creative commons
COMMONS DEED

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

dCollection

공학박사 학위논문

다변형 포장에서의 무인 트랙터 경운작업용 경로생성 및 강인 추종 제어기 개발

Development of Path Generation and Robust Tracking Controller for Unmanned Tillage Tractor in Polygonal Fields

2017년 2월

서울대학교 대학원

바이오시스템·소재학부 바이오시스템공학전공

한 웅 철

다변형 포장에서의 무인 트랙터 경운작업용 경로생성 및 강인 추종 제어기 개발

Development of Path Generation and Robust Tracking Controller for Unmanned Tillage Tractor in Polygonal Fields

지도교수 김 학 진

이 논문을 공학박사 학위논문으로 제출함
2017년 2월

서울대학교 대학원
바이오시스템·소재학부 바이오시스템공학전공
한 웅 철

한웅철의 공학박사 학위논문을 인준함
2017년 2월

위 원 장 조 성 인 (인)

부위원장 김 학 진 (인)

위 원 이 중 용 (인)

위 원 박 상 덕 (인)

위 원 문 희 창 (인)

다변형 포장에서의 무인 트랙터 경운작업용 경로생성 및 강인 추종 제어기 개발

한웅철

국문 초록

최근 농기계가 대형화 되면서 작업 성능은 과거에 비해 크게 높아졌지만 국내의 영농조건은 대부분 1 ha 이내의 좁은 면적의 포장에서 파종, 경운, 방제, 파종 등 다양한 농작업에서 사용되면서 작업의 정도도 일정 경로를 따라 겹치지 않고 적정 폭으로 운전하고 새머리 선회를 정밀하게 수행하는 농기계 작업기술이 필요하다. 특히 무인 농작업 기계기술을 고려했을 때 좁은 포장에서 포장 형상이 직교형을 포함한 비정형 형태에서 자동 농작업과 선회가 가능한 경로생성과 추종기술 개발은 중요하다. 반면에 국외의 상용 자동 주행제어시스템은 직진과 곡선 자동 주행 기능만 제공되고 새머리에서의 농기계 자동선회 기술은 가능하지 않기 때문에 국내 적용이 어려운 기술수준이다.

본 연구에서는 농업용 트랙터에 무인자율주행을 위한 각종 제어 및 조작 장치를 탑재하고 다양한 형태의 포장형상에서 무인 자율로 농작업이 가능하도록 작업경로 자동 생성과 추종 제어 시스템을 개발하고자 하였다. 이를 위해 직교형, 사다리형, “L”자형 등 다양한 형상의 포장 내에서 경운작업을 수행하고 자동선회가 가능한 경로생성 알고리즘과 주행 중 발생하는 노면의 슬립정도를 측정하면서 작업경로를 최적으로 추종하는 강인 조향제어기술을 개발하였으며, 가상현실에서 무인 농작업이 가능한 시뮬레이터를 이용하여 개발한 알고

리즘을 검증하였으며 최종적으로는 개발 기술을 실제 농용트랙터에 탑재하여 포장에서 무인 경운작업을 수행하여 작업 성능과 효율 등을 숙련된 작업자가 수행한 결과와 비교 개발기술의 효용성을 평가하였다.

농지형태 조건에 입력에 따른 자동 경로생성 알고리즘은 포장의 좌표와 트랙터 및 작업기의 주요치수를 입력하게 되면 작업경로와 트랙터가 주행하는 경로점인 웨이포인트(Waypoint)를 자동으로 생성하는 사용자 편이형 프로그램 형태로 개발하였다. 이를 통해 경로점 표시와 좌표 데이터 생성은 물론 예상 작업시간, 작업거리, 유효 작업시간, 유효 작업거리 및 포장 작업효율 등을 사용자가 사전에 알 수 있는 정보를 제공하도록 하였다. 정밀 추종에 영향을 주는 노면의 상태에 따른 측면 슬립 각은 GPS/INS 시스템을 이용하여 얻은 횡 편차 값과 방향각 오차정보를 기반으로 실시간으로 측정하고 이를 이용하여 강인 제어기 설계에 반영하였다. 무인 가상 농작업이 가능한 컴퓨터 시뮬레이터는 확장 기구모델을 기반으로 측면 슬립현상이 반영된 주행모델을 RoboticsLab 프로그램을 이용하여 개발하였으며 경사지에 따른 트랙터의 자세변화에 따른 위치 오차현상과 GPS 장착위치와 작업기 상/하강 작동에 따른 농작업 미경지 현상을 시뮬레이션 환경에 구현하였다. 이를 이용하여 개발한 3차원 위치보정기술과 작업기 제어알고리즘의 성능을 포장시험 전에 사전 검증하여 개발 시뮬레이터의 효용성을 확인하였다. 개발한 경로생성과 추종알고리즘의 실제 포장작업 적용성을 확인하기 위하여 26마력급의 소형 트랙터에 실시간 제어기에 주행 알고리즘을 탑재하였으며 제어명령에 따른 주행부와 작업기의 자동 작동을 위해 관련 작동기를 설계 제작하였다.

자율주행 쟁기와 로터리 포장 시험결과 개발한 동적 경로추종 알고리즘의 RMS 주행 위치 오차는 10 cm 이내로 기존 Look-ahead 추종방법과 사람 직접 주행하는 것 보다 우수한 결과를 나타내는 것으로 얻었다. 제안한 미경지 영역 방지를 위한 작업기 상승/하강 제어 시험을 통해 작업기 제어 성능을 평

가한 결과 트랙터가 새머리 영역으로 진입 및 새머리 영역에서 탈출 시 평균 RMS 위치 오차가 모두 15 cm 이내의 성능을 나타내는 것으로 확인하였다. 80 m×30 m의 직교형태 포장에서 쟁기작업을 수행한 결과 포장 작업시간은 48.6 min, 포장 작업효율은 79.9%, 포장 작업 면적의 비율은 90.3%로 관행적으로 전문 작업가가 수행하는 작업수준을 나타내었다. 또한, 다변형태인 사다리 형태와 “L”자 형태의 포장에서 진행한 결과 각 포장에서 14.9 cm과 21.8 cm의 RMS 위치 오차로 주행이 가능하였다. 사다리 형태의 포장에서는 자율주행 트랙터의 포장 작업효율은 62.1%, 포장 작업 면적비율은 91.1%로 나타났다. 반면에 “L”자 형태의 포장에서는 자율주행 트랙터의 포장 작업효율은 65.5%, 포장 작업 면적비율은 91.4%로 나타나 숙련된 작업자가 하는 작업시간 수준과 미경지 발생정도를 나타내어 개발한 경로생성과 추종기술은 자율주행 트랙터를 이용한 경운작업에 적용이 가능한 것으로 나타났다. 향후 보완 연구로서 자율주행 트랙터를 이용하여 경운 작업 시 토양조건으로 인하여 과도한 부하가 발생 시 작업기를 정밀 상/하강 조작하거나 트랙터의 주행속도를 제어하는 등 트랙터와 작업기의 강인제어기술 개발이 요구되며 경사지 조건의 밭 포장에서 작업을 수행 개발 기술의 경사지 무인 농작업 적용성을 구명하는 것이 필요할 것으로 판단된다.

주요어: 자율주행, 다변형 경로생성, 강인 경로추종, 가상 시뮬레이터, 포장 작업효율
학 번: 2013-31315

목 차

국문초록 ····· i

List of Figures ····· viii

List of Tables ····· xviii

1. 서론 ····· 1

1.1. 연구 배경 ····· 1

1.2. 문헌 연구 ····· 4

1.2.1. 경로생성 알고리즘 ····· 4

1.2.2. 경로추종 및 제어 알고리즘 ····· 10

1.2.3. 기술 개발 현황 ····· 15

1.3. 연구목적 ····· 18

2. 비정형 농지형태에 적합한 경운작업 경로생성 알고리즘 개발 ····· 19

2.1. 재료 및 방법 ····· 19

2.1.1. 개요 ····· 19

2.1.2. 주행 경로생성을 위한 좌표계 변환 ····· 20

2.1.3. Extended RDDF 구성 ····· 26

2.1.4. 자율주행 트랙터의 주행 경로생성 시스템 구성 ····· 29

2.1.5. MBR법을 이용한 직진주행 경로생성 ····· 32

2.1.6. 주행 경로생성을 위한 다변 형태 포장 분할 알고리즘 ····· 33

2.1.7. 무인 농작업 선회 경로유형 분석 및 수학적 모델링 ····· 39
2.1.8. 다변형태 포장에서 새머리의 선회경로 및 Coverage cost ····· 46
2.2. 결과 및 고찰 ····· 51
2.2.1. 경로생성 프로그램 개발 결과 ····· 51
2.2.2. 포장 면적 변화에 따른 경로생성 ····· 55
2.2.3. 다변형태 포장에서 주행 진행 각도 변화에 따른 경로생성 ····· 58
2.3. 요약 및 결론 ····· 69

3. 시뮬레이션을 이용한 동적 경로추종 기술 개발 ····· 71
3.1. 강인 제어 기반의 동적 경로추종 기술 ····· 71
3.1.1. 재료 및 방법 ····· 71
3.1.1.1. 개요 ····· 71
3.1.1.2. 자율주행을 위한 위치인지 시스템 ····· 72
3.1.1.3. 동적 경로추종 제어기 설계 ····· 74
3.1.1.4. 관측기를 이용한 슬라이딩 예측 알고리즘 ····· 79
3.1.1.5. 경사지면에서 위한 GPS 신호 보정 ····· 81
3.1.1.6. 미경지 영역 방지를 위한 작업기 제어 ····· 82
3.1.2. 자율주행 성능평가 방법 ····· 87
3.1.2.1. 주행 위치 오차 분석 ····· 87
3.1.2.2. 미 작업 영역 분석 ····· 87
3.2. 무인 농작업 시뮬레이터 개발 및 활용 기술 ····· 88
3.2.1. 재료 및 방법 ····· 88
3.2.1.1. 개요 ····· 88

3.2.1.2. 가상 트랙터 주행 모델 ····· 89
3.2.1.3. 3D 환경 경사지 ····· 98
3.2.1.4. 3D 지형에서 작업 경로 및 영역 표현 ····· 101
3.2.1.5. 서버와 클라이언트간의 통신 프로토콜 ····· 103
3.2.1.6. 시뮬레이터 활용 기술 ····· 109
3.3. 결과 및 고찰 ····· 114
3.3.1. 선회 패턴 변화에 따른 경로생성 성능 평가 ····· 114
3.3.2. 자율주행 동적 경로추종 알고리즘 성능 평가 ····· 116
3.3.3. 경사지면에서 경로추종을 위한 GPS 신호 보정 평가 ····· 122
3.3.4. 미경지 영역 방지를 위한 작업기 제어 평가 ····· 125
3.4. 요약 및 결론 ····· 130

4. 자율주행 트랙터 통합 시스템 개발 및 성능평가 ····· 132
4.1. 재료 및 방법 ····· 132
4.1.1. 개요 ····· 132
4.1.2. 자율주행 트랙터 시스템 개발 ····· 132
4.1.3. 항법 시스템 개발 ····· 146
4.1.4. 경로계획 및 추종 통합 제어기 개발 ····· 150
4.1.5. 경운성능 시험 평가용 자율주행 트랙터 시스템 구성 ····· 155
4.1.6. 시험 평가를 위한 계측 시스템 구성 ····· 157
4.2. 결과 및 고찰 ····· 162
4.2.1. 동적 경로추종 알고리즘 성능 요인실험 결과 ····· 162
4.2.2. 쟁기/로터리 자율 농작업 요인 시험 결과 ····· 165

4.2.3. 경사지면에서 경로추종을 위한 GPS 신호 보정 시험 평가 ····· 173
4.2.4. 미경지 영역 방지를 위한 작업기 제어 시험 평가 ····· 175
4.2.5. 직교형태 포장에서 자율주행 쟁기 작업 성능 평가 ····· 180
4.2.6. 다변형태 포장에서 자율주행 로터리 작업 성능 평가 ····· 194
4.3. 요약 및 결론 ····· 204

5. 종합 요약 및 결론 ····· 206

6. 참고 문헌 ····· 210

7. 부록 및 기타 ····· 222
7.1. 가상 시뮬레이터의 트랙터 주행 모델 코딩 ····· 222
7.2. 가상 시뮬레이터의 3D 지형 환경 구성 코딩 ····· 234
7.3. 가상 시뮬레이터의 작업경로 및 영역 표현 코딩 ····· 239

Abstract ····· 243

List of Figures

Fig. 2-1 Three-dimensional coordinates system ····· 20
Fig. 2-2 Process of coordinates conversion from WGS84 to TM ····· 22
Fig. 2-3 Geodetic and cartesian coordinates ····· 23
Fig. 2-4 Configuration of a RDDF format ····· 26
Fig. 2-5 Concept of LBO used in auto-guidance ····· 27
Fig. 2-6 Definition of working speed and implement status in the working and headland areas ····· 27
Fig. 2-7 Representation of a typical field structure in a polygonal field ···· 29
Fig. 2-8 Flowchart of coverage path algorithm in the polygonal field ····· 30
Fig. 2-9 Creation of headland passes in the polygonal field ····· 31
Fig. 2-10 A MBR method in the polygonal field ····· 32
Fig. 2-11 Slice movement from left to right (a), boundary creation by critical point and intersection points (b) ····· 34
Fig. 2-12 Cell creation for a concave shaped field ····· 37
Fig. 2-13 Flowchart for field decomposition ····· 38
Fig. 2-14 Path generation in the sub field or a whole field ····· 38
Fig. 2-15 Headland turning patterns in a paddy field ····· 40
Fig. 2-16 Separation of working area in the test field ····· 41
Fig. 2-17 Diagram of C-type turn on the straight lines and circular arcs 41
Fig. 2-18 Diagram of X-type turn on the straight lines and circular arcs 44
Fig. 2-19 Diagram of R-type turn on the straight lines and circular arcs 45
Fig. 2-20 Headland turning in a polygonal field ····· 46

Fig. 2-21 Configuration of headland turning in the polygonal field ····· 48
Fig. 2-22 R-type path turning in headland in the polygonal field ····· 48
Fig. 2-23 X-type path turning in headland in the polygonal field ····· 50
Fig. 2-24 Flowchart of the path generation program ····· 51
Fig. 2-25 Interface of the path generation program ····· 52
Fig. 2-26 Field boundary generation by field variable input ····· 53
Fig. 2-27 Input variables of tractor and implement ····· 54
Fig. 2-28 Input variables of waypoint setting and output of path analysis 54
Fig. 2-29 Result of path generation in the standard size field ····· 57
Fig. 2-30 Comparison of the field efficiency by changing the field size 58
Fig. 2-31 Convex shaped polygonal field for coverage path generation ····· 59
Fig. 2-32 Result of the path generation by changing the direction in convex shaped polygonal field ····· 60
Fig. 2-33 Simulation analysis by changing the direction in convex shaped polygonal field ····· 61
Fig. 2-34 Concave shaped polygonal field 1 for coverage path generation 63
Fig. 2-35 Result of the path generation by changing the direction in concave shaped field 1 ····· 64
Fig. 2-36 Simulation analysis by changing the direction in concave shaped polygonal field 1 ····· 65
Fig. 2-37 Concave shaped polygonal field 2 for coverage path generation 66
Fig. 2-38 Result of the path generation by changing the direction in concave shaped field 2 ····· 67
Fig. 2-39 Simulation analysis by changing the direction in concave shaped polygonal field 2 ····· 68

Fig. 3-1 Flowchart of high level control for path tracking ····· 71
Fig. 3-2 Configuration of the RTK GPS ····· 73
Fig. 3-3 Configuration of the VRS-RTK GPS ····· 74
Fig. 3-4 State a2 can converge to zero by choosing gain ····· 75
Fig. 3-5 Extended Ackerman Kinematic Model ····· 76
Fig. 3-6 Flowchart of robust control and slip observer loops ····· 79
Fig. 3-7 Position corrections of vehicle inclinations ····· 81
Fig. 3-8 Commands of implement control on the RDDF ····· 82
Fig. 3-9 Flowchart of the implement control ····· 83
Fig. 3-10 Kinematic bicycle model ····· 84
Fig. 3-11 Definition of parameters for implement control ····· 85
Fig. 3-12 Definition of sum of track errors ····· 87
Fig. 3-13 Method of calculating effective operation area in ArcGIS ····· 88
Fig. 3-14 Configuration of the developed virtual simulator system ····· 89
Fig. 3-15 Extended kinematics bicycle model ····· 90
Fig. 3-16 Definition of the slip angle for front and rear tire ····· 91
Fig. 3-17 Position of the extended kinematic model in field coordinates ··· 93
Fig. 3-18 Flowchart of determining slip angle and yaw rate ····· 94
Fig. 3-19 3D tractor model used in the study ····· 95
Fig. 3-20 Diagram views (tractor) for simulating engine ····· 95
Fig. 3-21 Navigation sensors in the virtual simulator ····· 96
Fig. 3-22 Diagram views (tractor & implements) for simulating engine ··· 96
Fig. 3-23 Various implements for farm work in simulator environment ···· 98
Fig. 3-24 3D map generation using Height Map Gen.exe ····· 99
Fig. 3-25 3D map generation using bit-map ····· 100

Fig. 3-26 Bit-map generation(a), 3D map generation in RoboticsLab(b) ·· 101
Fig. 3-27 Waypoint generation on the 3D plane ····· 102
Fig. 3-28 Coverage area generation in the simulator environment ····· 102
Fig. 3-29 Comparison of the coverage areas based on different driving path ····· 111
Fig. 3-30 Evaluation test in the presence of sliding effect ····· 112
Fig. 3-31 Environment editor for changing the road condition ····· 113
Fig. 3-32 Result of simulation analysis for path generation by C-type pattern ····· 115
Fig. 3-33 Result of simulation analysis for path generation by X-type pattern ····· 115
Fig. 3-34 Result of simulation analysis for path generation by R-type pattern ····· 116
Fig. 3-35 Result of simulation analysis by C-type, X-type and R-type pattern path ····· 116
Fig. 3-36 No slip environment on the developed simulator ····· 117
Fig. 3-37 Result of the simulation analysis of full path tracking trajectory by look-ahead (a) and robust control (b) from the no slip environment · 118
Fig. 3-38 Large slip environment on the developed simulator ····· 119
Fig. 3-39 Result of the simulation analysis of full path tracking trajectory by look-ahead (a) and robust control (b) from the large slip environment ····· 120
Fig. 3-40 Result of the simulation analysis of non working area by look-ahead (a) and robust control (b) from the large slip environment ··· 121
Fig. 3-41 Result of the simulation analysis of tracking performance by the

GPS uncorrected (a) and the corrected (b) on the left and right tilted plane ····· 123
Fig. 3-42 (a) Result of the simulation analysis of tracking trajectory and lateral deviation on the left tilted plane ····· 124
Fig. 3-42 (b) Result of the simulation analysis of tracking trajectory and lateral deviation on the right tilted plane ····· 125
Fig. 3-43 Result of the simulation analysis of implement control by no algorithm applied on the R-type(a) and X-type(b) pattern path ····· 127
Fig. 3-44 Result of the simulation analysis of implement control by the developed algorithm application on the R-type(a) and X-type(b) pattern path ····· 128
Fig. 3-45 Result of the simulation comparison of RMSE by no algorithm and the developed algorithm on the R-type pattern path ····· 129
Fig. 3-46 Result of the simulation comparison of RMSE by no algorithm and the developed algorithm on the X-type pattern path ····· 129
Fig. 4-1 A view of TS25 tractor used in the study ····· 133
Fig. 4-2 System configuration of the tractor ····· 135
Fig. 4-3 Configuration of power and signal for autonomous tractor ··· 135
Fig. 4-4 Design of electrical power steering for autonomous tractor ·· 138
Fig. 4-5 EPS module and control board ····· 138
Fig. 4-6 Encoder (a) and relationship between steering angle and output voltage (b) ····· 139
Fig. 4-7 Design of encoder for steering angle measurement ····· 140
Fig. 4-8 Smart motor for driving speed control ····· 141
Fig. 4-9 Motor reducer for driving speed control ····· 142

Fig. 4-10 Design of speed control actuator ····· 143
Fig. 4-11 Dynamixel motor for implement control ····· 143
Fig. 4-12 Design of implement control actuator ····· 144
Fig. 4-13 Encoder for speed measurement ····· 145
Fig. 4-14 Design of encoder for speed measurement ····· 146
Fig. 4-15 Configuration of a VRS-GPS system ····· 147
Fig. 4-16 Navigation system(left) and mounting position(right) ····· 148
Fig. 4-17 Schematic diagram of auto guidance system ····· 150
Fig. 4-18 Embedded controller for the low level control ····· 151
Fig. 4-19 Configuration of the auto guidance system ····· 152
Fig. 4-20 Block diagram of the steering controller ····· 152
Fig. 4-21 Block diagram of the velocity & implement controller ····· 153
Fig. 4-22 Integrated controller for autonomous tractor ····· 154
Fig. 4-23 Front panel of the high level control by LabVIEW ····· 154
Fig. 4-24 Front panel of the low level control by LabVIEW ····· 155
Fig. 4-25 A view of TX803 tractor used in the study ····· 156
Fig. 4-26 Auto-guidance system configuration of the tractor ····· 157
Fig. 4-27 3-point hitch dynamometer for draft force measurement ····· 158
Fig. 4-28 Load cell for draft force measurement ····· 158
Fig. 4-29 Flow meters installed in TX803 tractor ····· 159
Fig. 4-30 The principle of fuel consumption measuring system ····· 160
Fig. 4-31 Rotational speed sensor installed in TX803 tractor(a) and the principle of engine RPM measuring system (b) ····· 160
Fig. 4-32 Data acquisition system in the measuring tractor ····· 161
Fig. 4-33 Evaluation test on the test field by TS25 auto-guidance tractor

(a) and test field in Suwon from artificial satellite (b) ····· 163
Fig. 4-34 Result of full path tracking trajectory by look-ahead (a) and robust control (b) on the C-type pattern path (TS25 tractor) ····· 163
Fig. 4-35 Result of the position RMSE (a) & yaw rate (b) by look-ahead and robust control on the C-type pattern path (TS25 tractor) ····· 164
Fig. 4-36 Result of full path tracking trajectory by look-ahead (a) and robust control (b) on the R-type pattern path (TS25 tractor) ····· 164
Fig. 4-37 Result of the position RMSE (a) & yaw rate (b) by look-ahead and robust control on the R-type pattern path (TS25 tractor) ····· 165
Fig. 4-38 Test field in Gongju from artificial satellite (a) evaluation test on the test field with plowing operation by TX803 tractor (b) ····· 166
Fig. 4-39 Result of full path tracking trajectory by manual, look-ahead and robust control with plowing operation (TX803 tractor) ····· 167
Fig. 4-40 Result of the lateral deviation by manual, look-ahead and robust control with plowing operation (TX803 tractor) ····· 167
Fig. 4-41 Result of the yaw rate(a) and the enlarged part of curved to straight (b) by manual, look-ahead and robust control with plowing operation (TX803 tractor) ····· 168
Fig. 4-42 Evaluation test on the test field with tillage operation by TX803 tractor ····· 170
Fig. 4-43 Result of full path tracking trajectory by manual, look-ahead and robust control with tillage operation (TX803 tractor) ····· 171
Fig. 4-44 Result of lateral deviation by manual, look-ahead and robust control with tillage operation (TX803 tractor) ····· 171
Fig. 4-45 Result of yaw rate(a) and enlarged part of curved to straight (b)

by manual, look-ahead and robust control with tillage operation (TX803 tractor) ····· 172
Fig. 4-46 Test field for GPS signal correction test (a) and picture of evaluation test for GPS signal correction test (b) ····· 174
Fig. 4-47 Euler signals from IMU on the TS25 auto-guidance tractor (a) and result of GPS signal by no algorithm and correction algorithm (b) ·· 175
Fig. 4-48 Test field picture from artificial satellite (a) and evaluation test by TS25 auto-guidance tractor for implement control (b) ····· 176
Fig. 4-49 Result of implement up test by no algorithm and the developed algorithm on the field ····· 177
Fig. 4-50 Result of implement down test by no algorithm and the developed algorithm on the field ····· 177
Fig. 4-51 Result of position of implement operating point on tracking trajectory by no algorithm (a) and the developed algorithm (b) ····· 179
Fig. 4-52 Result of implement control RMSE by no algorithm and the developed algorithm ····· 179
Fig. 4-53 Simulation interface with autonomous plowing operation ····· 181
Fig. 4-54 Result of simulation analysis of full path tracking trajectory with autonomous plowing operation ····· 182
Fig. 4-55 Result of position RMSE with autonomous plowing operation by simulation ····· 183
Fig. 4-56 Test field in Suwon from artificial satellite (a) evaluation test on the test field with plowing operation by TX803 tractor(b) ····· 184
Fig. 4-57 Cone index with soil depth at different field section ····· 186
Fig. 4-58 Result of full path tracking trajectory with plowing operation in

the field test ····· 186
Fig. 4-59 Result of position RMSE with autonomous plowing operation in the field test ····· 187
Fig. 4-60 Result of heading RMSE with autonomous plowing operation in the field test ····· 188
Fig. 4-61 The most moisture part at headland A ····· 188
Fig. 4-62 Result of coverage area analysis by ArcGIS software ····· 190
Fig. 4-63 3D plot of draft force with autonomous plowing operation on field test ····· 191
Fig. 4-64 3D plot of consumption with autonomous plowing operation on field test ····· 192
Fig. 4-65 Countor plot of draft force with autonomous plowing operation in the field test ····· 193
Fig. 4-66 Countor plot of fuel consumption with autonomous plowing operation in the field test ····· 193
Fig. 4-67 Evaluation test pictures by autonomous (a) and manual (b) in the convex polygonal field ····· 195
Fig. 4-68 Result of travelling trajectory by autonomous(a) and manual(b) in the convex polygonal field ····· 196
Fig. 4-69 Implement status from the sensor values ····· 197
Fig. 4-70 Result of effective working trajectory at inside working area (a) and outside working area (b) in the convex field ····· 197
Fig. 4-71 Result of coverage area analysis by autonomous(a) and manual(b) in the convex polygonal field ····· 198
Fig. 4-72 Evaluation test picture by autonomous and manual in the

concave polygonal field ····· 199
Fig. 4-73 Result of travelling trajectory by autonomous (a) and manual (b) in the concave polygonal field ····· 200
Fig. 4-74 Result of effective working trajectory at inside working area(a) and outside working area(b) in the concave field ····· 201
Fig. 4-75 Result of coverage area analysis by autonomous(a) and manual(b) in the concave polygonal field ····· 202

List of Tables

Table 1-1 Summary of literature review on path generation method ····· 9
Table 1-2 Summary of literature review on path tracking method ····· 15
Table 2-1 Datum of TM Coordinate System ····· 21
Table 2-2 An example of RDDF format used in the autonomous system 28
Table 2-3 Input variables of tractor and implement used in the study ····· 56
Table 2-4 Results of the simulation analysis by changing the direction in convex shaped polygonal field ····· 61
Table 2-5 Results of the simulation analysis by changing the direction in concave shaped polygonal field 1 ····· 64
Table 2-6 Results of the simulation analysis by changing the direction in concave shaped polygonal field 2 ····· 68
Table 3-1 Configuration of the message header ····· 103
Table 3-2 Configuration of waypoint message packet ····· 104
Table 3-3 Configuration of velocity & steer angle message packet ····· 104
Table 3-4 Configuration of implement angle message packet ····· 105
Table 3-5 Configuration of implement up/down message packet ····· 105
Table 3-6 Configuration of implement width message packet ····· 106
Table 3-7 Configuration of working area feedback message packet ····· 106
Table 3-8 Configuration of slip feedback message packet ····· 107
Table 3-9 Configuration of lateral force feedback message packet ····· 107
Table 3-10 Configuration of GPS signal feedback message packet ····· 108
Table 3-11 Configuration of IMU signal feedback message packet ····· 108

Table 3-12 Configuration of velocity & steer angle feedback message packet ····· 109
Table 3-13 Results of the simulation comparison of full path tracking RMSE by look-ahead and robust control from the large slip environment ····· 122
Table 3-14 Results of the simulation comparison of operation performance by look-ahead and robust control from the large slip environment ····· 122
Table 4-1 Specifications of the TS25 tractor used in the study ····· 133
Table 4-2 Specifications of the obitrol used in the study ····· 136
Table 4-3 Specifications of the EPS used in the study ····· 137
Table 4-4 Specifications of the rotary potentiometer used in the study ·· 139
Table 4-5 Specifications of the smart motor used in the study ····· 141
Table 4-6 Specifications of the motor reducer used in the study ····· 142
Table 4-7 Specifications of the dynamixel motor used in the study ····· 144
Table 4-8 Specifications of the encoder used in the study ····· 145
Table 4-9 Hardware specification of navigation system ····· 148
Table 4-10 Specifications of the CompactRIO used in the study ····· 151
Table 4-11 Specifications of the TX803 tractor used in the study ····· 156
Table 4-12 Specifications of draft force load cell ····· 158
Table 4-13 Specifications of flow meter ····· 159
Table 4-14 Specifications of rotational speed sensor ····· 161
Table 4-15 Results of path tracking RMSE by manual, look-ahead and robust control with plowing operation (TX803 tractor) ····· 169
Table 4-16 Results of path tracking RMSE by manual, look-ahead and robust control with tillage operation (TX803 tractor) ····· 173

Table 4-17 Result of RMSE comparison of by no algorithm and the developed GPS signal correction algorithm ····· 175
Table 4-18 Result of error analysis of each repeated test by no algorithm and the developed algorithm ····· 180
Table 4-19 Results of the simulation comparison of operation performance with autonomous plowing operation ····· 183
Table 4-20 Soil properties and moisture contents of experimental field ··· 185
Table 4-21 Result of fuel consumption measurement with autonomous plowing operation in the field test ····· 189
Table 4-22 Results of the comparison of operation performance with autonomous plowing operation in the field test ····· 189
Table 4-23 Result of the comparison of operation performance between the autonomous and manual operation in the convex field ····· 196
Table 4-24 Result of the comparison of coverage area between the autonomous and manual operation in the convex field ····· 199
Table 4-25 Result of the comparison of operation performance between the autonomous and manual operation in the concave field ····· 201
Table 4-26 Result of the comparison of coverage area between the autonomous and manual operation in the convex field ····· 203

1. 서론

1.1. 연구 배경

농가 인구의 감소로 70마력급 이상의 중대형 농기계 보급률은 계속 증가하고 있으며, 경운, 파종, 이앙, 방제, 수확 등의 다양한 농작업에서 높은 생산성 확보가 가능하고 편리한 기계 조작이 가능한 자동화 기술에 대한 요구가 급속히 증가하고 있다. 그러나 단순히 수확량 목적으로 개발된 농업 생산방식은 적절한 비료와 농약을 사용하지 않기 때문에 농업의 지속 발전 가능성이 한계를 나타내고 있다. 이러한 문제를 해결하기 위해 농업의 많은 분야에도 로봇의 도입이 검토되고 있으며 농작업의 로봇화를 실현하기 위한 연구들이 진행되고 있다.

미래의 농업기계 기술은 미래지향적 농업여건과의 부합을 고려하였을 때 노동시간을 줄이고 편하게 농사를 지을 수 있는 경로화·쾌적화 지원, 규모화된 영농에서의 효율성 제고 및 환경 친화적인 생산 방식과 지속가능한 농업생산시스템으로 전환이 요구되고 있다. 지능화 기술로 각광받는 무인자율주행 기술은 자동차 분야에서 활발히 진행되어 왔으며 항법과 경로생성 등 무인자율주행에 필요한 요소기술이 아직까지는 완벽히 차량 스스로 주행이 가능한 무인자율주행차량은 개발되지 않았지만 이를 위한 요소기술의 발전은 빠르게 이루지고 있다. 차량의 경우 도로의 지면 마찰력이나 접촉 면적이 대부분 큰 차이가 없이 지속적이며 스스로 주행속도에 따른 안전거리를 유지하여 주행하는 조건이다. 이에 반해 경운, 파종, 방제, 수확 등 넓은 포장 내에서 농작업에 사용되는 트랙터, 콤바인 등의 농기계는 저속으로 운전하고 장애물의 문제가 비교적 큰 문제가 되지 않을 뿐만 아니라 직선 또는 완만한 곡선 경로를 따라 저속으로 움직이면서 작업이 이루어지고 이웃한 열에 일정 간격을 유지하면서 반복적으로 이루어지기 때문에 자율주행 또는 자동화 기술 적용성이

유리한 조건이다.

국내 농기계와 농작업 조건은 선진국에 비해 대형 기계보다는 중소형이 대부분이고, 농작업 조건의 경우도 미국의 경우는 옥수수, 콩과 같은 밭작물을 10 ha 이상의 대구획 작업용인데 반해 국내는 무논에서 이루어지면서 1 ha 이상의 상대적으로 작은 포장에서 기계가 사용된다. 대형 농형기계는 작업 성능은 우수하지만 파종, 경운 등의 농작업 시 일정 경로를 따라 겹치지 않고 적정 폭으로 운전하는 것이 난해한 문제가 있어 보급되고 있는 자동 주행제어 시스템을 국내 농기계시스템의 구조적 차이와 구획 크기 차이에 의한 문제로 국내 영농조건에 맞는 기술개발이 필요하다. 따라서 자율 주행 제어 시 지면과 타이어의 슬립 효과를 충분히 반영한 제어 기술이 필요하다.

한국의 경우 농업환경이 미국, 유럽등과 다르고(경지면적이 작고, 작목별 재배방식이나 재배환경이 상이) 사용하는 농업기계의 특성과 제원이 상이하므로 외국의 자율주행 제품을 국내에 바로 적용하기 어렵다. 특히, 국내 농업포장 조건에 적합한 무인자율주행을 의한 새머리(headland) 선회 알고리즘 연구가 필요하다.

작업 및 주행 경로 추종 시 대부분의 연구에서는 규격화된 농지 형태를 대상으로 실시하였으며, 경계의 형태, 고도의 차이, 기울기 등에 대한 정보는 누락되어있어 경로 추종을 위한 위치 보정 기술이 자율주행의 성능, 안전성과 연관되는 매우 중요한 사항으로 이를 향상시키기 위해 센서 혹은 시스템의 성능 향상에 초점을 둔 연구가 중요하다. 또한 현재 많은 농지가 규격화되고 있지만, 차량 분야의 일반적인 도로에 비해 편차는 매우 높으므로 다양한 농지의 환경에 대해 강인한 제어가 가능한 추종 알고리즘이 연구 개발이 되어야한다.

무인트랙터의 자율주행성능을 예측하고 분석하기 위해 실 환경에서 다양한 주행시험을 수행하여야 하지만 시험안전성과 다양한 시험환경의 인위적 제공

의 어려움 등 많은 제약이 있고 주행시험에 필요한 하드웨어를 필요로 하는 제한요소가 존재한다. 하지만, 시뮬레이션 기법을 이용하면 개발하는 경로 생성기 및 추종기의 효용성을 컴퓨터 환경에서 효율적으로 검증할 수 있으며 관련 매개변수를 최적화하는 것이 가능하다. 예를 들면, 실제 3차원 포장 환경을 시뮬레이터에서 구성하고 토양 슬립, 조향장치의 반응성 등을 쉽게 예측할 수 있으며 이를 통하여 관련 알고리즘을 쉽고 빠르게 개발할 수 있다.

개발된 경로생성과 추종 알고리즘을 트랙터 상위제어기에 탑재하기 위해서는 별도의 하드웨어 기술이 필요하다. 최근 전자트랙터의 보급에 따른 CAN통신 기반 통합제어기 적용이 가능해졌으며, 이를 기반으로 하는 자율주행을 위한 통합제어기 는 정밀한 위치를 인식하는 GPS와 연결이 되고, 자율주행 추종이 가능한 전자유압 제어장치와 운전 조건을 설정하고 모니터링하기 위한 화상제어기 설계기술 개발필요성이 제기되고 있다. 특히 화상제어기는 운전자에게 작업 상황에 대한 정보를 제공하고, 운전자로부터 농업기계를 자율제어하기 위한 각종 작업조건, 운전조건 등을 입력받는 화상 입출력 장치로서의 역할을 수행할 뿐 아니라 트랙터, 콤바인 등이 입력된 정보대로 자율 운전을 수행할 수 있도록 제어하는 두뇌역할을 하는 중요한 장치이며 경로생성과 추종 알고리즘이 탑재될 수 있는 하드웨어 플랫폼기술이다.

1.2. 문헌 연구

1.2.1. 경로생성 알고리즘

정인규 등(1999)은 트랙터의 자율주행에 필요한 장애물을 검출 및 회피 알고리즘에 대한 시뮬레이션 및 실제주행실험을 통하여 효과적인 장애물회피를 위해서는 DGPS의 정밀도가 5 cm이내가 되어야 하고, 최적의 주행을 위해서는 목표점 약 1.5 m 전방에서 그 다음 목표점을 향한 주행을 고려하는 것이 필요하다고 하였다.

정선옥 등(1999)은 무인트랙터 농작업 경로계획에서 직사각형의 포장에서 작업하는 최적 작업경로를 계획하고 최소 확장트리를 응용한 2가지 경로추종 알고리즘을 제안하였는데, 기존 관행방법에 의한 경로와 비교하여 40 m × 100 m 크기의 포장에 대하여 약 78 m의 경로를 단축시킨 것으로 나타났으며 brute-force알고리즘의 계산횟수에 대한 개발된 알고리즘의 계산 횟수비로 평가하였다.

Zhu 등(2007)은 로봇 트랙터가 경로 추적을 가능하게 하는 제어기를 설계하는 것을 목적으로 진행하였다. 경로 추적 제어기는 feedforward 와 feedback 부분으로 구성하였다. Feedforward 부분은 트랙터의 현재 위치와 주행 목표위치를 이용하여 조향각이 결정되고 feedback 부분은 최적화 설계된 제이기로부터 받는 구조이다. 경로 추적 성능을 검증위해 컴퓨터 시뮬레이션과 현장 필드 시험을 진행하였으며 직선경로에서 평균 오차와 표준편차 각각 3.1 cm 및 8.6 cm 로 나타났다.

Linker 등(2008)은 차량이 과수원에서 이용되기 위해 필요한 단순 최적화 경로를 개발하고자 하였다. 차량의 기하학적 구성과 A* 알고리즘을 적용하여 최적화 경로를 결정 하였으며 개발 알고리즘은 주변 환경에 의해 차량의 조향각과 주행 속도 결정됨을 제시해줬다. 다양한 사례연구를 통하여 최적경로를

결정하는 절차와 가중치 매개 변수를 결정한 결과를 제시하였다.

Bochits 등(2009)은 자율주행 농용 트랙터를 위해 포장 coverage 작업이 가능한 미션 플래너를 개발하였다. 자율적인 주행 미션을 완성위해 사용자 인터페이스를 통해서 XML 포맷 파일을 사용해 업로드하여 정의가 가능함을 보고하였다. 경로 계획 방법을 이용하면 전체 미작업 거리는 관행 방법보다 최대 50% 감소가 가능하다고 하였으며 3가지 다른 포장에서 시험을 진행하였으며 포장에서 왕복 작업순서를 최소화하는 기법을 제시하였다.

손영진 (2009)은 GPS, DGPS, Digital Compass, Encoder, 기타 차량의 위치정보를 획득할 수 있는 센서를 이용하여 정확하게 차량의 현재 위치좌표를 획득할 수 있는 알고리즘을 고안하고 지역경로생성 알고리즘인 Dijkstra, RRT, A*, D*를 비교하였으며 각 알고리즘을 검증하기 위하여 시뮬레이션환경도 구축하였다. 무인자율주행차량의 주행안정성을 높이는데 목적을 두고 실제 1.5 km의 주행시험을 통하여 무인자율주행차량에 대한 성능을 검증해 보았으며 최고시속 약 20 km/h의 속도로 주행 가능함을 확인하였다.

Jin 등(2010)은 좌우 교대 서법 경로를 사용하여 주어진 필드에서 커버리지 패턴을 경정하기 위해 전체 포장을 sub-region으로 분해하고 각 sub-region에 필요한 주행 방향 결정에 대해 연구를 진행하였다. 기하학적 모델에 기반으로 커버리지 경로 계획 알고리즘을 개발 하였으며 새머리 회전 시 사용할 회전 유형을 결정하였다. 개발 알고리즘 검증을 위해 단순한 정사변형 포장에서 진행하였고 장애물 있는 불규칙 다각형 포장에서 실험하였다. 제안된 알고리즘을 이용 sub-region으로 최적화 분리가 가능하였으며 주행 방향을 도출하여 효과적인 주행 경로를 생성하였다.

서일환 (2010)은 경운경로를 생성하기 위하여 포장의 작업순서를 결정하였고, 내측 왕복작업부와 외측 선화 작업부로 나누었으며, 왕복작업부의 왕복작업 경로를 C형, C-구획형, X형 및 R형 작업 4가지로 구분하여 개발하였다.

트랙터를 제어하기 위하여 PTO ON/OFF, 작업기 상승/하강, 엔진속도, 전 후진 변경, 주행단수 및 작업상태에 대한 제어명령어를 결정하였으며 이를 이용하여 시험포장에서 자율주행트랙터가 경운작업을 수행할 수 있는 데이터를 성공적으로 출력할 수 있음을 보고하였다. 경로생성 프로그램에서 생성된 주행경로를 검증하기 위하여 검증프로그램을 개발하였으며 가상시뮬레이션을 실시하여 미 경운 부분을 확인하고 이 결과를 경로생성 프로그램에 적용하여 성능을 향상시킬 수 있음을 제시하였다. 하지만, 개발한 경로생성 프로그램을 트랙터에 실제 장착하고 실험한 결과는 보고되지 않았으며 밭 포장과 같은 다변형태 포장에서 경로생성 연구는 진행하지 않았다.

Hameed 등(2011)은 유전 알고리즘을 이용하여 다변형태 포장에서 최적 경로생성에 관한 연구를 진행하였다. 첫번째 단계에서는 작업 겹쳐진 영역을 최소화 하는 것을 기반으로 주행 방향을 최적화 하였으며 두번째 단계에서는 미작업 거리를 최소화하는 주행 방향을 최적화 작업으로 진행하였다. 연구자가 제시한 sub-optimal 솔루션은 기존의 필드 작업 패턴에 비해 효율적인 것일 뿐만 아니라 실시간 농작업 구현을 위해 계산 시간도 감소함을 보고하였다.

김조환 등 (2011)은 실시간 경로 계획을 위해 전역경로 계획을 도입하여 장애물 지역과 비장애물 지역을 확인하였다. 각 노드 점을 이용하여 출발 지점과 목표 지점 사이의 경로를 생성하였으며 생성된 경로를 최적화 과정을 통해 최적 경로를 생성하였다. 연산최적화 알고리즘인 uDEAS을 이용하여 혼합다항식의 각 파라미터를 검색하여 부드러운 곡선을 생성하고 두 개의 고정 장애물 사이를 통과하는 실험을 수행한 결과 최적화된 경로 계획과 효율적인 충돌회피가 가능함을 시뮬레이션을 통하여 확인하였다.

Hameed 등(2012)은 작업시간과 비용의 측면에서 작업효율성이 높은 경로계획 방법을 연구하였다. 작업 시나리오의 평가를 위해 시뮬레이션 모델을 이용 핵심 운용 매개변수를 결정하는 것이 필수적임을 제시하였으며,시뮬레이션

모델을 사용해 CTF 시스템을 도입하는 것은 연산 시간을 약 5% 증가시킬 수 있음을 나타냈다. 또한 선택한 작업 시나리오에서 작업효율은 11.52 - 8.25% 범위로 감소하였으며 선택한 작업 시나리오에서 주행 방향의 변화에 따른 경로 생성을 통해 작업 시간을 4.5% 감소할 수 있음을 확인 하였다.

Matsuo 등(2012)은 검출 및 자동 추적 측량 장치 XNAV을 이용하여 트랙터 제어부를 개조해 자율주행 트랙터를 개발하였다. 개발한 트랙터는 소프트웨어 제어기를 통해 경로 계획이 가능하였으며 로봇 트랙터를 제어할 수 있도록 하였다. 본 연구에서 제안한 경로 생성 알고리즘은 포장의 내부 작업 영역에서 “대각선 작업” 왕복 작업경로를 생성 후 포장의 외부 작업 영역에 포장의 4개 변의 방향에 따라 “회전 작업” 경로를 생성하는 방식이다. 일정한 45도 주행경로 각도를 설정하고 대각선 작업 경로의 주행 효율은 관행 작업 경로 주행 효율보다 높음을 보고하였다.

류명진 (2013)은 자율주행 모워 트랙터 개발을 위해 잔디 포장에서 모워 트랙터를 인력으로 주행하면서 GPS 및 전자나침반 데이터를 계측하고 주행 경로 특성을 분석하였다. 포장 3 곳에서 측정된 좌표 데이터와 조향각 등을 이용하여 포장에서 총 주행, 직선 주행, 선회 주행, 중첩 주행한 거리 및 시간을 분석하였다. 주행 중 수집한 잔디를 버리기 위해 이동한 거리는 각각 35, 21, 39 m였으며, 포장 작업시간의 10% 이내에 해당함을 분석하였다. 전체 주행거리 중 중첩되어 잔디 작업의 경로는 3.5, 6.9, 1.7%인 35, 71, 26 m이었다. 본 연구에서는 경로형태를 사전에 입력하고 자동으로 경로를 생성하는 기술은 연구하지 않았다.

Hameed 등(2014)은 2D 경로 최적화 방법을 3D 경로생성방법에 적용되는 전역 경로생성에 대해 연구를 하였다. 경사지면의 필드에서 경로생성이 농기계의 에너지 소모와 관계를 가지고 있으며 가장 적은 에너지 소모 기반으로 완전한 작업을 수행하기 위해 적합한 왕복경로의 순서를 결정하는 것이 본 연

구의 주 목적이었다. 두 가지형태의 필드에서 적용한 결과를 보면 3D 지형을 고려한 경로생성 알고리즘을 사용 시 에너지 소모가 평균적으로 6.5% 감소하는 것으로 나타냈다. 또한 에너지 소모의 최소화는 연료소모와 이산화탄소의 배출과 관련 있어 농작업의 유효 작업시간을 향상하는 연구가 필요 하는 것으로 보고하였다.

Gonzalez-de-Soto 등(2015)은 GPS 위치 데이터에 따른 포장 기하정보 및 지형 높이 정보와 드론으로 찍은 잡초 위치정보를 이용하여 최소 연료 소비를 위한 최적경로 생성 알고리즘 연구를 진행하였다. 잡초 위치정보를 따라 로봇 트랙터의 출발 위치와 주행 방향을 변경하였으며 그 결과 연료소모량이 감소하는 것을 확인하였고 이산화탄소 및 아질산가스의 배출이 절감되어 공기 오염 문제를 해결할 수 있을 것으로 보고하였다.

Hameed 등(2016)은 쟁기작업 시 3D 환경으로 인해 발생하는 미작업 면적을 최소화하기 위한 경로생성 방법을 연구하였다. 현재의 3D 지형 필드에 경로 계획 알고리즘은 디지털 고도 모델 (DEM) 정보를 이용 경로를 생성하였다. 그 결과 경사지면에 따라 인접한 경로간의 거리가 증가 혹은 감소 영향을 받기 때문에 미 작업 또는 겹치는 면적이 발생하는 것을 확인하였으며 이를 개선하기 위하여 3D 커버리지 알고리즘을 이용하여 그 오차문제를 해결하였다. 또한, 적합한 주행 방향을 적용하게 되면 2-14%의 필드 공간을 더 확보하여 공간 사용 효율을 향상할 수 있음을 보고하였다.

신범수 (2016)은 전체 농경지 경작을 위한 최소 비용(최단 거리, 최단 시간 등) 경로를 구명하는 연구를 수행하였다. 경로 선정에 따라 작업기의 선회 시간이 변화를 분석하여 최적의 조합을 찾고자 하였으며 트랙터 이동 비용은 총 이동 거리 또는 총 이동 시간으로 간주하였으며 직선 주행의 속도와 곡선 주행의 속도가 다른 경우 각 속도를 고려하여 시간을 산정하여 계산하였다. 트랙터의 경로 변경을 위해 선회하는 방법은 크게 2가지를 고려하였는데 실제

트랙터의 속도를 감안하고 포장에서 주행하여 개발한 알고리즘의 효용성 검증 연구는 진행되지 않았다.

Table 1-1에 요약한 바와 같이 선행 연구에서 보고된 경로생성 관련 문헌들은 2D 혹은 3D 필드 환경에서 주행 시간과 연료소모 감소를 위해 최적화 경로생성에 대해 연구를 진행하였으나 대부분의 연구가 시뮬레이션을 통하여 그 결과를 검증하였으며 실제 트랙터와 같은 농기계에 탑재하여 포장시험을 통한 유효성을 확인하지 않았다. 또한 다변형태 포장에서 직선 경로생성을 하였고 선회는 대부분 포장 외부에서 진행하는 방식이었다. 따라서, 한국의 소구획의 직교형태 혹은 다변형태 포장에 작업효율을 향상을 위해 적합한 선회 경로를 포함된 경로생성 방법이 필요한 것으로 판단되었다.

Table 1-1 Summary of literature review on path generation methods

분류	세부 분류	기존연구	필요사항
차이점	포장형태	다변형 포장에서 직진경로만 포함, 선회경로는 포장외부에서 진행	직교형과 다변형 포장에서 모두 직진과 선회 경로 모두 포장 내부에서 진행
	선회경로	직교형태 포장에 적합한 선회 경로인 수학적 모델	직교 및 다변형 포장 경계선의 방향 인식 가능한 선회경로인 수학적 모델
	적용성	시뮬레이션 분석, 실제 자율주행 작업에 적용성 확인 없음	시뮬레이션 분석, 실제 자율주행 작업에 적용 및 분석
공통점	• MBR method 기반의 직진 경로생성 • Field Decomposition method		

1.2.2. 경로추종 및 제어 알고리즘

조성인 등(1996)은 스피드스프레이어의 무인주행을 위하여 퍼지제어기를 사용하여 자율주행 시뮬레이션을 수행하였다. 퍼지제어기의 판단 원리를 이용하여 기존 초음파 센서방식의 검출성능을 개선하고자 하는 것이 주요 내용으로 주요 스피드스프레이어의 모델은 기구학 모델을 사용하고 지면의 상태는 경사가 없는 것으로 하며 바퀴와 지면사이에서는 미끄러짐이 없는 것으로 가정 시뮬레이션 모델을 구성하였다. 그 결과 이상적인 경로와 비교하여 최대 53 cm 벗어나는 것으로 측정되었으며 이는 과수원의 주행방향 폭의 21% 이내에서 스피드스프레이어가 주행 가능함으로 보였다. 또한 스피드스프레이어의 폭에 때한 오차의 비율도 43% 이내의 성능을 나타냈다.

조성인 등(1999)은 스피드스프레이어의 자율주행을 위하여 머신 비전을 이용한 퍼지제어 기술을 개발하였다. 퍼지제어는 초음파 센서 신호를 입력으로 받으면서 유압 실린더의 작동시간을 추론 로직으로 변경하였다. CCD 카메라를 이용 신호처리 알고리즘을 개발하였고 유압시스템 기반의 제어 알고리즘을 구성하였다. 미리 결정된 목표점과 수신기로부터 받은 좌표 경로 점을 이용하여 스피드스프레이어의 조향각을 결정하였으며 유압실린더를 이용하여 조향하도록 하였다. 과수의 높이가 3 m 내외인 과수원에서 목표 지점을 설정하고 1.6 km/h의 속도로 평균 1.2 초의 제어주기 상태에서 주행실험을 한 결과 최대 RMS 위치 오차가 72 cm인 것으로 보고하였다. 개발한 주행 시스템을 이용 주행성능은 제시를 하였다.

Noguchi 등(2001)은 RTK-GPS, FOG(Fiber Optic Gyroscope) 및 IMU(Inertial Measurement Unit) 센서를 활용한 자율주행 트랙터에 대하여 연구하였다. 자이로스코프로 얻은 데이터를 GPS 신호의 오류보정에 활용하였으며, 작업방향에 대한 오류는 3 cm이내, 항법시스템을 활용하면 일반적 농작업 상태에서 농업용 로봇을 정확하게 사용할 수 있다고 하였다.

Zhang 등(2004)은 트랙터가 자동으로 원하는 경로를 추종하기 위해 Look-ahead 동적 경로추종 알고리즘을 개발하였으며 컴퓨터 시뮬레이션을 통해 알고리즘을 평가를 받았다. 경로추종 알고리즘의 적용성은 RTK-GPS와 FOG 센서를 이용 포장 시험을 통해 검증을 하였으며 RMS 값은 0.03 m 이내의 성능을 얻었다.

Lenain 등(2006)은 정밀한 자율주행 시스템을 개발위해 가장 큰 문제점은 측면 슬립을 극복하는 것이 우선순위라고 제시하였다. 모델예측제어 방법을 기반으로 제어 로직을 구성하고 주행 플랫폼이 회전 시 GPS과 IMU 기반에 주행 경로의 곡률을 인식하여 슬립을 극복 가능한 경로추종 알고리즘을 개발하고자 하였다. 그 결과 ±15 cm 오차 범위의 주행 정밀도를 가진다고 보고하였다.

문희창(2008)은 무인자율주행차량을 구성하기 위해 GPS에서 발생하는 위치오차를 줄이기 위한 위치보정알고리즘, 무인자율주행차량의 기본적인 주행알고리즘인 경로점 주행, 도심지와 같은 많은 장애물이 존재하는 지역을 주행하기 위하여 사용한 경로점과 이벤트 주행 알고리즘에 대하여 연구하였다. 또한 각 시스템에서 생성된 데이터를 이용 무인 자율차량의 주행을 위한 알고리즘을 개발 검증하였다. 개발된 기술은 도로와 지면 마찰력이 큰 변동이 없으며 안전거리 유지를 위해 주행 알고리즘이 요구되는 무인 자동차 기술에 활용될 수 있는 요인기술 연구였다.

서동현 등(2008)은 지자기센서를 장착된 자율주행 트랙터가 방위각을 획득할 수 있는 시스템을 구성하였고 제어 프로그램을 개발하였으며 이를 사용하여 직진주행성능에 미치는 영향을 분석하였다. 지자기센서의 자편각을 보정하기 전과 후의 직진 주행 비교 실험을 진행하였으며 보정 전에 오차 평균값이 22.7 cm로 보이며 최대 오차는 54.96 cm였으나 자편 각을 보정 후에는 오차 평균값이 2.05 cm이며 최대 오차는 25.63 cm로 감소하는 것으로 나타났다.

Huang 등(2010)은 새머리 선회에서 경로추종의 성능을 향상을 위해 개선된 Pure pursuit method에 대해 연구하였다. 신경망 이론을 적용하여 동적 look-ahead 거리를 결정하여 주행 모델의 성능이 향상하는 것을 시뮬레이션을 통해 확인하였다. 고정된 Look-ahead 거리를 동적으로 변화하는 것이 주행외에도 새머리 선회영역에서 주행 시간과 공간을 감소하고 주행 효율도 증가 하는 것을 확인하였다.

서동현(2010)은 CAN-bus 기반의 분산제어 방식을 이용한 조향제어 시스템을 개발하여 시스템에 장착함으로써 시스템의 제어주기를 단축하고 시스템의 제어성능을 향상 시켰다. 트랙터가 설정된 경로를 따라 자율주행할 때 제어주기에 따라 설정된 경로에 대해 벗어나는 경로 오차값은 제어 주기가 느릴수록 커지는 것을 확인하였다. 트랙터가 0.63 m/s의 속도로 주행할 경우에는 제어 주기가 1.5 초일 경우에도 오차의 표준편차 값과 RMS 값이 각각 -0.06 m과 0.59 m로 크지 않지만 트랙터의 주행 속도가 1.34 m/s로 빨라지면 제어 주기 1.5 초의 상태에서는 오차의 표준편차 값과 RMS 값이 각각 0.7 m과 0.72m로 커지는 것을 확인하였다.

Bakker 등(2011)은 RTK-DGPS에 기초한 자율주행 로봇은 RTK-DGPS 필드에서 직선 경로를 따라 로봇 차량, 평균 최소 및 최대 가로 오차는 각각 1.6, 0.1, -4.5, 3.4 cm. 스플라인 기반의 알고리즘을 사용하여 부드러운 필드는 작물에 따라 실시간으로 경로를 연결하고 생성하였다. 작물의 행 맵핑은 로봇의 안내를 위한 좋은 결과를 보여주었다.

Fang 등(2011)은 주행 플랫폼의 타이어와 노면간의 Cornering Stiffness를 예측하는 강인제어 알고리즘을 개발하고자 하였다. 측면 가속도와 회전 변화율을 측정해 강인 Luenberger 관측기 기반으로 타이어 강성을 예측하며 예측된 슬립각을 연속시스템 기반으로 개발된 조향 로직에 반영하여 실험을 진행하였다. 시뮬레이션을 통해 알고리즘 수렴한 것을 확인하고 실차 주행에서 측

면 오차는 ±20 cm 였으며 방향 오차는 ±5 deg을 확인하였다.

김상철(2011)은 농업로봇 모바일 플랫폼 구성을 위해 20 kw급 트랙터에 주행과 조향, 작업기 제어가 가능하도록 액추에이터를 개발하였다. 자북측정에 영향을 미치는 외부 자계의 간섭을 효과적으로 제거할 수 있는 보정 방법을 개발하고 이를 이용하여 지자기 방위 센서의 자동보정기술을 개발하였다. 작업선검출을 위해 영상처리를 이용한 경로추종 알고리즘을 개발하고 트랙터의 자율주행 알고리즘을 검증하기 위하여 실제와 유사한 조건으로 시뮬레이션을 수행하였다. 포장에서 실제 실험결과 영상처리와 지자기센서를 이용하여 자율주행이 가능하였다. 통합 제어시스템에 의한 경로추종성능은 출발점으로부터 2 m 정도 떨어진 측후방에서 트랙터가 출발하도록 했을 때 약 3 초후 목표경로와 30 cm 이내의 오차로 접근, 직선경로는 20 cm 이내, 선회나 곡선경로는 50 cm 정도의 오차를 보였다.

우훈재(2011)는 개별 무인 차량의 시스템 연결 및 다수 무인차량 위한 상호운용 시스템을 구성하기위해 KUL_NET 규약 및 메시지를 설계하고 무인 시스템을 구성하는 시스템 아키텍처를 설계하였다. 각 무인차량의 시스템에서 생성된 데이터를 관제할 수 있는 관제차량을 따로 두어 다수 무인 차량의 선행 경로 추종 주행 알고리즘을 구현 검증하였다.

Backman 등(2012)은 트랙터와 트레일러 시스템의 추적 비선형 모델 예측을 통해 경로가 표시되는 효율적 조향 제어에 관하여 연구하였다. 인접한 구동라인은 2D 레이저스캐너를 이용하여 국부적으로 인식하였으며 지역 측정 및 GPS 정보는 확장 칼만 필터를 보조로 하였다. 위치는 트랙터의 전차축 및 유압 제어 조인트의 사용에 의해 제어되었으며 비선형 모델예측 제어(NMPC)기반 추종 작업을 수행하였다. 그 결과 12 km/h의 속도로 최대 10 cm의 오차가 발생하였는데 이는 사람이 운전하는 것과 유사한 정확도 수준인 것으로 나타났다.

Vougioukas (2012)은 자율 농업 차량의 움직임을 조정하기 위한 분산 제어 프레임 워크를 제시하였다. 시뮬레이션 실험 모델 및 외부 장애물이 없는 상태에서 컨트롤러는 초기에 큰 오차에도 불구하고 위치 및 방향 추적오차가 0으로 수렴하는 것을 확인하였다. 시뮬레이션 CNMPT 컨트롤을 사용하여, 차량이 효과적으로 자신의 속도 프로파일및 경로의 모양을 변경하여 인근 차량과의 충돌을 방지하도록 하였다.

Kraus 등(2013)은 종방향 및 횡방향 슬립 요인과 적응모델을 기반으로 자율적인 트랙터 추종을 위한 MHE-NMPC 프레임 개발 연구가 수행되었다. 개발 MHE는 미지의 토양 조건 변화에 실시간으로 모델 매개 변수를 추정 할 수 있는 방식이다. NMPC는 MHE에 의해 제공된 추정결과에 기초하여 최적에 근접한 제어 출력을 나타내었다. MHE 성능은 파라미터의 정확한 추정을 제공하여 NMPC 컨트롤러의 경로추종 성능을 가능하게 했다고 보고하였다.

한웅철(2013)은 자율경운 작업을 진행하기 위해 3가지 경운 작업 중 C형 작업경로의 수학적 모델 개발하여 실제 포장에서의 자율 주행 경운작업을 개발된 경로생성 알고리즘과 경로추종 알고리즘으로 구현한 컴퓨터 시뮬레이션 프로그램을 개발하였다. 트랙터가 포장에서 선회주행 시 슬립영향을 고려하기 위해 시뮬레이터의 모델은 확장 기구학 모델을 개발하였고 트랙터 주행 시 look-ahead 추종 알고리즘을 적용했을 때 주행 성능은 만족스러운 것으로 나타났으나, 슬립정도의 영향성, 선회주행 후 작업로 진입에 따른 경로점 추종정확도에 대한 연구와 경운작업 성능에 대한 검증은 이루어지지 않았다.

Table 1-2는 경로추종 문헌을 분석한 결과로서 Pure-pursuit model, Bicycle model 및 Ackerman model 등과 같은 주행모델 기반 경로추종 알고리즘 연구가 주를 이루면서 자동차 분야에서는 도로 주행에서 슬립 현상을 극복 위한 제어기술을 다루고 있었으나 상대적으로 농기계 분야에서는 작업 주행 시 발생한 슬립을 극복하는 알고리즘 연구는 상대적으로 미흡한 것으로 나

타났다. 특히, 주행 추종성능에 영향을 주는 슬립이 무논과 같은 토양의 수분 함량이 비교적 높은 조건에서는 크게 작용하기 때문에 관련 연구가 필요한 것으로 나타났다.

Table 1-2 Summary of literature review on path tracking methods

분류	세부분류	기존연구	필요사항
차이점	조향	• 자동차 도로 주행의 슬립 극복 위한 조향 기술 제시 • 트랙터 포장 주행의 외란 극복 가능한 조향 기술 제시 없음	트랙터가 포장 작업에서 발생한 슬립을 극복이 가능한 조향 기술 적용
	작업기	• 경사진면에 따른 작업기 수평 제어 방법 제시 • 작업기 작업 깊이 제어 방법 제시	새머리 구역의 진입 및 탈출 점에서 미 경지 방지를 위한 작업기 상승/하강 제어
	작업 정보	농기계의 주행 위치 정보만 제공하였고 작업 정보 제시 없음	계측 시스템을 통해 자율주행을 이용한 작업 정보 제시
	작업 수준	경로추종의 정밀도 향상에 집중하였으나 작업 수준 제시 없음	자율주행과 농민 직접 작업 수준 비교 하여 작업 수준 제시
공통점	• 지형에 따른 GPS 위치 보정법 적용 • Ackerman model 이용한 조향 제어 • 관측기를 이용한 슬립 예측		

1.2.3. 기술 개발 현황

2004년 농촌진흥청과 LS전선(주)과 공동으로 직선경로에서 20 cm 오차를 갖고 인터넷을 통해서 실시간으로 논밭을 감시하여 경로 추종 및 생성할 수 있는 시스템을 개발하였으나 실제 작업에 적용하여 실용화하기 위해서는 다양한 시험을 통해 최적화가 필요하다고 보고하였다.

2012년부터 서울대학교에서는 최적 경로추종 알고리즘 개발을 위한 조향 시뮬레이터를 RoboticsLab 시뮬레이션 프로그램과 연동하여 실제 토양조건에 인해 동작 상태를 프로그램에서 확인할 수 있는 실내 모의 요인 실험 장치를 개발하여 다양한 경로추종과 조향장치 제어기술에 활용이 가능함을 보고했다.

2011년부터 2013년까지 동양물산기업(주), 언맨드솔루션, 서울대학교와 공동으로 83 kW급 트랙터에 자율적인 경운작업이 가능하도록 경로생성 및 추종 기술과 전자식 조향장치, 엑셀레이터, PTO on/off 등 다양한 액츄에이터를 CAN 통신을 통하여 작동되는 무인 경운작업기를 개발하였으나 다양한 경지 형태에 따른 경로 자동생성 기술과 위치 및 자세 정밀도를 높이기 위한 센서 기술의 연구가 필요함을 보고하였다.

2009년 일본의 Yutaka 연구팀은 기존 이앙기의 자율주행에서 육모의 라인 검출 방식인 일반적으로 사용되는 NIR 혹은 RGB camera가 마른 논에 비해 논담수에 의한 강한 태양 반사광으로 인해 라인검출이 매우 어렵다는 것을 감안해 이를 개선한 연구를 진행함. 저가의 monochrome 카메라 2대와 광학필터를 이용하여 듀얼 스펙트럼 카메라 시스템을 구성하여 실험을 수행하였으며 결과는 물에 의한 반사 노이즈를 줄이고 육묘의 라인을 정확하게 검출 하였다고 보고하였다.

2008년 일본의 구보다와 닛폰항공전자공업, 생연센터는 32 마력시판 트랙터를 기본으로 하여 경운 트랙터를 개발하였다. 이 트랙터는 항법시스템, 작업 소프트웨어가 설치된 메인 컨트롤러 , 차량 컨트롤러 및 차량제어 시스템으로 구성되어 있으며 항법시스템은 이용 상황에 따라서 광학측량방식, DGPS 방식, 전자유도방식 가운데 하나를 선택할 수 있도록 제작되었다.

미국의 경우는 농작업을 위한 자율주행 기술이 이미 실용화되어 관련 제품이 보급되고 있다. 존디어사의 경우 자회사인 Navicom에 의해 구축된 위성 레퍼런스 네트워크를 이용하여 별도의 지상 보정국 없이 15 cm이내의 오차수준의 측위정밀도를 갖는 GPS를 사용하고 있으며 자율주행 키트를 판매하고 있다. Trimble사는 AgGPS Autopilot system기반의 농기계를 위한 자율주행 솔루션을 개발하여 트랙터 및 콤바인에 장착할 수 있는 제품을 판매하고 있으며 농작업의 기계화, 정밀화, 대형화 그리고 지능화를 위한 농기계시스템 개발

을 목표로 지속적 연구개발을 진행 중에 있다.

존디어, C&H 등 농기계회사는 트랙터, 엔진, 부착, 작업기, 운반차, 건설용 중장비 등을 생산하고 있으며, GPS/GIS 기반과 자체 위성을 활용한 무인자율주행 적용기술을 장착하여 판매하고 있다. 특히, 미국에서 GPS 장착하여 트랙터로 자율주행 보조로 활용하는 농민이 약 30% 인 것으로 조사되었다. Starfire, Topcon, Trimble, Outback 등 다양한 회사에서 관련제품을 보급하고 있다.자동수확과 관련하여 미국 카네기멜론 대학은 Holland 콤바인에 DGPS, 관성항법시스템, 카메라를 장착하여 작물의 예취와 미예취부를 자동으로 인식하면서 수확하는 시스템을 개발하였으며 1997년에 알파파 수확에 적용하여 그 효용성을 증명하였다. 또한, 카메라를 이용하여 장애물을 감지하면서 콤바인의 주행을 제어하는 기술도 함께 연구하였다.

1.3. 연구목적

본 연구에서는 농업용 트랙터에 무인자율주행을 위한 각종 제어 및 조작 장치를 탑재하고 다양한 형태의 포장에서 무인 자율로 농작업이 가능하도록 작업경로 자동 생성과 추종 제어 시스템을 개발하고자 하였다. 이를 위해 직교형, 사다리형, “L”자형 등 다양한 형상의 포장 내에서 경운작업을 수행하고 자동선회가 가능한 경로생성 알고리즘과 주행 중 발생하는 노면의 슬립정도를 예측하면서 작업경로를 최적으로 추종하는 강인 조향제어기술을 개발한다. 가상현실에서 무인 농작업이 가능한 시뮬레이터를 이용하여 개발한 알고리즘을 검증하였으며 최종적으로는 개발 기술을 실제 농용트랙터에 탑재하여 포장에서 무인 경운작업을 수행하여 작업 성능과 효율 등을 숙련된 작업자가 수행한 결과와 비교 개발기술의 효용성을 평가한다. 본 논문의 목적을 달성하기 위해 세부 목표를 다음과 같이 설정하였다.

o 농지형태에 따른 주행 경로생성 알고리즘 개발

o 강인 제어 기반의 동적 경로추종 기술 개발

o 무인 농작업 시뮬레이터 개발 및 활용 기술

o 자율주행 트랙터 통합 시스템 개발 및 시험 평가

2. 비정형 농지형태에 적합한 경운작업 경로생성 알고리즘 개발

2.1. 재료 및 방법

2.1.1. 개요

일반적으로 경로생성 알고리즘은 무인자율주행차량이 출발하는 초기위치 좌표에서 목표점의 위치 좌표까지 이동하기 위한 경로를 따라 주행하면서 차량 주위의 장애물정보를 획득하여 차량이 장애물과 충돌위험성이 있을 경우에 그 장애물을 회피할 수 있는 경로를 의미하는 것으로, 사전에 지도가 존재 한다면 A*, Dijkstra, D*등과 같은 경로계획 알고리즘을 사용해 최적 경로를 생성하며 지도가 존재 하지 않는다면 사용자가 직접 경로를 설정하게 된다. 자율주행차량의 주행경로는 단순히 일정한 도로를 주행하거나 목적지를 이동하기 위한 경로 좌표를 생성하지만 자율주행 트랙터가 포장에서 경운 및 작업을 진행하기 위해서는 기본적으로 트랙터가 주행할 경로와 작업기의 사용유무에 대한 정보가 포함된 주행 경로생성이 필요하다. 또한 포장마다 포장의 크기 및 형태가 일정하지 않으므로 본 연구에서는 포장의 2D 좌표를 이용하여 자동으로 포장의 형태에 적합한 경로가 생성한다. 그리고 포장에서 자율주행 트랙터가 주행 시 작업 속도와 선회 속도가 서로 다르기 때문에 속도제어가 필요하며 비 작업부에서 작업부로 진입 및 탈출 시 특정된 위치에서 작업기 제어가 필요하다.

숙련된 트랙터 작업자는 포장에서 작업 시 환경 및 경제 측면을 고려하여 포장의 등고선의 방향 혹은 포장의 장변 방향으로 주행하는 것을 원칙으로 잡고 있다. 본 연구에서는 3차원 지형을 고려한 경로생성 기술은 포함하지 않았으며 경사지에 따른 등고선 생성 원칙은 적용하지 않았다. 대신 주어진 포장 정보를 이용 사용자가 선택한 주행의 방향에 따라 포장 내에서 직진과 선회 경로를 모두 포함하는 기술을 다루고자 하였다.

2.1.2. 주행 경로생성을 위한 좌표계 변환

트랙터의 움직임을 계산하기 위해서는 공간을 어떤 기준으로 나타낼 것인가를 결정하게 되고, 그 기준이 바로 좌표계이다. 좌표계는 사용자의 의도에 따라 임의의 결정이 가능하며 이러한 좌표계는 삼차원 좌표계와 평면 좌표계로 구분된다.

2.1.2.1. 삼차원 경위도 좌표계

경위도 좌표는 지구의 절대위치를 나타내는데 가장 많이 사용되는 좌표로 남북방향의 경선과 동서방향의 위선으로 표현한다. 기준타원체에 대한 지점위치를 경로, 위도 및 평균해수면에서 부터의 높이로 표시한 것이다. Fig. 2-1과 같이 경도는 본초자오선을 기준으로 동서로 각각 180도로 나누어 동경, 서경으로 표현한다. 위도는 적도를 기준으로 남북으로 90도로 나누어 남위, 북위로 표현한다.

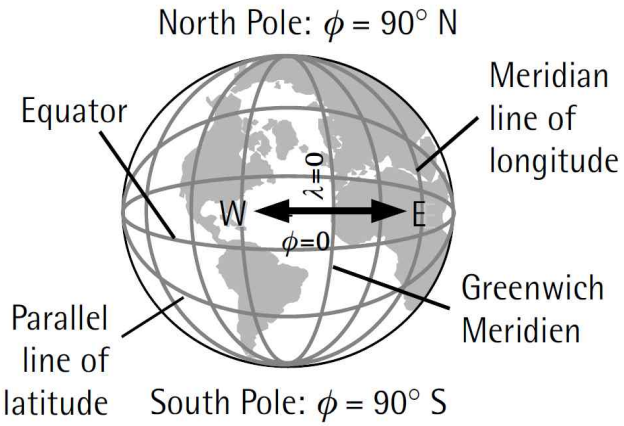


Fig. 2-1 Three-dimensional coordinates system (El-Rabbany, 2002)

2.1.2.2. 평면 TM 좌표계

평면직각 좌표계는 측량 범위가 넓지 않은 곳에서 일반측량을 위해 주로 사용되는 좌표계이다. 평면직각 좌표는 3차원 좌표를 투영법에 따라 투영하여 평면으로 만든 만큼 투영오차가 발생하지만, 넓지 않은 경우 작은 오차 범위를 갖는다. TM(Transverse Mercator)좌표는 임의의 지역에 대한 기준 지점을 좌표원점으로 정하고 원점을 중심으로 TM 투영한 평면상에서 원점을 지나는 자오선을 X축, 동서방향의 위도선을 Y축으로 하여 각 지점의 위치를 m 단위의 평면 직각좌표계로 표시한다(한국에스리, 2005). 일본의 경위도 원점을 기준하여 그 위치가 결정되었던 한국과 일본의 경도 값이 10.405초 커지게 되었다. 그래서 한국의 TM 좌표 값은 Table 2-1과 같이 TM 좌표계 원점들을 기준하여 WGS84 타원체를 바탕으로 동서방향의 거리 값(m)과 남북방향의 거리 값(m)으로 표시한다.

Table 2-1 Datum of TM Coordinate System (한국에스리, 2005)

데이텀	Korean 1985 데이텀(Tokyo D 데이텀)	
투영법	TM(Transverse Mercator)	
투영원점	서북:125도10.405초E, 38도N 중북:127도10.405초E, 38도N 동북:129도10.405초E, 38도N 동해:131도10.405초E, 38도N	원점 127도30초E, 38도N
투영원점 가산값	False Easting:200,000m False Northing:500,000m	False Easting:2,000,000m False Northing:1,000,000m
중앙자오선의 축척 계수	1.0000	0.9996

2.1.2.3. 좌표계 변환

GPS경위도와 한국의 경위도는 사용하는 타원체의 제원이 다르므로 한국에서 사용가능한 TM 직교좌표로 변환이 필요하다. 경위도는 삼각망의 정확도에 따라 지역별로 다소의 오차가 포함되어 있으며, 측지측량에 따른 오차와 일본의 측지계로부터 옮겨오는 과정에서 포함된 오차 등이 포함되어 있으므로 좌표변환 시 좌표축의 이동, 회전, 축척변경(Translation, Rotation, Scaling)이 필요하다. Fig. 2-2는 GPS출력인 WGS84(World Geodetic System 1984) 좌표계를 평면 직각 좌표계로 변경하는 방법이다.

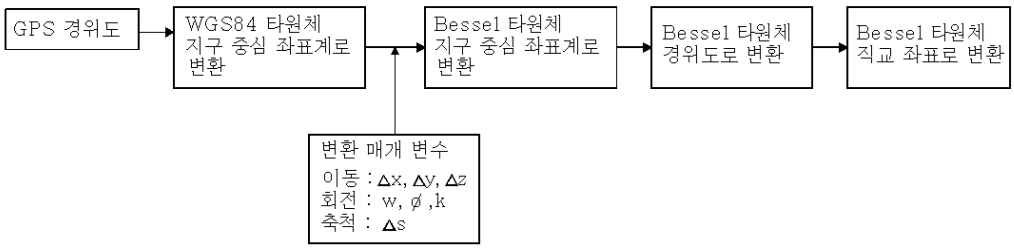


Fig. 2-2 Process of coordinates conversion from WGS84 to TM

Fig. 2-3과 같이 지심직각좌표계는 원점을 타원체중심과 일치하도록 하였으며 타원체의 중심으로부터 그리니치 자오선과 적도면이 만나는 점으로 연장하여 X축으로 정의한다(권대원, 1995). 또한 X축으로부터 적도면을 따라 동쪽으로 90°위치에 Y축을 정의한다. 타원체의 회전축을 Z축으로 한다.

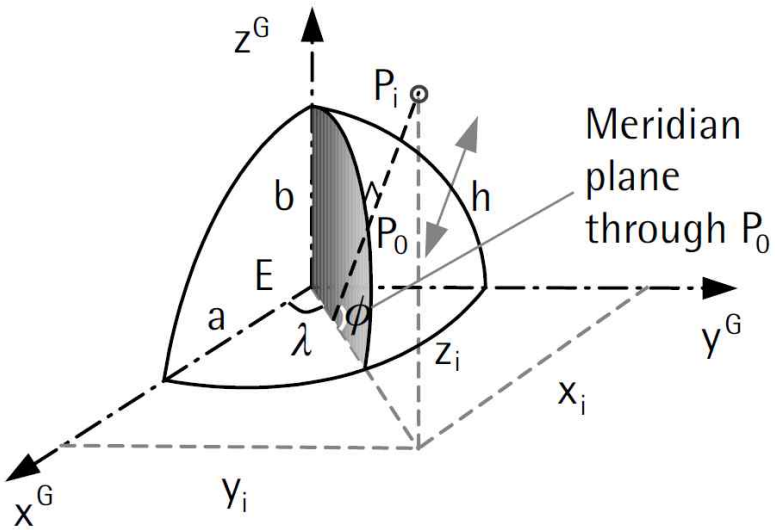


Fig. 2-3 Geodetic and cartesian coordinates (El-Rabbany, 2002)

점 Q의 지리좌표 Q(ϕ, λ, h)가 주어질 경우에 대응되는 지심직각 좌표 Q(X, Y, Z)는 Eq. (2-1), Eq. (2-2), Eq. (2-3)과 같이 계산한다. 횡 곡률 반경은 Eq. (2-4)과 같이 정의된다.

$$X = (N+h)\cdot\cos\phi\ \cos\lambda \tag{2-1}$$

$$Y = (N+h)\cdot\cos\phi\ \sin\lambda \tag{2-2}$$

$$Z = \left(\frac{b^2}{a^2}N+h\right)\cdot\sin\phi \tag{2-3}$$

$$N = \frac{a^2}{\sqrt{a^2\cos^2\phi+b^2\sin^2\phi}} = \frac{a}{\sqrt{(1-e^2\sin^2\phi)}} \tag{2-4}$$

여기서, Φ = 위도 (°)

λ = 경도 (°)

h = 고도 (m)

a = 타원체의 장반경 6378137 (m)

b = 타원체의 단반경 6356752.314 (m)

X, Y, Z = 지심직각 좌표 (m)

N = 횡 곡률 반경 (m)

점 Q의 지심직각좌표 Q(X, Y, Z)가 주어질 때 대응되는 지리좌표 Q(ϕ, λ, h)는 Eq. (2-5). Eq. (2-6), Eq. (2-7)과 같이 계산된다.

$$\phi = \arctan\left(\frac{Z}{L}\right) \tag{2-5}$$

$$\lambda = \arctan\frac{Y}{X} \tag{2-6}$$

$$\mathrm{h} = \frac{P}{\cos\phi} - N \tag{2-7}$$

여기서, $\mathrm{L} = \sqrt{\left(\frac{\mathrm{b}^2}{\mathrm{a}^2}\mathrm{N} + \mathrm{h}\right)^2 - \mathrm{Z}^2}$

$P = \sqrt{X^2 + Y^2} = (N + h) \cdot \cos\Phi$

제1 이심율과 제2 이심율은 Eq. (2-8), Eq. (2-9)과 같이 계산된다.

$$\mathrm{e}^2 = \frac{\mathrm{a}^2 - \mathrm{b}^2}{\mathrm{a}^2} \tag{2-8}$$

$$\mathrm{e}'^2 = \frac{\mathrm{a}^2 - \mathrm{b}^2}{\mathrm{b}^2} \tag{2-9}$$

여기서, e = 제1 이심율

e' = 제2 이심율

GPS WGS84 지심직각 좌표 Q(x, y, z) 좌표계를 Molodensky-Badekas 변환모델을 사용하여 7변수 상사변환과 왜곡모델링 결과의 보정에 의해서

Eq. (2-10)과 같이 Bessel 타원체에 근거한 한국의 좌표계를 새로운 좌표계 (X, Y, Z)로 변환한다.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} dX \\ dY \\ dZ \end{bmatrix} + (1+\triangle s)\begin{bmatrix} 0 & \kappa & -\phi \\ -\kappa & 0 & \omega \\ \phi & -\omega & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad (2\text{-}10)$$

여기서, $X, Y, Z =$ Bessel 타원체에 근거한 신좌표계 (m)

$x, y, z =$ WGS84의 구좌표계 (m)

$\omega, \phi, k =$ 회전각 (°)

$dX, dY, dZ =$ 이동량백터 (m)

$\triangle s =$ 축척차이 (ppm)

경위도에서 평면 직교좌표로 계산은 Eq. (2-11), Eq. (2-12)와 같다.

$$X = \triangle\lambda N cos\Phi + \frac{(\triangle\lambda)^3}{6} N cos^3\Phi(1-\tan^2\Phi+\eta^2) + \frac{(\triangle\lambda)^5}{120} N cos^5\Phi(5-18\tan^2\Phi+\tan^4\Phi) \qquad (2\text{-}11)$$

$$Y = B\Phi + \frac{(\triangle\lambda)^2}{2} N cos^2\Phi\tan\Phi + \frac{(\triangle\lambda)^4}{24} N cos^4\Phi\tan\Phi(5-\tan^2\Phi+9\eta^2+4\eta^4) + \frac{(\triangle\lambda)^5}{720} N sin\Phi\cos^3\Phi(61-58\tan^2\Phi+\tan^4\Phi) \qquad (2\text{-}12)$$

여기서, $\eta^2 = \frac{e^2}{1-e^2} cos^2\Phi,$

$\triangle\lambda = \lambda - \lambda_o$

2.1.3. Extended RDDF 구성

Waypoint는 경로 계획에 의해 생성한 경로점이며 트랙터 주행 경유점과 참고점의 의미로 사용하고 있다. 경유점은 주행하면서 그 점을 꼭 지나야만 하는 점이며 참고점은 그 점을 참고만 할 뿐이고 지나갈 필요는 없는 점이다. 본 연구에서는 조향을 생성하기 위해서 Waypoint를 참고점으로 사용하며 Waypoint 자체를 변경할 때는 경유점으로 사용하고 있다. 자율주행 트랙터는 스스로 어느 곳으로 갈 것인지를 결정할 수 없기 때문에 트랙터가 특정 목적지까지 주행하기 위해서는 목적지에 대한 위치정보를 트랙터에 입력을 해야 한다.

작업자는 트랙터 포장작업 주행 전 트랙터가 작업 할 목적지의 위치정보뿐만 아니라 그 지점에 대한 위치정보도 같이 입력 한다. Waypoint 정보는 Fig. 2-4와 같이 경로 점의 집합을 Extended RDDF(Route Data Definition File) 형태로 경로 점을 생성한다.

Index	Latitude (m)	Longitude(m)	LBO(m)	Curve	Velocity(km/h)	PTO
21	540648.8371	195572.4481	0.6	0	3	1

Fig. 2-4 Configuration of RDDF format

RDDF는 텍스트(.txt)파일에서 7개의 열로 이루어졌다. 본 연구에서 설계한 RDDV의 1열은 경로 점(Waypoint)의 순서를 나타내며, 2열은 경로 점의 X방향 좌표(TM)를 나타낸다. 3열은 경로 점의 Y방향 좌표(TM)를 의미하며, 4열은 LBO(Lateral Boundary Offset)를 나타내며 Waypoint의 오차 범위이다. Fig. 2-5와 같이 LBO의 범위는 원으로 나타내며 RDDF에서 반경 미터(m) 단위로 설정하였다. 원리적으로 트랙터가 설정된 범위에 들어오면 그다음 목표

점으로 갱신하여 주행한다. 그리고 자율주행 트랙터가 일반적으로 주행뿐만 아니라 그 외에 특정 위치에서 행동을 할 수 있도록 액추에이터를 작동 위해 명령을 주어야 한다. Fig. 2-6과 같이 트랙터가 새머리 영역으로 진입 및 탈출 시 각각 요구된 주행 속도 및 작업기 제어가 필요하다. 주행속도 및 작업기 제어를 위해 특정지점에서 수행할 명령을 제공해 주어야 한다. 즉 6 열과 7 열은 현 경로 점과 다음 경로 점간의 트랙터 주행속도와 작업기 상태를 나타낸다.

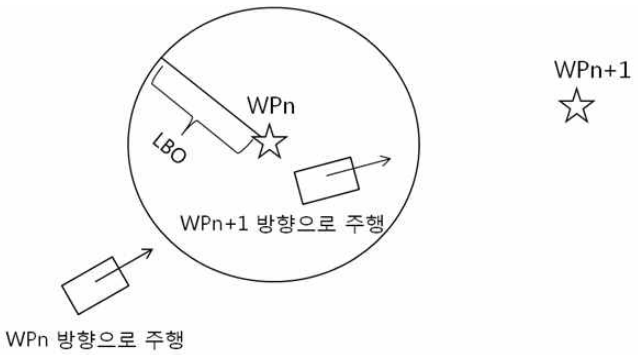


Fig. 2-5 Concept of LBO used in auto-guidance

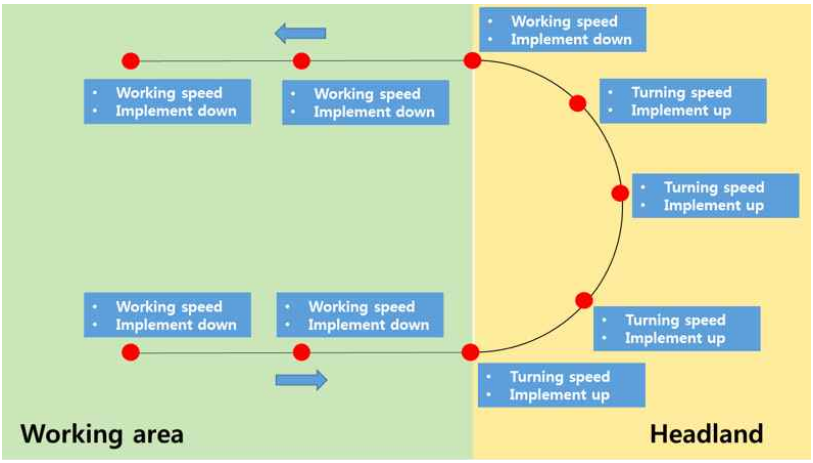


Fig. 2-6 Definition of working speed and implement status in the working area and headland area

계획한 주행 경로 위치 점과 작동 명령은 RDDF에 포함하게 되고 자율주행 트랙터가 RDDF에서 제공된 경로 정보를 바탕으로 주행 및 작업을 수행 하게 된다. Table 2-2는 주행을 위한 RDDF의 구성 예이다.

Table 2-2 An example of RDDF format used in the autonomous system

Index	Latitude (m)	Longitude (m)	LBO(m)	Speed (km/h)	PTO
1	540648.83714	195572.4481	0.6	4	0
2	540646.38641	195570.7177	0.6	4	1
3	540644.52135	195570.2375	0.3	2	0
.....	.....	.....	.....	.....	.....
19	540634.41698	195613.42345	0.3	2	0
20	540631.79841	195606.98206	0.3	4	1

2.1.4. 자율주행 트랙터의 경로생성 시스템 구성

트랙터가 출발 위치에서 목적지까지 주행하기 위해서는 설정한 작업 영역에서 지나가야 할 모든 위치점을 도출하기 위해 주행경로를 생성한다. 본 연구에서 수행한 다변형 포장에서 주행경로의 구성은 Fig. 2-7과 같다. 즉, 반시계 방향기준으로 포장 2차원 경계점을 모두 연결하여 빨간색선과 같이 포장 외부 경계선으로 표현하였다. 그 다음으로 트랙터가 포장에서 주행할 새머리 영역을 결정한 후 포장 내부 경계선을 표현하였다. 트랙터가 선정된 회전 패턴을 사용하여 내부 작업 영역부에서 파랑색선과 같이 왕복작업 경로를 생성 후 새머리 영역에서 흰색선과 같이 외부 회경 작업경로를 생성하도록 하였다.

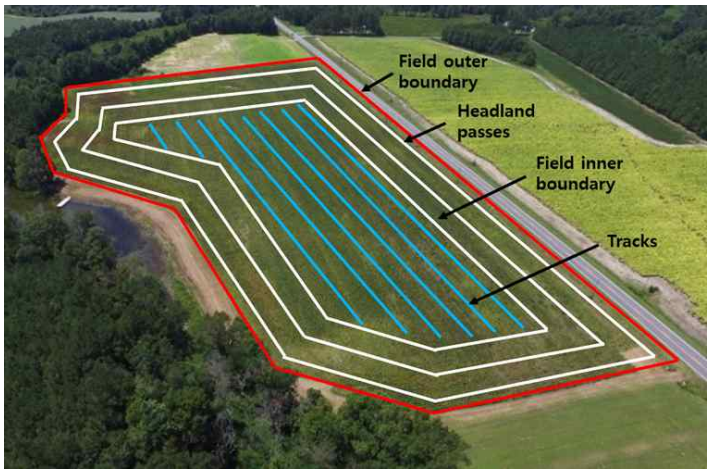


Fig. 2-7 Representation of a typical field structure in the study

트랙터가 포장에서 자율적으로 농작업을 수행하기 위한 주행 경로생성 시스템은 LabVIEW 2015 프로그램을 사용 개발하였으며 입력변수를 통해 작업 경로생성 후 RDDF 출력하여 경로추종에 사용할 수 있게 구성하였다.

자율주행 트랙터의 주행경로 생성 시스템 구성은 Fig. 2-8과 같다. 시스템의 전체적인 구성은 크게 네 개 부분으로 구성하였다. 첫 번째는 내부왕복작업부와 외부작업부로를 구분하기 위해 포장 외부 작업 경계선과 내부 작업 경계선

을 생성하였다. 변수 설정부분으로부터 트랙터 작업기의 넓이, 작업 겹침 량과 주행 방향 등을 수동 입력이 가능하도록 하였으며, 프로그램을 실행하면 Text 파일형태로 반 시계 방향으로 저장된 포장 좌표 정보를 순차적으로 불러 들어오게 하였다. 포장 외부 작업부에서 주행할 회경 횟수를 입력 받게 되면 포장 내부 경계 영역을 생성하였다.

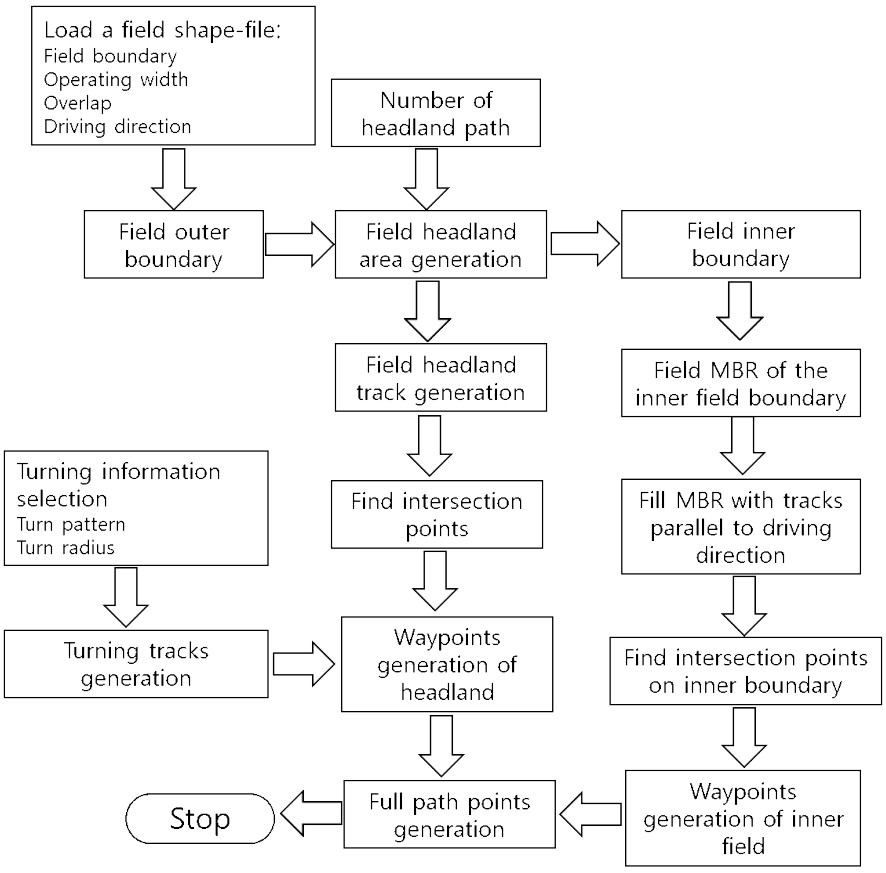


Fig. 2-8 Flowchart of the coverage path algorithm in polygonal field

다음 단계는 포장 내부 작업부에서 트랙터가 왕복작업을 진행하기 위한 직진 경로를 생성하는 것이다. 다변형 포장 내에서 미경지 작업 영역의 발생 방지를 위하여, MBR(Minimum Bounding Rectangle) 원리 기반으로 왕복 직진

작업 경로를 생성하였으며 사용자가 설정된 주행 방향을 이용 MBR 범위 내에서 평행선을 생성한 후 각 평행선과 포장 내부 경계선이 교차된 두 점을 트랙터 직진 주행의 시작점과 종료점으로 도출하였다. 포장 외부 작업부에서 트랙터의 중복작업 영역과 회전을 위한 미 작업 거리는 적합한 주행 방향에 따라 결정하였으며 내부 왕복작업부에 직진 경로와 포장경계선 방향이 수직방향 관계이면 중복작업 영역과 회전 미 작업시간이 가장 작으며 수직 방향 관계가 아닌 경우는 교차 각도 ($\theta = \theta_{path} - \theta_{field}$)를 이용 외부 작업부에서 회전 주행을 위한 미 작업 영역 경로를 생성하였다.

다음으로 포장 내부 경계선을 생성하기 위해 Fig. 2-9와 같이 포장 외부 작업부에서 결정된 회경 횟수를 이용 생성하는 방식으로 개발하였다. 회경 횟수는 3 회로 가정하였으며 첫 번째 회경 경로와 포장 외부 경계선간의 거리는 유효 작업 폭 절반 (w/2)로 유지하여 생성하고 나머지 회경 경로는 유효 작업 폭(w) 만큼의 거리를 유지하여 순차적으로 생성하는 방식을 사용하였다.

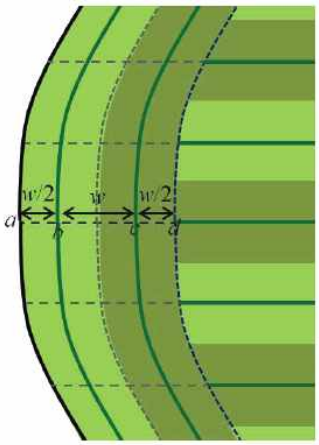


Fig. 2-9 Creation of headland passes in the polygonal field (Hameed et al., 2013)

네 번째는 트랙터가 새머리 영역에서 선회 경로를 추가하는 부분이다. 무인

농작업 시 왕복 작업부에서 선회 작업부로, 선회 작업부에서 다시 왕복 작업부로 이동 시 180도 회전하게 된다. 여기서 선회방식과 선회반경을 선정 후 선회 주행경로를 생성하여 MBR 기반으로 생성한 직진경로에 포함하게 된다.

2.1.5. MBR법을 이용한 직진주행 경로생성

다변형태 포장 내에서 미경지 작업 영역의 발생 방지를 위하여, Fig. 2-10과 같이 MBR 원리 (Zhou et al., 2014) 기반으로 왕복 직진 작업 경로를 생성하였다. MBR은 하나의 다변형태의 객체를 둘러싸고 있는 최소 경계 직사각형이다. 대각선 위의 두 점, 즉 객체의 가장 작은 X, Y 좌표 점과 가장 큰 X, Y 좌표 점으로 2개의 좌표로 지정된다. 객체의 형태를 미리 추정함으로써 객체 검색 시간을 줄이기 위해 사용된다. 지리사상의 모든 부분은 반드시 최소범위 사각형의 경계선을 포함한 내부 영역에 존재한다. 또한 (최소 점 X, 최소 점 Y, 최대 점 X, 최대 점 Y) 4개 정보를 이용 MBR를 표현이 가능하다.

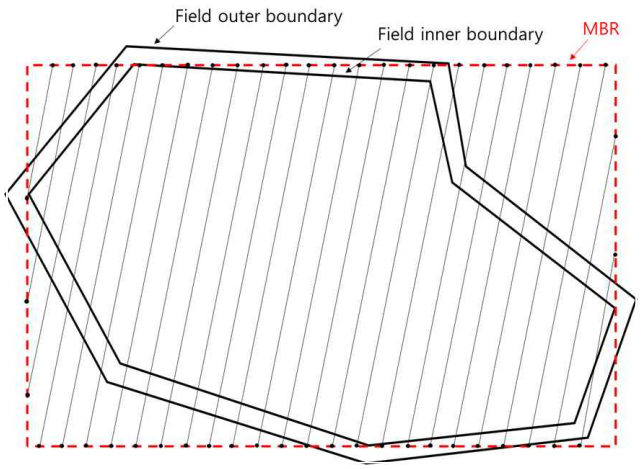


Fig. 2-10 A MBR method in the polygonal field

사용자가 설정된 주행 방향과 작업 폭을 이용 MBR 내에서 직진주행 할 평행선을 생성 후 각 평행선과 포장 내부 경계선이 교차된 두 점을 Eq. (2-13),

Eq. (2-14)와 같이 트랙터가 직진 주행의 시작점과 종료점으로 도출하였다. 본 연구에서는 트랙터가 포장 내에서 원활한 자율주행 작업을 수행하기 위해 트랙터가 생성된 첫 번째 평행선의 시작점 주변에서 자율주행 작업을 시작하게 한다. 다만 트랙터의 주행 시작 방향은 첫 번째 평행선의 시작점(x_i,y_i)과 끝점(x_{i+1},y_{i+1})으로 구성된 직선 방향의 ±90도 범위 내로 정해진다.

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & -\tan\theta \\ 1 & -\tan\psi_i \end{bmatrix} \times \begin{bmatrix} y_i - x_i \cdot \tan\theta - \dfrac{n(w-w_o)}{\cos\theta} \\ y_i - x_i \cdot \tan\psi_i \end{bmatrix} \quad (2\text{-}13)$$

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -\tan\theta \\ 1 & -\tan\psi_{i+1} \end{bmatrix} \times \begin{bmatrix} y_i - x_i \cdot \tan\theta - \dfrac{n(w-w_o)}{\cos\theta} \\ y_i - x_i \cdot \tan\psi_{i+1} \end{bmatrix} \quad (2\text{-}14)$$

여기서, θ = 포장 내부 직진주행 방향 (°)

ψ_i = 포장 내부 i번째 경계선의 방향 (°)

w = 작업기 폭 (m)

w_o = 작업 겹침 량 (m)

x_i, y_i = MBR의 평행선 시작점 (m)

x_{i+1}, y_{i+1} = MBR의 평행선 끝점 (m)

2.1.6. 주행 경로생성을 위한 다변 형태 포장 분할 알고리즘

복잡한 형태의 포장에서는 여러 개의 영역으로 나누어지는 분할(Choset et al., 1998) 작업이 필요하다. 볼록 형태의 포장에서는 구획작업이 필요 없으며 오목형태의 포장에서는 셀 분할 작업이 필요하다. 본 연구에서도 주어진 포장이 오목형태이면 경로생성 전에 영역을 분할하였다. 나누어진 영역을 셀(Cell)이라고 명명하였으며 주행 경로는 각 셀에 대한 경로를 생성 후 각 영역의 경로를 하나의 집합을 만들었다. 여기서 주어진 영역을 나누기 위해서는 하나의

스위프 라인 함수(Sweep line function)를 정하였다.

정의된 스위프 라인 함수는 y축과 평행한 직선의 함수로 Eq. (2-15)과 같이 표현된다.

$$f_{sweep\ |} = x_c \tag{2-15}$$

여기서, x_c = 스위프 라인 (m)

Fig. 2-11은 주어진 영역을 분리하는 원리를 나타낸 것이다. 정의된 스위프 라인 함수를 이용해 Fig. 2-11(a)과 같이 좌측부터 우측으로 이동시켜 준다. Fig. 2-11(b)과 같이 임계점(Critical point)을 만나면 이 경계점에서 위, 아래로 선을 연장하여 경계선(Boundary)을 만든다. 이렇게 만들어진 경계선이 기존 포장 경계선과 만나는 교차점을 이용 새로 만들어진 포장의 좌표점으로 사용한다. 스위프 라인이 이동 시 임계점을 만날 때 마다 새로운 경계선을 생성하기 때문에 영역을 분할하여 셀을 구성한다.

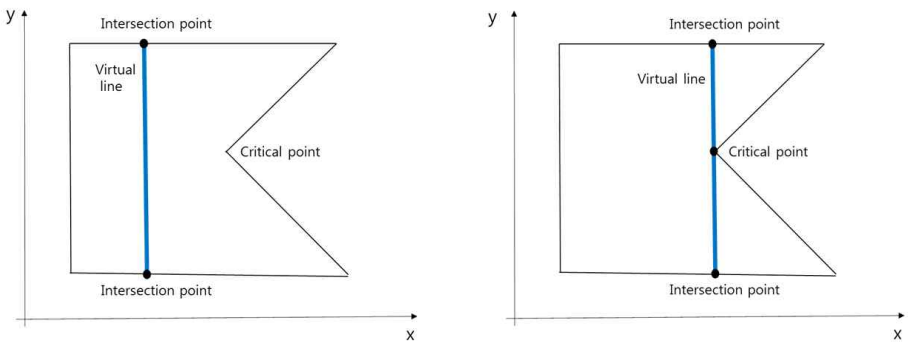
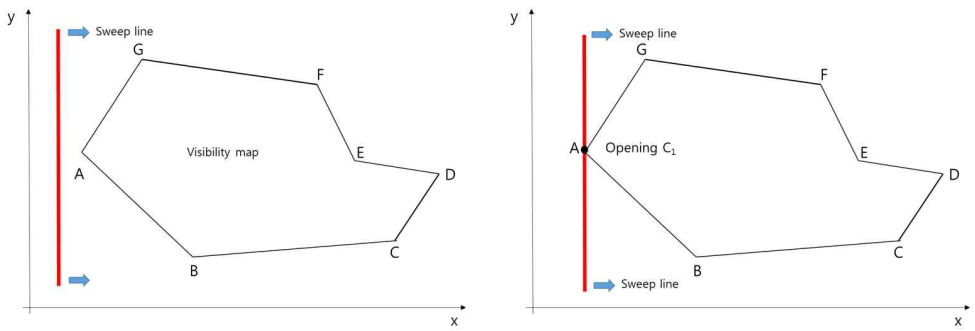


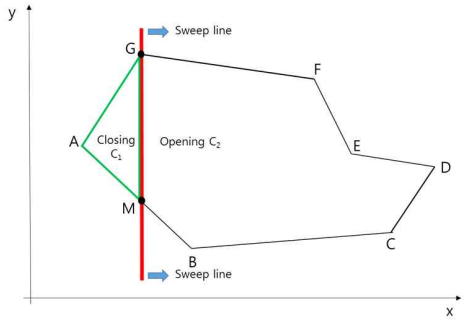
Fig. 2-11 Slice movement from left to right (a), boundary creation by critical point and intersection points (b)

Fig. 2-12는 주어진 영역을 분리하는 과정을 나타낸 것이다. Fig. 2-12(a)는 스위프 라인이 좌측부터 우측으로 이동 준비한다. Fig. 2-12(b)와 같이 스위프 라인이 첫 번째 임계점 A를 만나게 되며 셀 C_1을 열어준다. 스위프 라인이 계

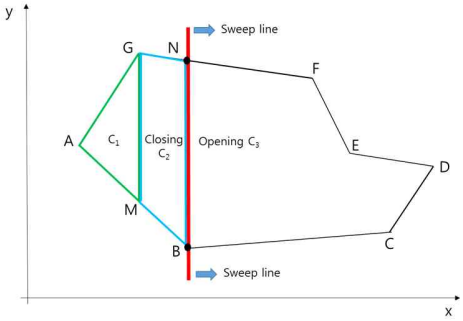
속 이동하며 Fig. 2-12(c)와 같이 임계점 G를 만나게 되면 셀 C_1을 닫아주고 교차점 M과 포장 경계점 A, G는 셀 C_1의 셀 경계점으로 정의한다. 여기서 셀 C_1을 생성 후 이어서 바로 셀 C_2를 열어주고 Fig. 2-12(d)와 같이 임계점 B를 만나게 되면 셀 C2를 닫아주고 교차점M, N과 포장 좌표 B, G는 셀 C_2의 셀 경계점으로 정의한다. 이와 같이 동일한 방법으로 스위프 라인이 우측으로 이동하면서 Fig. 2-12(e), (f), (g), (h)와 같이 임계점 F, E C, D를 순차적으로 만나 셀 C_3, C_4, C_5, C_6을 생성한다. 스위프 라인 함수를 이용 포장에서 각 모서리 간의 연결성을 하나 식 판별하여 판별된 결과 따라 영역을 합치게 된다. 본 연구에서 오목 형태의 포장을 셀 분할하는 것이 목적이기 때문에 포장의 내각이 180도 보다 큰 임계점을 이용 새로운 임계점으로 사용하여 셀 분할을 다시 한다. Fig. 2-12(i)와 같이 임계점 E의 내각은 180도 보다 큰 점이며 임계점 E에서 생성된 경계선을 기준으로 셀 C_1과 C_2로 셀 분할하여 각 셀에 대한 경로를 생성 후 각 영역의 경로를 하나의 집합을 만든다. 본 연구에서 포장의 내각이 180도 보다 큰 임계점이 5개 이상의 경우 개발한 포장 분할법을 적용 어려움이 있으므로 분할 어려움이 있는 복잡한 포장형태에서는 분할하지 않고 바로 경로생성 알고리즘을 적용하는 것을 고안하였다.



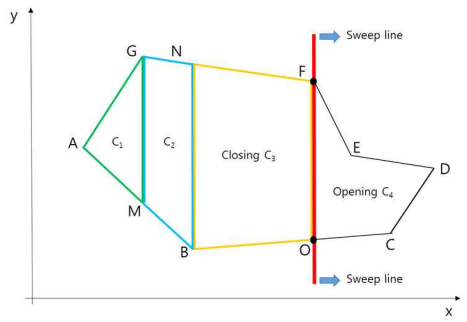
(a) (b)


y
Sweep line
G
F
A
Closing
C_1
Opening C_2
E
D
M
C
B
Sweep line
x

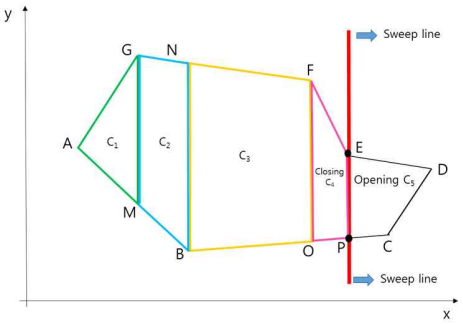
(c)


y
Sweep line
G
N
F
A
C_1
Closing
C_2
Opening C_3
E
D
M
C
B
Sweep line
x

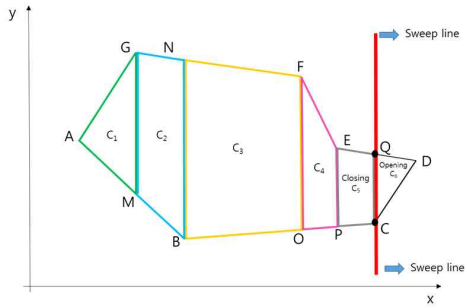
(d)


y
Sweep line
G
N
F
A
C_1
C_2
Closing C_3
E
D
Opening C_4
M
B
O
C
Sweep line
x

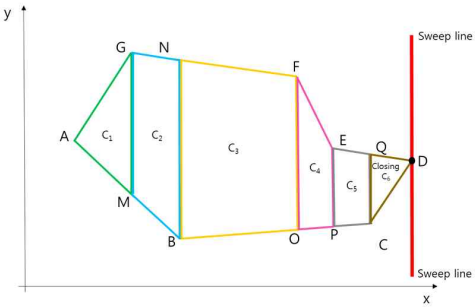
(e)


y
Sweep line
G
N
F
A
C_1
C_2
C_3
E
Closing
C_4
Opening C_5
D
M
B
O
P
C
Sweep line
x

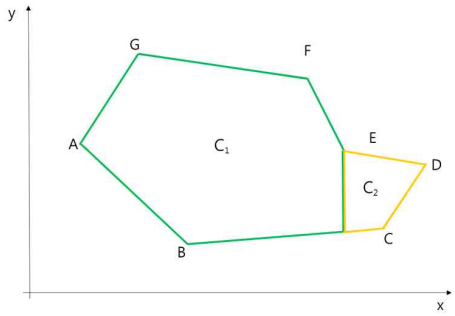
(f)


y
Sweep line
G
N
F
A
C_1
C_2
C_3
E
Q
D
C_4
Closing
C_5
Opening
C_6
M
B
O
P
C
Sweep line
x

(g)


y
Sweep line
G
N
F
A
C_1
C_2
C_3
E
Q
C_4
C_5
Closing
C_6
D
M
B
O
P
C
Sweep line
x

(h)



(i)

Fig. 2-12 Cell creation for a concave shaped field

MBR법을 사용하여 내부경로생성과 새머리 영역의 외부 회경 작업 경로생성과 포장의 셀 분할 절차는 Fig. 2-13과 같다. 다변형태 포장 정보를 입력받으면 스위프 라인을 이용 좌측으로부터 우측으로 이동하여 임계점을 찾게 된다. 임계점이 발생 경우 하나의 독립 셀을 생성 후 MBR법을 사용하여 내부경로를 생성한다. 임계점이 발생하지 아닌 경우 바로 MBR법을 사용하여 내부 경로를 생성한다. 셀 C_i에서 경로 생성 완료하면 셀 C_{i+1}로 이동하게 되며 아닌 경우 해당 셀C_i에서 경로생성 계속진행 한다. 해당 셀에서 경로생성 완료 후 다른 셀로 이동시켜 모든 셀에서 동일한 작업을 진행하여 주행 경로생성을 완성한다. 또한 본 연구에서는 Fig. 2-14와 같이 다변형 포장에서 경로생성을 위해 사용자가 포장을 분할 유무에 대해 선택가능하게 구성한다. 포장을 분할하는 경우 앞서 제시한 포장 분할법을 사용 후 각각 서브 필드에서 다변형 경로생성 알고리즘을 적용하여 최종 통합된 경로 점을 생성하게 된다. 그러나 포장을 분할하지 않는 상황에서는 전체 포장을 하나로 인식 받아 바로 경로생성 알고리즘을 적용하여 주행 경로를 생성한다.

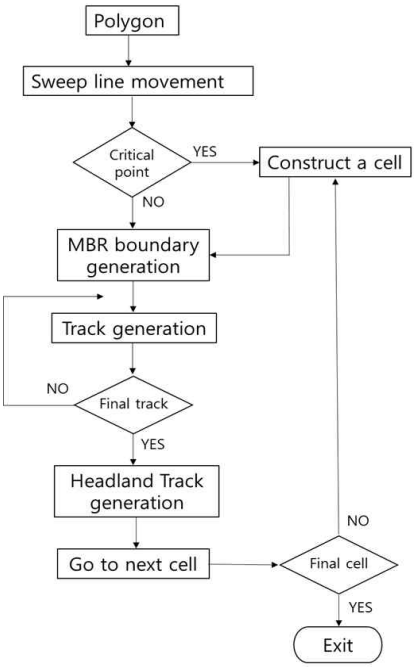


Fig. 2-13 Flowchart for field decomposition

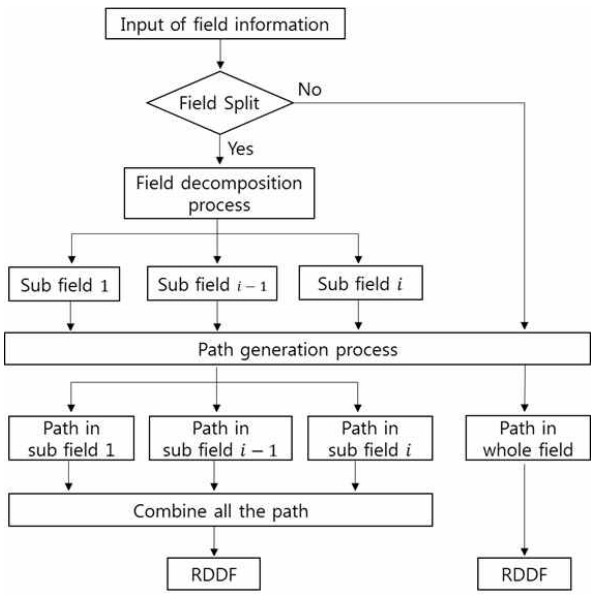


Fig. 2-14 Path generation in the sub field or a whole field

2.1.7. 무인 농작업 선회 경로유형 분석 및 수학적 모델링

2.1.7.1. 선회경로 경로유형 분석

트랙터를 이용하여 포장에서의 농작업은 작업별로 포장의 형태와 조건에 따라 작업순서가 다르게 된다. 예를 들면, 경운작업의 경우 관련 선행연구 결과(서일환, 2010)에 따르면 다음의 3가지 형태로 구분할 수 있다.

○ C형 선회패턴

Fig. 2-15(a)는 C형 농작업 유형을 나타낸 것으로 왕복작업하는데, 직진 작업을 미친 뒤 선회하여 인적 새머리 영역을 작업하지 않고 작업기 폭의 몇 배로 건너뛰어 작업하면서 하나의 구역을 작업하는 방법이다. C 형 작업은 포장의 회경 작업 영역을 제외한 전체를 진행 부분과 회귀부분으로 나누어 작업이 이루어지며, 진행부분에서 회귀부분으로 이동시 트랙터의 진행방향이 바뀜이 없이 연속적인 전진주행으로 작업이 이루어진다.

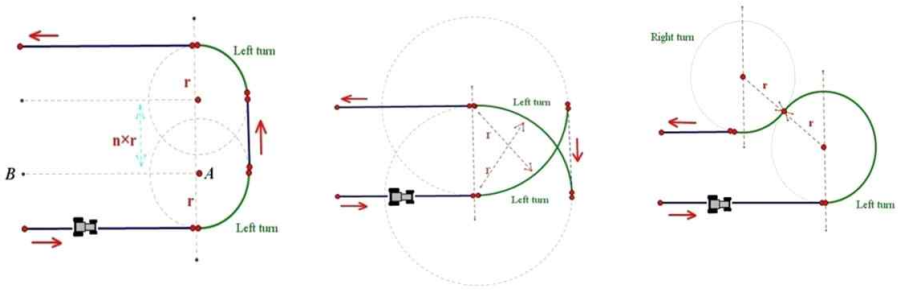
진행부분에서 회귀부분으로 진입 또는 회귀부분에서 진행부분으로 진입 시 내부 왕복부분의 1/2거리를 이동하여 진입하며, 트랙터의 작업기는 올려진 상태로 이동하고, 직진 작업 시 작업기를 하강하여 작업이 이루어진다. 이러한 C형 작업 경로는 왕복작업부과 회귀작업부의 진행방향은 후진이 없는 주행방법으로 연속적으로 작업이 이루어진다.

○ X형 선회패턴

작업 형태는 Fig. 2-15(b)와 같이 진입작업 후 전진작업과 회귀작업이 서로 인접하여 이루어지는 순차경법으로 작업이 이루어지며 왕복작업부에서 회귀작업부로의 이동 방법이 X형 선회 방법으로 선회하게 된다. 새머리부에서의 선회는 좌회전 다음 후진 그리고 좌회전의 순서로 이루어지며, 작업기는 상승하여 이동하게 된다. 이러한 방법은 작업자가 다른 작업부로 이동하기 위하여 트랙터의 좌회전, 후진, 좌회전 순으로 변경해야 하는 특징이 있다.

○ R형 선회패턴

작업순서는 Fig. 2-15(c)와 같이 왕복작업부분과 회귀작업부분이 서로 붙어 있으며 X형 작업과 동일하게 순차경법으로 작업이 진행되지만 진행부분과 회귀부분간의 이동시 선회는 R형태의 선회를 실시하고 있으며 트랙터 작업의 진행방향이 바뀌지 않고 연속적인 작업이 가능하지만 선회 시 핸들을 왼쪽과 오른쪽으로 연속하여 변경하여야 하는 특징이 있다.



(a) C-type turning (b) X-type turning (c) R-type turning

Fig. 2-15 Headland turning patterns in a paddy field

위와 같이 트랙터를 이용한 농작업에 따라 다양한 선회유형이 적용될 수 있는데, 농약살포, 써레질 등의 다른 농작업에 적용되는 선회유형별 농작업 경로 생성기술이 필요하다.

2.1.7.2. 작업 경로생성 수학적 모델 개발

트랙터가 포장에서 작업을 진행 시 Fig. 2-16과 같이 작업영역은 왕복작업부와 회경 작업부로 구분된다. 포장안쪽의 왕복작업부에서는 작업의 진행방향은 왕복작업에서 회귀작업으로 또는 회귀작업에서 왕복작업으로 경로 이동 시 트랙터가 180도를 회전하게 된다. 포장 주행 경로를 생성하기 위해 포장 4개 꼭지점의 좌표정보를 반 시계방향으로 입력도록하며 자동으로 입력된 점을 기반으로 포장의 길이(F_L)와 너비(F_W)의 계산이 가능하다. 포장의 꼭지점의 좌표

는 Google Earth에서 제공된 좌표점을 이용가능하며 더욱 정밀한 주행 경로를 생성위해 RTK-GPS를 이용 포장 4개 꼭지점을 측정하여 입력한다.

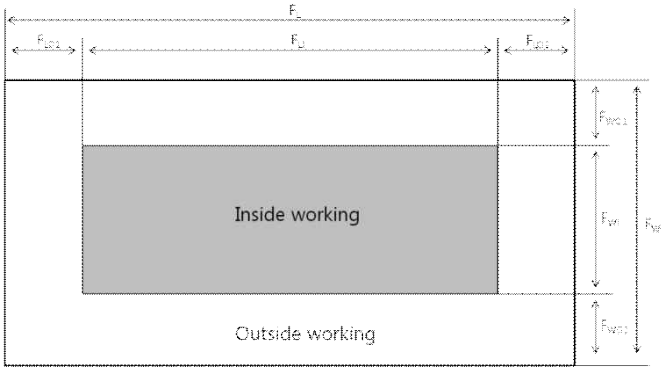


Fig. 2-16 Separation of working area in the test field

Fig. 2-17과 같이 C형 적업경로는 직진과 회전 경로로 구성되었으며 작업기 폭 크기를 고려하여 이동시켜 왕복작업경로를 생성한다.

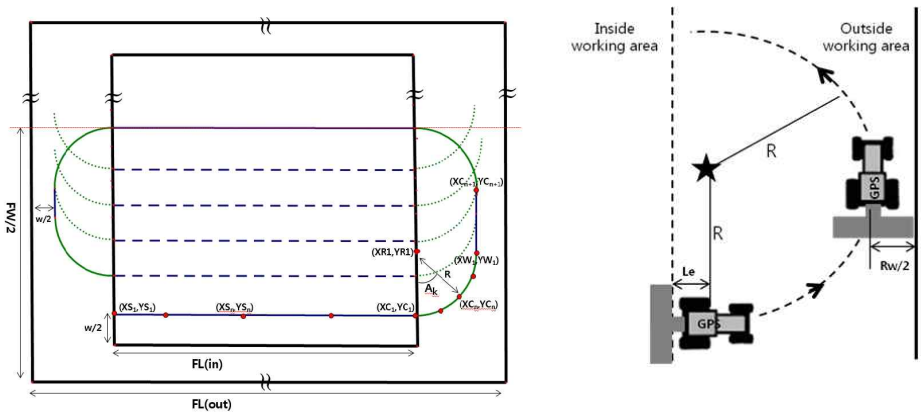


Fig. 2-17 Diagram of C-type turn on the straight lines and circular arcs

포장 변수, 트랙터 회전반경, 작업기 폭, 작업 겹침 량 및 GPS과 작업기의 중심위치간의 거리를 이용 Eq. (2-16), Eq. (2-17)와 같이 트랙터의 직진 시작점 XS_1, YS_1과 직진 종료점 XC_1, YC_1를 계산한다. X형과 R형의 작업경로에

서 같은 방법으로 직진 시작점과 종료점을 결정한다.

$$\begin{cases} XS_1 = R + \frac{W}{2} + L_e - \frac{W_r}{2} \\ YS_1 = W \times (0.5 + M) - W_o \times M \end{cases} \quad (2\text{-}16)$$

$$\begin{cases} XC_1 = FL_{(out)} - (R + \frac{W}{2}) \\ YC_1 = YS_1 \end{cases} \quad (2\text{-}17)$$

여기서, R= 회전 반경 (m)
W= 작업기 폭 (m)
W_o = 작업 겹침 량 (m)
$FL_{(out)}$ = 포장 장변 길이 (m)
W_r = 로터리 날의 반경 (m)
M= 회경 횟수
L_e = GPS와 작업기 중심간의 길이(m)

트랙터가 새머리 영역에서 선회주행 시 사용한 회전 중심 좌표 XR_1, YR_1과 회전 종료점 XW_1, YW_1의 결정은 Eq. (2-18), Eq. (2-19)와 같음. 트랙터가 몇 배의 폭을 건너뛰어 XC_{n+1}, YC_{n+1} 회전 시작점을 사용해 회전하여 왕복작업부에 진입하게 된다. X형 경로에서 동일한 방법으로 첫 번째 회전 중심과 종료점을 결정한다.

$$\begin{cases} XR_1 = XC_1 \\ YR_1 = YC_1 + R \end{cases} \quad (2\text{-}18)$$

$$\begin{cases} XW_1 = XR_1 + R \\ YW_1 = YR_1 \end{cases} \quad (2\text{-}19)$$

트랙터 작업주행의 직진 경로점 XS_n, YS_n 의 결정은 Eq. (2-20)과 같다. 사용자 원하는 직진 경로점 간격을 선택하면 직진 구간의 경로점이 생성하며

X, R형의 작업경로에서 같은 방법으로 직진 경로점 간격을 결정한다.

$$\begin{cases} XS_n = XS_1 + \dfrac{k(XC_1 - XS_1)}{D_{i.nterval}} \\ YS_n = YS_1 \end{cases} \tag{2-20}$$

여기서, $D_{i.nterval}$ = 직진 경로점 간격 (m)

k = 실수 (0, 1, 2.....)

트랙터가 새머리 영역에서 선회주행 시 선회 경로점의 결정은 그 다음 직진 경로로 진입에 영향을 준다. 선회의 시작점은 직진의 종료점을 이용하고 선회의 종료점은 트랙터 선회반경과 작업기 폭으로 결정한다. 또한 선회 경로점의 간격은 트랙터의 제원에 따라 변경 가능하며 90° 의 배수(45°, 30°, 22.5°)로 입력하여 선회 경로점을 도출한다. 선회 경로점은 Eq. (2-21), Eq. (2-22)와 같이 결정하며 X, R형의 선회주행 경로점도 동일한 방법으로 결정하게 된다.

$$\begin{cases} XC_n = XR_1 + R \times \cos(A_k) \\ YC_n = YR_1 + R \times \sin(A_k) \end{cases} \tag{2-21}$$

$$\begin{cases} \theta_1 = \dfrac{\tan^{-1}(YR_1 - YC_1)}{XR_1 - XC_1} \\ \theta_2 = \dfrac{\tan^{-1}(YR_1 - YS_{n+1})}{XR_1 - XS_{n+1}} \\ A_k = \theta_1 + \dfrac{k(\theta_2 - \theta_1)}{N_{i.nterval}} \end{cases} \tag{2-22}$$

여기서, $N_{i.nterval}$ = 회전 경로점 계수

A_k = 회전 경로점 간격 (°)

Fig. 2-18과 같이 X형 적업경로는 직선과 회전 경로로 구성되었으며 선회는 좌회전 다음 후진 그리고 좌회전의 순서로 이루어진다.

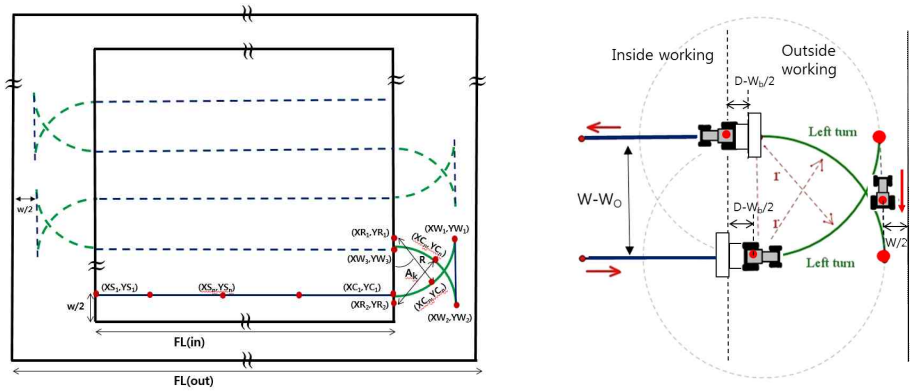


Fig. 2-18 Diagram of X-type turn on the straight lines and circular arcs

트랙터가 Headland에서 선회주행 시 첫 번째 선회경로에서 사용한 회전 중심은 C형과 동일하며 두 번째 선회경로에서 사용한 회전 중심은 XR_2, YR_2과 회전 시작점 XW_2, YW_2 및 종료점 XW_3, YW_3은 Eq. (2-23), Eq. (2-24), Eq. (2-25)와 같이 결정한다. 트랙터가 XW_3, YW_3 좌표를 이용 내부왕복작업의 시작점으로 사용한다.

$$\begin{cases} XR_2 = XR_1 \\ YR_2 = XC_1 - (R - (W - W_o) \end{cases} \quad (2\text{-}23)$$

$$\begin{cases} XW_2 = XR_2 + R \\ YW_2 = YR_2 \end{cases} \quad (2\text{-}24)$$

$$\begin{cases} XW_3 = XR_2 \\ YW_3 = YR_2 + R \end{cases} \quad (2\text{-}25)$$

Fig. 2-19와 같이 R형 작업의 진행방향은 후진 없이 연속적인 작업이 가능하지만 선회 시 핸들을 왼쪽과 오른쪽으로 연속하여 변경하여 주행경로를 생성한다.

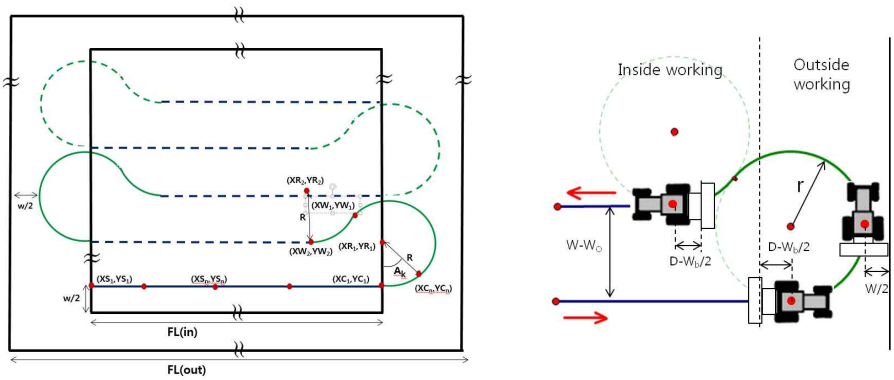


Fig. 2-19 Diagram of R-type turn on the straight lines and circular arcs

트랙터가 새머리 영역에서 선회주행 시 첫 번째 선회경로에서 사용한 회전 중심 좌표 XR_1, YR_1과 회전 종료점 XW_2, YW_2의 결정은 Eq. (2-26), Eq. (2-27)와 같다.

$$\begin{cases} XR_1 = XC_1 \\ YR_1 = YC_1 + R \end{cases} \tag{2-26}$$

$$\begin{cases} L = \sqrt{(2R)^2 - (W - W_o)^2} \\ XW_2 = XR_1 - L \\ YW_2 = YR_1 - (W - W_o) \end{cases} \tag{2-27}$$

트랙터가 새머리 영역에서 선회주행 시 두 번째 선회경로에서 사용한 회전 중심 XR_2, YR_2와 회전 시작점 XW_1, YW_1은 Eq. (2-28), Eq. (2-29)와 같이 결정한다. 트랙터가 XW_1, YW_1 좌표까지 좌회전 후 우회전하여 XW_2, YW_2 목표점으로 주행 후 내부왕복작업부로 진입한다.

$$\begin{cases} XR_2 = XW_2 \\ YR_2 = YW_2 + R \end{cases} \tag{2-28}$$

$$\begin{cases} XW_1 = (XR_1 + XR_2)/2 \\ YW_1 = (YR_1 + YR_2)/2 \end{cases} \tag{2-29}$$

2.1.8. 다변 형태 포장에서 새머리 선회경로 및 Coverage Cost

농지의 작업 경로는 일반적으로 작업자의 습관 및 경험과 상식적인 규칙에 따라 선택을 하게 된다. 적합한 경로 계획은 농작업의 질을 높일 수 있으며 에너지 소비도 절감이 가능하다. 새머리 선회경로는 Fig. 2-20과 같이 다변형태 농지에서 작업의 특성 및 필요에 따라 적합한 작업 주행 방향 (θ)값에 따라 여러 가지로 구별하게 되며 아래와 같이 경로 적합한 주행 방향을 위해 각각 목표를 제시하였다.

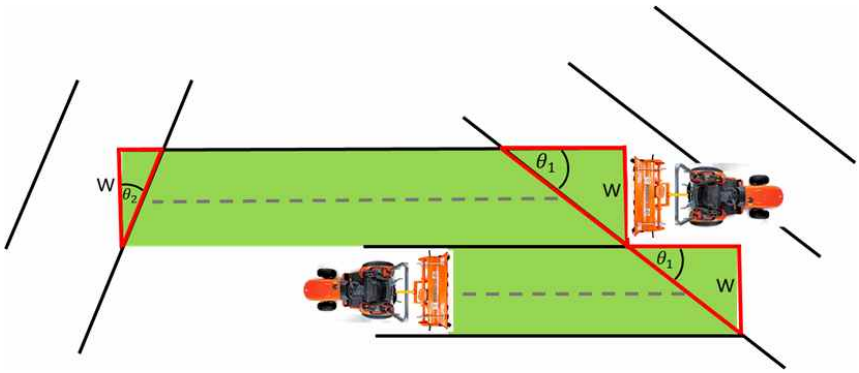


Fig. 2-20 Headland turning in a polygonal field

자율주행 트랙터의 포장 작업효율은 새머리 영역에서의 최소 선회 소모 시간으로 결정 된다. 계획된 경로에 대해 포장 작업효율을 개선하기 위해 선회 소모 시간의 감소화 작업이 필요 하다. 트랙터가 새머리 영역에서 일정한 속도 (v)로 주행하는 것을 가정하며 X, R형 두 가지 선회 패턴에 대해 작업 소모 시간을 계산하였다.

직교형태 포장의 경우는 포장 내부 왕복 작업 경로생성 후 포장 내부경계선에서 바로 회전 경로를 생성하게 된다. 그러나 다변형태 포장의 경우는 직진 경로와 포장 내부경계선이 서로 직교가 아닌 각도로 구성되어 있기 때문에 Fig. 2-21과 같이 구간을 나누어 회전경로를 생성하였다.

작업기가 포장 내부 경계선상의 a점에 도달하면 미경지 없이 완전한 직진 작업을 수행하기 위해 작업기가 b점까지 작업 상태로 주행 후 회전을 시작한다. 내부 경계선상의 a점과 e점은 앞서 설명한 MBR법을 기반의 직진경로생성 알고리즘에서 구성하게 되며 a점과 b점의 거리는 Eq. (2-30)과 같다.

$$|ab| = 0.5(W - W_o)\cot(\theta) \tag{2-30}$$

여기서, $\theta = \theta_{path} - \psi_i$

θ_{path} = 포장 내부 직진주행 방향 (°)

ψ_i = 포장 내부 i번째 경계선의 방향 (°)

W = 작업기 폭 (m)

W_o = 작업 겹침 량 (m)

작업기가 b점까지 도달하면 새머리 영역으로 완전히 진입된 상태이며 충분한 회전 공간을 확보를 위해 작업기가 비 작업 상태로 b점에서 c점까지 주행한다. b점과 c점의 거리는 Eq. (2-31)과 같다.

$$|bc| = (W - W_o)\cot(\theta) \tag{2-31}$$

작업기가 c점에 도달하면 선회경로를 통해 d점까지 주행하게 되며 d점으로부터 작업 상태로 e점까지 주행하여 왕복작업부로 진입하게 된다. 여기서 c점에서 d점까지의 선회 경로는 앞서 설명한 X, R형 선회 경로의 수학적 모델을 사용하게 되며 d점과 e점의 거리는 Eq. (2-32)과 같다.

$$|de| = 0.5(W - W_o)\cot(\theta) \tag{2-32}$$

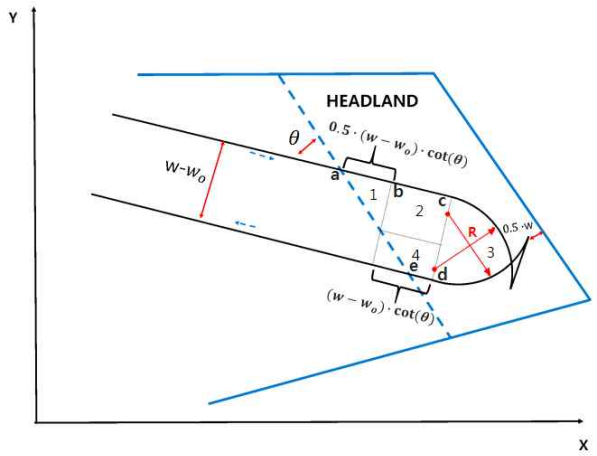


Fig. 2-21 Configuration of headland turning in the polygonal field

다변형 포장에서 R형 회전은 Fig. 2-22와 같이 작업기가 A점까지 도달하면 작업 완료지점이며 완전한 작업을 수행하기 위해 작업기가 작업 진행하는 상태로 A점부터 E점까지 직진 주행한다. 작업기가 작업 정지 상태로 E점에서 F점까지 α각도를 회전 후 F점부터 β각도로 반대방향으로 회전 하여 C점까지 회전 주행한다. 작업기가 C점부터 G점까지 작업 진행하는 상태로 왕복작업부로 진입하게 된다.

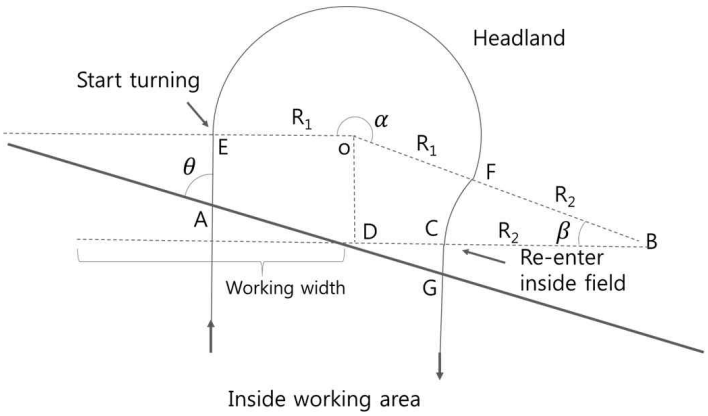


Fig. 2-22 R-type path headland turning in the polygonal field

여기서 R2는 R1보다 크며 R2의 정의는 Eq. (3-33)와 같으며 β는 Eq. (3-34)와 같다. 또한 $\alpha=\beta+180$ 때문에 회전 소모 시간은 1, 2, 3, 4 구역에서 사용한 총 회전 소모 시간은 Eq. (3-35)과 같다.

$$R_2 = \frac{W^2\cot^2\theta + W^2 - 2WR_1}{4R_1 - 2W} \tag{2-33}$$

$$\beta = \sin^{-1}\frac{4R_1 W\cot\theta - 2W^2\cot\theta}{4R_1^2 - 4WR_1 + W^2\cot^2\theta + W^2} \tag{2-34}$$

$$\begin{aligned} Cost_{turn} &= \frac{R_1\times\alpha + R_2\times\beta}{v} \\ &= \frac{R_1\pi + \frac{4R_1^2 - 4WR_1 + W^2\cot^2\theta + W^2}{4R_1 - 2W}\times\sin^{-1}\frac{4R_1 W\cot\theta - 2W^2\cot\theta}{4R_1^2 - 4WR_1 + W^2\cot^2\theta + W^2}}{v} \end{aligned} \tag{2-35}$$

여기서, θ = 직진주행 각도와 포장 경계선의 각도 차이 (°)

R_i = 선회반경 (m)

v = 선회속도 (m/s)

W = 작업기 폭 (m)

α = E점에서 F점까지 회전각도 (°)

β = F점에서 C점까지 회전각도 (°)

다변형 포장에서 X형 회전은 Fig. 2-23과 같이 작업기가 A점까지 도달하면 작업 완료지점이며 완전한 작업을 수행하기 위해 작업기가 작업을 진행하는 상태로 A점부터 B점까지 직진 주행하고 작업기가 작업 정지 상태로 B점에서 E점까지 회전 주행한다. 작업기가 E점부터 F점까지 비 작업 상태로 직진 주행 후 F점부터 G점까지 작업 진행하는 상태로 왕복작업부로 진입하게 된다.

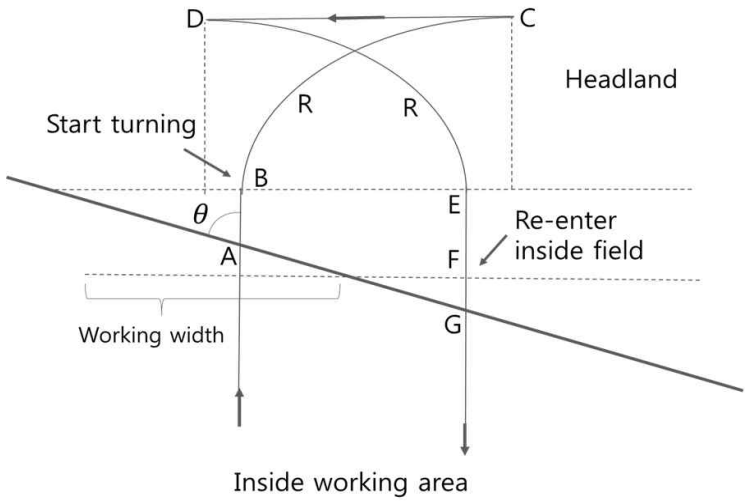


Fig. 2-23 X-type path headland turning in the polygonal field

회전 소모 시간은 1, 2, 3, 4 구역에서 사용한 총 회전 소모 시간은 Eq. (2-36)과 같다.

$$Cost_{turn} = \frac{\pi \times R + 2W \times \cot(\theta) + 2R - W}{v} \tag{2-36}$$

여기서, θ = 직진주행 각도와 포장 경계선의 각도 차이 (°)

R = 선회반경 (m)

v = 선회속도 (m/s)

W = 작업기 폭 (m)

2.2. 결과 및 고찰

2.2.1. 경로생성 프로그램 개발 결과

자율주행 트랙터의 농작업 경로생성을 위한 시스템의 전체적인 구성 결과는 Fig. 2-24와 같다. 앞서 개발한 농지형태에 따른 주행 경로생성 알고리즘을 적용하여 Fig. 2-25와 같이 LabVIEW® (Ver. 2015)를 이용 자율주행 트랙터 농작업을 수행하기 위해 경로생성 프로그램을 개발하였다. 개발한 경로생성 프로그램은 직교형태 포장에서 C형, X형, R형 세 가지 선회 유형별 경로생성이 가능하며 다변형태 포장에서 X형, R형 두 가지 선회 유형별 사용자가 선택할 수 있게 하였다.

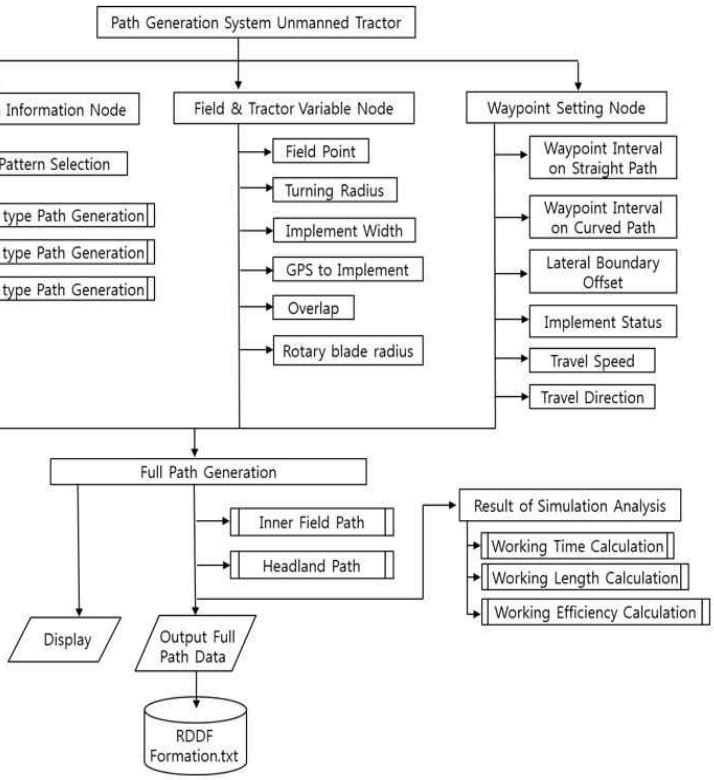


Fig. 2-24 Flowchart of the path generation program

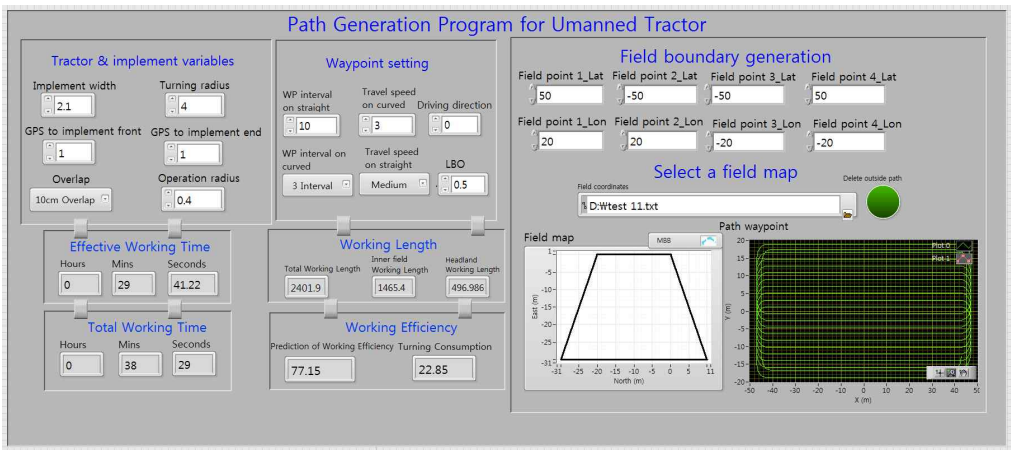


Fig. 2-25 Interface of the path generation program

Fig. 2-26은 포장정보를 받아 프로그램을 통해 포장형태를 출력해주는 과정이다. 포장의 작업영역은 왕복 작업부와 회경 작업부로 구분하였으며 직교형태 포장에서 경로생성 시 포장의 입력변수인 4개 꼭짓점의 경도/위도 TM좌표를 반시간 방향으로 입력 받는다. 다변형태 포장에서는 Text 파일로 구성된 포장의 TM좌표 정보를 반시계 방향으로 입력 받아 포장의 길이, 폭 및 면적을 계산하고 포장형태를 사용자가 알아 볼 수 있도록 지도로 출력하였다.

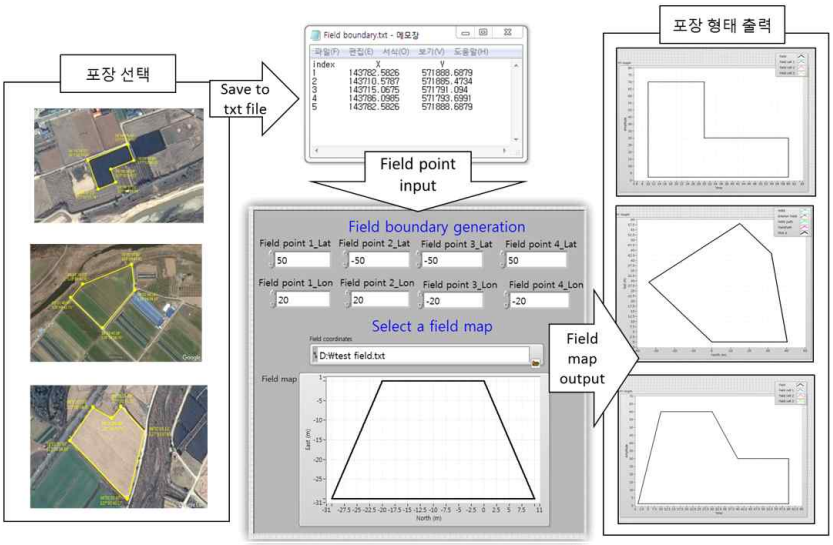


Fig. 2-26 Field boundary generation by field variables input

트랙터와 작업기의 주요수치는 여러 가지로 표시 가능하지만 본 연구에서 몇 가지 경로생성에 필요한 주요수치를 트랙터 및 작업기 변수로 시용하였다. 트랙터 및 작업기의 주요인자 중 트랙터의 회전 반경, GPS 설치 위치와 작업기까지의 거리, 작업기 폭 및 작업의 겹침 량을 입력변수로 설정하였다. 로터리 작업인 경우 로터리 날의 반경도 입력하도록 구성하였다. Fig. 2-27은 트랙터 및 작업기 변수를 입력하는 화면을 나타낸 것이다. 변수를 한번 입력되면 저장이 되며 다른 모델의 트랙터 혹은 작업기를 사용 시 기계의 제원을 이용 변수설정 프로그램을 실행하는 것이 가능하다.

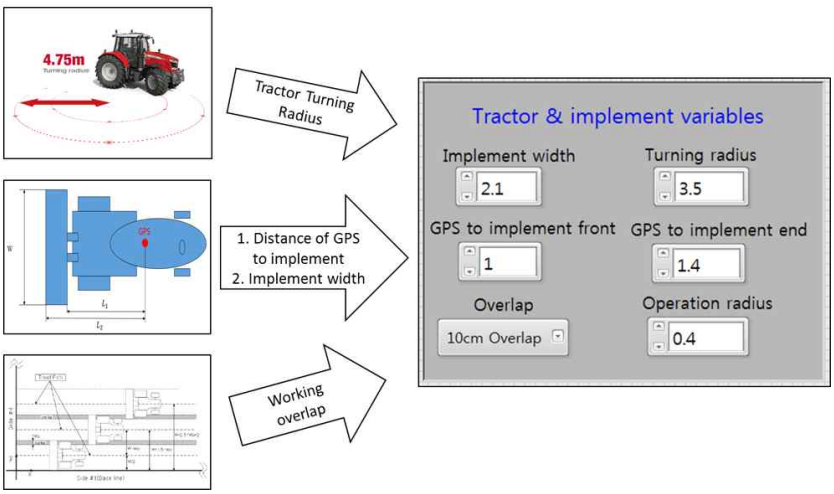


Fig. 2-27 Input variables of tractor and implement

경로점을 생성하기 위해 직진과 회전의 경로점 간격, 경로점 오차 범위, 작업영역의 작업 속도, 비 작업영역의 작업 속도 및 주행 방향 정보들을 사용자가 선택하면 포장의 내부와 외부를 포함된 주행경로를 생성한다. Fig. 2-28과 같이 프로그램에서 제시한 포장 작업시간, 거리 및 효율을 기반으로 적합한 주행 경로를 작업자가 선정이 가능하드록 하였다.

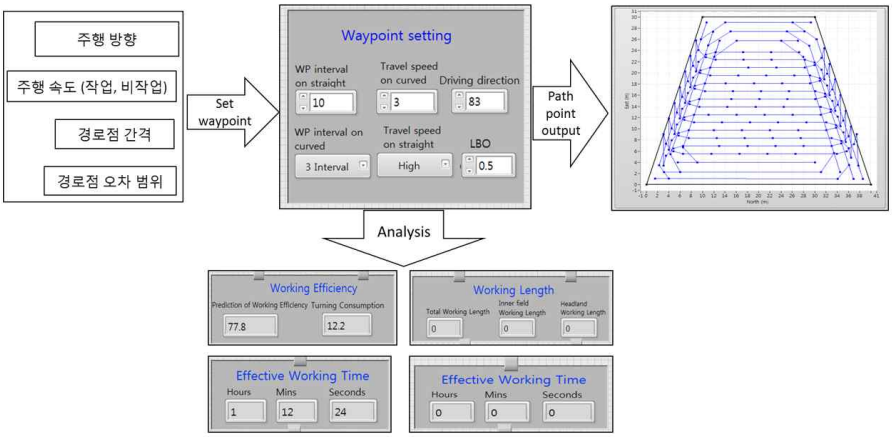


Fig. 2-28 Input variables of waypoint setting and output of path analysis

생성된 경로는 자율주행 트랙터가 포장에서 주행 및 작업을 수행할 수 있도록 주행 목표 경도 위도 좌표, 경로점 오차범위, 주행 속도, 작업기의 상승/하강 상태 정보를 포함된 확장형 RDDF 파일을 도출하여 경로추종에 사용하도록 하였다.

2.2.2. 포장 면적 변화에 따른 경로생성

본 연구에서 개발한 경로생성 프로그램을 이용하여 포장의 크기별로 일정한 방향으로 생성된 작업경로의 포장 작업효율을 시뮬레이션을 통해 분석하였다. 트랙터와 작업기의 변수는 Table 2-3과 같이 이용하였으며 표준구획 0.4 ha (100 m×40 m)에서 경로생성 결과는 Fig. 2-29와 같다. Fig. 2-29(a)는 C형 선회패턴으로 생성된 작업 경로를 나타낸 것이고, Fig. 2-29(b)는 X형 선회패턴으로 생성된 작업 경로를 나타낸 것이고, Fig. 2-29(c)는 R형 선회 패턴을 사용 생성된 작업 경로를 나타낸 것이다.

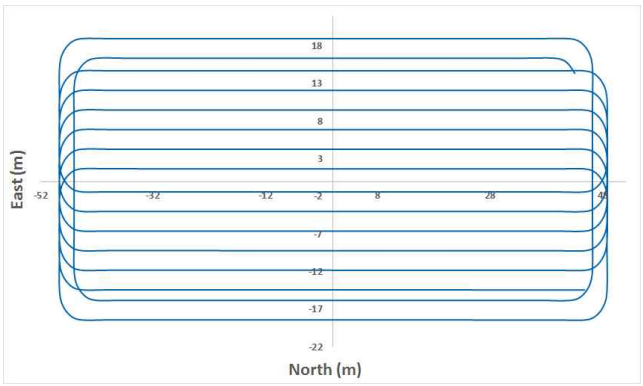
포장의 크기는 한국의 표준구획 0.4 ha (100 m×40 m)과 0.3 ha (100 m×30 m), 0.5 ha (100 m×50 m), 0.6 ha (100 m×60 m), 0.7 ha (100 m×70 m) 5가지로 선정하여 각각 포장 모형에서 C형, X형, R형 선회 패턴별로 사용하여 경로를 생성 하였다. Fig. 2-30은 포장구획 변화에 대한 3가지 선회 패턴을 사용 포장 작업효율을 나타낸 것이다. 포장의 단변 길이를 30, 40, 50, 60 ,70 m로 증가시킴에 따라 3가지 선회 패턴을 사용한 포장 작업효율은 모두 감소한 것을 나타내고 있다. 여기서 C형 선회 패턴을 사용한 작업 경로의 포장 작업효율은 90.99%에서 76.83%로 상대적 급격히 감소하는 것이 나타내고 있다. 이는 C형 선회 패턴 경로가 왕복작업부에서 직진 작업을 마친 뒤 선회하여 인접하게 다음 직진 작업경로로 진입하지 않고 작업기 폭의 몇 배로 건너뛰어 작업하게 되므로 비 작업 선회주행에 필요한 주행 거리는 X형, R형 선회 패턴 보다 길어졌음을 의미한다. 즉, 포장의 단변이 증가함에 따라 왕복작업 수

가 증가되어 C형 선회 패턴 개수도 같이 증가한다. 이러한 방식은 비 작업시간이 많이 발생하여 포장 작업효율도 급격히 감소하는 현상을 설명함을 의미한다.

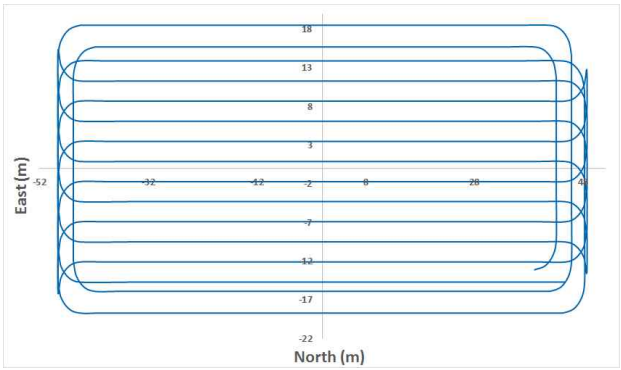
포장의 단변이 30 m일 경우 C형, X형, R형 선회 패턴 경로의 포장 작업효율은 각각 92.99%, 88.75%, 86.49%로 보이며 포장 단변이 40 m부터 증대됨에 따라 X형의 포장 작업효율은 87.37%, 86.14%, 85.46%, 85.05%로 C형과 R형 선회 패턴 경로보다 상대적 큰 포장 작업효율로 보인다. 즉 포장 단변이 40 m 이하의 소 구획 포장에서 C형 선회 패턴 경로를 사용하는 것이 적합하며 포장 단변이 40 m 이상의 경우 X형 선회 패턴 경로를 사용하는 것이 포장 작업효율 측면에서 유리한 것을 확인하였다.

Table 2-3 Input variables of tractor and implement used in the study

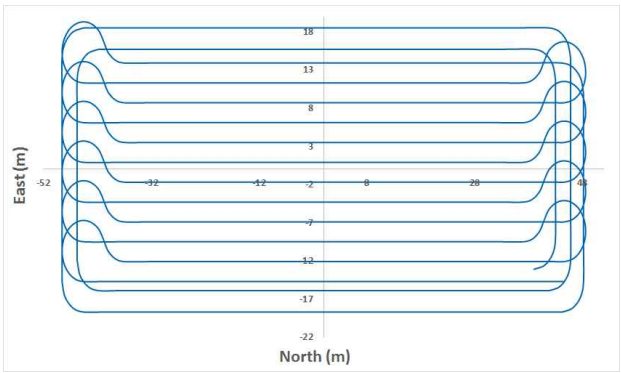
Items	Values
Manufacturer	TongYang
Model	TX803
Distance from front to rear axis	2.3 m
Distance from GPS to front axis	1.8 m
Distance from GPS to implement axis	1.3 m
Minimum turning radius	3.5 m
Width or implement	1.9 m
Width of working overlap	0.2 m



(a) C-type pattern path



(b) X-type pattern path



(c) R-type pattern path

Fig. 2-29 Result of the path generation in the standard size field

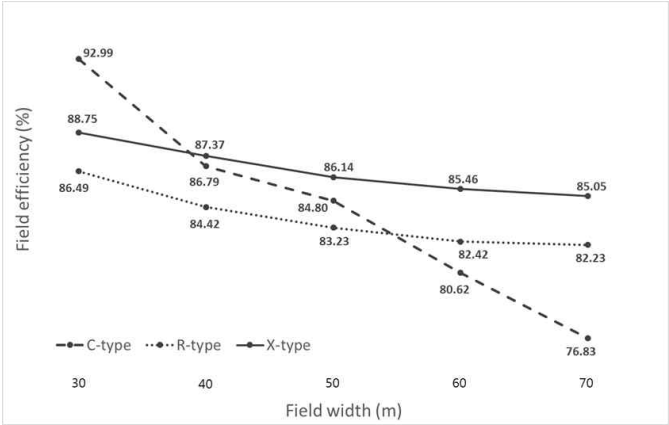


Fig. 2-30 Comparison of the field efficiency by changing the field size

2.2.3. 다변형태 포장에서 주행 진행 각도 변화에 따른 경로생성

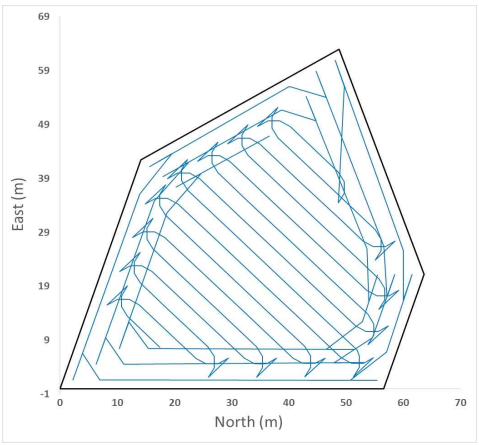
개발된 경로생성 프로그램을 이용 다변형태 포장에서 주행의 방향을 변경하여 트랙터의 포장 작업시간, 포장 작업거리 및 포장 작업효율 등을 시뮬레이션을 이용 분석하였다. 여기서 선회 방식은 X형 선회 패턴으로 고정하여 연구를 진행하였다. 트랙터와 작업기의 변수는 Table 2-3과 같이 설정하였으며 Fig. 2-31과 같은 볼록 형태의 포장에서 경로를 생성하였다. Fig. 2-32(a), 2-32(b), 2-32(c) 와 2-32(d)는 각각 0도, 65도, 120도, 170도 방향으로 생성된 작업경로를 나타낸 것이다. Table 2-4와 Fig. 2-33은 시뮬레이션을 실시하여 포장 작업시간, 유효 작업시간, 포장 작업거리, 포장 작업효율 등을 주행 방향별 나타낸 결과이다.

주행방향이 0도일 경우 트랙터의 포장 작업거리, 포장 작업시간과 포장 작업효율은 각각 1344 m, 1704 초, 77.8%로 분석 되었으며, 주행방향이 65도일 경우는 각각 1266 m, 1675 초, 83.9%로 나타나 주행 방향이 0도일 경우보다 포장 작업거리는 77 m 감소하였고, 포장 작업시간은 29 초 감소하였으며 포장 작업효율은 6.1% 증가하였다.

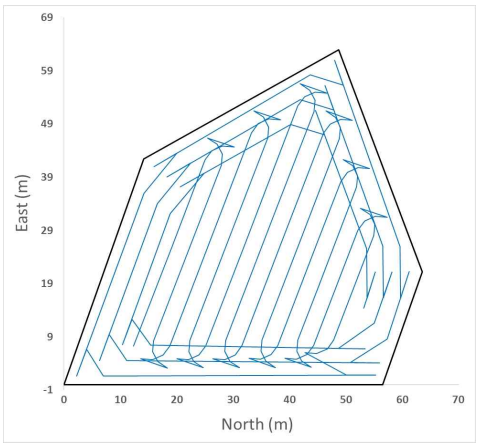
주행 방향이 0도일 경우 왕복 작업수가 16회이며 주행 방향이 65도에서는 12회로 작업을 수행하는 것으로 나타나, 왕복 작업수가 감소되면서 선회 소모 시간은 380 초에서 269초로 감소되어 작업효율이 높아지는 것으로 나타났다.

그러나 주행 방향이 65도에서 120도 및 170도로 증가하면 왕복작업 수가 높아지면서(12, 14, 16회) 작업거리(1266 m, 1290 m, 1342 m)가 증가하며 포장 작업효율은 각각 83.9%, 80.1%, 78.2%로 낮아진 것으로 나타났다. 즉 본 연구에서 사용한 볼록 형태의 포장에서는 65도 방향으로 주행경로를 생성 시 가장 효율적인 포장 작업효율을 얻을 수 있는 것으로 나타났다.

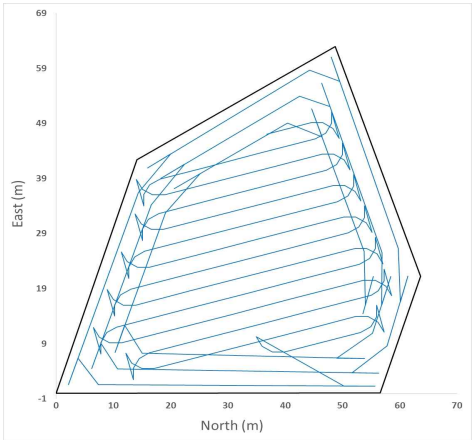
Fig. 2-31 Convex shaped polygonal field for coverage path generation



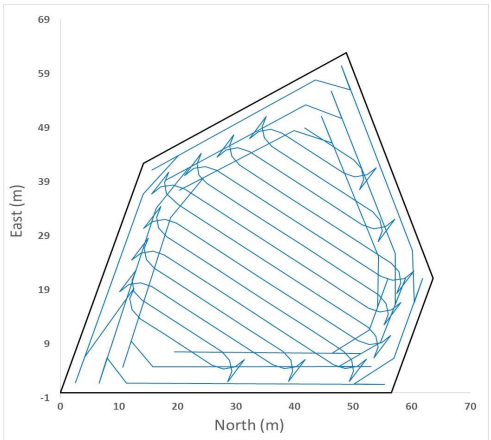
(a) 0 deg



(b) 65 deg



(c) 120 deg



(d) 170 deg

Fig. 2-32 Result of the path generation by changing the direction in convex shaped polygonal field

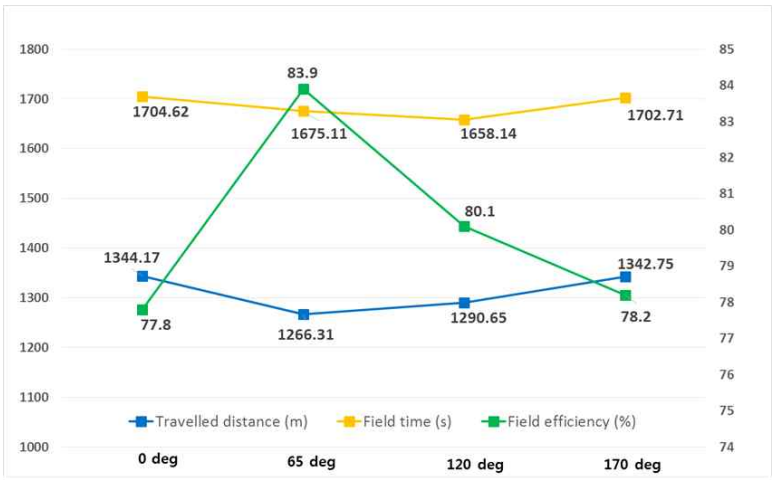


Fig. 2-33 Simulation analysis by changing the direction in convex shaped polygonal field

Table 2-4 Results of the simulation analysis by changing the direction in convex shaped polygonal field

Items	Values			
Driving angle (deg)	0	65	120	170
Travelled distance (m)	1344.17	1266.31	1290.65	1342.75
Effective field time (sec)	1323.87	1405.15	1327.88	1330.95
Field time loss in turning (sec)	380.74	269.96	330.26	371.76
Field time (sec)	1704.62	1675.11	1658.14	1702.71
Field efficiency (%)	77.8	83.9	80.1	78.2
Round numbers	16	12	14	16

개발된 경로생성 프로그램을 이용하여 오목 형태 포장 1에서 주행의 방향을 변경하면서 트랙터의 포장 작업시간, 포장 작업거리 및 포장 작업효율 등을 시뮬레이션 통해 분석하였다. Fig. 2-34와 같은 형태의 포장에서 주행 경로를

생성하기 위해 우선 포장 분할 작업하였다. 여기서 하나의 오목 형태의 포장을 두개의 볼록 형태 포장로 셀 분할 작업하여 Fig. 2-35와 같이 각각 포장에서 주행 경로를 생성하였다.

Fig. 2-35(a)는 subfield 1과 subfield 2에서 모두 90도 방향으로 생성된 작업 경로를 나타낸 것이고, Fig. 2-35(b)는 subfield 1에서 70도 방향, subfield 2에서 25도 방향으로 생성된 작업 경로를 나타낸 것이고, Fig. 2-35(c)는 subfield 1에서 90도 방향, subfield 2에서 45도 방향으로 생성된 작업 경로를 나타낸 것이고, Fig. 2-35(d)는 subfield 1에서 70도 방향, subfield 2에서 90도 방향으로 생성된 작업 경로를 나타낸 것이다. Table 2-5와 Fig. 2-36은 시뮬레이션을 실시하여 포장 작업시간, 유효 작업시간, 포장 작업거리, 포장 작업효율 등을 주행 방향별로 나타낸 결과이다.

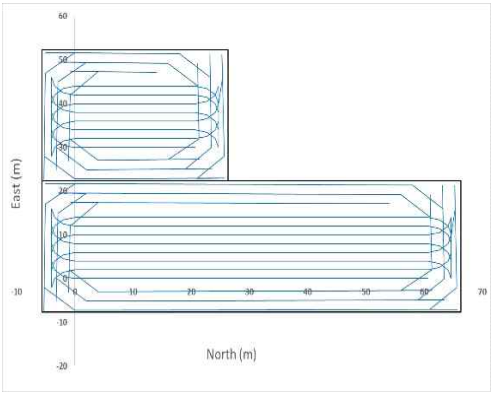
주행 방향이 Case 1일 경우 트랙터의 포장 작업거리, 포장 작업시간과 포장 작업효율은 각각 1935 m, 2375 초, 84.4%로 예측되었으며, 주행방향이 Case 2일 경우는 각각 1882 m, 2483 초, 80.2%로 조사되어 주행 방향이 Case 1일 경우보다 포장 작업거리는 52.5 m 증가하였고, 포장 작업시간은 108.1 초 감소하였으며 작업효율은 4.2% 증가하였다.

주행 방향이 Case 1일 경우 왕복 작업수가 16회이며 주행 방향이 Case 2에서는 20회로 작업을 수행하는 것으로 조사되었다. 주행 방향이 Case 1에서 포장 작업거리는 52.5 초 증가하였지만 왕복 작업수가 감소되면서 선회 소모시간이 120.5 초를 감소되어 유효 작업시간이 12.4 초가 높아지면서 포장 작업효율도 높다.

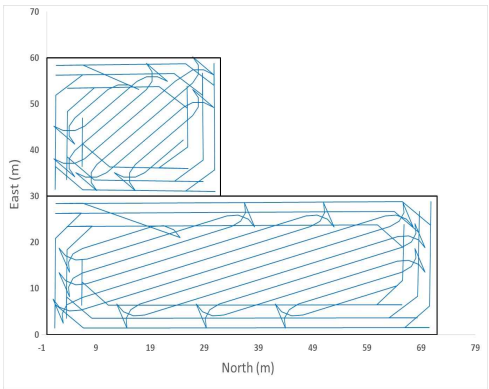
주행 방향이 Case 2에서 Case 3로 변화하면 왕복작업 수가 모두 20회에서 16회로 감소하였지만 포장 작업거리는 1882.94 m과 1887.36 m 크게 차이가 없는 것으로 조사되었다. 그러나 선회 소모시간은 112.24 sec가 감소하였기 때문에 포장 작업효율도 3.8% 증가하는 것으로 나타났다. 즉 본 연구에서 사용

한 오목 형태의 포장에서 Case 1 방향의 조합으로 주행 경로를 생성 시 가장 적합한 결과로 나타났다.

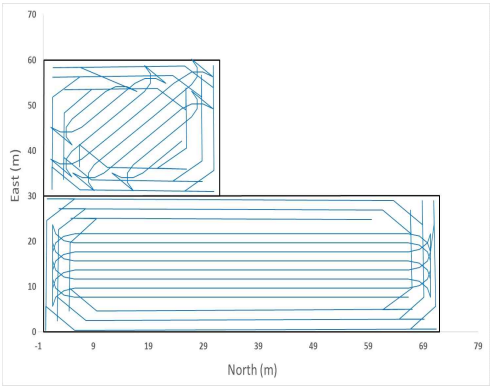
Fig. 2-34 Concave shaped polygonal field 1 for coverage path generation



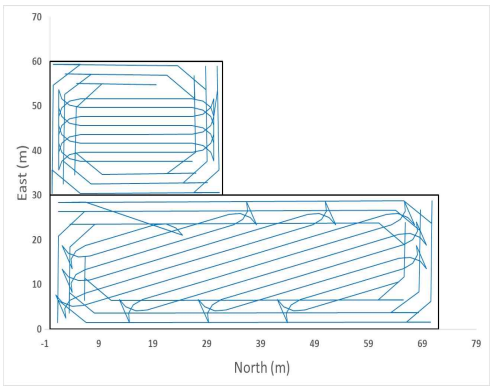
(a) Case 1



(b) Case 2



(c) Case 3



(d) Case 4

Fig. 2-35 Result of the path generation by changing the direction in concave shaped field 1

Table 2-5 Results of the simulation analysis by changing the direction in concave shaped polygonal field 1

Items	Values			
	Case 1	Case 2	Case 3	Case 4
Driving angle (deg)	90&90	70&45	90&45	70&90
Travelled distance (m)	1935.4	1882.9	1887.4	1930.9
Effective field time (sec)	2005.1	1992.6	1983.8	2013.8
Field time loss in turning (sec)	370.8	491.3	379.1	483.0
Field time (sec)	2375.8	2483.9	2362.9	2496.9
Field efficiency (%)	84.4	80.2	84.0	80.7
Round numbers	16	20	16	20

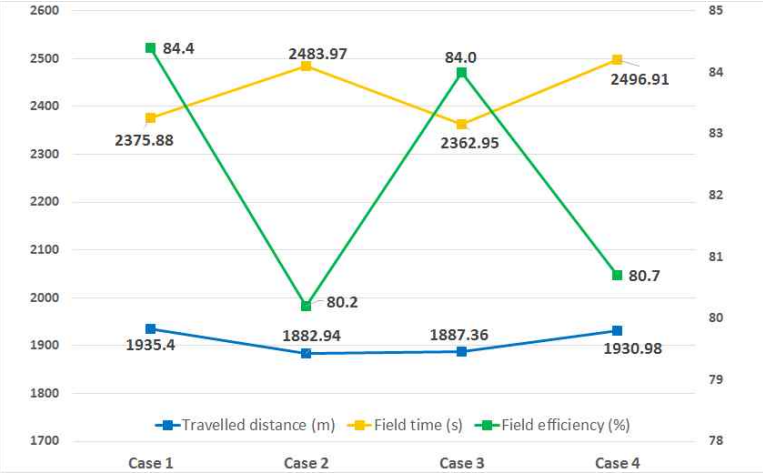


Fig. 2-36 Simulation analysis by changing the direction in concave shaped polygonal field 1

개발된 경로생성 프로그램을 이용하여 오목 형태 포장2에서 주행의 방향을 변경하면서 트랙터의 포장 작업시간, 포장 작업거리 및 포장 작업효율 등을 시뮬레이션 통해 분석하였다. Fig. 2-37과 같은 형태의 포장에서 주행 경로를 생성하기 위해 우선 포장 분할 작업하였다. 여기서 하나의 오목 형태의 포장을 두개의 볼록 형태의 포장을 셀 분할 작업하여 Fig. 2-38과 같이 각각 포장에서 주행 경로를 생성한다.

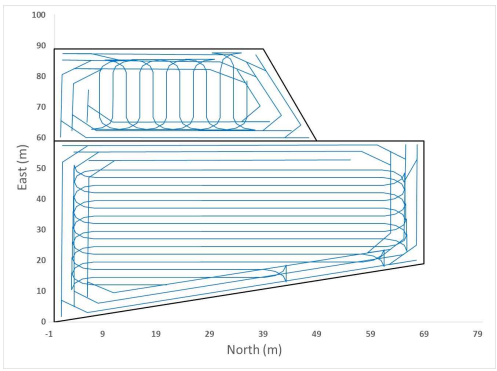
Fig. 2-38(a)은 subfield 1에서 90도 방향, subfield 2에서 0도 방향으로 생성된 작업 경로를 나타낸 것이고, Fig. 2-38(b)은 subfield 1에서 0도 방향, subfield 2에서 90도 방향으로 생성된 작업 경로를 나타낸 것이고, Fig. 2-38(c)은 subfield 1에서 145도 방향, subfield 2에서 120도 방향으로 생성된 작업 경로를 나타낸 것이고, Fig. 2-38(d)은 subfield 1과 subfield 2에서 모두 90도 방향으로 생성된 작업 경로를 나타낸 것이다.

Table 2-6과 Fig. 2-39는 시뮬레이션을 실시하여 포장 작업시간, 유효 작업

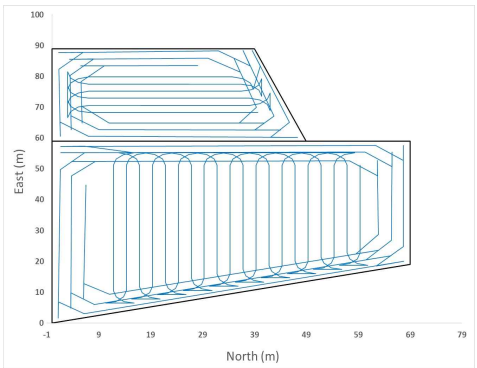
시간, 포장 작업거리, 포장 작업효율 등을 주행 방향별로 나타낸 결과이다.

주행 방향이 Case 3일 경우 트랙터의 포장 작업거리, 포장 작업시간과 포장 작업효율은 각각 2847 m, 3792 초, 77.5%로 예측 되었으며, 주행방향이 Case 4일 경우는 각각 2474 m, 3131 초, 84.5%로 주행 방향이 Case 3일 경우보다 포장 작업거리는 372 m 감소하였고, 포장 작업시간은 660 초 감소하였으며 포장 작업효율은 7% 증가하였다. 주행 방향이 Case 2에서 Case 4로 변화하면 왕복작업 수가 모두 26회에서 22회로 감소하였지만 총 주행거리는 14.3 m로 증가했다. 그러나 선회 비 작업 소모시간은 94 초가 감소하였기 때문에 포장 작업효율도 2.7% 증가하는 것으로 나타났다. 따라서, 즉 본 연구에서 사용한 오목 형태의 포장에서는 Case 4 방향의 조합으로 주행 경로를 생성 시 가장 적합한 결과로 나타났다.

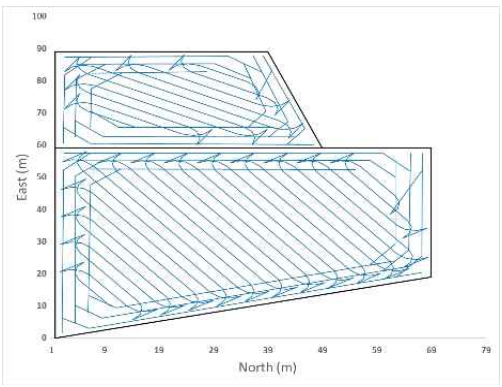
Fig. 2-37 Concave shaped polygonal field 2 for coverage path generation



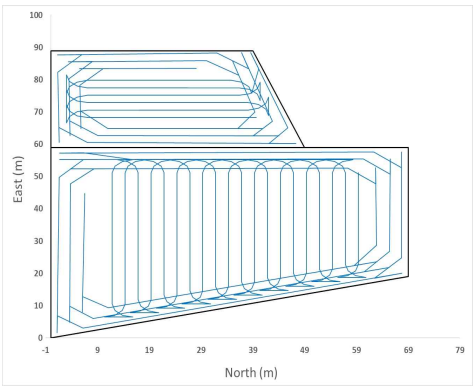
(a) Case 1



(b) Case 2



(c) Case 3



(d) Case 4

Fig. 2-38 Result of the path generation by changing the direction in concave shaped field 2

Table 2-6 Results of the simulation analysis by changing the direction in concave shaped polygonal field 2

Items	Values			
	Case 1	Case 2	Case 3	Case 4
Driving angle (deg)	90&0	0&90	145&120	90&90
Travelled distance (m)	2576.8	2469.0	2847.1	2474.5
Effective field time (sec)	2645.1	2664.9	2952.7	2625.0
Field time loss in turning (sec)	640.7	600.7	839.7	506.4
Field time (sec)	3285.7	3265.7	3792.4	3131.4
Field efficiency (%)	80.5	81.8	77.5	84.5
Round numbers	28	26	36	22

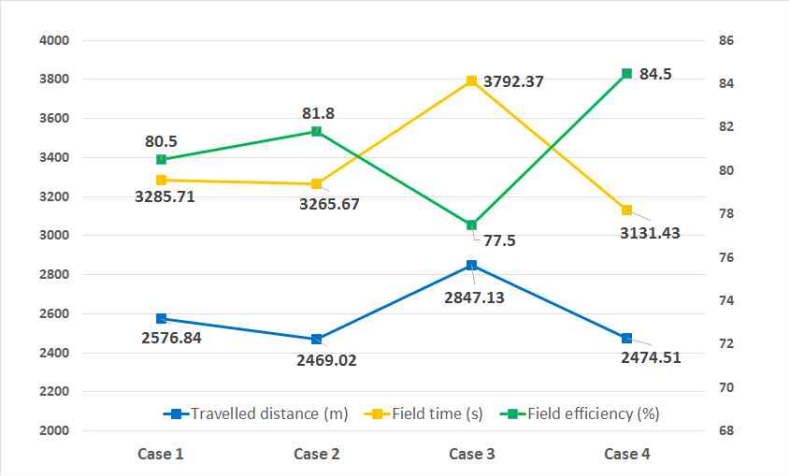


Fig. 2-39 Simulation analysis by changing the direction in concave shaped polygonal field 2

2.3. 요약 및 결론

농지형태에 따른 주행 경로생성 알고리즘을 적용하여 자율주행 트랙터 농작업을 수행하기 위한 경로생성 프로그램을 개발하였다. 개발한 경로생성 프로그램에서 포장의 좌표정보와 트랙터 및 작업기의 주요수치를 입력하는 방식이며 생성된 작업경로 데이터를 이용하여 사용자가 경로를 사전에 확인이 가능하게 하고 관련 주요 성능인 예상 포장 작업시간, 포장 작업거리, 유효 작업시간, 유효 작업거리 및 포장 작업효율 등을 함께 출력하여 경로생성의 정량적 정보를 확인할 수 있도록 하였다.

내부왕복작업부와 외부작업부로를 구분하기 위해 포장 외부 작업 경계선과 내부 작업 경계선을 생성하였다. 변수 설정부분으로부터 트랙터 작업기의 넓이, 작업 겹침 량과 주행 방향 등을 수동 입력이 가능하도록 하였으며, 주행할 회경 횟수를 입력 받게 되면 포장 내부 경계 영역을 생성하였다.

다변형 포장 내에서 미경지 작업 영역의 발생 방지를 위하여, MBR 원리 기반으로 왕복 직진 작업 경로를 생성하였으며 사용자가 설정된 주행 방향을 이용 MBR 범위 내에서 평행선을 생성한 후 각 평행선과 포장 내부 경계선이 교차된 두 점을 트랙터 직진 주행의 시작점과 종료점으로 도출하였다.

포장 내부 경계선을 생성하기 위해 포장 외부 작업부의 회경 횟수는 3 회로 가정하였으며 첫 번째 회경 경로와 포장 외부 경계선간의 거리는 유효 작업 폭 절반 (w/2)로 유지하여 생성하고 나머지 회경 경로는 유효 작업 폭(w) 만큼의 거리를 유지하여 순차적으로 생성하는 방식을 사용하였다.

볼록 형태의 포장에서는 구획작업이 필요 없으며 오목형태의 포장에서는 셀 분할 작업을 진행하였다. 본 연구에서도 주어진 포장이 오목형태이면 경로생성 전에 영역을 분할하였다. 포장 분할법 후 각각 서브 필드에서 다변형 경로생성 알고리즘을 적용하여 최종 통합된 경로 점을 생성하게 된다.

본 연구에서 개발한 경로생성 프로그램을 이용하여 다양한 포장형태별 생성

경로에 따른 예상 작업성능을 비교분석하여 활용성을 구명하였다.

볼록 형태의 다변형 포장에서 주행 방향을 변경하여 생성된 작업경로 특성을 시뮬레이션 통해 분석하였다. 주행 방향을 0도에서 65도로 변경 시 선회 소모시간이 380.74 초에서 269.96 초로 감소되어 유효 작업시간이 증가로 나타났다. 주행 방향이 65도에서 120도 및 170도로 증가하면 포장 작업효율이 각각 83.9%, 80.1%, 78.2%로 낮아진 것으로 나타났다. 본 연구에서 선정한 볼록 형태의 포장에서 65도 방향(포장의 장변 따라)으로 주행 경로를 생성 시 가장 적합한 결과로 판단되었다.

두 가지 오목형태의 다변형 포장에서 주행 방향을 변경하면서 생성된 작업경로 특성을 시뮬레이션 통해 분석하였다. 경로생성 전 우선 포장 분할법을 통해 오목형태의 포장을 셀 분할 작업을 수행 하였다.

첫 번째 오목형태 포장: 주행 방향이 Case 1일 경우는 트랙터의 포장 작업효율은 84.4%로 조사되었으며, 주행방향이 Case 2일 경우는 80.2%로 조사되어 주행 방향이 Case 1일 경우보다 포장 작업거리는 52.5 m 증가하였고, 총 주행시간은 108.1 초 감소하였다. 주행 방향이 Case 2에서 Case 3로 변화하면 포장 작업거리는 크게 차이가 없지만 포장 작업효율은 3.8% 증가하는 것으로 나타났다. 본 연구에서 사용한 오목형태의 포장에서 Case1 방향(포장의 장변 따라)의 조합으로 주행 경로를 생성 시 가장 적합한 결과로 판단된다.

두 번째 오목형태 포장: 주행 방향이 Case 3일 경우는 트랙터의 포장 작업효율은 77.5%로 조사되었으며, 주행방향이 Case 4일 경우는 84.5%로 조사되어 주행 방향이 Case 3일 경우보다 포장 작업시간은 660.94 초 감소하였다. 주행 방향이 Case 2에서 Case 4로 변화하면 포장 작업거리는 14.3 m로 증가하며 포장 작업효율도 2.7% 증가하는 것으로 나타났다. 본 연구에서 사용한 오목 형태의 포장에서 Case 4 방향(포장의 장변 따라)의 조합으로 주행 경로를 생성 시 가장 적합한 결과로 판단된다.

3. 시뮬레이션을 이용한 동적 경로추종 기술 개발

3.1. 강인 제어 기반의 동적 경로추종 기술

3.1.1. 재료 및 방법

3.1.1.1. 개요

Fig. 3-1은 경로생성과 경로추종의 관계를 나타낸 체계도이다. 첫 번째로 개발한 경로생성 프로그램을 이용하여 트랙터가 주행할 경로 정보인 RDDF가 생성된다. RDDF와 함께 트랙터의 실시간 위치 정보를 비교하여 적절한 경로추종 명령을 내리게 된다. 또한, 트랙터의 주행 속도, 조향 각, 및 작업기 조작에 필요한 작동 제어 신호도 함께 연동하여 주어진다. 본 연구에서는 트랙터의 경사지 주행성을 확보하기 위하여 경사지면에 따른 GPS 신호의 변화를 보상하는 알고리즘과 GPS 수신위치와 작업기 사이에 위치 차이에 따른 작업기 작동 지연시간을 고려한 작업기 제어 기술도 함께 개발하고자 하였다.

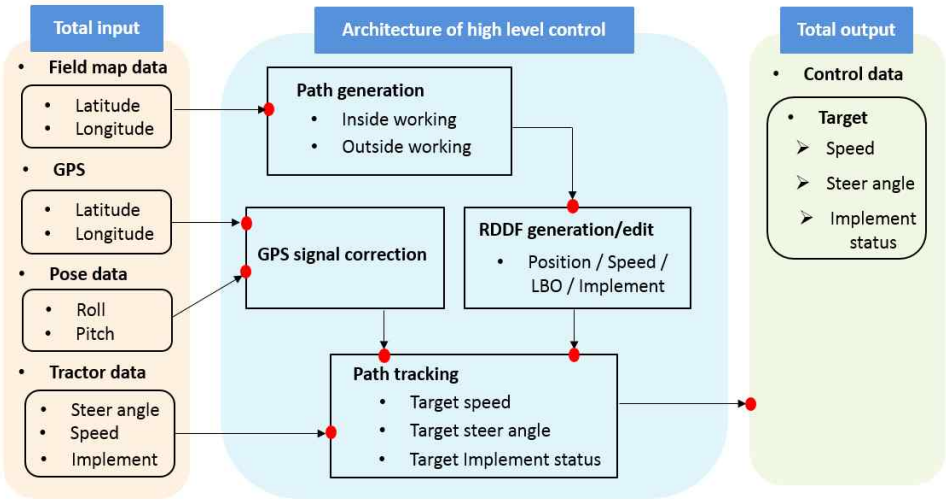


Fig. 3-1 Flowchart of high level control for path tracking

트랙터를 이용 포장에서 농작업 시 타이어의 측면강성계수(Cornering

Stiffness)는 트랙터의 중량, 관성 모멘트, 주행속도에 큰 영향을 받는다. 노면의 상태에 따라 생성된 슬립 각이 자율주행 트랙터의 정밀 추종에 영향을 주기 때문에 본 연구에서는 슬립각의 변화를 감지하여 경로추종을 연동하는 방식을 적용하였다. 슬립각의 측정과 관련한 최근 연구에서는 광, 음향 센서 등 이용 노면의 특성을 파악하는 방법()을 적용하였으나 기술적으로도 쉽지 않으며 슬립 각 센서도 고가인 단점이 있었기 때문에, 본 연구에서는 트랙터가 포장에서 주행 중 GPS/INS 기반으로 얻은 횡 편차 값과 방향각 오차정보를 기반으로 관측기를 통해 슬립 각을 예측하고 이를 이용하는 강인 제어 로직을 적용하였다.

하지만, 슬립의 정도는 선행 연구결과(Han et al., 2015)에 따르면 토양 함수율이 30% 이상의 포장에서는 선회시 트랙터가 외부로 밀리는 현상이 발생하여 주행하고자 하는 경로를 따라가지 못 하는 문제가 있는 것으로 나타났다. 따라서, 본 연구에서 적용한 강인 제어 로직은 트랙터가 포장에서 경운과 쟁기작업 시 토양의 상태를 고려한 조향제어 이지만 토양 함수율이 30%이하의 환경에서 적용이 되는 경로추종 알고리즘 개발에 집중을 하였다.

3.1.1.2. 자율주행을 위한 위치인지 시스템

Single GPS는 외부 환경에 따른 오차가 심하며 미터 이상의 오차를 있어 현재 위치의 정확하고 정밀한 정보가 요구되는 무인 트랙터에 사용이 부적합하다. 정확한 현재 위치의 인지를 위하여 RTK-GPS를 사용하였다. Fig. 3-2와 같이 RTK(Real Time Kinematic)는 수신기 1대만을 사용하였을 경우 발생하는 큰 수평오차를 줄이기 위하여 2대의 수신기를 동시에 사용하여 약측에서 동일하게 발생되는 오차를 상대적으로 소거하는 반송파신호를 사용한 상대측위 방식이다.

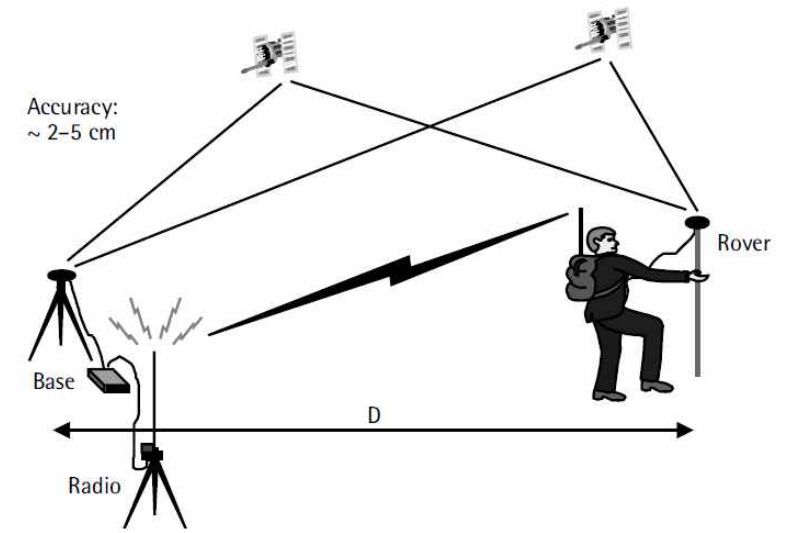


Fig. 3-2 Configuration of the RTK GPS (El-Rabbany, 2002)

또한 자율주행을 위한 위치인지 시스템은 Fig. 3-3과 같이 VRS(Virtual Reference Station) 구성방식으로 위치 정보를 취득하여 경로추종에 사용한다. 가상기준국은 실제로 존재하지 않지만 기존 기준국과 동일하게 트랙터에게 GPS 보정 정보를 제공이 가능하다. GPS 상시 관측소는 GPS수신 데이터를 인터넷을 통해 VRS 서버로 지속으로 전송하며 이동국 수신기는 자신의 현재 위치(NMEA 메시지)를 핸드폰을 통해 VRS서버로 전송한다. VRS 서버는 이동국 수신기 주변에 가상기준점을 생성하고 RTK측량에 필요한 RTCM 보정데이터를 생성 후 실시간으로 정밀한 위치정보를 자율주행 트랙터에 사용하게 된다.

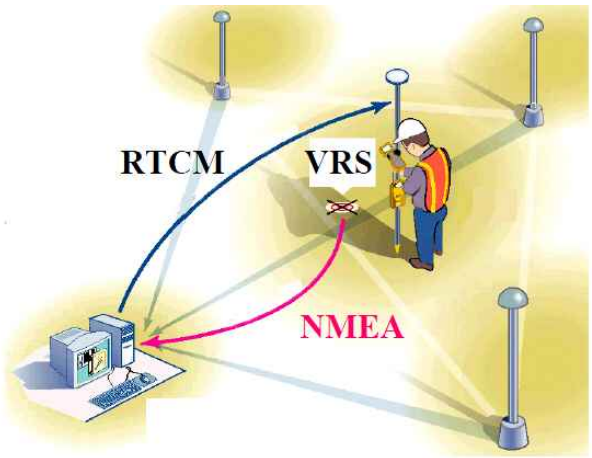


Fig. 3-3 Configuration of the VRS-RTK GPS (Landau, 2009)

3.1.1.3. 동적 경로추종 제어기 설계

(1) 체인 시스템 특성

본 연구에서는 체인 시스템 방법 기반으로 선형 시스템 이론을 이용 비선형 시스템 제어기를 설계하였다.(Samson, 1995). 일반적의 두 개의 제어입력을 가지고 있는 3차원의 비선형 시스템은 Eq. (3-1)과 같다.

$$\begin{cases} \dot{a_1} = m_1 \\ \dot{a_2} = a_3 m_1 \\ \dot{a_3} = m_2 \end{cases} \tag{3-1}$$

여기서, $[a_1, a_2, a_3]$= 상태 변수

$[m_1, m_2]$= 제어 입력

단일 입력을 가진 선형 시스템으로 변환하기 위해 Eq. (3-1)은 상태 변수 a_1에 대해서 미분하고 입력 m_1, m_2를 대신에 가상 제어 입력 m_3로 Eq. (3-2)와 같이 표현할 수 있다.

$$\begin{cases} a_1' = 1 \\ a_2' = a_3 \\ a_3' = m_3 \end{cases} \tag{3-2}$$

일반적으로 가상 제어 입력 m_3을 이용 Eq. (3-3)과 같이 PD제어기를 설계하고 Eq. (3-2)에 대입하면 Eq. (3-4)와 같은 2차 미분 방정식을 얻게 된다. 즉 P 와 D 이득 값을 조절하여 Fig. 3-4와 같이 상태 변수 a_2를 0으로 수렴하게 할 수 있다.

$$m_3 = -a_3 K_d - a_2 K_p \tag{3-3}$$

$$a_2'' + a_2' K_d + a_2 K_p = 0 \tag{3-4}$$

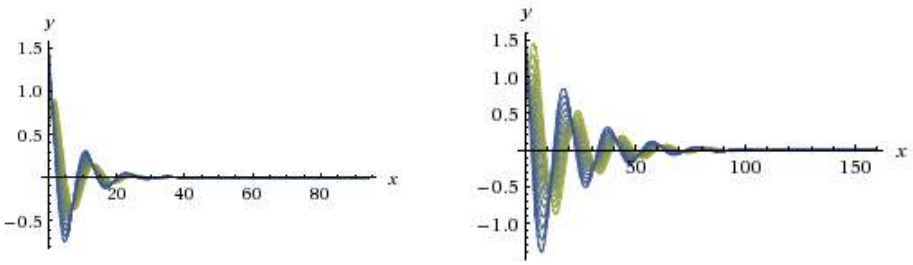


Fig. 3-4 State a_2 can converge to zero by choosing gain
(left: K_p=0.3, K_d=0.3 right: K_p=0.1, K_d=0.1)

(2) 강인제어 로직 설계

본 연구에서는 강인제어기를 설계하기 위해 Fig. 3-5와 같이 Extended Ackerman Kinematic Model (EAKM)를 사용하여 제어 로직을 구성하였다. 즉, EAKM은 횡 편차(Lateral Deviation) 값과 방향각 오차(Heading Error)정보만 고려하는 것이 아닌 트랙터 포장에서 주행 시 측면 외란(Lateral Disturbances)도 고려하여 Eq. (3-5)와 같이 슬립 각 β_f와 β_r을 극복이 가능한

주행 모델을 사용하였다(Lenain et al., 2007; Fang et al., 2011).

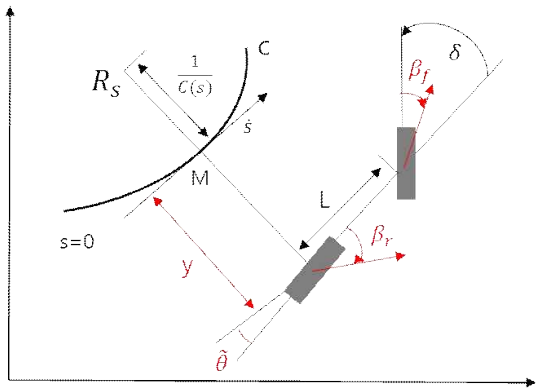


Fig. 3-5 Extended Ackerman Kinematic Model

$$\begin{cases} \dot{s} = \dfrac{v\cos(\tilde{\theta}+\beta_r)}{1-c(s)y} \\ \dot{y} = v\sin(\tilde{\theta}+\beta_r)+\varepsilon_1 \\ \dot{\tilde{\theta}} = v[\cos\beta_r \dfrac{\tan(\delta+\beta_f)-\tan\beta_r}{L} - \dfrac{c(s)\cos(\tilde{\theta}+\beta_r)}{1-c(s)y}]+\varepsilon_2 \end{cases} \quad (3\text{-}5)$$

여기서 s = 곡선 좌표 (m)

y = 횡 편차 (m)

θ = 방향 오차 (°)

$c(s)$ = 곡률 (m^{-1})

v = 주행 속도 (m/s)

δ = 조향 각도 (°)

β_f = 전륜 슬립 각도 (°)

β_r = 후륜 슬립 각도 (°)

$\varepsilon_i =$ 시스템 외란 (decimal)

상태 변환을 통해 Eq. (3-5)의 EAKM를 Eq. (3-6)과 같이 변환이 가능하며 또한 Eq. (3-7)과 같이 Perturbed 체인 시스템을 얻을 수 있다.

$$(a_1, a_2, a_3) = (s, y, (1-c(s)y)\tan(\tilde{\theta}+\beta_r)) \tag{3-6}$$

$$\begin{cases} \dot{a}_1 = \dfrac{v\cos(\tilde{\theta}+\beta_r)}{1-c(s)y} = m_1 \\ \dot{a}_2 = v\sin(\tilde{\theta}+\beta_r) + \varepsilon_1 = a_3 m_1 + \varepsilon_1 \\ \dot{a}_3 = \dfrac{d(1-yc(s))\tan(\tilde{\theta}+\beta_r)}{dt} = m_2 + \eta \end{cases} \tag{3-7}$$

여기서, $m_2 = -vc(s)\sin(\tilde{\theta}+\beta_r)\tan(\tilde{\theta}+\beta_r) - v\dfrac{dc(s)}{ds}\dfrac{\cos(\tilde{\theta}+\beta_r)}{1-yc(s)}tan(\tilde{\theta}+\beta_r)y$

$$+ v\frac{1-yc(s)}{\cos^2(\tilde{\theta}+\beta_r)}(\cos\beta_r\frac{\tan(\delta+\beta_f)-\tan\beta_r}{L} - c(s)\frac{\cos(\tilde{\theta}+\beta_r)}{1-yc(s)})$$

$$+ \frac{1-yc(s)}{\cos^2(\tilde{\theta}+\beta_r)}\frac{d\beta_r}{dt}$$

$$\eta = \frac{(1-yc(s))\varepsilon_2}{\cos^2(\tilde{\theta}+\beta_r)} - c(s)\varepsilon_1\tan(\tilde{\theta}+\beta_r)$$

여기서 ε_1과 η는 이상 체인 시스템에 외란 역할을 추가한 것이며 Eq. (3-7) 이용 상태 변수 a_1에 대해서 미분하여 Eq. (3-8)과 같이 단일 입력을 가진 선형 시스템으로 전환한다.

$$\begin{cases} a_1^{'} = 1 \\ a_2^{'} = a_3 + \dfrac{\varepsilon_1}{m_1} \\ a_3^{'} = \dfrac{m_2}{m_1} + \dfrac{\eta}{m_1} = u + \dfrac{\eta}{m_1} \end{cases} \tag{3-8}$$

가상 제어 입력 u를 이용 Eq. (3-9)와 같이 PD 제어기를 설계하고 Eq. (3-8)에 대입하면 Eq. (3-10)과 같은 2차 미분 방정식을 얻게 된다. 즉 P gain과 D gain을 조절하여 상태 변수 a_2를 0으로 수렴하는 것이 목적이다.

$$u = -zK_D - a_3K_P - \rho \tanh\left(\frac{0.2785\rho z}{\sigma}\right) \quad \sigma > 0 \tag{3-9}$$

$$a_2^{''} + a_2^{'}K_D + a_2K_P = 0 \tag{3-10}$$

여기서, $z = a_2K_P + a_3 = yK_P + (1 - yc(s))\tan(\tilde{\theta} + \beta_r)$

$\sigma =$ 양수

$K_P, K_D =$ 제어 매개 변수

Eq. (3-8)부터 Eq. (3-10)까지 수식을 이용 트랙터가 슬립을 고려한 실제 조향 각은 가상 강인 제어 입력 u를 역 변환에 의해 Eq. (3-11)와 같이 얻어진다.

$$\delta(y,\tilde{\theta}) = \arctan\left[L\left(\frac{\cos^3(\tilde{\theta}+\beta_r)}{(1-yc(s))^2\cos\beta_r}\left(\frac{dc(s)}{ds}ytan(\tilde{\theta}+\beta_r)\right.\right.\right. \\ \left. + u + c(s)(1-yc(s))\tan^2(\tilde{\theta}+\beta_r) \right. \\ \left.\left.\left. + \frac{c(s)\cos(\tilde{\theta}+\beta_r)}{(1-yc(s))^2\cos\beta_r}\right) - \frac{L}{vcos\beta_r}\frac{d\beta_r}{dt} + \tan\beta_r \right] - \beta_f \tag{3-11}$$

3.1.1.4. 관측기를 이용한 슬라이딩 예측 알고리즘

트랙터가 포장에서 주행 시 발생한 전 후륜 슬립 각이 알고 있는 경우 EAKM을 이용 트랙터의 슬라이딩 움직임 표현이 가능하기 때문에 이 두변수의 추정이 매우 중요한 부분이다.

Fig. 3-6은 강인제어기와 슬립관측 기반의 자율주행 시스템 개요도이다. 원리적으로 제어 루프가 앞서 경로생성 프로그램에서 생성된 RDDF를 입력받아 위치 정보를 이용 기준 경로로 사용한다. 관측 루프에서 생성된 횡 편차와 방향각 오차의 예측 값과 제어 루프에서 생성된 횡 편차와 방향각 오차의 측정값을 모두 관측 제어기(Observer Control)에 전송하여 전륜과 후륜의 슬립 각을 각각 예측한다. 예측된 슬립 각을 제어 루프에 강인제어로직 기반으로 개발된 제어기에 적용하여 조향 각을 계산 후 트랙터 움직임과 관측기의 입력값으로 사용하는 개념이다.

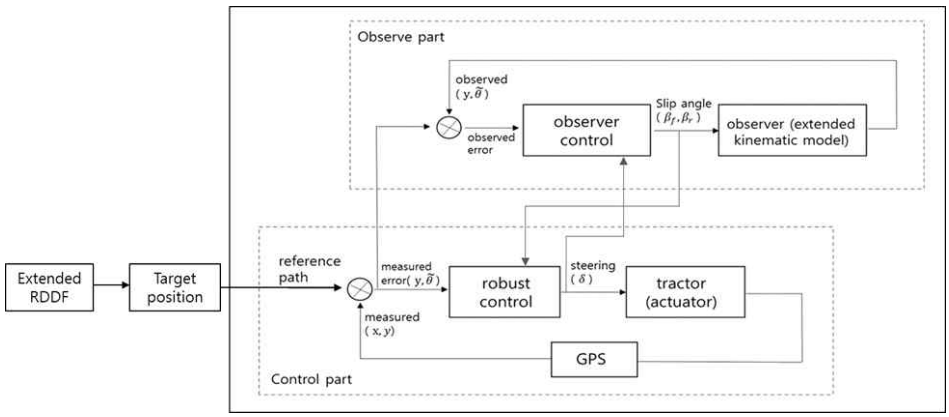


Fig. 3-6 Flowchart of robust control and slip observer loops

본 연구에서는 GPS/INS 기반으로 측정된 횡 편차 값과 방향각 오차정보를 기반으로 슬립관측 알고리즘을 통해 슬립 각을 간접적으로 추정한다(Stephant et al., 2004 and Lenain et al., 2006). 개발한 관측기는 제어 측면에서 문제를 해결 위해 전 후륜 슬립 각은 각각 제어 변수로 설계하여 EAKM 수식에서

예측된 횡 편차 값과 방향각 오차가 GPS/INS 기반으로 측정된 횡 편차 값과 방향각 오차로 수렴하게 한다. 슬립관측 알고리즘은 Eq. (3-12)과 같이 EAKM 비선형 시스템을 사용한다.

$$\dot{\hat{X}} = f\left[\hat{X}, \begin{pmatrix}\beta_f \\ \beta_r\end{pmatrix}\right] = \begin{cases} f_1 = v sin(\hat{\hat{\theta}} + \beta_r) \\ f_2 = v\left[\cos\beta_r \dfrac{\tan(\delta+\beta_f) - \tan\beta_r}{L} - c(s)\dfrac{\cos(\hat{\hat{\theta}} + \beta_r)}{1 - c(s)\hat{y}}\right] \end{cases} \quad (3\text{-}12)$$

비선형 시스템 Eq. (3-12)를 Eq. (3-13)과 같이 선형화 한다. 여기서 행렬 B는 역변환 가능한 행렬이며 이 행렬은 f를 슬립 각 행렬 $u = \begin{bmatrix}\beta_f \\ \beta_r\end{bmatrix}$에 대해 편미분 한 것이다.

$$\dot{\hat{X}} = f(\hat{X}, 0) + B(\hat{X}) \bullet \begin{pmatrix}\beta_f \\ \beta_r\end{pmatrix} \quad (3\text{-}13)$$

여기서, $B(\hat{X}) = \dfrac{\partial f}{\partial u}(\hat{X}, 0) = \begin{bmatrix} v cos\hat{\hat{\theta}} & 0 \\ v\dfrac{c(s)\sin\hat{\hat{\theta}}}{1 - c(s)\hat{y}} - \dfrac{v}{L} & \dfrac{v}{L}(1 + \tan^2\delta) \end{bmatrix}$

관측 오차 e는 X(관측) - X(측정)의 차이 값으로 정의가 되고 그의 목적은 X(관측)값이 X(측정)값으로 근접하게 생성하는 것이며 즉 관측 오차 e를 0으로 수렴하게 만드는 것이다. Error dynamics는 Eq. (3-14)와 같이 정의가 되고 여기서 관측기 이득(gain)인 행렬 G는 Hurwitz 행렬이다.

$$\dot{e} = G \bullet e = f(\hat{X}, 0) - \dot{\overline{X}} + B\begin{pmatrix}\beta_f \\ \beta_r\end{pmatrix} \quad (3\text{-}14)$$

여기서, $\dot{\overline{X}} = \dfrac{\overline{X}_t - \overline{X}_{t-1}}{T}$

행렬 B는 역변환 가능한 행렬이 때문에 슬립각의 예측 값은 Eq. (3-15)와 같이 유도를 가능하며 예측 값을 강인제어 로직에 반영하도록 하였다.

$$\begin{bmatrix}\beta_f\\ \beta_r\end{bmatrix}=\begin{bmatrix} v\cos\hat{\hat{\theta}} & 0\\ v\dfrac{c(s)\sin\hat{\hat{\theta}}}{1-c(s)\hat{y}}-\dfrac{v}{L} & \dfrac{v}{L}(1+\tan^2\delta)\end{bmatrix}^{-1}\left\{G\bullet e-f(\hat{X},0)+\dot{\bar{X}}\right\} \qquad (3\text{-}15)$$

3.1.1.5. 경사지면에서 GPS 신호 보정

트랙터에 장착되는 GPS (설치 높이 약 2 m)는 Fig. 3-7과 같이 포장의 경사지 혹은 불규칙한 노면 상태에 의해 롤, 피치의 영향을 크게 받게 되며 트랙터의 실제 위치를 제공하기 어렵다. 정확한 경로추종을 진행하기 위해 트랙터에 장착된 IMU에서 출력한 롤 피치 값을 이용 위치 오차를 보상하고자 하였다.

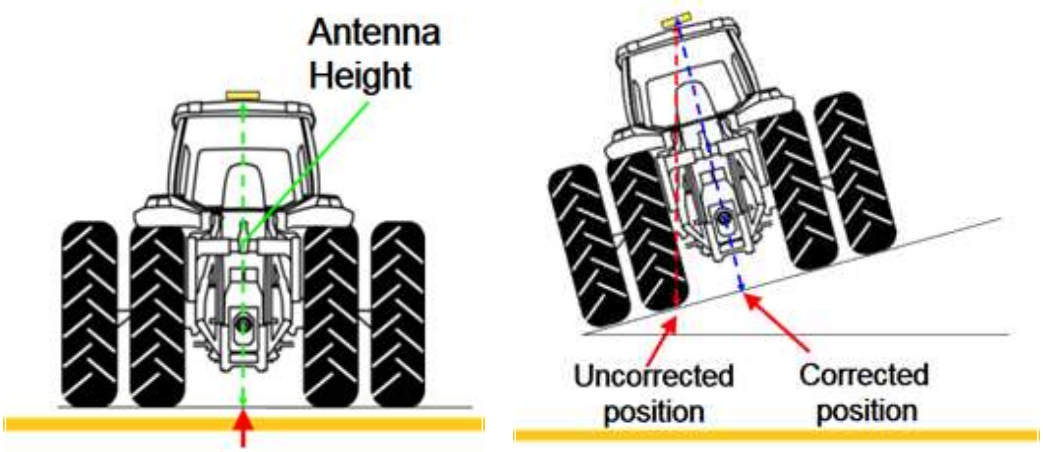


Fig. 3-7 Position corrections of vehicle inclinations

GPS 위치 오차 보상 방법은 Eq. (3-16)과 같으며 여기서 행렬 E는 트랙터 상태좌표를 절대좌표로 변환하기 위한 전환 행렬 이다.

$$\begin{pmatrix}X_c\\ Y_c\\ Z_c\end{pmatrix}=\begin{pmatrix}X_r\\ Y_r\\ Z_r\end{pmatrix}-\begin{pmatrix}a\\ b\\ c\end{pmatrix}\bullet E^{-1}(\psi,\theta,\alpha) \qquad (3\text{-}16)$$

$$E^{-1}(\psi,\theta,\alpha) = E(-\psi,-\theta,-\alpha)$$

$$= \begin{pmatrix} \sin\alpha\sin\theta\sin\psi + \cos\alpha\cos\psi & \cos\theta\sin\psi & \sin\alpha\cos\psi - \cos\alpha\sin\theta\sin\psi \\ \sin\alpha\sin\theta\cos\psi - \cos\alpha\sin\psi & \cos\theta\cos\psi & -\sin\alpha\sin\psi - \cos\alpha\sin\theta\cos\psi \\ -\sin\alpha\cos\theta & \sin\theta & \cos\alpha\cos\theta \end{pmatrix}$$

여기서, X_c, Y_c, Z_c = GPS 위치 보상 후 (m)

X_r, Y_r, Z_r = GPS 위치 보상 전 (m)

a, b, c = 트랙터 COG과 GPS간의 걸이 (m)

ψ = IMU의 요잉 각도 (°)

θ = IMU의 피치 각도 (°)

α = IMU의 롤 각도 (°)

3.1.1.6. 미경지 영역 방지를 위한 작업기 상승/하강 제어

트랙터가 생성된 주행 경로를 기반으로 농작업 진행 시 미 작업 영역 부분을 감소 및 포장 작업효율 향상을 위해 트랙터가 새머리 영역으로 진입 및 새머리 영역에서 탈출 시 작업기 작동 요구된 특정 경로점에서 정확히 작동하는 것이 필요하다. Fig. 3-8은 RDDF에서 작업기 제어명령이 구성된 하나의 예이며 4번 경로점에서 트랙터가 작업기 상승하고 16번째 경로점에서 작업기 하강 요구를 하고 있다.

1	9.5600000	-2.100000	0.5	1	0	4	1	
2	9.5600000	-17.10000	0.5	1	0	4	1	
3	9.5600000	-32.10000	0.5	1	0	4	1	
4	9.5600013	-35.3662105	0.5	1	0	2	0	Point of implement up
5	9.8884132	-36.9566735	0.5	1	0	2	0	
6	10.8220466	-38.285491	0.5	1	0	2	0	
7	12.2189914	-39.1379338	0.5	1	0	2	0	
8	13.8166324	-40.2058721	0.5	1	0	2	0	
9	14.7318551	-41.8956403	0.5	1	0	2	0	
10	14.7534175	-43.817224	0.5	1	0	2	0	
11	13.876343	-45.5271039	0.5	1	0	2	0	
12	12.3030684	-46.6306238	0.5	1	0	2	0	
13	10.3967202	-46.8730809	0.5	1	0	2	0	
14	8.5973012	-46.1985139	0.5	1	0	2	0	
15	7.3201343	-44.7626193	0.5	1	0	2	0	
16	6.8600013	-42.8968147	0.5	1	0	2	1	Point of implement down
17	6.8600013	-27.8968147	0.5	1	0	4	1	
18	6.8600013	-12.8968147	0.5	1	0	4	1	
19	6.8600013	2.1031853	0.5	1	0	4	1	
20	6.8600013	17.1031853	0.5	1	0	4	1	

Fig. 3-8 Commands of implement control on the RDDF

Fig. 3-9는 작업기의 효율적 작동을 위한 신호 제어 흐름도이다. 원리적으로 트랙터의 위치가 작업기 상승 요구된 위치에 도달하면 작업기 명령이 “1” 에서 “0”로 변환되어 작업기 상승을 위한 제어 위치를 계산한다. 제어 위치가 작업기 상승 요구된 경로점의 LBO 범위에 들어오면 작업기 상승하게 되며 아닌 경우 작업기 계속 하강상태로 주행한다. 트랙터의 위치가 작업기 하강 요구된 위치에 도달하면 작업기 명령이 “0” 에서 “1”로 변환되어 작업기 하강을 위한 제어 위치를 계산한다. 제어 위치가 작업기 하강 요구된 경로점의 LBO 범위에 들어오면 작업기 하강하게 되며 아닌 경우 작업기 계속 상승상태로 주행하는 개념이다.

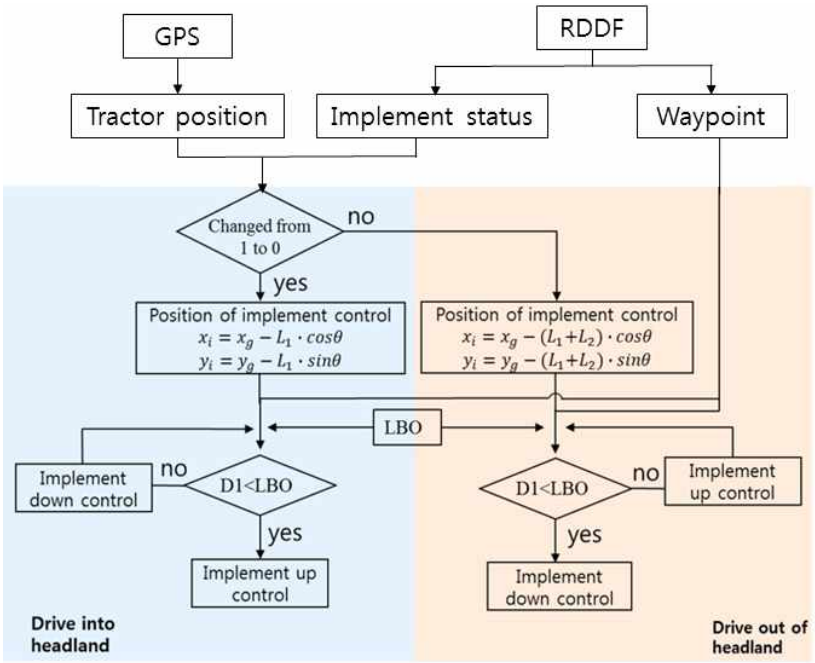


Fig. 3-9 Flowchart of the implement control

작업기 상승/하강 제어 위치의 계산은 Fig. 3-10과 같이 트랙터의 기구학적 거동 기술 기반의 모델을 사용하였다. 이 모델은 전륜과 후륜 중심으로 구성

하며 필드좌표에서 위치와 주행방향을 정의할 수 있다. 본 모델에서는 오직 전륜만 좌우로 조향가능하며 휠의 측면 슬립과 외력은 없는 것으로 가정하였다. Eq. (3-17)과 같이 트랙터의 거동을 X축과 Y축방향의 속도로 정의하고 트랙터의 방향각과 회전 각속도 및 조향 휠의 각속도를 정의하였다.

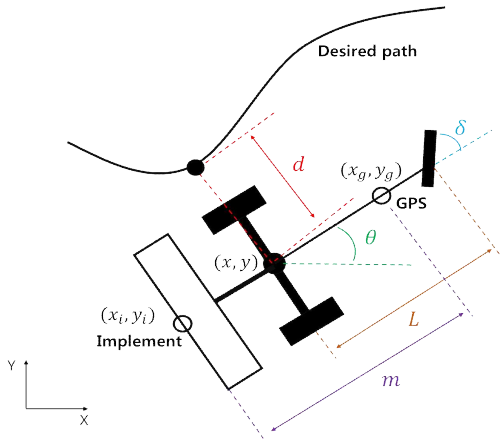


Fig. 3-10 Kinematic bicycle model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \dfrac{\tan(\theta)}{L} \\ 0 \end{bmatrix} \upsilon + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \delta \tag{3-17}$$

여기서, θ = 주행 방향 (°)

δ = 조향 각도 (°)

L = 축간 길이 (m)

υ = 주행 속도 (m/s)

GPS에서 수신된 위치 좌표(x_g, y_g)와 작업 위치 좌표(x_i, y_i)의 관계는 Eq. (3-18)과 같다. 여기서 m은 GPS 수신 위치와 작업기 위치간의 거리이다.

Fig. 3-11과 같이 트랙터가 새머리 영역으로 진입 시 GPS 설치 위치와 작업기 앞부분 간의 거리(L1)를 고려하여 사전에 만들어진 RDDF에 작업기 제어 명령을 인식하였다. 또한 트랙터가 새머리 영역에서 탈출 시 GPS 설치 위치와 작업기 앞부분 간의 거리 그리고 로터리 날의 직경을 포함된 거리(L1+L2)를 고려하여 사전에 만들어진 RDDF에 작업기 제어 명령을 인식하도록 하였다.

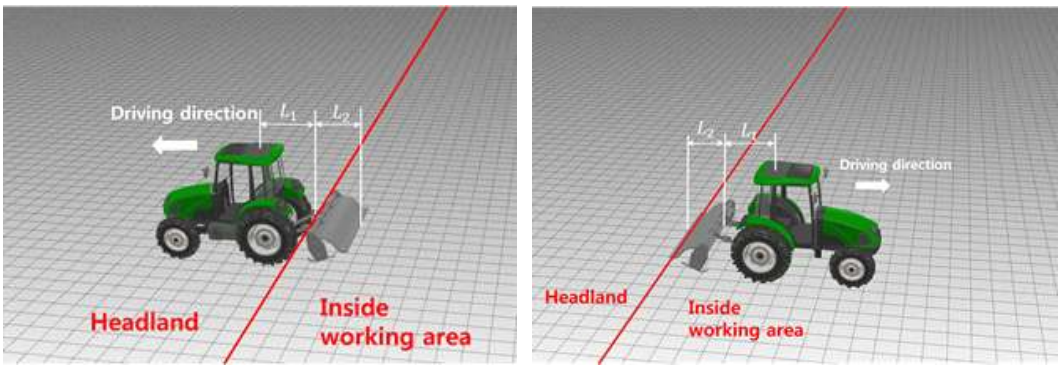


Fig. 3-11 Definition of parameters for implement control

$$\begin{cases} x_g = x_i + m\cos\theta \\ y_g = y_i + m\sin\theta \end{cases} \tag{3-18}$$

$$m = \begin{cases} L_1 & (drive\ into\ hedland) \\ L_1 + L_2 & (drive\ out\ of\ headland) \end{cases}$$

여기서, x_g, y_g = GPS 수신 위치 (m)

x_i, y_i = 작업기 위치 (m)

m = GPS와 작업기간의 길이 (m)

Eq. (3-18)를 Eq. (3-17)에 대입하면 아래 Eq. (3-19)와 같다

$$\begin{cases} \dot{x_g} = \dot{x_i} - msin\theta = vcos\theta - msin\theta \\ \dot{y_g} = \dot{y_i} + mcos\theta = vsin\theta + mcos\theta \end{cases} \tag{3-19}$$

Eq. (3-19)을 제곱하여 아래 Eq. (3-20)과 같이 정리한다.

$$\begin{aligned} \dot{x_g}^2 + \dot{y_g}^2 &= (vcos\theta - msin\theta)^2 + (vsin\theta + mcos\theta)^2 \\ &= v^2\cos^2\theta + m^2\dot{\theta}^2\sin^2\theta - 2va\dot{\theta}cos\theta\sin\theta + v^2\sin^2\theta \\ &\quad + m^2\dot{\theta}^2\cos^2\theta + 2va\dot{\theta}cos\theta\sin\theta \\ &= v^2 + m^2\dot{\theta}^2 \end{aligned} \tag{3-20}$$

Eq. (3-20)을 다시 정리하면 Eq. (3-21)와 같고 이를 이용 트랙터의 현재 방향을 Eq. (3-22)와 같이 얻게 된다.

$$\dot{\theta} = \pm \frac{\sqrt{\dot{x_g}^2 + \dot{y_g}^2 - v^2}}{m} \tag{3-21}$$

$$\theta_t = \theta_{t-1} \pm \frac{\triangle T}{m} \sqrt{\left(\frac{x_{g_t} - x_{g_{t-1}}}{\triangle T} \right)^2 + \left(\frac{y_{g_t} - y_{g_{t-1}}}{\triangle T} \right)^2 - v^2} \tag{3-22}$$

여기서, $\triangle T$= 샘플링 시간 (sec)

x_{g_t}, y_{g_t} = 시간 t에서의 GPS 위치 (m)

θ_t = 시간 t에서의 주행 방향 (°)

Eq. (3-22)에서 얻은 현재 방향각과 작업기 전 위치의 정보를 이용하여 Eq. (3-23)과 같이 작업기 현재 위치를 계산한다. 계산된 작업기 현재 위치를 이용 사전에 만들어진 RDDF에 작업기 제어 명령을 인지하여 작업기 제어를 한다.

$$\begin{cases} x_{i_t} = x_{i_{t-1}} + v\triangle Tcos\theta_t \\ y_{i_t} = y_{i_{t-1}} + v\triangle Tsin\theta_t \end{cases} \tag{3-23}$$

여기서, x_{i_t}, y_{i_t} = 시간 t에서의 작업기 위치 (m)

3.1.2. 자율주행 성능 평가 방법

3.1.2.1. 주행 위치 오차 분석

트랙터의 주행 오차는 Fig. 3-12와 같이 트랙터의 위치좌표로부터 설정된 주행 경로에 대한 수직거리를 계산하여 확인하였다. 오차는 직선주행 방향의 오른쪽에 있는 오차를 양수로, 왼쪽에 있는 경우를 음의 수로 정하였다. RMS 값은 설정된 경로와 실제 경로와의 차이를 이용하여 Eq. (3-24)에 의해 계산하였다.

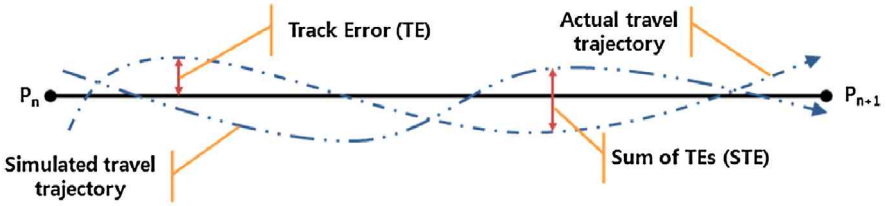


Fig. 3-12 Definition of sum of track errors

$$R.M.S.E = \sqrt{\frac{\sum (TEs)^2}{Nmber\,of\,data}} \tag{3-24}$$

3.1.2.2. 미 작업 영역 분석

트랙터가 실제 작업경로에 따라 작업 시 미경지 부분 또는 오버랩이 발생하게 된다. 본 연구에서 미경지 부분만 계산하였으며 오버랩부분은 계산하지 않았다. 미경지 영역을 도출하기 위해 ArcGIS (Ver. 10.1) 프로그램을 이용하여 Fig. 3-13과 같이 작업위치 정보를 이용 작업영역으로 표현하여 해당 작업면적을 계산하였다. 우선 작업기 위치 점을 라인으로 변환한 후 일정한 작업기 폭 크기를 적용하여 생성된 라인 기준으로 작업기가 경유한 작업영역을 생성한다. 생성 작업영역은 픽셀 면적으로 계산이 가능하며 이를 통하여 포장내의

유효 작업영역을 표현하고 유효 작업면적을 계산하였다.

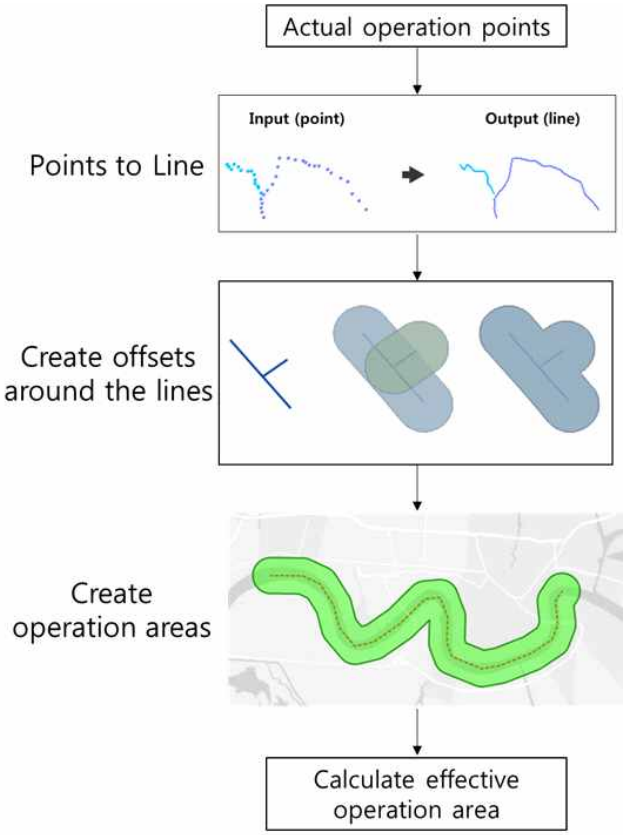


Fig. 3-13 Method of calculating effective operation area in ArcGIS

3.2. 무인 농작업 시뮬레이터 개발 및 활용 기술

3.2.1. 재료 및 방법

3.2.1.1. 개요

실제 포장에서 자율주행 경운 및 쟁기작업을 수행하기 위해 Fig. 3-14과 같은 환경구조로 가상 컴퓨터 시뮬레이션 프로그램을 개발하였다. 테스트 환경

은 크게 클라이언트 환경인 LabVIEW 프로그램에서 경로생성 및 경로추종 알고리즘을 구성하고 생성된 조향 각 및 속도정보는 Server 환경인 RoboticsLab 프로그램으로 전송하는 방식이다. 서버 환경에서 Extended Kinematic Bicycle Model (EKBM)을 이용 트랙터의 움직임을 표현하였다. 트랙터의 위치정보와 자세정보는 서버 환경에서 클라이언트 환경으로 피드백 전송하여 목표점과 비교 후 무인 주행에 필요한 조향 각과 속도정보를 다시 생성한다. 가상 컴퓨터 시뮬레이션 프로그램을 이용하여 개발된 경로생성 및 경로추종 알고리즘을 수정 및 검증을 위해 적합한 매개변수를 규명할 수 있도록 하였다.

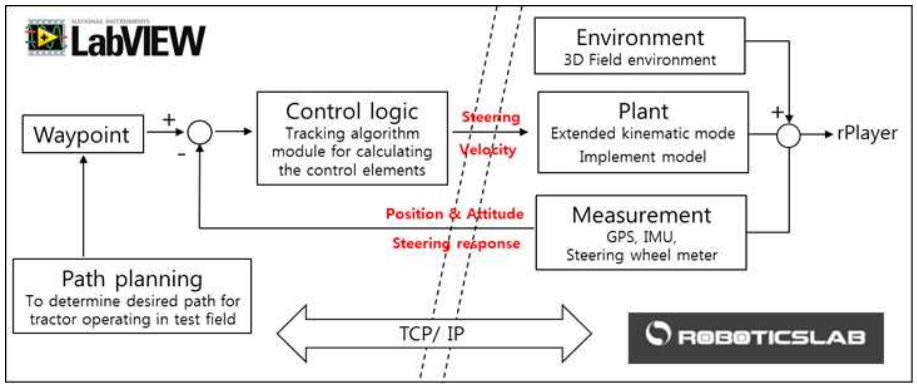


Fig. 3-14 Configuration of the developed virtual simulator system

3.2.1.2. 가상 트랙터 주행 모델

트랙터가 실제 포장에서 회전 주행 시 토양의 반력에 인해 슬립현상이 발생하게 되며 이러한 현상을 시뮬레이션에서 반영하기 위해 EKBM을 사용하였다(Zhang et al., 2009). Fig. 3-15와 같이 트랙터 회전 시 주행 속도(V)는 항상 진행 방향과 회전방향 사이의 벡터로 나타내며 이 때 진행방향과 속도 사이의 각도(β)는 회전 시 바퀴의 슬립 각에 의해 생성되고 트랙터 옆으로 미끄러지는 각도이다.

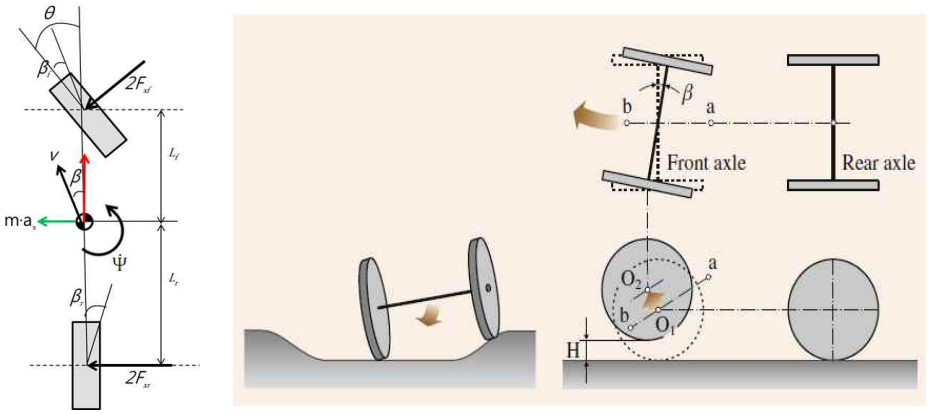


Fig. 3-15 Extended kinematics bicycle model (Zhang et al. 2009)

본 연구에서 트랙터의 주행속도는 정속으로(V_X =constant) 가정하였으며 속도와 슬립 각의 관계인 Eq. (3-25)을 미분하여 Eq. (3-26)으로 변환을 하였다.

$$v_y = v_x \times \beta \tag{3-25}$$

$$\dot{v_y} = v_x \times \dot{\beta} \tag{3-26}$$

여기서, v_x = 종방향 속도 (m/s)

v_y = 횡방향 속도 (m/s)

β = 트랙터 슬립 각도 (°)

뉴턴의 제 2법칙에 따르면 트랙터 역학은 다음의 Y-axis Motion Extended Kinematic방정식 Eq. (3-27)과 Yaw-axis Motion Dynamic 방정식 Eq. (3-28)으로 정리할 수 있다.

$$\sum F_y = ma_y = mv_x(\dot{\beta} + \dot{\psi}) = 2F_{yf} + 2F_{yr} \tag{3-27}$$

$$\sum M_z = I_z\ddot{\psi} = 2l_fF_{yf} - 2l_rF_{yr} \tag{3-28}$$

여기서, m= 트랙터 무개 (kg)

a_y = 트랙터 횡방향 가속도 (m/s^2)

I_z = 회전 관성 (kg-m^2)

$\dot{\psi}$= 트랙터 각속도 (°/s)

l_f, l_r = 트랙터 COG과 전/후륜의 휠 간의 길이 (m)

F_{yf}, F_{yr} = 전륜과 후륜의 횡방향 반력 (N)

트랙터 선회주행 시 전륜과 후륜의 측면 반력에 인해 생성된 슬립 각은 Fig. 3-16과 같이 정의 된다. 포장에서 트랙터 저속 주행 때문에 $\tan\delta = \delta$ 가정하여 전륜과 후륜의 슬립 각은 Eq. (3-29)와 (3-30)로 정리할 수 있다.

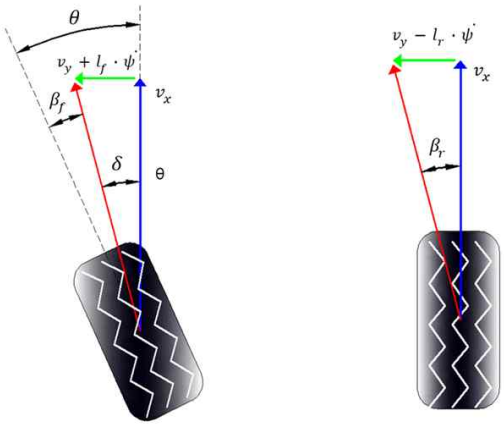


Fig. 3-16 Definition of the slip angle for front and rear tire

$$\beta_f = \theta - \delta = \theta - \tan^{-1}\frac{v_y + l_f\dot{\psi}}{v_x} = \theta - \frac{v_y + l_f\dot{\psi}}{v_x} \tag{3-29}$$

$$\beta_r = -\tan^{-1}\frac{v_y - l_r\dot{\psi}}{v_x} = -\frac{v_y - l_r\dot{\psi}}{v_x} \tag{3-30}$$

바퀴에 적용한 측면 반력은 Eq. (3-31), Eq. (3-32)에 정의된 것과 같이 슬립 각과 코너링 강성에 대한 함수이다.

$$F_{yf} = C_f\beta_f = C_f\left(\theta - \frac{v_y + l_f\dot{\psi}}{v_x}\right) \tag{3-31}$$

$$F_{yr} = C_r\beta_r = C_r\left(-\frac{v_y - l_r\dot{\psi}}{v_x}\right) \tag{3-32}$$

여기서, $C_{f,}C_r$ = 전륜과 후륜의 코너링 스티프니스 (N/rad)

$\beta_{f,}\beta_r$ = 전륜과 후륜의 슬립 각도 (°)

v_x, v_y = 종방향과 횡방향의 속도 (m/s)

θ = 조향 각도 (°)

Eqs. (3-25) - (3-32) 수식들을 이용하여 Eq. (3-33)와 같이 트랙터의 슬립 각(slip angle), 요잉 각(yaw angle), 요잉 율(yaw rate)을 결정 할 수 있다.

$$\begin{bmatrix}\dot{\beta}\\ \ddot{\psi}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} -\frac{2(C_f+C_r)}{mv_x} & -1-\frac{2(l_fC_f-l_rC_r)}{mv_x^2} & 0\\ -\frac{2(l_fC_f-l_rC_r)}{I_z} & -\frac{2(l_f^2C_f+l_r^2C_r)}{I_zv_x} & 0\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\beta\\ \dot{\psi}\\ \psi\end{bmatrix} + \begin{bmatrix}\frac{2C_f}{mv_x}\\ \frac{2l_fC_f}{I_z}\\ 0\end{bmatrix}\theta \tag{3-33}$$

Fig. 3-17과 같이 트랙터의 주행 위치는 슬립 각, 요잉 율 및 주행 속도에 의해 결정된다. 계산을 간소화하기 위해서 트랙터 역학의 모델은 필드 좌표의 기준점에서 출발한다는 가정으로 개발한다. 트랙터의 자율주행 추종 알고리즘에 적용하는 실시간 위치 데이터는 다음의 Eq. (3-34)으로 정의된 것과 같이

X 방향과 Y 방향에 대한 함수이다.

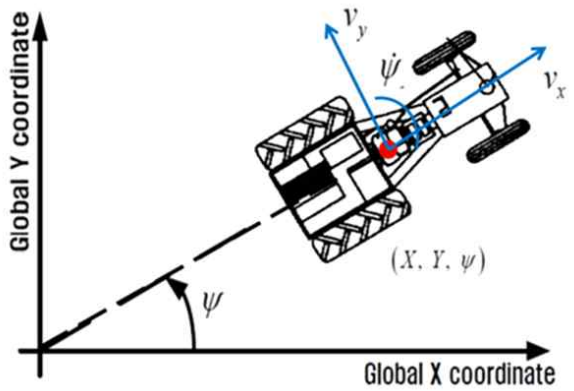


Fig. 3-17 Position of the extended kinematic model in field coordinates

$$\begin{cases} X = X_0 + \int v_x \cos\psi - v_x \beta \sin\psi \, dt \\ Y = Y_0 + \int v_x \sin\psi + v_x \beta \cos\psi \, dt \end{cases} \tag{3-34}$$

Fig. 3-18과 같이 트랙터 주행 모델을 시뮬레이터에 반영하였다. 이론적으로 시뮬레이션 작업 시작 전에 우선 트랙터의 특성수치를 입력이 필요하다. 토양의 조건을 반영하기 위해 전/후륜과 지면의 접촉특성을 Cornering Stiffness (Cf, Cr)로 설정하여 슬립 각을 증가 혹은 감소 가능하다. 또한 트랙터가 주행 시 발생된 슬립 각은 트랙터의 질량과 관성모멘트의 관계가 있어 초기설정이 필요하다. 초기상태에서 슬립 각, 요잉 각 및 요잉 율은 모두 0 이며 트랙터 선회 주행 시 경로추종 알고리즘에서 계산된 조향 각과 속도정보를 이용하여 트랙터의 슬립 각과 요잉 율 및 트랙터 위치정보를 계산한다. 트랙터 현재위치와 목표위치 판단하여 목표지점 도착 하지 않으면 계산된 슬립각과 요잉 율을 다시 피드백하게 되며 현재 트랙터에 발생된 슬립 각을 사용자에게 제공한다.

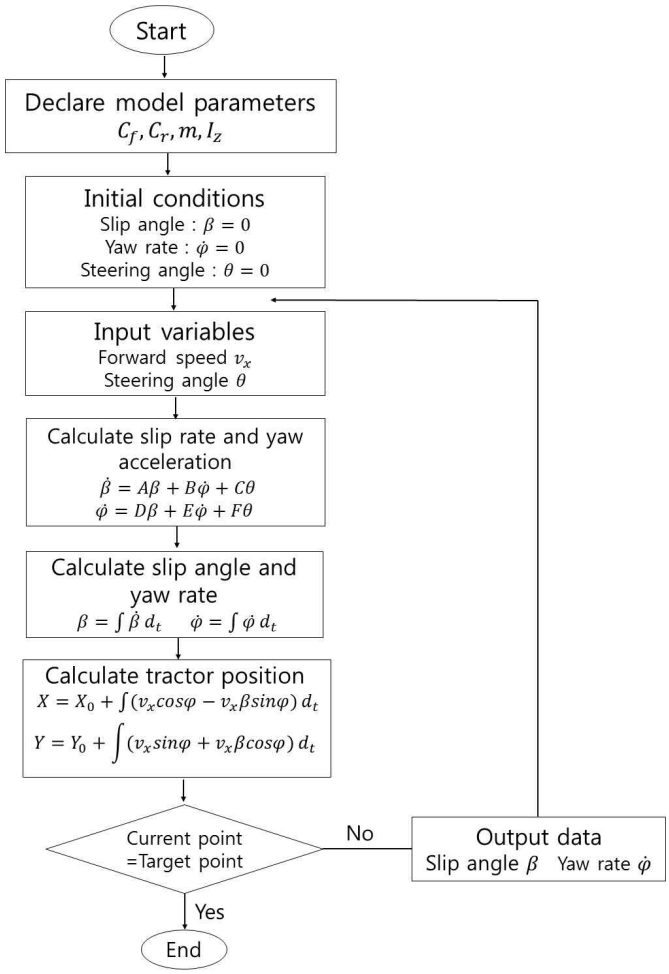


Fig. 3-18 Flowchart of determining slip angle and yaw rate

본 연구에서 동양 TX803 트랙터 CAD모델을 RoboticsLab rbuilder에 적용 및 모델 편집하기 위해 우선 모델 전환 작업을 진행하였다. rbuilder는 RoboticsLab을 이용한 트랙터 시뮬레이터 개발 과정에서 다양한 모델 파일을 작성하기 위한 가장 기본적인 모델링도구이며 전환된 모델 파일들은 XML을 기반으로 저장되며 일반적인 편집이 가능하다. Fig. 3-19는 CAD모델을 rbuilder에 적용할 수 있게 모델을 변환하는 모습이다.

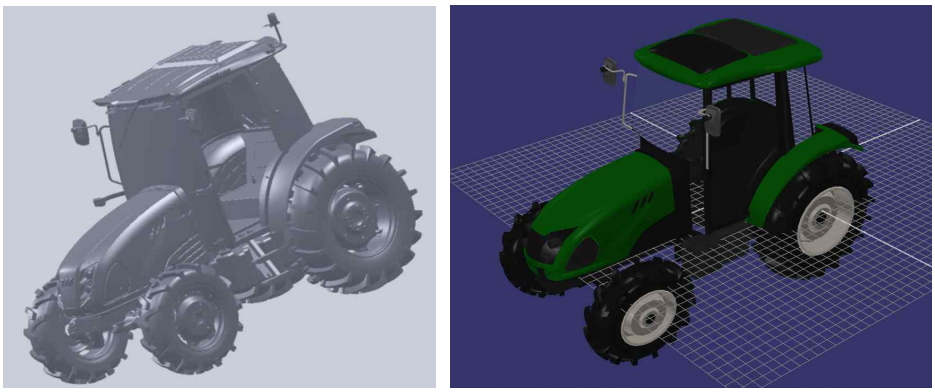


Fig. 3-19 3D tractor model used in the study

트랙터 본체의 각각의 부품관계를 편집하고, 링크와 조인트 속성을 설정하기 위해 다이어그램 편집기를 이용하였다. Fig. 3-20과 같이 시스템의 기구학적인 구조를 편집하고 트랙터 모델을 구축가능하며 자율주행 트랙터에 사용된 센서 및 액추에이터 등 확장하기 위해 모델 트리에서 수정이 가능한 구조이다. 본 연구에서는 GPS, IMU 및 조향 각 센서를 Fig. 3-21과 같이 장착하며 장착위치는 실제 자율주행 시스템과 동일하게 진행하였다.

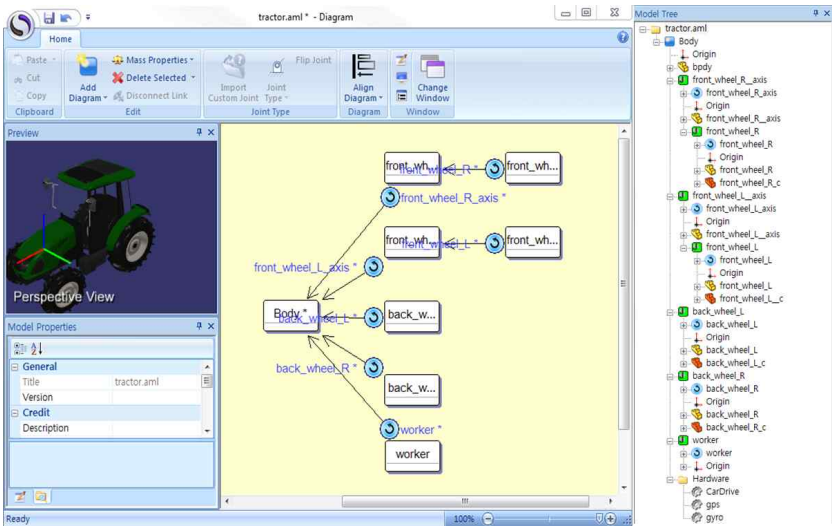


Fig. 3-20 Diagram view (tractor) for simulating engine

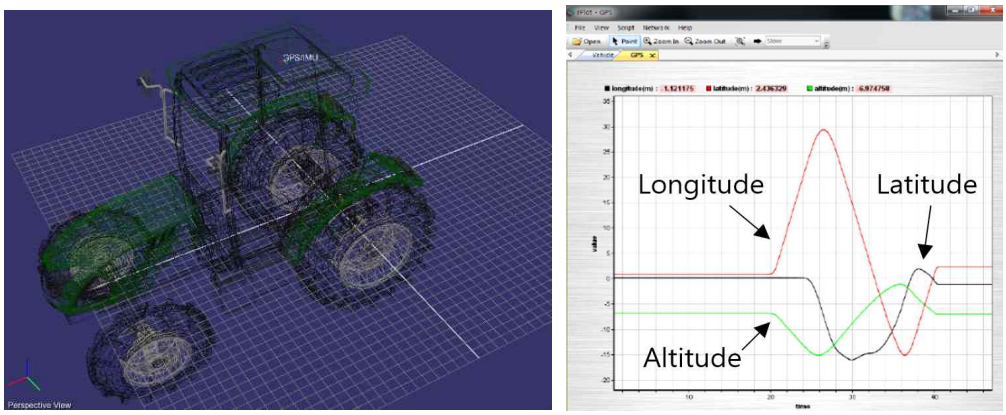


Fig. 3-21 Navigation sensors in the virtual simulator

트랙터 본체와 작업기 서로간의 편집과 부착은 Fig. 3-22와 같이 다이어그램 편집기를 이용하며 설계한 작업기를 트랙터 바디모델에 쉽게 연결가능하게 설계하였다. 관절간의 속성에 필요한 값도 사용자에 따라 재설정하며 작업기의 크기는 작업 경로생성에 큰 영향을 미치기 때문에 사용자가 원하는 크기로 조절이 가능하게 구축하였다.

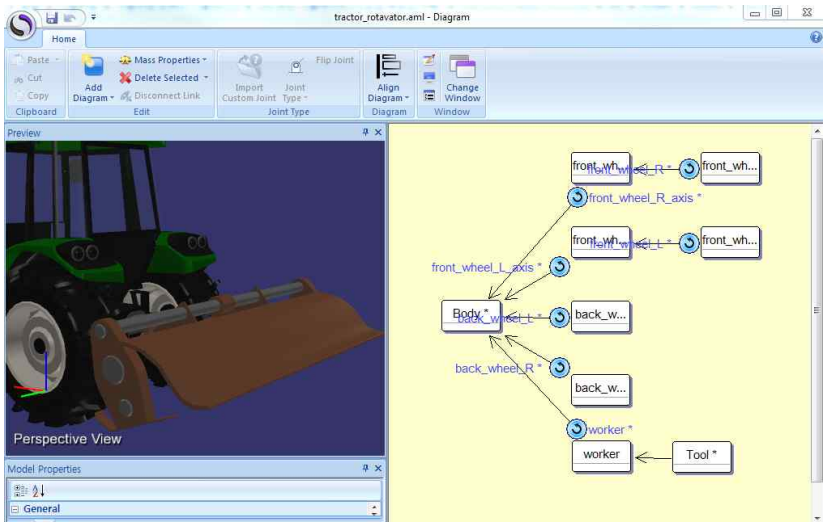
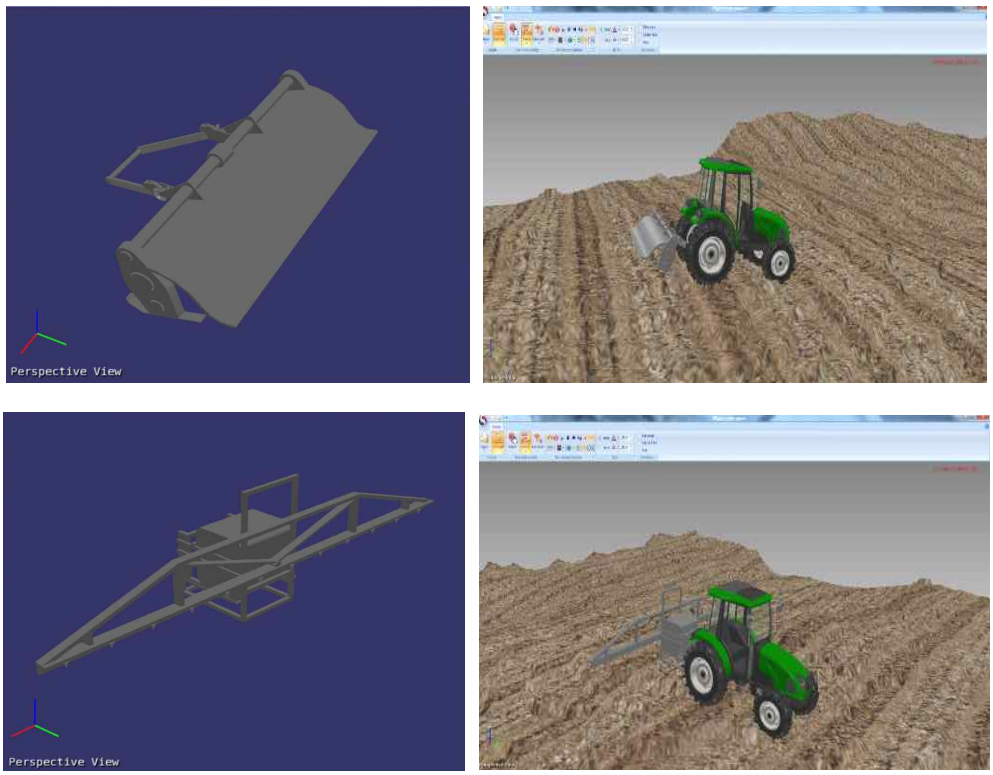


Fig. 3-22 Diagram view (tractor & implements) for simulating engine

경운/방제/파종 작업기를 사용자에 제공가능하게 개발하였다. 트랙터의 마력수에 따른 작업기 크기 변경이 가능하며 작업기 교체 및 추가는 구성한 작업기 모델을 선택 후 RoboticsLab의 환경에서 작업기 모델을 추가하여 사용가능하도록 구성했다(Fig. 3-23).



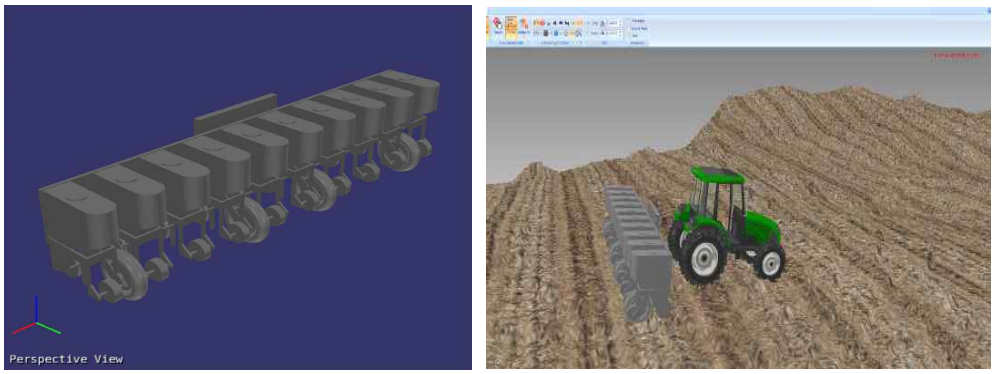


Fig. 3-23 Various implements for farm work in simulator environment

3.2.1.3. 3D 환경 경사지

경사지를 구성하기 위해 Height-map을 생성하였다. Height-map 생성은 시뮬레이터에 포함된 Height Map Gen.exe 프로그램을 사용하거나, bit-map 이미지 파일을 편집할 수 있는 이미지 제작 프로그램을 활용하였다. Fig. 3-24는 Height-map생성을 위한 설정화면이다. 생성하게 되는 Height-map의 이미지 크기를 설정하기 위해 Regular Square 체크박스를 이용 가로, 세로가 동일한 정사각형의 Height-map이 생성되며 Width, Height는 2보다 크거나 같은 값으로 요구하는 방식이다. Height-map의 평탄도를 결정하는 상수가 클수록 맵의 높낮이 차가 복잡해지고 작을수록 평탄한 지형이 만들어진다.

Height-map의 생성은 프로그램 내부에서 사용하는 보간 알고리즘(interpolator)을 적용하면 Linear, Cosine, Cubic을 선택 후 간단한 혹은 복잡한 지형이 생성이 가능하도록 하였다.

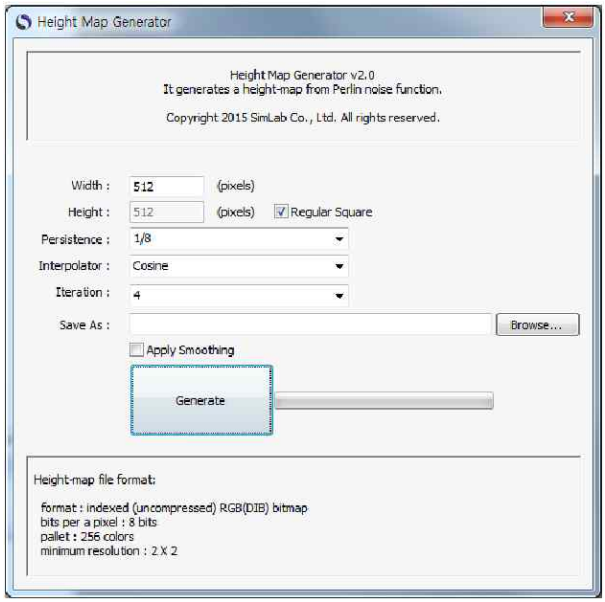


Fig. 3-24 3D map generation using Height Map Gen.exe

경사지 혹은 불규칙한 노면은 사용자가 비트맵 형태로 3차원 지도를 작성 후 시뮬레이터에서 3D 지형을 구성 가능하도록 하였으며, 3D 지형은 평탄한 지형에 Height-map을 이용해 높이 정보를 적용하여 구현되었다. Height-map은 전체 지형을 가로, 세로 일정 수의 격자로 나누고 각 격자의 높이를 2차원 배열(bit-map)에 저장한다. RoboticsLab의 환경 모델 파일인 EML은 Height-map 을 직접 지원하지 않지만 Fig. 3-25와 같이 Height-map 을 Import 하여 지형을 생성하는 기능을 제공하므로 맵을 생성하는 기술이다.

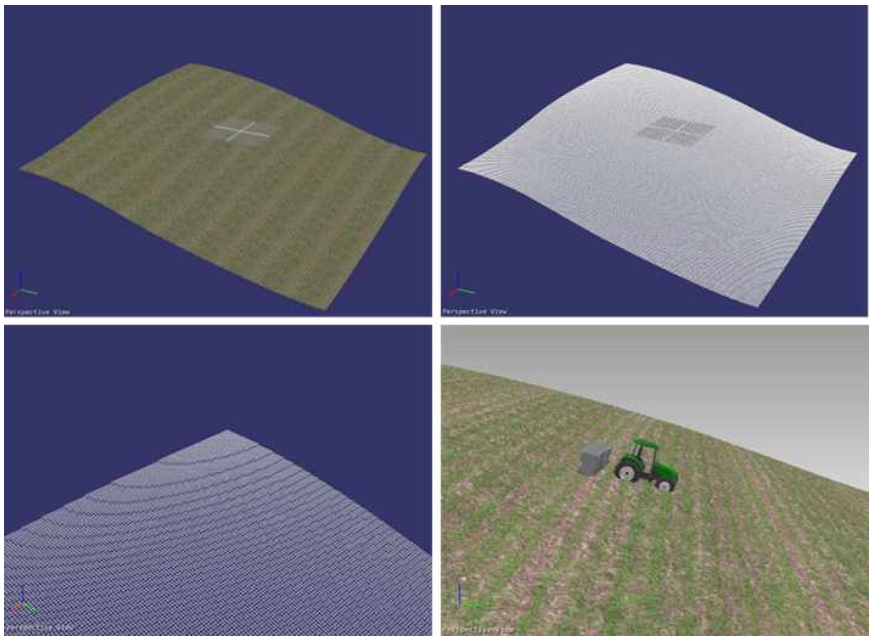


Fig. 3-25 3D map generation using bit-map

Height-map 생성 시 가로가 시뮬레이션 환경의 x축에 해당되며 세로는 y 축이다. Fig. 3-26(a)은 100x100(너비 100 픽셀, 세로 100 픽셀) 크기의 Height-map이다. 또한 Height-map을 일반 이미지 흑백 이미지로 나타난다. 여기서 색이 어두울 수 록 낮은 지형이며 밝을 수 록 고도가 높다. 실제 높이 값은 Height-map 설정 시 함께 설정하는 높이 최대와 최소 값 의해 결정하였다. Fig. 3-26(b)은 해당 Bit-map을 사용하여 생성된 3D 지형 예이다.

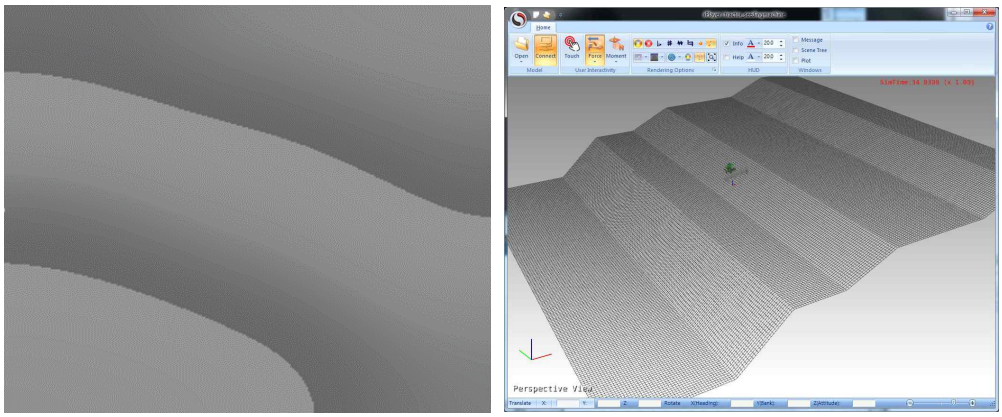


Fig. 3-26 Bit-map generation (a), 3D map generation in RoboticsLab (b)

3.2.1.4. 3D 지형에서 작업 경로 및 영역 표현

Waypoint를 시뮬레이터의 2D 평면환경에서 일정한 높이(0.5 m)를 두고 표시가 가능하며 3D 지형에서 way-point 표시를 위해 Fig. 3-27과 같이 Height-map의 높이 정보를 이용하도록 표시 모듈에서 구성 하였다.

또한 작업영역을 표현하기 위해 두 가지 장치(디바이스)를 구현하여 사용한다. GRID 디바이스를 사용하여 전체 작업영역 크기, grid 및 작업영역을 계산하는 방법 등을 정의하며 여기서 grid는 전체 작업영역의 크기와 분할 크기(resolution)를 정의한다. GRID 디바이스가 작업영역을 계산하기 위해서 작업기에 대한 정보가 필요하며 이 작업기는 GRID_TOOL 디바이스라는 별도의 디바이스로 트랙터의 작업기에 부착된다. 3D 지형에서 way-point 들의 표현과 작업영역 3D 표현을 Fig. 3-28과 같이 구성하였다.

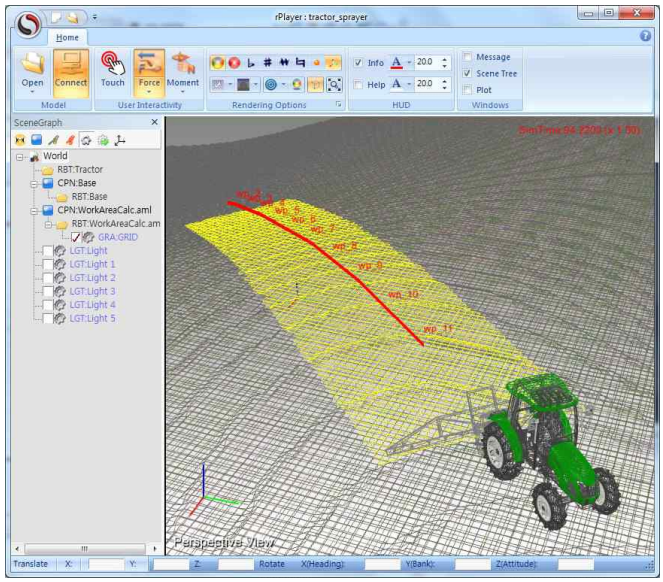


Fig. 3-27 Waypoint generation on the 3D plane

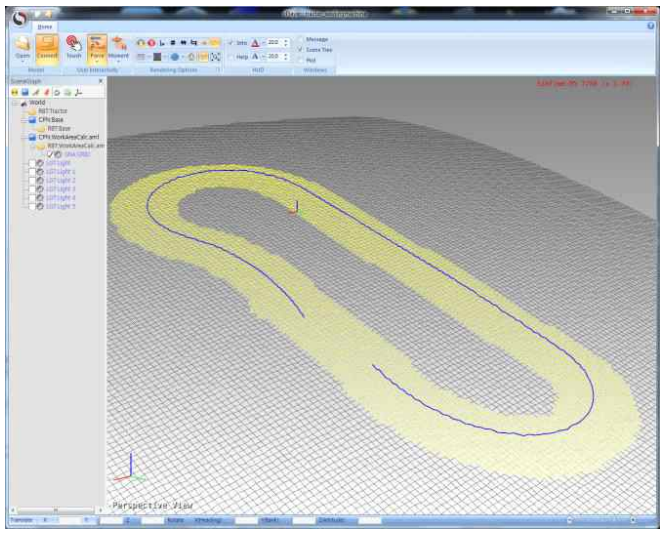


Fig. 3-28 Coverage area generation in the simulator environment

3.2.1.5. 서버와 클라이언트간의 통신 프로토콜

개발된 시뮬레이터에서 클라이언트와 서버간의 통신은 TCP/IP를 이용하였고, 데이터 통신을 위해 메시지 패킷(packet)을 구성하였다. 메시지는 헤더와 바디(데이터)로 구성되었으며 그 중 헤더는 Table 3-1과 같이 메시지의 처음 8 바이트이며, 메시지의 종류, 데이터 사이즈 등을 정의한다. 메시지 데이터는 헤더 다음에 추가되며 그 크기는 헤더의 Size 에 설정된 값과 같다. 현재는 16 바이트로 고정되어 있으며, 실제 패킷 정의에서 사용되지 않는 영역은 0으로 전송된다.

Table 3-1 Configuration of the message header

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	메시지 ID
	size	unsigned short	2byte	Data크기(bytes)
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0

메시지 패킷은 크게 송신부와 수신부 두 부분으로 구분되어있다. 각각 담당하는 역할에 따라 해당 패킷구성이 차이가 있다. 경로생성 및 추종 알고리즘의 제어기가 송신부로 담당하면 경로생성 추가(Table 3-2), 주행 속도 및 조향각 명령(Table 3-3), 작업기 각도 제어(Table 3-4), 작업기 상/하강 제어(Table 3-5), 작업기 폭(Table 3-6)과 같은 메시지 패킷을 구성하여 시뮬레이터로 전달하여 가성 시뮬레이션을 통해 트랙터 주행 상태를 보여준다. 시뮬레이터가 송신부로 담당하면 작업 영역(Table 3-7), 주행 슬립 각도(Table 3-8), 주행 반력(Table 3-9), GPS 위치 정보(Table 3-10), IMU 자세 정보(Table 3-11), 트랙터 실제 속도 및 조향 각도(Table 3-12)와 같은 메시지 패킷을 구

성하여 제어기로 전달하여 경로추종 알고리즘에 피드백 정보로 사용하도록 하였다.

Table 3-2 Configuration of waypoint message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=0
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	index	float	4byte	Index
	longitude	float	4byte	Longitude (m)
	latitude	float	4byte	Latitude (m)
	heading	float	4byte	Direction (deg)

Table 3-3 Configuration of velocity & steer angle message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=1
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	velocity	float	4byte	Speed (m/s)
	steer angle	float	4byte	Steer angle (deg)
	reserved_0	float	4byte	Unused
	reserved_1	float	4byte	Unused

Table 3-4 Configuration of implement angle message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=2
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	angle	float	4byte	PTO angle (deg)
	reserved_0	float	4byte	Unused
	reserved_1	float	4byte	Unused
	reserved_2	float	4byte	Unused

Table 3-5 Configuration of implement up/down message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=3
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	signal	float	4byte	“0”=up;
	reserved_0	float	4byte	“1”=down
	reserved_1	float	4byte	Unused
	reserved_2	float	4byte	Unused

Table 3-6 Configuration of implement width message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=4
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	width	float	4byte	Implement (m)
	reserved_0	float	4byte	Unused
	reserved_1	float	4byte	Unused
	reserved_2	float	4byte	Unused

Table 3-7 Configuration of working area feedback message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=5
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	total area	float	4byte	Field area (m^2)
	worked area	float	4byte	Worked area (m^2)
	reserved_0	float	4byte	Unused
	reserved_1	float	4byte	Unused

Table 3-8 Configuration of slip feedback message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=6
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	slip_CG	float	4byte	Slip_CG (deg)
	slip_front tire	float	4byte	Slip_front tire (deg)
	slip_rear tire	float	4byte	Slip_rear tire (deg)
	reserved_0	float	4byte	Unused

Table 3-9 Configuration of lateral force feedback message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=7
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	force_front tire	float	4byte	Force_front tire (N)
	force_rear tire	float	4byte	Force_rear tire (N)
	reserved_0	float	4byte	Unused
	reserved_1	float	4byte	Unused

Table 3-10 Configuration of GPS signal feedback message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=8
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	x	float	4byte	Longitude_GPS(m)
	y	float	4byte	Latitude_GPS(m)
	z	float	4byte	Altitude_GPS(m)
	reserved_0	float	4byte	Unused

Table 3-11 Configuration of IMU signal feedback message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=9
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	yaw	float	4byte	Yaw (deg)
	pitch	float	4byte	Pitch (deg)
	roll	float	4byte	Roll (deg)
	reserved_0	float	4byte	Unused

Table 3-12 Configuration of velocity & steer angle feedback message packet

	Name	Type	Size	Description
Header (8 byte)	id	unsigned short	2byte	ID=10
	size	unsigned short	2byte	16
	reserved	short	2byte	0
	flag	unsigned char	1byte	0
	checksum	char	1byte	0
Body (16 byte)	velocity	float	4byte	Actual speed(m/s)
	steer angle	float	4byte	Actual steer angle (deg)
	x	float	4byte	Longitude_COG (m)
	y	float	4byte	Latitude_COG (m)

3.2.1.6. 시뮬레이터 활용 기술

(1) 포장 작업효율 분석

개발한 주행 알고리즘을 실제 트랙터에 적용 전에 자율주행 성능을 분석하기 위해 개발 시뮬레이터를 이용하여 농작업 선회패턴별 작업 성능과 효율을 가상 환경 조건에서 비교 분석하고자 하였다. 이를 진행하기 위해 100 m × 40 m 크기의 포장에서 C형, X형, R형 선회 방식별 포장 작업효율의 차이를 분석하고 미경지 부분의 발생 여부를 확인하고자 하였다. 또한 주행 경로생성 프로그램을 이용 포장의 크기별 경로생성 후 시뮬레이션을 수행 하여 포장 크기별 선회유형에 따라 포장 작업효율을 비교 분석하였다. 포장 작업효율은 선행연구 결과 (김학규, 1995)에 참고하여 계산하였다.

한 포장 내에서 포장작업 소요시간은 작업 시작부터 완료까지 총 소요 시간(T)을 의미하는 것으로 Eq. (3-35)과 같이 정의된다. 포장 내에서 작업 손실시간(T_{loss})은 작업 준비, 선회, 포장 진출입, 조정 및 정비 등으로 발생하게 되

는데 본 연구에서는 자율주행 트랙터가 작업 시작부터 완료까지 정지 없이 진행하기 때문에 작업 손실은 선회 소모 시간만 고려하였다.

$$T = T_e + T_{loss} = T_e + T_{turn} \tag{3-35}$$

여기서, T= 포장 작업 시간 (min)

T_e = 포장 유효 작업 시간 (min)

T_{loss} = 소모 시간 (min)

T_{turn} = 회전 소모 시간 (min)

유효 작업시간은 작업기가 목적하는 작업을 수행하는데 소요되는 시간으로서 유효 작업 폭과 유효 작업 속도로 인해 Eq. (3-36)과 같이 나타낸다.

$$T_e = \frac{xy}{60vw} \tag{3-36}$$

여기서, x= 포장 장변 길이 (m)

y= 포장 단변 길이 (m)

v= 작업 속도 (m/s)

w= 유효 작업 폭 (m)

유효 작업시간은 중복 작업을 포함한 작업시간으로서 중복 작업을 제외하면 이론 작업시간은 Eq. (3-37)과 같이 나타낸다.

$$T_{th} = \frac{xy}{60vW} = \frac{\varepsilon_w xy}{60vw} = \varepsilon_w T_e \tag{3-37}$$

여기서, W= 이론 작업 폭 (m)

ε_w = 유효 작업 폭의 비율 (decimal)

포장작업 소요시간과 이론 작업시간으로써 포장 작업효율은 Eq. (3-38)와 같이 나타낸다.

$$\varepsilon_f = \frac{T_{th}}{T} = \frac{\varepsilon_w T_e}{T} \tag{3-38}$$

여기서, ε_f = 포장 작업효율 (decimal)

(2) 작업 미경지 구역 분석

개발한 시뮬레이터는 시간 기반의 포장 작업효율 분석이 가능하며 면적 기반의 작업면적 비율도 함께 사용자에게 제공한다. 포장 작업면적 비율은 (작업 완료 영역 면적)/(포장 전체 영역 면적)으로 정의했다. 작업 전체 영역 면적은 GRID 장치에 의해 정의된 전체 작업 면적으로 단위가 픽셀 이므로 (GRID row 개수) X (GRID column 개수)로 정의했다. 작업 완료 영역 면적은 작업기 모델에 의해 계산된 작업 영역을 누적한 것으로 작업 완료 영역의 면적은 coverage 픽셀의 개수로 계산하였다. Fig. 3-29은 한 예로서 노란색은 작업한 면적을 의미하고 검은색은 미작업한 부분을 의미한다.

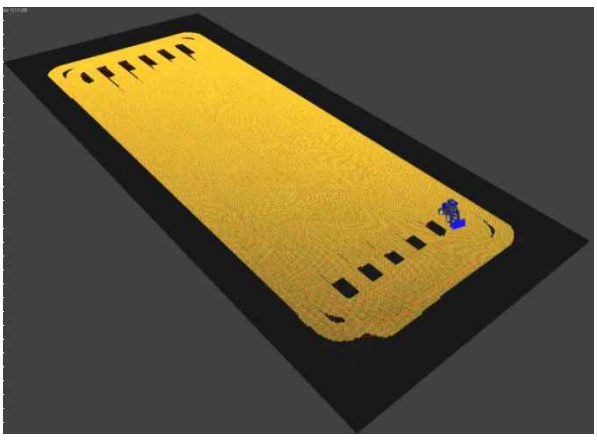


Fig. 3-29 Comparison of the coverage areas based on different driving path

(3) 동적 경로추종 알고리즘의 성능 분석

트랙터의 중량, 관성 모멘트, 주행 속도 및 토성에 따라 타이어와 노면간의 접촉성 Cornering Stiffness가 변화가 있고 이러한 변화가 노면 반력에 의해 슬립 각이 발생하게 된다. 실제 포장에서 회전주행 시 이러한 슬라이딩 현상을 표현하기 위해 사용한 Extended Kinematic Bicycle Model에서 표현하는 슬립 각의 생성은 앞쪽과 뒤쪽 코너링 강성인 Cf, Cr 두 개 요소에 주 영향을 받는다. 실제 포장에서 자율주행 시 타이어와 노면간의 특성을 파악하기 어려움이 있어 시뮬레이터에서는 Cf, Cr 값을 변경하면서 시뮬레이션에서 슬립의 효과를 부가하고자 하였다. Cf, Cr 값을 이용 토양수분 높은 혹은 낮은 상태인 다종 노면상태로 설정이 가능하게 구성하여 트랙터 슬라이딩 효과를 나타낸다(Fig. 3-30). 즉 앞 & 후륜 슬립 각, 바디 슬립 각, 조향각과 전방-횡방 속도 취득이 가능하게하며 Cornering Stiffness의 변화에 따른 주행 슬립현상 구현이 가능하다.

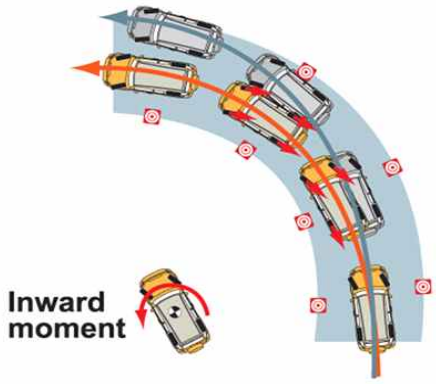


Fig. 3-30 Evaluation test in the presence of sliding effect

Fig. 3-31은 시뮬레이터 환경의 노면상태를 변화 설정하는 화면이다. 개발한 가상 시뮬레이터는 주행 모델 기반으로 반력과 슬립 각을 계산하기 때문에 트

랙터 모델이 변경하게 되면 설정 화면에서 변경된 트랙터의 제원기반으로 변경 설정이 필요하다. 시뮬레이터 상에서 슬립 환경이 재연이 가능하기 때문에 앞서 설계된 주행 슬립 예측기반의 동적 경로추종 제어기를 시뮬레이터 상에서 주행 성능 평가를 진행하였다.

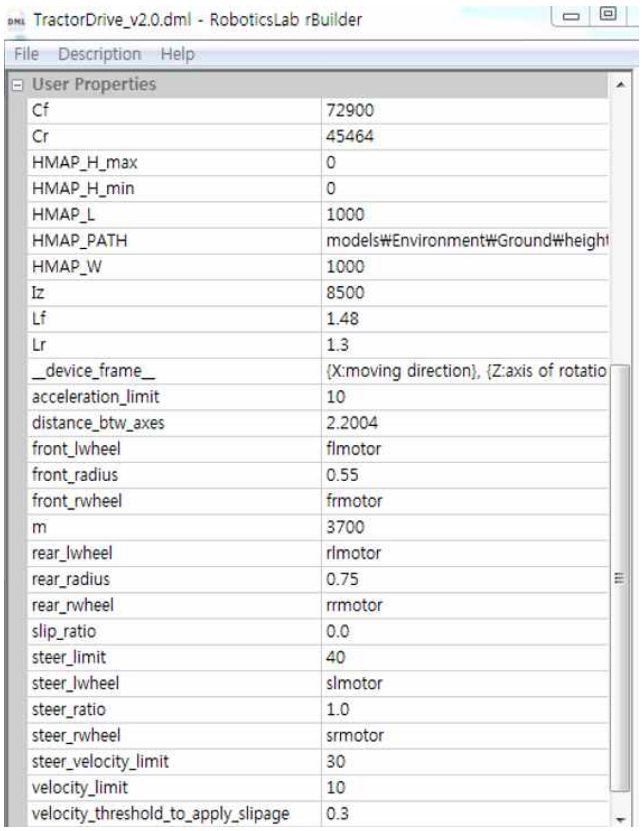


Fig. 3-31 Environment editor for changing the road condition

(4) 자율주행 농작업을 위한 요인 실험

개발한 자율주행 가상 시뮬레이터를 사용하여 두 가지 자율주행 농작업 요인 가상실험을 수행하였다. 첫 번째로 개발한 시뮬레이터에서 3D 환경지면이 구성 가능하기 때문에 경로 추종 알고리즘에 앞서 소개한 GPS 신호보정 알고

리즘을 적용하여 주행 성능평가를 진행하였다. 개발한 가상 환경에서 복잡한 지면환경을 구축이 가능하기 때문에 효율적으로 GPS 신호 보정 효과를 분석하였다. 두 번째 실험은 개발한 자율주행 가상 시뮬레이터에서 작업기 제어 및 작업영역이 표시 가능하기 때문에 개발한 미경지 영역을 방지를 위한 작업기 제어 알고리즘을 평가하였다. 세 번째로는 슬립의 발생하는 경우 개발한 강인 경로추종 알고리즘이 효과가 있는지를 시뮬레이션에서 확인하고자 하였다. 관련 매개변수는 주어진 트랙터 및 작업기 제원을 이용 입력하였고 포장의 조건도 다양하게 변경하면서 요인연구를 수행하였다.

3.3. 결과 및 고찰

3.3.1. 선회 패턴 변화에 따른 경로생성 성능 평가

개발한 경로생성 알고리즘을 이용하여 가상 시뮬레이터 환경에서 경로생성 성능을 평가하였다. 트랙터와 작업기의 제원은 Table 2-3과 같이 사용 하였으며 한국의 표준 포장 구획 0.4 ha (100 m × 40 m)에서 경로생성 후 시뮬레이션 평가하였다. Fig. 3-32는 C형 선회패턴으로 생성된 작업 경로를 사용 주행 궤적을 나타낸 것이며 시뮬레이터 환경에서 상대적 작은 미경지 영역이 있는 것을 나타났다. Fig. 3-33은 X형 선회패턴으로 생성된 작업 경로를 사용 주행 궤적을 나타낸 것이며 시뮬레이터 환경에서 C형 선회 패턴과 비슷하게 상대적 작은 미경지 영역이 있는 것을 나타났다. Fig. 3-34는 R형 선회패턴으로 생성된 작업 경로를 사용 주행 궤적을 나타낸 것이며 시뮬레이터 환경에서 상대적 많은 미경지 영역이 있는 것을 확인할 수 있었다.

Fig. 3-35와 같이 개발한 시뮬레이터를 이용 한국의 표준 포장 구획에서 3가지 선회 패턴 경로를 기반으로 포장 작업시간, 유효 작업시간 및 포장 작업효율을 분석하였다. X형 선회 패턴 기반으로 생성된 경로는 유효 작업시간은 26.2 min과 포장 작업시간은 30 min으로 나타났으며 포장 작업효율은 87.38%

로 가장 높은 것으로 나타났다. R형 선회 패턴 기반으로 생성된 경로는 순 작업시간은 24.9 min과 전체 작업시간은 29.5 min으로 나타났으며 작업효율은 84.43%로 가장 낮았으며 시뮬레이터 환경 상에서 미경지부분이 가장 많은 것으로 판단하였다. 3가지 경로를 비교하여 X형 선회 기반의 작업 경로가 가장 접합한 것으로 나타났다.

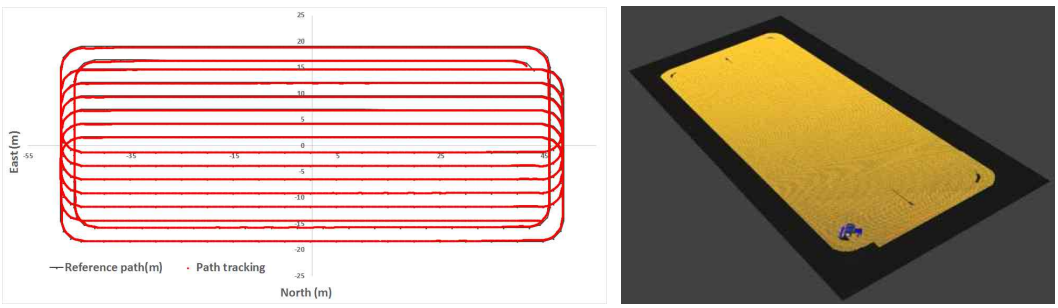


Fig. 3-32 Result of simulation analysis for path generation by C-type pattern

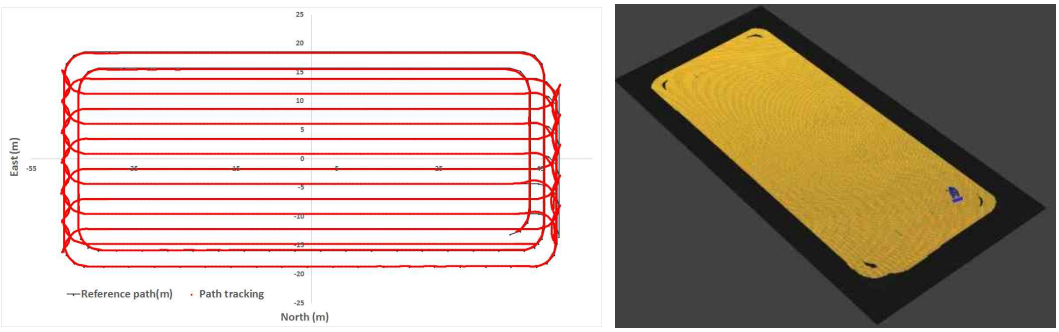


Fig. 3-33 Result of simulation analysis for path generation by X-type pattern

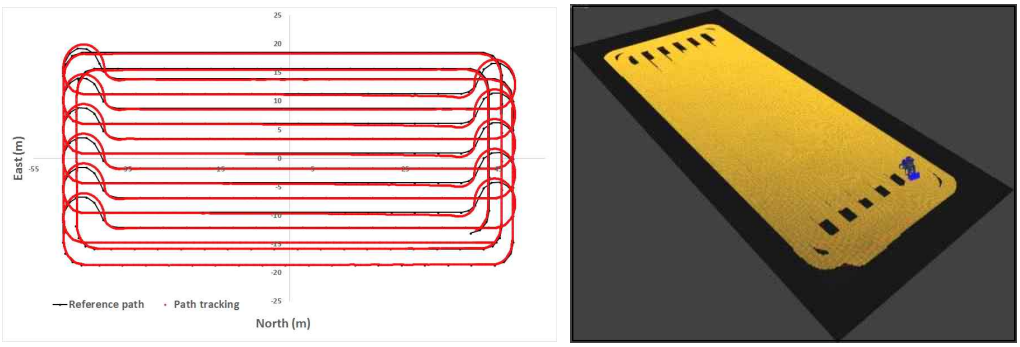


Fig. 3-34 Result of simulation analysis for path generation by R-type pattern

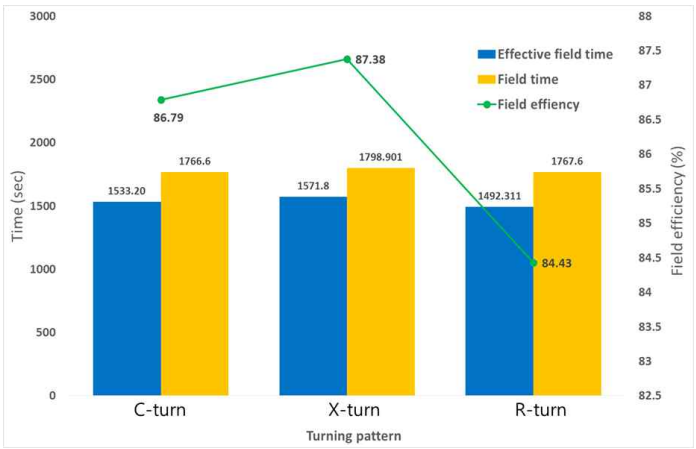


Fig. 3-35 Result of simulation analysis by C-type, X-type and R-type pattern path

3.3.2. 자율주행 동적 경로추종 알고리즘 성능 평가

본 절에서는 개발한 강인 제어기반의 동적 경로추종 알고리즘의 주행 성능을 평가하기 위해 한웅철 석사학위논문(2013)에 자율주행 트랙터(TX803 모델)에 적용된 Look-ahead 경로추종 알고리즘을 이용 가상 시뮬레이터 환경에서 비교분석을 진행하였다.

전체 포장에 대해 자율주행 작업 성능을 평가하기 위해 길이가 50 m, 폭이 20 m 크기의 포장에서 C형 선회 패턴 기반의 내부 왕복 작업 경로를 생성하였다. 시뮬레이터 환경에서 슬립 없는 노면과 슬립 존재한 노면 상태를 구성하여 각 노면상태에서 두 가지 경로추종 알고리즘의 주행 성능을 각각 평가하였다. 개발한 시뮬레이터에서 노면 상태 변수인 Cornering stiffness를 Cf=72900 N/rad, Cr=45464 N/rad로 설정하여 Fig. 3-36과 같이 타이어와 노면 간에 슬립이 존재하지 않는 환경으로 구성하였다. 슬립이 존재하지 않는 환경에서 Fig. 3-37(a), (b)과 같이 Look-ahead 경로추종과 강인 제어 기반의 동적 경로추종 알고리즘을 이용한 자율주행 궤적이 큰 차이 없는 것으로 나타내었다.

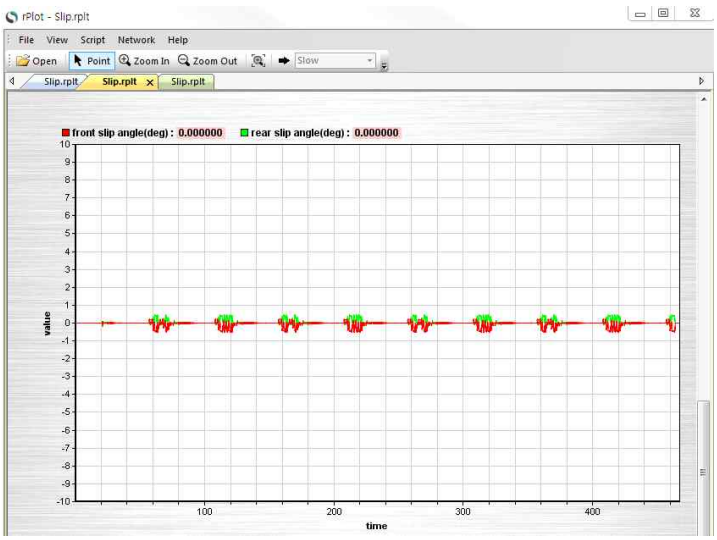
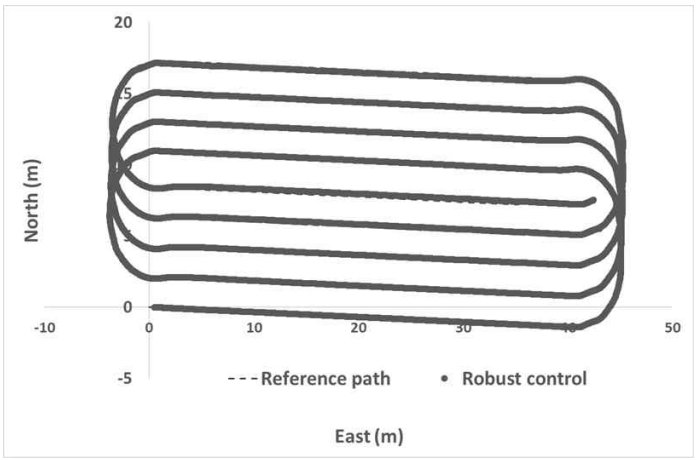
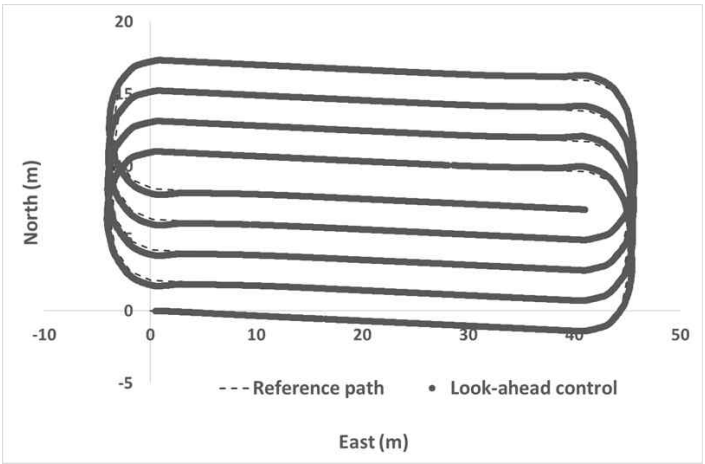


Fig. 3-36 No slip environment on the developed simulator



(a)



(b)

Fig. 3-37 Result of the simulation analysis of full path tracking trajectory by look-ahead (a) and robust control (b) from the no slip environment

노면 상태 변수인 Cornering stiffness를 Cf=5000 N/rad, Cr=3000 N/rad로 설정하여 Fig. 3-38과 같이 타이어와 노면 간에 슬립이 존재하는 환경으로 구성하였다. 슬립이 존재하는 환경에서 Fig. 3-39(a), (b)와 같이 Look-ahead 경

로추종과 강인 제어 기반의 동적 경로추종 알고리즘을 이용한 자율주행 궤적이 차이가 있는 것으로 나타났다. 특히 선회 구역에서 왕복 작업구역으로 진입 시 동적 경로추종 알고리즘을 이용할 때 목표 경로와 더 근접하는 궤적을 보였다.

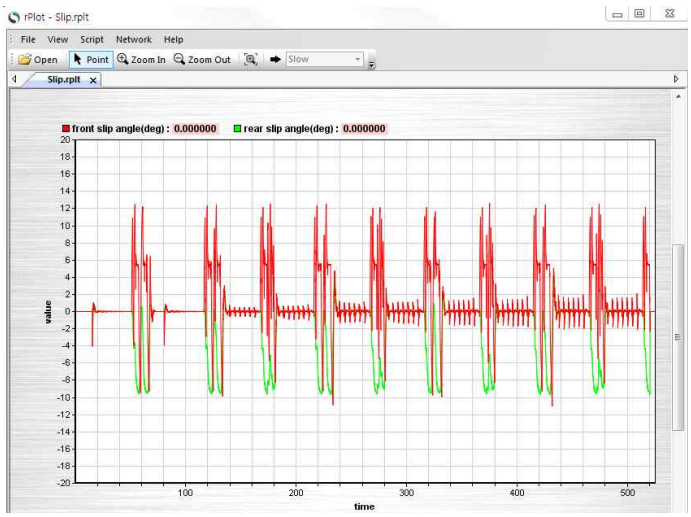
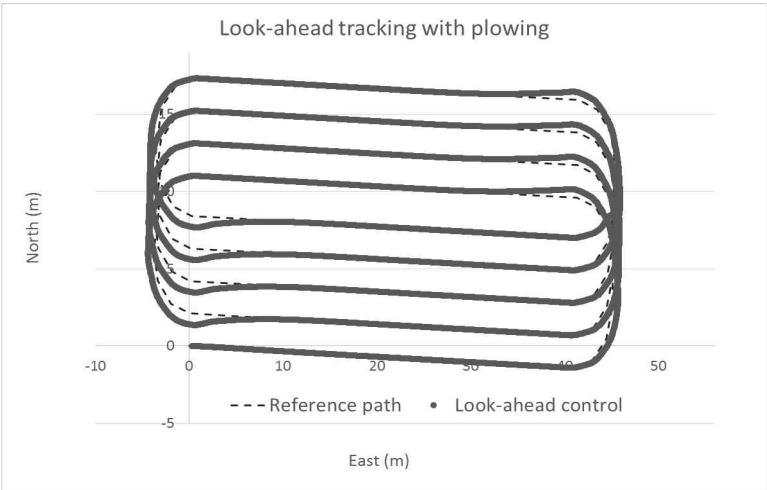
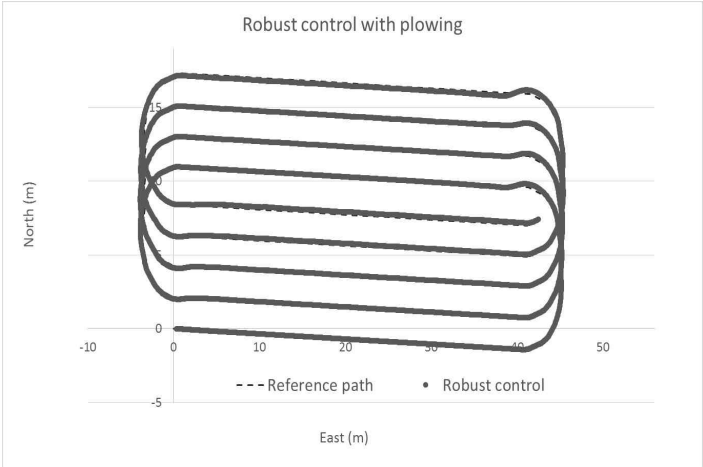


Fig. 3-38 Large slip environment on the developed simulator



(a)

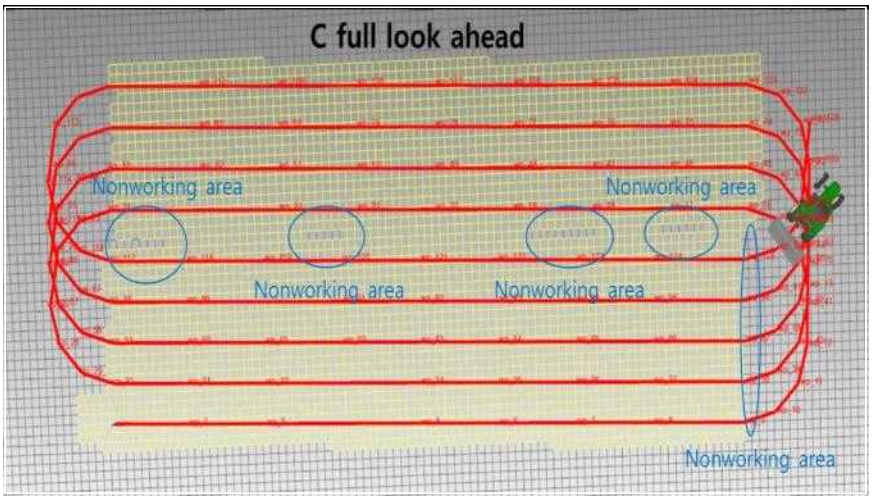


(b)

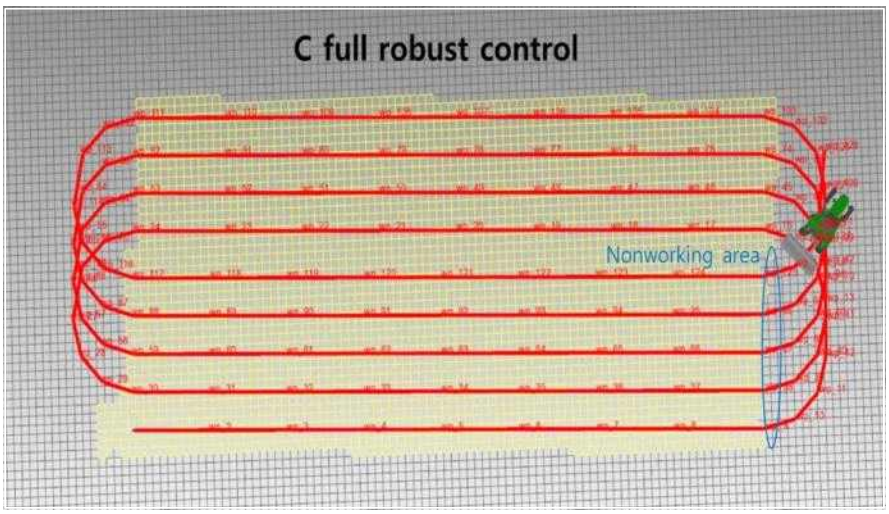
Fig. 3-39 Result of the simulation analysis of full path tracking trajectory by look-ahead (a) and robust control (b) from the large slip environment

전체 주행의 RMS 위치 오차는 Table 3-13과 같이 동적 경로추종 알고리즘은 10.9 cm로 나타났으며 Look-ahead 경로추종 기반의 RMS 위치 오차인 16.4 cm 보다 작게 나타낸 것으로 판단되었다. 또한 개발한 가상 시뮬레이터 환경에서 작업 영역을 표시 가능하기 때문에 Fig. 3-40과 같이 미 작업 영역을 사용자에게 제공이 가능하였다. 동적 경로추종 알고리즘을 사용 시 트랙터가 새머리 진입 영역 부근에서 미 작업 영역을 11.8 m^2로 생겼으나 Look-ahead 경로 추종 알고리즘을 사용 시 새머리 영역 진입과 내부 왕복 작업부에서 발생한 미 작업 영역을 포함하여 총 18.3 m^2로 동적 경로 추종 보다 더 많은 미 작업 영역을 발생한 것을 나타났다.

포장의 전체 작업 성능을 평가하기 위해 Table 3-14와 같이 트랙터의 실제 작업 면적, skip면적 및 포장 작업효율 등을 제시하여 비교분석하였다. 그 결과, 동적 경로 추종 알고리즘을 사용 시 skip 면적은 7.4 m^2 감소하고 선회 시간은 12.9 초 절약하여 포장 작업효율이 3.9% 상승하는 것으로 나타났다.



(a)



(b)

Fig. 3-40 Result of the simulation analysis of non working area by look-ahead (a) and robust control (b) from the large slip environment

Table 3-13 Results of the simulation comparison of full path tracking RMSE by look-ahead and robust control from the large slip environment

Working types	RMSE
Look-ahead control	16.4 cm
Robust control	10.96 cm

Table 3-14 Results of the simulation comparison of operation performance by look-ahead and robust control from the large slip environment

	Ideal field area	Skip area	Field time loss in turning	Field efficiency	Ratio of skip area
Look ahead	840 m^2	18.3 m^2	149.2 sec	68%	2.2%
Robust control	840 m^2	11.8 m^2	136.3 sec	71.9%	1.4%

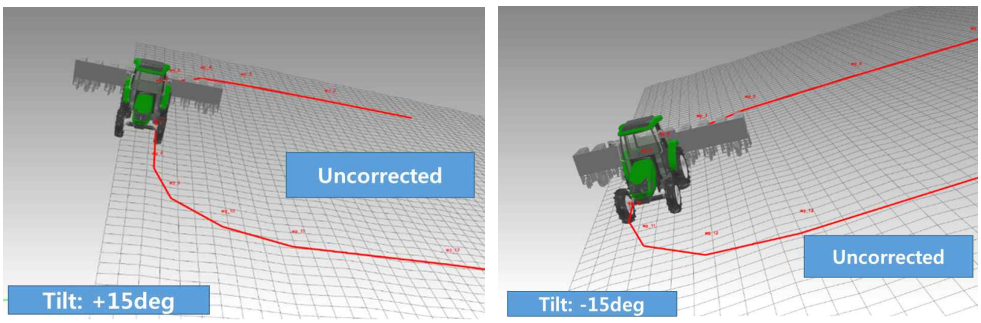
3.3.3. 경사지면에서 경로추종을 위한 GPS 신호 보정 평가

본 절에서는 포장의 경사지면 혹은 불규칙한 노면에서 IMU의 롤, 피치 값을 이용 정밀한 경로추종을 위해 앞서 제시한 GPS 위치 보정 알고리즘을 적용하였으며 그의 성능평가는 가상 시뮬레이터 환경에서 진행하였다.

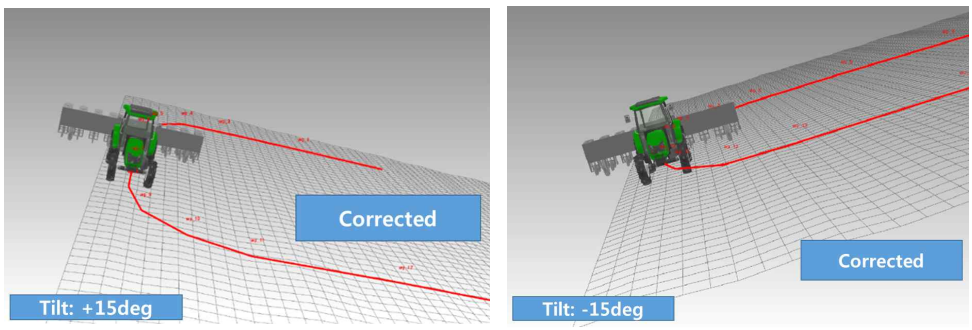
Fig. 3-41은 가상 시뮬레이터 환경에서 C형 선회 패턴의 일부분 경로를 생성하고 좌/우 15도의 경사지면을 각각 구성하여 트랙터가 기울어진 상태에서 알고리즘을 적용 전과 후의 경로추종의 시뮬레이션 모습이다. Fig. 3-41(a)은 트랙터가 좌/우 15도의 경사지면에서 GPS 위치 오차 보정 알고리즘을 적용 전 결과를 각각 나타내었다. 트랙터가 기울어진 상태에서 주행 시 GPS 안테나의 수직방향으로 투영된 좌표와 트랙터의 무게중심 점이 동일한 수직선상에 겹쳐져 있지 않아 GPS 오차가 발생하게 되어 트랙터가 목표 경로선(빨간색)

상에서 주행이 안되는 것을 확인할 수 있었다.

Fig. 3-41(b)은 트랙터가 좌/우 15도의 경사지면에서 GPS 위치 오차 보정 알고리즘을 적용 후 결과를 각각 나타낸 것이다. 트랙터가 기울어진 상태에서 주행 시 GPS 안테나의 수직방향으로 투영된 좌표와 트랙터의 CG점이 동일한 수직선상에 보상하였기 때문에 GPS 오차를 감소하여 트랙터가 목표 경로선(빨간색) 상에서 주행이 가능한 것으로 나타났다.



(a)



(b)

Fig. 3-41 Result of the simulation analysis of tracking performance by the GPS uncorrected (a) and the corrected (b) on the left and right tilted plane

Fig. 3-42(a)와 (b)는 트랙터가 좌/우 15도의 경사지면에서 GPS 위치 오차 보정 알고리즘 적용전과 후의 주행 궤적을 각각 나타낸 것이다. 트랙터가 선

회 구간에 진입 전, 즉 롤 방향에 영향을 받지 않을 때 알고리즘 적용 전(빨간색)과 후(검정색)의 주행 궤적이 큰 차이가 보이지 않았으나 트랙터가 선회 구간으로 진입하여 롤 방향의 영향을 받게 되면 알고리즘 적용 후의 주행 궤적이 목표 경로(노란색)와 더 근접한 것을 나타났다. 좌/우 15도의 경사지면에서 목표 경로와 각각 30 cm / 20 cm 이내의 오차로 접근할 수 있어 알고리즘 적용 전 보다 빨리 감소하여 안정된 주행 상태로 유지하는 것을 확인하였다.

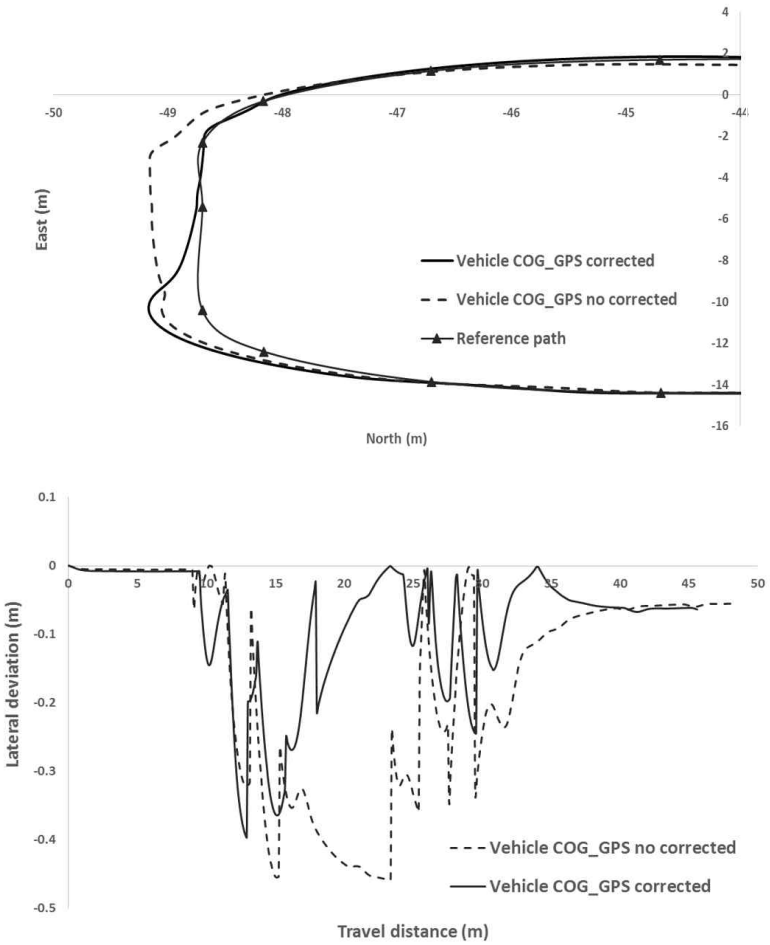


Fig. 3-42 (a) Result of the simulation analysis of tracking trajectory and lateral deviation on the left tilted plane

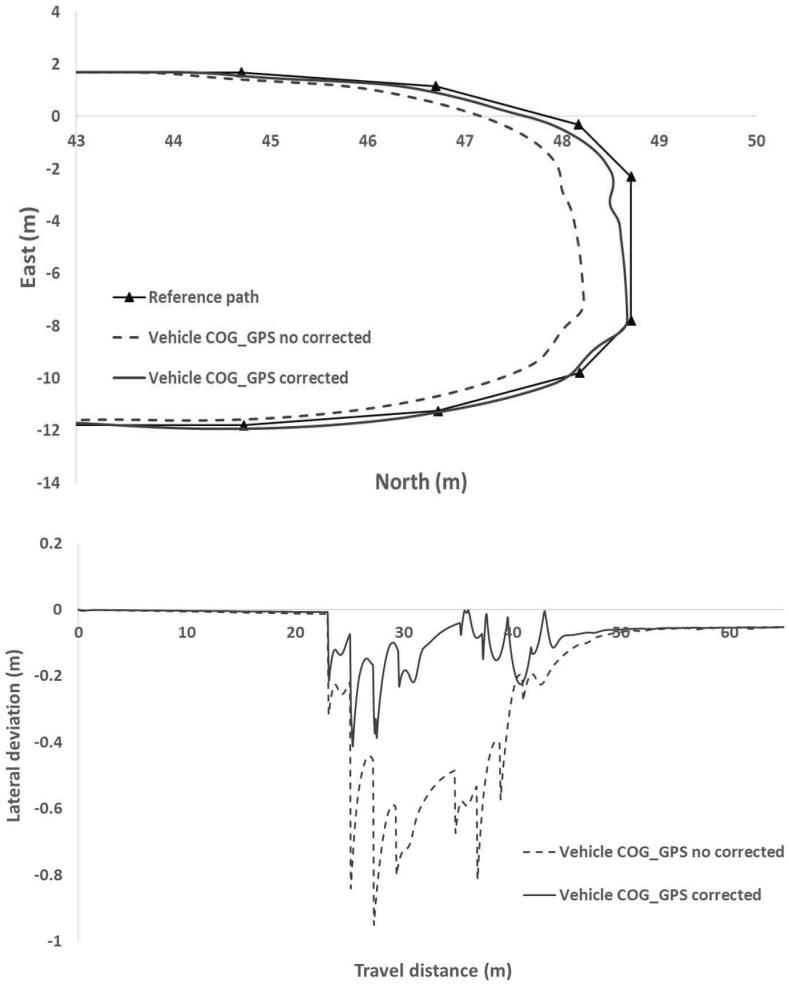


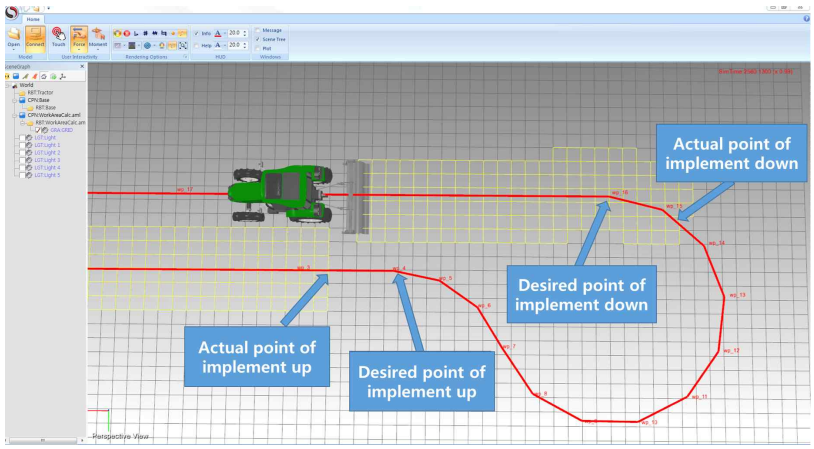
Fig. 3-42 (b) Result of the simulation analysis of tracking trajectory and lateral deviation on the right tilted plane

3.3.4. 미경지 영역 방지를 위한 작업기 제어 평가

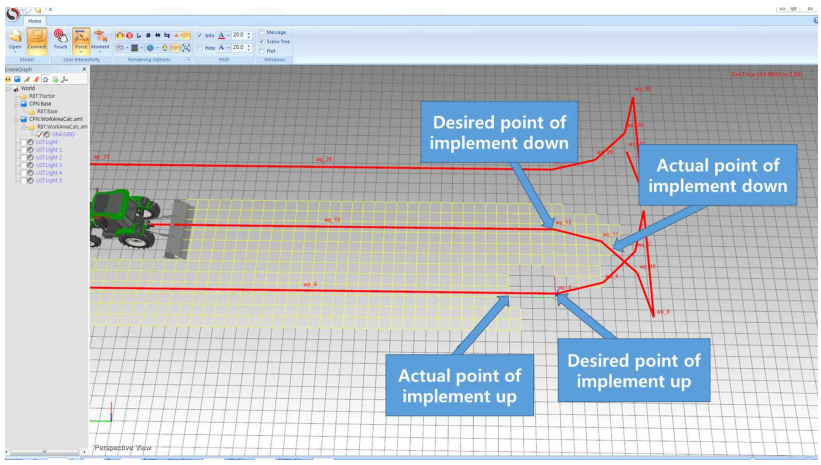
트랙터가 새머리 영역으로 진입 및 새머리 영역에서 탈출 시 작업기 작동 요구된 특정 경로점에서 정확히 작업기를 상승/하강 작동하기 위해 개발한 알고리즘을 적용하여 가상 시뮬레이터 환경에서 성능 평가하였다. 성능을 평가

하기 위해 R형 선회 패턴과 X형 선회 패턴의 일부분 경로를 적용하여 진행하였다. 가상 시뮬레이터 환경에서 작업영역을 통해 작업기 적시 작동역부를 판단이 가능하였다. Fig. 3-43(a), (b)은 R형 선회 패턴과 X형 선회 패턴 경로에서 알고리즘 적용 전 결과를 각각 나타내었다. 작업기의 작업 영역(노란색)을 통해 트랙터가 새머리 영역으로 진입할 때 작업기의 상승 명령을 요구된 경로점에 도달 하지 않고 상승하였으며, 새머리 영역에서 탈출 시 작업기의 하강 명령을 요구된 경로점에 도달 못해 하강하였다. 이러한 문제로 무인 농작업 진행 시 미경지 부분이 증가하는 문제로 포장 작업효율이 낮아지며 동역소모 증가의 단점이 있는 것으로 판단된다.

Fig. 3-44(a)와 (b)는 R형 선회 패턴과 X형 선회 패턴 경로에서 알고리즘 적용 후 결과를 각각 나타내었다. 작업기의 작업 영역(노란색)을 통해 트랙터가 새머리 영역으로 진입할 때 작업기의 상승 명령을 요구된 경로점에 도달하여 상승하였으며, 새머리 영역에서 탈출 시 작업기의 하강 명령을 요구된 경로점에 도달 후 하강하였다.

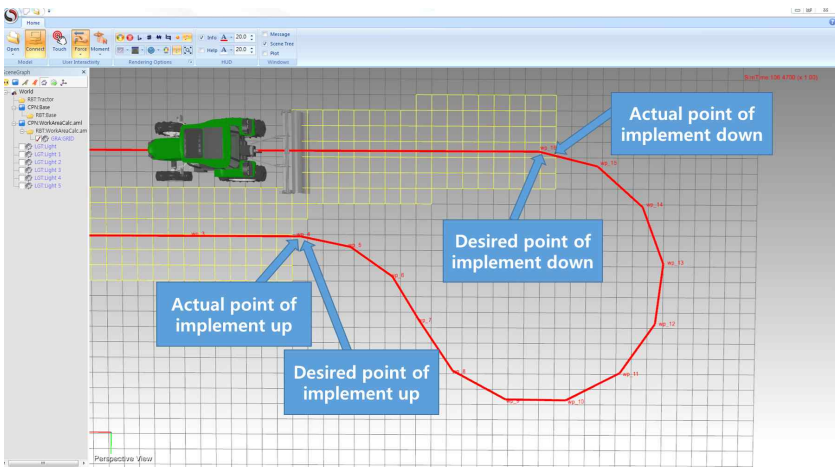


(a)

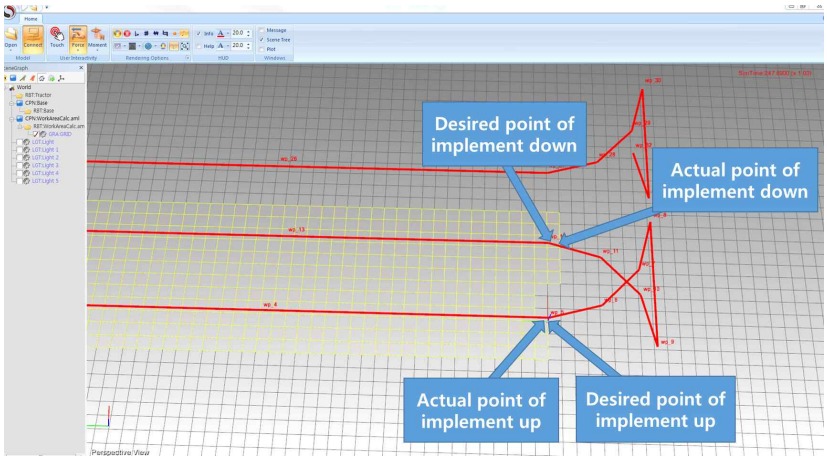


(b)

Fig. 3-43 Result of the simulation analysis of implement control by no algorithm applied on the R-type(a) and X-type(b) pattern path



(a)



(b)

Fig. 3-44 Result of the simulation analysis of implement control by the developed algorithm application on the R-type(a) and X-type(b) pattern path

Fig. 3-45는 R형 선회 패턴 경로를 사용한 주행 궤적에서 작업기 제어 알고리즘 적용 전과 후의 작업기 실제 작동 위치점을 나타내었다. 알고리즘 적용 후 트랙터가 새머리 영역으로 진입할 때 RMS 위치 오차가 2.15 m에서 0.14 m로 감소하였으며, 새머리 영역에서 탈출 시 RMS 위치 오차가 2.01 m에서 0.35 m로 감소하는 것을 확인하였다. Fig. 3-46은 X형 선회 패턴 경로를 사용한 주행 궤적에서 작업기 제어 알고리즘 적용 전과 후의 작업기 실제 작동 위치점을 나타내었다. 알고리즘 적용 후 트랙터가 새머리 영역으로 진입할 때 RMS 위치 오차가 2.1 m에서 0.3 m로 감소하였으며, 새머리 영역에서 탈출 시 RMS 위치 오차가 2.2 m에서 0.17 m로 감소하는 것을 확인하였다. 따라서, 개발한 작업기 제어 알고리즘을 적용하면 포장에서 무인 농작업 진행 시 미경지 영역 및 동력 소모량이 감소하여 포장 작업효율을 향상이 가능함을 시뮬레이션에서 확인할 수 있었다.

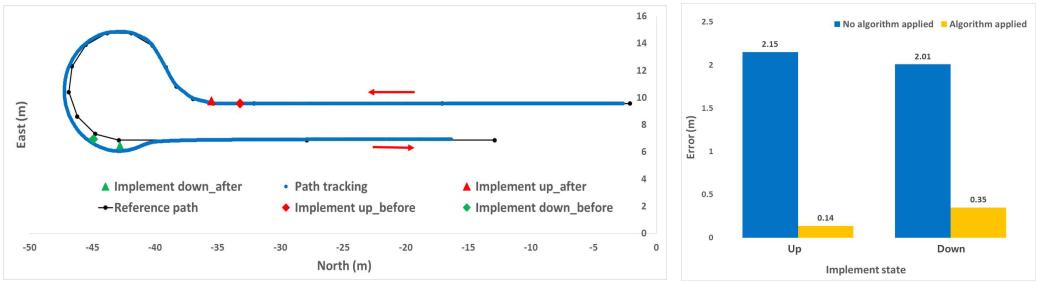


Fig. 3-45 Result of the simulation comparison of RMSE by no algorithm and the developed algorithm on the R-type pattern path

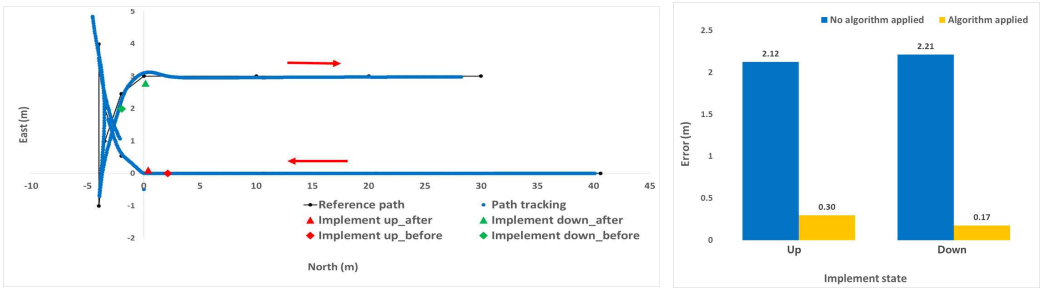


Fig. 3-46 Result of the simulation comparison of RMSE by no algorithm and the developed algorithm on the X-type pattern path

3.4. 요약 및 결론

실제 포장에서 자율주행 농작업을 수행하기 전에 개발한 경로생성과 경로추종 알고리즘의 유효성과 작업 특성을 분석하기 위해 컴퓨터 시뮬레이터를 개발하였다. 트랙터가 실제 포장에서 토양의 반력에 인해 회전 주행 시 슬립현상이 발생하기 때문에 이러한 현상을 시뮬레이션에서 반영하기 위해 EAKM을 사용하였다. Cf, Cr 값을 변경하면서 시뮬레이션에서 슬립의 효과를 부가가능하기 때문에 적용한 동적 설계된 주행 슬립 예측기반의 동적 경로추종 제어기를 시뮬레이터 상에서 주행 성능 평가를 진행하였다.

경사지 혹은 불규칙한 노면은 사용자가 비트맵 형태로 3차원 지도를 작성 후 시뮬레이터에서 3D 지형을 구성 가능하도록 하였다. 개발한 시뮬레이터에서 3D 환경지면이 구성 가능하기 때문에 GPS 신호보정 알고리즘을 적용하여 주행 성능평가를 진행하였으며 복잡한 지면환경에서 효율적으로 GPS 신호 보정 효과를 분석하였다.

개발한 시뮬레이터는 시간 기반의 포장 작업효율 분석이 가능하며 면적 기반의 작업면적 비율도 함께 사용자에게 제공한다. 개발한 자율주행 가상 시뮬레이터에서 작업기 제어 및 작업영역이 표시 가능하기 때문에 개발한 미경지 영역을 방지를 위한 작업기 제어 알고리즘을 평가하였다.

본 연구에서 개발한 시뮬레이터를 이용하여 주요 활용한 기술의 결과는 다음과 같다.

농작업 선회 유형 및 수학적 모델링 기반으로 개발한 경로를 가상 시뮬레이터 환경 통해 경로생성 성능 평가하였다. 한국의 표준 포장 구획 0.4 ha (100 m×40 m)에서 3가지 선회 패턴 경로를 기반으로 포장 작업시간, 유효 작업시간 및 포장 작업효율을 분석하였다. 포장 작업효율 기준으로 보았을 때 X형 선회 패턴 기반으로 생성된 경로는 포장 작업효율이 87.38%로 가장 높고 가장 적합한 것으로 나타났다.

개발한 동적 경로추종 알고리즘의 주행 성능을 평가하기 위해 기존 개발한 Look-ahead 경로추종 알고리즘과 비교분석을 진행하였다. 선회구역에서 왕복 작업구역으로 진입 시 동적 경로추종 알고리즘이 목표 경로와 더 근접하는 궤적을 나타났다. 동적 경로추종 알고리즘의 주행 RMS 위치 오차는 5.44 cm로 더 작게 나타났으며 skip 면적이 6.5 m^2 감소하는 것으로 나타났다.

트랙터가 새머리 영역으로 진입 및 새머리 영역에서 탈출 시 작업기 제어 성능 평가를 진행하였다. 제어 알고리즘 적용하면 새머리 영역으로 진입과 탈출 시 RMS 위치 오차가 모두 30 cm 이내로 감소하는 것을 확인할 수 있다. 개선된 알고리즘을 자율주행 트랙터에 적용하여 실제 농작업 시 미경지 영역 및 동력 소모량을 감소하여 포장 작업효율을 향상이 가능함이 판단된다.

개발한 GPS 위치 보정 알고리즘의 성능 평가하기 위해 가상 시뮬레이터에서 경사지면을 구현하여 트랙터를 기울어진 상태로 경로추종 특성을 분석하였다. 좌/우 15도의 경사지면에서 각각 목표 경로와 30 cm / 20 cm 이내의 오차로 보여 알고리즘 적용 전 보다 더 빨리 감소하여 안정된 주행 상태로 나타났다.

4. 자율주행 트랙터 통합 시스템 개발 및 성능평가

4.1. 재료 및 방법

4.1.1. 개요

본 연구에서는 실차 트랙터를 이용하여 개발한 경로생성과 추종 알고리즘을 탑재하여 포장성능시험을 수행하였다. 비 작업 상태에서 동적 경로추종 알고리즘의 기초 성능 평가, 경사지면에서 경로추종을 위한 GPS 신호 보정 평가, 미경지 영역을 방지를 위한 작업기 제어의 성능 평가 및 다변형태 포장에서 자율주행 작업 평가가 진행되었으며, 쟁기/로터리 작업 상태에 자율주행 요인 성능평가, 계측시스템을 이용 자율주행 쟁기작업의 성능 평가도 진행하였다. 또한, 개발한 다변형 포장 경로추종 알고리즘의 실제 작업 가능성을 실차 실험을 통하여 구명하였다.

4.1.2. 자율주행 트랙터 시스템 개발

무인 자율주행 트랙터 구축을 위하여 동양물산기업(주)에서 생산중인 25마력 TS25 트랙터를 대상으로 자율주행 기능을 탑재, 작업이 가능하도록 Layout 설계, 시스템 통합 설계 등을 수행하였다. 해당기종은 유압무단변속(HST)이 가능하며 대부분 기계식으로 작업 장치를 구동하고 있어 전자 작동 장치를 탑재하여 자동 작업이 가능하도록 관련 브라켓과 작동기를 장착하였다. Fig. 4-1은 트랙터 외형 사진을 나타낸 것이고 Table 4-1은 트랙터 TS25의 제원을 나타낸 것이다.

Fig. 4-1 A view of TS25 tractor used in the study

Table 4-1 Specifications of the TS25 tractor used in the study

Items		Specifications
Engine	Type	Four stroke, indirect injection water-cooled diesel
	Cylinders	3
	Compression Ration	22:1
	Displacement cubic	61.5 inch
	Fuel Tank Capacity	6.1 gallons
Electrical	Battery Capacity	12V 50AH
	Alternator	12V 50A
Transmission	Type	HST-2 ranges
	Numberf of Gears/Speeds	Infinite
Steering	Type	Hydrostatic Power
	Pump Output	2.6 gallons per minute

Hydraulic System	Type	Open center, full live hydraulic
	Pump Output	4.5 gallons per minute
	3-point Linkage	Category 1
	Lift Capacity	1320 lbs
PTO	Type	Independent
	PTO RPM	540 @ 2893 engine rpm
	PTO HP	19.3 kW (max)
Operating Weight	Total	1540 lbs
Dimensions	Overall Length	102.4 inch
	Overall Width	47.2 inch
Tire sizes	Ag (Front / Rear)	6x12 / 9.5x16
	Industrial (Front / Rear)	20x8-10 / 27x12.5-15
	Turf (Front / Rear)	20x8-10 / 29x12.5 -15

전기 또는 유압을 이용하여 전자제어가 가능하도록 변경하는 주요 부품은 크게, 조향장치, 가감속 페달, 작업기제어 레버 등으로 Fig. 4-2와 같다. 전자제어기와 센서 및 솔레노이드 밸브를 이용한 전자유압제어가 일반적으로 선호되고 트랙터 특성 상 유압을 많이 사용하고 있으므로 가장 편리하게 제어 시스템을 구축할 수 있지만 본 연구에서 TS25 모델 트랙터는 내부 유압 시스템 설계 변경의 어려움이 있어 전기식 모터를 이용하여 위치제어 방식을 채용하였다. 제어관련 부품들은 자율주행 제어를 위한 인터페이스 모듈에 연결되고 인터페이스 모듈은 자율주행 제어기로부터 제어명령을 받아 관련 제어하도록 시스템을 구성하였다. Fig. 4-3은 제어부품과 인터페이스 모듈간의 제어흐름도이다.

Fig. 4-2 System configuration of the tractor

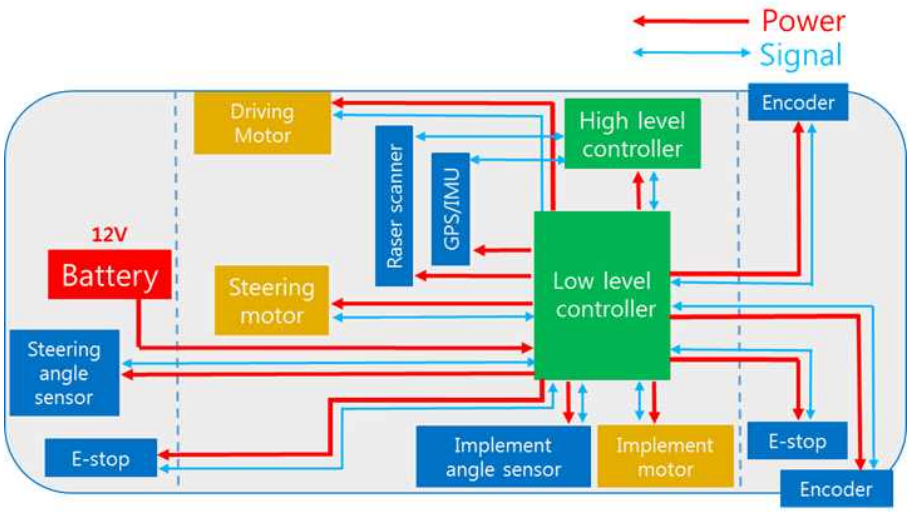


Fig. 4-3 Configuration of power and signal for autonomous tractor

트랙터의 주행모드는 사람이 직접 운전하는 수동 모드, 원격 조종기를 사용하여 수행하는 원격 모드와 자율주행을 수행하는 자동 모드로 나누어 제어하도록 구성하였다. 수동 모드에서는 모든 전기 구동부품이 토크를 풀어진 상태로 사용자가 기계식에 의하여 트랙터를 구동하게 되며 원격 모드인 경우 트랙터는 조종기의 제어명령을 받아 자율주행 low-level 제어기에 의하여 주행하는 방식이다. 자동 모드인 경우 트랙터는 자율주행 high-level 제어기에 제어명령을 받아 자율주행 low-level 제어기에 의하여 주행한다. Low-level과 high level 제어기간은 UDP 통신기반으로 연결하였다.

4.1.2.1. 조향 시스템 설계

트랙터가 무인 자율주행을 수행하기 위해서 high-level 제어기의 조향 명령을 받아 자동으로 조향을 수행하는 시스템을 장착하였다. 일반적으로 트랙터의 조향 시스템은 오비트롤방식을 사용하고 있으며 트랙터의 작업 특성상 논이나 밭에서 조향을 위해선 전륜조향을 위하여 큰 힘이 요구되므로 유압방식의 조향시스템을 채택하고 있다. 운전자가 운전대(steering wheel)를 조작하면 운전대와 연결된 조향펌프가 작동하고 있어 자율주행 중 제어기를 통한 조향을 위해서는 모터를 이용하여 조향펌프를 작동하였다. 트랙터의 전륜 실린더를 작동하기 위한 오비트롤의 사양은 Table 4-2와 같다.

Table 4-2 Specifications of the obitrol used in the study

Items	Specifications
Max. Pressure	17.1 Mpa
Rated Flow	16 L/min
Max. Back Pressure	3.4 Mpa
Input Torque	2.0 Nm
Max	17.5 Nm

트랙터가 자율주행 운전 중 스티어링 조작을 위하여 EPS (Electrical Power Steering)를 장착하였다. EPS(Unmanned Solution Co, Korea)는 모터구동부와 제어부로 구성되어 있으며 제어부는 자율주행 제어기로부터 제어명령을 받아 위치제어를 하는 구조이다. EPS 사양(Table 4-3)은 오비트롤 제원에 제시된 입력 토크를 만족시키면서 조향 휠의 위치 및 토크 제어가 가능한 사양의 제품을 선정하였으며 트랙터에 장착이 가능 하도록 마운팅 직경을 수정하였다. Fig. 4-4는 트랙터 조향 밸브 축에 EPS를 장착하기 전과 장착한 후의 그림이며 Fig. 4-5는 자율주행 트랙터에 장착된 EPS이다.

Table 4-3 Specifications of the EPS used in the study

Items	Specifications
Power	12V
Max. Drive Current	7A
Drive Torque	2.5 Nm
Reduction Ratio	1:15
Rotational Speed (No load)	150 RPM
Weight	3.05Kg
Operating Temperature	20ºC~80 ºC
Control Board	Baud rate : 500kbps Analog Input 2 channel 0Vdc, 5Vdc Output DIO 2 channel RS232 1 channel CAN 1 channel
Input of Steering Angle Sensor	Analog 0V: 100º / 2.5V : 0º

Fig. 4-4 Design of electrical power steering for autonomous tractor

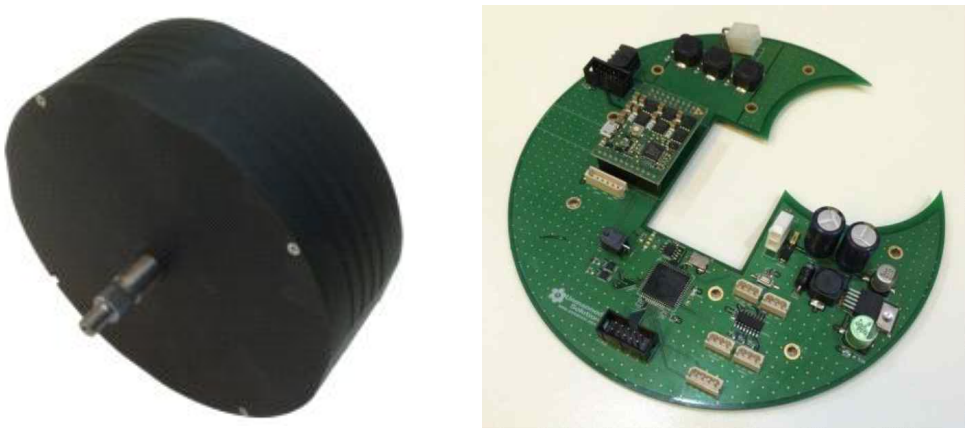


Fig. 4-5 EPS module and control board

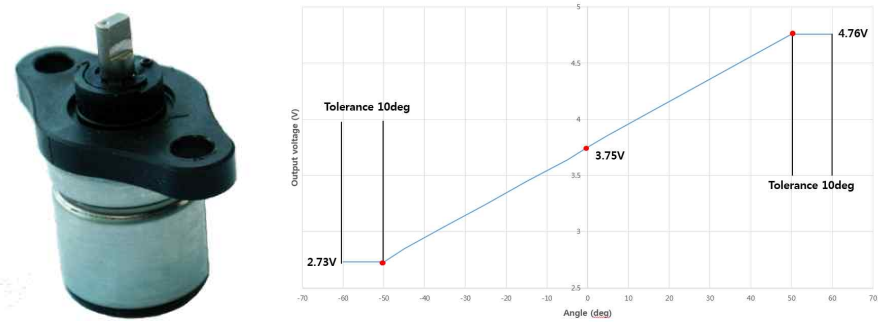
4.1.2.2. 조향각 센서

자동차에서 일반적으로 사용하는 랙-피니언 방식의 경우 조향각과 조향핸들의 위치가 일정하므로 모터를 사용한 제어가 용이한 반면, 트랙터는 오비트롤 방식으로 구동하여 조향핸들과 전륜 조향 각은 기하학적인 관계를 갖지 않으므로 조향 각 위치판별을 위해서는 별도의 각도 측정을 위한 센서를 사용하였다. 조향실린더 회전에 따라 센서의 변형이 수반되는 문제를 고려하여 조향각 측정 센서는 회전식 포텐쇼미터를 사용하며 전륜 휠의 조향 회전중심이 되는

킹핀축 중심에 설치하였다. Fig. 4-6 (a)은 사용한 회전식 포텐쇼미터이며 센서 사양은 Table 4-4와 같다. Fig. 4-6 (b)과 같이 각도와 전압 간의 관계가 선형으로 나타내고 양쪽 끝 10도에서는 출력전압의 dead band를 설정하며 이는 제어 시 조향각의 큰 변화에 따른 과도한 진동을 방지한다. Fig. 4-7은 조향 각 센서를 전륜에 장착할 수 있도록 설계된 도면이다.

Table 4-4 Specifications of the rotary potentiometer used in the study

Items	Specifications
Product Life	3M Cycle (Full Travel Cycles)
Seal Integrity	IP65
ESD	Tested in accordance with IEC 61000-2-2
EMS	As per ISO 11452-2 (2004E)
Input Voltage	5/12/24/48/80 VDC as requested
Steer Sensor Travel	360 deg
Operation Temp.	-40 ~ +85°C
Storage Temp.	-40 ~ +105°C
Salt fog Test	After Exposed to Salt Fog 96 hours



(a) (b)

Fig. 4-6 Encoder (a) and relationship between steering angle and output voltage (b)

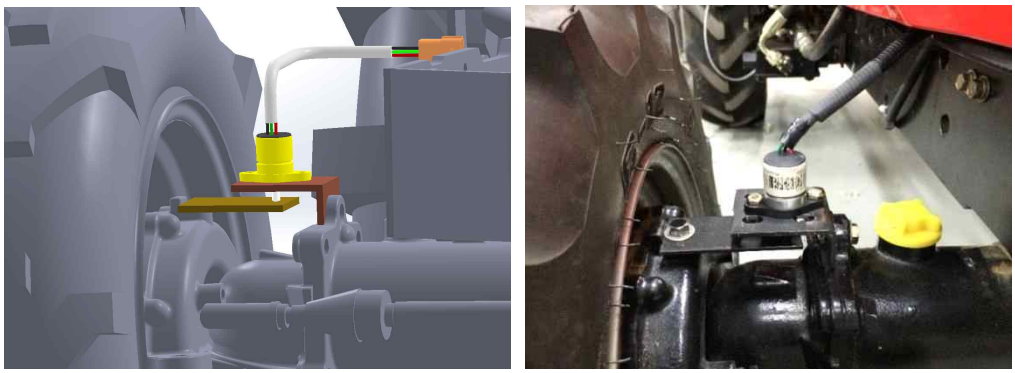


Fig. 4-7 Design of encoder for steering angle measurement

4.1.2.3. 가감속 제어 장치

트랙터의 가감속은 가속페달과 감속페달을 기계적으로 조작하여 이뤄지게 된다. 이러한 메커니즘을 전자제어가 가능하도록 하 기 위해 운전석 아래에 스마트모터를 장착하여 제어기의 명령에 따라 목표 위치로 회전하여 기구적으로 페달에 힘을 전달함으로서 가감속이 이루어지도록 하였다.스마트 모터가 HST에 속도 제어 축과 연결하여 회전하는 방식으로 전진과 후진의 속도를 제어한다. 제어 모터는 트랙터 후방에 설치된 인터페이스 모듈과 RS-232 방식의 통신을 통하여 컨트롤러로부터 모터의 제어명령 뿐 아니라 위치 초기화 및 비상정지 모드를 구현할 수 있도록 설계하였다. Fig. 4-8은 사용한 Animatics사의 스마트 모터이며 Table 4-5는 속도제어에 사용된 스마트 모터의 사양이다.

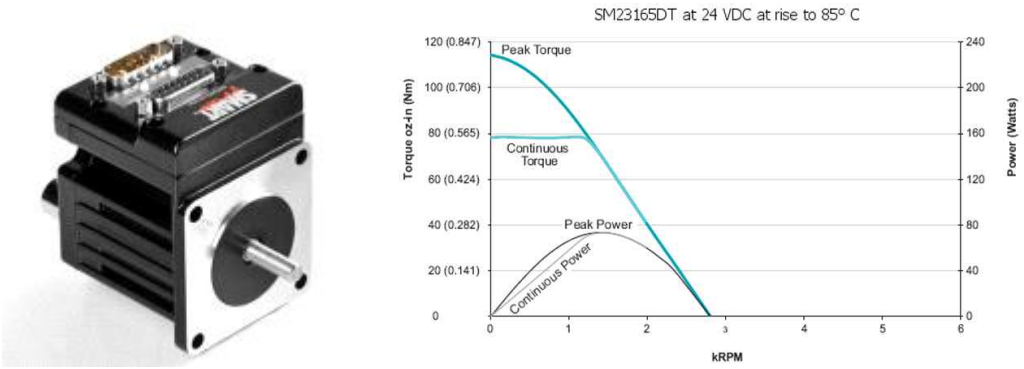


Fig. 4-8 Smart motor for driving speed control

Table 4-5 Specifications of the smart motor used in the study

Items	Specifications
Continuous Torque	0.52 N-m
Peak Torque	0.84 N-m
Nominal Continuous Power	204 Watt
No Load Speed	5,200 RPM
Continuous Current	6.1 Amps
Voltage Constant	9.08 V/kRPM
Winding Resistance	0.7 ohms
Encoder Resolution	4,000 Counts/Rev
Rotor Inertia	0.706 10-5 Kg-m2
Weight	0.59 kg
Shaft Diameter	6.35 mm
Shaft, Radial Load	3.18 kg
Shaft, Axial Thrust Load	1.36 kg

또한 HST의 속도제어 축과 가감속 페달간은 기계적으로 이루어졌기 때문에 스마트 모터를 이용 HST의 속도제어 축 회전을 위해 Fig. 4-9와 같이 ATG사의 PGX-62-H-100모델의 감속기를 사용하며 Table 4-6은 감속기의

사양이다. Fig. 4-10은 트랙터의 속도제어 모터를 장착하는 모습을 보여주는 3차원 캐드를 나타낸다.

Fig. 4-9 Motor reducer for driving speed control

Table 4-6 Specifications of the motor reducer used in the study

Items	Specifications
Rated Output Torque	40 Nm
Rated Input Speed	3,000 rpm
Max. Input Speed	6,000Nm
Torsional Rigidity	6 Nm/arc-min
Max. Radial Force	1120 N
Max. Axial Force	560N
Service Life	10,000 hr (Continuous Operation 4,000 hrs)
Efficiency	≧97%
Operating Temperature	-25°C ~ +90°C
Noise Level	≦60 dB
Weight ±3%	1.35 Kg

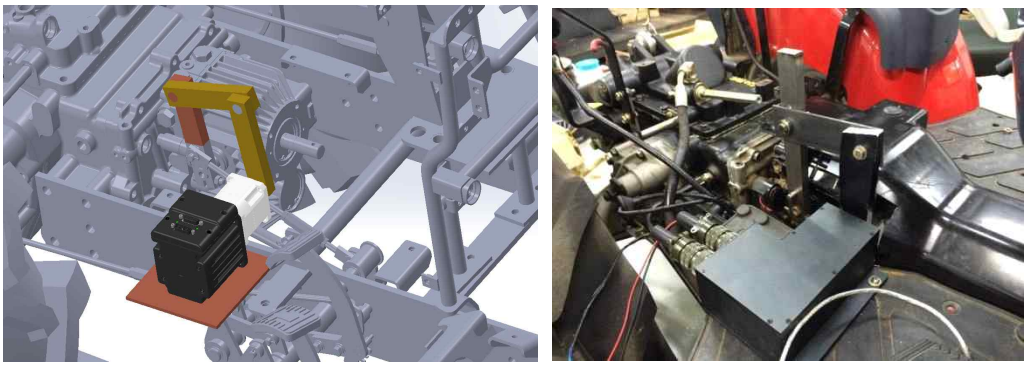


Fig. 4-10 Design of speed control actuator

4.1.2.4. 작업기 제어 장치

트랙터 작업기 제어는 트랙터의 변속기에 장착되어 있는 작업기 제어 레버를 통해 앞/뒤 움직이면 유압 벨브를 통해 작동하게 된다. 본 트랙터의 작업기 제어용 유압 벨브는 기계식으로 작동되기 때문에 모터를 장착하여 작업기 레버와 연결해 전기신호를 통하여 작업기를 제어하는 방식으로 설계변경하였다. 트랙터의 작업기 제어를 위해 적용된 모터는 Fig. 4-11과 같이 Robotics사의 MX-106 다이나믹셀 모터를 사용하였다.

Fig. 4-11 Dynamixel motor for implement control

다이나믹셀 모터는 감속기, 제어기, 구동기 및 Network 기능 등을 일체형으로 구성하여 하나의 모듈로 구성되어 있고 RS-485 방식의 통신을 통하여 컨트롤러와 연동하는 방식이다. Table 4-7은 다이나믹셀 모터의 사양이며 Fig. 4-12는 트랙터의 작업기 제어 모터를 장착할 수 있도록 설계된 도면이다.

Table 4-7 Specifications of the dynamixel motor used in the study

Items	Specifications
Model Name	MX-106R
Size	40.2mm × 65.1mm × 46mm
Gear Ratio	225:1
Operation Voltage	12 V
Stall Torque	8.4 Nm
Stall Current	5.2 A
No Load Speed	345 RPM
Range of Motion	360 Degree
Operating Temperature	-5°C ~ +80°C
Communication Protocol	RS485 Asynchronous Serial
ID	254 ID (0~253)
Communication Speed	8000bps~4.5Mbps
Position Sensor	Contactless absolute encoder

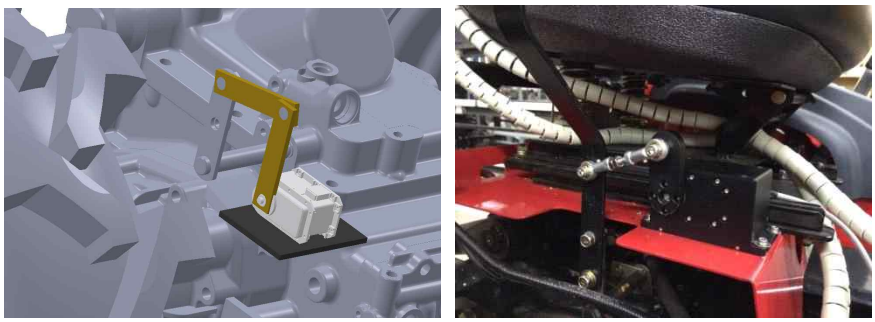


Fig. 4-12 Design of implement control actuator

4.1.2.5. 주행 속도 측정용 엔코더 센서

트랙터가 작업 주행 중 지면과의 슬립이 발생하여 이론주행 속도 및 거리와 실제 주행 속도 및 거리에 차이가 발생된다. 따라서 현재 제어하는 트랙터의 실제 주행 속도와 주행거리와 이론 주행거리 및 주행 속도를 알 경우 슬립에 대한 고려가 가능해지므로 트랙터의 정확한 주행 제어가 가능해 진다. 기계적인 원인에 의하여 순간적인 GPS 신호의 수신에 문제가 발생하는 경우에도 바퀴의 정확한 동작 상태를 측정함으로써 트랙터의 위치정보 및 속도정보를 제공 가능하도록 해주며, 안정적인 트랙터 무인제어를 가능하도록 구성한다. 이를 달성하기 위해 트랙터의 후륜 휠에 Autonics사의 E58CS10-CS 엔코더(Fig. 4-13)를 설치하여 제어 안정성을 높였다. Table 4-8은 DMI의 제원을 나타낸 것이며 Fig. 4-14은 트랙터의 주행 속도 측정을 가능하게 장착할 수 있도록 설계된 도면이다.

Fig. 4-13 Encoder for speed measurement

Table 4-8 Specifications of the encoder used in the study

Items	Specifications
Resolution (Pulse/Round)	100
Phase	T/4 ± T/8
NPN Control Output	Load Current < 30mA, Residual

	Voltage< 0.4VDC
Max. Response frequency	300kHz
Insulation Resistance	>100MΩ
Access Method	Wiring withdrawal
Operating Torque	< 0.002Nm
Moment of Inertia	<2×10-6kgm2
Capacity of Axis Load	<2kgf
Max. transmission	5000rpm
Operating Temperature	-10°C ~ +70°C
Weight	80g

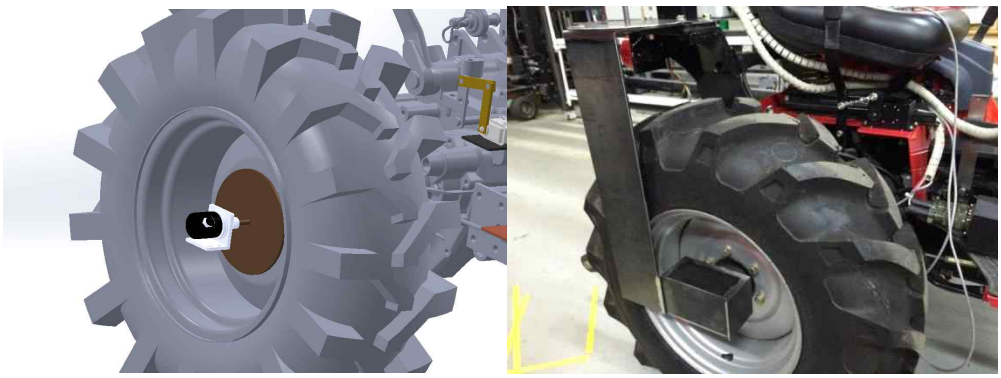


Fig. 4-14 Design of encoder for speed measurement

4.1.3. 항법 시스템 개발

4.1.3.1. VRS-GPS 시스템 구성

일반적으로 GPS는 4개 이상의 저궤도 위성과 신호 수신을 위한 안테나, 그 신호들을 처리하는 모듈로 구성된다. 사용하는 위성 수신 안테나와 수신 모듈은 수평 오차가 수십 미터 이상 큰 오차를 보여 자율주행 트랙터에 사용하기에는 어려움이 있어 본 연구에서는 정확도가 높은 VRS-GPS를 사용하여 정확한 위치정보를 수집하였다. VRS-GPS는 가상의 기준국을 이용 서버와 전

역 통신망을 통해 보정신호를 받아 위치 오차를 보정하는 방식으로 Fig. 4-15와 같이 시스템을 구성하였다.

VRS-GPS를 구성하기 위해 Android OS탑재된 스마트폰(Galaxy S5)과 GNSS receiver RT-2기능의 Firmware가 탐재된 GPS 수신기(Bluetooth 내장)를 사용하였다. Novatrip ver.1 App을 통해 시스템을 설정 후 국립지리정보원 VRS sever에 접속하여 사용하였다.

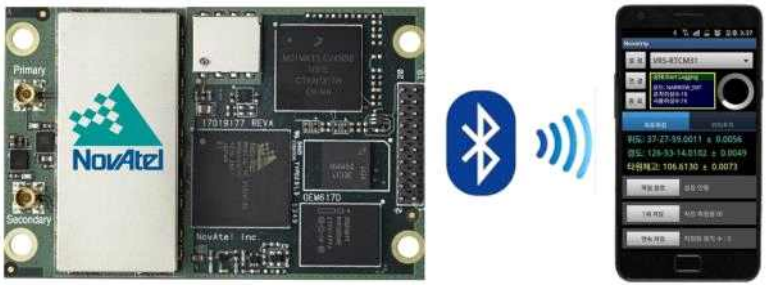


Fig. 4-15 Configuration of a VRS-GPS system

4.1.3.2. 하드웨어 구성

VRS-GPS는 42G1215A-XT-1-2-CERT 안테나와 OEM-617-D 수신기를 사용한다. 42G1215A-XT-1-2-CERT 안테나는 L1과 L2를 통해서 보정신호를 받는다. OEM-617-D 수신기는 L1, L2, WASS, DGPS, SBAS, BeiDou 등을 받을 수 있기 때문에 관측정의 좌표들은 최대 1~2 cm의 정밀도를 갖는다. 또한 최대 20 Hz의 갱신속도를 가지고 있기 때문에 실시간 처리속도와 cm급 정밀도와 True heading제공으로 자율주행 트랙터에 적용할 수 있는 성능이다.

Ellipse-E 수신기는 고성능 관성 항법시스템이며 roll과 pitch는 최대 0.2 deg, yaw는 0.5 deg의 정밀도를 갖는다. 외부 GPS에 연결하여 확장 칼만필터 기반으로 센서 퓨전 기술로 위치 추적이 가능하다. 최대 200 Hz의 갱신속도로 데이터 제공이 가능하며, 본 연구에서는 경사지면에서 경로추종을 위한 GPS

신호 보정에 사용하였다. 항법 센서 데이터 획득 및 알고리즘 처리를 위한 ultra-compact controller를 사용하였으며, 시스템 구성 및 장착 모습은 Fig. 4-16과 같다. 항법시스템의 자세한 제원은 Table 4-9와 같다.

Fig. 4-16 Navigation system(left) and mounting position(right)

Table 4-9 Hardware specification of navigation system

Items	Figures	Specifications
42G1215A-XT -1-2-CERT Antenna		Frequency: L1, L2 Performance: LNA Gain (dB) Tracking: L-Band, SBAS, QZSS Max. number of frequency: Dual Input voltage: 2.5 to 24 VDC Consumption: 35 mA VSWR:<2.0:1
OEM-617-D		-Position accuracy Single point L1: 1.5m CEP Single point L1/L2: 1.2m CEP SBAS: 60 cm DGPS: 40 cm RT2: 1 cm+1 ppm -Signal tracking GPS: L1, L2, L2C

		GLONASS: L1, L2 Galileo: E1, E5b BeiDou: B1, B2 -Data rate Measurements: 20Hz Position: 20Hz
Ellipse-E		-Accuracy Roll/Pitch:0.2 deg Yaw: <0.5 deg Velocity:0.1 m/s Heave period: up to 15 sec Heave accuracy: 10% -Interfaces Aiding sensors: GNSS, DMI, RTCM Output rate: 200 Hz Main serial: RS-232 Serial protocols: NMEA, ASCII
POC-200 ultra-compact controller		-System core Processor: Intel® Atom™ E3845 1.91 GHz quad-core Graphics: Integrated Intel® HD Memory: 8G -Panel I/O Interface Ethernet: Gigabit Ethernet ports Serial port: RS-232/422/485 USB: 3xUSB 3.0 , 1x USB 2.0 port DIO: 4-ch isolated DI 4-ch isolated DO
Parani-SD100 0 - SENA		Serial interface: speed > 921.6kbps Bluetooth Interface: working distance=100m Max. Tx power: +18dBm Receive sensitivity: -88dBm Antenna gain: Stub : +1dBi, Dipole : +3dBi, Dipole : +5dBi, Patch : +9dBi

4.1.4. 경로계획 및 추종 통합 제어기 개발

자율주행 트랙터 시스템을 구성하기 위해 Fig. 4-17과 같이 경로 생성과 경로 추종 알고리즘을 구축위한 High level 제어기과 트랙터에 장착된 각종 액추에이터와 센서를 통합하는 low level 제어기로 구성하였다. Low level 제어기는 임베디드 형태의 실시간 컨트롤러가 장착되었으며 리얼타임 컨트롤러는 National Instrument사의 CompactRIO (Fig. 4-18) 임베디드 컨트롤 및 디바이스를 수집하는 리얼타임 프로세서를 사용하여 FPGA(Field Programmable Gate Arrary) 및 I/O를 사용자가 재구성이 가능하도록 하였다. CompactRIO 컨트롤러의 역할은 자율주행 제어기의 명령을 받아 트랙터 내부에 설치된 각종 액추에이터를 제어하고 센서의 데이터를 수집하는 역할을 한다. Table 4-10은 low level 제어기에 사용한 CompacRIO의 제원이다.

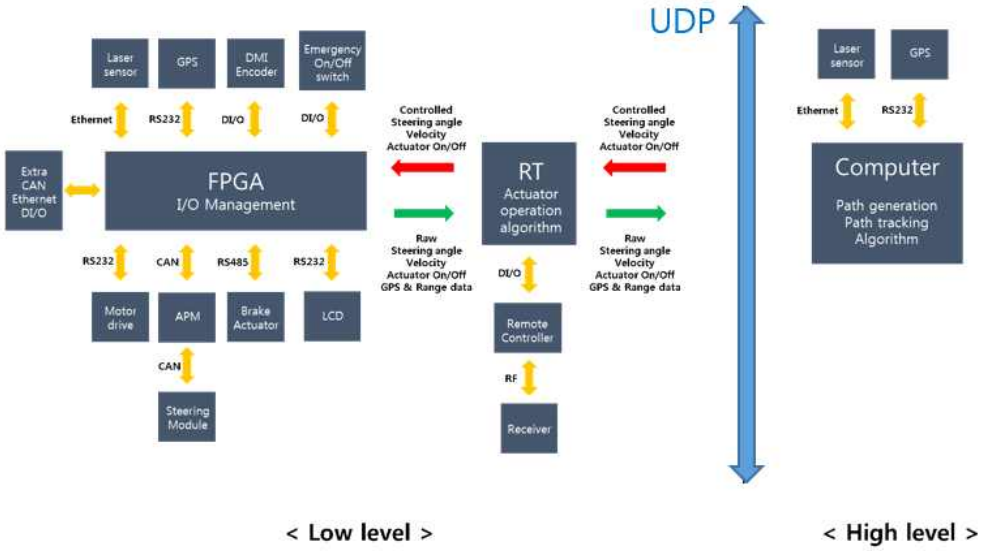


Fig. 4-17 Schematic diagram of auto guidance system

Fig. 4-18 Embedded controller for the low level control

Table 4-10 Specifications of the CompactRIO used in the study

Items	Specifications
Processor Type	Intel Atom Processor E3825
Frequency of CPU Clock	1.33GHz
Number of Cores	Dueal Core
Operation System	Linux
Novolatile Memory	4 GB
System Memory	1 GB
Number of Ethernet Port	2
Ethernet	10 BastT/100 BaseTX/1000 BaseT
Serial Ports (RS232)	1
Serial Ports (RS485)	1
Range of Input Power	9V ~ 30V
FPGA	Kintex-7

트랙터의 움직임을 제어하기 위해 사용하는 중요 요소는 조향제어, 가감속 페달 제어 및 작업기 제어이다. CompacRIO에 FPGA 기반의 확장 슬롯을 이용하여 Fig. 4-19와 같이 Digital I/O, Analog Input, RS-232, CAN등 다양한 interface를 이용 자율주행 시스템을 구성하였다. 트랙터 조향제어를 위해 NI

9862 CAN 모듈을 사용하여 EPS의 CAN frame을 참조하여 메시지 형식을 NI-XNET database editor에서 구성하여 EPS를 구동하였다. 조향핸들을 정확한 목표치에 도달하기 위해 EPS에 feedback 신호를 받아 회전식 포텐쇼미터를 NI 9239 Analog 모듈에 연결하여 타이어의 실제 조향각을 측정하였다.

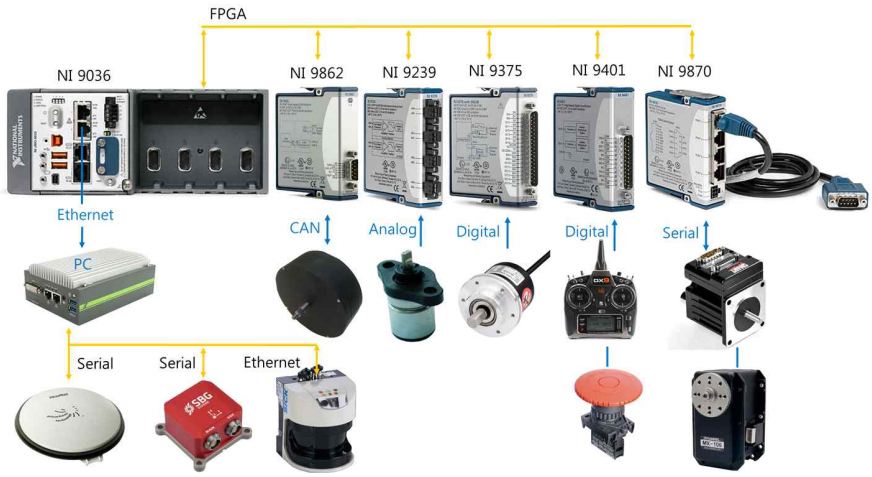


Fig. 4-19 Configuration of the auto guidance system

통합 제어 시스템에서 주행경로를 계산해서 트랙터가 목적지로 주행위해 조향 각을 생성하고 이를 유지하기 위해 Fig. 4-20과 같이 제어 방법을 적용하였다. 안전사항으로 트랙터의 조향각의 변화량 제한을 두어 주행 중에 급격히 조향 각이 변화하여 트랙터가 불안해 지는 것을 방지하도록 하였다.

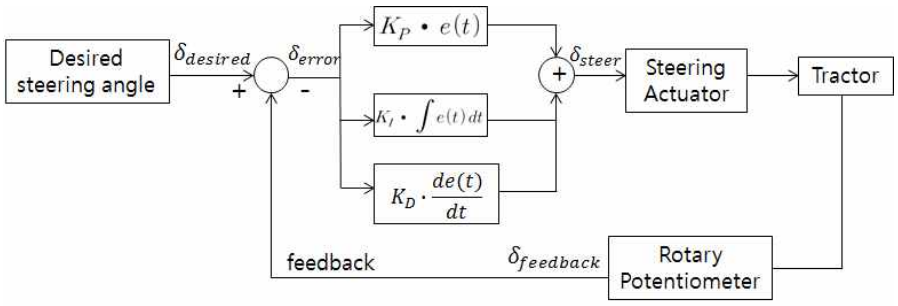


Fig. 4-20 Block diagram of the steering controller

트랙터의 주행 목적 속도가 전달되면 제어기는 트랙터에 장착된 엔코더를 이용하여 NI 9375 Digital 모듈을 통해 측정된 현재속도와 비교하여 속도차이를 계산하게 된다. 계산된 속도 차이값을 NI 9870 RS232 모듈을 이용해 트랙터의 속도제어 모터로 전달한다. 요구하는 트랙터의 속도를 유지하기 위한 최적의 제어 값을 계산하고 계산된 값은 각 구동기로 전달되어 차량의 엔진회전속도를 변경하였다. 또한 트랙터가 포장에서 주행 시 직진에서 선회 구간으로 진입과 선회에서 직진 구간으로 출입 시 작업기를 상승과 하강 작동을 하게 된다. High level에서 작업기의 작동상태를 UDP 통신을 이용 low level로 전달하게 되며 이러한 작동신호를 NI 9870 R3232 모듈을 이용해 트랙터의 작업기 제어 모터로 전달하였다. Fig. 4-21은 속도 제어와 작업기 제어 방법을 제시한 것이다.

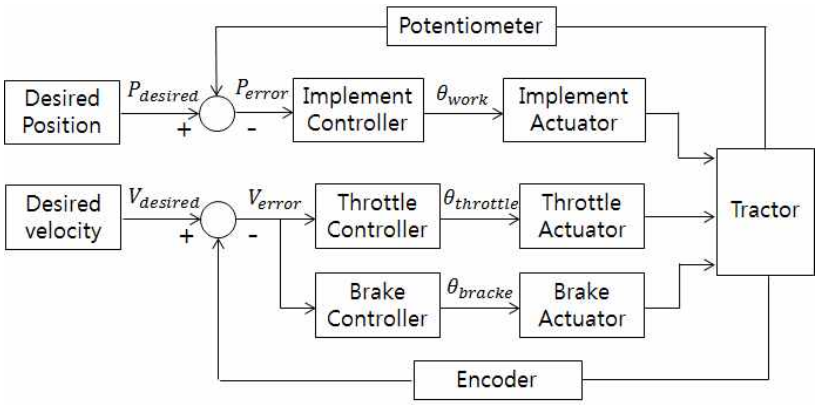


Fig. 4-21 Block diagram of the velocity & implement controller

무인자율주행 트랙터가 주행 중 위험 판단을 위해 트랙터의 긴급정지와 시스템의 정지를 위하여 원격 제어기와 레이저 스캐너를 이용하여 Emergency Stop시스템을 구성하였다. 트랙터제어시스템에 Horizon Hobby사에서 판매하는 DX9 리모트 컨트롤러, NI 9041 TTL digital 모듈을 이용하여 E-stop기능을 구현하였다. Fig. 4-22는 TS25 트랙터에 탑재한 자율주행을 위한 통합제어

기의 실제 모습이다. Fig. 4-23은 High level 제어기에 사용한 개발된 경로추종 프로그램이며 Fig. 4-24는 Low level 제어기에 각종 엑츄에이터를 구동하기 위한 프로그램이다.

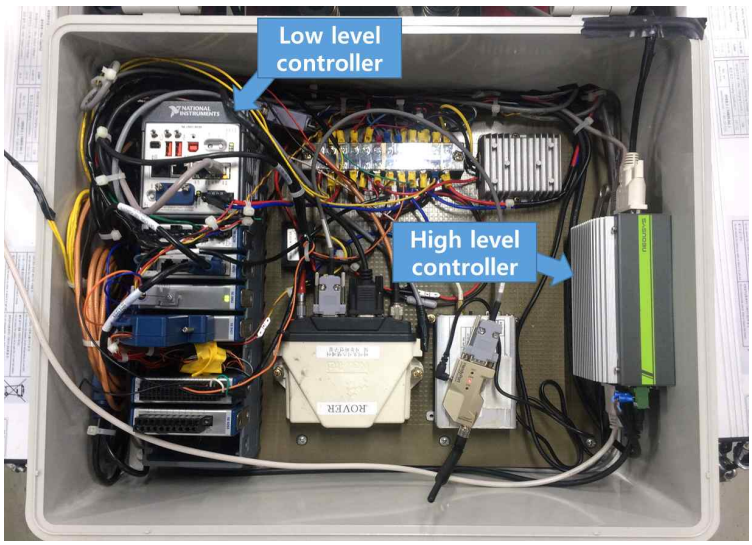


Fig. 4-22 Integrated controller for autonomous tractor

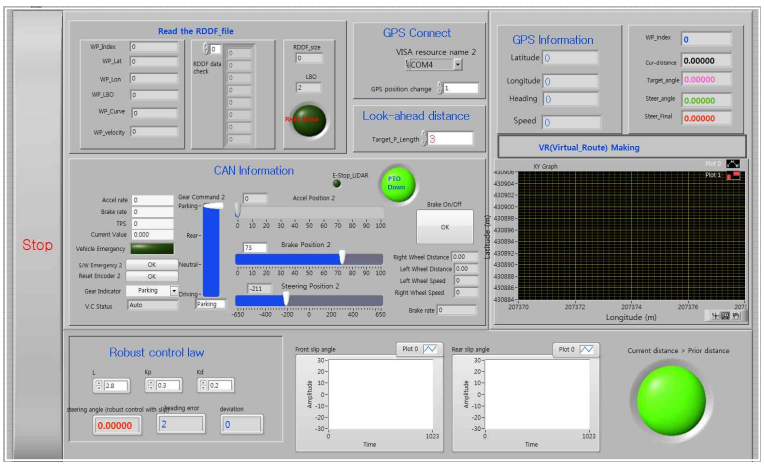


Fig. 4-23 Front panel of the high level control by LabVIEW

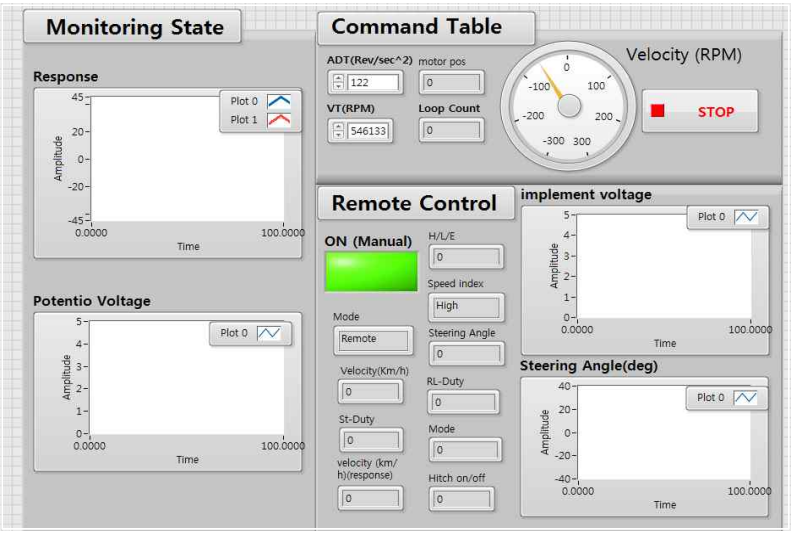


Fig. 4-24 Front panel of the low level control by LabVIEW

4.1.5. 경운성능 시험 평가용 자율주행 트랙터 시스템 구성

개발한 25마력 자율주행 트랙터외에 개발한 경로생성과 추종알고리즘을 이용하여 경운작업의 정량적 성능평가를 위해 선행 연구에서 개발한 83마력급 자율주행 트랙터에 쟁기를 장착하고 관련 계측 시스템을 장착하였다. 사용 트랙터는 동양물산기업(주)에서 생산중인 83마력 TX803으로 RTK-GPS와 고가의 IMU가 포함된 자율주행시스템이다. 또한, 시험기대는 전자유압 장치에 의하여 자동 작업이 가능한 구조이다. Fig. 4-25는 외형 사진을 나타낸 것이고 Table 4-11은 트랙터 TX803의 제원을 나타낸 것이다. 조향장치의 작동은 전륜의 회전 각도가 30도 이상 되면 작업기가 자동 상승하며 30도 이하 되면 자동 하강하도록 구성되어 있으며 작업기 상/하강에 따른 지연시간을 고려한 제어 알고리즘은 탑재되지 않은 시스템이다.

Fig. 4-25 A view of TX803 tractor used in the study

Table 4-11 Specifications of the TX803 tractor used in the study

Items		Specifications	Items		Specifications
Engine	Rated power	58.8kW/ 2400rpm	Body	Tire(front)	11.2-24
	Displacement	3.3 m		Tire(rear)	16.9-30
Tractor	Length	3.9 m		Steer type	Hydraulic
	Width	1.9 m		Fuel tank	110L
	Height	2.7 m	Speed	Forward	0.25~35.6 km/h
	Wheelbase	2.2 m		Reverse	0.24~34.7 km/h

시험 자율주행 트랙터 시스템 구성은 Fig. 4-26과 같으며 자세한 시스템 사양은 한웅철 석사학위논문(2013)에서 기술되어 있다.

Fig. 4-26 Auto-guidance system configuration of the tractor

4.1.6. 시험 평가를 위한 계측 시스템 구성

4.1.6.1. 쟁기작업의 견인력 측정

자율주행 트랙터가 쟁기작업 시 전체 포장의 작업특성을 제시하기 위해 수평 견인력을 측정하고자 하였다. 이를 위해 쟁기와 3점 히치 사이에 수평 견인력을 측정할 수 있는 장비를 구성하였다. 시험 TX803 트랙터에 Fig. 4-27과 같이 로드셀 기반의 육분력 측정 시스템을 장착하였다. 사용한 육분력 측정 시스템은 6개의 로드셀을 이용하여 3점 히치 점에 위치한 3개의 로드셀은 수평 축 방향의 힘에만 반응하며, 수평축과 수직을 이루는 평면에 삼각형의 형태로 자리한 3개의 로드셀은 수평 분력 이외의 힘에 민감하게 반응을 하는 구조이다(이진웅, 2015). 본 연구에서는 수평 방향의 3개의 로드셀에 작용하는 견인력의 합을 이용하여 수평 견인력으로 산출하였다. Table 4-12은 육분력

시스템에 사용된 로드셀의 제원이며 Fig. 4-28은 트랙터에 장착된 Dacell사 로드셀의 모습을 보여주는 그림이다.

Fig. 4-27 3-point hitch dynamometer for draft force measurement

Table 4-12 Specifications of draft force load cell

Items	Specifications
Manufacturer	Dacell, Korea
Rated capacity (kgf)	2000
Rated output (mV/V)	2.0±1%
Temperature range, safe	-20~80℃
Excitation recommended	10V

Fig. 4-28 Load cell for draft force measurement

4.1.6.2. 연료 소모량 측정

연료 소모량은 2개의 펄스 출력 유량계를 이용하여 엔진으로 유입되는 유량과 연료탱크로 리턴 되는 유량을 측정하여 그 차이로서 결정하였다. Table 4-13은 본 연구에서 사용한 유량계의 제원이고, Fig. 4-29은 트랙터의 엔진룸 내부에 설치한 Wintech Process사의 유량계를 나타낸 것이며 Fig. 4-30은 연료 소모량을 계측하는 원리를 나타낸 개략도이다. 한 개의 유량계는 연료펌프와 필터 사이에 설치하여 엔진으로 유입되는 유량을 측정하였으며, 나머지 하나는 엔진으로 부터 탱크로 리턴 되는 곳에 설치하였다(동양물산기업, 2016).

Table 4-13 Specifications of flow meter

Items	Specifications
Manufacturer	Wintech Process, Korea
Model	OG2-SS5-VHQ-B
Flow range	0.03 - 4.0LPM
Accuracy	± 0.75% FSD
Reliability	± 0.1%

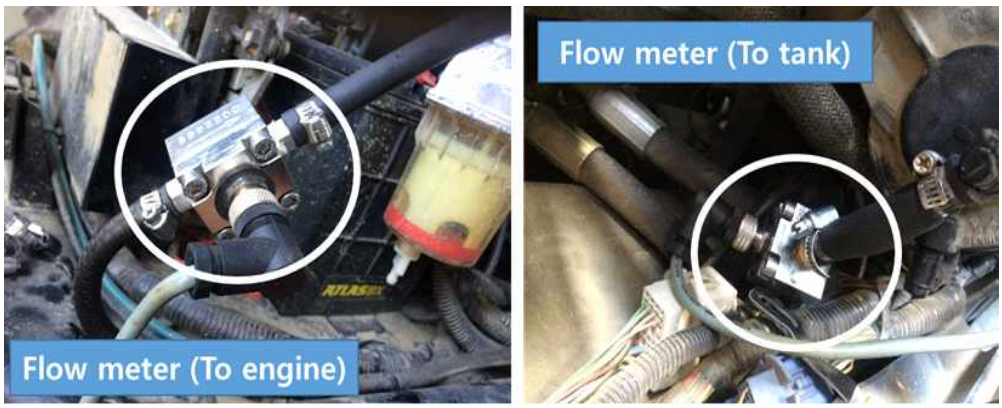


Fig. 4-29 Flow meters installed in TX803 tractor

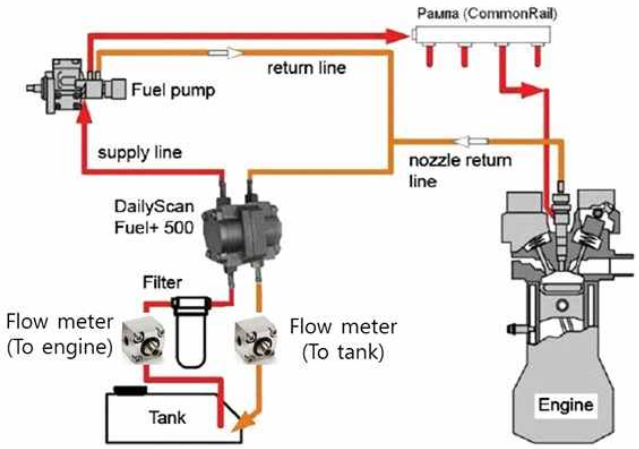
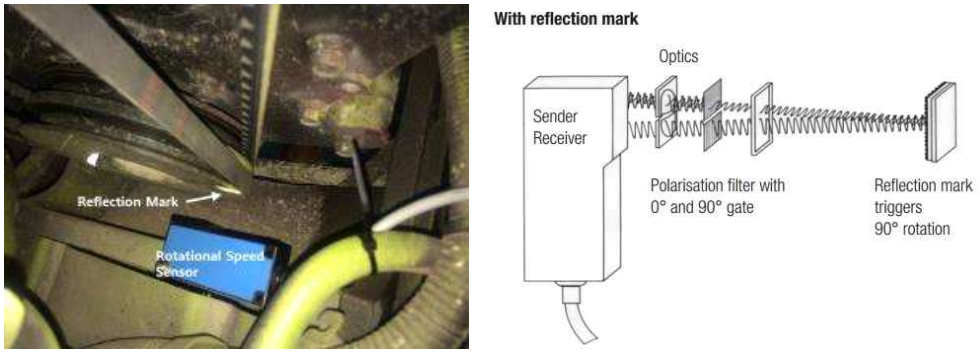


Fig. 4-30 The principle of fuel consumption measuring system

4.1.6.3. 엔진 RPM 측정

엔진 RPM을 측정하기 위해 Fig. 4-31 (a)과 같이 펄스 출력인 회전속도 센서를 사용하였다. 반사 마크를 엔진 출력축판에 장착해 회전속도 센서가 반사 마크를 인식하여 회전수를 측정하는 원리를 이용하였다. Fig. 4-31 (b)은 회전속도 센서를 사용하여 엔진 RPM 계측하는 원리를 나타낸 개략도이며 Table 4-14는 본 연구에서 사용한 엔진 RPM 측정 장치의 제원이다.



(a) (b)

Fig. 4-31 Rotational speed sensor installed in TX803 tractor(a) and the principle of engine RPM measuring system (b)

Table 4-14 Specifications of rotational speed sensor

Items	Specifications
Manufacturer	Hydrotechnik, German
Model	DS03
Light source	Power LED with visible red light
Range	50mm to 500mm
Signal repetition frequency	Max. 500Hz
Output signal	Analogue signal 4 to 20 mA

4.1.6.4. 계측 데어터 수집 장치

엔지 RPM, 견인력, 유량 소모량을 측정하기 위해 센서 및 데이터 수집이 가능한 계측 시스템을 구성하였다. 데이터 수집 장치는 Fig. 4-32과 같이 NI(National Instrument) 사의 c-DAQ 9174를 사용하였으며 3가지 센서가 모두 아날로그 신호를 출력하기 때문에 NI 9201 모듈을 이용 개발하였다.

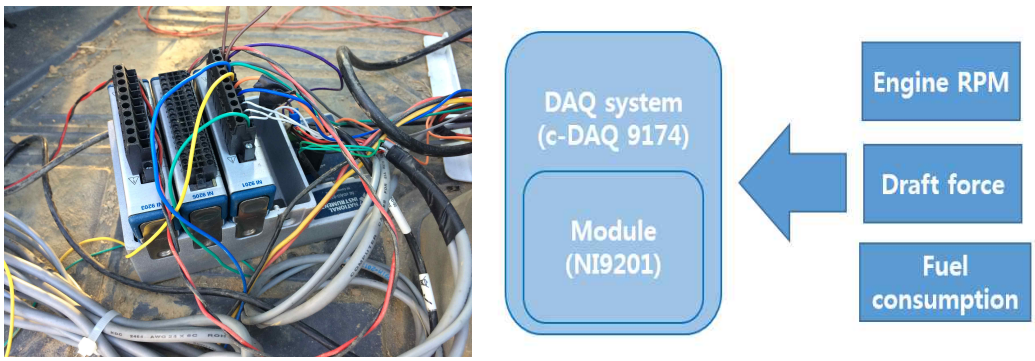


Fig. 4-32 Data acquisition system in the measuring tractor

4.2. 결과 및 고찰

4.2.1. 동적 경로추종 알고리즘 성능 요인실험 결과

개발한 동적 경로추종 알고리즘을 실차에서 성능 평가하기 위해 Fig. 4-33 (a)과 같이 TS25 모델 자율주행 트랙터 시스템에 탑재하여 비 작업 상태로 실시하였다. 성능실험 실시 장소는 서울대학교 농업생명과학대의 수원에 위치한 밭 포장에서 시행하였다. 시험포장의 크기는 길이가 70 m, 폭이 30 m이며 Fig. 4-33 (b)에 포장의 위성사진을 나타내었다.

개발한 강인 제어 기반의 동적 경로추종 알고리즘의 주행 성능을 평가하기 위해 기존연구의 TX803 모델 자율주행 트랙터에 적용된 Look-ahead 경로추종 알고리즘과 비교분석을 진행하였다. 주행 경로는 C형과 R형 선회 패턴 기반으로 왕복작업 경로를 생성하였다.

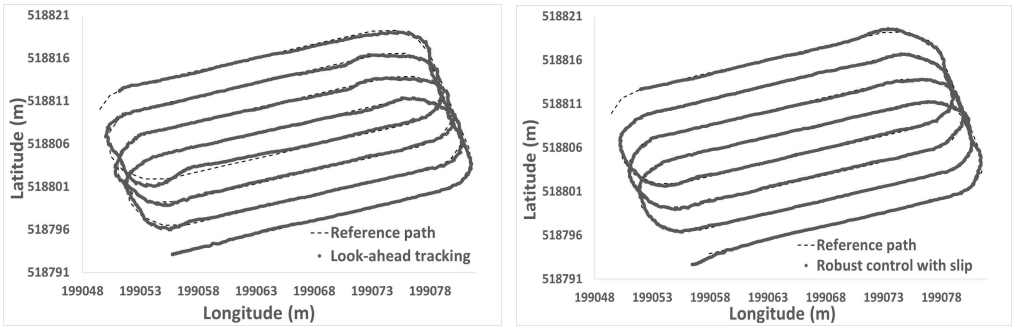
트랙터가 1.38 m/s의 속도로 주행하였으며 Fig. 4-34, Fig. 4-36과 같이 C형과 R형 선회 패턴 경로에서 Look-ahead과 동적 경로추종 알고리즘이 모두 주행이 가능한 것이 나타내고 있으나 직진 주행 후 선회하여 다시 직진 경로로 진입하는 구간에서는 강인 제어 기반의 동적 경로 추종 알고리즘이 목표 경로와 더 근접하며 오실레이션이 없는 주행 궤적을 나타났다.

성능 평가에서 전체 주행의 RMS 위치 오차는 Fig. 4-35 (a)에 나타낸 것과 같이 C형 선회 패턴 경로에서 동적 경로추종 알고리즘은 10.64 cm를 나타내었고 이것이 Look-ahead 추종기반의 RMS 위치 오차 17.69 cm 보다 작게 나타낸 것으로 판단되었다. Fig. 4-37 (a)에 나타낸 것과 같이 R형 선회 패턴 경로에서 동적 경로추종 알고리즘은 9.32 cm를 나타내었고 이것이 Look-ahead 추종기반의 RMS 위치 오차 18.30 cm 보다 작게 나타낸 것으로 판단되었다.

트랙터의 주행 안정성은 Fig. 4-35 (b)와 Fig. 4-37 (b)에 나타낸 것과 같이 C형과 R형 선회 패턴 경로에서 동적 경로추종은 Look-ahead 추종 알고리즘보다 작은 yaw rate를 유지하여 더 안정적으로 주행하는 것을 나타났다.

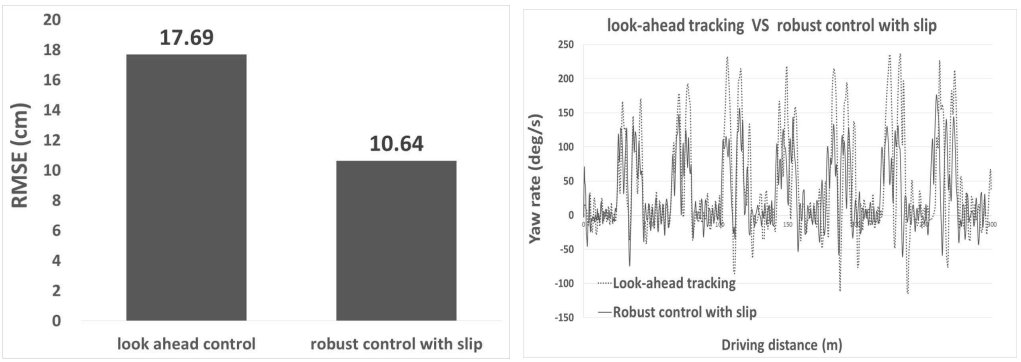
(a) (b)

Fig. 4-33 Evaluation test on the test field by TS25 auto-guidance tractor (a) and test field in Suwon from artificial satellite (b)



(a) (b)

Fig. 4-34 Result of full path tracking trajectory by look-ahead (a) and robust control (b) on the C-type pattern path (TS25 tractor)



(a) (b)

Fig. 4-35 Result of the position RMSE (a) & yaw rate (b) by look-ahead and robust control on the C-type pattern path (TS25 tractor)

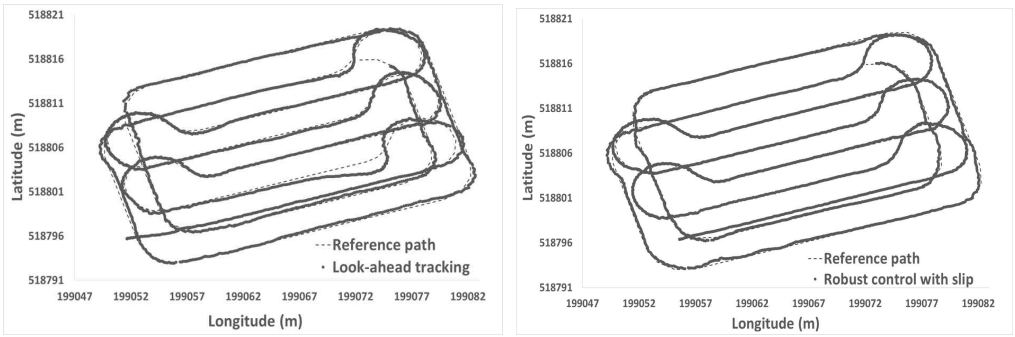
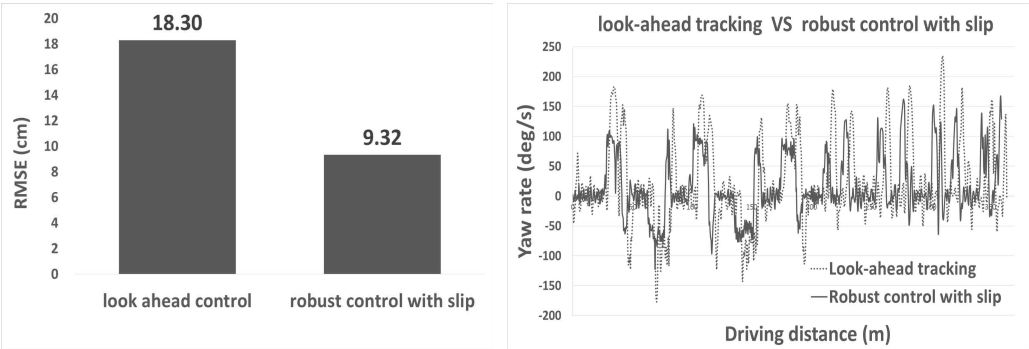


Fig. 4-36 Result of full path tracking trajectory by look-ahead (a) and robust control (b) on the R-type pattern path (TS25 tractor)



(a) (b)

Fig. 4-37 Result of the position RMSE (a) & yaw rate (b) by look-ahead and robust control on the R-type pattern path (TS25 tractor)

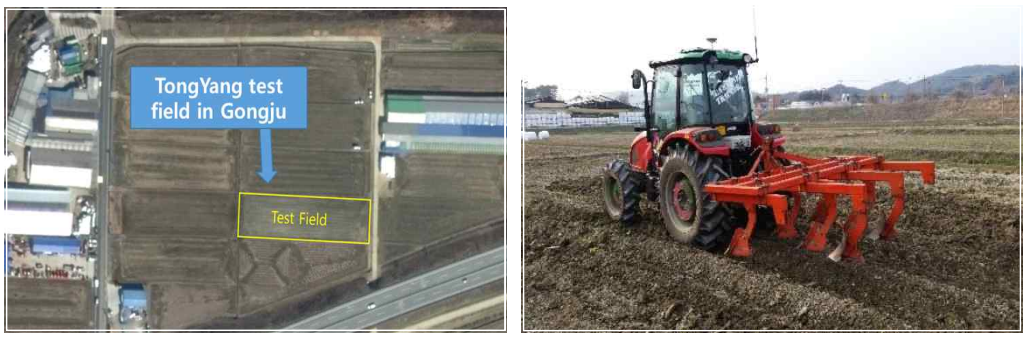
4.2.2. 쟁기/로터리 자율 농작업 요인 시험 결과

자율주행 트랙터(TX803 모델) 시스템에 탑재하여 Look-ahead 경로추종 알고리즘과 비교 실험을 진행하였다. 성능실험 실시 장소는 동양물산 중앙연구소의 공주에 위치한 실험 포장에서 진행하였다. Fig. 4-38 (a)에 포장의 위성사진을 나타내었으며 Fig. 4-38 (b)은 포장에서 쟁기작업 상태로 실험하는 모습이다. 성능 평가는 C형 선회 패턴 경로를 사용하였으며 1.12 m/s의 주행 속도로 유지하였다. Fig. 4-39는 쟁기작업을 사용 취득한 주행 궤적이다. 주행 모드는 수동(회색), Look-ahead 경로추종(노란색) 및 강인 제어 기반의 동적 경로추종(청색) 알고리즘을 이용하여 C형 선회 패턴 경로의 일부분 목표 경로(검정색 점선)를 사용하여 경로추종 성능을 비교하였다. 트랙터가 3가지 주행 모드를 사용한 주행 궤적을 나타내었다. 그림에 나타난 바와 같이 트랙터가 직진 주행 후 선회하여 다시 직진 경로에 진입하는 구간에서 운전자의 판단으로 선회 후 직진 경로 진입의 판단 여부가 난해하여 똑바른 주행이 어려운 것으로 나타났다. 반면에 강인 제어 기반의 동적 경로 추종은 수동 모드와

Look-ahead 추종 알고리즘 보다 모두 목표 경로와 더 근접하는 주행 궤적을 나타내었다.

Fig. 4-40에 나타낸 것과 같이 쟁기작업 기반으로 자율주행 시 Look-ahead 추종 알고리즘을 사용하면 직진 구간에서 약 20 cm의 주행 오차를 유지하며 선회 구간에서 약 1.4 m의 최대 오차를 보였다. 그러나 동적 경로추종 알고리즘을 사용하면 진진 구간에서 10 cm 이내의 주행 오차를 유지하며 선회에서 직진 구간으로 진입 시 20 cm 이내의 오차로 주행이 가능하였다.

Fig. 4-41에 나타낸 것과 같이 선회 구간에서 동적 경로추종은 매뉴얼 주행과 Look-ahead 추종 알고리즘들 보다 작은 yaw rate를 가지고 주행하는 것을 나타냈으며 특히 트랙터가 선회에서 직진 구역으로 진입 시 yaw rate는 청색과 같이 다른 주행 모드 보다 더 빨리 감소하여 안정된 주행상태로 유지하는 것으로 나타났다. 성능 평가결과 전체 주행의 RMS 위치 오차는 Table 4-15에 나타낸 것과 같이 운전자가 직접 매뉴얼로 운전 시 48.54 cm 비교적 큰 오차를 나타났다. 반면에 동적 경로추종 알고리즘은 9.5 cm를 나타내었으며 이는 Look-ahead 추종기반의 RMS 오차 22.27 cm 보다도 작은 오차였다.



(a) (b)

Fig. 4-38 Test field in Gongju from artificial satellite (a) evaluation test on the test field with plowing operation by TX803 tractor (b)

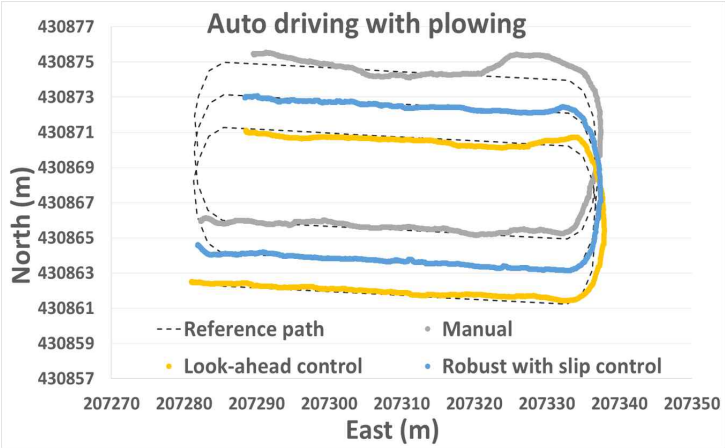


Fig. 4-39 Result of full path tracking trajectory by manual, look-ahead and robust control with plowing operation (TX803 tractor)

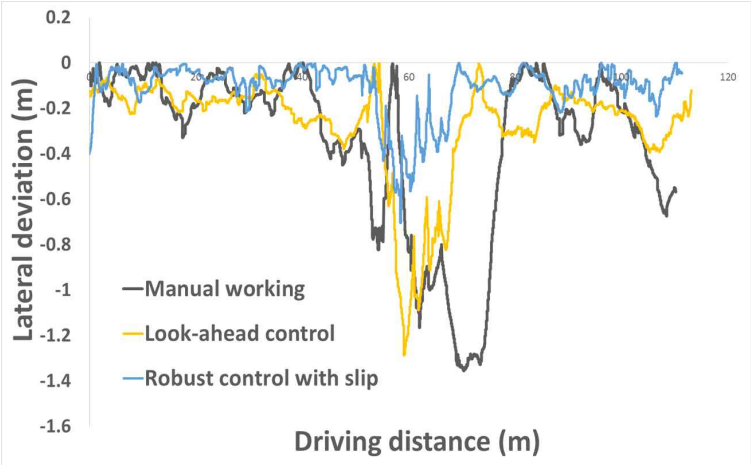
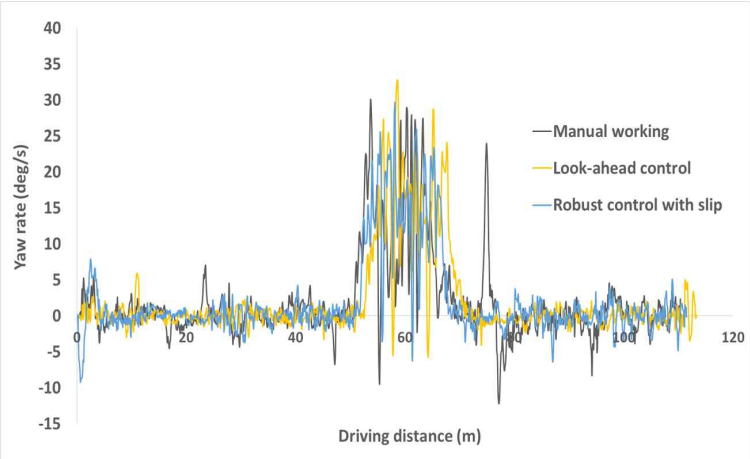
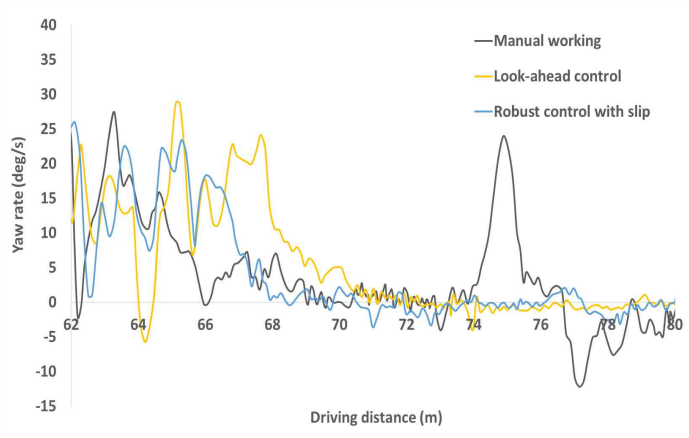


Fig. 4-40 Result of the lateral deviation by manual, look-ahead and robust control with plowing operation (TX803 tractor)



(a)



(b)

Fig. 4-41 Result of the yaw rate(a) and the enlarged part of curved to straight (b) by manual, look-ahead and robust control with plowing operation (TX803 tractor)

Table 4-15 Results of path tracking RMSE by manual, look-ahead and robust control with plowing operation (TX803 tractor)

Working types	RMSE
Manual with plowing	48.5 cm
Look-ahead control with plowing	22.2 cm
Robust control with plowing	9.5 cm

Fig. 4-42는 실험 포장에서 로터리 작업 상태로 실험하는 모습이다. 성능 평가하기 위해 포장에서 C형 선회 패턴 경로를 생성하였다. Fig. 4-43은 쟁기작업을 이용하여 취득한 주행 궤적이다. 주행 모드는 매뉴얼(회색), Look-ahead 경로추종(노란색) 및 강인 제어 기반의 동적 경로추종(청색) 알고리즘을 이용하여 C형 선회 패턴 경로의 일부분 목표 경로(검정색 점선)를 사용하여 경로추종 성능을 비교하였다. 트랙터가 1.12 m/s의 속도로 주행하였으며 3가지 주행 모드를 사용한 주행 궤적을 나타내었다. 로터리 작업의 주행 궤적은 앞서 설명한 쟁기 작업과 비슷한 결과를 나타내고 있다. 강인 제어 기반의 동적 경로 추종을 사용할 때 매뉴얼 모드와 Look-ahead 추종 알고리즘 보다 모두 목표 경로와 더 근접하는 주행 궤적을 나타내었다.

Fig. 4-44에 나타낸 것과 같이 로터리 작업 기반으로 자율주행 시 Look-ahead 추종 알고리즘을 사용하면 직진 구간에서 약 20 cm의 주행 오차를 유지하며 선회 구간에서 약 1.5 m의 최대 오차를 보였다. 그러나 동적 경로추종 알고리즘을 사용하면 진진 구간에서 10 cm 이내의 주행 오차를 유지하며 선회에서 직진 구간으로 진입 시 20 cm 이내의 오차로 주행이 가능하였다. 또한 Fig. 4-45에 나타낸 것과 같이 선회 구간에서 동적 경로추종은 매뉴얼 주행과 Look-ahead 추종 알고리즘들 보다 작은 yaw rate를 가지고 주행하는 것을 나타냈으며 특히 트랙터가 선회에서 직진 구역으로 진입 시 yaw

rate는 청색과 같이 다른 주행 모드 보다 더 빨리 감소하여 안정된 주행상태를 유지하였다.

성능 평가에서 전체 주행의 RMS 위치 오차는 Table 4-16에 나타낸 것과 같이 운전자가 직접 매뉴얼로 주행 시 36.26 cm의 큰 오차를 나타내었다. 그러나 동적 경로추종 알고리즘은 9.41 cm를 나타내었고 이것이 Look-ahead 추종기반의 RMS 위치 오차 23.12 cm 보다 작게 나타낸 것으로 우수한 성능으로 판단하였다. 본 요인 실험을 통해 동적경로 추종 알고리즘은 쟁기와 로터리 작업 상태에서 모두 작은 RMS 위치 오차와 높은 안정성이 가지므로 향후 포장에서 완전한 자율주행 농작업에 사용 적합한 것으로 판단하였다.

Fig. 4-42 Evaluation test on the test field with tillage operation by TX803 tractor

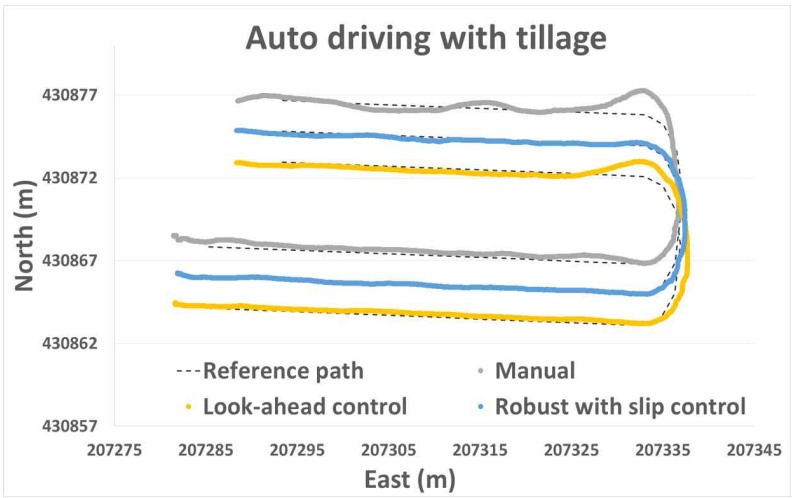


Fig. 4-43 Result of full path tracking trajectory by manual, look-ahead and robust control with tillage operation (TX803 tractor)

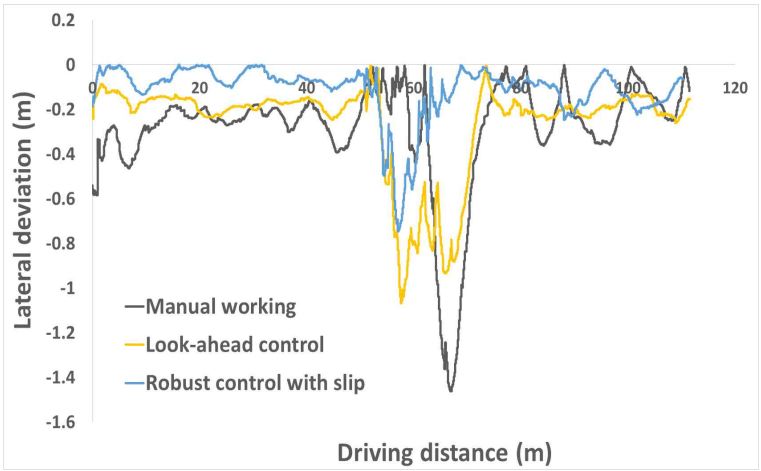
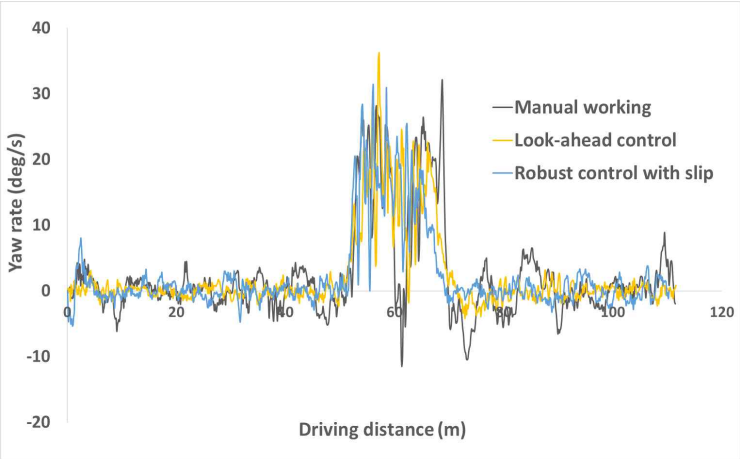
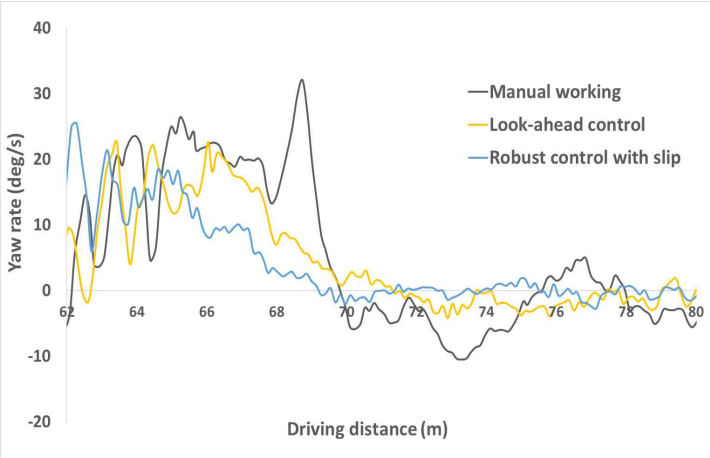


Fig. 4-44 Result of lateral deviation by manual, look-ahead and robust control with tillage operation (TX803 tractor)



(a)



(b)

Fig. 4-45 Result of yaw rate(a) and enlarged part of curved to straight (b) by manual, look-ahead and robust control with tillage operation (TX803 tractor)

Table 4-16 Results of path tracking RMSE by manual, look-ahead and robust control with tillage operation (TX803 tractor)

Working types	RMSE
Manual with tillage	36.26 cm
Look-ahead control with tillage	23.12 cm
Robust control with tillage	9.41 cm

4.2.3. 경사지면에서 경로추종을 위한 GPS 신호 보정 성능 평가

경사지면에서 GPS 신호 보정의 성능 평가를 진행하기 위해 자율주행 트랙터(TS25 모델)를 사용하여 서울대학교 농업생명과학대의 수원 실험 농장에서 실험을 진행하였다. Fig. 4-46 (a)과 같이 경사지면을 모사하기 위해 사다리를 두 개 사용하여 트랙터가 전진 주행 시 좌우 연속적으로 기울어질 수 있게 배열하였다. 트랙터의 전진 주행은 원격 조종기를 사용해 Fig. 4-46 (b)과 같이 트랙터의 한쪽 바퀴가 사다리에 올라가 트랙터를 기울어지게 하였다. Fig. 4-47 (a)에 나타낸 바와 같이 트랙터가 경사지 1과 2에서 좌/우 약 9도를 기울어진 상태에서 GPS 신호 보정 효과를 관찰하였다.

Fig. 4-47 (b)은 GPS 위치 보정 알고리즘 적용 전(청색)과 후(노란색)의 주행 궤적을 나타낸 것이며 위치 보정 후 GPS에서 얻은 위치점이 주행방향에 따라 안정적으로 취득이 가능하였다.

(a) (b)

Fig. 4-46 Test field for GPS signal correction test (a) and picture of evaluation test for GPS signal correction test (b)

트랙터 전진 주행의 시작점과 종료점을 사용해 기준 선을 만들어 Table 4-17과 같이 위치 정밀도를 분석하였다. 경사지 1에서 위치 보정 알고리즘을 적용 후 RMS 위치 오차가 33.26 cm에서 1.94 cm로 감소하였고, 경사지 2에서 RMS 위치 오차가 32.21 cm에서 1.96 cm로 감소한 것을 나타내었다. 본 실험에서 취득한 GPS 신호 보정 성능 평가의 결과로 개발한 알고리즘을 자율주행 트랙터에 탑재하여 향후 경사지면 혹은 불규칙한 노면 환경에서 자율주행 농작업에 사용 가능한 것으로 판단 된다.

(a) (b)

Fig. 4-47 Euler signals from IMU on the TS25 auto-guidance tractor (a) and result of GPS signal by no algorithm and correction algorithm (b)

Table 4-17 Result of RMSE comparison of by no algorithm and the developed GPS signal correction algorithm

	No algorithm	Algorithm
RMSE_Tilt 1	33.26 cm	1.94 cm
RMSE_Tilt 2	32.21 cm	1.96 cm

4.2.4. 미경지 영역 방지를 위한 작업기 제어 시험 평가

자율주행 트랙터(TS25 모델) 시스템을 이용하여 작업 위치 인지기반의 로터리 작업기 상승/하강 제어 성능 실험은 강화군청 농업기술센터에 위치한 실험 포장에서 진행하였다. Fig. 4-48 (a)은 실험 포장의 위성사진이며 Fig. 4-48 (b)은 현장에서 실험하는 모습이다.

트랙터가 포장에서 자율주행 로터리 작업을 하면서 설정된 경로를 따라 미경지 영역 방지를 위한 작업기 제어 알고리즘 적용 전과 후 각각 3반복 주행

실험하였다. 주행 경로는 C형 선회 패턴 경로의 일부분으로 이용하였으며 주행 속도는 0.83 m/s로 설정하였다.

Fig. 4-49에서는 트랙터가 새머리 영역으로 진입 시 제어 알고리즘 적용 전과 후의 결과를 보여줬다. 알고리즘 적용 전은 작업기가 새머리 영역에 도달 전에 상승하게 되어 미경지 발생한 것을 확인되었다. 반대로 알고리즘 적용하게 되면 트랙터가 RDDF에서 요구된 작업기가 상승 명령을 인식하여 정확히 경계 선(청색)에서 작업기가 상승하였다. Fig. 4-50에서는 트랙터가 새머리 영역에서 탈출 시 제어 알고리즘 적용 전과 후의 결과를 보여줬다. 알고리즘 적용 전은 작업기가 경계선에 도달 전에 하강한 것을 확인하였다. 이러한 방식으로는 작업할 필요 없는 영역에서 작업기 하강되어 동력 소모가 증가한다. 반대로 알고리즘 적용하게 되면 트랙터가 RDDF에서 요구된 작업기가 하강 명령을 인식하여 정확히 경계 선(청색)에서 작업기가 하강하였다.

(a) (b)

Fig. 4-48 Test field picture from artificial satellite (a) and evaluation test by TS25 auto-guidance tractor for implement control (b)

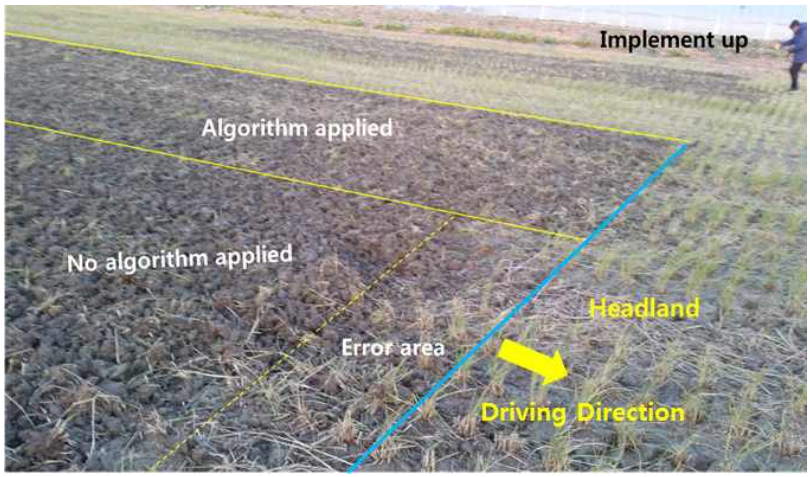


Fig. 4-49 Result of implement up test by no algorithm and the developed algorithm on the field

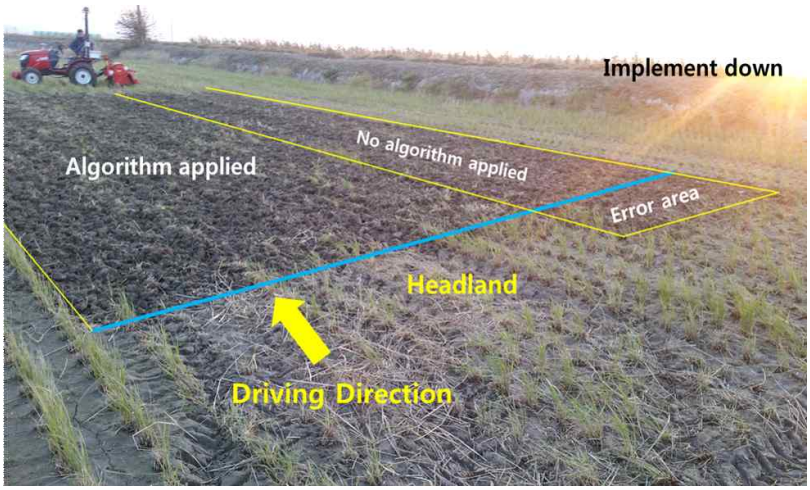
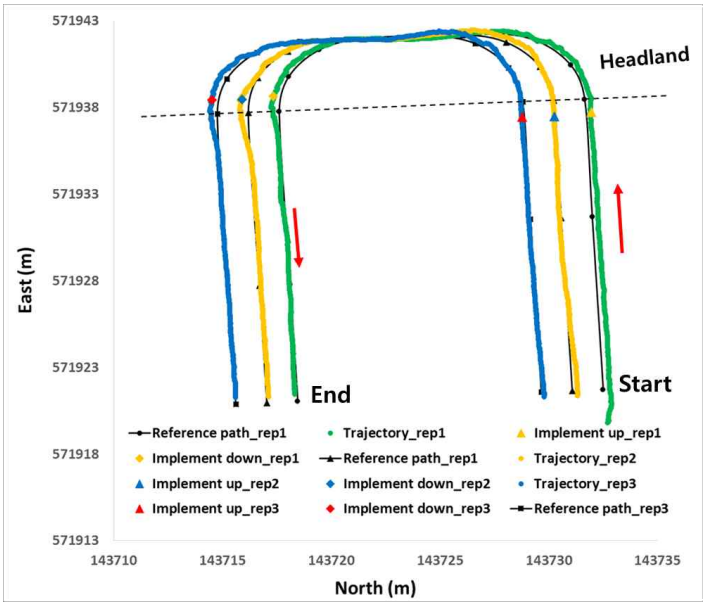


Fig. 4-50 Result of implement down test by no algorithm and the developed algorithm on the field

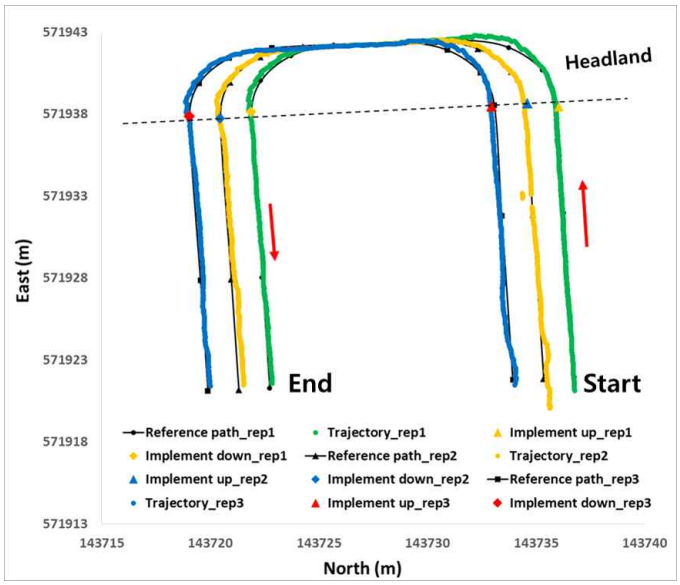
Fig. 4-51 (a)은 트랙터의 자율주행 궤적에서 작업기 제어 알고리즘을 적용 전의 작업기 실제 작동 위치점을 나타내었으며 새머리 영역의 경계선(검정색

점선)에서 작업기가 작동하지 못한 것을 확인했다. Fig. 4-51 (b)은 트랙터의 자율주행 궤적에서 작업기 제어 알고리즘을 적용 후의 작업기 실제 작동 위치점을 나타내었으며 새머리 경계선(검정색 점선)에서 작업기가 작동 가능한 것을 확인하였다. 여기서 작업기의 실제 작동 위치점은 작업기 상승/하강 제어에 피드백신호로 사용하는 포텐쇼미터를 이용하여 작업기 상태를 판단하였다.

작업기 상승/하강의 제어 위치 오차는 작업기가 실제 작동된 위치와 목표 위치를 비교하여 계산하였다. 트랙터가 새머리 영역으로 진입할 시 평균 RMS 위치 오차가 0.91 m에서 0.14 m로 감소하였으며, 새머리 영역에서 탈출 시 평균 RMS 위치 오차가 0.86 m에서 0.13 m로 감소하는 것을 Fig. 4-52에서 확인할 수 있었다. Table 4-18은 3반복 실험 결과를 보여 주었으며 알고리즘 적용 후의 작업기 상승과 하강의 표준편차는 각각 0.05 m과 0.06 m 작은 값으로 나타내었다.



(a)



(b)

Fig. 4-51 Result of position of implement operating point on tracking trajectory by no algorithm (a) and the developed algorithm (b)

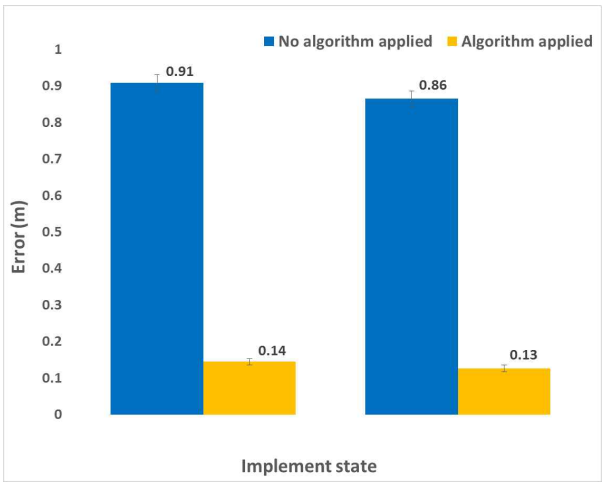


Fig. 4-52 Result of implement control RMSE by no algorithm and the developed algorithm

Table 4-18 Result of error analysis of each repeated test by no algorithm and the developed algorithm

	No algorithm			Algorithm		
	Rep 1	Rep 2	Rep 3	Rep 1	Rep 2	Rep 3
Error (m)	Up: 0.93 Down: 0.93	Up: 0.92 Down: 0.81	Up: 0.88 Down: 0.85	Up: 0.20 Down: 0.16	Up: 0.10 Down: 0.19	Up: 0.13 Down: 0.02
AVG (m)	Up: 0.91 / Down: 0.86			Up: 0.14 / Down: 0.13		
STDEV (m)	Up: 0.03 / Down: 0.06			Up: 0.05 / Down: 0.07		

4.2.5. 직교형태 포장에서 자율주행 쟁기 작업 성능 평가

본 절에서는 쟁기작업 기반으로 동적 경로추종 알고리즘을 적용하여 전체 포장에서 무인 주행 및 작업 성능을 평가한 내용을 다루고자 한다. 실험 실시 장소는 서울대학교 농업생명과학대의 수원에 위치한 107번 포장에서 시행하였다. 시험 포장의 크기는 길이가 80 m, 폭이 30 m이다. 포장에서 경로생성을 위해 포장의 입력변수인 4개 꼭짓점의 경도/위도 TM좌표는 VRS-GPS를 이용 취득하였으며 트랙터와 작업기의 변수는 Table 3-3과 같이 이용하였다.

실제 자율주행 쟁기작업을 실시하기 전 개발한 가상 시뮬레이터 환경에서 생성된 C형 선회 패턴 경로를 기반으로 미리 자율주행 시뮬레이션 하여 주행 위치 정밀도, 포장 작업시간 및 포장 작업효율을 예측하였다. 트랙터가 직진 왕복 작업부에서 주행 속도는 1.12 m/s로 유지하였으며 선회구역에서는 0.56 m/s로 저속상태로 주행하였다. Fig. 4-53은 시뮬레이터 환경에서 주행 상태 및 미 작업 영역을 보여주었으며 Fig. 4-54에서 동적 경로추종 알고리즘을 기반의 자율주행 궤적을 나타낸 것이다.

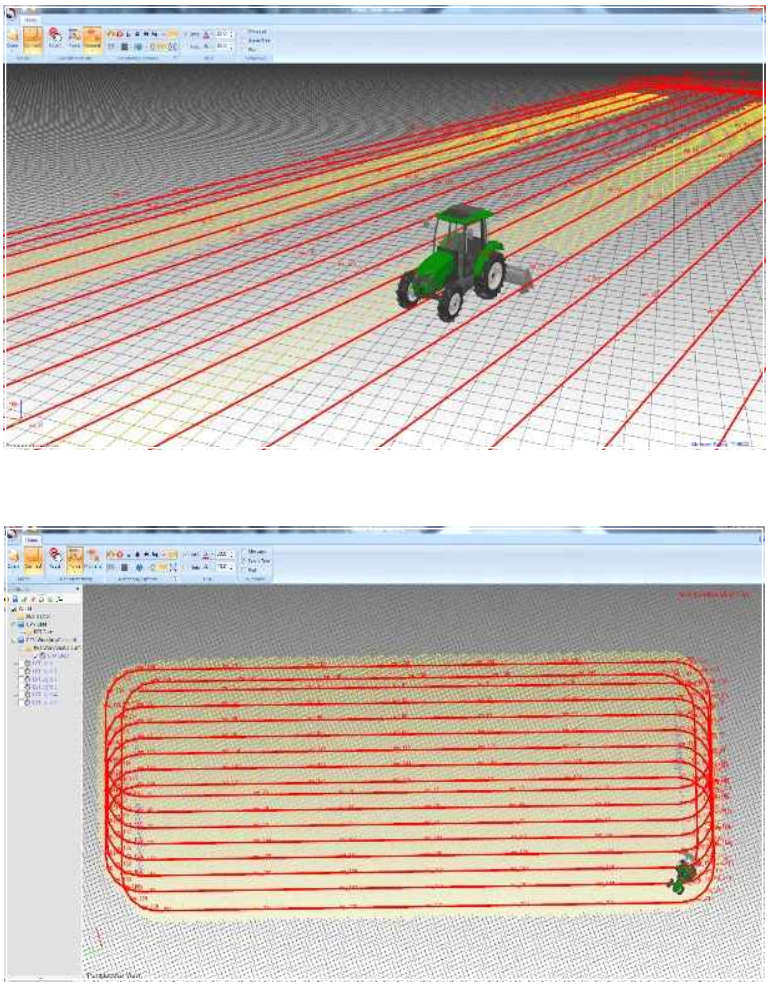


Fig. 4-53 Simulation interface with autonomous plowing operation

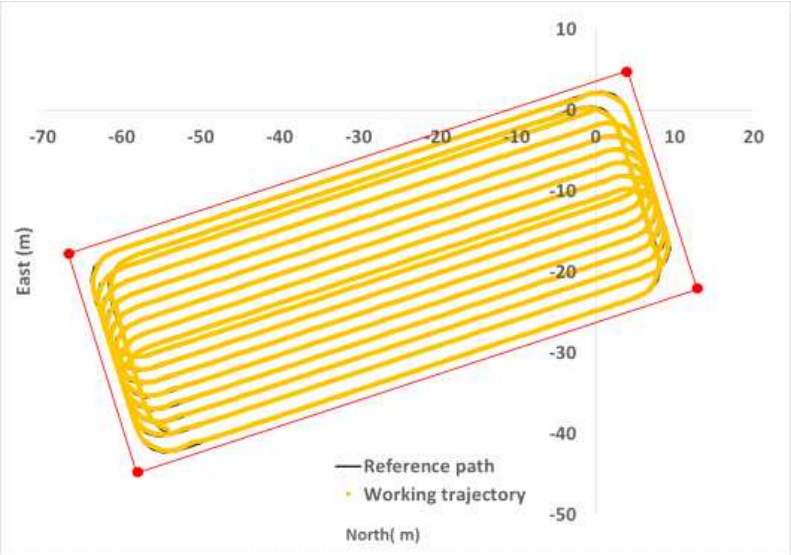


Fig. 4-54 Result of simulation analysis of full path tracking trajectory with autonomous plowing operation

실험 포장에서 생성된 C형 선회패턴 작업경로의 왕복 직진 주행부분은 총 21개로 포함되며 트랙터가 각각 진선 경로 상에서 자율주행 RMS 위치 오차는 Fig. 4-55와 같이 나타내고 있다. 자율주행 시 직진 경로에서 RMS 위치 오차의 평균값은 11.5 cm였으며 각각 직진 경로에 대해 살펴보면 최소 오차는 5.08 cm이고 최대 오차는 13.08 cm로 나타내었다. 여기서 첫 번째 직진 경로에서 RMS 위치 오차가 상대적으로 작게 나타낸 원인은 트랙터가 작업 경로의 시작점에서 출발했기 때문이다.

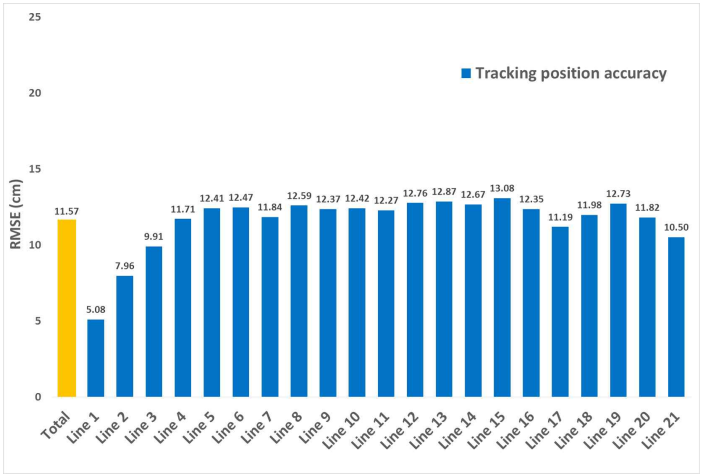


Fig. 4-55 Result of position RMSE with autonomous plowing operation by simulation

자율주행 트랙터의 주행 시간과 작업 면적에 대해 성능 평가 결과는 Table 4-19와 같이 정리하였다. 주행 시간의 측면에서 살펴보면 포장 작업시간은 37.38 min, 선회 소모시간은 5.91 min이며 포장 작업효율은 84.2%로 나타났다. 또한 작업 면적의 측면에서 살펴보면 전체 포장 면적 2100 m^2 중에서 미 작업 면적은 64.59 m^2로 나타났다. 시뮬레이션을 통해 예측된 주행 및 작업 성능의 유효함이 확인되어 실험하고자하는 포장에서 개발한 동적 경로추종 알고리즘을 이용하여 자율주행을 통한 쟁기 작업을 실시하였다.

Table 4-19 Results of the simulation comparison of operation performance with autonomous plowing operation

Time	Field time	Effective field time	Field time loss in turning	Field efficiency
	2243.26 sec	1888.91 sec	354.35 sec	84.2%
Area	Ideal field area	Effective operation area	Skip area	Ratio of effective operation area
	2100 m^2	2035.41 m^2	64.59 m^2	96.9%

Fig. 4-56 (a)에 포장의 위성사진을 나타내었으며 Fig. 4-56 (b)은 포장에서 자율주행을 통한 쟁기작업 상태로 실험하는 모습이다. 포장 통양의 물리적 특성을 구명하기 위하여 새머리 선회부분인 A, D 구역과 직전 주행부분인 B, C 구역에서 각각 일정한 간격으로 3개위치를 선정하여 토양에 대하여 토성, 원추지수 및 토양의 수분함량을 측정하였다.

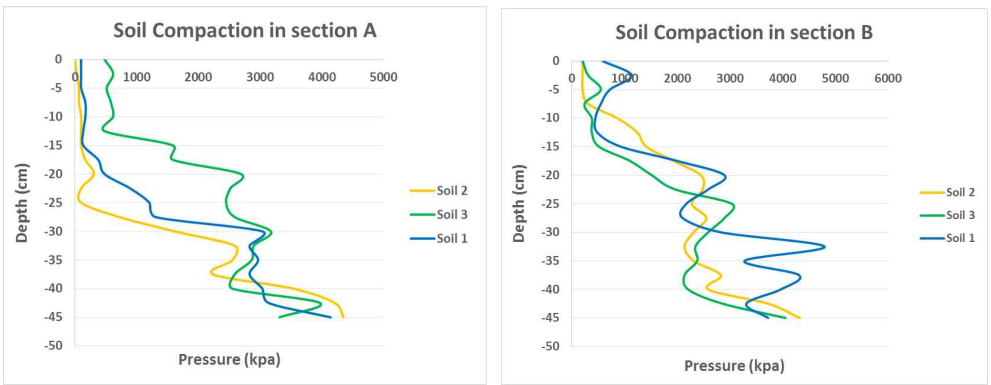
Fig. 4-56 Test field in Suwon from artificial satellite (a) evaluation test on the test field with plowing operation by TX803 tractor (b)

Table 4-20과 같이 포장 A 구역의 경우 모래 37.80%, 실트 39.72%, 점토 22.48%, 함수율 26.3% 이었으며, B 구역의 경우 모래 36.88%, 실트 38.68%, 점토 24.44%, 함수율 22.8% 이었으며, C 구역의 경우 모래 34.96%, 실트 39.04%, 점토 26.00%, 함수율 19.41% 이었으며 C 구역의 경우 모래 37.32%, 실트 37.40%, 점토 25.28%, 함수율 22.3%로 분석되어 미국 농무성법(USDA)에 의하면 4개 구역의 토성이 모두 양질 사토(Sandy Loam)로 분류되었다.

Table 4-20 Soil properties and moisture contents of experimental field

	Items	Values
Section A	Sand (%)	37.80
	Silt (%)	39.72
	Clay (%)	22.48
	Moisture contents (%)	26.3
Section B	Sand (%)	36.88
	Silt (%)	38.68
	Clay (%)	24.44
	Moisture contents (%)	22.8
Section C	Sand (%)	34.96
	Silt (%)	39.04
	Clay (%)	26.00
	Moisture contents (%)	19.41
Section D	Sand (%)	37.32
	Silt (%)	37.40
	Clay (%)	25.28
	Moisture contents (%)	22.3

토양의 원추지수는 포장 내에 왕복 작업부와 새머리 선회부에 포함된 9개 지점에서 원추 관입 시험기를 이용하여 30 mm/s의 토양 관입 속도로서 지면으로부터 2.5 cm 간격으로 35 cm 깊이까지 측정하였다. Fig. 4-57은 A, B, C, D 각각 구역에서 측정한 원추지수를 측정 깊이에 따라 나타낸 것이다.



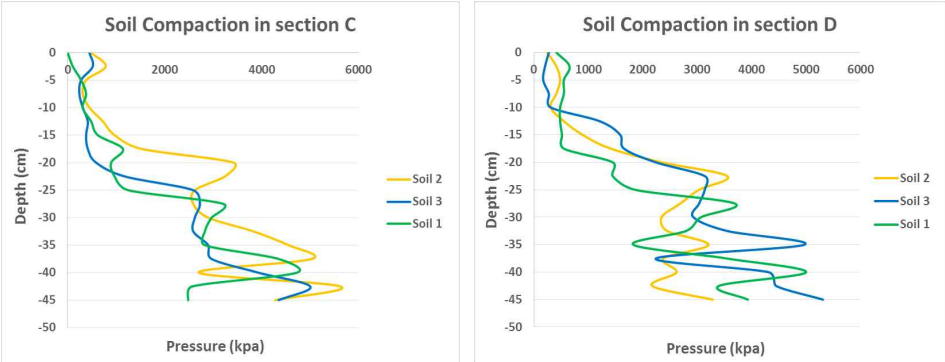


Fig. 4-57 Cone index with soil depth at different field section

본 실험에서 트랙터를 작업 경로의 시작 위치 점으로 이동 후 자율주행 기반의 쟁기작업을 실시하였다. 작업 속도는 직진 왕복 작업부에서 1.12 m/s로 유지하였으며 선회구역에서는 0.56 m/s로 저속상태로 주행하였다. 작업기의 상승과 하강은 전자트랙터에 탑재된 제어기능을 사용하여 전륜의 회전 각도가 30도 이상 시 작업기가 자동 상승하며 30도 이하 시 자동 하강한다. Fig. 4-58은 동적 경로추종 알고리즘을 기반의 자율주행 궤적을 나타낸 것이다.

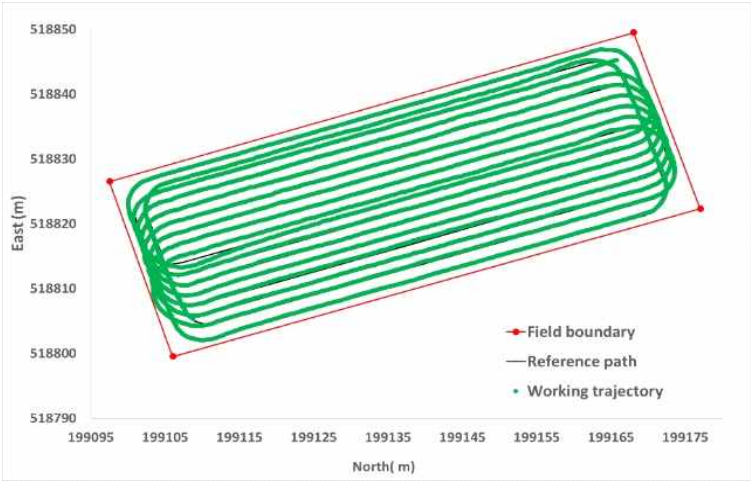


Fig. 4-58 Result of full path tracking trajectory with plowing operation in the field test

Fig. 4-59는 21개의 직진 경로에 대해 각각 직진 작업 주행 RMS 위치 오차를 나타낸 것이다. 직진 경로에서 RMS 평균 위치 오차는 13.66 cm이며 각각 직진 경로에 대해 살펴보면 최소 오차는 6.70 cm이고 최대 오차는 23.24 cm로 나타내었다. Fig. 4-60은 21개의 직진 경로에 대해 각각 작업 주행 RMS 방향 오차를 나타낸 것이다. 직진 경로에서 RMS 평균 방향 오차는 4.91 deg이며 각각 직진 경로에 대해 살펴보면 최소 오차는 3.61 deg이고 최대 오차는 6.71 deg로 나타났다. RMS 최대 위치와 방향 오차의 발생 구역은 두 번째 직진 경로에서 나타내었으며 자율주행 궤적에서 살펴보면 포장 구역 A 중앙 부분에서 트랙터 궤적이 일정하지 못 하고 계획된 경로(검정색)에서 이탈 현상이 있다. 이것이 Fig. 4-61과 같이 포장 구역 A의 일부분의 토양 이의 함수율이 26.3%로 상대적 높기 때문에 트랙터가 새머리 영역에서 탈출하여 내부 왕복 작업부로 진입 시 타이어가 측면으로 밀리는 원인으로 판단되었다.

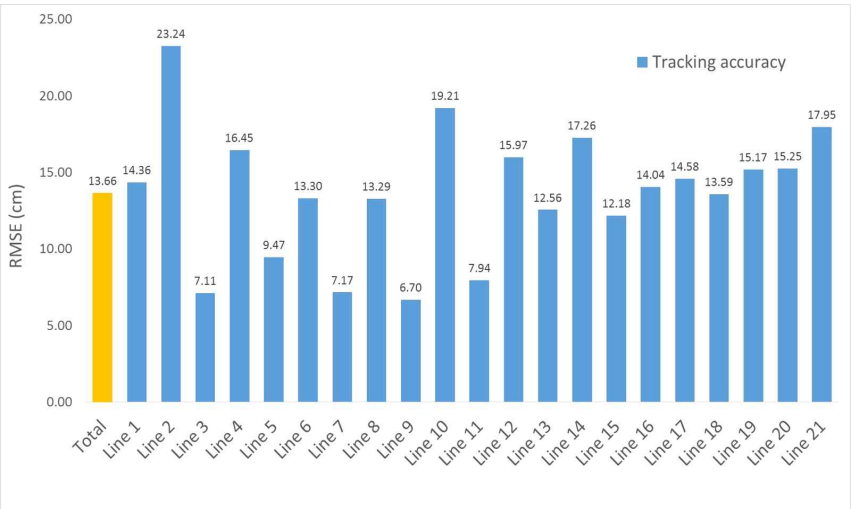


Fig. 4-59 Result of position RMSE with autonomous plowing operation in the field test

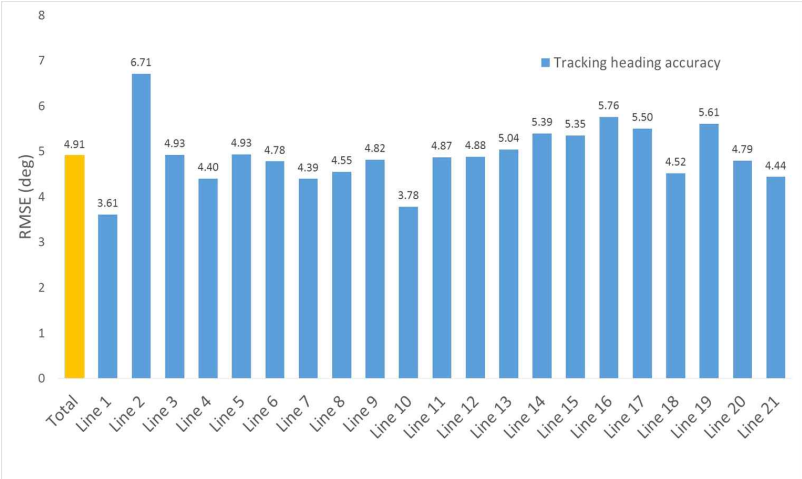


Fig. 4-60 Result of heading RMSE with autonomous plowing operation in the field test

Fig. 4-61 The most moisture part at headland A

Table 4-21은 트랙터가 자율주행을 통한 쟁기 작업 면적과 시간에 대한 연료 소비량을 나타낸 것이다. 자율주행 트랙터의 주행 시간과 작업 면적에 대해 성능 평가 결과는 Table 4-22와 같이 정리하였다. 주행 시간의 측면에서

살펴보면 포장 작업시간은 48.61 min, 선회 소모시간은 9.75 min이며 포장 작업효율은 79.9%로 나타났다. 김학규 (1995)의 박사학위논문의 대 구획 포장의 벼농사 기계화 시스템에 관한 연구에서 길이가 80 m, 폭이 30 m 크기의 포장에서 6련 쟁기를 사용할 때 포장 작업효율이 약 64%로 제시하여 본 실험의 포장 작업효율의 타당성이 있는 것으로 판단하였다. 미경지 영역을 분석하기 위해 Arc GIS (Ver. 10.1) 프로그램을 이용하여 Fig. 4-62와 같이 작업기의 작업 위치 정보를 사용하여 작업영역을 표시하였다. 즉 작업 면적의 측면에서 살펴보면 전체 포장 면적 2100.4 m^2 중에서 미 작업 면적은 204.3 m^2로 나타냈으며 포장 작업면적의 비율은 90.3%로 판단되었다.

Table 4-21 Result of fuel consumption measurement with autonomous plowing operation in the field test

Units	Values
L/h	8.7
L/ha	26.6

Table 4-22 Results of the comparison of operation performance with autonomous plowing operation in the field test

Time	Field time	Effective field time	Field time loss in turning	Field efficiency
	2917.19 sec	2331.596 sec	585.301 sec	79.9%
Area	Ideal field area	Effective operation area	Skip area	Ratio of effective operation area
	2100.4 m^2	1895.7 m^2	204.3 m^2	90.3%

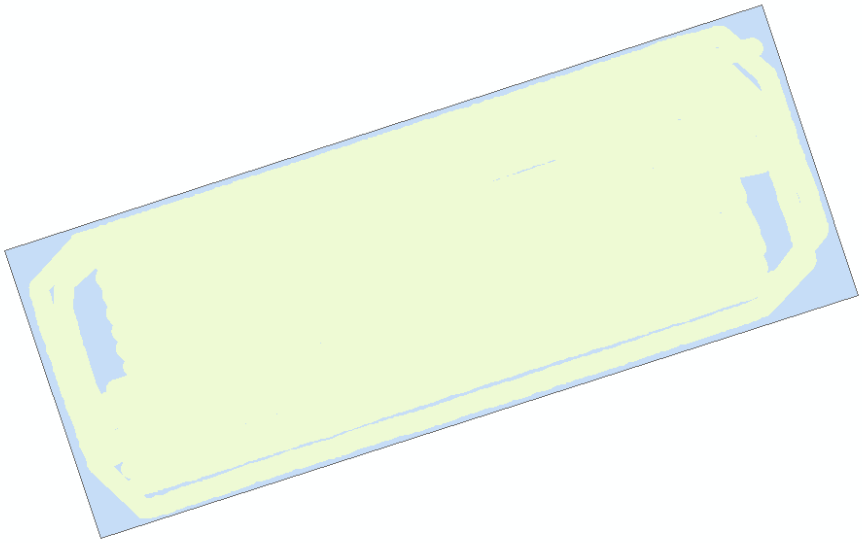


Fig. 4-62 Result of coverage area analysis by ArcGIS software

트랙터에 탑재된 계측 시스템을 이용 쟁기 작업의 위치 별 견인력과 연료소모량을 Fig. 4-63, Fig. 4-64와 같이 각각 나타내었다. 본 연구에서 쟁기 작업을 통해 취득한 견인력과 연료 소모량을 관찰하기 위해 모든 선회구역의 비작업부분은 제외하여 작업구역만 분석하였다.

앞서 설명과 같이 포장 A 구역에서 토양의 함수율이 상대적으로 높기 때문에 트랙터가 새머리 선회 작업 시 A 구역에서 측정된 견인력이 상대적 낮은 값으로 나타났다. 이와 같은 원인으로 견인력 낮은 A 구역에서 연료 소모량도 동일하게 적은 것으로 나타났다.

본 시험 평가에서 왕복 작업부에서 트랙터가 작업 시 작업 깊이를 15 cm로 설정하였으나 위치별로 토양의 토질이 다르고 트랙터 기울어진 상태에서 쟁기 작업기가 땅속에 박혀 트랙터가 전진 어려운 현상이 있었다. 이러한 현상은 Fig. 4-63에 빨간색으로 표시된 위치에서 발생한 것을 판단이 가능하였다. 향

후 이러한 문제점을 극복하기 위해 토양의 반력 및 트랙터 주행 자세에 의해서 작업기 자동 상승 하강 제어가 관련 연구가 필요한 것으로 판단되었다.

또한 본 시험 평가에서 사용한 트랙터의 작업기 제어는 이미 탑재된 제어기능을 사용하여 전륜의 회전 각도가 30도 이상 되면 작업기가 자동 상승하도록 구성이 되어 high level의 경로추종 알고리즘에서 전송된 작업기 제어명령을 인지 못하게 되어있다. 그러므로 Fig. 4-63에 보라색 주행 궤적과 같이 새머리 진입구역에서 작업기가 해당영역에 도달하지 못하는 위치에서 미리 상승하게 되어 미경지부분이 발생하였다. 미경지 영역 방지를 위한 작업기 제어 알고리즘은 자율주행 트랙터(TS25 모델)에서 평가가능하며 작업기가 정확하게 새머리 진입구역에서 작동이 가능한 결과를 검증한다. 향후에는 본 논문에서 검증된 미경지 발생 방지를 위한 작업기 제어 알고리즘을 자율주행 트랙터(TX803 모델) 시스템에 탑재하면 미경지 부분이 제거가 가능할 것으로 사료된다.

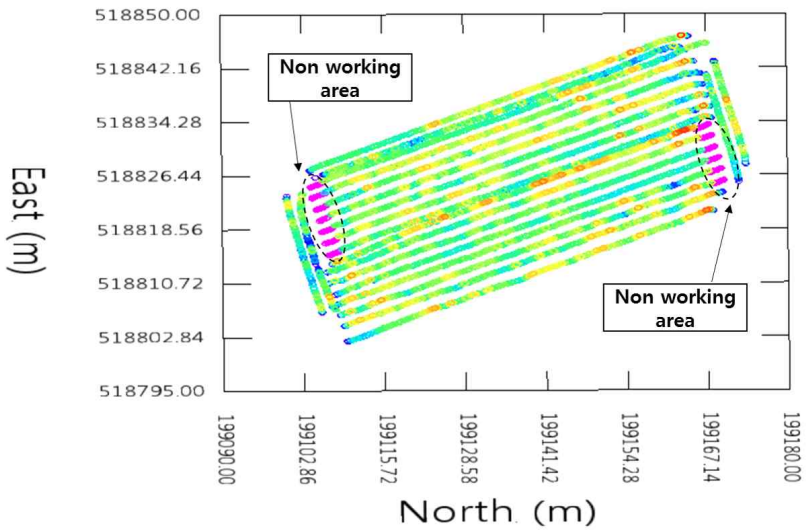


Fig. 4-63 3D plot of draft force with autonomous plowing operation on field test

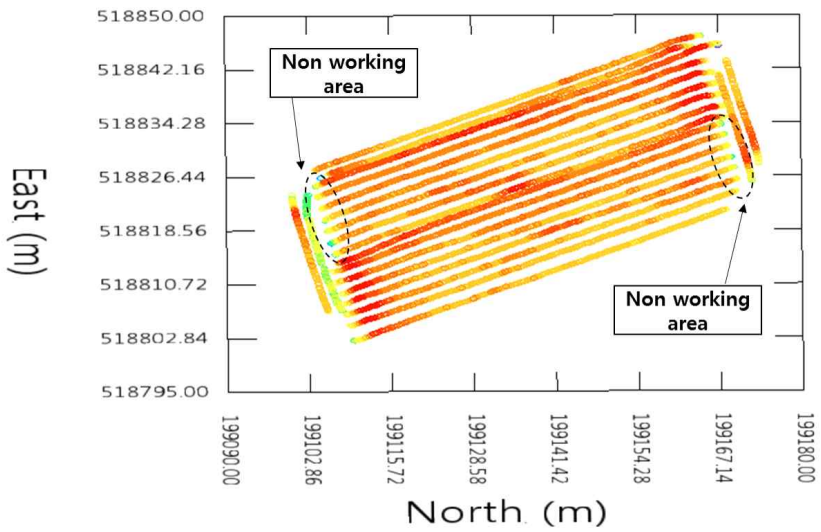


Fig. 4-64 3D plot of consumption with autonomous plowing operation on field test

특정한 지점에서 특성치를 분석하기 위해 정규 크리깅법을 이용하여 위치별의 견인력과 연료 소모량의 산술 평균값을 취득하여 그 대표 값을 계산하였다. Surfer 13 프로그램을 이용 주위의 실측값들을 선형으로 조합하며 통계적으로 분석하여 Fig. 4-65, Fig. 4-66과 같이 자율주행을 이용한 쟁기 작업의 견인력과 연료 소모량에 대한 등고선 선도를 나타내었다.

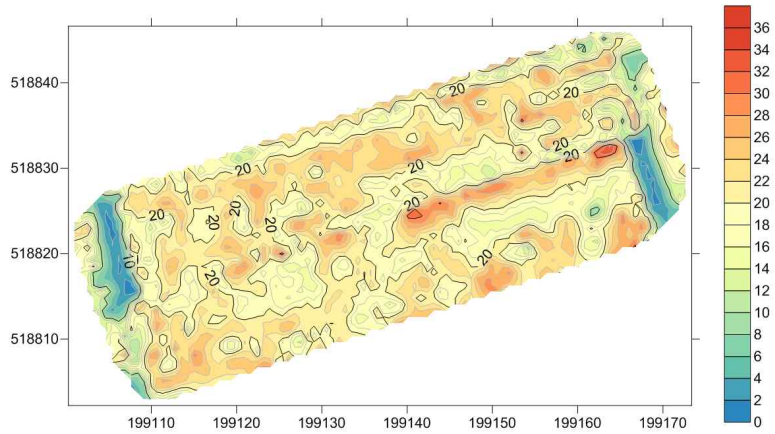


Fig. 4-65 Countor plot of draft force with autonomous plowing operation in the field test

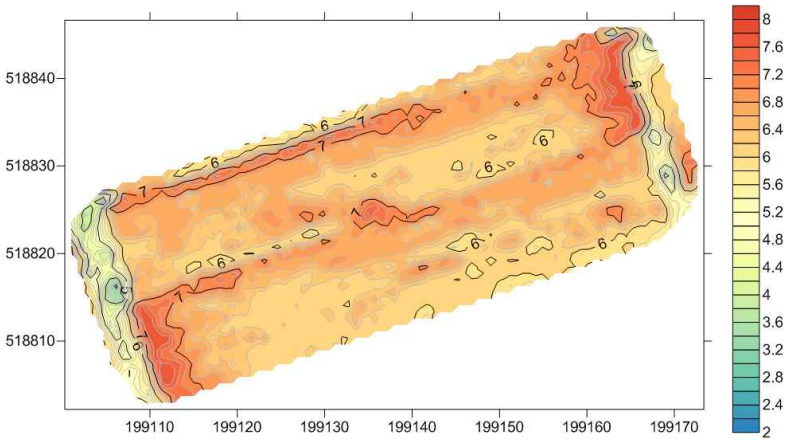


Fig. 4-66 Countor plot of fuel consumption with autonomous plowing operation in the field test

4.2.6. 다변형태 포장에서 자율주행 작업 평가

본 절에서는 자율주행 트랙터(TS25 모델) 시스템을 이용하여 사다리와 “L” 자의 두 가지 형태 포장에서 자율주행 농작업을 실시하였다. 실험은 강화군청 농업기술센터에 위치한 실험 포장에서 진행하였으며 두 가지 형태의 다변형태 영역을 각각 두 개식 구성하여 자율주행과 숙련된 트랙터 운전자의 작업 성능을 비교하였다.

(1) 사다리 형태의 포장에서 자율주행 작업 평가

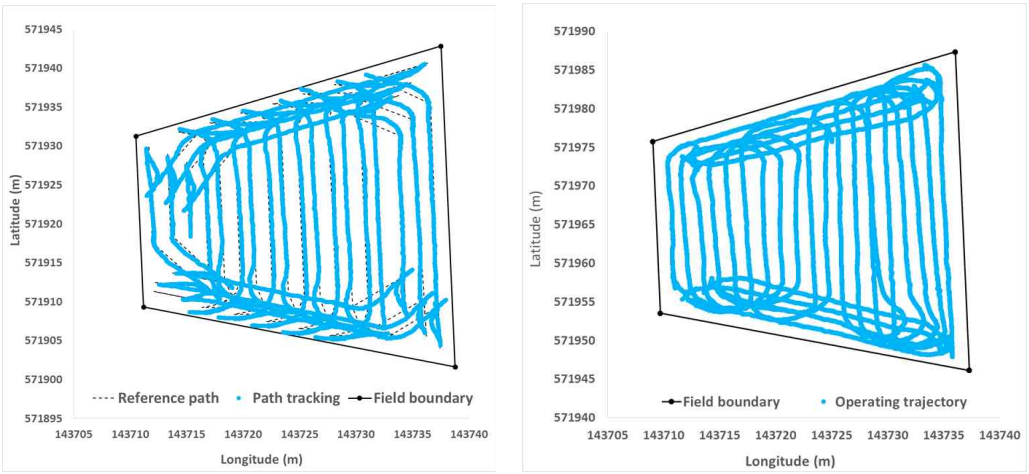
사다리 형태의 포장 면적은 약 0.09 ha이며 포장 좌표정보를 취득 후 생성된 경로를 따라 자율주행 로터리 작업을 실시하였다. 주행속도는 약 3 km/h이며 작업 겹침 량은 10 cm이었다. Fig. 4-67(a)과 (b)는 각각 사다리 형태 포장영역에서 자율주행과 숙련된 트랙터 운전자의 작업진행 사진 결과 모습을 나타낸 것이다.

(a)

(b)

Fig. 4-67 Evaluation test pictures by autonomous (a) and manual (b) in the convex polygonal field

Fig. 4-68(a)과 (b)는 각각 사다리 형태 포장영역에서 자율주행과 숙련된 트랙터 운전자의 작업 궤적을 나타낸 것이며 Table 4-23은 작업 성능 비교 결과를 나타낸 것이다. 자율주행 기반의 로터리 작업의 주행 RMS 위치 오차는 14.9 cm 이었으며 숙련된 트랙터 운전자는 작업의 겹침 량을 약 15 cm로 진행하였기 때문에 자율주행에서 설정된 겹침 량 10 cm 보다 더 넓기 때문에 직진 작업 왕복 수가 증가하여 순 작업시간이 자율주행 작업 보다 큰 것으로 나타났다. 그러나 본 실험에서 자율주행 트랙터의 포장 작업시간은 48.6 min 소요하였으며 숙련된 트랙터 운전자 보다 9.1 min 감소함을 확인하였고 포장 작업효율(62.1%)도 숙련된 트랙터 운전자의 수준(65.7%)으로 도달하는 것으로 판단되었다.



(a) (b)

Fig. 4-68 Result of travelling trajectory by autonomous(a) and manual(b) in the convex polygonal field

Table 4-23 Result of the comparison of operation performance between the autonomous and manual operation in the convex field

	Field time	Effective field time	Field time loss in turning	Field efficiency	RMSE
Auto	48.6 min	30.2 min	18.5 min	62.1%	14.9 cm
Manual	57.7 min	37.7 min	19.6 min	65.7%	

미경지 영역 발생 방지를 위한 작업기 제어 기술을 본 실험에서 포함하여 진행하였다. 작업기의 하강 작업 상태 및 상승 비 작업 상태의 판단은 Fig. 4-69에 나타낸 것과 같이 작업기 위치 센서를 이용 확인하였다. Fig. 4-70(a)은 내부 작업경로에서 작업기 하강 상태의 작업 궤적을 나타낸 것이며 Fig. 4-70(b)은 외부 작업경로에서 작업기 하강 상태의 작업 궤적을 나타낸 것이다. 작업기가 RDDF에서 요구된 작업기 상승(녹색) 및 하강(빨간색) 위치에

도달 전 작동하지 않았으며 새머리 영역으로 진입 및 새머리 영역에서 탈출 시점에서 정확히 작업기 작동하는 것을 확인하였다.

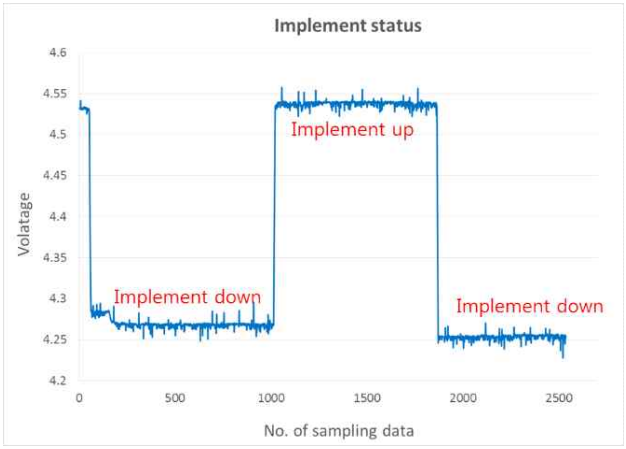
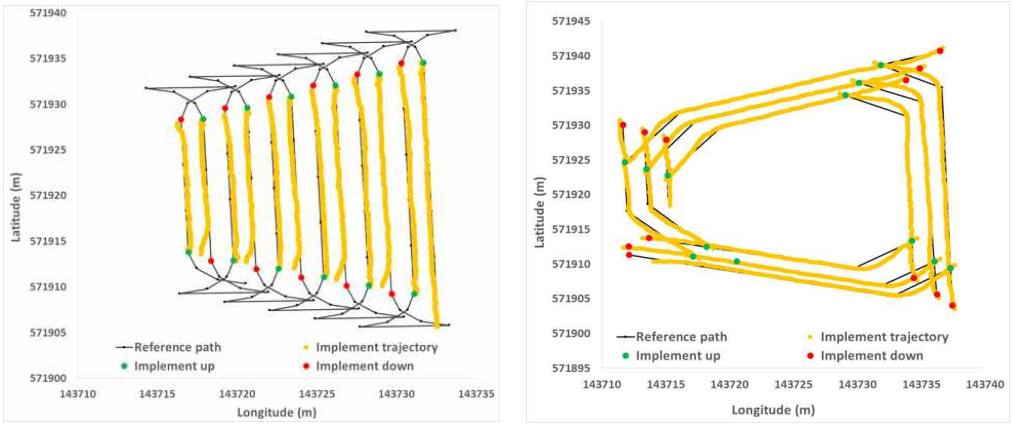


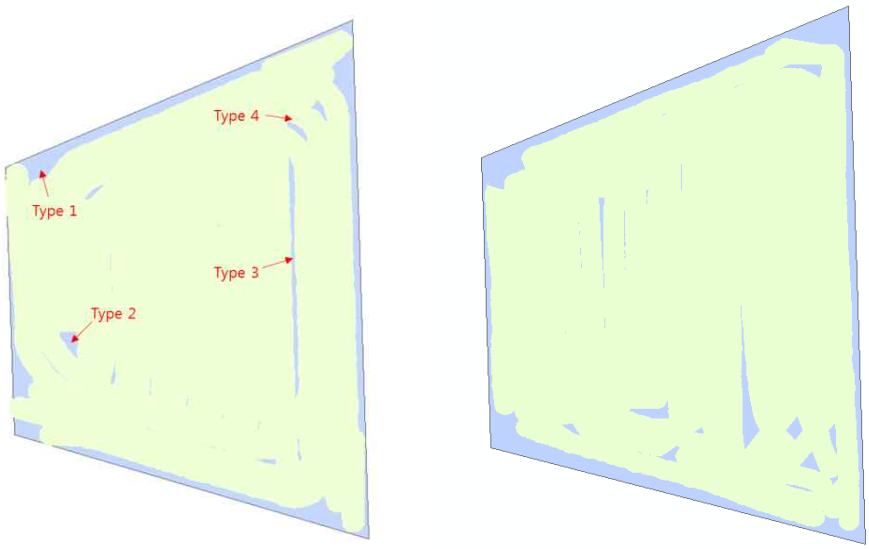
Fig. 4-69 Implement status from the sensor values



(a) (b)

Fig. 4-70 Result of effective working trajectory at inside working area (a) and outside working area (b) in the convex field

미경지 영역은 Fig. 4-71과 같이 작업기의 작업 위치 정보를 사용 분석하였다. Fig. 4-71(a)은 자율주행 상태에서 작업영역을 나타낸 것이며 Table 4-24와 같이 미 작업 영역은 75.3 m^2, 포장 순 작업면적의 비율은 91.1%이다. 미경지 발생 구역을 네 가지로 구분하여 분석하였다. Type 1, 2의 미경지는 모두 주행경로에 대해 부족하게 생성하여 발생 원인으로 판단되었다. Type 3, 4의 미경지는 경로간의 거리가 작업기 폭 만큼 인접하지 못하여 발생 원인으로 판단되었다. Fig. 4-71(b)은 작업자가 직접 작업인 작업영역 나타낸 것이며 Table 4-24와 같이 미 작업 영역은 93.1 m^2, 포장 순 작업면적의 비율은 89.0%이다. 본 실험에서 자율주행 트랙터의 작업면적 효율은 숙련된 트랙터 운전자의 수준으로 도달하는 것으로 판단되었다.



(a) (b)

Fig. 4-71 Result of coverage area analysis by autonomous(a) and manual(b) in the convex polygonal field

Table 4-24 Result of the comparison of coverage area between the autonomous and manual operation in the convex field

	Ideal field area	Effective operation area	Skip area	Ratio of effective operation area
Autonomous	842.5 m^2	767.2 m^2	75.3 m^2	91.1%
Manual	846.7 m^2	753.6 m^2	93.1 m^2	89.0%

(2) “L”자 형태의 포장에서 자율주행 작업 평가

“L”자 형태의 포장 면적은 약 0.14 ha이며 포장 좌표정보를 취득 후 생성된 경로를 따라 자율주행 로터리 작업을 실시하였다. 주행속도는 약 3 km/h 이며 작업 겹침 량은 10 cm이었다. Fig. 4-72는 각각 “L”자 형태 포장영역에서 자율주행과 숙련된 트랙터 운전자의 작업진행 사진 모습을 나타낸 것이다.

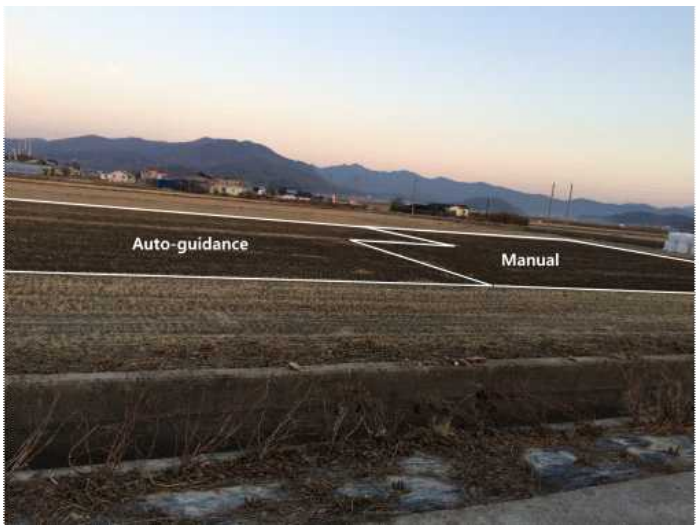
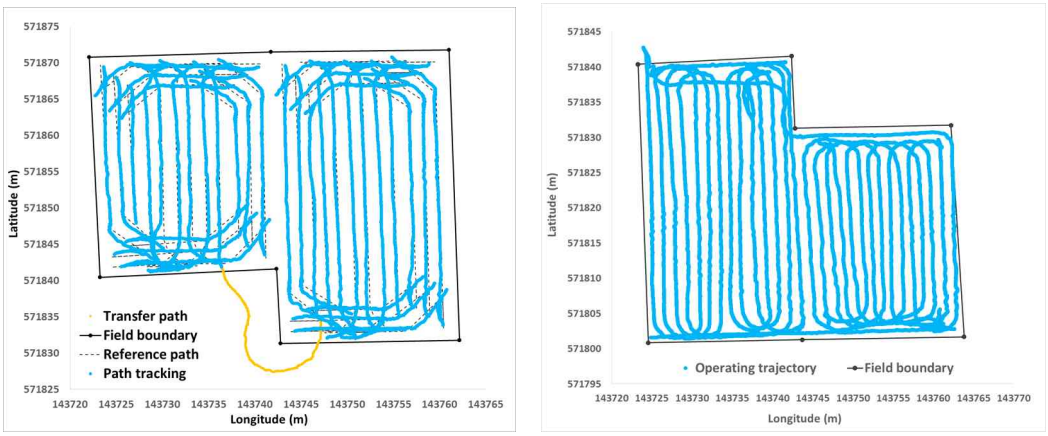


Fig. 4-72 Evaluation test picture by autonomous and manual in the concave polygonal field

Fig. 4-73(a)과 (b)는 각각 “L”자 형태 포장영역에서 자율주행과 숙련된 트랙터 운전자의 작업 궤적을 나타낸 것이며 Table 4-25는 작업 성능 비교 결과를 나타낸 것이다. 자율주행 기반의 로터리 작업의 주행 RMS 위치 오차는 21.8 cm 이었으며 숙련된 트랙터 운전자는 작업의 겹침 량을 약 15 cm로 진행하였기 때문에 자율주행에서 설정된 겹침 량인 10 cm 보다 더 넓어 직진 작업 왕복 수가 증가하여 순 작업시간이 자율주행 작업 보다 큰 것으로 나타났다. 본 실험에서 concave형태 포장에서 sub 포장간의 이동 경로생성은 미완성로 인해 매뉴얼 상태로 이동하였다. Table 4-25와 같이 자율주행 농작업 시 포장 작업시간은 69.5 min 소요하였으며 포장 작업효율(65.5%)은 숙련된 트랙터 운전자의 수준(70.6%)으로 도달하는 것으로 판단되었다.



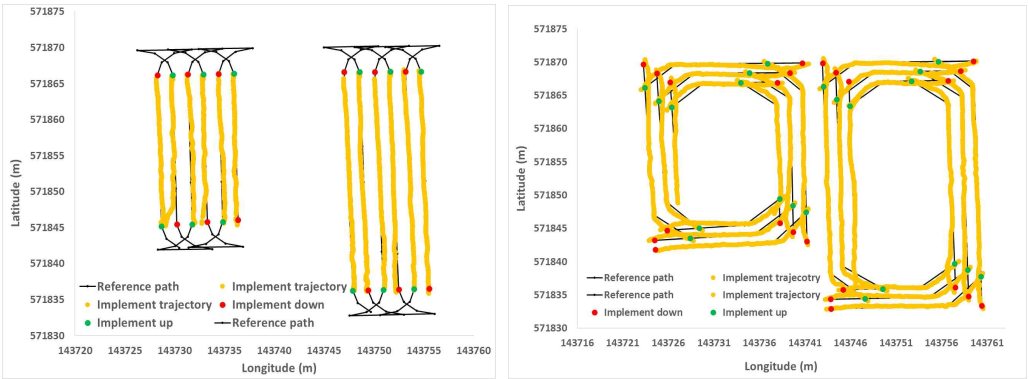
(a) (b)

Fig. 4-73 Result of travelling trajectory by autonomous (a) and manual (b) in the concave polygonal field

Table 4-25 Result of the comparison of operation performance between the autonomous and manual operation in the concave field

	Field time	Effective field time	Field time loss in turning	Field efficiency	RMSE
Auto	69.5 min	45.5 min	24.0 min	65.5%	21.8 cm
Manual	65.3 min	46.2 min	19.1 min	70.6%	

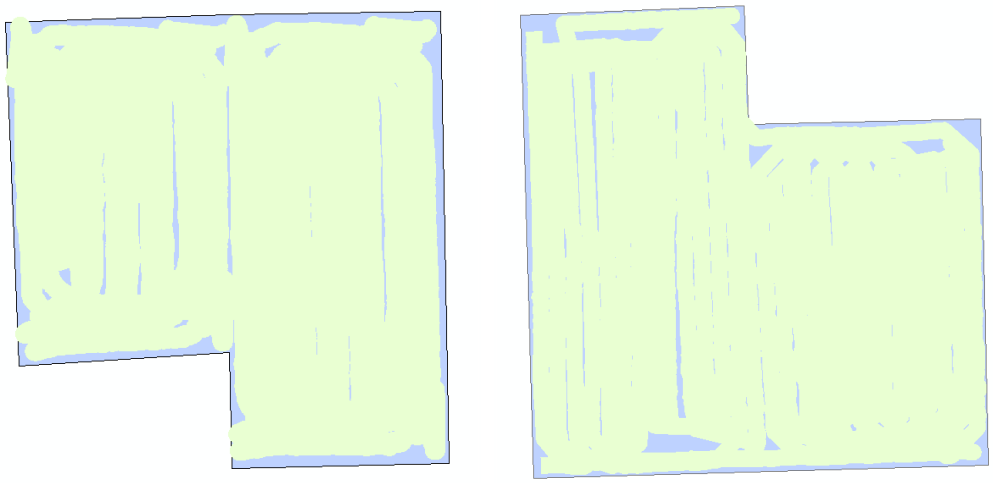
미경지 영역 발생 방지를 위해 앞서 개발한 작업기 제어 기술을 본 실험에 포함하여 진행하였다. Fig. 4-74(a)는 내부 작업경로에서 작업기 하강상태의 작업 궤적을 나타낸 것이며 Fig. 4-75(b)는 외부 작업경로에서 작업기 하강상태의 작업 궤적을 나타낸 것이다. 작업기가 RDDF에서 요구된 작업기 상승(녹색) 및 하강(빨간색) 위치에 도달 전 작동하지 않았으며 새머리 영역으로 진입 및 새머리 영역에서 탈출 시점에서 정확히 작업기 작동하는 것을 나타났다.



(a) (b)

Fig. 4-74 Result of effective working trajectory at inside working area(a) and outside working area(b) in the concave field

미경지 영역을 분석하기 위해 Arc GIS (Ver. 10.1) 프로그램을 이용하여 Fig. 4-75와 같이 작업기의 작업 위치 정보를 사용하여 작업영역을 표시하였다. Fig. 4-75(a)는 자율주행 상태인 작업영역 나타낸 것이며 Table 4-26과 같이 미 작업 영역은116 m^2, 포장 순 작업면적의 비율은 91.4%이다. Fig. 4-75(b)는 작업자가 직접 작업인 작업영역 나타낸 것이며 Table 4-26과 같이 미 작업 영역은 98.2 m^2, 포장 순 작업면적의 비율은 92.7%이다. 본 실험에서 자율주행 트랙터의 작업면적 효율은 숙련된 트랙터 운전자의 수준으로 도달하는 것으로 판단된다.



(a) (b)

Fig. 4-75 Result of coverage area analysis by autonomous(a) and manual(b) in the concave polygonal field

Table 4-26 Result of the comparison of coverage area between the autonomous and manual operation in the convex field

	Ideal field area	Effective operation area	Skip area	Ratio of effective operation area
Autonomous	1340.9 m^2	1224.9 m^2	116 m^2	91.4%
Manual	1380.3 m^2	1282.0 m^2	98.2 m^2	92.7%

4.3. 요약 및 결론

무인 자율주행 트랙터 구축을 위하여 동양물산기업(주)에서 생산중인 25마력 TS25 트랙터를 대상으로 자율주행 기능을 탑재, 작업이 가능하도록 Layout 설계, 시스템 통합 설계 등을 수행하였다. 주요 부품은 조향장치, 가감속 페달, 작업기제어 레버 등으로 전기식 모터를 이용하여 위치제어 방식을 채용하였다. 제어관련 부품들은 자율주행 제어를 위한 인터페이스 모듈에 연결되고 인터페이스 모듈은 자율주행 제어기로부터 제어명령을 받아 관련 제어하도록 시스템을 구성하였다.

개발된 자율주행 트랙터 시스템의 주행 및 작업 성능을 분석하기 위해 시험평가를 수행하였고 시험 결과는 다음과 같이 요약할 수 있다.

개발한 동적 경로추종 알고리즘의 경로추종 성능 평가하기 위해 비 작업 상태에서 Look-ahead 추종과 비교하였다. R형 선회 패턴 경로에서 동적 경로추종 알고리즘의 RMS 위치 오차는 9.32 cm로 나타났고 이것이 Look-ahead 추종기반의 RMS 위치 오차 18.30 cm 보다 우수한 것으로 판단하였다. 또한 쟁기와 로터리 작업 상태에서 개발한 동적 경로추종 알고리즘의 주행 성능을 평가하기 위해 Look-ahead 경로 추종 알고리즘과 비교 실험을 진행하였다. 동적 경로추종 알고리즘을 사용 시 쟁기 작업과 로터리 작업에서 모두 10 cm 이내의 RMS 위치 오차로 Look-ahead 추종 알고리즘 보다 우수한 성능으로 보였다.

경사지면에서 GPS 신호 보정의 성능 평가를 진행하기 위해 약 9도의 경사지면을 구성하여 트랙터가 기울어진 상태에서 주행 시 GPS 신호 보정 효과를 관찰하였다. 경사지면에서 위치 보정 알고리즘 적용하면 RMS 위치 오차가 약 30 cm에서 2 cm 이내로 감소한 것을 나타났다. 또한 미경지 영역 발생 방지를 위한 작업기 제어의 시험 평가는 C형 선회 패턴 경로를 이용 일정한 속도로 성능평가 하였다. 트랙터가 새머리 영역으로 진입 시 평균 RMS 위치 오차

가 0.91 m에서 0.14 m로 감소하였으며, 새머리 영역에서 탈출 시 평균 RMS 위치 오차가 0.86 m에서 0.13 m로 감소하는 것을 확인하였다.

동적 경로추종 알고리즘을 적용하여 직교형태의 포장에서 자율주행 기반으로 쟁기작업을 실시하였다. 새머리 구역에서 토양의 수분함수량이 상대적으로 높게 나타났으나 RMS 평균 위치 오차는 13.66 cm이며 RMS 평균 방향 오차는 4.91 deg로 정밀 추종하여 농작업이 가능하였다. 주행 시간의 측면에서 포장 작업효율은 78.7%로 나타났고 사람의 수준(64%)이상으로 나타났다. 계측 시스템을 이용하여 위치별 견인력과 연료소모량을 측정하여 위치 별 토양 경도 작업 수준 평가를 진행이 가능하고 위치 별 토양 경도 특성 정보를 제공가능 했다.

사다리와 “L”자 형태의 포장에서 생성된 경로를 따라 각각 포장에서 자율주행 작업이 가능하였으며 자율주행의 포장 작업효율이 숙련된 트랙터 운전자의 수준으로 도달하는 것으로 판단되었다. 작업기의 작업 위치를 기반으로 포장 작업 영역을 분석하였으며 각각 형태의 포장에서 모두 90% 이상의 포장 작업 면적 비율을 나타나 미경지 영역 발생에 대한 문제를 해결되었다.

5. 종합 요약 및 결론

본 연구에서는 농업용 트랙터에 무인자율주행을 위한 각종 제어 및 조작 장치를 탑재하고 다양한 형태의 포장형상에서 무인 자율로 농작업이 가능하도록 작업경로 자동 생성과 추종 제어 시스템을 개발하고자 하였다. 이를 위해 직교형, 사다리형, 삼각형 등 다양한 형상의 포장 내에서 경운작업을 수행하고 자동선회가 가능한 경로생성 알고리즘과 주행 중 발생하는 노면의 슬립정도를 측정하면서 작업경로를 최적으로 추종하는 강인 조향제어기술을 개발하였으며, 가상현실에서 무인 농작업이 가능한 시뮬레이터를 이용하여 개발한 알고리즘을 검증하였으며 최종적으로는 개발 기술을 실제 농용트랙터에 탑재하여 포장에서 무인 경운작업을 수행하여 작업 성능과 효율 등을 숙련된 작업자가 수행한 결과와 비교 개발기술의 효용성을 평가하였다.

5.1. 농지형태에 따른 주행 경로생성 알고리즘 개발

농지형태에 따른 주행 경로생성 알고리즘을 적용하여 자율주행 트랙터 농작업을 수행하기 위한 경로생성 프로그램을 개발하였다.

내부왕복작업부와 외부작업부로를 구분하기 위해 포장 외부 작업 경계선과 내부 작업 경계선을 생성하였다. MBR 원리 기반으로 왕복 직진 작업 경로를 생성하였으며 사용자가 설정된 주행 방향을 이용 MBR 범위 내에서 평행선을 생성한 후 트랙터 직진 주행의 시작점과 종료점으로 도출하였다. 포장 내부 경계선을 생성하기 위해 포장 외부 작업부에서 첫 번째 회경 경로와 포장 외부 경계선간의 거리는 유효 작업 폭 절반 (w/2)로 유지하여 생성하고 나머지 회경 경로는 유효 작업 폭(w) 만큼의 거리를 유지하여 순차적으로 생성하는 방식을 사용하였다. 볼록 형태의 포장에서는 구획작업이 필요 없고 오목형태의 포장에서는 셀 분할 작업을 진행하였다. 본 연구에서는 주어진 포장이 오

목형태이면 경로생성 전에 영역을 분할하였다.

개발한 경로생성 프로그램에서 포장의 좌표정보와 트랙터 및 작업기의 주요 수치를 입력하는 방식이며 생성된 작업경로 데이터를 이용하여 사용자가 경로를 사전에 확인이 가능하게 하고 관련 주요 성능인 예상 포장 작업시간, 포장 작업거리, 유효 작업시간, 유효 작업거리 및 포장 작업효율 등을 함께 출력하여 경로생성의 정량적 정보를 확인할 수 있도록 하였다.

숙련된 트랙터 작업자는 포장에서 작업 시 환경 및 경제 측면을 고려하여 포장의 등고선의 방향으로 주행하는 것이 원칙적이다. 향후 3D 지형을 고려한 경로생성 기술에 대한 연구가 이루어지면 농작업 시 3D 환경으로 인해 발생한 미경지 영역을 최소화 가능 하는 것으로 판단되며 자율주행 트랙터 경로생성 구성을 위해서 트랙터 포장외부에서 트랙터 작업 시작점까지의 진입경로를 추가하는 것을 고려하여 포장 전역 경로생성에 대해 계속적으로 연구가 이루어져야 할 것으로 사료된다.

5.2. 동적 경로추종 기술 개발 및 시뮬레이션 분석

실제 포장에서 자율주행 농작업을 수행하기 전에 개발한 경로생성과 경로추종 알고리즘의 유효성과 작업 특성을 분석하기 위해 컴퓨터 시뮬레이터를 개발하였다. 트랙터가 실제 포장에서 토양의 반력에 인해 회전 주행 시 슬립현상이 발생하기 때문에 이러한 현상을 시뮬레이션에서 반영하기 위해 EKBM을 사용하였다. Cf, Cr 값을 변경하면서 시뮬레이션에서 슬립의 효과를 부가 가능하기 때문에 적용한 동적 설계된 주행 슬립 예측기반의 동적 경로추종 제어기를 시뮬레이터 상에서 주행 성능 평가를 진행하였다. 동적 경로추종 알고리즘의 주행 RMS 위치 오차는 5.44 cm로 더 작게 나타났으며 skip 면적이 6.5 m^2 감소하는 것으로 나타났다.

경사지 혹은 불규칙한 노면은 사용자가 비트맵 형태로 3차원 지도를 작성

후 시뮬레이터에서 3D 지형을 구성 가능하도록 하였다. 개발한 시뮬레이터에서 3D 환경지면이 구성 가능하기 때문에 GPS 신호보정 알고리즘을 적용하여 주행 성능평가를 진행하였으며 복잡한 지면환경에서 효율적으로 GPS 신호 보정 효과를 분석하였다. 좌/우 15도의 경사지면에서 각각 목표 경로와 30 cm / 20 cm 이내의 오차로 보여 알고리즘 적용 전 보다 더 빨리 감소하여 안정된 주행 상태로 나타났다.

개발한 시뮬레이터는 시간 기반의 포장 작업효율 분석이 가능하며 면적 기반의 작업면적 비율도 함께 사용자에게 제공한다. 개발한 자율주행 가상 시뮬레이터에서 작업기 제어 및 작업영역이 표시 가능하기 때문에 개발한 미경지 영역을 방지를 위한 작업기 제어 알고리즘을 평가하였다. 제어 알고리즘 적용하면 새머리 영역으로 진입과 탈출 시 RMS 위치 오차가 모두 30 cm 이내로 감소하는 것을 확인할 수 있다.

향후는 가상 시뮬레이터의 트랙터 조향 성능이 실제와 유사한 동작하기 위해 실제 트랙터의 조향 동특성을 모델링하여 시뮬레이터에 반영하는 것이 필요하다.

5.3. 자율주행 트랙터 통합 시스템 개발

무인 자율주행 트랙터 구축을 위하여 TS25 모델 트랙터를 대상으로 자율주행 기능을 탑재, 작업이 가능하도록 Layout 설계, 시스템 통합 설계 등을 수행하였다. 주요 부품은 조향장치, 가감속 페달, 작업기제어 레버 등으로 전기식 모터를 이용하여 위치제어 방식을 채용하였다. 제어관련 부품들은 자율주행 제어를 위한 인터페이스 모듈에 연결되고 인터페이스 모듈은 자율주행 제어기로부터 제어명령을 받아 관련 제어하도록 시스템을 구성하였다.

동적 경로추종 알고리즘의 요인 성능을 평가하기 위해 쟁기와 로터리 작업의 성능 평가를 하였다. 동적 경로추종 알고리즘은 기존 Look-ahead 추종방

법 보다 더 정밀하게 추종하는 것을 확인하였으며 농작업에 적용 적합성이 높은 것으로 판단하였다.

동적 경로추종 알고리즘을 적용하여 직교형태의 포장에서 자율주행 기반으로 쟁기작업을 실시하였다. 주행 RMS 평균 위치 오차는 13.66 cm이며 RMS 평균 방향 오차는 4.91 deg로 정밀 추종하여 농작업이 가능하였다. 포장 작업 효율은 78.7%로 나타났고 사람의 수준(64%)이상으로 나타났다. 계측시스템을 이용하여 위치별 견인력과 연료소모량을 측정하여 위치 별 토양 경도 작업 수준 평가를 진행이 가능하고 위치 별 토양 경도 특성 정보를 제공가능 했다.

사다리와 “L”자 형태의 포장에서 생성된 경로를 따라 각각 포장에서 자율주행 작업이 가능하였으며 자율주행의 포장 작업효율이 숙련된 트랙터 운전자의 수준으로 도달하는 것으로 판단되었다. 작업기의 작업 위치를 기반으로 포장 작업 영역을 분석하였으며 각각 형태의 포장에서 모두 90% 이상의 포장 작업 면적 비율을 나타나 작업기 제어 알고리즘 적용이 미경지 영역 발생에 대한 문제를 해결되었다.

향후 보완 연구로서 자율주행 트랙터를 이용하여 경운 작업 시 토양조건으로 인하여 과도한 부하가 발생 시 작업기를 정밀 상/하강 조작하거나 트랙터의 주행속도를 제어하는 등 트랙터와 작업기의 강인제어기술 개발이 요구되며 경사지 조건의 밭포장에서 작업을 수행 개발 기술의 경사지 무인 농작업 적용성을 구명하는 것이 필요할 것으로 판단된다.

6. 참고 문헌

1. ASAE S495.1 Nov2005, Uniform Terminology for Agricultural Machinery Management.

2. Acar, E. U., H. Choset, Y. Zhang, and M. Schervish. 2003. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. The International Journal of Robotics Research 22(7-8):441-466.

3. Alvarez, A., A. Caiti, and R. Onken. 2004. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. Oceanic Engineering, IEEE Journal of 29(2):418-429.

4. Backman, J., T. Oksanen, and A. Visala. 2012. Navigation system for agricultural machines: nonlinear model predictive path tracking. Computers and Electronics in Agriculture 82:32-43.

5. Baffet, G., A. Charara, and D. Lechner. 2009. Estimation of vehicle sideslip, tire force and wheel cornering stiffness. Control Engineering Practice 17(11):1255-1264.

6. Barawid, O. C., A. Mizushima, K. Ishii, and N. Noguchi. 2007. Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application. Biosystems Engineering 96(2):139-149.

7. Bayar, G., M. Bergerman, A. B. Koku, and E. ilhan Konukseven. 2015. Localization and control of an autonomous orchard vehicle. Computers and Electronics in Agriculture 115:118-128.

8. Bayar, G., M. Bergerman, and A. B. Koku. 2016. Improving the

trajectory tracking performance of autonomous orchard vehicles using wheel slip compensation. Biosystems Engineering.

9. Bochtis, D., and S. Vougioukas. 2008. Minimising the non-working distance travelled by machines operating in a headland field pattern. Biosystems Engineering 101(1):1-12.

10. Bochtis, D., S. Vougioukas, and H. W. Griepentrog. 2009. A mission planner for an autonomous tractor. Transactions of the ASABE 52(5):1429-1440.

11. Bochtis, D., C. Sørensen, and S. Vougioukas. 2010. Path planning for in-field navigation-aiding of service units. Computers and Electronics in Agriculture 74(1):80-90.

12. Cariou, C., R. Lenain, B. Thuilot, and M. Berducat. 2009. Automatic guidance of a four wheel steering mobile robot for accurate field operations. Journal of Field Robotics 26(67):504-518.

13. Cho, S., and N. Ki. 1999. Autonomous speed sprayer guidance using machine vision and fuzzy logic. Transactions of the ASAE 42(4):1137-1143.

14. Choset, H., and P. Pignon. 1998. Coverage path planning: The boustrophedon cellular decomposition. In Field and Service Robotics. Springer.

15. Choudhury, S., and S. Singh. 2009. A Coverage Planning Algorithm for Agricultural Robots. Technical Report, IIT Kharagpur, India.

16. Dix, P. J., B. Ni, and K. Moaveninejad. 2010. Method and apparatus for creating curved swath patterns for farm machinery. U.S. Patent No. 7,689,356.

17. El-Rabbany, A. 2002. Introduction to GPS: the global positioning

system. Artech House.
18. Fang, H. 2004. Automatic guidance of farm vehicles in presence of sliding effects. A scientific report for the post-doc research project of all-terrain autonomous vehicle control, Le LASMEA.
19. Fang, H., L. Dou, J. Chen, R. Lenain, B. Thuilot, and P. Martinet. 2011. Robust anti-sliding control of autonomous vehicles in presence of lateral disturbances. Control Engineering Practice 19(5):468-478.
20. Fujimoto, Y., H. Araki, and Y. Uoya. 2016. Work Vehicle Cooperation System. US Patent 20,160,120,095.
21. Godoy, E. P., G. T. Tangerino, R. A. Tabile, R. Y. Inamasu, and A. J. V. Porto. 2012. Networked control system for the guidance of a four-wheel steering agricultural robotic platform. Journal of Control Science and Engineering 2012:4.
22. Gomez-Gil, J., S. Alonso-Garcia, F. J. Gomez-Gil, and T. Stombaugh. 2011. A simple method to improve autonomous GPS positioning for tractors. Sensors (Basel) 11(6):5630-5644.
23. Gonzalez-de-Soto, M., L. Emmi, I. Garcia, and P. Gonzalez-de-Santos. 2015. Reducing fuel consumption in weed and pest control using robotic tractors. Computers and Electronics in Agriculture 114:96-113.
24. Gray, S. A., S. L. Hansen, and N. S. Flann. 2006. Path planner and a method for planning a path of a work vehicle. Google Patents. U.S. Patent No. 7,010,425.
25. Hameed, I., D. Bochtis, C. Sørensen, and M. Nøremark. 2010. Automated generation of guidance lines for operational field planning. Biosystems Engineering 107(4):294-306.

26. Hameed, I., D. Bochtis, and C. Sorensen. 2011. Driving angle and track sequence optimization for operational path planning using genetic algorithms. Applied Engineering in Agriculture 27(6):1077-1086.
27. Hameed, I., D. Bochtis, C. Sørensen, and S. Vougioukas. 2012. An object-oriented model for simulating agricultural in-field machinery activities. Computers and Electronics in Agriculture 81:24-32.
28. Hameed, I., D. Bochtis, and C. A. G. Sørensen. 2013. An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas. International Journal of Advanced Robotic Systems 10(231):1-9.
29. Hameed, I., D. Bochtis, C. G. Sørensen, A. L. Jensen, and R. Larsen. 2013. Optimized driving direction based on a three-dimensional field representation. Computers and Electronics in Agriculture 91:145-153.
30. Hameed, I., A. la Cour-Harbo, and O. Osen. 2016. Side-to-side 3D coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths. Robotics and Autonomous Systems 76:36-45.
31. Han, X. Z., H. J. Kim, J. Y. Kim, S. Y. Yi, H. C. Moon, J. H. Kim, and Y. J. Kim. 2015. Path-tracking simulation and field tests for an auto-guidance tillage tractor for a paddy field. Computers and Electronics in Agriculture 112:161-171.
32. Jensen, M. A. F., D. Bochtis, C. G. Sørensen, M. R. Blas, and K. L. Lykkegaard. 2012. In-field and inter-field path planning for agricultural transport units. Computers & Industrial Engineering 63(4):1054-1061.
33. Jin, J., and L. Tang. 2010. Optimal coverage path planning for arable

farming on 2D surfaces. Transactions of the ASABE 53(1):283-295.

34. Jin, J., and L. Tang. 2011. Coverage path planning on three dimensional terrain for arable farming. Journal of Field Robotics 28(3):424-440.

35. Jung, D., & Tsiotras, P. (2007). Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle. AIAA Infotech at Aerospace, AIAA, 07-2763.

36. Kayacan, E., E. Kayacan, H. Ramon, and W. Saeys. 2014a. Distributed nonlinear model predictive control of an autonomous tractor - trailer system. Mechatronics 24(8):926-933.

37. Kayacan, E., E. Kayacan, H. Ramon, and W. Saeys. 2014b. Nonlinear modeling and identification of an autonomous tractor - trailer system. Computers and Electronics in Agriculture 106:1-10.

38. Kayacan, E., E. Kayacan, H. Ramon, and W. Saeys. 2015. Towards agrobots: Identification of the yaw dynamics and trajectory tracking of an autonomous tractor. Computers and Electronics in Agriculture 115:78-87.

39. Kondekar, R. 2013. Method and system for determining a planned path of a vehicle. U.S. Patent No. 8,498,788.

40. Kraus, T., H. J. Ferreau, E. Kayacan, H. Ramon, J. De Baerdemaeker, M. Diehl, and W. Saeys. 2013. Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles. Computers and Electronics in Agriculture 98:25-33.

41. Landau, H., U. Vollath, and X. Chen. 2009. Virtual reference station systems. Journal of Global Positioning System 1(2):137-143.

42. Lenain, R., B. Thuilot, C. Cariou, and P. Martinet. 2006. Mobile robot

control in presence of sliding: Application to agricultural vehicle path tracking. In Decision and Control, 2006 45th IEEE Conference on. IEEE. p.6004-6009.

43. Lenain, R., B. Thuilot, C. Cariou, and P. Martinet. 2006. Sideslip angles observer for vehicle guidance in sliding conditions: Application to agricultural path tracking tasks. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on. IEEE. p.3183-3188.

44. Lenain, R., B. Thuilot, C. Cariou, and P. Martinet. 2006. High accuracy path tracking for vehicles in presence of sliding: Application to farm vehicle automatic guidance for agricultural tasks. Autonomous Robots 21(1):79-97.

45. Lenain, R., B. Thuilot, C. Cariou, and P. Martinet. 2007. Adaptive and predictive path tracking control for off-road mobile robots. European journal of control 13(4):419-439.

46. Lenain, R., B. Thuilot, O. Hach, and P. Martinet. 2011. High-speed mobile robot control in off-road conditions: a multi-model based adaptive approach. In Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE. p.6143-6149.

47. Letherwood, M. D., & Gunter, D. D. (2001). Ground vehicle modeling and simulation of military vehicles using high performance computing. Parallel Computing, 27(1), 109-140.

48. Liang, J. 2011. A path planning algorithm of mobile robot in known 3D environment. Procedia Engineering 15:157-162.

49. Liangliang, Y., and N. NOGUCHI. 2014. Development of a Wheel-Type

Robot Tractor and its Utilization. IFAC Proceedings Volumes 47(3):11571-11576.

50. Linker, R., and T. Blass. 2008. Path-planning algorithm for vehicles operating in orchards. Biosystems Engineering 101(2):152-160.

51. Liu, Y., and N. Noguchi. 2016. Development of an unmanned surface vehicle for autonomous navigation in a paddy field. Engineering in Agriculture, Environment and Food 9(1):21-26.

52. Lucet, E., R. Lenain, and C. Grand. 2015. Dynamic path tracking control of a vehicle on slippery terrain. Control Engineering Practice 42:60-73.

53. Matsuo, Y., O. Yukumoto, and N. Noguchi. 2012. Enhanced Adaptability of Tilling Robot (Initial Report). Japan Agricultural Research Quarterly: JARQ 46(4):295-303.

54. Mailer, R. L. 2005. Vehicle positioning apparatus and method. U.S. Patent No. 6,876,920.

55. Matikainen, V., J. Backman, and A. Visala. 2014. Semi-automatic coupling of an agricultural tractor and a trailer. IFAC Proceedings Volumes 47(3):11583-11588.

56. Matveev, A. S., M. Hoy, J. Katupitiya, and A. V. Savkin. 2013. Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation. Robotics and Autonomous Systems 61(9):973-987.

57. McClure, J. A. 2014. System and method for integrating automatic electrical steering with GNSS guidance. U.S. Patent No. 8,781,685.

58. Motz, D. S., and M. D. Staub. 2001. System and method for planning the operations of an agricultural machine in a field. U.S. Patent No.

6,236,924.

59. Noguchi, N., J. F. Reid, Q. Zhang, J. D. Will, and K. Ishii. 2001. Development of robot tractor based on RTK-GPS and gyroscope. 2001 ASAE Annual International Meeting, Paper Number:01-1195.

60. Park, G., Y. Chang, J. Ryu, S. Jeong, H. Song, S. Park, C. Lee, & S. Hong. 2003. Estimation of vehicle cornering stiffness via GPS/INS. ICCAS 2003:1706-1709.

61. Samson, C. 1995. Control of chained systems application to path following and time-varying point-stabilization of mobile robots. IEEE transactions on Automatic Control 40(1):64-77.

62. Soltani, A. R., H.Tawfik, J. Y. Goulermas, and T. Fernando. 2002. Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms. Advanced Engineering Informatics 16(4):291-303.

63. Song, M.-Z., S.-W. Kang, S.-O. Chung, K.-D. Kim, Y.-S. Chae, D.-H. Lee, Y.-J. Kim, S.-H. Yu, and K.-H. Lee. 2013. Development of Path Planning Algorithm for an Autonomous Mower Tractor. IFAC Proceedings Volumes 46(18):154-158.

64. Stephant, J., A. Charara, and D. Meizel. 2004. Virtual sensor: application to vehicle sideslip angle and transversal forces. Industrial Electronics, IEEE Transactions on 51(2):278-289.

65. Taïx, M., P. Souères, H. Frayssinet, and L. Cordesses. 2006. Path planning for complete coverage with agricultural machines. In Field and service robotics. Springer.

66. Thanpattranon, P., T. Ahamed, and T. Takigawa. 2016. Navigation of

autonomous tractor for orchards and plantations using a laser range finder: Automatic control of trailer position with tractor. Biosystems Engineering 147:90-103.

67. Tuncer, A., and M. Yildirim. 2012. Dynamic path planning of mobile robots with improved genetic algorithm. Computers & Electrical Engineering 38(6):1564-1572.

68. Volger, T., L. Grandl, and B. Kieninger. 2015. Path planning method for vehicle guidance. US Patent 20,150,331,423.

69. Vougioukas, S. G. 2012. A distributed control framework for motion coordination of teams of autonomous agricultural vehicles. Biosystems Engineering 113(3):284-297.

70. Wang, X., J. Taghia, and J. Katupitiya. 2016. Robust Model Predictive Control for Path Tracking of a Tracked Vehicle with a Steerable Trailer in the Presence of Slip. IFAC-Papers OnLine 49(16):469-474.

71. Zakaria, M. A., H. Zamzuri, R. Mamat, and S. A. Mazlan. 2013. A Path Tracking Algorithm Using Future Prediction Control with Spike Detection for an Autonomous Vehicle Robot. International Journal of Advanced Robotic Systems 10.

72. Zhang, Q., and H. Qiu. 2004. A dynamic path search algorithm for tractor automatic navigation. Transactions of the ASAE 47(2):639.

73. Zhang, C., L. Yang, and N. Noguchi. 2014. Development of a Human-driven tractor following a Robot System. IFAC Proceedings Volumes 47(3):11559-11564.

74. Zhang, C., L. Yang, and N. Noguchi. 2015. Development of a robot tractor controlled by a human-driven tractor system. Engineering in

Agriculture, Environment and Food 8(1):7-12.

75. Zhang, C., N. Noguchi, and L. Yang. 2016. Leader-follower system using two robot tractors to improve work efficiency. Computers and Electronics in Agriculture 121:269-281.

76. Zhou, K., A. L. Jensen, C. G. Sørensen, P. Busato, and D. Bothtis. 2014. Agricultural operations planning in fields with multiple obstacle areas. Computers and Electronics in Agriculture 109:12-22.

77. Zhu, Z. X., J. Chen, T. Yoshida, R. Torisu, Z. H. Song, and E.-r. Mao. 2007. Path tracking control of autonomous agricultural mobile robots. Journal of Zhejiang University SCIENCE A8 (10):1596-1603.

78. 김조환, 김만석, 최민구, 김종욱. 2011. uDEAS 를 이용한 이동 로봇의 최적 전역 경로 계획. 한국지능시스템학회 논문지 21(2):268-275.

79. 김상철. 2011. 농업용 로봇 플랫폼의 자율주행 시스템 연구. 공주대학교 농업기계공학 박사학위논문.

80. 김학규. 1995. 大區劃 圃場의 벼農事 一貫 機械化體系에 關한 硏究. 서울대학교 농업기계전공 박사학위논문.

81. 기노훈, 조성인, 최창현. 1996. 기계시각과 퍼지논리를 이용한 스피드스프레이어의 자율주행 (II)-실제 주행. 한국농업기계학회 학술발표논문집 21(2):175-181.

82. 권대원. 1995. 우리나라 측지계와 WGS84의 좌표변환. 성균관대학교 토목공학 석사학위논문.

83. 농촌진흥청. 2008. 트랙터 무인 경운을 위한 작업경로의 생성과 성능 평가 기술 개발. 과제 완결보고서.

84. 농촌진흥청. 2009. 농업기계 이용과 정비기술.

85. 동양물산기업(주). 2013. 자율주행 농용트랙터를 위한 지능형 운전제어시스

템 개발. 최종보고서.
86. 류명진, 조명진, 송명장, 정선옥, 이경환, 유승화, 김용주. 2013. 모워 트랙터 자율주행 경로 생성을 위한 관행 작업 경로 분석. 한국농업기계학회 학술발표논문집 18(2):75-76.
87. 문희창. 2008. 무인자율주행차량을 위한 분산제어시스템 설계 및 자율주행 알고리즘 개발. 국민대학교 자동차공학전문대학원 박사학위논문.
88. 서일환, 김찬수, 오종진, 조성찬, 이영규, 김기대. 2007. 트랙터의 무인경운을 위한 작업경로 생성과 성능평가 기술개발(3)-작업경로 생성을 위한 알고리즘 및 프로그램 개발-.한국농업기계학회 하계학술대회논문집 12(2):364-372.
89. 서일환, 이영규, 김찬수, 조성찬, 나우정, 조종승, 김기대. 2008. 트랙터의 무인경운을 위한 작업경로 생성과 성능평가 기술개발(5)-경로생성프로그램의 성능향상-.한국농업기계학회 동계학술대회논문집13(1):456-462.
90. 서일환. 2010. 자율주행 트랙터 경운경로 생성시스템. 충남대학교 농업기계공학 박사학위논문.
91. 서일환, 서동현, 김기대. 2010. 자율 주행 트랙터 경운경로생성 프로그램 개발. 농업과학연구 37(1):113-121.
92. 서동현, 윤상진, 김찬수, 임중호, 하종규, 손대원, 김기대. 2008. 자율주행 트랙터의 성능 향상을 위한 연구(1)-자편각(磁偏角)이 직진주행성능에 미치는 영향. 한국농업기계학회 학술대회논문집 13(1):468-473.
93. 서동현, 서일환, 정선옥, 김기대. 2010. 자율 주행 트랙터 시스템의 성능 향상을 위한 CAN 기반의 조향제어시스템 개발. 농업과학연구 37(1):123-130.
94. 서동현. 2010. Network RTK GPS와 CAN-bus 基盤의 操向 制御를 利用한 트랙터 自律 走行 시스템 開發. 충남대학교 생물생산기계공학 박사학위논문.
95. 손영진. 2009. 무인자율주행차량을 위한 경로생성 방법 및 위치추정 방법

에 대한 연구. 국민대학교 자동차공학전문대학원 박사학위논문.
96. 신범수, 김기동. 2016. 무인 농작업기 작업경로 생성 알고리즘 개발. 한국생산제조학회 학술발표대회 논문집:152-152.
97. 이재훈, 조성인, 이정엽. 1998. DGPS와 퍼지제어를 이용한 스피드스프레이어의 자율주행. 한국농업기계학회지 23(1):75-82.
98. 우훈제. 2011. 다수 무인 차량을 위한 시스템 아키텍처 설계 및 차량 간 선행 경로 추종 구현. 국민대학교 자동차공학전문대학원 박사학위논문.
99. 정선옥, 박원규, 장영창, 이동현, 박우풍. 1999. 소구획 경지에서의 벼 수확량 지도 작성. 한국농업기계학회지 24(2):135-144.88.
100. 정인규, 박우풍, 김상철, 정선옥, 황성준. 1999. 자율주행 트랙터의 장애물 검출 및 회피알고리즘 개발. 한국농업기계학회 동계학술대회 논문집 5(1):554-559.
101. 조성인, 기노훈. 1996. 기계시각과 퍼지논리를 이용한 스피드스프레이어의 자율주행(1)-그래픽시뮬레이션.한국농업기계학회학술발표논문집 21(2):167-174.
102. 조성인, 이정엽, 박영식. 1999. DGPS와 GIS를 이용한 스피드스프레이어의 자율주행. 한국농업기계학회 동계학술대회논문집 4(1):698-703.
103. 조성인, 박영식, 최창현, 황헌, 김명락. 2001. DGPS와 기계시각을 이용한 자율주행 콤바인의 개발. 한국농업기계학회 학술발표논문집 26(1):29-38.
104. 한국에스리. 2005. 세계측지계와 ArcGIS를 이용한 좌표변환. 기술 노트.
105. 한웅철. 2013. 자율경운 농용트랙터를 위한 경로생성 및 경로추종 기술 개발. 서울대학교 바이오시스템공학 석사학위논문.

7. 부록 및 기타

7.1. 가상 시뮬레이터의 트랙터 주행 모델 구성 코딩

```
#ifndef __RDEVICETRACTORDRIVE_H__
#define __RDEVICETRACTORDRIVE_H__

#include "rMath/rMath.h"
using namespace rMath;

#include "rDevice/rDeviceBase.h"
#include "../common/HeightMap.h"

class rDeviceTractorDrive : public RD_DEVICE_CLASS(Base)
{
		RD_VERSION(1.0);
		RD_AUTHOR(SimLab);

public:
		RD_DECLARE_CTOR(TractorDrive);
		RD_DECLARE_DTOR(TractorDrive);

public:
		RD_DECLARE_createDevice;
		RD_DECLARE_initDevice;
		RD_DECLARE_terminateDevice;
		RD_DECLARE_writeDeviceValue;
		RD_DECLARE_monitorDeviceValue;
		RD_DECLARE_exportDevice;

		void InitParams();
		void PrintParams();
		void InitHeightMap();

private:
		// base body id
		rID		_nodeid;

		// simulation time
		rTime		_prevTime;
		float		_dT;

		// wheel/steer motor(kinematic) motors
		rID		_front_lwheel;
		rID		_front_rwheel;
		rID		_rear_lwheel;
		rID		_rear_rwheel;
		rID		_steer_lwheel;
		rID		_steer_rwheel;

		// tractor kinematics
		float		_distance_btw_axes;
		float		_front_radius;
		float		_rear_radius;

		// current/desired wheel/steer position
		float		_front_ltheta;
		float		_front_rtheta;
		float		_rear_ltheta;
		float		_rear_rtheta;
		float		_steer_ltheta;
		float		_steer_rtheta;

		// limits
		float		_steer_limit;
		float		_steer_velocity_limit;
		float		_velocity_limit;
		float		_acceleration_limit;

		// desired(commanded) forward velocity and steer angle
		float		_v_des;
		float		_theta_des;
```

```
            // current forward velocity and steer angle (wrt vehicle coordinate frame)
            float       _v;
            float       _theta;

            // current forward and lateral velocity considering slip (wrt vehicle coordinate frame)
            float       _vx; // forward(longitudinal) speed
            float       _vy; // lateral speed

            // current pose of tractor (wrt global coordinate frame)
            float       _psi;   // yaw angle
            float       _x;             // global position x
            float       _y;             // global position y
            float       _z;             // global position z
            float       _pitch; // pitch angle
            float       _roll;  // roll angle

            // initial transform of tractor
            float       _r0[3];     // initial axis displacement
            float       _R0[9];     // initial axis rotation

            // current transformation of tractor
            float       _R[9];
            float       _r[3];

            // model parameters to determine the slip angle, yaw angle, yaw rate
            float       _Cf;        // cornering stiffness of front wheels (N/rad)
            float       _Cr;        // cornering stiffness of rear wheels (N/rad)
            float       _m;         // mass of the tractor (kg)
            float       _Iz;        // turning inertia WRT tractor COG (kg m^2)
            float       _Lf;        // distance from front wheel to tractor COG (m)
            float       _Lr;        // distance from rear wheel to tractor COG (m)

            float       _beta;      // slip angle of front and rear wheels
            float       _betadot;   // slip rate
            float       _betaf;     // slip angle of front wheel
            float       _betar;     // slip angle of rear wheel
            float       _Ff;        // lateral force at front wheel
            float       _Fr;        // lateral force at rear wheel

            // current velocity (wrt global coordinate frame)
            float       _xdot;
            float       _ydot;
            float       _zdot;
            float       _psidot;    // yaw rate
            float       _psiddot;   // yaw angular acceleration

            float       _slip_ratio;

            // minimum forward velocity to apply extended bicycle model with lateral vehicle dynamics
            float       _v_threshod_to_apply_slipage;

            // height map
            HeightMap _hmap;
            int _hmap_normal_window_size;
};

#endif

#include "rDeviceTractorDrive.h"
#include "rDeviceTractorDriveCmd.h"

#define A(i, j)         A_[3*(i) + (j)]
#define B(i, j)         B_[3*(i) + (j)]
#define At(i, j)        A_[3*(j) + (i)]
#define Bt(i, j)        B_[3*(j) + (i)]
#define C(i, j)         C_[3*(i) + (j)]

void MultMat(float* C_, const float* A_, const float* B_)
{
            assert(C_ != A_ && C_ != B_);
            memset(C_, 0, sizeof(float) * 9);
            for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
            for (int k = 0; k < 3; k++)
            C(i, j) += A(i, k) * B(k, j);
            }
            }
```

```
}

void TrMultMat(float* C_, const float* A_, const float* B_)
{
            assert(C_ != A_ && C_ != B_);
            memset(C_, 0, sizeof(float) * 9);
            for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
            for (int k = 0; k < 3; k++)
            C(i, j) += At(i, k) * B(k, j);
            }
            }
}

void MultMatTr(float* C_, const float* A_, const float* B_)
{
            assert(C_ != A_ && C_ != B_);
            memset(C_, 0, sizeof(float) * 9);
            for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
            for (int k = 0; k < 3; k++)
            C(i, j) += A(i, k) * Bt(k, j);
            }
            }
}

void TrMultMatTr(float* C_, const float* A_, const float* B_)
{
            assert(C_ != A_ && C_ != B_);
            memset(C_, 0, sizeof(float) * 9);
            for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
            for (int k = 0; k < 3; k++)
            C(i, j) += At(i, k) * Bt(k, j);
            }
            }
}

void MultVec(float* c, const float* A_, const float* b)
{
            assert(c != b);
            memset(c, 0, sizeof(float) * 3);
            for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++)
            c[i] += A(i, j) * b[j];
            }
}

void TrMultVec(float* c, const float* A_, const float* b)
{
            assert(c != b);
            memset(c, 0, sizeof(float) * 3);
            for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++)
            c[i] += At(i, j) * b[j];
            }
}

void AddVec(float* c, const float* a, const float* b)
{
            for (int i = 0; i < 3; i++)
            c[i] = a[i] + b[i];
}

void SubtractVec(float* c, const float* a, const float* b)
{
            for (int i = 0; i < 3; i++)
            c[i] = a[i] - b[i];
}
//implementation of rDeviceTractorDrive

RD_IMPLE_FACTORY(TractorDrive)

rDeviceTractorDrive::rDeviceTractorDrive()
: _nodeid(INVALID_RID)
, _dT(0.001), _prevTime(0)
, _distance_btw_axes(1)
```

```
, _front_lwheel(INVALID_RID), _front_rwheel(INVALID_RID), _front_radius(1)
, _rear_lwheel(INVALID_RID), _rear_rwheel(INVALID_RID), _rear_radius(1)
, _steer_lwheel(INVALID_RID), _steer_rwheel(INVALID_RID)
, _front_ltheta(0), _front_rtheta(0)
, _rear_ltheta(0), _rear_rtheta(0)
, _steer_ltheta(0), _steer_rtheta(0)
, _steer_limit(-1), _steer_velocity_limit(-1)
, _velocity_limit(-1), _acceleration_limit(-1)
, _v_des(0), _theta_des(0), _v(0), _theta(0)
, _vx(0), _vy(0)
, _x(0), _y(0), _z(0), _psi(0), _pitch(0), _roll(0)
, _xdot(0), _ydot(0), _zdot(0)
, _psidot(0), _psiddot(0)
, _beta(0), _betadot(0)
, _betaf(0), _betar(0)
, _Ff(0), _Fr(0)
, _slip_ratio(0)
, _v_threshod_to_apply_slipage(0.3f)
, _hmap_normal_window_size(1)
{
}

rDeviceTractorDrive::~rDeviceTractorDrive()
{
}

void rDeviceTractorDrive::onCreate(const rDeviceContext& rdc)
{
            RD_DEVICE_CLASS(Base)::onCreate(rdc);
}

void rDeviceTractorDrive::onInit()
{
            RD_DEVICE_CLASS(Base)::onInit();

            // load device parameters
            InitParams();
            PrintParams();

            // get tractor base body
            _nodeid = _rdc.m_deviceAPI->getBodyID(_rdc.m_robotname, _rdc.m_nodename);

            // get initial device frame
            //    R_G_C = R_G_V * R_V_C
            //    r_G_C = R_G_V * r_V_C + r_G_V
            //        where,
            //            G := Global frame
            //            V := Vehicle frame
            //            C := COG of vehicle
            //
            float r0[3];
            float R0[9];
            _rdc.m_deviceAPI->getBodyPosition(_nodeid, r0);
            _rdc.m_deviceAPI->getBodyOrientation(_nodeid, R0);
            MultMat(_R0, R0, R);

            MultVec(_r0, R0, r);
            AddVec(_r0, _r0, r0);

            // initialize vehicle position and orientation
            _x = _r0[0];
            _y = _r0[1];
            _z = _r0[2];
            _psi = atan2(-_R0[1], _R0[0]);

            memset(_r, 0, sizeof(float)*3);
            memset(_R, 0, sizeof(float)*9);
            _R[0] = _R[4] = _R[8] = 1;

            _dT = (float)(period * 1e-6);

            _v = _v_des = 0;
            _theta = _theta_des = 0;
            _vx = _vy = 0;
            _beta = 0;
            _betadot = 0;
            _xdot = 0;
```

```
		_ydot = 0;
		_zdot = 0;
		_psidot = 0;
		_psiddot = 0;

		// initiate height map
		InitHeightMap();
}

void rDeviceTractorDrive::onTerminate()
{
		RD_DEVICE_CLASS(Base)::onTerminate();
}

int rDeviceTractorDrive::writeDeviceValue(void* buffer, int len, int port)
{
		switch (port)
		{
		case TRACTORDRIVE_DATAPORT_SET_POSE:
		{
		if (len >= 3*sizeof(float))
		{
		_lock.lock();
		{
		float *value = (float*)buffer;
		_x = value[0];
		_y = value[1];
		_psi = value[2];

		_v = _v_des = 0;
		_theta = _theta_des = 0;
		_beta = 0;
		_betadot = 0;
		_xdot = 0;
		_ydot = 0;
		_zdot = 0;
		_psidot = 0;
		_psiddot = 0;
		}
		_lock.unlock();
		return 3*sizeof(float);
		}
		}
		break;

		case TRACTORDRIVE_DATAPORT_SET_VEL_IMMEDIATE:
		{
		if (len >= 2*sizeof(float))
		{
		_lock.lock();
		{
		float *value = (float*)buffer;
		_v = _v_des = value[0];           // forward velocity
		_theta = _theta_des = value[1];   // steer angle
		}
		_lock.unlock();
		return 2*sizeof(float);
		}
		}
		break;

		case TRACTORDRIVE_DATAPORT_SET_VEL_TARGET:
		default:
		{
		if (len >= 2*sizeof(float))
		{
		_lock.lock();
		{
		float *value = (float*)buffer;
		_v_des = value[0];     // forward velocity
		_theta_des = value[1]; // steer angle
		}
		_lock.unlock();
		return 2*sizeof(float);
		}
		}
		}
```

```
		return 0;
}

int rDeviceTractorDrive::monitorDeviceValue(void* buffer, int len, int port)
{
		switch (port)
		{
		case TRACTORDRIVE_MONITORPORT_SLIP_ANGLE:
		{
		if (len >= 3*sizeof(float))
		{
		_lock.lock();
		{
		float* value = (float*)buffer;
		value[0] = _beta;
		value[1] = _betaf;
		value[2] = _betar;
		}
		_lock.unlock();
		return 3*sizeof(float);
		}
		}
		break;

		case TRACTORDRIVE_MONITORPORT_LATERAL_FORCE:
		{
		if (len >= 2*sizeof(float))
		{
		_lock.lock();
		{
		float* value = (float*)buffer;
		value[0] = _Ff;
		value[1] = _Fr;
		}
		_lock.unlock();
		return 2*sizeof(float);
		}
		}
		break;

		case TRACTORDRIVE_MONITORPORT_POSE_LOCAL:
		{
		if (len >= 3*sizeof(float))
		{
		_lock.lock();
		{
		float* value = (float*)buffer;
		value[0] = _theta;
		value[1] = _vx;
		value[2] = _vy;
		}
		_lock.unlock();
		return 3*sizeof(float);
		}
		else if (len >= 2*sizeof(float))
				{
		_lock.lock();
		{
		float* value = (float*)buffer;
		value[0] = _theta;
		value[1] = _v;
		}
		_lock.unlock();
		return 2*sizeof(float);
		}
		}
		break;

		case TRACTORDRIVE_MONITORPORT_VELOCITY:
		{
		if (len >= 3*sizeof(float))
		{
		_lock.lock();
		{
		float* value = (float*)buffer;
		value[0] = _xdot;
```

```
		value[1] = _ydot;
		value[2] = _psidot;
		}
		_lock.unlock();
		return 3*sizeof(float);
		}
		}
		break;

		case TRACTORDRIVE_MONITORPORT_POSE:
		default:
		{
		if (len >= 5*sizeof(float))
		{
		float* value = (float*)buffer;
		value[0] = _x;
		value[1] = _y;
		value[2] = _psi;
		value[3] = _pitch;
		value[4] = _roll;
		}
		else if (len >= 3*sizeof(float))
		{
		_lock.lock();
		{
				float* value = (float*)buffer;
		value[0] = _x;
		value[1] = _y;
		value[2] = _psi;
		}
		_lock.unlock();
		return 3*sizeof(float);
		}
		}
		break;
		}
		return 0;
}

void rDeviceTractorDrive::exportDevice(rTime time, void* mem)
{
		_lock.lock();
		{

		_dT = float(time - _prevTime);
		_prevTime = time;
		if (_dT < 1.0e-6)
		{
		_lock.unlock();
		return;
		}

		// calcurate current velocity(_v) considering velocity and acceleration limit
		if (_acceleration_limit > 0)
		{
		float vdot_des = (_v_des - _v) / _dT;
		if (vdot_des > _acceleration_limit)
		vdot_des = _acceleration_limit;
		else if (vdot_des < -_acceleration_limit)
		vdot_des = -_acceleration_limit;
		_v += vdot_des*_dT;
		}
		else
		_v = _v_des;

		if (_velocity_limit > 0)
		_v = RD_BOUND(_v, -_velocity_limit, _velocity_limit);

		// calcurate current steering angle(_theta) considering velocity and position limit
		if (_steer_velocity_limit > 0)
		{
		float thetadot_des = (_theta_des - _theta) / _dT;
		if (thetadot_des > _steer_velocity_limit)
		thetadot_des = _steer_velocity_limit;
		else if (thetadot_des < -_steer_velocity_limit)
		thetadot_des = -_steer_velocity_limit;
		_theta += thetadot_des*_dT;
```

```
}
else
_theta = _theta_des;

if (_steer_limit > 0)
_theta = RD_BOUND(_theta, -_steer_limit, _steer_limit);

//calcurate wheel velocity and angle
float w_front_wheel = _v / _front_radius;
float w_rear_wheel = _v / _rear_radius;

_front_ltheta += w_front_wheel * _dT;
_front_rtheta += w_front_wheel * _dT;

_rear_ltheta += w_rear_wheel * _dT;
_rear_rtheta += w_rear_wheel * _dT;

_steer_ltheta = _theta;
_steer_rtheta = _theta;

//calcurate new position and orientation
if (_v >= _v_threshod_to_apply_slipage)
{
if (_Iz == 0 || _m == 0)
{
_vx = _v;
_vy = 0;

_xdot = _vx * cos(_psi);// - _vy * sin(_psi);
_ydot = _vx * sin(_psi);// + _vy * cos(_psi);
_psidot = _v * tan(_theta) / _distance_btw_axes;

_x += _xdot * _dT;
_y += _ydot * _dT;
//_psi += _psidot * _dT;
_psi += _psidot * _dT * (1.0f - _slip_ratio);
}
else
{
_vx = _v;
_vy = 0;

float A = -2 * (_Cr + _Cf) / (_m * _vx);
float B = -1 - 2 * ((_Cf * _Lf - _Cr * _Lr) / (_m * _vx * _vx));
float C = (2 * _Cf) / (_m * _vx);
float D = -2 * ((_Cf * _Lf - _Cr * _Lr) / _Iz);
float E = -2 * ((_Cf * _Lf * _Lf + _Cr * _Lr * _Lr) / (_Iz * _vx));
float F = (2 * _Cf * _Lf) / _Iz;

_betadot = (A * _beta) + (B * _psidot) + (C * _theta);
_psiddot = (D * _beta) + (E * _psidot) + (F * _theta);

_beta    += _betadot * _dT;
_psidot  += _psiddot * _dT;
_psi     += _psidot  * _dT;

float tbeta = tan(_beta);
//float tbeta = _beta; // tan(th) ~= th when th << 1
_vy = _vx * tbeta;

_xdot = _vx*cos(_psi) - _vy*sin(_psi);
_ydot = _vx*sin(_psi) + _vy*cos(_psi);

_x += _xdot*_dT;
_y += _ydot*_dT;

// slip angles
// from A DYNAMIC PATH SEARCH ALGORITHM FOR TRACTORI AUTOMATIC NAVIGATION
//_betaf = _theta - atan2(_Lf * _psidot + _ydot, _xdot);
//_betar = atan2(_Lr * _psidot - _ydot, _xdot);
          ////////////////////////////////////////////////////////////
// from ROBUST ANTI-SLIDING CONTROL OF AUTONOMOUS VEHICLES IN PRESENCE OF LATERAL DISTURBANCES
//_betaf = _theta - (_Lf*_psidot + _v*tan(_beta)) / _v;
//_betar = (-_Lr*_psidot + _v*tan(_beta)) / _v;
//from LATERAL VEHICLE DYNAMICS (2012 VDCL SNU)
_betaf = _theta - (_Lf*_psidot + _vy) / _vx;
_betar = (-_Lr*_psidot + _vy) / _vx;
```

```
// lateral forces
_Ff = _Cf * _betaf;
_Fr = _Cr * _betar;
}
}
else
{
// for the backward direction, apply normal kinematics model.
_beta = 0.0f;
_betadot = 0.0f;
_betaf = 0.0f;
_betar = 0.0f;
_Ff = 0.0f;
_Fr = 0.0f;

if (abs(_v)  >= 1.0e-4)
{
_vx = _v;
_vy = 0;

_xdot = _vx * cos(_psi);// - _vy * sin(_psi);
_ydot = _vx * sin(_psi);// + _vy * cos(_psi);
_psidot = _v * tan(_theta) / _distance_btw_axes;

_x += _xdot * _dT;
_y += _ydot * _dT;
          //_psi += _psidot * _dT;
_psi += _psidot * _dT * (1.0f - _slip_ratio);
}
else
{
_vx = _vy = 0;
_xdot = 0;
_ydot = 0;
_zdot = 0;
_psidot = 0;
_psiddot = 0;
}
}

// new vehicle(tractor) trasformation
float c = cos(_psi);
float s = sin(_psi);
float height;
_hmap.GetHeight(_x, _y, height);

_R[0] = c;
_R[1] = -s;
_R[3] = s;
_R[4] = c;
_r[0] = _x;
_r[1] = _y;
//_r[2] = _z;
_r[2] = height + _r0[2];

// set model(base body) transformation
//    T_G_V = T_G_C ^ T_V_C^-1
// it leads to,
//    R_G_V = R_G_C * R_V_C^-1
//    r_G_V = -R_G_V * R_V_C^-1 * r_V_C + r_G_C
//        where,
//            G := Global frame
//            V := Vehicle frame
//            C := COG of vehicle
//
float R1[9];
float r1[3], r2[3];
MultMatTr(R1, _R, R);
MultVec(r2, R1, r);
SubtractVec(r1, _r, r2);
//_rdc.m_deviceAPI->setBodyHTransform(_nodeid, R1, r1);
// apply face normal to model(robot) transform (pitch, roll):
float normal[3];  // terrain face normal wrt global frame
float normal2[3]; // terrain face normal wrt robot heading frame (R1_t * normal)
float Ry[9], Rx[9];
```

```
		_hmap.GetNormalEx(_x, _y, _hmap_normal_window_size, normal);

		TrMultVec(normal2, R1, normal);

		_pitch = (float)(M_PI*0.5 - atan2(normal2[2], normal2[0]));
		c = cos(_pitch);
		s = sin(_pitch);
		Ry[0] = c; Ry[1] = 0; Ry[2] = s;
		Ry[3] = 0; Ry[4] = 1; Ry[5] = 0;
		Ry[6] =-s; Ry[7] = 0; Ry[8] = c;

		_roll = -(float)(M_PI*0.5 - atan2(sqrt(pow(normal2[0],2) + pow(normal2[2],2)), normal2[1]));
		c = cos(_roll);
		s = sin(_roll);
		Rx[0] = 1; Rx[1] = 0; Rx[2] = 0;
		Rx[3] = 0; Rx[4] = c; Rx[5] =-s;
		Rx[6] = 0; Rx[7] = s; Rx[8] = c;

		float R1_pitch[9];
		float R1_pitch_roll[9];

		MultMat(R1_pitch, R1, Ry);
		MultMat(R1_pitch_roll, R1_pitch, Rx);
		_rdc.m_deviceAPI->setBodyHTransform(_nodeid, R1_pitch_roll, r1);

		// visualizing rotating wheel and steer angles:
		if (_front_lwheel != INVALID_RID)
		_rdc.m_deviceAPI->setDeviceValue(_front_lwheel, &_front_ltheta, sizeof(float));
		if (_front_rwheel != INVALID_RID)
		_rdc.m_deviceAPI->setDeviceValue(_front_rwheel, &_front_rtheta, sizeof(float));
		if (_rear_lwheel  != INVALID_RID)
		_rdc.m_deviceAPI->setDeviceValue(_rear_lwheel, &_rear_ltheta, sizeof(float));
		if (_rear_rwheel  != INVALID_RID)
		_rdc.m_deviceAPI->setDeviceValue(_rear_rwheel, &_rear_rtheta, sizeof(float));
		if (_steer_lwheel != INVALID_RID)
		_rdc.m_deviceAPI->setDeviceValue(_steer_lwheel, &_steer_ltheta, sizeof(float));
		if (_steer_rtheta != INVALID_RID)
		_rdc.m_deviceAPI->setDeviceValue(_steer_rwheel, &_steer_rtheta, sizeof(float));
		}
		_lock.unlock();
}

void rDeviceTractorDrive::InitParams()
{
		// initialize device parameters
		const TCHAR* distance_btw_axes = getProperty(_T("distance_btw_axes"));
		if (distance_btw_axes)
		_distance_btw_axes = (float)_tstof(distance_btw_axes);

		const TCHAR* front_radius = getProperty(_T("front_radius"));
		if (front_radius)
		_front_radius = (float)_tstof(front_radius);

		const TCHAR* front_lwheel = getProperty(_T("front_lwheel"));
		if (front_lwheel)
		_front_lwheel = _rdc.m_deviceAPI->getDeviceID(_rdc.m_robotname, front_lwheel);

		const TCHAR* front_rwheel = getProperty(_T("front_rwheel"));
		if (front_rwheel)
		_front_rwheel = _rdc.m_deviceAPI->getDeviceID(_rdc.m_robotname, front_rwheel);

		const TCHAR* rear_radius = getProperty(_T("rear_radius"));
		if (rear_radius)
		_rear_radius = (float)_tstof(rear_radius);

		const TCHAR* rear_lwheel = getProperty(_T("rear_lwheel"));
		if (rear_lwheel)
		_rear_lwheel = _rdc.m_deviceAPI->getDeviceID(_rdc.m_robotname, rear_lwheel);

		const TCHAR* rear_rwheel = getProperty(_T("rear_rwheel"));
		if (rear_rwheel)
		_rear_rwheel = _rdc.m_deviceAPI->getDeviceID(_rdc.m_robotname, rear_rwheel);

		const TCHAR* steer_lwheel = getProperty(_T("steer_lwheel"));
		if (steer_lwheel)
		_steer_lwheel = _rdc.m_deviceAPI->getDeviceID(_rdc.m_robotname, steer_lwheel);
```

```
		const TCHAR* steer_rwheel = getProperty(_T("steer_rwheel"));
		if (steer_rwheel)
		_steer_rwheel = _rdc.m_deviceAPI->getDeviceID(_rdc.m_robotname, steer_rwheel);

		const TCHAR* steer_limit = getProperty(_T("steer_limit"));
		if (steer_limit)
		_steer_limit = (float)(_tstof(steer_limit) * DEGREE);

		const TCHAR* steer_velocity_limit = getProperty(_T("steer_velocity_limit"));
		if (steer_velocity_limit)
		_steer_velocity_limit = (float)(_tstof(steer_velocity_limit) * DEGREE);

		const TCHAR* velocity_limit = getProperty(_T("velocity_limit"));
		if (velocity_limit)
		_velocity_limit = (float)_tstof(velocity_limit);

		const TCHAR* acceleration_limit = getProperty(_T("acceleration_limit"));
		if (acceleration_limit)
		_acceleration_limit = (float)_tstof(acceleration_limit);

		const TCHAR* Cf = getProperty(_T("Cf"));
		if (Cf)
		_Cf = (float)_tstof(Cf);

		const TCHAR* Cr = getProperty(_T("Cr"));
		if (Cr)
		_Cr = (float)_tstof(Cr);

		const TCHAR* Iz = getProperty(_T("Iz"));
		if (Iz)
		_Iz = (float)_tstof(Iz);

		const TCHAR* m = getProperty(_T("m"));
		if (m)
		_m  = (float)_tstof(m);

		const TCHAR* Lf = getProperty(_T("Lf"));
		if (Lf)
		_Lf = (float)_tstof(Lf);

		const TCHAR* Lr = getProperty(_T("Lr"));
		if (Lr)
		_Lr = (float)_tstof(Lr);

		const TCHAR* slip_ratio = getProperty(_T("slip_ratio"));
		if (slip_ratio)
		_slip_ratio = (float)RD_BOUND(_tstof(slip_ratio), 0, 1.0);

		const TCHAR* v_threshold = getProperty(_T("velocity_threshold_to_apply_slipage"));
		if (v_threshold)
		_v_threshod_to_apply_slipage = (float)RD_LBOUND(_tstof(v_threshold), 1.0e-4);
}

void rDeviceTractorDrive::InitHeightMap()
{
		const TCHAR* prop;
		string_type hmap_path = _T("");
		float hmap_map_width = 0.0f;
		float hmap_map_length = 0.0f;
		float hmap_map_height_min = 0.0f;
		float hmap_map_height_max = 0.0f;

		prop = getProperty(_T("HMAP_PATH")); if (prop) hmap_path = prop;
		prop = getProperty(_T("HMAP_W")); if (prop) hmap_map_width = (float)_tstof(prop);
		prop = getProperty(_T("HMAP_L")); if (prop) hmap_map_length = (float)_tstof(prop);
		prop = getProperty(_T("HMAP_H_min")); if (prop) hmap_map_height_min = (float)_tstof(prop);
		prop = getProperty(_T("HMAP_H_max")); if (prop) hmap_map_height_max = (float)_tstof(prop);
		prop = getProperty(_T("HMAP_NORMAL_WND_SIZE")); if (prop) _hmap_normal_window_size = _tstoi(prop);

		if (!_hmap.Create(hmap_path, hmap_map_width, hmap_map_length, hmap_map_height_min,
hmap_map_height_max))
		{
		assert(0 && "ERROR! rDeviceTractorDrive: failed to load height map.₩n");
		}
}

void rDeviceTractorDrive::PrintParams()
```

```
{
        printf("********** rDeviceTractorDrive ********₩n");
        printf("₩n₩t== VEHICLE KINEMATICS ==₩n₩n");
        printf("Distance between front and rear axis: ₩t%.3f ₩t(m)₩n", _distance_btw_axes);
        printf("Radius of front wheel: ₩t%.3f ₩t(m)₩n", _front_radius);
        printf("Radius of rear wheel: ₩t%.3f ₩t(m)₩n", _rear_radius);
        printf("Steering angle limit: ₩t%.3f ₩t(degree)₩n", _steer_limit*RADIAN);
        printf("Steering velocity limit: ₩t%.3f ₩t(degree/s)₩n", _steer_velocity_limit*RADIAN);
        printf("Longitudinal velocity limit: ₩t%.3f ₩t(m/s)₩n", _velocity_limit);
        printf("Longitudinal acceleration limit: ₩t%.3f ₩t(m/s^2)₩n", _acceleration_limit);
        printf("₩n₩t== VEHICLE DYNAMICS ==₩n₩n");
        printf("Cornering stiffness of front wheels(Cf): ₩t%.3f ₩t(N/rad)₩n", _Cf);
        printf("Cornering stiffness of rear wheels(Cr): ₩t%.3f ₩t(N/rad)₩n", _Cr);
        printf("Mass of the tractor(m): ₩t%.3f ₩t(kg)₩n", _m);
        printf("Turning inertia WRT tractor COG(Iz): ₩t%.3f ₩t(kg m^2)₩n", _Iz);
        printf("Distance from front wheel to tractor COG(Lf): ₩t%.3f ₩t(m)₩n", _Lf);
        printf("Distance from rear wheel to tractor COG(Lr): ₩t%.3f ₩t(m)₩n", _Lr);
        printf("₩n");
        printf("₩n");
        printf("**************************************₩n");
}
```

7.2. 가상 시뮬레이터의 3D 지형 환경 구성 코딩

```
#ifndef __rlab_HEIGHTMAP_H__
#define __rlab_HEIGHTMAP_H__

#include "rType.h"

class HeightMap
{
public:
		HeightMap();
		~HeightMap();

		bool Create(const string_type& path, float width, float length, float height_min, float height_max);
		bool GetHeight(float x, float y, float& height);
		bool GetNormal(float x, float y, float normal[3]);
		bool GetNormalEx(float x, float y, int wnd_size, float normal[3]);

private:
		string_type _path;
		float	_map_width;
		float	_map_length;
		float	_map_height_min;
		float	_map_height_max;
		float	_map_width_scale;
		float	_map_length_scale;
		float	_map_height_scale;
		float*	_heights;  // vertex height(z)
		float*	_normals;  // vertex normal
		int		_bmp_pixel_width;
		int		_bmp_pixel_length;
};

#endif

#include "HeightMap.h"
#include "rMath/rMath.h"
#include <windows.h>
#include <assert.h>
#include <math.h>

#define ALIGNEDWIDTHBYTES(bits) (((bits)+31)/32*4)

HeightMap::HeightMap()
: _heights(NULL)
, _normals(NULL)
, _path(_T(""))
, _map_width(0.0f)
, _map_length(0.0f)
, _map_height_min(0.0f)
, _map_height_max(0.0f)
, _map_width_scale(1.0f)
, _map_length_scale(1.0f)
, _map_height_scale(1.0f)
, _bmp_pixel_width(0)
, _bmp_pixel_length(0)
{
}

HeightMap::~HeightMap()
{
		if (_heights) delete[] _heights;
		if (_normals) delete[] _normals;
}

bool HeightMap::Create(const string_type& path, float width, float length, float height_min, float height_max)
{
		_path = path;
		_map_width = width;
		_map_length = length;
		_map_height_min = height_min;
		_map_height_max = height_max;

		FILE* pf = NULL;
		long fsize;
		char* fbuffer = NULL;
```

```
            size_t bytesRead;
            int bmp_width_aligned;
            int bmp_byte_per_pixel;
            int bmp_image_size;
            int pindex, pcolor;
            rMath::Vector3D v1, v2, face_n[4], vertex_n;

            //open bitmap:
            if (0 != _tfopen_s(&pf, _path.c_str(), _T("rb")))
            {
            return false;
            }

            //obtain file size:
            fseek (pf , 0 , SEEK_END);
            fsize = ftell(pf);
            rewind (pf);

            //read bitmap:
            fbuffer = (char*)malloc(sizeof(char)*fsize);
            bytesRead = fread(fbuffer, 1, fsize, pf);
            if (fsize != bytesRead)
            goto __HeightMap_Create_Error;

            fclose(pf); pf = NULL;

            //read bitmap file header and bitmap info header:
            BITMAPFILEHEADER* bmfh = (BITMAPFILEHEADER*)fbuffer;
            BITMAPINFOHEADER* bmih = (BITMAPINFOHEADER*)(fbuffer + sizeof(BITMAPFILEHEADER));
            unsigned char* bmp = (unsigned char*)(fbuffer + bmfh->bfOffBits);

            assert(bmih->biPlanes == 1 && "HeightMap::Create() : bmih->biPlanes != 1₩n");
            //assert(bmih->biBitCount == 8 && "HeightMap::Create() : bmih->biBitCount != 8");
            assert(bmih->biCompression == BI_RGB && "HeightMap::Create() : bmih->biCompression != BI_RGB₩n");
            //assert(bmih->biSizeImage == bmih->biWidth*bmih.biHeight && "HeightMap::Create() : bmih->biSizeImage
!= bmih->biWidth*bmih.biHeight₩n");
            assert(bmih->biWidth >= 2 && "HeightMap::Create() : bmih->biWidth < 2₩n");
            assert(bmih->biHeight >= 2 && "HeightMap::Create() : bmih->biHeight < 2₩n");

            _bmp_pixel_width = bmih->biWidth;
            _bmp_pixel_length = bmih->biHeight;
            bmp_image_size  = _bmp_pixel_width*_bmp_pixel_length;

            //calculate memory width of one row of bitmap, etc:
            bmp_width_aligned = ALIGNEDWIDTHBYTES(_bmp_pixel_width*bmih->biBitCount);

            switch(bmih->biBitCount)
            {
            case 8:  bmp_byte_per_pixel = 1; break;
            case 24: bmp_byte_per_pixel = 3; break;
            case 32: bmp_byte_per_pixel = 4; break;
            default:
            goto __HeightMap_Create_Error;
            }

            //initialize variables from loaded bitmap info:
            _map_width_scale = _map_width / (float)(_bmp_pixel_width-1);
            _map_length_scale = _map_length / (float)(_bmp_pixel_length-1);
            _map_height_scale = (_map_height_max - _map_height_min) / 255.0f;

            //allocate memory for vertex heights and vertex normals:
            if (_heights) delete[] _heights;
            _heights = new float[bmp_image_size];
            if (_normals) delete[] _normals;
            _normals = new float[bmp_image_size*3];

            //calculate vertex heights:
            for (int py=0; py<_bmp_pixel_length; py++) {
            for (int px=0; px<_bmp_pixel_width; px++) {
            pindex = py*_bmp_pixel_width + px;

            pcolor = 0;
            for (int i=0; i<bmp_byte_per_pixel; i++)
            pcolor += (int)bmp[py*bmp_width_aligned + px*bmp_byte_per_pixel + i];

            _heights[pindex] = _map_height_min + (float)(pcolor) / (float)(bmp_byte_per_pixel) *
_map_height_scale;
```

```
}
}

//calculate vertex normals:
for (int py=0; py<_bmp_pixel_length; py++) {
for (int px=0; px<_bmp_pixel_width; px++) {

pindex = py*_bmp_pixel_width + px;

// face normal for the 1st quadrant:
v1[0] = _map_width_scale;
v1[1] = 0.0f;
v1[2] = (px+1 < _bmp_pixel_width)
?  (_heights[py*_bmp_pixel_width + (px+1)] - _heights[pindex])
: -(_heights[py*_bmp_pixel_width + (px-1)] - _heights[pindex]);

v2[0] = 0.0f;
v2[1] = _map_length_scale;
v2[2] = (py+1 < _bmp_pixel_length)
?  (_heights[(py+1)*_bmp_pixel_width + px] - _heights[pindex])
: -(_heights[(py-1)*_bmp_pixel_width + px] - _heights[pindex]);

// face_normal = v1 x v2;
face_n[0] = v1.Cross(v2);
face_n[0].Normalize();

// face normal for the 2nd quadrant:
v1 = v2;

v2[0] = -_map_width_scale;
v2[1] = 0.0f;
v2[2] = (px-1 >= 0)
?  (_heights[py*_bmp_pixel_width + (px-1)] - _heights[pindex])
: -(_heights[py*_bmp_pixel_width + (px+1)] - _heights[pindex]);

face_n[1] = v1.Cross(v2);
face_n[1].Normalize();

// face normal for the 3rd quadrant:
v1 = v2;

v2[0] = 0.0f;
v2[1] = -_map_length_scale;
v2[2] = (py-1 >= 0)
?  (_heights[(py-1)*_bmp_pixel_width + px] - _heights[pindex])
: -(_heights[(py+1)*_bmp_pixel_width + px] - _heights[pindex]);

face_n[2] = v1.Cross(v2);
face_n[2].Normalize();

// face normal for the 4th quadrant:
v1 = v2;

v2[0] = _map_width_scale;
v2[1] = 0.0f;
v2[2] = (px+1 < _bmp_pixel_width)
?  (_heights[py*_bmp_pixel_width + (px+1)] - _heights[pindex])
: -(_heights[py*_bmp_pixel_width + (px-1)] - _heights[pindex]);

face_n[3] = v1.Cross(v2);
face_n[3].Normalize();

vertex_n.zero();
for (int i=0; i<4; i++)
vertex_n += face_n[i];
vertex_n.Normalize();

_normals[3*pindex+0] = (float)vertex_n[0];
_normals[3*pindex+1] = (float)vertex_n[1];
_normals[3*pindex+2] = (float)vertex_n[2];
}
}

free(fbuffer);

return true;
```

```
__HeightMap_Create_Error:
            if (pf) {
            fclose(pf);
            }
            if (_heights) {
            delete[] _heights;
            _heights = NULL;
            }
            if (_normals) {
            delete[] _normals;
            _normals = NULL;
            }
            if (fbuffer) {
            free(fbuffer);
            }
            return false;
}

bool HeightMap::GetHeight(float x, float y, float& height)
{
            // by default, it return "0".
            height = 0.0f;

            if (!_heights) {
            return false;
            }

            int px = (int)((_map_width * 0.5f + x) / _map_width_scale);
            if (px < 0) px = 0;
            else if (px >= _bmp_pixel_width) px = _bmp_pixel_width-1;

            int py = (int)((_map_length * 0.5f + y) / _map_length_scale);
            if (py < 0) py = 0;
            else if (py >= _bmp_pixel_length) py = _bmp_pixel_length-1;

            // bilinear interpolation:
            float x1 = (float)px * _map_width_scale - 0.5f * _map_width;
            float y1 = (float)py * _map_length_scale - 0.5f * _map_length;
            float x2 = x1 + _map_width_scale;
            float y2 = y1 + _map_length_scale;
            float Q11 = _heights[py                            *_bmp_pixel_width + px];
            float Q12 = _heights[min(py+1, _bmp_pixel_length-1)*_bmp_pixel_width + px];
            float Q21 = _heights[py                            *_bmp_pixel_width + min(px+1,
_bmp_pixel_width-1)];
            float Q22 = _heights[min(py+1, _bmp_pixel_length-1)*_bmp_pixel_width + min(px+1,
_bmp_pixel_width-1)];
            height = (Q11 * (x2-x) * (y2-y) + Q21 * (x-x1) * (y2-y) + Q12 * (x2-x) * (y-y1) + Q22 * (x-x1) *
(y-y1)) / ((x2-x1) * (y2-y1));

            return true;
}

bool HeightMap::GetNormal(float x, float y, float normal[3])
{
            // by default, it returns "Z" unit vector.
            normal[0] = 0.0f;
            normal[1] = 0.0f;
            normal[2] = 1.0f;

            if (!_heights) {
            return false;
            }

            int px = (int)((_map_width * 0.5f + x) / _map_width_scale);
            if (px < 0) px = 0;
            else if (px >= _bmp_pixel_width) px = _bmp_pixel_width-1;

            int py = (int)((_map_length * 0.5f + y) / _map_length_scale);
            if (py < 0) py = 0;
            else if (py >= _bmp_pixel_length) py = _bmp_pixel_length-1;

            // bilinear interpolation:
            float x1 = (float)px * _map_width_scale - 0.5f * _map_width;
            float y1 = (float)py * _map_length_scale - 0.5f * _map_length;
            float x2 = x1 + _map_width_scale;
            float y2 = y1 + _map_length_scale;
            float Q11, Q12, Q21, Q22;
```

```
            for (int i=0; i<3; i++) {
            Q11 = _normals[3*(py                            *_bmp_pixel_width + px)                            +
i];
            Q12 = _normals[3*(min(py+1, _bmp_pixel_length-1)*_bmp_pixel_width + px)                            +
i];
            Q21 = _normals[3*(py                            *_bmp_pixel_width + min(px+1, _bmp_pixel_width-1)) +
i];
            Q22 = _normals[3*(min(py+1, _bmp_pixel_length-1)*_bmp_pixel_width + min(px+1, _bmp_pixel_width-1)) +
i];
            normal[i] = (Q11 * (x2-x) * (y2-y) + Q21 * (x-x1) * (y2-y) + Q12 * (x2-x) * (y-y1) + Q22 * (x-x1) *
(y-y1)) / ((x2-x1) * (y2-y1));
            }

            // normalize:
            float norm = sqrt(pow(normal[0], 2) + pow(normal[1], 2) + pow(normal[2], 2));
            normal[0] /= norm;
            normal[1] /= norm;
            normal[2] /= norm;

            return true;
}

bool HeightMap::GetNormalEx(float x, float y, int wnd_size, float normal[3])
{
            // by default, it returns "Z" unit vector.
            normal[0] = 0.0f;
            normal[1] = 0.0f;
            normal[2] = 1.0f;

            if (!_heights) {
            return false;
            }

            int px = (int)((_map_width * 0.5f + x) / _map_width_scale);
            if (px < 0) px = 0;
            else if (px >= _bmp_pixel_width) px = _bmp_pixel_width-1;

            int py = (int)((_map_length * 0.5f + y) / _map_length_scale);
            if (py < 0) py = 0;
            else if (py >= _bmp_pixel_length) py = _bmp_pixel_length-1;

            // bilinear interpolation:
            float x1 = (float)px * _map_width_scale - 0.5f * _map_width;
            float y1 = (float)py * _map_length_scale - 0.5f * _map_length;
            float x2 = x1 + _map_width_scale;
            float y2 = y1 + _map_length_scale;
            float Q11, Q12, Q21, Q22;
            for (int i=0; i<3; i++) {

            Q11 = 0.0f; Q12 = 0.0f; Q21 = 0.0f; Q22 = 0.0f;

            for (int r=0; r<wnd_size; r++) {
            for (int c=0; c<wnd_size; c++) {
            Q11 += _normals[3*(max(py-r, 0)                    *_bmp_pixel_width + max(px-c, 0))
  + i];
            Q12 += _normals[3*(min(py+1+r, _bmp_pixel_length-1)*_bmp_pixel_width + max(px-c, 0))
  + i];
                        Q21 += _normals[3*(max(py-r, 0)                    *_bmp_pixel_width + min(px+1+c,
_bmp_pixel_width-1)) + i];
            Q22 += _normals[3*(min(py+1+r, _bmp_pixel_length-1)*_bmp_pixel_width + min(px+1+c,
_bmp_pixel_width-1)) + i];
            }
            }

            normal[i] = (Q11 * (x2-x) * (y2-y) + Q21 * (x-x1) * (y2-y) + Q12 * (x2-x) * (y-y1) + Q22 * (x-x1) *
(y-y1)) / ((x2-x1) * (y2-y1));
            }

            // normalize:
            float norm = sqrt(pow(normal[0], 2) + pow(normal[1], 2) + pow(normal[2], 2));
            normal[0] /= norm;
            normal[1] /= norm;
            normal[2] /= norm;

            return true;
}
```

7.3. 가상 시뮬레이터의 작업경로 및 영역 표현 코딩

```
#include "rDeviceTractorGrid.h"
#include "rCommon/rParseUtil.h"
#include "rMath/rMath.h"

// MAX_SINGLE_DEVICE_BUFFER_SIZE_TO_SEND is definded in rxsdk(rxDeviceManager.cpp) as...
#define MAX_SINGLE_DEVICE_BUFFER_SIZE_TO_SEND    (32*1024)
//
// One readDeviceValue() function call should not return data greater than this.
// If so, assertion will fail in debug build or the device value will not be deliverd to
// rgd(graphics plug-in).
// To avoid this limit, one readDeviceValue() function call will split the whole data and
// return only the partial data buffer.
// If the user want to read whole device buffer, readDeviceValue() should be called multiple times
// until it gets whole device buffer.
//
#define RDEVICEGRID_READSIZE_MAX 32000

// implementation of rDeviceTractorGrid

RD_IMPLE_FACTORY(TractorGrid)

rDeviceTractorGrid::rDeviceTractorGrid()
: _tool_robotName(_T(""))
, _tool_bodyName(_T(""))
, _tool_deviceName(_T(""))
, _tool_bodyID(INVALID_RID)
, _tool_deviceID(INVALID_RID)
, _tool_width(1.0f)
, _tool_type(TOOL_LINE)
, _tool_activated(false)
, _rows(1)
, _cols(1)
, _cell_count(1)
, _cell_count_occupied(0)
, _cell_size(0.1f)
, _grid(NULL)
, _read_index(0)
, _read_offset(0)
, _read_size_max(RDEVICEGRID_READSIZE_MAX)
{
}

rDeviceTractorGrid::~rDeviceTractorGrid()
{
        delete[] _grid;
}

void rDeviceTractorGrid::onCreate(const rDeviceContext& rdc)
{
        RD_DEVICE_CLASS(Base)::onCreate(rdc);

        const TCHAR* prop;

        // CellSize
        prop = getProperty(_T("CELL_SIZE"));
        if (prop)
                _cell_size = (float)_tstof(prop);
        if (_cell_size < 0)
                _cell_size = 0.1f;

        // row size
        prop = getProperty(_T("row"));
        if (prop)
                _rows = _tstoi(prop);
        if (_rows < 0)
                _rows = 1;

        // column size
        prop = getProperty(_T("col"));
        if (prop)
                _cols = _tstoi(prop);
        if (_cols < 0)
                _cols = 1;
```

```
		// tool info
		prop = getProperty(_T("TOOL_SYS_NAME"));
		if (prop)
			_tool_robotName = prop;

		prop = getProperty(_T("TOOL_BODY_NAME"));
		if (prop)
			_tool_bodyName = prop;

		prop = getProperty(_T("TOOL_DEVICE_NAME"));
		if (prop)
			_tool_deviceName = prop;

		// maximum buffer size to read data once
		prop = getProperty(_T("READ_SIZE_MAX"));
		if (prop)
			_read_size_max = min(_tstoi(prop), RDEVICEGRID_READSIZE_MAX);

		// Initialize
		_grid = new unsigned char[_rows * _cols];

		// Grid Initialize
		memset(_grid, 0x0, sizeof(unsigned char) * (_rows * _cols));
}

void rDeviceTractorGrid::onInit()
{
		RD_DEVICE_CLASS(Base)::onInit();

		_tool_bodyID = _rdc.m_deviceAPI->getBodyID(_tool_robotName.c_str(), _tool_bodyName.c_str());
		if (_tool_bodyID == INVALID_RID)
		{
			_tprintf(_T("ERROR: rDeviceVCRGrid: cannot find tool (system: %s, body: %s)\n"),
_tool_robotName.c_str(), _tool_bodyName.c_str());
			enable(false);
		}

		_tool_width = 1.0f;

		_tool_deviceID = _rdc.m_deviceAPI->getDeviceID(_tool_robotName.c_str(), _tool_deviceName.c_str());
		if (_tool_deviceID == INVALID_RID)
		{
			_tprintf(_T("ERROR: rDeviceVCRGrid: cannot find tool. (system: %s, device: %s)\n"),
_tool_robotName.c_str(), _tool_deviceName.c_str());
			enable(false);
		}
		else
		{
			TCHAR prop[1024];

			if (_rdc.m_deviceAPI->getDeviceProperty(_tool_deviceID, _T("TOOL_TYPE"), prop))
			{
				if (!_tcsicmp(prop, _T("line")))
					_tool_type = TOOL_LINE;
				else
					_tool_type = TOOL_NONE;
			}
			if (_tool_type == TOOL_NONE)
			{
				_tprintf(_T("ERROR: rDeviceVCRGrid: tool type is not known. (system: %s,
device: %s)\n"), _tool_robotName.c_str(), _tool_deviceName.c_str());
				enable(false);
			}

			if (_rdc.m_deviceAPI->getDeviceProperty(_tool_deviceID, _T("TOOL_OFFSET_R"), prop))
			{
				float R[9];
				const TCHAR* prop_ptr = (const TCHAR*)prop;
				parse_vector(R, 9, prop_ptr);
				_tool_offset.R.Set(R[0], R[1], R[2], R[3], R[4], R[5], R[6], R[7], R[8]);
			}
			if (_rdc.m_deviceAPI->getDeviceProperty(_tool_deviceID, _T("TOOL_OFFSET_r"), prop))
			{
				float r[3];
				const TCHAR* prop_ptr = (const TCHAR*)prop;
				parse_vector(r, 3, prop_ptr);
				_tool_offset.r.Set(r[0], r[1], r[2]);
```

```
                }

                // width of worker(tool)
                if (_rdc.m_deviceAPI->getDeviceProperty(_tool_deviceID, _T("TOOL_WIDTH"), prop))
                        _tool_width = (float)_tstof(prop);
                if (_tool_width < 0)
                        _tool_width = 1.0f;
        }

        _cell_count = _cols * _rows;
        _cell_count_occupied = 0;
}

void rDeviceTractorGrid::onTerminate()
{
        RD_DEVICE_CLASS(Base)::onTerminate();
}

int rDeviceTractorGrid::readDeviceValue(void* buffer, int len, int port)
{
        int read = 0;

        if (len > (int)(_read_size_max + sizeof(int) * 3))
        {
                _lock.lock();
                {
                        int datasize = (int)(sizeof(unsigned char) * (_rows * _cols)) - _read_offset;
                        if (datasize < 0)
                        {
                                read = 0;
                        }
                        else if (datasize <= _read_size_max)
                        {
                                ((int*)buffer)[0] = _read_index;
                                ((int*)buffer)[1] = _read_offset;
                                ((int*)buffer)[2] = datasize;
                                memcpy((char*)buffer + 12, _grid + _read_offset, datasize);
                                _read_index = 0;
                                _read_offset = 0;
                                read = datasize + 12;
                        }
                        else
                        {
                                datasize = _read_size_max;
                                ((int*)buffer)[0] = _read_index;
                                ((int*)buffer)[1] = _read_offset;
                                ((int*)buffer)[2] = datasize;
                                memcpy((char*)buffer + 12, _grid + _read_offset, datasize);
                                _read_index += 1;
                                _read_offset += datasize;
                                read = datasize + 12;
                        }
                }
                _lock.unlock();
                return read;
        }
        else
                return 0;
}

int rDeviceTractorGrid::writeDeviceValue(void* buffer, int len, int port)
{
        if (len == sizeof(int))
        {
                _tool_activated = *(int*)buffer == 0 ? false : true;
                return sizeof(int);
        }
        else
                return 0;
}

int rDeviceTractorGrid::monitorDeviceValue(void* buffer, int len, int port)
{
        if (len == (int)(sizeof(int) * 2))
        { // read (occupied cell count , whole cell count)
                _lock.lock();
                ((int*)buffer)[0] = _cell_count_occupied;
```

```
                        ((int*)buffer)[1] = _cell_count;
                        _lock.unlock();
                        return len;
                }
                else
                        return 0;
}

void rDeviceTractorGrid::importDevice(rTime time, void* mem)
{
                float r_w[3];
                float R_w[9];

                _lock.lock();
                {
                        if (_tool_activated)
                        {
                                _rdc.m_deviceAPI->getBodyPosition(_tool_bodyID, r_w);
                                _rdc.m_deviceAPI->getBodyOrientation(_tool_bodyID, R_w);
                                _tool_bodyHT.R.Set(R_w[0], R_w[1], R_w[2],
                                                                                R_w[3], R_w[4], R_w[5],
                                                                                R_w[6], R_w[7], R_w[8]);
                                _tool_bodyHT.r.Set(r_w[0], r_w[1], r_w[2]);

                                rMath::HTransform Tg = _tool_bodyHT * _tool_offset;
                                rMath::Displacement v1(-_tool_width*0.5, 0, 0);
                                rMath::Displacement v2( _tool_width*0.5, 0, 0);
                                rMath::Displacement v1g = Tg.Transform(v1);
                                rMath::Displacement v2g = Tg.Transform(v2);
                                rMath::Vector3D dir = v2g - v1g;
                                rMath::Vector3D p;
                                dir.Normalize();
                               // to calculate number of segments,
                               // we multiply by 2 considering the tool is lying in diagonal direction of the cell.
                                unsigned int nSegments = static_cast<unsigned int>(2.0*_tool_width/_cell_size);
                                float segLength = (nSegments > 0 ? _tool_width / (float)nSegments :
_tool_width);
                                for (unsigned int i = 0; i <= nSegments; ++i)
                                {
                                        p = v1g + dir*i*segLength;

                        int col = (int)(p[0] / _cell_size) + (int)(_cols * 0.5) + (p[0] < 0 ? -1 : 0);
                                        int row = (int)(p[1] / _cell_size) + (int)(_rows * 0.5) + (p[1] < 0 ? -1
: 0);

                                        if (col >= 0 && col < _cols &&
                                                row >= 0 && row < _rows &&
                                                0x00 == _grid[row*_cols + col])
                                        {
                                                _grid[row*_cols + col] = 0x1;
                                                _cell_count_occupied++;
                                        }
                                }
                        }
                }
                _lock.unlock();
}

void rDeviceTractorGrid::Reset()
{
                //_read_offset = 0;
                //_read_index = 0;
                memset(_grid, 0, _cell_count);
                _cell_count_occupied = 0;

}
```

Development of Path Generation and Robust Tracking Controller for Unmanned Tillage Tractor in Polygonal Fields

XiongZhe Han

Abstract

Auto-guidance tractor navigation requires the integration of various technologies including path planning and tracking, position and orientation sensing, vehicle control and obstacle avoidance. High-precision auto farming is rapidly becoming a reality with the requirements of agricultural applications. The application of an auto-guided tractor to rice cultivation in Korean paddy fields may be limited by tire slippage and a necessity for automatic headland turning algorithms due to relatively soft and wet soil conditions and the use of a small-sized field < 1 ha. This study reports on the development of a path generation and robust tracking system for an unmanned tillage tractor in polygonal fields. A path generation program was developed to automatically generate the full coverage of a path map for the auto-guided tractor to follow in rectangular and polygon fields. When parameters related to the used tractor and implements were input into the developed path generation program, useful information about the performance of auto tillage tasks such as field time, field travelled distance,

and field efficiency, could be provided. To enhance the capability of the autonomous tractor based on a previously developed look-ahead distance method to follow along a predefined path, a robust path tracking controller was developed by considering sliding effects in agricultural fields. A predictive algorithm was designed for determining the sideslip angles of the tractor using GPS and IMU devices, which were used as adaptive variables for the path tracking algorithm. A 3D computer simulator was developed for the tractor to virtually perform tillage operations based on predefined path generation commands and robust path tracking algorithms. An extended kinematic model that could change the coefficients of cornering stiffness according to different road conditions was used to describe the motion of the tractor in the presence of slippage. In addition, algorithms for compensating for changes in GPS locations due to sloping grounds and delayed GPS signals due to a distance between a GPS mounting location and an implement were studied. The feasibility using the simulator for the development of path generation and tracking algorithms was investigated by studying the effects of various driving parameters such as a slippage condition, a change in ground slope, a distance between GPS mounting location and an tillage implement on the performance of an auto-guided tillage tractor. A prototype autonomous tillage tractor equipped with the developed path generation and tracking algorithms were developed and tested in paddy fields with various shapes of field boundary. The use of the developed robust path tracking algorithm provided an reduction in RMSE tracking error to <10 cm as compared to the previously developed look-ahead distance method due to its ability to take the effect of

tire-sliding at headland areas into account. When using the developed implement control algorithm, the errors that occurred due to the delayed response of the 3-point hitch implement were reduced to < 15 cm, which were considered as an acceptable level of accuracy for autonomous operations. The results of the plowing operation performance test in a rectangular field of 80 m x 30 m showed that the field efficiency was 79.9% and the ratio of the effective operation area was 90.3%, which were similar to those obtained from a manual-based tillage operation method. The field tests performed with the autonomous tractor in a trapezoid and an "L" shaped field showed the tracking errors of 14.9 cm and 21.8 cm, respectively, the field efficiencies of 62.1 ~ 70.6% and the ratios of effective operation areas of 91.4 ~ 92.7%, thereby implying that the developed path generation algorithm and robust tracking controller could guide the tractor with acceptable accuracy and quality. Further studies on the robust control of the 3-point hitch implement were needed to provide the ability of the implement control in the presence of various load disturbances by spatial variability in soil strength.

Key words: Auto-guidance tractor, Path generation in polygonal field, Robust path tracking, Virtual simulator, Field efficiency
Student number: 2013-31315