	차륜형장갑차 가변피스톤형 유압펌프 개발
	사업명	중소벤처기업부 구매조건부신제품개발사업
	연구기간	2016.09.22. ~ 2018.09.21.
	참여기관	피엠씨, 울산대학교
3	과제명	선박 기관실용 1.5톤 지능형 Manipulator 기술 개발
	사업명	한국산업기술진흥원 지역주력산업육성사업
	연구기간	2017.06.01. ~ 2018.05.31.
	참여기관	영광공작소, 하일시스템, 한국조선해양기자재연구원, 울산대학교
4	과제명	시뮬레이션 기반 6자유도 굴삭기 고르기 자동화 제어 기술 개발
	사업명	기타사업
	연구기간	2019.03.01. ~ 2020.06.30.
	참여기관	현대건설기계(주), 울산대학교
5	과제명	미래형 듀얼암 소형 굴삭기 Hardware 플랫폼 개발
	사업명	건설기계전문인력양성사업
	연구기간	2018.07.01. ~ 2019.02.28.
	참여기관	울산대학교, ECS 프라임
6	과제명	미래형 필드로봇의 플랫폼 및 시뮬레이션 기반 제어 알고리즘 개발
	사업명	건설기계전문인력양성사업
	연구기간	2019.08.01. ~ 2020.01.31.
	참여기관	울산대학교, (주)제일피엠씨

7	과제명	3D 프린팅 기법을 이용한 자동차 초고강도부품(1500MPa) 용 컷팅 프로세스 및 핫스탬핑 금형개발 기초연구
	사업명	한국연구재단 지역신산업선도인력양성사업
	연구기간	2016.06.01. ~ 2019.05.31.
	참여기관	울산대학교, 엠디티(주)
8	과제명	세단 승용차량(2000cc급) 트렁크용 휠체어 이지-업로드 시스템 개발에 관한 연구
	사업명	LINC+ 사회맞춤형 융복합기술개발과제
	연구기간	2018.07.01. ~ 2018.12.31.
	참여기관	울산대학교, 이든모터스(주)
9	과제명	리클라이닝이 가능한 복지차량용 전동시트와 휠체어 탑재장치 개발
	사업명	중소벤처기업부 창업성장기술개발사업
	연구기간	2018.10.15. ~ 2019.12.31.
	참여기관	성우, 울산대학교
10	과제명	MC기반 소형굴착기 틸트로테이터 기초연구
	사업명	스마트 건설기계 전문인력양성사업
	연구기간	2020.05.01. ~ 2021.02.28.
	참여기관	울산대학교, (주)에이티엠
11	과제명	1.5톤 소형 굴착기 무선 원격 조종 시스템 개발
	사업명	스마트 건설기계 전문인력양성사업
	연구기간	2021.05.01. ~ 2022.02.28.
	참여기관	울산대학교, (주)에이티엠
12	과제명	안정성 향상을 위한 차량용 이지-업 자전거 루프 캐리어 RU-모델 사용화 개발
	사업명	한국산업단지공단 현장맞춤형기술개발사업
	연구기간	2015.11.01. ~ 2016.10.31.
	참여기관	주경산업, 명솔산업, 울산대학교

13	과제명	전기자동차 상위제어기(EVCU)의 발열을 고려한 케이스 최적설계
	사업명	울산대학교 기타사업 (외부민간기업 연구수탁과제)
	연구기간	2016.01.31. ~ 2016.03.31.
	참여기관	성산VCC, 울산대학교
14	과제명	자율주행자동차 경진대회 차량제작 지원
	사업명	울산대학교 기타사업 (외부민간기업 연구수탁과제)
	연구기간	2016.01.26. ~ 2017.01.31.
	참여기관	현대엔지비(주), 울산대학교
15	과제명	자동차용 범용 제어기 개발을 위한 열 해석
	사업명	울산대학교 기타사업 (외부민간기업 연구수탁과제)
	연구기간	2016.10.05. ~ 2016.12.04.
	참여기관	성산VCC, 울산대학교
16	과제명	직교로봇용 2축(회전-틸팅) 증설 유닛 유효자세 분석
	사업명	울산대학교 기타사업 (외부민간기업 연구수탁과제)
	연구기간	2016.12.19. ~ 2017.02.18.
	참여기관	모던테크, 울산대학교
17	과제명	자동차 배기열을 이용한 원전비상사태 대비용 200bar 봉산수 공급기술 개발
	사업명	한국산업기술진흥원 지역주력산업육성사업
	연구기간	2017.03.01. ~ 2018.02.28.
	참여기관	하일시스템, 울산대학교
18	과제명	3D 프린터 검증을 위한 적층 챔버 유동해석
	사업명	울산대학교 기타사업 (외부민간기업 연구수탁과제)
	연구기간	2018.12.03. ~ 2018.12.31.
	참여기관	원포시스(주), 울산대학교

APPENDIX III. 지식재산권 실적

1	구분	특허 출원
	명칭	원격조종 굴삭기 모니터링 시스템 및 그 시스템을 이용한 모니터링 방법
	번호	출원번호 10-2016-0066498
	출원인/발명인	울산대학교 산학협력단 / 양순용, 레광호안, 최성웅
2	구분	특허 출원
	명칭	굴삭기용 조명장치 및 그 제어방법
	번호	출원번호 10-2016-0080744
	출원인/발명인	울산대학교 산학협력단 / 양순용, 레광호안, 최성웅
3	구분	특허 등록
	명칭	차량용 휠체어 수납 시스템
	번호	출원번호 10-2019-0002149 / 등록번호 10-2124539
	출원인/발명인	울산대학교 산학협력단 / 양순용, 김용석, 최성웅, 김영재
4	구분	특허 등록
	명칭	휠체어 수납장치
	번호	출원번호 10-2019-0110903 / 등록번호 10-2267843
	출원인/발명인	울산대학교 산학협력단 / 양순용, 김용석, 최성웅, 김영재
5	구분	특허 출원 진행중
	명칭	미니 소형 굴삭기용 틸트-로테이터 장치 / 유니버설 틸트 장치
	번호	특허 출원 진행중
	출원인/발명인	울산대학교 산학협력단 / 양순용, 김용석, 최성웅, 곽경신

A Study on Building a Simulation Model of a Smart Field Robot and Learning the Work Path using Deep Reinforcement Learning

Seong-Woong Choi

Department of Construction Machinery Engineering
Graduate School, University of Ulsan

Abstract

Field robots are being used in various industries such as agriculture, forestry, and manufacturing, as well as in the construction industry, and their scope is expanding to the subsea area. The field robot referred to here is a robot that works in the field rather than in a factory. Representative field robots include excavators, wheel loaders, and forklifts. In this paper, excavators are expressed as field robots.

Representative works of field robots include excavation work, leveling work, and demolition work. However, when working in a narrow work space, due to the limitation of the range of motion, the number of work hours and work time increase due to unnecessary motion, and the work efficiency is lowered. Therefore, it is necessary to increase the degree of freedom of the field robot to increase work efficiency and work convenience. Likewise, a new mechanism such as tilting and rotation of the bucket to replace the micro-turn is needed. By applying the tiltrotator, a new mechanism to increase work efficiency, work efficiency is increased by reducing work time.

Recently, in the construction field, research on core technologies for realizing smart construction is being actively conducted. As a field of smart construction machinery, we are conducting research on intelligent construction equipment that combines artificial intelligence and AI with construction equipment. These studies

are being used to predict the characteristics that occur while performing real field robot tasks using a field robot model in a virtual space.

Therefore, in this paper, a simulation model is built for a 6DOF field robot to which a tilt rotator is applied for a 1.5-ton small excavator, a field robot, and a simulation to check and analyze its motion characteristics and a kinematic model of a field robot and a reinforcement learning algorithm are used. The following conclusions were drawn by proposing a deep reinforcement learning model and conducting basic research on path learning that simulated the working path scenario for the flattening task that is often performed in real field robots.

1. Research on simulation models for building simulators of various field robots
 - Presenting 6DOF field robot system with tiltrotator applied
 - Set the reference coordinate system for the 6DOF field robot system, present mathematical models of forward kinematics and inverse kinematics, and verify validity
 - 3D model, hydraulic model, multibody model (mechanism/dynamics model), and control model construction by checking the specifications and design specifications for the 6DOF field robot system
 - Confirm that the output value behaves similarly with respect to the input value through three simulations of sine waveform, single operation and complex operation for excavation operation

2. Basic study of deep reinforcement learning model for driver assistance machine guidance
 - Establishment of basic concepts such as reinforcement learning, a part of artificial intelligence and AI, and definitions and theories of algorithms
 - Construction of a work path learning model with URDF model of field robot, GYM environment model, and PPO algorithm for application to reinforcement learning engine
 - As a result of learning the work path of the flattening work, it is confirmed that the position of the end of the bucket is learned and operated within the set error range of 0.1 m
 - Apply the learned results to the simulation model to check the operation results