

차량 시뮬레이터의 운동시스템 구동소프트웨어 개발

Development of a Motion System Operating Software for a Driving Simulator

박경관, 박일경*, 최연철*, 조준희*, 안태환**, 이운성*, 김정하*

* 국민대학교 자동차공학과(Tel : +82-2-919-3714; E-mail : m962404@kmu1.kookmin.ac.kr)

** 국민대학교 전자공학과(Tel : +82-2-910-4709; E-mail : m962413@kmu1.kookmin.ac.kr)

Abstract This paper describes the operating software of a motion system developed for a driving simulator, consisting of a six degree of freedom Stewart platform driven hydraulically. The drive logic, consisting of an washout algorithm, inverse kinematic analysis, and a control algorithm, has been developed and applied for creating high fidelity motion cues. The basic environment of the operating software is based on LabVIEW 4.0 and DLL modules compiled by Fortran.

Keywords Driving simulator, Motion System, Hydraulic System, Inverse Kinematics, Washout Algorithm

1. 서론

차량 시뮬레이터(Driving Simulator)는 운전자가 자동차를 운전하는 동안 수행하는 조향 휠 조작, 액셀레이터와 브레이크 페달 조작을 통해 야기되는 차량의 운동을 실시간 시뮬레이션을 수행해 예측하고 그 결과를 운동, 시각 및 음향 큐를 통해 운전자에게 피드백하여 차량 시뮬레이터에 탑승한 운전자가 실제로 자동차를 운전하고 있다는 느낌을 받게 하는 가상현실 장비이다. [1]

차량 시뮬레이터에서 운동시스템은 유압서보시스템을 이용해 운전자로부터 입력받은 값을 토대로 차량 동력학모델로부터 예측되는 차량의 운동을 현실감 있게 재현하는 중요한 역할을 한다. 이 유압서보시스템은 유압펌프, 서보밸브, 액츄에이터, LVDT등을 포함하는 복잡한 회로로 구성된다.

운동시스템에 사용되는 스투어트 플랫폼은 병렬구조형 기구로서 오차변위가 적어 정확도가 높고, 하중이 축방향으로 걸리게 되므로 큰 하중을 구동시킬 수 있다. 그러나 운동범위가 작기 때문에 탑승한 운전자에게 실제와 같은 느낌을 부여하기 위해 Washout 알고리즘 모듈을 거쳐서 제한된 공간상에서 현실감을 표현하도록 하고, 여기서 생성된 데이터를 역기구학 해석 알고리즘에 적용한 후 최종적으로 6개의 각 유압액츄에이터를 구동하게 된다. [2]

본 연구에서는 National Instrument사의 LabVIEW 4.0 프로그램[3]을 이용해 운동시스템을 실시간으로 구동할 수 있는 구동소프트웨어를 개발하고, 이를 운동시스템 성능평가를 위한 실험 및 차량 시뮬레이터를 이용한 주행시뮬레이션에 적용하였다.

2. 운동시스템 구동로직

2.1 시스템 개요

그림 1, 2는 유압액츄에이터, 서보밸브, 어큐뮬레이터등으로 구성된 운동플랫폼과 각종신호를 처리할 수 있는 유압제어기 콘솔을 보인다. 그림 3은 본 연구에서 개발한 운동시스템의 제어블록도를 나타내고 있다.



Fig. 1 Motion Platform

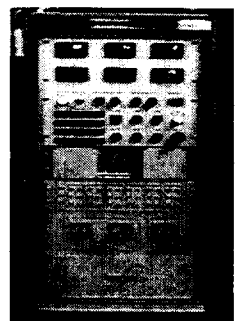


Fig. 2 Control Console

운동시스템의 구동 제어 및 데이터 수집을 위한 기본 환경으로는 National Instruments사의 DAQ Board와 LabVIEW 4.0 프로그램을 사용하였으며, 역기구학, Washout 알고리즘 및 시스템 특성상 LabVIEW로 구현이 불가능한 부분들은 각각 Fortran으로 프로그래밍하여 DLL (Dynamic-Link Library)로 모듈화한 후, LabVIEW와 연계하여 사용하였다.

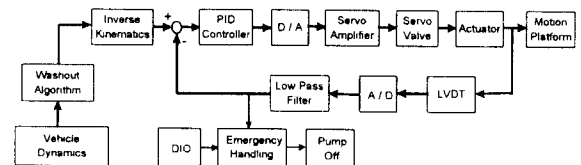


Fig. 3 Motion System Functional Block Diagram

2.2 Washout Algorithm

운동영역의 제한없이 수행하는 차량의 운동을 모사하는 운동시스템은 기구학적인 운동영역과 시스템의 동특성에 있어서 한계를 갖고 있으므로 정확한 차량 운동의 재현이 어렵다. 그러므로, 운동시스템의 물리적인 한계 내에서 운동범위를 제한하는 것과 운전자에게 전달되는 차량 운동을 인체 감지 영역 내로 유지시키는 Washout 알고리즘은 운동시스템의 현실감을 확보하는데 필수적이다.

운동플랫폼에서 모사할 차량의 명령 큐는 운전자가 주행 시나리오를 민감하게 체험할 수 있는 운전석의 각 방향 선형가속도와 각 속도이다. 명령 큐의 성분 중 저주파 신호는 운동시스템을 기구학적

인 한계에 이르게 하여 더 이상의 운동재현이 불가능하므로, 고주파 필터링을 통해 인체감각이 감지할 수 있는 고주파 성분만을 재현한다. 또한 고주파 필터링은 임의의 운동 이후 다음 명령 큐를 재현하기 위해 운동 플랫폼을 운동원점으로 되돌리는 역할도 수행한다. Washout 알고리즘에서 고주파 필터링과 함께 중요한 설계요소로는 Tilt Coordination이다. J-Turn과 같은 주행상황에서 발생하는 지속적인 차량의 횡가속도는 저주파 신호이므로 고주파 필터링을 거치면 재현을 할 수가 없게 된다. 선형가속도와 중력 벡터와의 차이인 비력(Specific Force)을 이용하면 또 다른 가속도를 재현할 수 있다 [4]. 필터링된 저주파 가속도를 재현하기 위해 운동플랫폼을 더 기울여 주면 운전자는 비력에 의해 지속적인 가속도를 체험하게 된다. 그러나, 너무 빨리 기울이면 잘못된 각속도와 이에 따라 발생하는 가속도를 운전자가 감지하게 되며, 시각 시스템과도 불일치 되어 심할 경우 멀미를 발생시키는 등의 시뮬레이터 병(Sickness)을 발생시키므로 주의해야 한다.

Washout 알고리즘의 검증을 위해 비력에 민감한 Otolith계와 가속도에 민감한 Vestibular계에 대한 인체모델을 적용하여 차량 모델에서와 Washout 알고리즘을 거쳐 모사되는 운동플랫폼간의 가속도/각속도를 비교 분석한다. 최종적으로 운동시스템 구동에 필요한 신호 발생을 위해 역기구학 모델에 운동플랫폼의 변위(X_p, Y_p, Z_p)와 오일러각(ϕ_p, θ_p, ψ_p)을 입력신호로 넘겨준다. 그림 4는 Pitch/Surge 나 Roll/Sway 운동에 대한 간단한 Washout 필터의 블록도를 보이고 있으며, 그림 5는 횡가속도와 롤각에 대한 명령 큐와 Washout 알고리즘을 거쳐 여과했을 때의 재현 큐를 비교하여 보이고 있다.

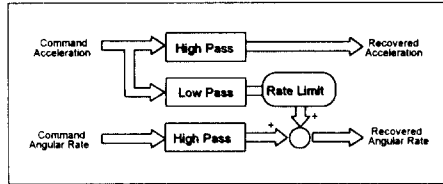


Fig. 4 Washout Filter Block Diagram

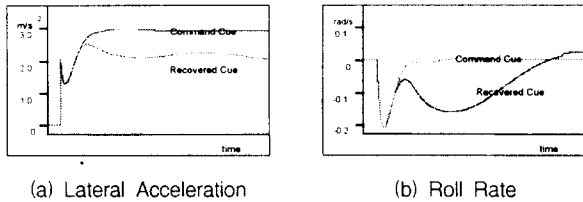


Fig. 5 Effects of Washout Filtering

2.3 스튜어트 플랫폼의 역기구학 해석

역기구학 해석은 Washout 알고리즘을 거쳐 나온 이동플랫폼의 회전엔 대한 롤(Roll), 요(Yaw), 피치(Pitch) 그리고 X, Y, Z축의 좌표로부터 각 액추에이터 길이를 구하는데 이용된다. 구해진 액추에이터의 길이는 피드백 제어시스템의 기준입력이 된다.

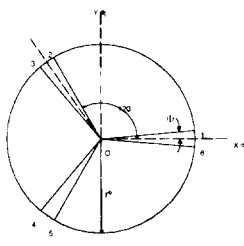


Fig. 6 Base Coordinate System

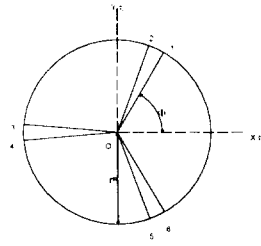


Fig. 7 Platform Coordinate System

액추에이터의 길이를 구하기 위해 베이스와 이동플랫폼의 중심점 O_b, O_p 에서 조인트의 위치를 정의하였다. 각 한 쌍의 조인트들의 각도는 120° 로 설정하였다. 그림 6과 7은 베이스와 이동플랫폼에 있는 각 조인트의 위치를 나타낸다. 그리고 X축은 중심점에서 조인트 1과 6사이를 지나가고, Z축은 윗방향으로 설정되어 있다.

베이스 좌표계의 각 조인트의 중심점은 아래와 같다.

$$B_i = r_b [\cos(A_{bi}) \quad \sin(A_{bi}) \quad 0] \quad (i = 1, \dots, 6)$$

$$A_{b1} = \phi_B \quad A_{b2} = \frac{2}{3}\pi - \phi_B$$

$$A_{b3} = \frac{2}{3}\pi + \phi_B \quad A_{b4} = -(\frac{2}{3}\pi + \phi_B)$$

$$A_{b5} = -(\frac{2}{3}\pi - \phi_B) \quad A_{b6} = -\phi_B$$

이동플랫폼 좌표계의 각 조인트의 중심점은 아래와 같다.

$$P_i = r_p [\cos(A_{Pi}) \quad \sin(A_{Pi}) \quad 0] \quad (i = 1, \dots, 6)$$

$$A_{P1} = \phi_P \quad A_{P2} = \frac{2}{3}\pi - \phi_P$$

$$A_{P3} = \frac{2}{3}\pi + \phi_P \quad A_{P4} = -(\frac{2}{3}\pi + \phi_P)$$

$$A_{P5} = -(\frac{2}{3}\pi - \phi_P) \quad A_{P6} = -\phi_P$$

베이스 좌표계의 중심점에서 각 조인트 중심점 B_i ($i=1-6$)와 운동플랫폼 좌표계의 중심점에서 각 조인트 중심점 P_i ($i=1-6$)의 벡터를 \vec{b}_i, \vec{p}_i 라고 정의한다. 여기서, 운동플랫폼에서 정의된 벡터 \vec{p}_i 를 글로벌 좌표계로 변환시키기 위해서 $z-y-x$ 오일러 각도를 사용하였다. 이때 베이스 좌표계와 운동플랫폼간의 각 액추에이터의 길이벡터 \vec{l}_i 는 다음과 같다.

$$\vec{l}_i = [R]_p \vec{p}_i + \vec{p}_0 - \vec{b}_i \quad (i = 1, \dots, 6) \quad (5)$$

그러므로 각 액추에이터의 길이는 벡터 \vec{l}_i 의 크기로 나타낼 수 있으며, 이와 같이 결정된 6개 액추에이터 길이가 LVDT 길이 변위에 따른 전압의 형태로 변환되어 유압시스템 구동을 위한 서보밸브의 입력 신호가 된다 [5].

2.4 PID 제어기

제어를 위한 피드백 신호를 측정하기 위하여 액추에이터의 변위를 측정하는 6개의 LVDT를 사용하였다. A/D 변환기를 거친 신호는 서보밸브 및 주변 잡음 등의 영향으로 고주파 성분을 포함하고 있는데, 2차의 버터워스 저역통과 필터를 이용하여 필터링 하였다. 피드백 제어를 수행하기 위하여 제어가 용이하고 비교적 좋은 성능을 얻을 수 있는 식 (6)과 같은 PID 제어기를 사용하였다 [6].

$$G(s) = K_p + K_i/s + K_d s \quad (6)$$

여기서 K_p 는 비례 이득을 나타내고, K_i 는 적분 이득을 나타내며, K_d 는 미분 이득을 나타낸다. 적분 이득은 정상상태에서 오차를 보정하는 역할을 하며 미분 이득은 과도응답 속도와 관계되므로 이 두 값을 적절히 하면 원하는 정상상태 오차와 과도응답 속도를 얻을 수 있다. PID 제어에서 원하는 성능을 얻기 위한 이득값 조정이 중요한 문제인데, 시스템 모델링을 통해 전달함수를 구하였으나, 시스템에 적용한 서보밸브의 유량-압력계수, 압력 민감계수, 밸브이득 등의 주요 설계인자들을 제작사의 사정으로 구할 수 없었으므로, 시스템 특성상 시스템에 부리가 가지 않는 범위에서 실험적인 방법으로 PID 이득값을 조정하였다. 그림 8은 PID 제어기를 사용했을 때와 사용하지 않았을 경우 계단과 입력에 대한 운동시스템의 응답을 나타내고 있으며, 그림 9는 실시간으로 차량 동력학 모델의 해석 결과를 Washout 알고리즘으로 필터링한 후 생성한 모션 큐에 대한 운동시스템의 응답을 나타내고 있다.

Ziegler-Nichols 페루프 이득 조정방법[7]을 사용하여 이득값 조정을 시도하였으나 원하는 제어 성능에는 미치지 못하였다. 이 과정에

서 부하가 없는 유압 시스템은 제어 밸브가 무부하 압력 근처에서 동작하고 그 동작점이 일정하다고 가정할 수 있지만, 운동플랫폼 상판의 질량이나 관성모멘트 등의 작용으로 시스템 변수가 불확실하거나 예상치 못한 외란이 발생할 경우에는 이 가정을 만족하지 않는다는 것을 알 수 있었다. 따라서 이와 같이 제어밸브와 파라미터의 동작점이 부하 외란에 의해 변하는 시스템의 경우 고정상수 이득회환 제어기(Constant Gain Feedback Controller)가 적합하지 않다는 것을 인지하고, 현재 동작 조건과 부하 외란의 변화에도 원하는 제어 레적을 만족하는 적응 제어기(Adaptive Controller)를 설계하고 있다.

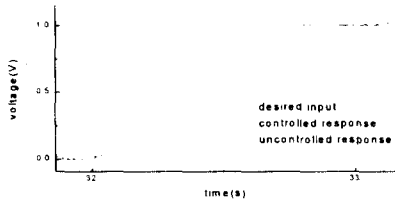


Fig. 8 Step Response

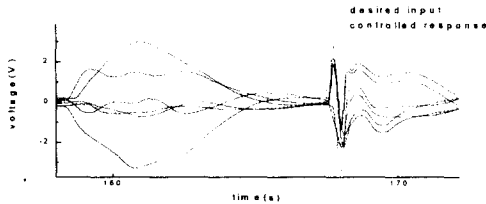


Fig. 9 Dynamic Response by Motion Cues

3. 구동 소프트웨어

3.1 프로그램 구성 및 특징

본 프로그램은 A/D와 D/A보드의 정상적인 작동상태의 확인 모드, 시스템의 각종 성능평가를 위한 기능시험 모드, 그리고 네트워크를 통해 전송된 차량 동력학 해석 결과를 바탕으로 실시간으로 시스템을 구동하는 시뮬레이션 모드로 구성되어 있다. 그림 10은 본 연구에서 개발한 구동 소프트웨어의 구성을 도시하고 있다. 각 서브프로그램은 운용자의 조작 편의를 위하여 GUI(Graphic User Interface)로 구성하였다.

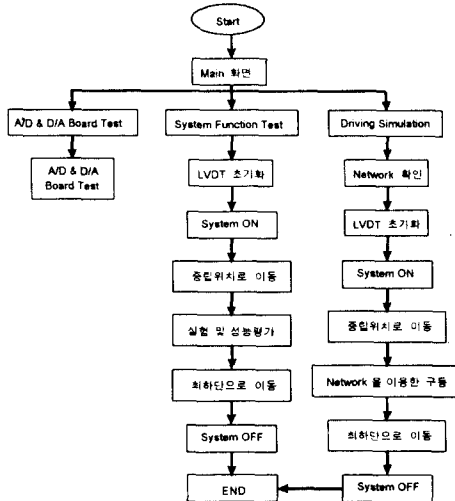


Fig. 10 Operating Software Functional Flow

3.2 서브 프로그램

3.2.1 DAQ 시스템 시험 모듈

운동시스템을 구동하기 전에 DAQ (Data Acquisition) System의 이상유무를 점검하기 위한 기능이다. A/D 보드는 유압컨트롤 패널에 입력된 전압과 PC에 입력된 전압을 비교하고, D/A 보드는 PC로부터 출력된 전압과 유압컨트롤 패널에서의 전압을 상호 비교하므로써 DAQ의 이상유무를 판단한다.

3.2.2 시스템 성능평가 시험 모듈

- LVDT 초기화

LVDT가 갖는 전기적인 오차 전압을 보정하기 위한 기능으로 유압시스템을 작동시키기 전에 최하단에 위치해 있는 운동플랫폼의 위치값을 여러번 검출하여 평균값을 구한 후, 실제 원하는 위치값인 기준 전압과의 차를 구해 LVDT 오차전압을 구한다. 이후 모든 구동프로그램에 이때 계산된 오차전압이 보정되어 적용된다.

- 유압시스템 구동

LVDT의 오차전압이 보정되면 유압펌프를 가동시키고 A/D 보드의 아날로그 채널을 이용해 전압을 서서히 증가시켜, 공급압력을 1300psi로 유지한다. 유압시스템의 구동이 완료되면 각종 성능시험 및 시뮬레이션을 위해 운동플랫폼을 중립위치로 서서히 이동시킨다.

- 실험 및 성능평가

운동시스템의 성능을 평가하기 위해 계단과 입력, 속도 및 가속도 시험과 주파수 응답시험 등을 할 수 있도록 프로그램을 작성하였으며, 이때 발생하는 각종 데이터를 저장할 수 있도록 구성하였다. 또한 유압시스템에 부착되어있는 필터, 실린더의 상·하한 한계스위치, 히터 및 쿨러 등의 각종 센서 신호를 검출할 수 있도록 하였다. 만일 구동 중에 각 실린더가 미리 설정되어 있는 운동 영역 범위를 벗어나거나 유압시스템에 이상이 발생하였을 경우에는 펌프를 정지시키고, 운동플랫폼을 최하단으로 서서히 이동시키는 이상상태 처리루틴을 수행한다. 모든 실험이 끝난 후 운동플랫폼을 최하단으로 이동시키며, 공급압력을 0psi로 조절한 후 유압펌프를 정지시킨다. 그림 11과 12는 LabVIEW로 작성된 프로그램의 조작 패널과 내부 구동회로를 도시하고 있다.

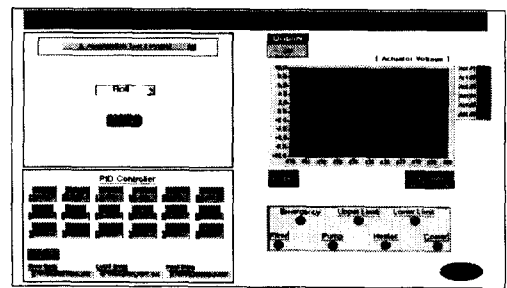


Fig. 11 Control Console for Functional Test

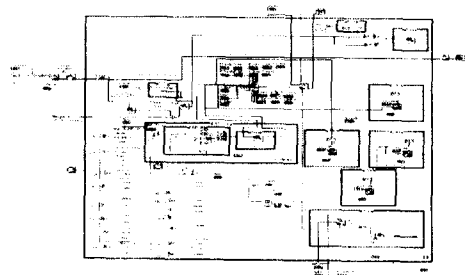


Fig. 12 Block Diagram for Functional Test

3.2.3 차량 시뮬레이션 모듈

- 통신모듈 및 네트워크

차량 동역학 해석을 수행하는 컴퓨터에서 전달되는 각종 입출력 신호는 네트워크를 통해 전달하게 되는데, 데이터의 전송을 위하여 10Mbps의 전송속도를 갖는 Ethernet 카드와 UDP (User Datagram Protocol)를 사용하였으며, 네트워크 통신을 위해 LabVIEW에서 지원하는 통신 모듈을 사용하여 프로그래밍하였다. UDP는 고속으로 데이터를 전송할 수 있으며, 1 대 다수의 통신에 특히 유리한 방법이다. 전송된 데이터가 오류 없이 정확하게 전달되었는지를 확인하지 않는다는 단점을 가지고 있으나, 아주 작은 스텝마다 반복적으로 데이터를 전송하는 실시간 시스템에서 시간적 구속 요건을 만족시키는 것이 오류의 검증 절차 보다 더 중요하므로 일반적으로 차량 시뮬레이터에서는 UDP를 사용한다. 네트워크 초기화를 위하여 가상의 통신 소켓을 만든 후 네트워크의 종류 및 포트번호, 주소 등을 확인하는 절차를 거친다. 네트워크 초기화는 시스템을 원활히 구동하기 위한 일종의 시작 신호이며, 이 과정을 통해 각 서브시스템의 구동 준비 여부를 확인할 수 있다. 그림 13은 운동시스템 구동을 위해 LabVIEW를 이용하여 작성된 통신모듈의 구성을 보이고 있다.

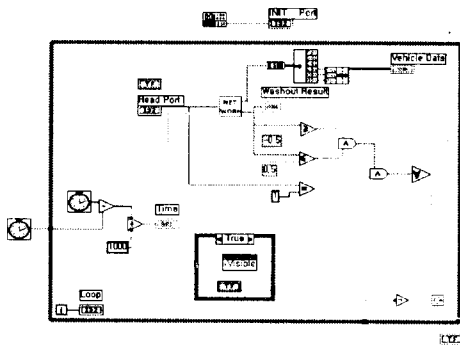


Fig. 13 Communication Module

- 시뮬레이션 수행

이상과 같이 구성된 구동 소프트웨어와 각 서브시스템을 통합하여 차량 시뮬레이터를 구동하였다. 그림 14는 운전자가 탑승하여 주행 시뮬레이션을 수행하는 차량 시뮬레이터 운동시스템의 구동 상태를 보이고 있다.

운동시스템을 통해 운전자에게 실제 차량의 운동과 유사한 느낌을 갖도록 하기 위하여 시스템을 직접 구동하면서 Washout 알고리즘의 각 파라미터를 튜닝하였다. 시각시스템의 주행시나리오에 따른 운전자의 주행조작을 통해 slalom 및 J-Turn 운동을 테스트 해본 결과, 운동시스템이 운동 큐를 실시간으로 적절히 재현하는 것을 볼 수 있었다. 그림 15는 네트워크를 이용해 운동시스템을 구동하기 위해 LabVIEW를 이용하여 작성된 프로그램의 구성을 도시하고 있다.



Fig. 14 Driving Simulator in Operation

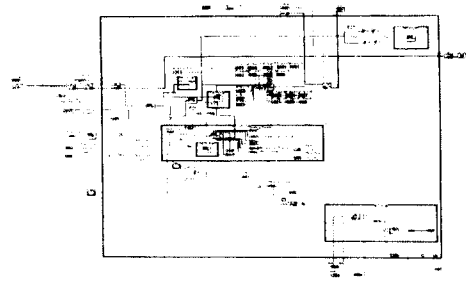


Fig. 15 Block Diagram for Driving Simulator Operation

- 안전도 및 위급상황 처리

유압장치를 사용하는 운동시스템은 이상상태 발생시에 안전하게 대처할 수 있는 위급상황 처리 기능을 필수적으로 포함하고 있어야 한다. 운동시스템 작동시 설정된 운동범위를 벗어나거나, 네트워크를 통한 정상적인 데이터 교환 유무 및 유압시스템의 각종 센서 신호를 주기적으로 감시한다. 만일의 오류가 발생시 이상상태 처리 알고리즘을 수행, 유압펌프를 정지시키고 서보밸브에 약간의 전압을 인가하여 운동플랫폼의 하중에 의해 서서히 최하단으로 이동하도록 프로그래밍하였으며, 초기 유압펌프 가동시부터 펌프 정지시까지 모든 상황에 능동적으로 대비할 수 있도록 하였다.

4. 결론

차량 시뮬레이터의 운동시스템을 구동하기 위하여 LabVIEW 4.0 프로그램을 기본 환경으로 하는 구동 소프트웨어를 개발하였으며, 네트워크를 이용하여 차량 동역학 해석에 의해 생성된 운동 큐를 운동시스템에서 실시간으로 재현하였다. 차량 동역학 해석 결과를 제한된 운동영역 안에서 충실히 모사하기 위하여 차량 운동의 선형 가속도와 각속도를 인체감각이 감지할 수 있도록 고주파 필터링하고, Tilt Coordination을 통해 지속적인 차량의 횡가속도를 재현하기 위한 Washout 알고리즘을 개발 적용하였다. 시스템의 안전성을 확보하기 위하여 위급상황 발생시 적극적으로 대처할 수 있는 이상상태 처리 알고리즘을 개발 적용하였다. 또한 실험을 통해 PID 제어기의 제어이득을 조정하여 응답성능을 향상시켰으며, 동작 조건과 부하외란의 변화에도 원하는 제어계적을 만족하는 적응제어기의 설계 적용을 추진하고 있다.

참고문헌

- [1] 이윤성, 김정하, 조준희, "실시간 차량시뮬레이터 개발," '97 춘계 한국자동차공학회 학술대회 논문집 Vol. 2, pp. 464-469.
- [2] 박경관 외, "차량시뮬레이터의 운동시스템 개발," '97 춘계 한국자동차공학회 학술대회 논문집 Vol. 2, pp. 488-493.
- [3] LabVIEW User Manual, National Instruments, 1996.
- [4] 전용민, "6자유도 비행시뮬레이터용 구동 신호처리에 관한 연구," 인하대학교 공학석사학위 논문, 1994.
- [5] 김정하 외, "병렬구조형 1/2 Size 차량운전 모사장치의 설계 및 특성연구," 제어계측·자동화·로보틱스 연구회 합동발표회 논문집, pp. 49-53, 1997.
- [6] E.F.Fichter, 1986, "A Stewart Platform-based Manipulator General theory and Practical construction", Int. J. of Robotics Research, Vol.5, No.2, pp.157~182.
- [7] Gene F. Franklin, J. David Powell, A. Emami-Naeini, Feedback Control of Dynamic Systems 3rd Edition, 1994.