

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3



InnoVR Motion SDK Manual

Rev (1.3)



Restricted Distribution – All data or information contained in or disclosed by this document are proprietary and confidential information of INNOSIMULATION Co., Ltd. and all rights therein are expressly reserved. By accepting this material, the recipient agrees that this material and the information contained therein are held in confidence and in trust and will not be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of INNOSIMULATION Co., Ltd. Information in this document is subject to change and does not represent a commitment on the part of INNOSIMULATION Co., Ltd.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

문서 개정 내역

변경일	버전	변경 내용	비고
2017.09.01	1.0.0	모션 SDK 매뉴얼 초안	1 차 릴리즈
2018.03.01	1.1.0	모션 필터 객체 추가	
2018.08.01	1.2.0	Telemetry 모션 플레이어 와 플러그인 추가	
2019.04.01	1.3.0	모션 플레이어 추가	

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

목 차

1	문서 개요	9
1.1	개요	9
1.2	VR 모션 시스템.....	10
1.3	VR 멀미(MOTION SICKNESS).....	11
1.4	VR 모션 플랫폼.....	11
2	VR 모션 SDK	13
2.1	SDK 구성.....	13
2.1.1	가이드 문서(DOC).....	15
2.1.2	모션 샘플 데이터.....	16
2.1.3	모션 편집기(InnoME)	16
2.1.4	모션 제어 프로그램(InnoMP).....	18
2.1.5	모션 라이브러리 (InnoML).....	19
2.2	환경 설정	21
2.2.1	장치 연결	21
2.2.2	IP 설정	21
2.2.3	모션 제어 실행	22
2.3	요구사항 및 설정	23
2.3.1	하드웨어 요구사항	23
2.3.2	소프트웨어 요구사항	24
2.3.3	비주얼 스튜디오 설정 (C/C++)	24
2.3.4	비주얼 스튜디오 설정 (C#)	25
3	모션 라이브러리	26
3.1	모션 라이브러리 소개	27
3.1.1	개요	27
3.1.2	모션 신호	27
3.1.3	모션 기본 개념	28
3.1.4	모션 데이터 (PCM).....	29
3.2	모션 버퍼(IMBUFFER)	30
3.2.1	버퍼 생성	31
3.2.2	버퍼 채우기(enqueue)	32
3.2.3	버퍼 읽기(dequeue)	33
3.2.4	주요 API.....	33
3.3	모션 컨텍스트(IMCONTEXT).....	34
3.3.1	모션 장치 연결	34

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

3.3.2	모션 장치 상태	37
3.3.3	모션 장치 제어	38
3.3.4	모션 중첩(Superposition)	41
3.3.5	주요 API.....	43
3.4	모션 소스(IMSOURCE)	44
3.4.1	소스 생성	45
3.4.2	소스 버퍼 설정	46
3.4.3	재생 제어	47
3.4.4	모션 버퍼 참조	48
3.4.5	주요 API.....	48
3.5	모션 입력(IMINPUT)	49
3.5.1	입력 생성	49
3.5.2	입력 버퍼 설정	50
3.5.3	입력 스트리밍.....	51
3.5.4	입력 샘플링	51
3.5.5	사용자 입력	52
3.5.6	주요 API.....	53
3.6	모션 필터(IMFILTER)	54
3.6.1	필터 생성	55
3.6.2	필터 빌드	57
3.6.3	필터 처리	58
3.6.4	워시아웃 필터.....	58
3.6.5	사용자 필터	60
3.6.6	주요 API.....	62
3.7	샘플 프로젝트.....	63
3.7.1	프로젝트 열기.....	63
3.7.2	IMotion 시작하기	67
3.7.3	InnoML 시작하기	70
4	모션 플러그인.....	73
4.1	유니티 플러그인	74
4.1.1	Motion Context Component	74
4.1.1.1	컴포넌트 설정 값	75
4.1.1.2	주요 프로퍼티	75
4.1.1.3	주요 함수	75
4.1.2	Motion Source Component.....	76
4.1.2.1	컴포넌트 설정 값	77
4.1.2.2	주요 함수	77
4.1.3	Motion Input Component	78

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1.3.1	컴포넌트 설정 값	79
4.1.3.2	주요 함수	80
4.1.4	Motion Filter Component.....	81
4.1.4.1	컴포넌트 설정 값	81
4.1.5	샘플 프로젝트.....	82
4.1.6	플러그인 패키지	83
4.1.7	주의사항.....	83
4.2	언리얼 플러그인	84
4.2.1	Motion Context Actor Component	84
4.2.1.1	컴포넌트 에디터 설정 값	85
4.2.1.2	주요 블루프린트 함수.....	85
4.2.2	Motion Source Actor Component.....	86
4.2.2.1	컴포넌트 에디터 설정 값	87
4.2.2.2	주요 블루프린트 함수.....	87
4.2.3	Motion Input Actor Component.....	88
4.2.3.1	컴포넌트 에디터 설정 값	89
4.2.3.2	주요 블루프린트 함수.....	90
4.2.4	Motion Filter Actor Component	91
4.2.4.1	컴포넌트 에디터 설정 값	91
4.2.4.2	주요 블루프린트 함수.....	92
4.2.5	샘플 프로젝트.....	92
4.2.6	주의사항.....	93
5	모션 플레이어	94
5.1	텔레메트리(TELEMETRY).....	95
5.1.1	게임 텔레메트리	96
5.1.2	InnoMI	96
5.2	시뮬레이션 프로파일	97
5.2.1	시스템 설정	98
5.2.2	모델 설정	99
5.2.3	모션 필터 컴포넌트	101
5.3	모션 프레임워크	102
5.3.1	그래픽스 시스템	102
5.3.2	네트워크 시스템	103
5.3.3	모션 시스템	104

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

그 림 목 차

그림 1. VR 모션 시스템 구성도	10
그림 2. 모션 플랫폼 자유도 (DOF).....	12
그림 3. VR 모션 플랫폼	12
그림 4. VR 콘텐츠 연동	13
그림 5. VR 모션 SDK 구성	14
그림 6. API Reference (InnoML.chm)	15
그림 7. 모션 편집기 화면 구성	17
그림 8. 4-DOF 모션 장치 제어 용 모션 편집기 설정	17
그림 9. 모션 제어 프로그램 화면 구성	18
그림 10. 모션 Player 화면 구성	19
그림 11. 모션 플랫폼 연결.....	21
그림 12. 네트워크 IP 설정 예 (모션 제어 PC, 모션 플랫폼)	22
그림 13. IMotion_Testbed.....	23
그림 14. 프로젝트 설정	25
그림 15. C# 개발 환경	26
그림 16. 모션 SDK 아키텍처	26
그림 17. 디지털 신호 처리.....	27
그림 18. 모션 신호 처리 과정	27
그림 19. 샘플링과 양자화 예시	28
그림 20. 모션 데이터 (모션 파일과 PCM 데이터)	30
그림 21. 모션 버퍼 주요 기능.....	31
그림 22. 모션 버퍼 큐잉 (좌: 인큐, 우: 디큐)	33
그림 23. 모션 컨텍스트 주요 기능.....	34
그림 24. 모션 상태 머신	37
그림 25. 모션 신호 동기화.....	39
그림 26. 역기구학 개념	39
그림 27. 모션 중첩 처리 구조	42
그림 28. 모션 파형의 합성.....	43
그림 29. 모션 소스 주요 기능	45
그림 30. 모션 파형의 진폭과 주기	45
그림 31. 모션 소스 재생 파라미터 (1x, 2x, 4x)	46
그림 32. 모션 입력 주요 기능	49
그림 33. 실시간 샘플링 방식	50
그림 34. 모션 입력 샘플링 처리 흐름.....	51
그림 35. 모션 필터 주요 기능	54
그림 36. 워시아웃 필터 구성도	59
그림 37. High Pass Filter에 의한 입력 변화	60
그림 38. LowPass Filter에 의한 입력 변화와 Tilt 적용 개념도	60
그림 39. 모션 샘플 프로젝트	64
그림 40. IMotion 샘플 파일들.....	65
그림 41. IMotion_Testbed 화면 구성	66
그림 42. InnoML 샘플 파일들.....	66

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

그림 43. Motion Context Component.....	74
그림 44. Motion Source Component	76
그림 45. Motion Input Component	78
그림 46. Motion Filter Component.....	81
그림 47. 모션 필터 에셋의 생성 방법.....	81
그림 48. Script Execution Order Example	83
그림 49. Motion Context Actor Component.....	84
그림 50. Motion Source Actor Component.....	86
그림 51. Motion Input Actor Component	88
그림 52. Motion Filter Actor Component.....	91
그림 53. 모션 필터 데이터 에셋의 생성 방법.....	91
그림 54. Telemetry 지원 모션 시스템 구성	95
그림 56. 워시아웃 및 텔트 보정 필터.....	102
그림 57. 모션 플레이어 화면 구성.....	103

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

표 목 차

표 1. 모션 SDK 패키지 구성.....	14
표 2. 모션 라이브러리 구성.....	20
표 3. 모션 데이터 용어 정리.....	28
표 4. Format 구조체 (IM_FORMAT).....	32
표 5. Buffer 주요 API.....	33
표 6. Description 구조체 (IM_DEVICE_DESC).....	35
표 7. 컨텍스트 옵션 (IM_CFG_*)	36
표 8. 에러 코드.....	40
표 9. Diagnostic 구조체 (IM_DIAGNOSTIC_AXIS_INFO)	41
표 10. Context 주요 API.....	43
표 11. Source 주요 API.....	48
표 12. Input 주요 API	53
표 13. 필터 처리기 (IM_FILTER_*)	55
표 14. Filter 주요 API.....	63
표 15. IMotion 샘플 예제 목록.....	65
표 16. InnoML 샘플 예제 목록.....	66
표 17. Motion Context 컴포넌트 설정 값.....	75
표 18. Motion Context 컴포넌트 프로퍼티	75
표 19. Motion Context 컴포넌트 함수.....	75
표 20. Motion Source 컴포넌트 설정 값	77
표 21. Motion Source 컴포넌트 함수.....	77
표 22. Motion Input 컴포넌트 설정 값.....	79
표 23. Motion Input 컴포넌트 함수	80
표 24. Motion Filter 컴포넌트 설정 값	81
표 25. 유니티 플러그인 샘플 폴더.....	82
표 26. 유니티 플러그인 샘플 씬	82
표 27. Motion Context Actor 컴포넌트 에디터 설정 값.....	85
표 28. Motion Context Actor 컴포넌트 함수.....	85
표 29. Motion Source Actor 컴포넌트 설정 값	87
표 30. Motion Source Actor 컴포넌트 함수	87
표 31. Motion Input Actor 컴포넌트 설정 값	89
표 32. Motion Input Actor 컴포넌트 함수	90
표 33. Motion Filter Actor 컴포넌트 설정 값	91
표 34. Motion Filter Actor 컴포넌트 함수	92
표 35. 언리얼 플러그인 샘플 폴더.....	92
표 36. 언리얼 플러그인 샘플 맵	92

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

1 문서 개요

1.1 개요

본 문서는 이노시뮬레이션의 모션 시스템을 구성하는 모션 플랫폼과 모션 제어 SW의 개념, 그리고 사용 절차 및 규격에 대하여 서술하고 있으며, 특히 VR 콘텐트에 모션 플랫폼을 연동하여 개발할 수 있는 VR 모션 라이브러리에 관하여 설명합니다.

모션 라이브러리는 다양한 모션 플랫폼의 추상화(HAL)를 지원하며 정밀 모션 제어가 가능한 영상용 라이브러리(IMotion)와 게임 등의 실시간 인터랙션이 강화된 라이브러리 (InnoML), 그리고 최근 VR 엔진으로 각광 받고 있는 유니티 3D 등의 모션 제어 플러그인으로 구성 됩니다.

이노시뮬레이션 모션 라이브러리의 신호 처리(Buffer), 장치 관리(Context), 모션 효과(Source), 모션 입력(Input), 모션 필터(Filter)의 기본 개념을 설명합니다. 이노 모션 라이브러리 (InnoML)에서 그 구조를 가장 잘 표현하고 있으므로, 본 문서는 InnoML의 5 가지 모션 객체를 중심으로 기본 개념을 설명합니다.

본 문서의 구조는 다음과 같습니다.

✓ 개요

VR 모션 SDK를 구성하는 모션 미들웨어, 모션 제어 프로그램 및 편집기를 소개합니다. VR 모션 시스템의 소개와 기본 개념, 그리고 콘텐츠 연동 방법을 모션 라이브러리 사용 중심으로 설명합니다.

✓ SDK 구성

어플리케이션 및 컴포넌트 개발을 위한 개발 환경 구축 방법 및 샘플을 통한 통합 연동 예제를 설명합니다.

✓ 모션 라이브러리

모션 라이브러리의 역할과 기능을 설명하고, 어플리케이션 및 컴포넌트 개발에 필요한 API 사용 방법을 예시와 함께 설명합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

1.2 VR 모션 시스템

VR 모션 시뮬레이션 시스템의 유효성을 결정하는 현실감을 확보하기 위하여 각 서브시스템간의 정보 및 데이터를 동적으로 상호 교환하고, 동기화 하는 전체 시스템의 관점에서 관리하는 시스템 통합 기술은 가상현실에서 중요하게 고려되어야 합니다.

VR 모션 시스템은 모션 플랫폼의 정밀 제어의 단순한 목적을 넘어, 가상현실 환경에서 인간에게 현실감을 줄 수 있는 체감형 장치의 하나로 가상 세계의 일부가 될 수 있는 하나의 요소 기술로 쉽게 융합 될 수 있어야 합니다. 즉, VR 모션 시스템은 기존 모션 시스템의 하드웨어 장치 제어의 물리적(Physical) 관점 보다, 가상 세계를 시뮬레이션 하는 논리적(Logical) 관점에서 설계 되어야 합니다.

현실감은 사용자에게 전달되는 각종 큐(Cue)의 견실도에 좌우되며, 사용자의 제어 응답과 큐를 생성하는 각 서브시스템간의 관계에 의해 결정됩니다. VR 멀미(Motion Sickness) 또한 각종 큐의 불일치에서 오는 인간의 인지 부조화에 의해 발생 된다고 할 수 있습니다. 즉, VR 모션 시스템의 유효성은 각종 시스템의 특성과 체험 조건 변화에 대해 사용자가 모션 시뮬레이터에서 얼마나 실제와 유사하게 느끼느냐에 좌우되며, 최대한 현실감을 보장하기 위해서는 각 서브시스템의 성능 향상과 함께 하나의 시스템 관점에서 주요 구성 요소들이 사용목적에 맞도록 통합 관리되어야 합니다.

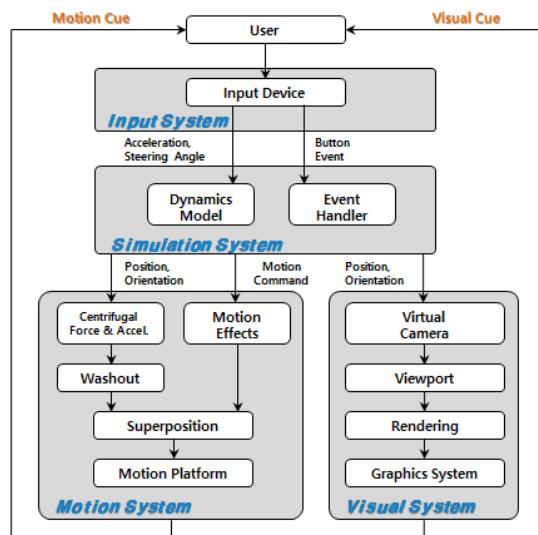


그림 1. VR 모션 시스템 구성도

VR 모션 시스템을 기능별로 분류하면 그림과 같이 3 가지로 간략히 구분할 수 있습니다.

- 물리(동역학) 모델이나 상호작용을 처리하는 실시간 시뮬레이션 시스템
- 사용자가 실제 느끼는 운동, 환경 및 소음을 각각 재현 하는 운동, 시각 및 음향 시스템
- 사용자와 시뮬레이터의 인터페이스 역할을 하는 입력 시스템

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

가상현실은 인간과 가상 세계를 연결하는 여러 요소 기술들의 융합 기술이고, 사용자의 입력 신호에 따라 시뮬레이션 된 가상 세계의 정보는 가시화 이미지와 모션의 움직임으로 다시 사용자에게 전달됩니다.

1.3 VR 멀미(motion sickness)

VR 모션 플랫폼을 체험하는 동안, 인간의 전정기관(vestibular system)은 평형감각(heave, roll, pitch)을 감지합니다. 우리는 눈으로 surge, sway, heave 및 roll, pitch, yaw 를 감지하고, 그에 따라 위가 반응합니다. 전정기관에서 제공하는 평형 신호들과 시각 시스템의 모션 신호 사이의 불일치가 멀미의 원인이 될 수 있습니다.

다행히 모든 사람이 멀미로 고통 받는 것은 아니지만, 그것이 가지고 있는 영향을 무시할 수는 없고, 작은 아이디어가 그 효과를 최소화 할 수 있습니다.

모션 플랫폼은 전정기관이 운동감을 느끼도록 하여 시각 시스템의 모션 신호와의 차이를 줄이는데 도움을 줄 수 있습니다.

멀미를 줄이고 몰입도를 높이기 위해 시각 신호(Visual cue)과 모션 신호(Motion Cue)의 동기화 기술과 모션 정밀 제어 기술이 중요합니다.

1.4 VR 모션 플랫폼

모션 플랫폼은 다른 말로 모션 베이스, 모션 의자, 모션 시뮬레이터 등으로 불립니다. 모션 플랫폼의 주된 목적은 사용자에게 실제와 같은 운동감을 전달해 주는 것입니다. 운동은 시각적 이미지와 동기화되며, 촉각이나 비디오 게임의 한 요소, 시뮬레이션 또는 가상현실 분야에서 사용됩니다. 모션이 음향과 시각적 이미지와 함께 동기화 될 때 시각, 청각, 촉각의 조화를 이루게 됩니다.

모션 플랫폼은 4D 효과의 가장 큰 부분을 차지하는 운동감을 재현하는 장비로, 가상현실 기술과 결합하여 인간에게 적절한 운동감을 제공함으로써 사실적인 사용자 경험(User Experience)을 극대화할 수 있도록 합니다.

이러한 대표적인 예가 비행기 시뮬레이션 또는 자동차 시뮬레이션으로 3 개의 축을 따라 수평 이동과 회전이 가능합니다. 이 같은 운동을 자유도 (DOF)라고 부르는데, 일반적인 모션 플랫폼은 6 자유도의 운동을 갖는 Stewart 모션 플랫폼입니다. 그림에서 보이는 것처럼, 첫 번째 축은 앞으로, 두 번째 축이 위쪽으로, 세 번째 축이 옆으로 향하도록 3 개의 축을 표시 하고 있습니다. 회전(Rotation) 운동을 각각 roll, pitch, yaw 라고 하고, 각 축을 따라 병진(Translation) 운동을 각각 surge, sway, heave 라고 합니다.

surge (forward and backward translation along its x-axis)

sway (sideways translation along its y-axis)

heave (vertical translation along its z-axis)

roll (tilting rotation around the x-axis)

pitch (tilting rotation around the y-axis)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

yaw (horizontal rotation around the z-axis)

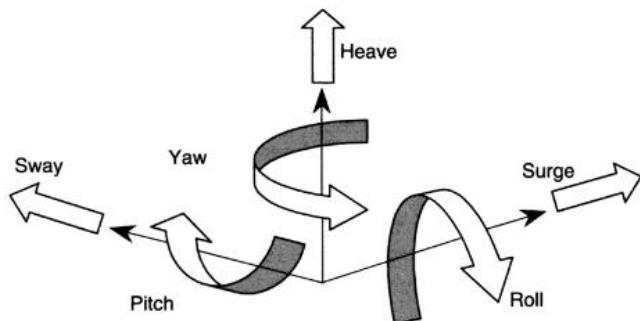


그림 2. 모션 플랫폼 자유도 (DOF)

VR 용 모션 플랫폼은 엔터테인먼트 시뮬레이터로 저가의 보급형 3 자유도(DOF) 사용이 일반적입니다. 3 개의 AC 모터로 구동되는 3 자유도 모션 플랫폼으로 구성된 VR 모션 시스템에서는 가상 객체의 운동을 모션플랫폼의 운동 범위 내에서 구동 명령에 따라 실행하여 현실감 있는 모션 큐를 재현 합니다. VR 용 3 자유도 모션 플랫폼 중 모션 시트의 경우 상하 수직 이동(heave)을 -30mm ~ 30mm 의 운동 범위를 갖습니다. 또한 좌우 회전(roll) 및 앞뒤 회전(pitch)이 -10deg ~ 10deg 의 운동 범위로 모션 플랫폼이 회전할 수 있습니다.



그림 3. VR 모션 플랫폼

모션 플랫폼은 모션 베이스(Motion Base) 라는 핵심 부품 위에 다양한 형태의 기구 부를 장착하여, 인간의 탑승 자세를 제공할 수 있도록 합니다. 모션 테이블 위에 서서 진동을 느끼거나 압력 센서를 부착하여 스노우보드 등의 게임을 즐길 수 있으며, 시트 위에 앉아서 롤러코스터나 자동차 라이딩을 체험 할 수 있습니다. 또한 눌거나 엎드려서 봅슬레이, 패러글라이딩 등의 실감나는 체험을 할 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

2 VR 모션 SDK

이번 장에서는 VR 모션 플랫폼을 제어하는 VR 모션 SDK의 소개와 구성, 환경 설정 방법에 대해 알아봅니다.

VR 콘텐츠는 최근 마케팅 용으로 많이 사용하는 360 도 파노라마 영상의 Passive VR 콘텐츠와 유니티(Unity3D)나 언리얼(Unreal)과 같은 게임 엔진 기반의 Interactive VR 콘텐츠로 크게 구분할 수 있습니다. Passive VR의 모션은 주로 영상내의 정적인 카메라 위치로부터 얻어질 수 있기 때문에, 모션 편집기(InnoME)로 미리 생성된 모션 데이터를 영상과 동기 재생(InnoMP)하여 쉽게 모션 시스템을 구성할 수 있습니다. 360 VR Player는 영상 파일을 재생할 때 마치 자막처럼 모션 파일을 재생하는 IMotion에 직접 연동하거나 모션 입력 플러그인을 통해 모션 플랫폼을 제어할 수 있습니다.



그림 4. VR 콘텐츠 연동

Interactive VR 콘텐츠의 모션 제어는 조금 더 복잡한 처리가 필요한데, 동적인 모션 객체의 위치와 자세로부터 모션 위치를 실시간 제어 하는 입력 스트리밍과 모션 효과의 재생을 사용자 인터랙션과 함께 연동해야 합니다. 이 기능 개발을 위해 InnoML의 입력 및 소스 인터페이스를 사용하거나, 입력 플러그인의 Telemetry를 통한 원격 제어를 수행할 수 있습니다.

2.1 SDK 구성

VR 모션 SDK는 모션 라이브러리와 모션 제어 프로그램, 모션 편집기, 모션 입력 플러그인으로 구성됩니다.

아래 그림과 같이 2 가지 VR 모션 시스템 구성 방법을 제공합니다. 그림 좌측 구성은 VR 콘텐츠를 모션 입력 플러그인에 연동하여 개발하고, 플러그인이 모션 Player와 이더넷 통신으로 연결되어 원격 제어하는 방법입니다. 이 연동 방법은 플러그인의 Telemetry 입력 필드를 시뮬레이션 하는 방법이기 때문에 PID 제어와 같은 모션 명령 이벤트 처리에 적합합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

그림 우측 구성은 모션 라이브러리에 직접 연동하여 VR 콘텐츠를 구성하는 방법으로 실시간 모션 플랫폼 정밀 제어에 적합 합니다. 모션 라이브러리는 게임과 같이 상호작용이 용이한 InnoML 와 모션 플랫폼 직접 정밀 제어가 가능한 IMotion 을 필요에 따라 선택하여 연동할 수 있습니다.

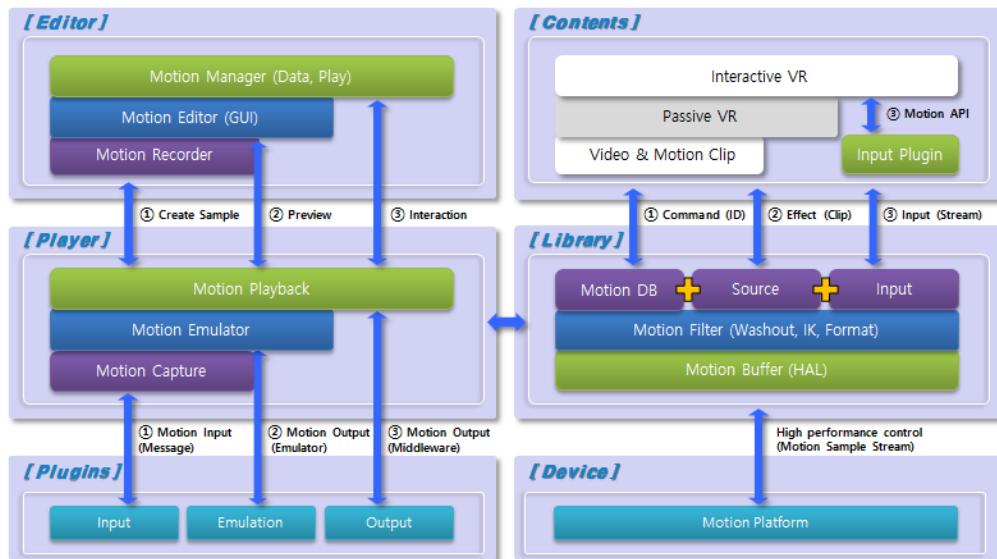


그림 5. VR 모션 SDK 구성

다운로드 받은 압축 파일(.zip) 을 풀면 SDK 설치는 완료 됩니다.

SDK 는 가이드 문서, 모션 제어 프로그램(MCP), 모션 데이터(샘플 모션), 모션 편집기, 모션 라이브러리 기본 구성과, 개발 지원을 위한 매뉴얼과 API 레퍼런스 문서, 샘플 등 추가 구성을 표 1 과 같이 구성하고 있습니다.

표 1. 모션 SDK 패키지 구성

폴더명	설명
DOC (가이드 문서)	SDK 소개 문서, 개발자 매뉴얼, API 레퍼런스 문서(doxygen), 모션 플랫폼 가이드 등의 개발 가이드 문서를 포함합니다.
MotionData (모션 샘플 데이터)	라이브러리 샘플 프로젝트에서 사용하는 기본 사인파 모션 데이터 파일과 Editor 및 Player의 샘플 모션 파일
MotionEditor (모션 데이터 편집기)	모션 데이터 생성 및 편집, 저장 기능을 하는 GUI 편집 프로그램입니다. (QEditor 설명서 참고)
MotionPlayer (모션 제어 프로그램)	저장된 모션 데이터를 모션 플랫폼에서 구동 시키는 모션 제어 프로그램입니다.
MotionLibrary (모션 라이브러리)	모션 제어 어플리케이션 개발을 위한 IMotion, InnoML의 포함 파일(include), 라이브러리 및 실행파일(lib) 등 개발 바이너리입니다.
MotionLibrary (샘플 프로젝트)	모션 라이브러리 사용 예제 소스 및 샘플 프로젝트, 테스트 프로그램 등으로 구성 됩니다. (C/C++/C#)
MotionPlugins (플러그인)	모션 입력 플러그인 바이너리, 샘플 프로젝트, 테스트 프로그램 등으로 구성 됩니다. (패키지 구성에 따라 다릅니다)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

2.1.1 가이드 문서(DOC)

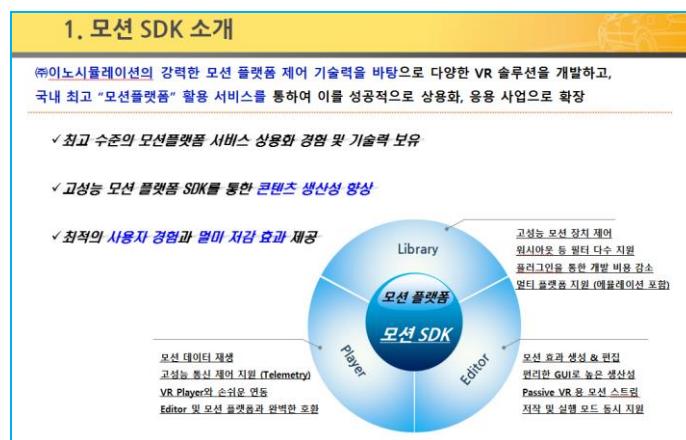
VR 모션 SDK 개발자를 위한 매뉴얼(본 문서) 및 모션 라이브러리 API 레퍼런스를 포함 합니다.

- SDK 매뉴얼

본 문서입니다.

- SDK 소개 문서 (pdf)

InnoVR 모션 SDK 를 소개 하는 쿠 가이드 문서 입니다. 본 매뉴얼의 요약을 포함하고 있으며 SDK 로드맵을 확인할 수 있습니다.



- API 레퍼런스 문서 (doxygen)

IMotion 및 InnoML 모션 라이브러리 API 인터페이스 설명과 예제를 포함 합니다. (chm 형식의 파일)



그림 6. API Reference (InnoML.chm)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

2.1.2 모션 샘플 데이터

모션 편집기의 샘플 프로젝트 및 데이터, 그리고 모션 라이브러리 샘플 프로젝트의 기본 모션 데이터를 포함 합니다.

- qep : Demo 프로젝트 파일
- csv : Demo 모션 파일 및 샘플 파일
- avi : Demo 영상 (SDK/MotionData/GreatWall/GreatWall.avi)

2.1.3 모션 편집기(InnoME)

모션 데이터 생성 및 편집, 저장 기능을 하는 GUI 편집 프로그램 입니다. (QEditor 설명서 참고)

모션 편집기는 주로 다음 2 가지 목적으로 사용 됩니다.

- Passive VR 용 모션 데이터 파일 생성
- Interactive VR 용 모션 효과 파일 생성 (모션 DB)

모션 샘플은 콘텐츠에서 모션 라이브러리를 통해 직접 전달되는 모션 각축의 운동 범위 값의 모션 데이터입니다. 그리고 모션 버퍼는 시간 축에 따라 미리 생성된 모션 샘플의 집합으로, 모션 편집기에 의해 생성됩니다. (모션 데이터와 관련 상세한 설명은 3.1.4 모션 데이터를 참고 하세요.)

모션 편집기는 순간적인 운동 이벤트(충돌, 포격)나 정적인 운동감을 위한 운동 효과를 미리 생성할 수 있는 편리한 저작 환경을 제공 합니다. 모션 버퍼는 모션 조건(강도, 주기)에 따라 실행되고 다른 모션 신호와 혼합되어 모션 플랫폼에 전달됩니다.

모션 편집기를 통해 다음 과정으로 모션 효과 데이터를 쉽게 생성할 수 있습니다.

- VR 영상 콘텐츠 준비

모션 데이터 생성을 위해 영상 파일이 mp4 인 경우 코덱 설치 후(ex. StarCodec) 동영상 재생이 가능합니다.

- 프로젝트 열기

배포된 프로젝트 파일(.qep) 파일을 열고, 다운 받은 샘플 영상을 불러 옵니다. 영상과 동기화된 모션 데이터를 생성 합니다.

- 모션 연동

편집기에서 작업 중인 모션 데이터를 모션 플랫폼에 연동하여 확인할 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

모션 플랫폼에 연결하기 위해 우측 상단의 Wi-Fi 모양의 버튼()을 클릭합니다. 성공적으로 장치에 연결되었을 경우 아이콘이 활성화 되고, 모션이 중립 위치로 이동합니다. 연결 실패 메시지가 나오는 경우 본 문서의 2.2 환경 설정에 따라 장치 연결을 확인해야 합니다.

모션 플랫폼 실행 준비가 완료 되었으면, Wi-Fi 버튼 옆에 우측 화살표() 버튼을 클릭합니다. 아이콘이 활성화 되고, 모션이 편집된 데이터에 따라 이동합니다. 모션 데이터 편집을 다시 수행하기 위해서는 모션 제어를 정지()하고 재시작 해야 합니다.

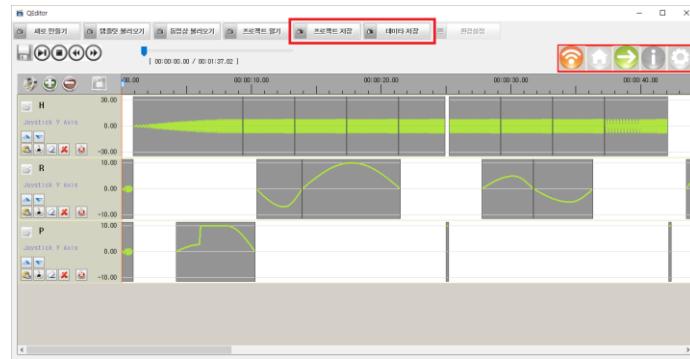


그림 7. 모션 편집기 화면 구성

- 프로젝트 저장

작업 완료 후 저장은 현재 작업 환경을 저장 하는 프로젝트 저장과 실제 모션 플레이어에서 사용 하는(실제 모션 구동 하는 파일) 데이터 저장으로 나눠지며, 모션 데이터 파일은 모션 라이브러리 또는 모션 제어 프로그램에서 실행 할 수 있습니다.

- 4DOF 모션 연동

4 축 모션 파형을 생성하고 제어하기 위해 Yaw 채널은 “레이어 속성>>최대 입력값”을 1 로 설정 하여 파형을 생성 합니다. 다음 모션 연동을 위해 아래 그림과 같이 환경 설정 대화 상자에서 모션 장치 번호를 3 으로 지정 합니다. 그리고 “인자값 개수”와 “[4]인자값”을 각각 “4 Asix”와 “[04] Yaw”로 설정 합니다. (**※주의, 장치 번호 3 설정은 4-DOF 모션 장치 프로파일을 생성 합니다. 개발용 임시 버전으로 향후 기준 장치와 통합 될 수 있습니다.**)

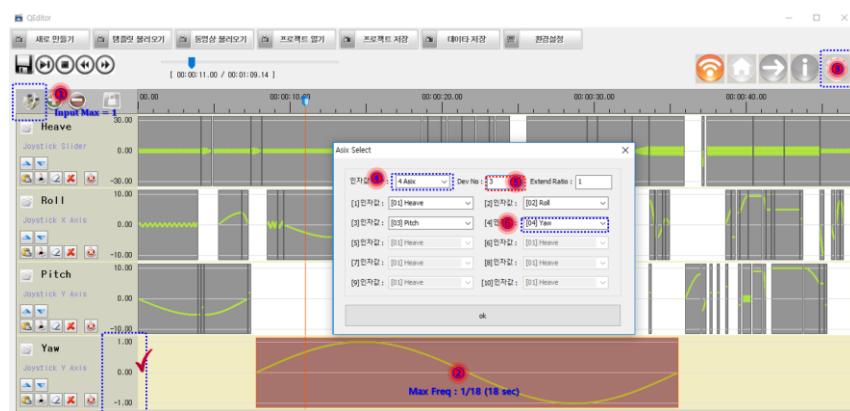


그림 8. 4-DOF 모션 장치 제어 용 모션 편집기 설정

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

2.1.4 모션 제어 프로그램(InnoMP)

저장된 모션 데이터를 모션 플랫폼에서 구동 시키는 모션 제어 프로그램입니다.

VR 어플리케이션이 모션을 적용하기 위해 모션 라이브러리를 링크하여 빌드 되어야 하지만, 모션 프로그램을 통해 쉽게 모션 시스템을 구성할 수 있습니다.

모션 Player는 주로 다음 2 가지 목적으로 사용 됩니다.

- 모션 플랫폼 직접 제어 및 상태 모니터링 (IMotion_Testbed.exe)
- 게임 등 응용프로그램의 Telemetry 데이터 실시간 시뮬레이션 (InnoMP.exe)

모션 시스템은 콘텐츠에서 전달하는 모션 신호를 받아서 모션 플랫폼에 전달하는 일련의 과정으로 볼 수 있는데, 발생된 모션 신호를 처리하는 프로그램을 통해 쉽게 모션을 제어 할 수 있습니다. 즉, 장치 연결부터 제어와 진단까지 독립적인 프로그램에서 수행 하고, 실시간 모션 신호는 네트워크를 통한 Telemetry 와 같은 특정 메시지에 따라 모션을 동작 시킬 수 있습니다. 이때 모션 입력 플러그인이 유용하게 사용됩니다.

- 모션 제어 프로그램 (IMotion_Testbed)

모션 장치 번호(IP:11)와 모션 장치 타입을 선택한 후, 열기(Open) 버튼을 눌러 모션 장치 연결을 수행 합니다. 모션 장치 연결에 실패 한 경우, 환경 설정 절차에 따라 장치 연결을 수행 합니다.

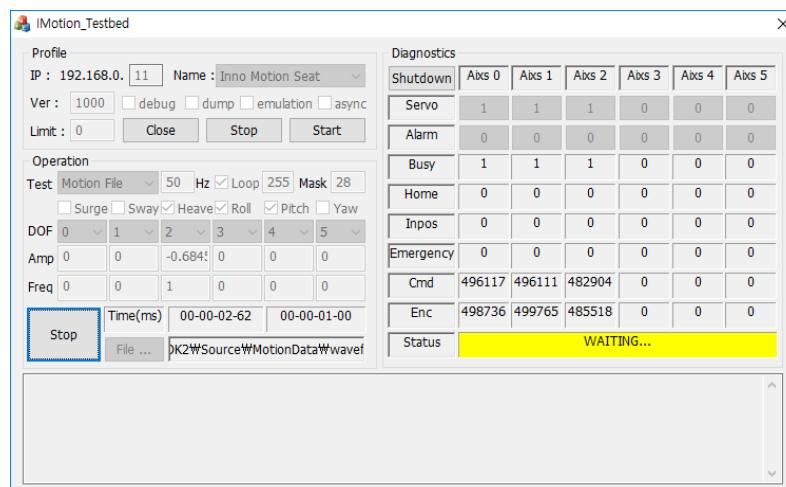


그림 9. 모션 제어 프로그램 화면 구성

File 버튼을 클릭 한 후 저장된 데이터 파일을 선택 하여 데이터를 불러 옵니다. 정상적으로 Load가 완료 되었을 경우 파일 이름(File Name)이 출력 됩니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

(IMotion_Testbed 환경 설정에 관한 자세한 사항은 2.2.3 모션 제어 실행을 참고 하세요.)

- 모션 Player (InnoMP)

모션 플레이어는 Telemetry 데이터를 UDP 및 IPC 연결로 실시간 모션 시뮬레이션하는 프로그램입니다. 현재까지 상용 레이싱 게임 10 종 및 동영상 Player 연동을 지원하고, 모션 라이브러리의 IPC 출력 및 모션 플러그인(유니티, 언리얼)과 연동을 지원합니다. 기존 PC 뿐만 아니라 모바일 기기 등과 같은 타 플랫폼에서도 모션 플러그인을 연결하기만 하면 쉽게 모션 제어가 가능합니다.

모션 Player 를 실행하고 Telemetry 데이터의 시뮬레이션 프로파일 (MotionData/Profile/*.ini)을 로드해야 합니다. 프로파일을 프로그램 창위에 드래그 앤 드롭하거나, 마우스 우측 클릭으로 프로파일을 선택 합니다. Telemetry 자원 프로그램이 실행 중이라면 아래 화면과 같이 수신된 데이터를 실시간 파형 그래프로 표시합니다. 화면 구성은 Profile 정의에 따라 변경되며, 아래 그림에서는 좌측부터 Telemetry 신호 12 채널, 모션 입력 신호 6 채널, 모션 장치 출력 신호 3 채널을 실시간으로 표시 합니다.

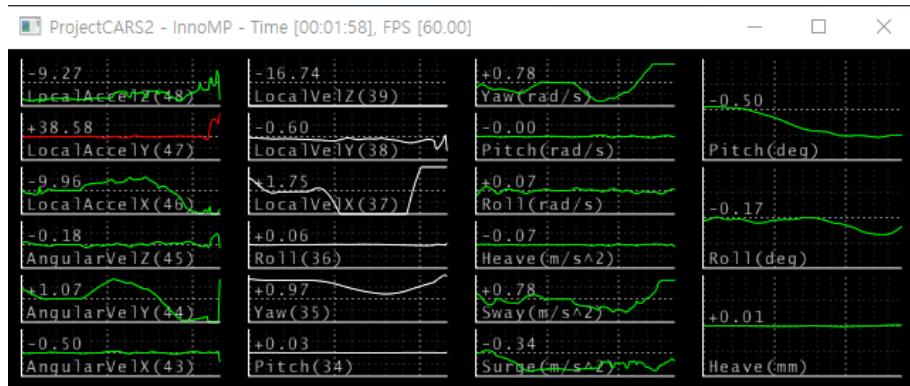


그림 10. 모션 Player 화면 구성

모션 시뮬레이션은 Telemetry 수신, 입력 축 매핑, 출력 필터의 3 단계로 수행 됩니다. 수신 단계는 UDP/IPC 연결과 수신 패킷으로부터 축 선택이 수행되고, 축 매핑 단계에서는 수신 축과 값을 적절히 변환하여 모션 입력의 각 축으로 매핑 합니다. 마지막, 필터링에서는 Force Simulation 모드일때 워시아웃 필터 등을 적용하여 최종 모션 제어 신호를 시뮬레이션 합니다. (모션 Player 에 관한 자세한 사항은 5 장 모션 Player 를 참고 하세요.)

2.1.5 모션 라이브러리 (InnoML)

모션 라이브러리는 영상 콘텐츠 및 모션 정밀 제어를 위한 라이브러리(IMotion)와 게임 같은 실시간 시뮬레이션을 지원하는 라이브러리(InnoML)를 제공 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

모션 라이브러리는 개발용 헤더 파일 및 정적 링크 라이브러리(.lib)와 실행용 동적 링크 라이브러리(.DLL)를 실행 경로에 포함하고 있으며, 각 라이브러리를 시험할 수 있는 샘플 프로젝트의 소스를 포함하고 있습니다.

표 2. 모션 라이브러리 구성

변수	설명	비고
IMotion.h	C 헤더 파일	IMotion
IMotion.hpp	C++ 헤더 파일	IMotion
IMotion_csv.h	모션 파일 로더 헤더 파일	IMotion
IMotion.cs	C# Wrapper	IMotion
IMotion.dll	실행 바이너리 (x86, x64)	IMotion
IMotion.lib	개발 라이브러리 (x86, x64)	IMotion
IMotion_Testbed.exe	Testbed GUI 프로그램 (x86)	IMotion
InnoML.h	C/C++ 헤더 파일	InnoML
InnoML.cs	C# Wrapper	InnoML
InnoML.dll	실행 바이너리 (x86, x64)	InnoML
InnoML.lib	개발 라이브러리 (x86, x64)	InnoML
InnoML_Test.exe	샘플 Console 프로그램 (x86)	InnoML
IMotion_types.h	모션 라이브러리 타입 선언 헤더 파일	공용
IMotion_types.cs	C# Wrapper	공용
NMC2.dll	모션 컨트롤러 실행 바이너리 (x86, x64)	공용
pthreadVC2.dll	pthread 실행 바이너리 (x86, x64)	공용

- IMotion API

모션 시뮬레이터는 기구부의 종류, 운동범위 등에 따라 다양한 종류가 있으며, DOF 수에 따라 달라질 수 있습니다. 이와 같이 다양한 형태의 모션 플랫폼을 제어하는 SW 는 각 모션 컨트롤러와 파라미터가 다르기 때문에 추상화된 모션 제어 라이브러리가 필요합니다.

IMotion은 모션 플랫폼의 각 축에 모션 신호를 직접 전달하고, 현재 상태를 조회할 수 있습니다. 모션 제어는 모션 제어 PC 와 모션 플랫폼 컨트롤러가 이더넷 통신으로 연결되며, 다수의 모션 플랫폼 컨트롤러의 동시 제어도 가능 합니다.

- InnoML API

게임과 같이 실시간 상호작용이 용이하도록 InnoML API 를 지원 합니다. VR 콘텐츠에서 전달하는 복수의 모션 신호를 처리할 수 있으며, 모션 제어 정보에 따라 연속적인 모션 신호를 사용자에게 전달할 수 있습니다.

또한, 모션 데이터 스트림의 직접 전달은 사용자 인터랙션 처리에 유용하게 사용됩니다. 이를 모션 입력이라고 정의하고 모션 신호를 생성하는 방법에 따라 다양한 플러그인이 제공 됩니다.

IMotion 과의 주요 차이점은 모션 소스 막상, 멀티 모션 입출력 장치, Telemetry 지원, 모션 데이터베이스 등을 들 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

2.2 환경 설정

모션 SDK를 사용하기 전에 모션 플랫폼 실행 환경 설정이 필요합니다. 모션 플랫폼 연결, 이더넷 설정 및 연결, 모션 제어의 과정으로 환경 설정을 모두 완료해야 합니다.

SDK를 이용한 응용 프로그램 개발은 크게 다음과 같은 세 과정으로 처리됩니다.

2.2.1 장치 연결

모션 SDK를 사용하기 전에 모션 플랫폼의 환경을 먼저 설정해야 합니다. 모션 플랫폼은 이노시뮬레이션 모션 제품군의 해당 모델명과 버전을 확인하신 후 다음 절차에 따라 설치를 진행 합니다.

- 모션 플랫폼 전원 스위치를 Off (O 놀림) 합니다.
- 모션 제어 PC와 모션 플랫폼의 각각의 랜 포트를 랜 선으로 직접 연결 합니다.
(공유기를 통한 PC와 모션 플랫폼 연결도 가능합니다.)
- 모션 플랫폼의 전원 플러그를 꽂은 다음, 전원 스위치를 On (I 놀림) 합니다.
(주의, 전원 스위치 Off 상태에서 전원 플러그를 꽂는 것이 좋습니다.)

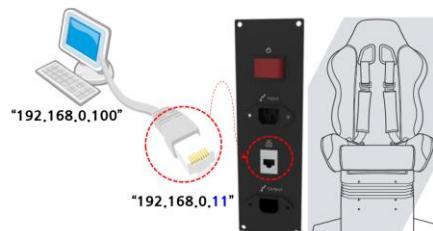


그림 11. 모션 플랫폼 연결

2.2.2 IP 설정

모션 플랫폼을 이더넷 통신으로 로컬 네트워크에 연결될 수 있도록, 모션 제어 PC와 모션 플랫폼의 이더넷 IP 주소를 올바르게 설정해야 합니다.

- PC의 네트워크 설정에서 IP 주소를 고정(수동) “192.168.0.100”으로 설정 합니다.
(모션 플랫폼에 설정된 IP와 중복되지만 않으면, 꼭 100번이 아니라도 무관 합니다.)
- 모션 플랫폼의 IP는 기본값으로 “192.168.0.11”로 설정 되어 있습니다.
(측면 뒷개를 열고 수동으로 IP를 변경 할 수 있습니다.)
- 명령 창에서 “ping 192.168.0.11” 입력하여, 모션 플랫폼에 연결이 되는지 확인합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

(응답이 없다면 PC가 고정 IP 인지 그리고 랜선 및 랜포트가 정상인지 확인해 주세요)

- 모두 정상적으로 설정 했는데도 HW 연결이 불가능한 경우,
측면 덮개를 열고 내부의 차단기를 확인해야 합니다.

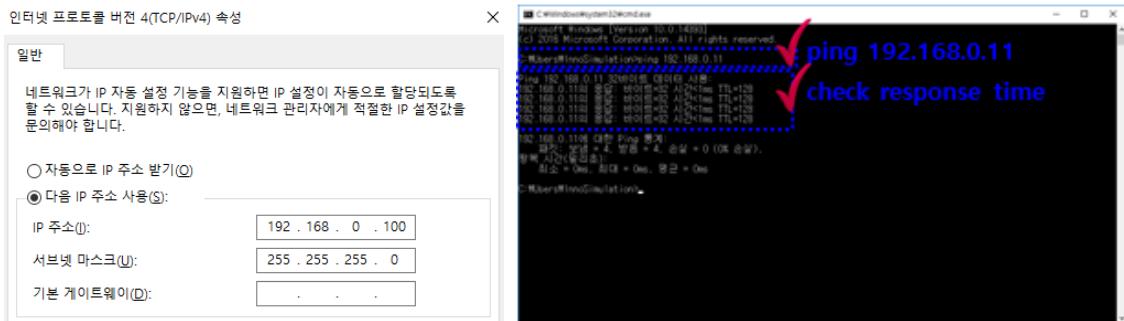


그림 12. 네트워크 IP 설정 예 (모션 제어 PC, 모션 플랫폼)

2.2.3 모션 제어 실행

모션 플랫폼 테스트 프로그램으로 동작을 확인 합니다. 샘플 프로젝트는 Sine 파형을 각 모션의 운동 범위 (DOF)에 적용하여 모션을 특정 속도로 제어 하는 자가 진단 프로그램으로, 진폭과 진동수를 설정할 수 있습니다. 모션 장치의 연결 상태와 각 축의 동작 상태를 확인 할 수 있습니다.

※ 주의, 모션 플랫폼 프로파일에서 제품 이름과 버전 정보의 확인이 필요 합니다. 프로파일은 테스트 프로그램 최초 연결 시 자동 생성됩니다(*run_path/IMotion_#id.in*).

- 제어 PC에서 테스트 프로그램("MotionPlayer/InnoMP/IMotion_Testbed.exe")를 실행 합니다.
- 장치 번호에 IP인 “11” 입력을 확인 후, “Open” 클릭하여 디바이스에 연결합니다.
- “Stop” 클릭 후 모션이 시작 위치로 이동을 확인 합니다.
(Cmd : 2000, Enc : 2000 근사치 정상)
- “Start” 클릭 후 모션이 중립 위치로 이동을 확인 합니다.
(Cmd : 240000, Enc : 240000 근사치 정상)
- “Heave” 또는 “Roll”, “Pitch”의 Amp 와 Freq를 입력한 다음, “Operation” 클릭 하여 모션 제어를 확인 합니다.
(Amp 는 -1 ~ 1 사이 값, Freq 는 재생 주파수 - 예 : Amp (1), Freq (1))
(Freq 는 Amp 에 반비례하여 최대값 제한(Max = 1/Amp) - 예 : Amp (0.1), Freq (10))
- “Yaw”의 모션 제어도 동일하지만 이동 거리가 크기 때문에 속도를 제한해야 합니다.
(사인파(sine wave) Freq 는 최대 1/18 재생 주파수 권장 - 예 : Amp (1), Freq (1/18))

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

(무한턴을 위한 톱니파(sawtooth wave)는 최대 1/6 주파수 권장 - 예 : Amp (1), Freq (1/6))

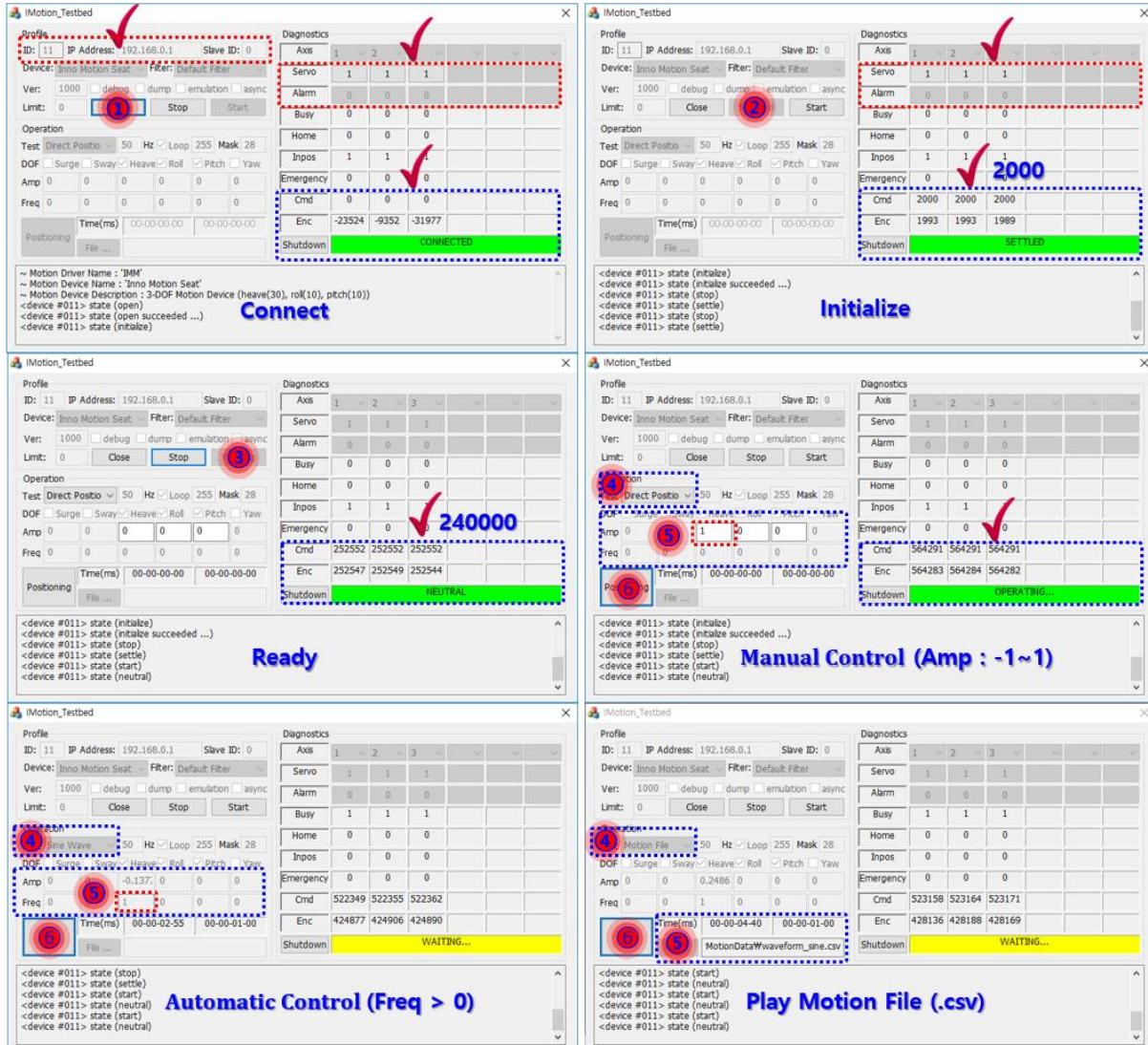


그림 13. IMotion_Testbed

2.3 요구사항 및 설정

VR 모션 시스템 개발 환경 설정 및 하드웨어 요구사항을 설명 합니다.

2.3.1 하드웨어 요구사항

VR 모션 시스템 구성은 모션 제어 PC, 모션 플랫폼, HMD 가 필요합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

다음은 모바일 기반 HMD 와 같이 PC 에서 모션 제어만 수행 하는 시스템 구성에서 모션 제어 PC 최소 요구 사항 입니다.

- OS : Windows 7 SP1 32비트 이상
- CPU : 싱글코어 2GHz 이상
- RAM : 2GB 이상
- HDD : 32GB 이상

Oculus Rift, HTC Vive 과 같은 HMD 용 기존 PC 에서 모션 제어를 함께 구성을 위해 고성능 그래픽 칩이 있는 고사양 PC 가 필요할 수 있습니다.

VR 용 모션 플랫폼의 요구 사항입니다.

- 허용 하중 : 180Kg
- 제품 무게 : 100Kg
- Degrees of Freedom : 3DOF
- 크기(L×W×H) : 800mm × 800mm × 1400mm (의자포함)
- 연결 유형 : Ethernet
- Heave : ± 30mm
- Roll : ± 10degree
- Pitch : ± 10degree
- 전원 : AC 220~240V, 50/60Hz, 3KW

시스템 구성에 따라 유무선 공유기, TV, 컨트롤러 등이 추가 구성 될 수 있습니다.

2.3.2 소프트웨어 요구사항

VR 모션 시스템 개발을 위해 다음과 같이 Visual Studio 또는 유니티 개발 환경이 필요합니다.

프로젝트는 비주얼 스튜디오 2010 이상 닷넷 프레임워크 2.0 이상, 32 비트 및 64 비트에서 개발됩니다.

- Language : C, C++, C#
- Tool : Visual Studio 2010 이상, Unity3d 5.4 이상, Unreal 4.0 이상

2.3.3 비주얼 스튜디오 설정 (C/C++)

비주얼 스튜디오 2010 이상에서 생성된 프로젝트에서 다음 절차를 통해 모션 라이브러리 개발 환경을 설정 할 수 있습니다.

- Visual C++ 2010에서 프로젝트를 생성한 후, “Project>>속성” 을 선택합니다 (Alt+F7).

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

- 속성 창의 구성 속성에서 “C/C++>>일반>>추가 포함 디렉토리”에서 (SDK)/include 경로를 추가 합니다.
- 속성 창의 구성 속성에서 “링커>>일반>>추가 라이브러리 디렉토리”에서 (SDK)/lib 경로를 추가 합니다.
- “링커>>입력>>추가 종속성”에 IMotion.lib 또는 InnoML.lib 를 입력하면 설정이 완료 됩니다.

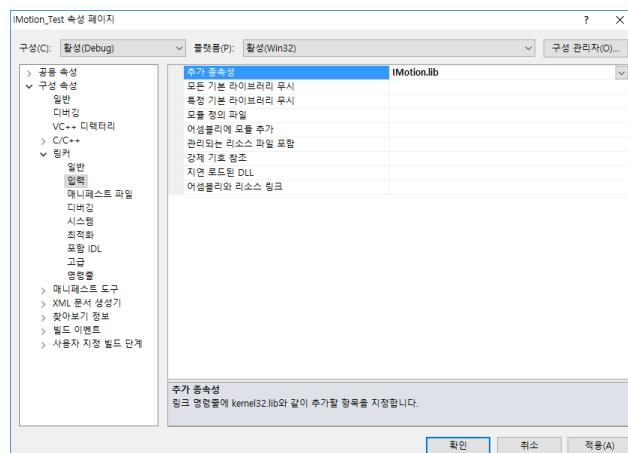


그림 14. 프로젝트 설정

빌드 후 실행을 위해서 실행 바이너리(.dll 파일)가 필요 합니다. (SDK)/lib 경로를 실행 경로에 추가하거나, 바이너리 파일을 복사해서 사용하실 수 있습니다.

2.3.4

비주얼 스튜디오 설정 (C#)

C# 프로젝트를 생성하고 ‘참조추가’로 배포된 모션 라이브러리 C# Wrapper 파일(.cs)을 지정하면 API 를 using 키워드를 통해 불러들여 사용 할 수 있습니다. 라이브러리 include 폴더에 Native DLL 파일을 C#으로 래핑한 파일들(IMotion_Types.cs, IMotion.cs, InnoML.cs)이 포함되어 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

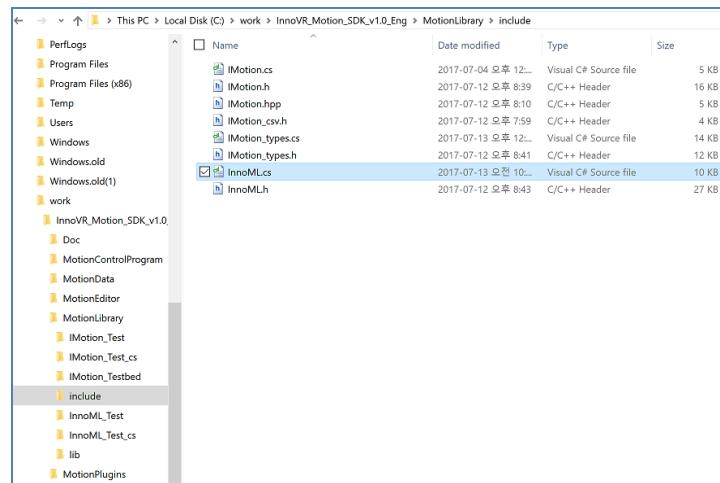


그림 15. C# 개발 환경

3 모션 라이브러리

3 장에서는 모션 라이브러리의 기본 개념을 설명하고 사용 예를 통해 VR 모션 시스템 구성 방법을 알아봅니다.

VR 모션 시스템은 현실세계의 모션 입력 신호를 인간이 느낄 수 있는 운동 신호로 변환하는 디지털 신호 처리 과정으로 볼 수 있는데, 모션 디지털 신호의 입력, 합성, 출력의 3 과정으로 구분 할 수 있습니다. 모션 컨텍스트 관리 객체를 통해 모션 신호 처리 과정을 수행 하고, 모션 입력 스트림 처리와 모션 효과 관리를 위해 모션 입력 객체와 모션 소스 객체를 추가로 제공 합니다.

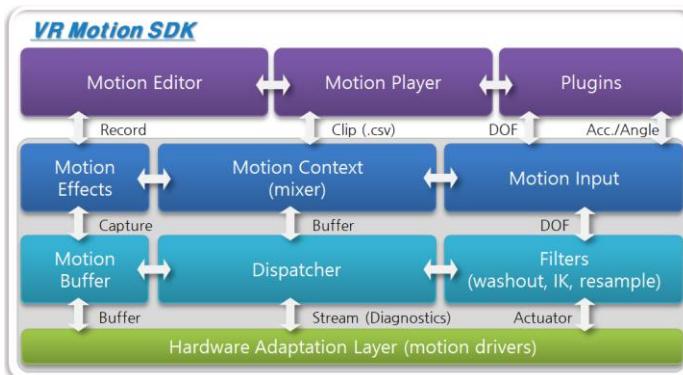


그림 16. 모션 SDK 아키텍처

모션 입력으로 전달되는 운동과 힘에 따라 모션 시스템이 제어 되는데, 게임 컨트롤러와 같은 센서를 갖는 장치를 통해 가상 세계에 모션 신호를 전달합니다. 가상 세계의 시뮬레이션 결과는 다시 시각 시스템과 모션 시스템을 통해 인간에게 시각 및 운동감이 전달됩니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

3.1 모션 라이브러리 소개

가상현실을 구성하는 다른 서브시스템들과 같이 모션 시스템 또한 입력된 디지털 신호에 특별한 처리 과정을 거친 다음 다시 출력 장치를 통해 인간에게 감각을 전달하도록 해야 합니다. 모션 신호는 가상 객체에 힘을 전달하여 운동을 제어할 수 있는데, 힘의 모션 신호를 시뮬레이션 하여 모션을 제어하는 방법을 많이 사용합니다.

모션 라이브러리는 이 같은 실시간 시뮬레이션 환경을 구성하는 개발자가 직접 모션 플랫폼을 제어할 수 있는 인터페이스(API)를 제공 합니다.

3.1.1 개요

일반적으로 디지털 신호 처리는 ADC에서 입력 받은 디지털 신호(PCM)을 처리하여 다시 DAC로 전달하는 일련의 SW 처리 과정을 보이고 있습니다.

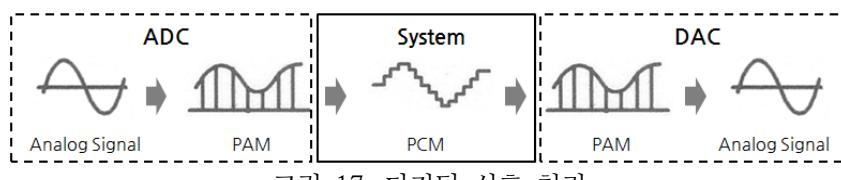


그림 17. 디지털 신호 처리

VR 모션 시스템의 모션 제어 처리 과정도 디지털 신호 처리와 동일한 과정을 수행 합니다. ADC는 다양한 모션 입력 장치에서, DAC는 모션 출력 장치의 주요 기능입니다. 모션 장치는 종류에 따라 ADC, DAC 를 모두 지원할 수 도 있고 하나의 기능만 제공할 수도 있는데, 모션 플랫폼 장치는 주로 모션 출력 기능만 담당하고, 입력은 다른 모션 입력 컨트롤러와 함께 사용됩니다.

게임 컨트롤러와 같이 다양한 입력 컨트롤러에서 전달된 입력 신호는 디지털 신호로 변조되어 시뮬레이션 되고, 그 처리 결과는 출력 컨트롤러에 의해 다시 아날로그 신호로 변조되어 인간에게 운동감(모션 Cue)을 전달합니다.

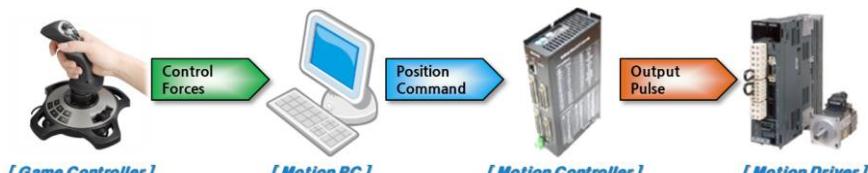


그림 18. 모션 신호 처리 과정

3.1.2 모션 신호

VR 모션 시스템은 AD 변환기로부터 전달 받은 디지털 신호를 처리합니다. 대부분의 센서 데이터들은 센싱된 모션에 대한 연속적인 전압의 파형이며, 해당 모션의 특성은 파형의 진폭(amplitude)의 패턴으로 표현 됩니다. 연속된 아날로그 모션 파형으로부터 디지털 모션 파형을 만들기 위해 디지털 형식으로 바꿔야 하는데, 그 과정을 각각 샘플링(sampling)과 양자화(quantization)라고 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

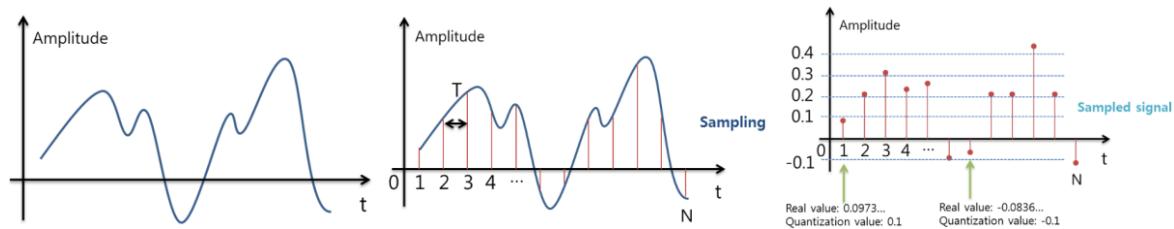


그림 19. 샘플링과 양자화 예시

그림의 좌측 이미지는 가로축의 시간과 세로축의 진폭으로 표시된 연속된 모션의 아날로그 신호입니다. 여기서 샘플링은 가운데 그림과 같이 일정한 시간 간격의 주기로 신호의 강도를 수집하는 것입니다. 빨간 줄이 샘플링 지점이 되고 그 간격이 작을수록 원래 신호에 가까워집니다.

양자화는 샘플링된 아날로그 신호를 디지털화하는 작업을 말합니다. 양자화 수준은 데이터 포맷의 범위에 따라 결정되는데, 그림의 우측에서 0.1의 양자화 오차의 사용 예를 보이고 있습니다. VR 모션 시스템에서는 16비트 정수형(-32768 ~ 32767)을 사용하여 양자화 오차를 최소화 합니다.

PCM은 샘플링 되고 양자화된 디지털 신호를 메모리에 이진수로 인코딩된 데이터입니다. 인코딩(부호화) 과정은 채널 막싱과 압축 과정이 있습니다. 모션 데이터는 각 모션의 DOF만큼 독립된 모션 신호의 채널이 존재 합니다. 3 자유도 모션 플랫폼의 경우 항상 3쌍의 모션 파형 신호를 받아 처리해야 하기 때문에, VR 모션 시스템은 샘플링 시점에 생성된 각 샘플마다 모든 채널 데이터를 함께 PCM 데이터로 저장합니다.

3.1.3 모션 기본 개념

모션 시스템은 모션 플랫폼의 각 자유도(DOF)마다 독립된 모션 신호를 처리하는데, 모션 위치는 시간 축 상의 각 채널 모션 신호(운동 값)가 항상 동시에 제어 되어야 합니다. 이를 모션 샘플이라고 부르고 각 자유도 별 모션 신호를 모션 채널이라 부릅니다.

모션 시스템은 시간에 따라 어떤 모션 샘플을 전달하느냐에 따라 모션 플랫폼의 연속된 움직임을 만들어 낼 수 있습니다.

모션 위치는 모션 플랫폼의 좌표계 상의 각 축의 운동 값을 말하는데, 모션 플랫폼의 실제 좌표계 상의 물리적 위치와 가상세계 좌표계 상의 논리적 위치로 구분할 수 있습니다. VR 모션 시스템에서는 다양한 모션 플랫폼 장치 호환성을 위해 논리적 위치 값을 사용합니다.

표 3. 모션 데이터 용어 정리

명칭	설명
물리적 위치 (Physical Position)	물리적인 모션 테이블 상판의 좌표계에서 각 축의 실제 위치나 회전각 - heave(-30~30 mm), roll & pitch (-10~10 degree)
논리적 위치 (Logical Position)	모션 테이블 상판의 최대 운동 범위의 백분율(16 비트 양자화)로 계산된 논리적 좌표계에서 각 축의 상대적인 위치나 회전각 - heave & roll & pitch (-32768 ~ 32767)
Channel	모션 샘플을 구성하는 특정 축의 모션 신호. (heave 채널, ...) 모션 플랫폼의 자유도(DOF) 수에 따라 채널 수 결정
Sample	특정 시간에 동시 제어할 축 별 모션 위치의 묶음 (heave, roll, pitch) 모션 스트림으로 모션 장치를 제어하는 최소 단위.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

Buffer	시간축 상의 임의 구간 동안의 모션 샘플들의 집합 (<u>PCM 데이터</u>) 모션 효과(Bump, Vibration, ...)를 하나의 버퍼에 저장.
--------	--

3.1.4 모션 데이터 (PCM)

모션 시스템에서 모션 데이터는 모션 플랫폼 또는 시뮬레이션 모델의 논리적 위치(Logical Position)를 16 비트로 양자화한 값을 사용합니다. 아래 좌측 그림에서 모션 신호를 50Hz로 샘플링한 모션 데이터 파일("waveform_sine.csv")의 내용을 보이고 있고, 우측 그림과 같이 각 모션 채널 신호가 양자화된 값이 모션 버퍼에 PCM 데이터로 저장됩니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

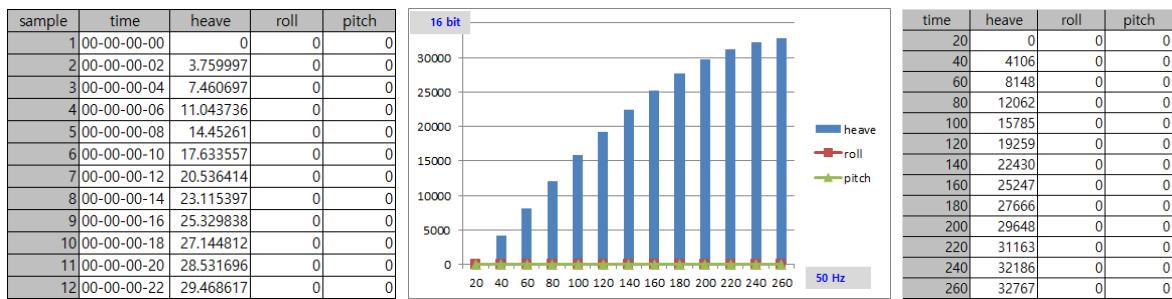


그림 20. 모션 데이터 (모션 파일과 PCM 데이터)

모션 버퍼는 PCM 데이터로 변환하면서 양자화 오차가 발생하는데, 모션의 정밀도는 샘플링 주기(가로축 시간 간격)와 양자화 수준(세로축 진폭 간격)에 의해 결정됩니다. VR 모션 시스템에서의 양자화 수준은 16 비트로 높은 수준이므로, 주로 샘플링 주기에 의한 모션 시스템의 시간 분해능에 의해 모션 제어 성능이 결정된다고 볼 수 있습니다. 모션 시스템의 샘플링 주기는 시각 시스템(25~60hz)과 음향 시스템(44khz) 사이의 주파수 대역(50 ~ 1000hz)을 사용해야 하는데, 이는 모션 플랫폼이 처리할 수 있는 성능 범위 안에서 설정되어야 합니다.

```
short PCM_BUF[50][3] = {0,};
for(i=0; i<50; i++) {
    PCM_BUF[i][0] = (short)(sin(2 * IM_PI * (float)i/50) * MOTION_MAX_16);
}
```

그림에서 보여진 PCM 버퍼의 생성을 API로 구현한 예를 보이고 있습니다. 1 초 주기로 반복되는 연속된 sin 파형의 아날로그 신호로부터, 50Hz로 모션 샘플을 추출합니다(PAM). 다시 그 값을 16 비트로 양자화 한 결과를 heave 채널에 인코딩한 PCM 데이터 생성 코드 예를 보이고 있습니다.

3.2 모션 버퍼(imBuffer)

시각 시스템이나 음향 시스템처럼 모션 시스템 또한 PCM 데이터로 모션 신호를 처리합니다. 모션 라이브러리에서는 이를 모션 버퍼라고 정의하여 모션 데이터를 관리합니다.

또한, 1 장에서 언급한 것과 같이 모션 시스템의 성능은 모션 신호 처리 결과가 아날로그 신호로 다시 인간에게 전달되는, 시각 신호와 모션 플랫폼을 통한 운동감의 동기화에 따라 결정된다고 볼 수 있습니다. 모션 지연 시간(latency)과 분해능(resolution)을 모션 시스템의 성능 기준으로 볼 수 있는데, 모션 버퍼의 포맷 설정에 따라 모션 신호의 처리 방법이 결정됩니다.

3.2 장에서는 모션 데이터 메모리 및 포맷 관리의 모션 버퍼 주요 기능을 알아봅니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

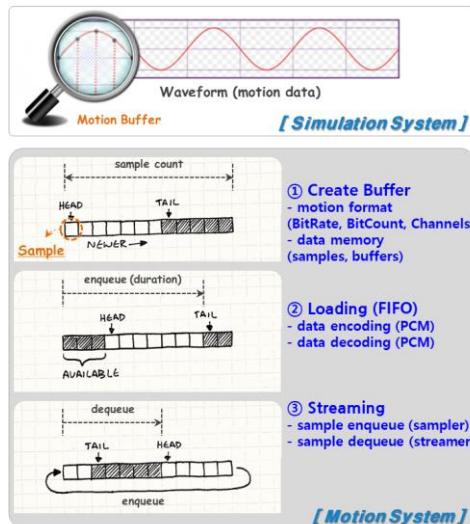


그림 21. 모션 버퍼 주요 기능

3.2.1 버퍼 생성

모션 버퍼는 모션 신호를 인코딩하기 위한 입력용으로, 그리고 인코딩된 데이터를 다시 디코딩하기 위한 출력용의 2 가지 목적으로 사용됩니다.

모션 버퍼는 샘플 단위로 인코딩(encode)되거나 디코딩(decode) 될 수 있는데, 모션 시스템은 샘플들의 집합인 버퍼(buffer) 단위로 콘텐츠 동기화 및 연산, 모션 장치 출력을 수행합니다.

모션 장치 출력 버퍼(마스터 버퍼)의 경우 버퍼 단위로 모션 신호를 모션 플랫폼에 전달하기 때문에 모션 버퍼의 시간 간격만큼 지연이 발생할 수 있습니다. 그림 좌측의 수직 막대는 샘플링 된 모션 샘플들을 보이고 있는데, 모션 장치로 한 번에 전송되고 내부적으로 주기에 맞게 하나씩 처리 됩니다. 버퍼의 처리 시간은 샘플 갯수와 샘플 시간(1/sampleRate)의 곱에 의해 계산되는데, 이 시간이 길수록 모션 신호의 지연이 발생할 수 있습니다.

$$\text{Motion Latency (Buffer Time)} : \text{samples} * 1000 / \text{sampleRate}$$

예를 들어, 그림에서 샘플 수(빨간 점선의 수)가 4이고 샘플 시간이 10ms(100hz)인 경우의 지연 시간은 최대 40ms입니다. 지연에도 불구하고 모션 버퍼를 사용하는 이유는 일련의 모션 위치를 매번 동기화하여 직접 제어 하는 것 보다, 연속적으로 동일한 시간 간격 동안에 모션 위치를 더 자주 전달하여 모션 제어 정밀도(resolution)를 높일 수 있기 때문입니다.

모션 제어 PC에서 모션 플랫폼(컨트롤러)의 위치 제어 성능을 최대로 활용하지 못하는 이유는 두 컴퓨터간의 통신 동기화 지연에 그 원인이 있습니다. 모션 버퍼를 통해 가장 큰 오버헤드인 통신 횟수를 줄이고 저장된 신호를 대기 없이 실시간 처리하여 모션 제어 처리 성능을 높일 수 있습니다. 모션 샘플 직접 제어(모션 샘플 스트리밍) 보다 모션 버퍼를 통한 자동 제어(모션 버퍼 스트리밍) 사용을 권장합니다. (InnoML은 모션 버퍼 스트리밍 방식을 사용합니다.)

```
IMBuffer master = imCreateBuffer(100, 0, 0, 4, 2);
ctx = imCreateContext(master, 0, &desc);
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

모션 버퍼는 모션 효과 데이터, 사용자의 입력 데이터, 모션 장치 출력용으로 모션 소스, 모션 입력, 모션 컨텍스트에서 각각 사용됩니다. 샘플 주기(100hz), 데이터 포맷(기본 16 비트), 채널 수(기본 3), 샘플 수(4 개), 버퍼 수(2 개)를 사용하여 컨텍스트의 마스터 버퍼를 생성하는 API 사용 예를 보이고 있습니다. 마스터 버퍼나 입력 버퍼의 경우 데이터 저장과 읽기 과정이 각각의 스레드에서 동시에 처리 될 수 있으므로 더블버퍼링 사용을 권장하며, 샘플들을 모션 장치에 한번에 제출 하므로 통신비용을 절감할 수 있습니다.

모션 버퍼는 생성 함수 3개의 인자로 표 4의 포맷 정보를 저장하고, 마지막 2개의 인자로 모션 버퍼의 실제 메모리를 생성하여 데이터베이스에 등록 관리합니다.

표 4. Format 구조체 (IM_FORMAT)

변수	설명	기본값
nType	모션 데이터의 타입을 정의합니다. DOF 타입은 축 위치, AXIS는 모터 회전량, MATRIX는 DOF의 변환행렬입니다. 현재 DOF 타입만 사용 가능합니다. ("imotion_types.h"의 IM_FORMAT_TYPE_* 참고)	DOF
nSampleRate	초당 샘플의 수. 1초 동안 처리할 샘플의 수를 정의 합니다. 역으로 샘플당 처리 시간을 계산할 수 있습니다. 이 값은 모션 제어 성능(시간 분해능)을 결정하기 때문에 장치의 성능을 고려하여 설정해야 합니다.	50
nChannels	모션 플랫폼의 자유도 수. 3 자유도 모션 플랫폼에서 기본 값으로 3 채널이 사용 됩니다. 지원 범위 이상(8)의 채널은 라이브러리에서 무시되는 사용자 채널 정보입니다.	3
nDataFormat	PCM 데이터의 양자화 수준. 라이브러리는 논리적 위치 값을 사용하여, 항상 16비트로 양자화 됩니다.	S16
nBlockAlign	모션 샘플의 바이트 크기. 각 채널 정보를 포함하는 인코딩된 샘플의 크기만큼 이동하여 데이터를 처리 합니다. (0으로 지정하면 내부에서 포맷 정보로 크기를 자동 계산 합니다)	6

3.2.2 버퍼 채우기(enqueue)

모션 버퍼 생성으로 확보된 메모리에 필요한 만큼 모션 데이터를 저장해야, 인코딩 크기(duration)만큼 재생이 가능합니다.

모션 소스의 경우 한번 로딩된 모션 데이터는 추가적인 로딩 과정이 없는 단순 재생 목적이 많기 때문에, 보통 버퍼의 수를 1로 하고 샘플 수만큼 생성하여 사용합니다. 모션 파일 로딩 함수(imLoadBuffer)로 데이터를 로딩 할 수 있고, 생성된 버퍼에 데이터를 직접 저장(imBufferEnqueue) 할 수 있습니다. 아래 예제는 샘플 50개를 저장할 수 있는 단일 버퍼에 직접 데이터를 저장하는 예를 보이고 있습니다.

```
IMBuffer effect = imCreateBuffer(50, 0, 0, 50, 1);
imBufferEnqueue(effect, raw_data, raw_size); // write data
int duration = imBufferGetDuration(effect);
IMSource source = imCreateSource(effect);
```

모션 소스의 버퍼 참조 방법은 아래 그림의 좌측과 같은 선형 큐 방식이기 때문에 미리 저장된 메모리 영역(메모리 주소 공간상의 처음부터 저장 위치까지)을 순차적으로 참조 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

반면 모션 입력의 경우 오른쪽 그림과 같이 생성된 모션 버퍼에 모션 샘플을 실시간으로 저장하는 환형 큐 구조로 순환 처리되기 때문에, 버퍼가 차지 않도록 버퍼를 읽어야(dequeue) 연속적으로 다음 데이터를 실행 중에 저장(enqueue) 할 수 있습니다. 즉, 최소 샘플 크기부터 최대 가능 영역(TAIL~HEAD) 크기까지 저장 가능 합니다.

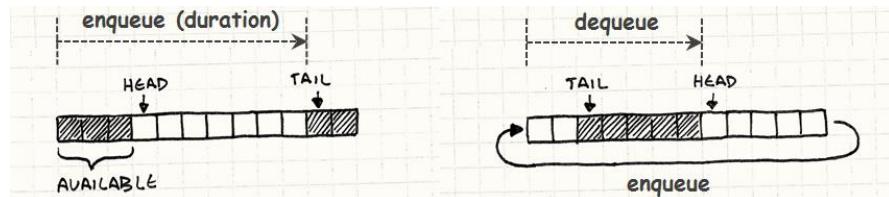


그림 22. 모션 버퍼 큐잉 (좌: 인큐, 우: 디큐)

3.2.3 버퍼 읽기(dequeue)

모션 버퍼에 저장된 모션 샘플 데이터를 읽어오는 작업을 디큐(dequeue)라고 하는데 그림에서 HEAD는 읽기 시작 위치를 표시하고 있습니다. 좌측 그림은 이미 3개의 버퍼 블록을 사용한(디큐) 상태를 보이고 있는데, 다음 입력에서 다시 저장할 수 있도록 인큐 가능 상태로 체크됩니다. 몇 번의 저장이 추가되면 TAIL은 우측 그림과 같이 다시 처음으로 이동하게 되고, 이 과정이 계속 진행되면 HEAD는 TAIL을 따라 다니면서 2개의 위치가 계속 오른쪽으로 순회합니다.

HEAD와 TAIL이 만나면 큐잉 지연이나 입력 데이터의 손실(loss)이 발생 할 수 있어 주의가 필요합니다. 어플리케이션에서 전송되는 모션 데이터를 실시간으로 저장하고 모션 장치에 그 신호를 전달하는 연속된 처리를 모션 스트리밍이라고 합니다. 모션 스트리밍에서 모션 입력 주기와 모션 장치의 출력 주기를 일치시키고, 큐잉 대기 시간이나 입력 신호의 손실이 없도록 충분한 최적의 크기로 버퍼를 생성하는 것이 좋습니다.

```
short stream[4*3];
int len = imBufferDequeue(input_buffer, stream, sizeof(stream)); // read
imBufferEnqueue(effect_buffer, stream, len); // append data
```

입력 버퍼에 저장된 모션 데이터를 읽어 모션 효과 버퍼를 채우는 예를 보이고 있습니다. 이 예를 활용하면 반대로 모션 효과 파일을 읽어 모션 입력 버퍼에 채우거나, 다른 모션 컨텍스트에 모션 데이터를 공유하는 등으로도 활용이 가능 합니다.

3.2.4 주요 API

모션 데이터 저장 및 포맷 관리의 책임이 있는 Buffer는 다음과 같은 주요 API 기능을 갖습니다. (상세한 설명은 API Reference를 참고하세요.)

표 5. Buffer 주요 API

API	설명
imCreateBuffer(sample_rate, format, channels, samples, buffers)	모션 버퍼 객체를 생성 합니다. 샘플 수, 버퍼 수 지정으로 환형 큐 구조의 모션 데이터 메모리를 생성합니다. 버퍼 재생을 위한

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

	샘플링 주기, 포맷, 채널수의 포맷 정보를 정의합니다.
imLoadBuffer(url, key) imLoadBufferMemory(data, size, key)	모션 파일을 로딩 하여 모션 버퍼 객체를 생성 합니다. 로딩할 파일 이름 문자열이나 파일 메모리를 정의 하고 암호키를 지정할 수 있습니다.
imBufferConvert(buffer, adjusted, desired, filter)	마스터 버퍼의 포맷에 맞게 변환된 버퍼를 생성 합니다. (필터를 적용하여 다양한 모션 효과를 만들 수 있습니다)
imBufferEnqueue/Dequeue	버퍼에 샘플 데이터를 저장하거나 읽습니다.
imBufferGetSize/Duration/Format	버퍼의 크기, 재생 시간, 포맷 정보를 얻습니다.
imBufferGetQueuedCount	현재 버퍼에 큐잉된 샘플 수를 얻습니다.
imDeleteBuffer	모션 버퍼 객체를 제거 합니다.

3.3 모션 컨텍스트(imContext)

VR 모션 시스템은 콘텐츠에서 모션 플랫폼을 제어하여 인간에게 운동감을 전달하는 것으로, VR-용 운동 시뮬레이터 장치를 관리하는 컨텍스트 객체가 있습니다. 다수의 시뮬레이션 객체로 구성된 대규모 가상현실 시스템 구성에서는 각기 다른 운동 시뮬레이터가 요구 될 수 있기 때문에, 각 운동 객체와 상호작용이 가능한 환경이 필요합니다.

가상 세계의 대상 운동 객체마다 모션 장치를 연동 하여 실시간 모션 시스템이 구성 할 수 있습니다. 컨텍스트는 가상 세계의 운동이 현실 세계에서 운동감으로 느끼도록 상호작용하는 전반적인 모션 제어 흐름을 관리 합니다.

3.3 장에서는 모션 장치 및 마스터 버퍼 관리의 모션 컨텍스트 주요 기능을 알아봅니다.

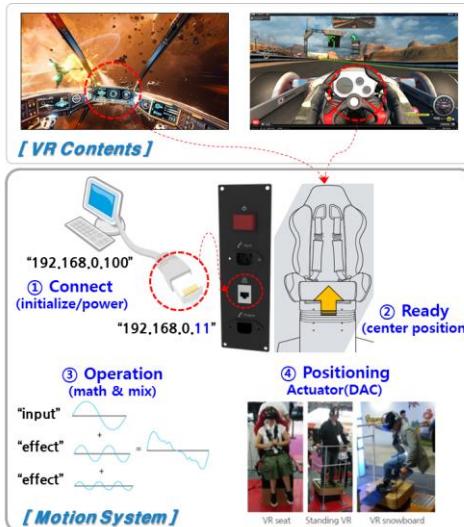


그림 23. 모션 컨텍스트 주요 기능

3.3.1 모션 장치 연결

VR 모션 시스템은 모션 플랫폼 시뮬레이터와 모션 플랫폼을 제어하는 시뮬레이션 컴퓨터(모션 제어 PC)로 구성 됩니다. 모션 제어 PC는 콘텐츠의 시뮬레이션 환경을 제공하고, 모션 플랫폼은 전달된 모션 제어 명령을 수행 합니다. 두 컴퓨터가 네트워크로 서로 연결되어 메시지를 처리하는데, 이더넷

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

통신으로 연동 제어 됩니다. 모션 제어 PC와 모션 플랫폼은 외부 게이트웨이 연결 없이 로컬 네트워크로 연결이 가능하도록 동일한 이더넷 IP("192.168.0.xxx")로 설정해야 하고, 마지막 값은 이더넷 허브 등의 LAN으로 연결된 모든 장치를 식별할 수 있는 서로 다른 값으로 설정 되어야 합니다.

모션 플랫폼 컨트롤러의 이더넷 IP는 "192.168.0.11"로 초기 설정되어 있고, 모션 제어 PC는 11번을 제외한 다른 IP를 사용할 수 있습니다. "2.2 환경 설정"을 참고로 제어 PC의 환경 설정을 진행하는 것이 좋습니다.

모션 플랫폼 이더넷 IP의 마지막 자리(1~255)를 모션 장치 ID로 사용하여 컨텍스트를 생성해야 합니다. 기본값으로 11번이 사용되고, 멀티 컨텍스트에서 최대 32개까지 지원 가능합니다 (중복 ID 사용 불가).

```
IMContext ctx;
IM_DEVICE_DESC desc;
memset(&desc, 0, sizeof(IM_DEVICE_DESC));
desc.szName = "Inno Motion Seat"; // device name
desc.nVersion = 1000; // device version number
desc.nOptions |= IM_CFG_DEBUG_MODE; // device options
ctx = imCreateContext(0, 11, &desc); // open device "192.168.0.11"
...
imDestroyContext(ctx, 0); // close device
```

모션 플랫폼 장치에 맞는 정보를 설정하여 컨텍스트를 생성할 수 있습니다. 모션 장치 정보는 사용자 설정 변수(표 6. Description 구조체의 RW 변수) 만 변경이 가능하고, 다른 변수는 해당 장치의 정보를 조회만 가능합니다. 모션 장치에 연결되면 해당 장치 번호로 설정 파일 ("IMotion_#id.ini")이 실행 경로에 생성 또는 갱신되는데, Description 구조체를 NULL로 지정하면 기존 프로파일로부터 정보를 로딩 합니다(코드 수정과 재빌드 없이 프로파일 수정으로 장치 옵션 변경 가능).

아래 표의 모션 장치 기술 정보에서 장치 버전, 옵션, 이름은 연결 시 설정이 가능한 필드입니다. 별도의 설정 없이 연결되는 모션 장치는 모두 기본 값으로 연결됩니다. 생성된 장치 컨텍스트의 실행 중에 정보 변경은 불가능 하므로 컨텍스트를 다시 생성해야 합니다.

표 6. Description 구조체 (IM_DEVICE_DESC)

변수	설명	비고
nId	모션 장치 식별 번호. 지원 가능한 장치 목록의 순서입니다.	R
nType	모션 장치의 캡처 지원 여부. 현재 출력 전용 장치만 지원 합니다.	R
nMask	지원 가능한 자유도(채널) 수와 비트 마스크. 하위 8 bit는 모션 플랫폼의 각 축으로 예약되고, 상위는 사용자 정의 채널입니다. 3 DOF(heave, roll, pitch)일 때 0xe를 얻습니다. ("imotion_types.h"의 비트 마스크 참고)	R
nVersion	모션 장치 버전. HW 버전에 따라 제어 SW 파라미터가 조정됩니다. $version = (major * 1000) + (minor * 100) + patch$	RW
nOptions	모션 장치 제어 옵션(LSB)입니다. 장치 연결 시 옵션 설정이 필요할 수 있습니다. ("imotion_types.h"의 IM_CFG_* 참고)	RW
	모션 장치 초기 설정(MSB)입니다. 장치 연결 시 옵션 설정이 필요할 수 있습니다. ("imotion_types.h"의 IM_CFG_DEVICE_* 참고)	RW

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

	※ 주의, MSB는 프로파일에서만 변경할 수 있습니다. (run_path/IMotion_#id.ini)	
szName	모션 장치 이름(ascii). 식별 번호마다 다른 장치 이름이 부여되고, 장치 연결 시 이름이 필요합니다.	RW
szDetail	해당 모션 장치의 상세 설명(ascii). 지원 운동 범위나 자유도 정보 및 특징 등을 설명합니다.	R

모션 장치 이름은 시트 타입("Inno Motion Seat")과 스탠드 타입("Inno Motion Stand")의 모션 장치를 선택할 수 있고, 기본 값은 모션 시트입니다. 장치 버전은 정식 양산용 1.0(1000)과 개발용 0.7(700)을 지원 하며, 기본값은 1000입니다. 또한 기타 모션 장치 실행 옵션(IM_CFG_*)을 설정할 수 있습니다.

표 7. 컨텍스트 옵션 (IM_CFG_*)

변수	설명	비고
IM_CFG_DEBUG_MODE	디버깅 모드. 모션 컨텍스트의 제어 위치를 콘솔에 출력합니다.	common
IM_CFG_FILE_LOG	파일 텁프. 모션 컨텍스트의 제어 위치를 로그 파일에 기록합니다. 컨텍스트 소멸시 로그 파일이 생성 됩니다. (run_path/IMotion_#id.log)	common
IM_CFG_ASYNC_MODE	상태 제어 명령을 비동기적으로 수행 합니다. 현재 상태 제어를 수행 중에 사용자 로직 수행이 가능 합니다. (3.3.2 모션 장치 상태를 참고하세요)	common

모션 라이브러리가 지원하는 장치 종류 수와 정보를 조회할 수 있는데, 조회한 상세 정보를 참고하면 장치 컨텍스트 생성과 사용에 힌트를 얻을 수 있습니다. 또한 활성 컨텍스트가 있고 요청 장치 수가 1인 경우에는 현재 활성 장치의 정보를 조회 할 수 있습니다.

```
IM_DEVICE_DESC desc[16];
int nDev = imGetDescriptionCount();
imGetDescription(desc, nDev); // all supported devices
Desc[1].nOptions |= IM_CFG_DEBUG_MODE;
...
ctx = imCreateContext(0, 11, &desc[1]);
imSetContext(ctx);

IM_DEVICE_DESC profile;
imGetProfile(&profile); // active device
```

멀티 컨텍스트를 사용할 때 여러 컨텍스트를 동시에 제어할 수 없고, 활성 컨텍스트로 전환 되어야 제어 될 수 있습니다. 어플리케이션의 모션 서브시스템을 구성에서 각 모션 컨텍스트별로 활성 컨텍스트 전환 후 일괄 처리하는 방법을 권장합니다. (참고, 단일 컨텍스트에서는 imSetContext 없이도 사용 가능 합니다.)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

3.3.2 모션 장치 상태

모션 플랫폼은 장치 연결부터 연결이 해제될 때 까지의 모션 플랫폼 상판 위치에 따라 모션 장치 상태를 정의 합니다. 사용자에게 콘텐츠에 동기화된 모션의 운동감을 전달하기까지 준비 절차가 필요한데 다음과 같은 모션 장치의 상태를 정의 할 수 있습니다.

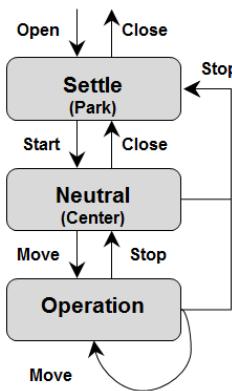


그림 24. 모션 상태 머신

- 원점 상태(Zero)

원점 복귀는 모션 시스템의 장치를 리셋할 때 자동으로 이루어지고 정확한 원점을 찾기 위해 센서부 근처에서 여러 번의 원점 위치 이동을 통해 이루어집니다. 모션 제어에 있어서 위치 이동은 위치 감시 및 모션제어를 위하여 정확한 원점 위치 설정을 필요로 합니다. 인크리멘탈 엔코더는 절대위치를 측정하지 못하고 상대적인 위치만을 측정할 수 있기 때문에 이동 거리에 따라 원점 검출 시간이 소요 될 수 있습니다. 원점 복귀를 수행하지 않았거나 잘못 설정된 원점으로 리밋 센서에 의한 알람이 발생 할 수 있으므로 주의가 필요합니다. 원점 복귀는 서보 모터(bServoOn)가 처음 켜질 때만 수행됩니다.

※ 주의, HW 버전 1.0 부터 인크리멘탈 엔코더를 사용하므로, 장치 연결시 원점 복귀를 위해 다소 시간이 소요 될 수 있습니다.

- 초기 상태(Settle)

모션 라이브러리는 모션 플랫폼 연결 직후 또는 해제 직전에는 모션 장치의 초기 위치로 자동 이동 합니다. 초기 위치는 모션 상판이 수평이고 수직 위치(heave)가 최저인 상태를 말합니다. 주차 (Park) 상태로 불리기도 하는데, 안전을 위해 체험자는 항상 이 상태일 때 모션 플랫폼에 앉거나 내려야 합니다. 초기 상태는 원점 위치 보다 약간 위쪽에 위치합니다.

- 중립 상태(Neutral)

모션 초기 상태에서 사용자의 탑승 후 체험 준비가 완료되면, 모션 플랫폼의 동작을 시작하기 전에 모션 장치를 중립 위치로 이동 시켜야 합니다. 중립 위치는 모션 상판이 수평이고 수직 위치가 중앙(Center)인 상태를 말합니다(각 축 위치가 모두 0 인 상태).

```

ctx = imCreateContext(); // Zero & Settle
// Ready
  
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

```
imStart(); // Neutral
// Operation (Move ...)
imStop(IM_DEVICE_MOVE_NEUTRAL); // Neutral
imDestroyContext(ctx,0); // Settle
```

모션 파형은 대부분 중립 위치에서 시작하고 준비 상태인 중립 위치로 회귀하는 특성이 있기 때문에, 모션 제어를 시작하기 전에 항상 중립 위치로 이동해야 합니다. (참고, 모션 파일 또한 스트림의 시작과 끝 부분을 중립 위치로 생성하는 것이 좋습니다.)

- 운행 상태(Operation)

컨텍스트에 전달된 모션 신호들은 연속된 모션 샘플 스트리밍으로 모션 플랫폼을 제어합니다. 운행 모드에서는 초기 상태와 중립 상태로의 위치 이동 속도보다 더 빠르게 이동하는데, 최초의 모션 샘플이 전달되어 모션 장치가 실행 상태로 변경 될 때 서보 모터의 제어 속도가 변경 됩니다.

- 비동기 상태 모드

모션 플랫폼의 상태 변경은 HW 상태 변경이 완료 될 때 까지 어플리케이션의 대기 시간이 발생 할 수 있기 때문에 실시간 시뮬레이션 시스템에 적합하지 않을 수 있습니다. 이를 위해 모션 컨텍스트의 생성 옵션으로 비동기(IM_CFG_ASYNC_MODE) 모드를 지정하면 현재 요청된 상태를 바로 완료하지 않고 다음 상태 변경 명령 시작 시점까지 완료합니다.

예를 들어, 장치 연결시 수행되는 장치 초기 위치로의 이동 명령(Create)은 어플리케이션에서 다음에 지시한 중립 위치 이동 명령(Neutral)을 수행하기 직전에, 이전 초기 위치 이동 명령이 완료되기를 기다린 후 새로운 중립 위치로의 이동 명령을 시작 합니다.

※ 주의, 비동기 모드에서 모션 장치의 상태 동기는 사용자가 직접 Diagnostic bBusy 체크를 수행해야 합니다.

```
desc.nOptions |= IM_CFG_ASYNC_MODE;
imCreateContext(0,0,&desc); // Open -> Settle Start
// process other system or biz logic ...
imSetContext(ctx, IM_DEVICE_MOVE_NEUTRAL); // Settle End -> Neutral Start
```

3.3.3 모션 장치 제어

모션 라이브러리는 생성된 개별 모션 샘플의 직접 제어 방식과 모션 베퍼 자동 제어 방식의 2 가지 모션 장치 제어 방법을 지원합니다. 모션 샘플 스트리밍은 연속된 샘플을 전달하는 시간을 직접 제어하기 때문에 자연 없이 정밀 제어가 가능하지만, 어플리케이션에서 모터의 처리 성능을 고려하여 직접 제어해야 하는 어려움이 있습니다. 반면, 모션 베퍼 스트리밍은 정의된 샘플링 주기 마다 자동으로 모션이 제어되기 때문에, 어플리케이션의 동기화가 용이하지만 약간의 자연이 발생 할 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

- 모션 디스패처 (Dispatcher)

VR 모션 시스템의 모션 제어는 샘플링 주기에 따라 모션 베퍼의 샘플 신호를 모션 플랫폼에 전달하여 제어 됩니다. 어플리케이션은 시뮬레이션 주기(frame rate)에 맞게 모션 시스템의 샘플링 주기를 일치 시키면 서브시스템간 동기화에 도움을 줄 수 있습니다.

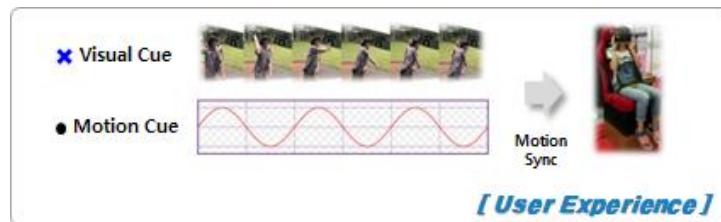


그림 25. 모션 신호 동기화

모션 디스패처 모듈은 어플리케이션의 프레임 레이트에 동기화된 모션 베퍼의 각 샘플들을 내부 스레드에서 모션 제어 주기에 따라 자동으로 실행합니다. 인간이 느끼는 운동감은 시각과 같은 다른 감각보다 더 민감하게 느낄 수 있기 때문에, 모션 베퍼를 구성하는 샘플수와 샘플링 주기는 성능적으로 중요한 기준이 될 수 있습니다.

- 역기구학(inverse kinematics)

모션 플랫폼의 자유도는 모션 테이블 상판의 중심을 원점으로 하는 위치와 자세(각도)를 나타내고, 모션 테이블에 연결된 액츄에이터 길이에 따라 변경됩니다. 모션 상판의 상대적인 변환 행렬로부터 액츄에이터의 길이를 구하고, 제 2 코사인 법칙을 통해 서보 모터의 회전각을 구할 수 있는데, 이 과정은 역기구학이라고 합니다.

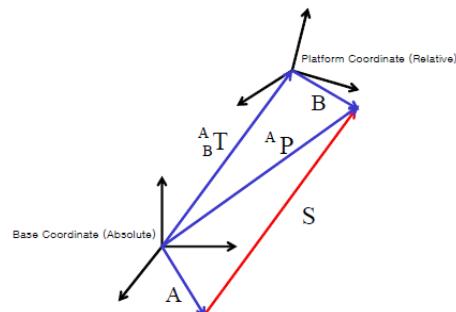


그림 26. 역기구학 개념

다음 수식을 통해 모션 자유도인 회전 행렬(R)과 이동 벡터(P)로 모션플랫폼의 A(초기위치)에서 B(현재위치)로의 상대적인 변환 행렬(Transform Matrix)를 구할 수 있고, 수학적 계산을 통해 그림과 같은 액츄에이터의 길이(S)를 구할 수 있습니다.

$$\begin{aligned} {}^A_B T &= \begin{bmatrix} {}^A_B R & P \\ 0 & 0 & 1 \end{bmatrix} & {}^A P &= {}^A_B T \times B \\ && S &= {}^A P - A \end{aligned}$$

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

- 모션 위치 제어

모션 제어는 역기구학으로부터 얻어진 모터 회전각(위치 지령)에서 회전 전압의 아날로그 신호를 컨트롤러에서 얻고, 모션 드라이버에 의해 AC 모터의 위치를 제어합니다. 3 자유도 VR 모션 플랫폼은 3 축 액츄에이터를 사용하고, 각 축별 지령(Command)에 맞게 서보 모터를 구동시키고 그 결과는 엔코더(Encoder) 값으로 확인할 수 있습니다.

(표 9. Diagnostic 구조체를 참고하세요.)

- 모션 장치 진단

모션 제어에 의해 처리된 모션 장치의 상태 조회 기능이 제공됩니다. 모션 플랫폼 각 축별로 장치의 장애 상태를 진단하거나 위치 제어 명령의 처리 상태를 조회 할 수 있습니다.

```
IM_DIAGNOSTIC_AXIS_INFO info[3];
int error = imGetDiagnostic(info, 3);
if(error > 0) exit(0); // critical error
if(error != 0) return; // connection failed
```

모션 장치 연결 실패시 -1 의 에러 코드를 반환하고, 각 축 별 알람(alarm) 및 명령 처리 상태(busy) 등의 상세 정보를 얻을 수 있습니다. 특히 아래 표의 1 이상 에러 코드는 장치 진단 구조체 없이도 조회 가능하도록, 즉시 정지되는 중요 장애 코드를 따로 반환 합니다.

표 8. 에러 코드

code	설명	비고
0	모션 장치 제어가 가능한 정상 응답 상태입니다.	
-1	모션 장치와 연결이 끊어진 경우 (인터넷 IP 설정이나 연결 상태 확인 필요)	
-2	소켓 초기화에 실패한 경우	
-3	지원되지 않는 함수를 호출한 경우	
-4	함수 인자 값 오류	
-5	함수 호출 시 구문 오류	
-6	함수 호출 시 통신 에러 발생	
-7	구동중인 축에 이동 명령을 전달한 경우	
-8	함수 호출 시 응답이 없는 경우	
-9	함수 호출 시 식별코드 에러가 발생한 경우	
-10	연결 시 네트워크 장치가 없을 경우 (방화벽 설정이나 연결 상태 확인 필요)	
1	라이브러리의 critical 런타임 오류 (모션 객체 메모리 확인 필요) (라이브러리 관련 상세 에러 코드(0x1000 ~ 0x8000)가 추가될 예정입니다. - IBase(0x1000 ~ 0x5000), IMotion(0x6000), InnoML(0x7000 ~ 0x8000))	
2	비상 정지 상태. 모션 드라이버의 긴급 상황 발생으로 긴급 정지 합니다.	
3	알람 상태. 모션 드라이버의 알람 발생으로 해당 축은 즉시 정지 합니다.	

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

그리고, 각 축 cmd 과 enc 는 역기구학을 통해 계산된 서보 모터 회전각의 제어 명령과 그 처리 결과입니다. 위치 이동에 시간이 소요 되는 경우 두 값의 편차가 클 수 있고 명령이 완료 될 때(busy off) 까지 다음 요청 명령은 대기 합니다.

표 9. Diagnostic 구조체 (IM_DIAGNOSTIC_AXIS_INFO)

변수	설명	비고
bBusy	펄스 출력 상태. 이 정보를 활용하여 모션 구동의 완료 여부를 확인합니다.	R
bHome	홈 센서 상태. 홈 센서는 모터에 가까운 쪽에 있는 원점 센서로 최초 구동시 각 축의 원점 운전시 사용 됩니다.	R
bAlarm	알람 상태. 서보 모터나 드라이버에 이상이 있는 경우 활성 되고, 해당 축만 정지 합니다.(참고, 비상 정지는 모든 축 정지)	R
bInpos	위치 결정 완료 상태. 지령한 위치에 도달한 완료 상태를 알 수 있지만, 기구적인 관성에 의한 시간 소요로 위치 제어 동기화에 부적합 할 수 있습니다. (bBusy 사용 권장)	R
bEmer	비상 정지 상태. 긴급 상황 발생 여부로 모터는 긴급 정지 합니다.	R
dCmd	위치 지령(모터 회전량). 역기구학으로 계산된 위치(서보 모터의 회전각) 요청 값입니다.	R
dEnc	엔코더 모터 회전량. 모션 드라이버에 의해 수행된 회전 펄스의 위치 응답 값. dCmd 값과의 편차는 급격한 위치 이동에 의해 발생 되므로 주의가 필요 합니다. (참고, 엔코더는 모터 회전량을 검출하는 장치입니다)	R
bCurrentOn	모터 전류 출력 설정. 모터가 기구적으로 체결되어 외력으로 움직일 수 없는 상태가 되는데, 비활성시 모터를 Free 상태로 변경 합니다.	RW
bServoOn	서보 운전 가능 상태. 지속적인 HW 장애 발생시 서보 모터 운전 상태의 재활성이 필요할 수 있습니다. 서보 ON 신호 전달시 “빼~”하는 초음이 발생할 수 있습니다.	RW
bDCCOn	편차 카운터 클리어(DCC) 활성 여부. 모터의 구동은 지령과 엔코더의 편차에 의해 제어되는데, 이 신호로 카운트를 소거합니다.	RW
bAlarmResetOn	알람 상태 리셋. 컨트롤러 내부에 활성된 알람 상황을 해제 합니다. (알람 발생 원인을 제거 후 리셋을 권장하고, 해제 되지 않는 알람은 전원 리셋이 필요합니다)	RW

3.3.4 모션 중첩(Superposition)

지금까지 살펴본 모션 장치 컨텍스트 기본 기능은 IMotion 라이브러리의 기능을 InnoML 의 컨텍스트에서도 연동하여 제공하는 공통 기능입니다. InnoML 라이브러리의 컨텍스트 객체는 이 IMotion 의 기본 기능을 모두 포함하면서 상위 다른 모션 객체와의 연동을 위해 모션 중첩이라는 기능을 추가로 지원 합니다.

IMotion 의 마스터 소스 관리 객체(IMotion_Source)는 어플리케이션이나 InnoML 로부터 전달 받은 하나의 마스터 베퍼(최종 중첩 결과)를 재생하는 기능을 수행합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

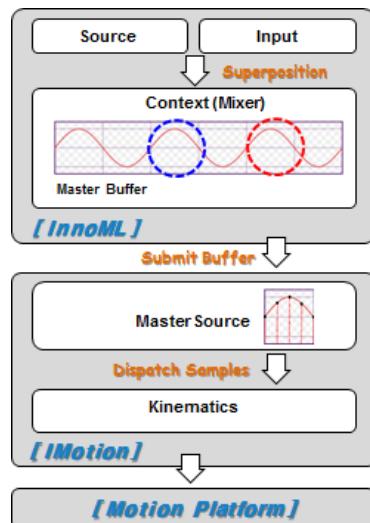


그림 27. 모션 중첩 처리 구조

어플리케이션이 제어하는 모션 신호는 지속적으로 가상 객체의 운동을 측정하여 활용하는 방법과 순간적인 운동 효과를 재현해 주는 방법이 사용될 수 있습니다. 실시간 시뮬레이션 시스템에서 동역학 모델에 대한 복잡한 연산에 따른 처리 성능 때문에 좀 더 단순한 수학적 모델의 시뮬레이션이 필요합니다.

모션 중첩은 모션 신호를 샘플링하여 만들어진 모션 데이터베이스와 입력 스트림으로부터, 실시간으로 모션 조건(강도, 주기)에 따라 중첩하여 모션을 제어 하는 방법입니다. 모션 컨텍스트는 다음과 같은 과정으로 모션 제어를 수행합니다.

- 소스 입력

컨텍스트는 여러 모션 소스들을 입력으로 받아 믹싱(중첩) 하고, 그 결과를 마스터 버퍼에 저장하여 모션 장치로 출력을 준비 합니다. 컨텍스트의 믹서는 주로 모션 효과의 참조 버퍼들과 모션 입력 버퍼들을 믹싱의 소스로 사용 합니다.

(모션 소스와 입력의 상세 설명은 “3.4 모션 소스”와 “3.5 모션 입력”을 참고 하세요.)

또한, 모션 입력 스트리밍과 유사한 방법으로 어플리케이션에서 모션 데이터를 마스터 버퍼에 직접 저장하여 믹싱 소스를 생성할 수 있습니다.

```

short buf[SAMPLE_RATE][SAMPLE_CHANNELS] = {0,};
memset(buf, 0, sizeof(short)*SAMPLE_RATE*SAMPLE_CHANNELS);
for(int i=0; i<SAMPLE_RATE; i++) {
    buf[i][0] = (short)(sin(2 * IM_PI * (float)i/SAMPLE_RATE)*MOTION_MAX_16);
    imBufferEnqueue(master_buffer, buf[i], sizeof(int16)*SAMPLE_CHANNELS);
    Sleep(1000/SAMPLE_RATE);
}

```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

- 모션 버퍼 혼합 (Mixer)

추출된 서로 다른 모션 파형들은 실시간으로 각 채널별로 혼합되어 최종 모션 샘플이 만들어집니다. 컨텍스트는 모션 소스들을 중첩하여 저장한 마스터 버퍼를 IMotion 라이브러리에 연동하여 모션 장치를 제어합니다.

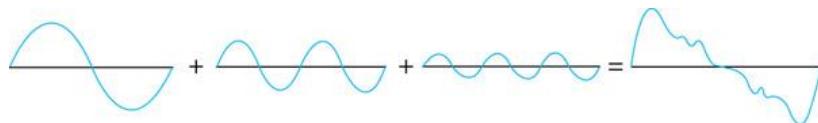


그림 28. 모션 파형의 합성

그림처럼 모션 파형 y_1, y_2, y_3 의 합성파 y 는 $y = y_1 + y_2 + y_3$ 로 중첩(superposition) 되어 나타납니다. 일반적으로 임의의 파형은 다수의 단진동(Sine 파)의 중첩으로 표시될 수 있습니다.

모션 입력 스트리밍의 경우 합성할 입력 버퍼를 어플리케이션으로부터 직접 전달받기 위해, 모션 컨텍스트는 멀티 스레드 실행 주기마다 사용자 입력 콜백 함수를 내부적으로 호출합니다. 모션 입력(예, y_1)의 경우 모션 효과(예, y_2, y_3)에 의해 저주파 신호가 주로 사용되기 때문에, 이 콜백 함수에서 저주파 통과 필터(LPF)와 같은 필터를 적용할 수 있습니다.

모션 입력에 대한 상세 설명은 “3.5 모션 입력”을 참고하세요.

- 모션 버퍼 제출

어플리케이션에서 정의한 마스터 버퍼의 크기만큼 합성이 완료되어 모션 장치에 제출할 때, 이전에 제출된 버퍼가 모션 컨트롤러에서 완료 될 때 까지 우선 대기 합니다. 모션 컨텍스트는 별도의 모션 스레드 (thread)에서 마스터 버퍼를 관리하는데, 컨텍스트 생성에서 정의된 모션 버퍼 처리 시간마다 작업 버퍼(back buffer)와 제출 버퍼(front buffer)를 서로 교환하여 모션 컨트롤러에 제출합니다.

3.3.5 주요 API

모션 컨텍스트는 가상 세계의 운동을 현실 세계에서 운동감을 느끼도록 전체적인 모션 제어 흐름(맥락)을 관리합니다. 이전 장에서 논의한 장치 컨텍스트 생성, 모션 장치 상태 변경 및 진단, 마스터 버퍼 관리 등의 주요 기능을 담당합니다. (상세한 설명은 API Reference 를 참고하세요.)

표 10. Context 주요 API

API	설명
imCreateContext (buffer, id, desc)	모션 장치 컨텍스트 객체를 생성합니다. 마스터 버퍼 및 IP, 장치 옵션을 지정할 수 있습니다.
imSetContext()	지정된 컨텍스트 객체를 활성 컨텍스트로 전환 합니다.
imGetContext()	현재 활성 객체를 얻습니다.
imGetBuffer()	마스터 버퍼를 조회합니다. 마스터 버퍼의 변경은 지원하지 않으며, 컨텍스트를 다시 생성해야 합니다.
imSetFilter(filter)	마스터 버퍼의 출력을 변환하는 필터를 설정(조회) 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

imGetFilter()	(필터 그래프를 사용하면 필터들이 동시에 처리됩니다.)
imStart()	모션 장치를 준비(중립) 상태로 변경하고, 컨텍스트(마스터 버퍼 스트리밍)를 시작합니다.
imStop()	컨텍스트(마스터 버퍼 스트리밍)를 정지 하고, 모션 장치를 중립 또는 초기 상태로 변경합니다.
imGetDescription()	지원 모션 장치들의 정보를 조회합니다
imGetDiagnostic()	모션 장치 상태 정보를 조회합니다.
imGetPlayingSourceCount	재생 중인 모션 소스의 수를 얻습니다.
imSetMasterVolume()	마스터 볼륨을 설정 합니다. 볼륨을 0으로 하여 모션 플랫폼을 일시정지(Silence) 할 수 있습니다.
imStopAllSources()	재생 중인 모든 모션 소스를 정지합니다.
imDestroyContext()	모션 장치 컨텍스트 객체를 제거합니다. 모션 디바이스 연결을 해제합니다.

3.4 모션 소스(imSource)

모션 버퍼를 처리 하는 과정은 모션 버퍼에 데이터를 저장하고 읽는 인코딩과 디코딩 과정을 수행합니다. 모션 파일 로딩이나 미리 생성된 모션 데이터(imBufferEnqueue)를 중복 사용 할 때마다 인코딩 없이도 재사용할 수 있도록, 모션 버퍼를 모션 데이터베이스에서 관리합니다. 모션 데이터베이스는 모든 모션 관리 객체들 간에 공유가 가능하고, 유일한 모션 버퍼를 ID로 재사용할 수 있습니다.

모션 소스는 모션 데이터베이스에 등록된 모션 버퍼를 참조하여, 정의된 포맷에 따라 특정 시간의 샘플 취득 후 모션 위치를 복원하는 디코딩 과정을 수행합니다. 모션 소스는 모션 버퍼의 디큐(imBufferDequeue)와는 다르게 HEAD 위치의 변화 없이 디코딩 위치를 따로 관리하여 재생 제어 합니다.

모션 소스는 참조 모션 버퍼의 재생 관리 객체이며, 모션 효과를 재생하는 다양한 방법을 제공합니다. 모션 버퍼는 데이터베이스에서 식별된 동일 모션 파형을 재생하는 파라미터에 따라 다양한 새로운 파형이 변조 되어 출력될 수 있습니다.

이번 장에서는 모션 소스의 모션 효과 파일 및 재생 제어 관리 주요 기능을 알아봅니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

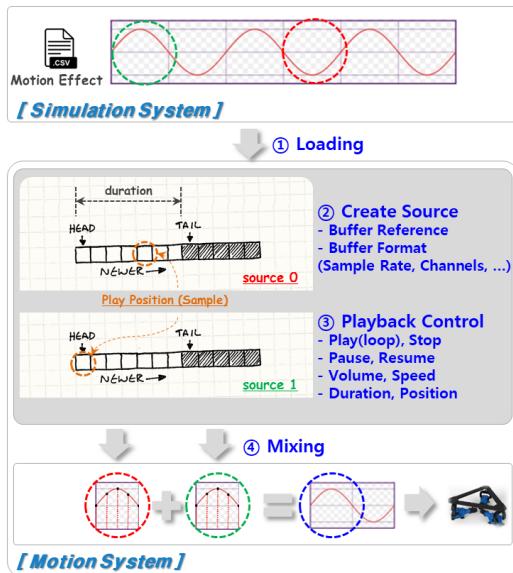


그림 29. 모션 소스 주요 기능

3.4.1 소스 생성

모션 소스는 주로 미리 저작된 모션 효과 파일이 저장된 모션 버퍼의 재생을 관리합니다. 모션 소스의 생성을 위해 모션 효과를 저장하는 모션 버퍼를 생성해야 합니다.

모션 데이터 각 채널의 모션 파형을 시간 도메인으로 표현할 때 시간과 운동량을 축으로 하는 파형을 나타낼 수 있습니다. 아래 그림과 같은 Sine 파를 모션 버퍼로 생성할 때, 재생 파라미터는 진폭(운동 범위)과 주기(진동수)로 정의할 수 있습니다. 모션 효과의 파형은 모션 입력 파형에 비해 고주파 성질을 갖습니다.

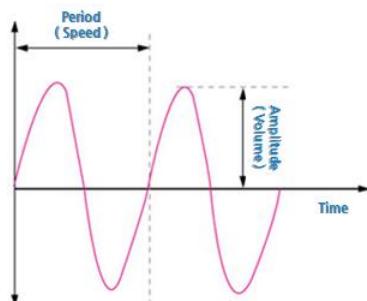


그림 30. 모션 파형의 진폭과 주기

모션 효과 파일은 자유도 각 채널마다 모션 파형을 구성하도록 텍스트 편집기나 오디오 편집 툴과 같이 파형을 저작할 수 있는 편집도구에서 텍스트 파일(.CSV)을 생성할 수 있지만, SDK에 포함된 모션 데이터 편집기를 통해 쉽게 모션 효과 파일을 생성 할 수 있습니다.

모션 파일 생성에 관한 자세한 설명은 “2.1.3 모션 편집기”를 참고 하세요.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

```
IMBuffer effect = imLoadBuffer("waveform_sine.csv"); // load motion effect file
```

모션 시스템을 사용하는 시뮬레이션 환경에 따라 모션 파일의 이름이 아닌 파일 메모리로부터 직접 로딩을 지원하며, 저작권 보호를 위해 암호화된 모션 효과 파일의 로딩도 지원합니다.

```
fread(data, 1, size, fp); // file open & read  
IMBuffer effect = imLoadBufferMemory(data, size, "decrypt key"); // load motion file memory
```

모션 효과 파일 로딩이나 어플리케이션에서 생성한 모션 버퍼의 실제 메모리는 내부적으로 전역적인 공간에 리소스(asset)로 관리됩니다. 모션 소스는 앞에서 생성한 모션 버퍼를 참조하여 다음과 같이 생성되어야 합니다.

```
int duration = imBufferGetDuration(effect);  
IMSource source = imCreateSource(effect);  
...
```

3.4.2 소스 버퍼 설정

모션 소스에 할당된 참조 버퍼는 언제든지 변경 될 수 있어야 합니다. 예를 들어, 모션 소스의 생성 시점에는 어떤 모션 효과를 재생할지 알지 못하고 콘텐츠의 상호작용에 따라 적절한 효과를 사용자에게 전달해야 할 경우가 있는데, 게임 같은 시뮬레이션 환경에서 모션 효과 변경과 조회는 실시간에 수행 될 수 있어야 합니다.

```
IMSource source = imCreateSource();  
imSourceSetBuffer(source, effect);
```

```
IMBuffer buffer = imSourceGetBuffer(source);
```

충돌이나 폭발과 같이 sine 파형 상하 진동 효과는 자주 사용되는 유용한 모션 효과가 될 수 있는데, 단순한 Sine 파형을 재사용하여 다양한 모션 효과를 구현할 수 있습니다. 아래 그림은 동일한 sine 파형의 모션 버퍼를 참조하는 3 가지의 모션 소스의 출력 파형을 도시 하고 있습니다.

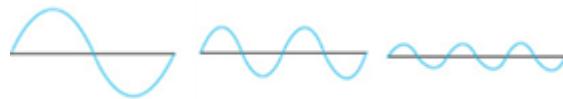


그림 31. 모션 소스 재생 파라미터 (1x, 2x, 4x)

이와 같이 미리 등록된 모션 데이터베이스의 특정 모션 파형은 실시간으로 파라미터가 적용되어 모션 파형을 재구성할 수 있는 기능을 제공합니다. 모션 소스는 모션 버퍼를 재생하는 파라미터를 재생 시점에 변경할 수 있는데 진폭과 주기의 비율을 각각 볼륨(volume)과 속도(speed)로 정의합니다. 아래에 Sine 파의 모션 버퍼의 재생 파라미터를 1x, 2x, 4x 배속을 적용하여 범프 효과(bump)나 진동 효과(vibration) 등의 파형을 실시간으로 재생할 수 있는 구현 예를 보이고 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

```
IMSource source1x = imCreateSource(effect);
IMSource source2x = imCreateSource(effect);
IMSource source4x = imCreateSource(effect);
// source2x : volume (0.5x), speed (2x)
imSourceSetVolume(source2x, 50);
imSourceSetSpeed(source2x, 200);
// source4x : volume (0.25x), speed (4x)
imSourceSetVolume(source4x, 25);
imSourceSetSpeed(source4x, 400);
...
```

3.4.3 재생 제어

모션 소스의 재생(play), 정지(stop), 일시정지(pause), 재시작(resume) 기능으로 모션 효과 재생 상태를 제어 합니다. 또한 모션 파형의 특정 패턴을 무한 반복 옵션으로 재생하거나 원하는 시간 까지 모션 파형을 반복적으로 재생할 수 있습니다.

```
imSourcePlay(source, IM_LOOP_INFINITE); // loop count
Sleep(1000);
imSourcePause(source, true); // pause (1 sec)
Sleep(1000);
imSourcePause(source, false); // resume (1 sec)
Sleep(1000);
imSourceStop(source); // stop (break)
```

모션 소스의 재생 시간은 모션 버퍼의 포맷 정보에 의해 계산되는데, 재생 위치는 디코딩된 모션 버퍼의 위치를 재생 시간 기준으로 계산하여 얻을 수 있습니다.

```
// duration(ms) : (encoded_size * 1000) / (sample_rate * sample_size)
duration = imBufferGetDuration(buffer);

// position(ms) : (decoded_size * 1000) / (sample_rate * sample_size)
position = imSourceGetPosition(source);
```

모션 소스의 재생 위치가 재생 시간에 도달하면 재생 완료를 판단할 수 있고, 반복 횟수를 지정하면 재생 위치를 다시 처음 위치로 재설정하여 반복 재생이 가능합니다. 또한 어플리케이션은 특정 모션 소스의 재생 완료 이벤트의 수신이 필요할 수 있는데, 라이브러리 내부에서 완료된 모션 소스의 콜백 함수를 호출하여 완료 이벤트를 수신할 수 있습니다.

```
static bool completed = false;
static void motion_event_handler(void* obj, unsigned int state) {
    if(state == IM_END_OF_STREAM) completed = true;
}
...
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

```
imSourcePlay(motion1, 0, motion_event_handler, &listener);
Sleep(500);
```

3.4.4 모션 베퍼 참조

모션 소스는 모션 베퍼를 읽기 전용으로 참조 하여 디코딩하는데, 동일 베퍼의 참조 방식에 따라 2 가지 경우를 생각해 볼 수 있습니다. 먼저 동일 베퍼를 참조하는 단일 소스를 여러 번 재생하는 경우, 이전 명령의 재생 위치는 다시 처음으로 초기화 되어 재시작 됩니다.

```
IMSource motion1 = imCreateSource(buffer);
imSourcePlay(motion1);
Sleep(500); // motion1 play_time : 500 ms
imSourcePlay(motion1);
int pos = imSourceGetPosition(motion1); // pos = 0
```

반면 동일 베퍼를 참조하는 복수 소스를 각기 재생 하는 경우, 서로 다른 모션 파형으로 취급되어 모두 중첩되어 출력됩니다. 즉, 모션 베퍼와 무관하게 모션 소스마다 독립적인 모션 데이터가 모두 모션 플랫폼으로 전달됩니다.

```
IMSource motion1 = imCreateSource(buffer);
IMSource motion2 = imCreateSource(buffer);
imSourcePlay(motion1);
Sleep(500); // motion1 play_time : 500 ms
imSourcePlay(motion2);
int pos = imSourceGetPosition(motion1); // pos = 500
```

3.4.5 주요 API

모션 효과 (베퍼) 재생 관리의 책임이 있는 Source 는 다음과 같은 주요 API 기능을 갖습니다. (상세한 설명은 API Reference 를 참고 하세요.)

표 11. Source 주요 API

API	설명
imCreateSource(buffer)	모션 소스 객체를 생성 합니다.
imSourceSetBuffer(source, buffer)	소스의 참조 베퍼를 변경(조회) 합니다.
imSourceGetBuffer(source)	베퍼의 변경 전 재생을 정지 합니다.
imSourceSetFilter(source, filter)	소스 객체의 출력을 변환하는 필터를 설정(조회) 합니다.
imSourceGetFilter(source)	(필터 그래프를 사용하면 필터들이 동시에 처리됩니다.)
imSourcePlay(source, loops, notify_cb, user_data)	소스의 재생을 시작 합니다. 반복 횟수 +1만큼 재생되고, 이벤트 리스너를 등록 할 수 있습니다.
imSourceStop(source)	소스의 재생을 정지 합니다.
imSourcePause(source, paused)	재생을 일시 정지 또는 재시작 합니다.
imSourceSetVolume(source, volume)	소스의 볼륨을 설정 합니다.
imSourceSetSpeed(speed)	소스의 재생 속도를 설정 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

imSourceGetPosition(source)	소스의 재생 시간을 얻습니다.
imDeleteSource(source)	모션 소스 객체를 제거 합니다.

3.5 모션 입력(imInput)

모션 입력은 어플리케이션에서 모션 데이터를 실시간 획득하는 방법과 모션 장치로의 스트리밍을 관리합니다. 모션 소스가 생성된 모션 버퍼의 출력을 위해 필요하다면, 이름에서 알 수 있는 것처럼 모션 입력은 모션 데이터 입력 스트림을 관리하는 객체입니다.

모션 입력 처리 과정은 입력 버퍼에 PCM 데이터를 저장하는 샘플링 과정과 모션 컨텍스트를 통해 모션 신호를 출력 하는 스트리밍 과정으로 이루어집니다. 전달되는 입력 신호가 이미 샘플링된 신호인 경우, 입력 버퍼에 큐잉(Enqueue) 동작만으로 모션 스트리머를 통해 샘플링 주기에 따라 입력 신호는 자동으로 출력 됩니다. 반면 전달되는 입력 신호가 아날로그 신호인 경우, 입력 버퍼에 저장 할 수 있도록 입력 샘플러를 통해 샘플링 되어야 합니다. 모션 입력 스트리밍 과정에서 필터를 함께 적용할 수 있는데, 이를 위해 콜백 함수를 미리 등록해야 합니다.

3.5 장에서는 모션 입력의 스트리밍 및 샘플링 등의 주요 기능을 알아봅니다.

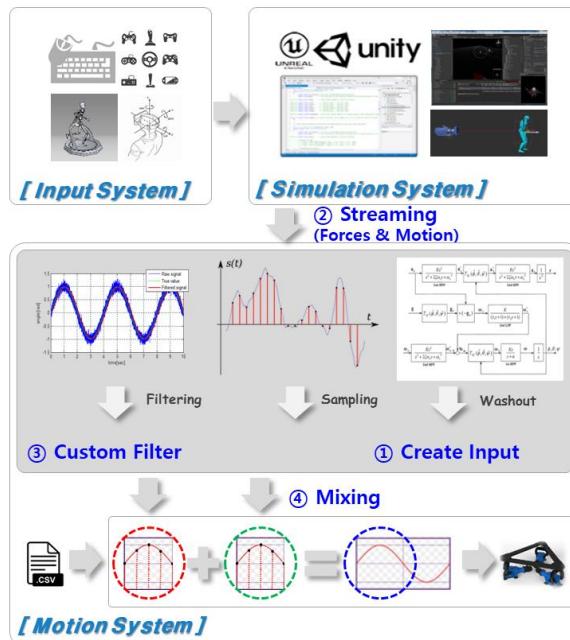


그림 32. 모션 입력 주요 기능

3.5.1 입력 생성

게임과 같은 Interactive VR 콘텐츠에서 게임 컨트롤러와 같은 실시간 상호작용에 따라 운동 객체의 위치와 자세가 동적으로 변경됩니다. 어플리케이션에서 전달된 모션 데이터를 가동하여 모션

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

컨텍스트를 통해 모션 장치에 전달하는 과정을 모션 입력 스트리밍이라고 합니다. 모션 입력 객체를 생성한 다음 스트리밍을 시작(Start)하면 종료(Stop)될 때까지, 모션 입력 버퍼에 데이터를 읽고 쓰는 입력 스트리밍 과정을 지속적으로 수행 합니다.

```
IMBuffer input_buffer = imCreateBuffer(50, 0, 0, 4, 2);
IMInput input = imCreateInput(input_buffer);
...
imDeleteInput(input);
```

어플리케이션에서 전달되는 모션 입력 신호의 제어를 위해 모션 플랫폼의 제어 위치를 저장할 수 있는 모션 버퍼 메모리 공간이 필요하고, 연속적인 신호를 실시간 처리하기 위해서는 최소 버퍼의 수를 2 이상의 환형 큐 구조로 생성해야 합니다("3.2 모션 버퍼" 참고).

3.5.2 입력 버퍼 설정

입력 버퍼가 설정되면, 어플리케이션에서 전달되는 모션 입력 신호의 레코딩을 시작할 수 있습니다. 모션 입력은 입력 버퍼에 데이터를 저장 하는 과정과 입력 버퍼를 모션 장치로 전달하는 과정으로 처리됩니다.

첫째, 어플리케이션에서 입력 버퍼에 저장(큐잉)한 입력 신호를 읽어서(디큐잉), 연속적으로 모션 장치에 전달하는 기본 과정을 입력 스트리밍이라고 합니다. 모션 입력을 시작하면 내부적으로 모션 컨텍스트의 믹싱 스레드에서 주기적으로 입력 버퍼로부터 획득한 입력 샘플을 마스터 버퍼에 자동으로 중첩하여 모션 플랫폼을 제어하게 됩니다.

아래 예와 같은 모션 입력의 명시적인 시작과 종료의 기본 기능은 입력 스트리밍 처리 구간을 명시하는 것입니다. 이때 저장된 신호는 샘플링된 PCM 데이터로 간주 합니다.

```
imInputSetBuffer(input, input_buffer);
imInputStart(input);
// sampling or streaming loop
...
imInputStop(input);
```

둘째, 어플리케이션에서 실시간으로 전달하는 불규칙한 모션 신호에서는 실시간 샘플링 단계를 거친 결과를 스트리밍 해야 합니다. 모션 입력의 샘플링 방법은 아래 그림과 같은 "순차적 등가 시간 샘플링" 방법을 사용한 실시간 샘플링을 수행 합니다. 아래 그림 하단의 입력 버퍼에 정의된 샘플링 주기에 따라 동일 시간마다 단일의 샘플을 수집하여 입력 버퍼에 저장(큐잉) 하는 방식을 사용합니다. 실제적인 입력 버퍼에 입력 신호의 저장(큐잉)은 내부 입력 샘플러가 수행 합니다.

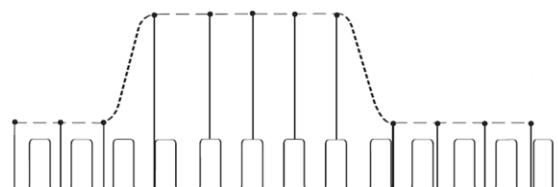


그림 33. 실시간 샘플링 방식

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

3.5.3 입력 스트리밍

360 video 과 같은 Passive VR 콘텐츠에서의 운동 객체는 주로 카메라 위치 및 자세 변화에 따라 정적으로 변경 됩니다. 정적인 모션 스트림의 경우 콘텐츠 재생 시간에 따라 수 분에서 수 시간의 비교적 많은 데이터 처리를 요구 하는데, 영상이나 네트워크로부터 수신되어 인코딩된 모션 데이터 블록을 연속적으로 처리해야 합니다.

어플리케이션은 입력 샘플러에 실시간 모션 샘플 스트림을 전달 하지만, 입력 스트리머는 이미 샘플링된 모션 버퍼 스트림을 전달하여 콘텐츠와 동기화 합니다. 이와 같이 이미 인코딩되어 전달된 모션 데이터 블록은 그대로 모션 버퍼에서 사용할 수 있기 때문에 생성된 입력 버퍼에 입력된 데이터를 큐잉 하는 것만으로 모션 입력 데이터 스트리밍이 가능합니다.

입력 스트리밍은 시작과 종료의 명시적인 실행 구간동안 수행 되며, 어플리케이션에서 큐잉(Enqueue)한 모션 입력 샘플은 내부 스트리머에서 막싱 주기 마다 자동으로 디큐잉(Dequeue) 합니다.

모션 입력 스트리머를 통한 모션 데이터 스트리밍 예는 다음과 같습니다.

```
imInputStart(input, motion_filter_callback, NULL);
...
short stream[SAMPLE_COUNT * SAMPLE_CHANNELS];
while(time) {
    imBufferEnqueue(input_buffer, stream, size);
}
imInputStop(input);
```

3.5.4 입력 샘플링

1 장의 VR 모션 시스템 구성도에서 설명한 것과 같이 시뮬레이션 시스템으로부터 2 가지 입력 방법을 지원 합니다. 운동 객체의 선형 가속도와 원심력을 입력 신호로 받아 모션 플랫폼의 위치/자세로 시뮬레이션 하는 방법(Force Simulation Mode)과 모션 플랫폼의 위치/자세를 입력 신호로 받아 모션의 위치를 직접 제어 하는 방법(Direct Positioning Mode)을 지원합니다.

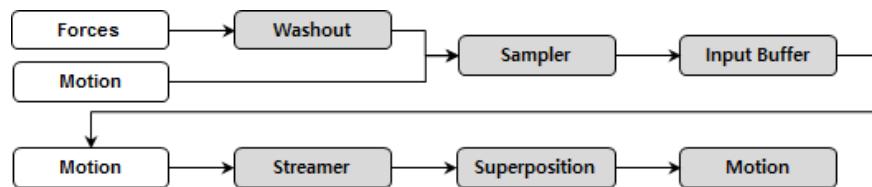


그림 34. 모션 입력 샘플링 처리 흐름

모션 입력 샘플러를 통한 모션 데이터의 샘플링 예는 다음과 같습니다.

입력 스트리밍 실행 구간동안 입력 스트림 전달(imInputSendStream 사용) 시점을 기준으로 시간 도메인의 모션 입력 파형이 입력 버퍼에 저장되게 됩니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

```

imInputStart(input, motion_filter_callback, NULL);
...
// Direct Positioning Mode (Sample Stream)
short sample[SAMPLE_CHANNELS];
while(time) {
    imInputSendStream(input, sample, size);
}
imInputStop(input);

```

하지만, 이 방법은 샘플이 획득된 시점과 API 호출 시점을 동일시점으로 가정하는 논리적 오류가 있고, 대부분의 게임 엔진과 같은 실시간 시뮬레이션 환경에서 물리(동역학) 서브시스템의 실행 주기는 고정되어(예: FixedUpdate) 이미 샘플링된 PCM 데이터이므로 실시간 샘플링의 필요성은 점차 줄어들고 있습니다. 시뮬레이션 주기에 맞는 모션 입력 주기 설정과 PCM 데이터 스트리밍(imBufferEnqueue)을 권장합니다. (주의, 모션 입력 샘플러는 향후 제거 될 수 있습니다.)

3.5.5 사용자 입력

어플리케이션으로부터 전달된 모션 입력 신호는 불규칙한 노이즈 신호를 제거하는 저주파 통과 필터(LPF)나 워시아웃 필터로 제한된 운동 범위에 맞는 고주파 및 저주파 신호로 변조가 필요합니다. 또한, 입력 신호로 전달된 원격 측정 데이터(Telemetry)의 경우 다양한 모션 제어 파라미터를 모션 위치로 매핑하는 스케일러 사용자 정의 필터가 필요할 수 있습니다.

```

typedef struct {
    ...
    float wx, wy, wz;      /* < roll, pitch, yaw (speed, radians/s) */
    float ax, ay, az; /* < surge, sway, heave (acceleration, m/s^2) */
} SIM_TELEMETRY_MESSAGE;

static int telemetry_filter_callback(void* netstream, void* stream, int len)
{
    IMBuffer input_buffer = (IMBuffer)netstream;
    SIM_TELEMETRY_MESSAGE message[SAMPLE_COUNT];
    If(imBufferDequeue(input_buffer, message, sizeof(message)) == 0)
        return 0;

    // simple scaler (Telemetry -> DOF)
    PLATFORM_POSITION_MESSAGE* dof = (PLATFORM_POSITION_MESSAGE *)stream;
    for(int i=0; i<SAMPLE_COUNT; i++) {
        doff[i].heave = message[i].ay * MOTION_MAX_16;
    }
    return len; // mix size
}

```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

이처럼 모션 입력 필터를 적용하기 위해 사용자 입력 콜백 함수를 등록하여 저장된 입력 버퍼의 데이터를 읽고 필터링 한 후 모션 장치에 전달할 수 있습니다. 모션 입력 스트리밍은 마스터 버퍼의 샘플링 주기마다 이 콜백 함수를 실행하게 되는데, 스트리밍 데이터를 읽어 필터링한 결과를 다시 전달하는 과정으로 처리됩니다.

```
input_buffer = imCreateBuffer(SAMPLE_RATE,
    IM_FORMAT_DATA_F32, TELEMETRY_CHANNELS, SAMPLE_COUNT, 2);
imInputSetBuffer(input, input_buffer);
// Forces Simulation Mode (Telemetry Stream)
imInputStart(input, telemetry_filter_callback, (void*)input_buffer);
SIM_TELEMETRY_MESSAGE telemetry;
memset(&telemetry, 0, sizeof(SIM_TELEMETRY_MESSAGE));

unsigned int dt = 1000/SAMPLE_RATE;
for(int time=0; time<=1000; time+=dt) {
    telemetry.ay = sin(2 * IM_PI * (float)time/1000);
    imInputSendStream(input, &telemetry, sizeof(SIM_TELEMETRY_MESSAGE));
    Sleep(dt);
}
imInputStop(input);
```

모션 입력 필터의 주요 기능은 시뮬레이션 데이터(Force, 6 DOF, ...)를 VR 모션 플랫폼 데이터(3 DOF)로 매핑(스케일) 처리로 요약할 수 있습니다. 시뮬레이션 시스템의 처리 결과(가상 객체의 운동 정보)를 모션 입력으로 매핑하는 모션 입력 필터의 추가만으로 VR 모션 서브시스템을 구성 할 수 있고, 시뮬레이션 로직의 수정 없이도 모션 플랫폼에 의존성 없이 쉽게 연동할 수 있습니다.
(Telemetry 사용에 대한 상세한 설명은 main_input.cpp 예제를 참고하세요.)

3.5.6 주요 API

VR 콘텐츠로부터 모션 데이터 스트림 입력 관리의 책임이 있는 Input 는 다음과 같은 주요 API 기능을 갖습니다. (상세한 설명은 API Reference 를 참고하세요.)

표 12. Input 주요 API

API	설명
imCreateInput(buffer)	모션 입력 객체를 생성(제거) 합니다
imDeleteInput(input)	입력 버퍼를 지정해야 스트리머가 작동합니다.
imInputSetBuffer(input, buffer)	입력의 참조 버퍼를 변경(조회)합니다.
imInputGetBuffer(input)	imBufferEnqueue 함수로 PCM 샘플을 직접 저장할 수 있습니다.
imInputSetFilter(input, filter)	입력 객체의 출력을 변환하는 필터를 설정(조회)합니다.
imInputGetFilter(input)	(필터 그래프를 사용하면 필터들이 동시에 처리됩니다.)
imInputStart(input, streamer_cb, streamer_data)	입력 스트리밍을 시작합니다. 마스터 버퍼의 헙성 주기마다 스트리밍 콜백이 호출됩니다.
imInputStop(input)	입력 스트리밍을 정지합니다.
imInputSendStream(input, data, size)	입력 객체에 모션 샘플 스트림을 전달합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

	모션 입력 샘플 데이터와 그 크기를 지정합니다.
--	----------------------------

3.6 모션 필터(imFilter)

모션 필터는 모션 신호의 변환을 관리하는 객체입니다. 3.5 장에서 알아본 것처럼 시뮬레이션 결과를 모션 플랫폼의 위치로 변환하는 다양한 알고리즘이 모션 필터로 추가 될 수 있습니다.

모션 플랫폼의 DOF 변환과 운동 범위 및 속도 제한 등의 장치 지원 스펙에 맞는 호환성을 유지하거나, 막상 완료된 최종 신호에 특별한 효과를 적용하기 원하는 경우 마스터 필터를 사용할 수 있습니다. 또한, 모션 효과를 더욱 부드럽게 만들어서 진동, 범프, 충돌 등 모션 효과의 수준을 다시 결정하는 다양한 DSP 필터들이 사용될 수 있습니다.

라이더를 움직이는 모션 플랫폼의 목적은 실세계에서 느껴지는 것과 유사한 가속 감각을 재현하는 것입니다. 실세계를 시뮬레이션하기 위해 모션을 큐잉하는 가장 일반적인 방법은 고전적인 워시아웃 알고리즘을 사용하는 것입니다. 이번 장에서는 가장 많이 사용되는 워시아웃 필터와 자주 사용되는 몇 가지 모션 효과 필터에 대해 알아 봅니다. 그리고, 필터들을 모션 라이브러리의 4 가지 관리 객체(모션 컨텍스트, 모션 버퍼, 모션 소스, 모션 입력)에 적용하는 방법을 알아 봅니다.

3.6 장에서는 모션 필터의 구성 및 처리 과정 을 알아봅니다.

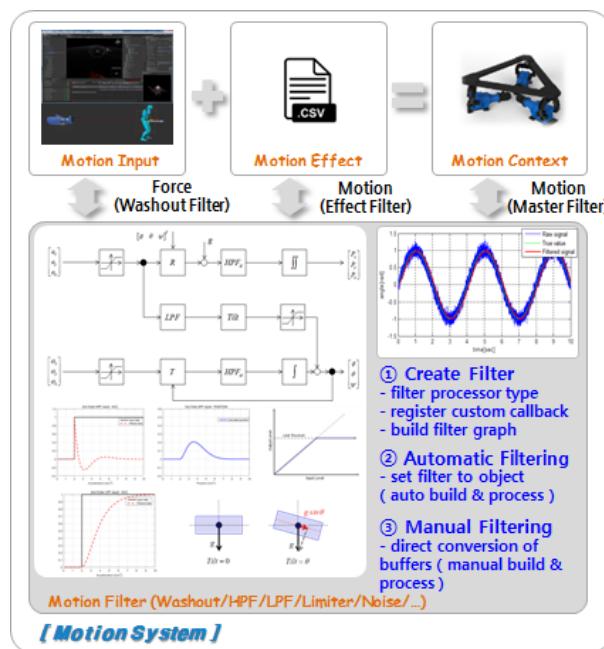


그림 35. 모션 필터 주요 기능

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

3.6.1 필터 생성

디지털 신호 처리에서 입력 신호를 변환하기 위해 다양한 필터 처리기가 사용될 수 있습니다. 모션 라이브러리는 모션 신호의 필터링을 위해 공통되고 자주 사용하는 모션 필터 처리기 (Processor)를 내장하고 있으며 사용자 필터로 확장을 지원합니다. ("IMotion_types.h"의 IM_FILTER_TYPE 참고)

모션 큐잉을 위해 자주 사용되는 내장 필터 처리기는 다음과 같습니다.

표 13. 필터 처리기 (IM_FILTER_*)

타입	설명	PARAMS
DEFAULT	기본 모션 필터 타입을 정의합니다. 필터 처리기와 파라미터 없이 단순히 필터 그룹을 정의하는데 사용됩니다. 필터를 벌드(SetFilter)하기 전에 자주 사용되는 필터들을 미리 등록하여 사용을 권장합니다.	None
NOISE	모션 신호에 포함된 노이즈를 제거하여 부드러운 파형을 얻기 위해 사용 됩니다. 예측 및 측정 노이즈의 공분산을 상수로 정의합니다.	$nCovariance$ (5, 0~100)
MEAN	모션 신호의 평균값으로 부드러운 파형을 얻습니다. 이동 평균 필터로 알려진 이 필터는 최근 $nCount$ 개 값 만을 계산하므로 실시간 처리에 적합합니다.	$nCount$ (4, 0~16)
HIGHPASS	모션 신호에서 차단 주파수 이상의 고주파 성분을 통과 합니다. 주로 범프, 진동 효과와 같은 불규칙적이고 빠른 변화를 얻기 위해 사용 됩니다. 최대 3차까지 필터의 차수($nOrder$) 사용이 가능합니다.	$fCutoffFrequency$ (5, 0~10)
		$nOrder$ (1, 0~3)
LOWPASS	모션 신호에서 차단 주파수 이하의 저주파 성분을 통과합니다. 주로 모션 입력에서 불규칙한 잦은 변화를 줄이고 주된 운동감을 얻는데 사용됩니다. 자연이 발생될 수 있기 때문에 주의가 필요합니다. 최대 2차까지 필터의 차수($nOrder$) 사용이 가능합니다.	$fCutoffFrequency$ (5, 0~10)
		$nOrder$ (1, 0~2)
INTEGRAL	모션 값을 적분합니다. 가속도에서 속도를 얻거나 위치를 얻는데 최대 2차까지 적분 차수($nOrder$)가 사용됩니다.	$nOrder$ (1, 0~2)
TILT	틸트 보정을 수행합니다. 중력에 의한 가속력을 느낄 수 있도록 틸팅 각도만큼 추가해 줍니다. $Tilt += \text{asin}(a/g)$	None
WASHOUT	6-DOF의 논리적 데이터(가속도, 각속도)를 3-DOF의 물리적 데이터(위치, 각도)에 맞게 변환 합니다. 각 축의 차단 주파수가 클 수록 반응의 크기가 작고/수렴 속도가 빠릅니다.	$fCutoffFrequency$ (5, 0~10)
SCALE	진폭을 스케일 합니다. 입력 또는 신호의 범위를 조정할 수 있습니다. $Output = \text{Clamp}(\text{Input} * fScaleFactor, -32767, 32767)$	$fScaleFactor$ (1, ~)
LIMIT	특정 범위로 그 값을 잘라냅니다. 운동 범위를 제한하기 위해 사용 합니다. $Output = \text{Clamp}(\text{Input}, nRangeMin, nRangeMax)$	$nMin$ (-32768, ~)
		$nMax$ (32767, ~)
RATELIMIT	변화량이 최대 속도를 넘지 않도록 그 값을 잘라냅니다. 운동 속도를 제한하기 위해 사용 합니다. 주로 장치에 주는 데미지를 방지 하기 위해 사용됩니다. (밀리초 당 동작량으로, 변환에서 이 값에 샘플시간을 곱하여 사용합니다.) $Output = \text{Clamp}(\text{Input}, Input0 - nRateMax * dt, Input0 + nRateMax * dt)$	$nRateMax$ (256, ~)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

FORMAT	데이터 타입을 변환합니다. (내부 필터)	None
CHANNEL	채널수 변환 및 채널을 매핑합니다. (내부 필터)	None
RESAMPLE	샘플링 주기를 변환합니다. (내부 필터)	None
CUSTOM	사용자 콜백 함수로 필터링을 수행합니다.	User Data

모션 포맷 변환기는 데이터 타입 변환, 채널 변환, 샘플링 주기 변환을 지원하며, 필터의 빌드 과정에서 등록된 필터가 포맷 변환에 필요한지 내부적으로 판단하여 처리기를 작동 시킵니다. 예를 들어, 소스 버퍼에 사용된 6 채널 신호는 3 DOF 모션 장치로의 출력을 위해, 입력 신호 중 어떤 3 개의 채널을 어떤 순서로 각 축에 매핑할 것인가를 정의한 채널 변환기가 필요합니다. 이 경우 모션 입력 신호의 채널 수를 항상 6 채널로 처리하고, 최종 믹싱된 결과를 디바이스에 전달할 때 각 축에 맞게 채널을 매핑하는 것은 모션 플랫폼 의존성을 제거하는 좋은 방법입니다. 포맷 변환기는 모션 버퍼의 크기를 변경할 수 있어 주의가 필요한 중요한 변환기 이므로, 사용자가 직접 추가하지 않아도 항상 마지막 기본 처리기로 자동 추가됩니다.

이처럼 특정 필터를 사용하기 위해, 필터의 타입 지정으로 적합한 내장 필터 처리기를 생성하거나 사용자 필터 처리기 등록으로 필터 객체를 생성해야 합니다. 기본값으로 설정된 필터 처리기로도 사용이 가능하지만, 사용자가 처리기의 파라미터를 설정하거나 조회 할 수 있는 별도의 함수를 제공합니다. 파라미터의 크기는 필터 처리기의 버전을 찾기 위해 사용되며 크기가 일치 하지 않으면 설정은 무시됩니다. 그리고, 모션 신호의 채널에 맞게 지정된 파라미터의 수까지 설정하고 나머지 채널들은 마지막 파라미터를 그대로 복사 합니다. 즉, 필터는 모션 버퍼의 각 채널별로 다른 설정이 적용 될 수 있는데, 그 수를 1로 설정하면 모든 채널에 동일한 파라미터를 사용합니다.

```
IMFilter scaler = imCreateFilter(IM_FILTER_SCALE); // default x1
IM_FILTER_SCALE_PARAMS scale_params[] = {20}; // x20
imFilterSetParams(scaler, scale_params, sizeof(IM_FILTER_SCALE_PARAMS), 1);
...
imDeleteFilter(scaler);
```

모션 신호의 변환 작업은 단일 필터를 사용하는 경우 보다 여러 필터들이 함께 사용되는 경우가 더 많습니다. 반복적으로 여러 번 필터를 적용하여 모션 신호를 변환하는 것보다 먼저 여러 필터들을 그룹으로 만들어 대상 버퍼에 한번에 적용하는 것이 효율적일 수 있습니다. 모든 필터 객체는 다른 필터들을 자식으로 추가하거나 삭제 할 수 있기 때문에 자주 사용하는 필터 목록을 프리셋(preset)으로 생성하여 재사용하고, 필터를 다른 객체에 설정할 때 기본 타입 (IM_FILTER_DEFAULT)의 필터 그룹에 필요한 필터 처리기들을 등록 하여 사용하는 것이 좋습니다. 자식으로 다른 필터 그룹을 추가 하면 좀 더 복잡한 필터 그래프를 생성 할 수도 있습니다.

```
IMFilter filter = imCreateFilter(IM_FILTER_DEFAULT);
imFilterAppend(filter, scaler);
...
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

3.6.2 필터 빌드

모션 라이브러리의 다른 관리 객체(imContext, imSource, imInput)처럼 필터 객체(imFilter) 또한 모션 버퍼(imBuffer)를 대상으로 필터 변환을 수행 합니다. 필터의 빌드 과정은 필터 처리에 사용될 소스 버퍼와 대상 버퍼를 미리 지정하여 변환에 필요한 정보를 수집하고 활성 처리기 목록 구성과 처리기에 파라미터를 설정합니다. 빌드 이후의 파라미터 설정은 유효하지 않기 때문에 주의가 필요합니다.

다음과 같이 변환 버퍼를 지정하여 사용자가 직접 필터를 빌드하고 필터링 할 수 있습니다. 수동으로 변환할 때 크기와 메모리를 지정할 수 있습니다.

```
imFilterBuild(filter, src_buffer, dst_buffer); // build (src, dst)
...
imFilterProcess(filter); // process
```

또한, 모션 버퍼를 직접 필터링하여 여러가지 다른 모션 효과를 미리 생성할 수 있습니다. 주로 노이즈 필터, 고주파 통과 필터, 저주파 통과 필터 등을 사용하여 범프나 충돌 효과의 수준을 결정하는데 사용됩니다. 이 함수는 필터링 크기 지정이 불가능하고 인코딩된 버퍼 크기만큼 한번에 변환 합니다.

```
imBufferConvert(effect, bump, ref, filter); // build & process(effect, bump)
```

앞의 두 가지 방법처럼 필터 빌드 및 처리를 직접 호출하여 특정 버퍼를 수동으로 필터링할 수 있지만, 대부분 모션 라이브러리에 있는 다른 관리 객체의 런타임 출력에 필터를 적용하여 사용됩니다.

이 경우 각 객체에 설정된(SetFilter) 필터는 런타임 시작 함수(Start/Play)에서 빌더가 사용되며, 이후 런타임 동안 자동으로 처리기가 호출되기 때문에 별도의 필터 API 호출은 불필요합니다. 런타임 필터링의 경우 항상 마스터 버퍼의 크기만큼 샘플 시간 마다 필터링 됩니다.

모션 컨텍스트에서 믹싱된 마스터 버퍼의 모션 신호를 장치로 큐잉 하기전에 전역 필터를 적용할 수 있습니다.

```
imSetFilter(filter);
imStart(); // build (master, device)
...
imStop();
```

다음과 같이 모션 소스에서 재생되는 모션 효과 신호를 믹싱하기 전에 효과 필터를 적용할 수 있습니다.

```
imSourceSetFilter(source, filter);
imSourcePlay(source); // build (source, master)
...
imSourceStop();
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

마지막으로 모션 입력에서 스트리밍하는 모션 신호를 믹싱하기 전에 입력 필터를 적용할 수 있습니다. 주로 실시간으로 시뮬레이션된 모션 신호를 모션 플랫폼의 위치로 변환하는 워시아웃 필터가 많이 사용됩니다.

```
imInputSetFilter(input, filter);
imInputStart(input); // build (input, master)
... // process
imInputStop();
```

3.6.3 필터 처리

사용자가 구성한 필터 그래프는 빌드 과정에서 전위 순회(preorder) 방법으로 필터 처리기 목록을 구성하고, 그 순서대로 필터링을 수행 합니다. 이는 이전 필터의 처리 결과가 다음 필터의 입력으로 사용되기 때문에 추가된 필터의 순서에 따라 다른 결과를 얻을 수 있어 주의가 필요합니다.

예를들어, 워시아웃의 처리 결과를 플랫폼 운동 범위로 스케일하려는 경우, 스케일러(Scaler)를 워시아웃 전에 적용하면 원하는 결과를 얻을 수 없습니다. 또한, 이 스케일러를 속도 제한기(Rate Limiter) 이후에 추가하면 모션 플랫폼의 운동 범위를 벗어나 데미지를 입을 수도 있습니다. 다음은 워시아웃 필터를 사용하는 대표적인 필터 적용 순서를 보이고 있습니다.

(주의, 가속도 및 각속도의 입력 범위는 시뮬레이션 환경에 따라 다를 수 있기 때문에 입력 범위를 양자화 수준에 맞게 스케일해야 오차를 최소화 할 수 있습니다.)

- 가속도 및 각속도의 입력 범위를 스케일하여 인코딩 합니다
(ex : *surge = acceleration * 256*) // input range (-127~127)
- 입력 노이즈를 제거 하기 위해 필터를 추가 합니다.
(ex : *imFilterAppend(filter, noise)*) // smoothing
- 가속도 및 각속도를 모션 플랫폼의 위치나 각도로 변환 합니다.
(ex : *imFilterAppend(filter, washout)*) // washout + tilt(a/g)
- 모션 플랫폼 운동 범위에 맞는 최적의 값으로 스케일러를 추가 합니다.
(ex : *imFilterAppend(filter, scaler)*) // value x 20
- 모션 플랫폼 운동 속도 제한기를 추가 합니다.
(ex : *imFilterAppend(filter, rate_limiter)*) // 15% / sample time

3.6.4 워시아웃 필터

워시아웃 필터은 모션플랫폼의 제한적인 동작 반경으로 가상 환경 내 객체의 움직임을 모사하기 위한 플러그인입니다. 물리엔진에서 발생하는 객체의 동역학적 상태를 모션 플랫폼의 입력으로 변환하는 워시아웃필터 (Washout Filter)로 구성되며, 일반적으로 객체의 병진 가속도(Translational

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

Acceleration)와 회전 각속도(Rotational Velocity)를 필터의 입력으로 병진, 회전 위치(Position)를 출력합니다.

워시아웃필터는 병진 가속도, 회전 각속도의 크기를 완화시키는 역할(Washout)과 시간에 지남에 따라 원점 위치로 복귀시키는 기능(Position Back)을 수행합니다. 또한 전방향, 측 방향 가속도 성분을 각도를 통해 구현합니다.

워시아웃필터는 기능에 따라 High Pass Filter(HPF)와 Low Pass Filter(LPF)로 구성됩니다. HPF 는 Washout 과 Position Back 기능을 담당하고, LPF 는 자세에 따른 중력가속도 구현 역할을 합니다. 더불어 입력과 출력의 상태 단위를 맞추기 위한 수치 적분기가 적용되며 중력가속도를 계산을 위해 Tilt 식이 적용됩니다.

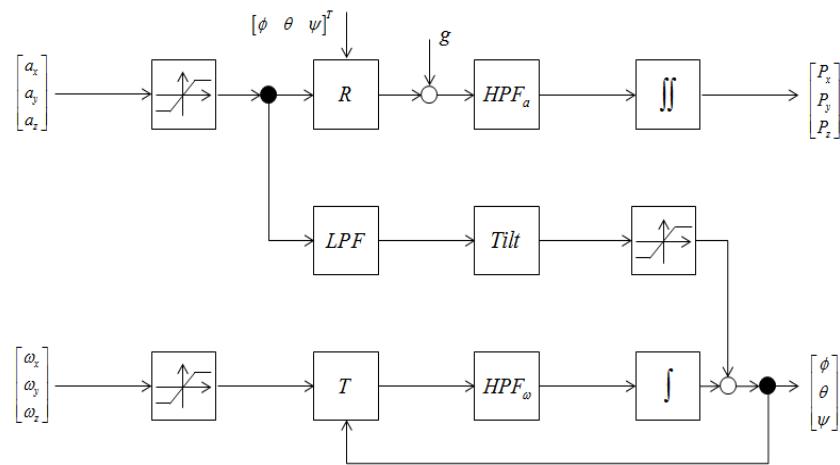


그림 36. 워시아웃 필터 구성도

- High Pass Filter

High Pass Filter 는 입력되는 신호의 고주파(High Frequency) 성분을 통과시키고, 저주파(Low Frequency) 성분을 제한(Cut-off)하는 필터입니다.

일반적으로 물리엔진에서 발생하는 동역학적 생태값은 10Hz 미만의 저주파 특성을 가집니다. 따라서 워시아웃필터로 입력되는 병진 가속도, 회전 각속도의 성분은 HPF 를 거치며 제한됩니다.

HPF 는 입력된 신호를 초기값으로 수렴시키는 방법을 사용하여 고주파 성분을 제한합니다. 이때 HPF 를 구성하는 Parameter 에 따라 수렴하는 모양과 시간이 결정됩니다. 또한 수렴에 걸리는 시간에 의해 지연이 발생합니다.

HPF 를 거쳐 제한된 병진 가속도, 회전 각속도는 수치적분기를 통해 Position 상태로 계산되어 모션플랫폼의 입력으로 적용됩니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

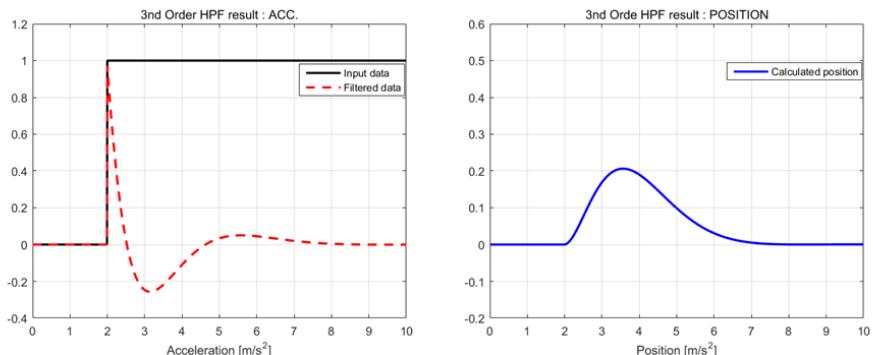


그림 37. High Pass Filter에 의한 입력 변화

- Low Pass Filter

Low Pass Filter는 입력되는 신호의 저주파 성분을 통과시키고, 고주파 성분을 제한(Cut-off)하는 필터입니다.

일반적으로 물리엔진에서 발생하는 동역학적 생태값은 10Hz 미만의 저주파 특성을 가집니다. 따라서 워시아웃필터로 입력되는 병진 가속도 성분은 LPF를 거치며 값이 유지됩니다.

LPF는 입력된 신호를 입력값으로 수렴시키는 방법을 사용하여 저주파 성분을 유지합니다. 이때 LPF를 구성하는 Parameter에 따라 수렴하는 모양과 시간이 결정됩니다. 또한 수렴에 걸리는 시간에 의해 지연이 발생합니다.

LPF를 거쳐 유지된 병진 가속도는 Tilt 식을 통해 Position 상태로 계산되어 모션플랫폼의 입력으로 적용됩니다.

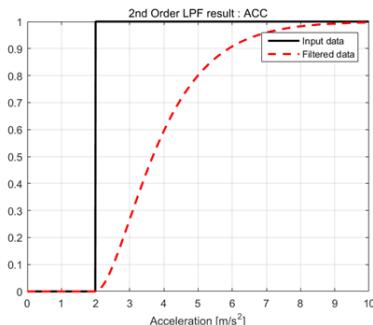


그림 38. LowPass Filter에 의한 입력 변화와 Tilt 적용 개념도

3.6.5 사용자 필터

모션 라이브러리는 디지털 필터들 중 제한된 필터 처리기 만을 지원하고 있고, 워시아웃 필터의 경우에도 다양한 알고리즘과 구성방법이 존재 합니다. 콜백 함수 등록으로 생성된 사용자 정의 필터는 내장 필터를 개선, 재구성, 교체, 추가 할 수 있는 필터의 확장 기능을 제공합니다. 앞에서 설명한

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

필터의 빌드와 처리 함수가 사용될 때마다 여러 번 필터 처리기가 호출되기 때문에, 아래와 같이 필터의 빌드와 변환 로직을 콜백 함수에 포함하여 사용자 필터 처리기를 정의 해야 합니다.

```
float custom_processor(void* context, void* data, int size,
    IM_FORMAT* src_format, const IM_FORMAT* dst_format)
{
    // 1. BUILD logic
    // 2. CONVERT logic
    return convert_rate;
}
```

필터 처리기는 필터 빌드 과정에서 먼저 한번 호출된 다음, 런타임 중에 여러 번 호출될 수 있습니다. 사용자의 필터 파라미터, 변환 데이터와 크기, 소스 및 대상 버퍼의 포맷이 함수의 인자로 전달 됩니다. 변환 데이터와 크기 인자가 유효하지 않을 때만 파라미터 초기화와 같은 빌드 로직이 처리 되어야 합니다. 아래 예와 같이 포맷 변경이 필요한 경우 소스 포맷에 그 결과를 갱신하고 변환율을 반환해 주어야 합니다. 하지만, 사용자 정의 필터의 경우 포맷 변경 없이 입력 파형을 변형하는데 주로 사용되므로, 대부분 포맷 변경 없이 1을 반환합니다.

```
// 1. BUILD : For the filter build process, first specify the supported conversion results in the source format.
if(src_format->nChannels != 6 || dst_format->nChannels != 3)
    return 0; // If the conversion format is not supported by the build process of the filter ...

float cvt_ratio = dst_format->nChannels / (float)src_format->nChannels;
if(data == 0 && size == 0) {
    // Where it initializes the custom context data and informs the support conversion format.
    src_format->nChannels = dst_format->nChannels;
    return cvt_ratio; // Notify that it is a supported conversion format for the build process of the filter
}
```

실제 필터 변환 과정에서 주어진 결과 크기를 얻기 까지 입력 데이터를 처리해야 하는데, 처리 결과가 저장되는 메모리가 오염되지 않도록 주의가 필요합니다. 예를 들어, 채널 교환과 같이 채널 순서를 변경 하는 경우 이전 채널 변환의 오염된 결과를 다음 채널 변환의 입력으로 사용하는 오류가 발생 할 수 있습니다. 또한, 변환율이 1 보다 큰 경우 변환 결과는 항상 더 큰 메모리 공간을 필요로 하므로 동일한 문제를 유발할 수 있습니다.

이 문제의 원인은 동일한 메모리를 읽기와 쓰기에 사용하는데 있습니다. 임시 저장 메모리를 사용하거나, 소스와 대상 위치를 각 크기만큼 이동 후 역방향으로 변환하여 이 문제를 해결할 수 있습니다. 불필요한 연산 오버헤드가 없애고 입력 신호의 손실이 없도록, 상황에 맞게 사용자가 직접 메모리를 관리하도록 권장하고 있습니다.

```
// 2. CONVERT : It processes the actual user filter logic here.
int mask = *(int*)context; // ex) IM_BIT_MASK_DEFAULT
const short* src = (const short*)data;
short* dst = (short*)data; // Create temporary memory as needed
int samples = size/(sizeof(short)*6);
while(samples--) {
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

```

for(int i=0; i<6; i++) {
    if(mask & (1<<i)) {
        (*dst) = (*src);
        dst++;
    }
    src++;
}
src_format->nChannels = dst_format->nChannels;
return cvt_ratio;

```

내장 필터 처리기의 사용자 재구성 기능은 위시아웃과 같은 복합 필터인 경우 유용할 수 있습니다. 복합 필터의 특정 처리기를 교체하거나 세부 파라미터를 설정하기 원한다면, 다음과 같이 사용자 재구성 필터를 고려해 보는 것이 좋습니다.

```

/* simulate cues of initial acceleration (translational and rotational) */
// 3rd HPF (heave acceleration), 2nd HPF (roll/pitch velocity)
IMFilter highpass_filter = imCreateFilter(IM_FILTER_HIGHPASS);
// 2nd integral (heave position), 1st integral (roll/pitch angle)
IMFilter integrator = imCreateFilter(IM_FILTER_INTEGRAL);
/* simulate sustaining accelerations (G-Force) */
// 2nd LPF (surge/sway acceleration)
IMFilter lowpass_filter = imCreateFilter(IM_FILTER_LOWPASS);
// tilt-coordination (add special force to pitch/roll)
IMFilter tilt_coordinator = imCreateFilter(IM_FILTER_TILT);
/* build custom washout filter preset */
IMFilter custom_classical_washout = imCreateFilter();
imFilterAppend(custom_classical_washout, highpass_filter);
imFilterAppend(custom_classical_washout, integrator);
imFilterAppend(custom_classical_washout, lowpass_filter);
imFilterAppend(custom_classical_washout, tilt_coordinator);
...
IMFilter filter = imCreateFilter();
imFilterAppend(filter, noise_filter);
imFilterAppend(filter, custom_classical_washout);
...
imInputSetFilter(input, filter);
imInputStart(input);

```

3.6.6 주요 API

입력된 모션 데이터 신호 변환의 책임이 있는 Filter 는 다음과 같은 주요 API 기능을 갖습니다. (상세한 설명은 API Reference 를 참고하세요.)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

표 14. Filter 주요 API

API	설명
imCreateFilter(type, callback, data) imDeleteFilter(filter)	타입에 맞는 필터 처리기로 모션 필터 객체를 생성(제거)합니다. 콜백 등록으로 사용자 정의 필터를 생성 할 수 있습니다. (기본 파라미터 설정이 적용됩니다)
imFilterSetParams/GetParams (filter, data, size, count)	필터 처리기에 파라미터를 설정(조회)합니다. 이 함수로 설정(조회)된 파라미터의 수를 반환 합니다.
imFilterAppend(filter, child) imFilterRemove(filter, child)	필터 객체를 필터 그룹 객체에 추가(삭제)합니다. 자주 사용되는 필터 그룹을 미리 생성하여 쉽게 사용할 수 있습니다. (등록 순서에 따라 처리 됩니다)
imFilterBuild(filter, src_buffer, dst_buffer)	소스 버퍼와 대상 버퍼를 지정하여 수동 변환에 필요한 정보와 처리기 목록을 구성 합니다. 변환율을 반환 하고, 값이 0인 경우 변환이 불가능 합니다. (※ 주의, 필터가 컨텍스트, 소스 및 입력에 설정 되어 있으면(SetFilter), 이 빌드 함수 와 아래의 처리 함수 사용은 불필요합니다.)
imFilterProcess(filter, data, size)	빌드된 변환 정보를 이용 소스 버퍼에서 크기만큼 변환하고 (size가 0인 경우 전체 크기), 그 결과를 메모리(data)에 저장 합니다. 변환 완료된 크기를 반환 합니다. (※ 주의, size는 요구하는 변환 결과의 크기이며, data가 NULL인 경우 대상 버퍼에 큐잉됩니다.)

3.7 샘플 프로젝트

샘플 프로젝트는 IMotion 및 InnoML 라이브러리의 사용 예제입니다.

장치 연결 및 진단, 모션 위치 직접 제어, 자동 제어, 모션 파일 재생에 대한 예제를 설명합니다.

예제코드는 C++ 기반으로 작성되었고 C 및 C# 코드가 함께 제공됩니다.

3.7.1 프로젝트 열기

SDK 사용법을 쉽게 설명하기 위한 샘플 프로젝트입니다. 3 가지 샘플의 솔루션 파일은 각 라이브러리의 기능을 시험하는 프로젝트와 소스 파일(c, cpp, cs)로 구성됩니다.

- Visual Studio 개발 환경

Visual Studio 2010 이상의 C++ 및 C# 개발 환경을 지원합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

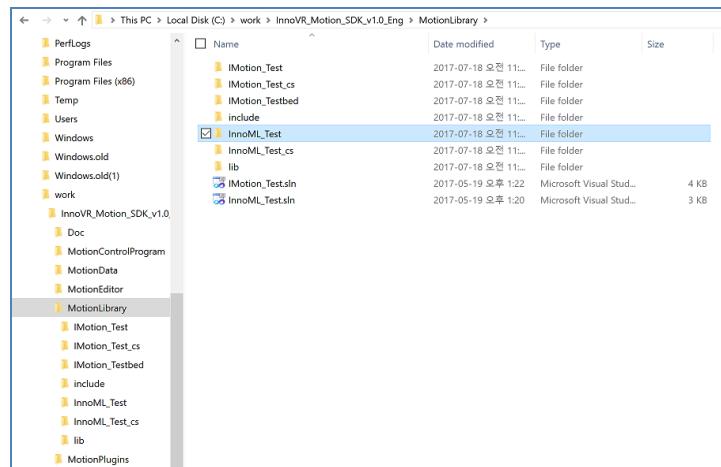


그림 39. 모션 샘플 프로젝트

샘플 프로젝트 폴더 내의 솔루션 파일을 비주얼 스튜디오 2010에서 오픈하여 빌드한 후 실행해 봅니다. 샘플 프로젝트는 비주얼 스튜디오 2010 이상 닷넷 프레임워크 2.0 이상, 32비트 및 64비트에서 빌드됩니다.

SDK의 IMotion 및 InnoML 솔루션 내에 포함된 샘플 프로젝트와 예제 파일이 포함되어 있습니다. 솔루션 파일을 열고 다음 절차로 활성 프로젝트를 선택할 수 있습니다.

프로젝트 선택>>마우스 우측 클릭>>시작 프로젝트로 설정 선택

활성 프로젝트에 있는 샘플 예제 파일(main*.cpp)은 다음 절차로 파일 선택 후 튜토리얼을 선택적으로 실행해 볼 수 있습니다.

선택(예/아니오)에 따라 해당 샘플이 활성(C++) 또는 비활성(C++) 됩니다.

샘플 파일 선택>>마우스 우측 클릭 후 속성(Alt+F7)>>빌드에서 제외>>예/아니오 선택

- IMotion 샘플 프로젝트

“IMotion_Test.sln” 솔루션 파일은 3개의 샘플 프로젝트를 포함하고 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

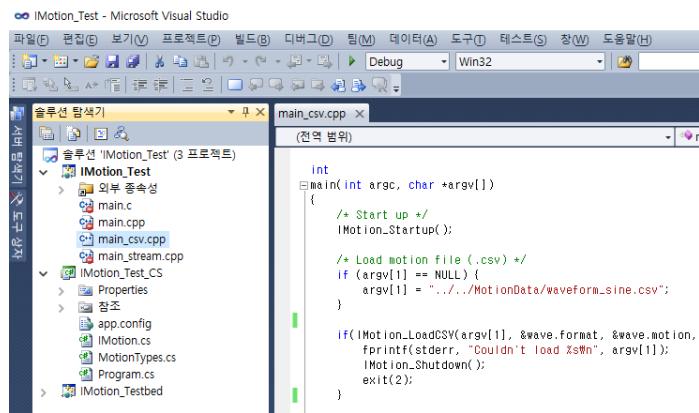


그림 40. IMotion 샘플 파일들

기본 프로젝트인 IMotion_Test에는 4개의 C/C++ 샘플 예제 파일이 포함되어 있고, 앞서 설명한 방법으로 선택적으로 실행해 볼 수 있습니다.

표 15. IMotion 샘플 예제 목록

File	설명
main.c	IMotion 라이브러리 기본 사용 방법을 설명 합니다. (3.6.2 상세 설명 참고)
main.cpp	main.c의 C++ 용 샘플 예제입니다.
main_csv.cpp	모션 효과 파일의 로딩과 실행 방법을 설명 합니다.
main_stream.cpp	모션 데이터의 스트리밍 방식 제어를 설명 합니다.

IMotion_Test_CS 프로젝트 파일은 기본 예제(main.c)의 C# 샘플 예제입니다. C# Wrapper 인 IMotion.cs 파일을 통해 라이브러리를 동적으로 연결하여 사용합니다. (유니티 개발에 참고 할 수 있습니다)

마지막 IMotion_Testbed 프로젝트는 IMotion 라이브러리의 기능을 검증 하는 MFC 기반 GUI 프로젝트입니다. 모든 예제 소스 및 리소스가 포함되어 있습니다.

화면 좌측에서는 모션 장치의 연결, 상태 제어, 장애 및 상태 진단 등의 기능을 확인 할 수 있습니다. 우측에서는 모션 위치 직접 정밀 제어, 축별 사인파 자동 제어, 모션 효과 파일 자동 제어, 재생 시간 동기 등의 주요 기능을 시험할 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

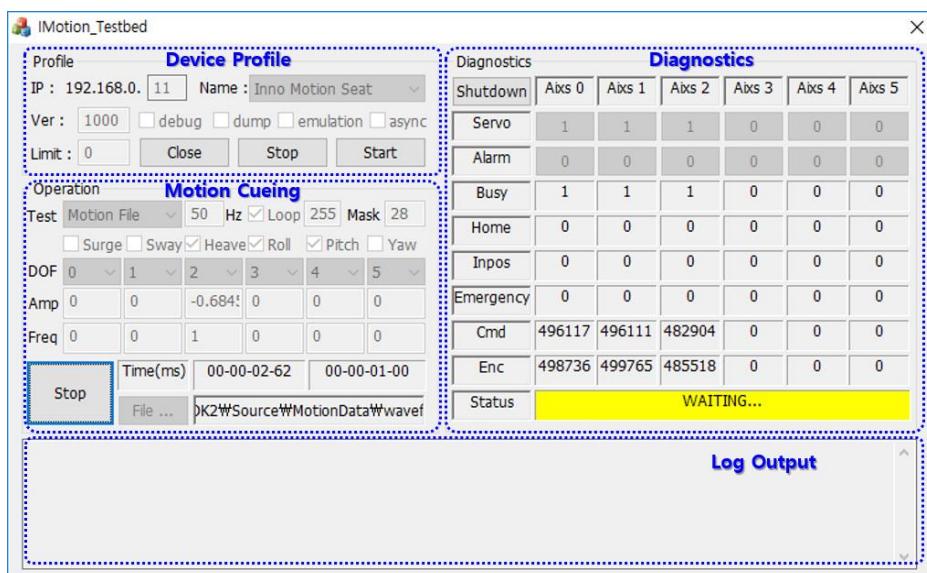


그림 41. IMotion_Testbed 화면 구성

- InnoML 샘플 프로젝트 (InnoML_Test.sln)

“InnoML_Test.sln” 솔루션 파일은 2 개의 샘플 프로젝트를 포함하고 있습니다.

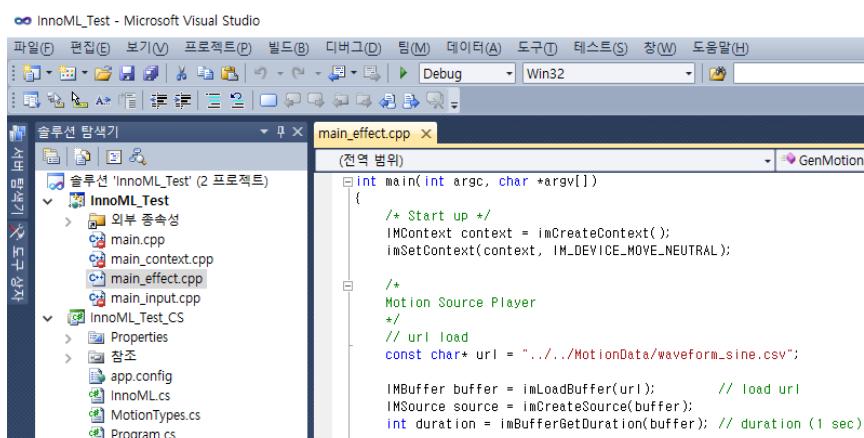


그림 42. InnoML 샘플 파일들

기본 프로젝트인 InnoML_Test에는 4 개의 C/C++ 샘플 예제 파일이 포함되어 있고, 앞서 설명한 방법으로 선택적으로 실행해 볼 수 있습니다.

표 16. InnoML 샘플 예제 목록

File	설명
main.cpp	InnoML 라이브러리 기본 사용 방법을 설명 합니다. (3.6.3 상세 설명 참고)
main_context.cpp	멀티 모션 컨텍스트 관리 샘플 예제입니다.
main_effect.cpp	모션 파일의 로딩과 소스 재생 방법을 설명 합니다.
main_input.cpp	모션 입력 스트리밍 및 샘플링 방법을 설명 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

InnoML_Test_CS 프로젝트 파일은 기본 예제(main.cpp)의 C# 샘플 예제입니다. C# Wrapper 인 InnoML.cs 파일을 통해 라이브러리를 동적으로 연결하여 사용합니다. (유니티 개발에 참고 할 수 있습니다)

3.7.2 IMotion 시작하기

IMotion_Test 샘플 프로젝트의 “main.c” 예제 소스를 통해, IMotion 라이브러리의 기본 동작을 알아봅니다.

모션 장치의 직접 위치 제어와 마스터 소스를 통한 자동 위치 제어 방법을 예제와 함께 설명합니다.

예제코드는 C 언어로 작성되었고 C++ 및 C# 코드가 함께 제공됩니다.

- IMotion 시동

IMotion 라이브러리의 시동과 종료는 프로그램의 처음과 끝에 사용할 수 있습니다. 지원 모션 장치 목록 생성 등의 초기화 작업을 처리하여 모션 라이브러리를 시작합니다.

```
IMotion_Startup(0, 0); // IMotion_Shutdown
...

```

- IMotion 디바이스 생성하기

IMotion 라이브러리의 생성을 위해 장치 연결을 위한 장치 종류 선택과 ID 를 설정해야 합니다. (ID 는 중복사용이 불가능합니다.)

모션 장치의 ID 는 제어 PC 와 연결되는 모션 플랫폼 장치의 이더넷 IP 의 마지막 값으로 지정해야 합니다. (내부적으로 "192.168.0.xxx" 로 설정, 기본값 11)

지원 장치 이름은 IMotion_GetDeviceDescription 함수를 통해 조회가 가능 하고, 기본 장치는 모션 시트로 정의 되어 있습니다.

모션 장치 연결 없이 라이브러리를 에뮬레이션 하거나 모션 장치 버전 등의 라이브러리 실행 모드 및 옵션을 지정합니다.

모션 장치의 초기(중립) 위치로 이동하고 IMotion 인터페이스를 통한 제어를 시작 합니다. (IMotion_Start 는 초기 위치 재설정을 위해 중복 사용 가능합니다.)

```
int devid = IMotion_Create(0, 0);
IMotion_Start(devid, 0, 0);
...

```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

- IMotion 디바이스 수동 정밀 제어하기

모션 장치 별로 지원 가능한 모션 자유도(DOF) 와 운동 범위가 다를 수 있기 때문에, 정규화된 모션 위치 데이터 정의가 필요합니다.

DOF(Degrees Of Freedom)는 모션 플랫폼 xyz 각 축의 직선 운동(surge, sway, heave)과 회전 운동(roll, pitch, yaw)을 의미합니다.

IMotion 에서는 장치 호환성을 위해 DOF 순서로 모션 데이터의 각 채널을 선택적으로 매핑하고, 운동 범위를 16 비트 정수형(-32768 ~ 32767)에 양자화 하여 제어합니다.

3DOF 모션 시트의 경우 정규화된 모션 샘플(각 채널의 모션 데이터)은 모션 장치의 자세(Heave(-30~30), Roll(-10~10), Pitch(-10~10))에 연동됩니다.

모션 샘플들을 일정한 시간 간격으로 연속적으로 전달하여, 모션 디바이스의 위치를 직접 변경시켜 정밀 제어할 수 있습니다. (HW 성능 이상의 위치 제어는 오동작의 원인이 될 수 있습니다.)

```
int devid = IMotion_Create(0, 0); // IMotion_Destroy
IMotion_Start(devid, 0, 0); // IMotion_Stop
...
#define SAMPLE_CHANNELS 3
#define SAMPLE_RATE      50
#define MAX_MOTION_VAL(bits) ((1 << (bits - 1)) - 1)
short buf[SAMPLE_RATE][SAMPLE_CHANNELS];
for(int i=0; i<SAMPLE_RATE; i++) {
    double wave = sin(2 * M_PI * (float)i/SAMPLE_RATE);
    buf[i][0] = (short)(wave*MAX_MOTION_VAL(16));
    IMotion_SendStream(devid, (uint8*)&buf[i], sizeof(short)*SAMPLE_CHANNELS, 1);
}
```

- IMotion 소스 생성하기

미리 정의된 모션 포맷에 맞는 모션 버퍼(모션 샘플 들)를 모션 디바이스에 전달하여 자동 제어할 수 있습니다.

모션의 자동 제어를 위해 모션 샘플 스트림을 재생 하는 방법을 기술하는 모션 포맷 정의로 모션 소스를 생성해야 합니다. (디바이스 당 1 개의 마스터 소스 생성 가능)

일정한 제어 속도를 정의 하는 샘플 레이트는 모션 장치의 성능 한계에 따라 정의 되어야 하고, 기본 값은 초당 50 Samples 이 사용 됩니다. (HW 성능 이상의 설정은 오동작의 원인이 될 수 있습니다.)

채널 수는 모션 장치의 특성에 따라 지원 가능한 자유도 (DOF) 에 따라 정의되며, 기본은 3 채널이 사용됩니다. (지원 채널 이상의 값은 무시됩니다.)

```
{
...
}
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

```

IM_FORMAT format;
memset(&format, 0, sizeof(IM_FORMAT));
format.nSampleRate = SAMPLE_RATE;
format.nChannels = SAMPLE_CHANNELS;
...
int source = IMotion_CreateSource(devid, &format, NULL, NULL);
...
}

```

- IMotion 디바이스 자동 제어하기

모션 소스에서 자동으로 제어할 모션 데이터 스트림을 버퍼 형태로 지정합니다.

모션 버퍼의 데이터 배열과 바이트 크기를 지정하면, 모션 소스는 내부적으로 short 형 데이터 포맷으로 변환하여 사용합니다.

모션 버퍼는 버퍼 재생 정보(반복 재생 등)와 재생 상태 콜백 함수에 전달할 사용자 데이터를 지정할 수 있습니다.

준비된 모션 버퍼를 모션 소스의 작업 큐(Queue)에 추가 하면, 내부 스레드는 자동으로 모션 제어를 수행합니다.

모션 스트림을 고정 크기의 여러 개의 버퍼로 나누어 작업 큐에 추가 하는 스트리밍 방식을 권장 합니다. (초기 지연을 고려하여 버퍼의 크기를 설정해야 합니다)

```

{
...
IM_BUFFER buffer;
memset(&buffer, 0, sizeof(IM_BUFFER));
buffer.nMotionBytes = SAMPLE_RATE*SAMPLE_CHANNELS*sizeof(int16);
buffer.pMotionData = (uint8*)buf;
IMotionSource_SubmitBuffer(source, &buffer);
...
}

```

- IMotion 소스 재생 제어하기

모션 소스의 내부 스레드의 시작과 종료 명령에 따라 자동으로 모션 제어를 수행합니다. (수동 제어는 작업 큐가 없을때 사용 권장)

비동기 작업 큐의 수 및 콜백 함수의 상태 조회를 통해 모션 버퍼 제어 상태를 조회합니다.

```

{
...
IMotionSource_Start(source, 0);
while(IMotionSource_GetQueuedBufferCount(source) > 0)
    Sleep(10);
IMotionSource_Stop(source, 0);

```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

```
{
    ...
}
```

- IMotion 라이브러리 종료

생성된 모션 소스를 종료 합니다. 모션 소스의 정지를 내부적으로 수행합니다.

모션 장치의 연결을 해제 합니다. IMotion_Destroy 함수에서 IM_MOTOR_POWER_OFF 인자를 명시적으로 사용하면 모터를 끌 수 있습니다.

모든 장치 연결이 해제 되어 더이상 모션 제어가 필요 없을때 라이브러리를 종료하면 라이브러리의 모든 메모리를 해제합니다.

```
{
    ...
    IMotion_DestroySource(devid, source);
    IMotion_Destroy(devid, 0);
    IMotion_Shutdown();
}
```

3.7.3 InnoML 시작하기

InnoML_Test 샘플 프로젝트의 “main.cpp” 예제 소스를 통해, InnoML 라이브러리의 기본 동작을 알아봅니다.

모션 장치 컨텍스트, 모션 버퍼, 모션 소스를 통한 모션 데이터 재생 방법을 예제와 함께 설명합니다.

예제코드는 C/C++ 언어로 작성되었고 C# 코드가 함께 제공됩니다.

- InnoML 디바이스 연결 하기 (모션 컨텍스트)

모션 장치로 샘플 스트림을 전송하기 위한 마스터 버퍼와 실행 모드를 지정하여 컨텍스트를 생성 합니다. (0 이면 기본 포맷의 버퍼 사용)

마스터 버퍼의 포맷은 실행 중인 모션 버퍼들을 혼합하는 방법과 모션 장치 제어의 성능을 결정하므로, 최적의 값으로 설정 되어야 합니다.

모션 장치의 ID 는 제어 PC 와 연결되는 모션 플랫폼 장치의 이더넷 통신 IP 의 마지막 값으로 지정해야합니다. (내부적으로 "192.168.0.xxx" 로 설정, 기본값 11이며 중복 사용이 불가합니다.)

장치 라이브러리 생성을 위해 모션 장치의 이름과 버전, 디버깅 옵션을 설정 할 수 있습니다. (기본 값은 모션 시트 최신버전입니다.)

장치 버전에 따라 제어 파라미터가 변경되어 HW 오동작의 원인이 될 수 있습니다. (v0.7 과 v1.0 이 있으며, 양산용 모션 플랫폼(v1.0) 부터 후면부 명판 모델 유무로 버전 확인이 가능 합니다.)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

생성된 장치를 활성 컨텍스로 지정하고 장치를 실행 준비 위치로 이동 시켜서, 모션 재생을 준비해야 합니다.

모션 디바이스의 진단 상태 코드를 얻고, 모션 장치 연결이 실패한 경우라도 에뮬레이션 모드로 자동으로 전환되어 실행됩니다.

```
IM_DEVICE_DESC desc;
memset(&desc, 0, sizeof(IM_DEVICE_DESC));
desc.nOptions |= IM_CFG_DEBUG_MODE;

IMBuffer master_buffer = imCreateBuffer(SAMPLE_RATE, 0, 0, SAMPLE_COUNT, 2);
IMContext context = imCreateContext(master_buffer, 0, &desc);
imSetContext(context);
imStart();

int error = imGetDeviceDiagnostic(0, 0, 0);
...
```

- InnoML 데이터 생성하기 (모션 베퍼)

인간이 느낄 수 있는 연속적인 모션 패형과 동일한 운동감을 모션 장치를 통해 인간에게 전달하는 것은 매우 중요합니다.

1 초 동안의 연속적인 모션 신호로 부터 얼마나 많은 이산적인 모션 신호를 생성할 것인가를 정의 하는 것을 샘플 레이트(Sample Rate)라고 합니다.

모션 베퍼는 샘플 수와 샘플링 레이트 등의 포맷 정의로 생성되고, 이산적인 신호인 샘플들로 구성된 데이터를 저장합니다. (샘플 데이터는 16 비트 정수형 포맷에 양자화 됩니다)

```
short buf[SAMPLE_COUNT][IM_FORMAT_CHANNELS_DEFAULT] = {0,};
for(i=0; i<SAMPLE_COUNT; i++)
    buf[i][0] = (short)(sin(2 * IM_PI * (float)i/SAMPLE_COUNT)*MOTION_MAX_16);
...
```

- InnoML 컨텍스트 수동 제어하기

모션 스트림을 순차적으로 모션 장치 수동 제어하기 위해 마스터 베퍼에 샘플을 직접 추가하여 위치를 제어할 수 있습니다.

다수의 모션 스트리밍을 위해 모션 입력 객체(imInput)를 사용해야 합니다.

```
for(i=0; i<SAMPLE_COUNT; i++) {
    buf[i][0] = (short)(sin(2 * IM_PI * (float)i/SAMPLE_COUNT)*MOTION_MAX_16);
    imBufferEnqueue(master_buffer, buf[i], sizeof(int16)*SAMPLE_CHANNELS);
    Sleep(1000/SAMPLE_RATE);
}
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

- InnoML 컨텍스트 자동 제어하기

모션 스트림을 일괄적으로 전달하여 모션 장치 자동 제어를 위해 모션 소스 객체가 필요합니다.

모션 스트림과 모션 제어 정보를 갖고 있는 모션 버퍼를 지정하여 모션 소스가 생성 되어야 합니다.

모션 소스의 재생으로 모션 버퍼의 샘플들이 순차적으로 읽혀진 후, 다른 모션 재생 샘플들과 혼합(mix) 되어 모션 장치에 전달됩니다.

모션 버퍼의 마지막 샘플 실행이 완료되면 해당 모션 소스는 컨텍스트의 재생 중인 소스 목록에서 제거됩니다.

```
IMBuffer buffer = imCreateBuffer(0,0,0,SAMPLE_COUNT,1);
imBufferEnqueue(buffer, buf, sizeof(buf));
IMSource source = imCreateSource(buffer);
imSourcePlay(source,0,0,0);
while(imGetPlayingSourceCount())
    Sleep(10);
```

- InnoML 컨텍스트 종료

어플리케이션에서 제거 하지 않고 있는 모션 소스와 버퍼의 사용을 정지하고 모두 제거합니다.

사용이 끝난 모션 객체(특히, 모션 버퍼)는 바로 명시적으로 제거하여 메모리 낭비를 줄이는 것이 좋습니다.

제거되지 않은 모든 소스와 모션 입력은 컨텍스트가 제거될 때, 그리고 모션 버퍼는 라이브러리 종료시 자동으로 해제됩니다.

모션 장치의 위치를 최저 위치로 이동 후 연결을 해제 합니다. IM_MOTOR_POWER_OFF 인자를 명시적으로 사용하면 모터를 끌 수 있습니다.

```
...
imDeleteSource(source);
imDeleteBuffer(buffer);
imStop();
imSetContext(NULL);
imDestroyContext(context, 0);
```

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4 모션 플러그인

2 장에서 알아본 모션 시스템은 VR 모션 시뮬레이션 시스템에서 하나의 3rd party 미들웨어 입니다. 최근 VR 게임 엔진으로 많이 사용하는 Unity3D 나 언리얼 엔진에서 모션 시스템을 적용하기 위해 VR 모션 SDK 에서 제공 되는 C# Wrapper 또는 Native DLL 을 임포트하여 사용할 수 있다. 그러나 VR 게임 엔진이 제공하는 컴포넌트 시스템은 VR 콘텐츠 개발의 생산성 향상에 중요한 역할을 담당하는 것이 사실이기 때문에, 타 플랫폼을 위한 모션 시스템의 컴포넌트 제공이 필요합니다.

4 장에서는 Unity3D, Unreal Engine 4 용 모션 컴포넌트의 구성 및 사용방법을 알아봅니다.

모션 플러그인은 모션 라이브러리의 4 개 관리 객체(Context, Source, Input, Filter)를 컴포넌트로 정의하고, 컴포넌트 내부에서 모션 버퍼를 관리합니다.

기본적으로 모션 버퍼의 사용은 숨겨져 있으나, 마스터 버퍼의 포맷의 수정은 공개 되어 있습니다. 기본값의 수정을 원하는 경우 모션 버퍼에 대한 충분한 이해 후에 사용하기를 권장합니다.

모션 플러그인은 Motion Context, Motion Source, Motion Input, Motion Filter 4 개의 컴포넌트를 조합하여 모션을 운용할 수 있습니다. 각 컴포넌트의 목적은 아래와 같습니다.

- Motion Context : 모션 장치에 대한 연결 및 제어를 담당
- Motion Source : 모션 데이터의 재생과 그 제어를 담당
- Motion Input : 직접적인 모션 데이터 값의 스트리밍과 제어를 담당
- Motion Filter : 모션 필터의 생성과 제어를 담당

기본적으로 한 개의 Motion Context 컴포넌트에는 다수의 Motion Source, Motion Input 컴포넌트와 관계를 갖게됩니다. 이를 통해 자연스러운 믹싱 기능을 지원하며, 이 컴포넌트들이 동시에 동작하게 될 때 각자가 소유하는 모션 버퍼가 Motion Context 의 마스터 버퍼에 믹싱되어 동작하게 됩니다.

※ 주의사항

모션 플러그인은 SDK 기본 구성이 아니며, 패키지 구성에 따라 기능이 제한될 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1 유니티 플러그인

유니티 플러그인은 유저의 유니티 프로젝트에 Import 하기 위한 플러그인 패키지와 샘플 프로젝트를 제공합니다. 플러그인 패키지는 플러그인 사용을 위한 필수요소만 포함합니다. 샘플 프로젝트는 제공되는 컴포넌트가 어떻게 사용되는지에 대해 볼 수 있는 예제 씬과 C# 샘플 스크립트를 제공합니다.

※ 주의 : 유니티 플러그인은 Unity 5.4.0 버전 이상 사용을 요구합니다.

4.1.1 Motion Context Component

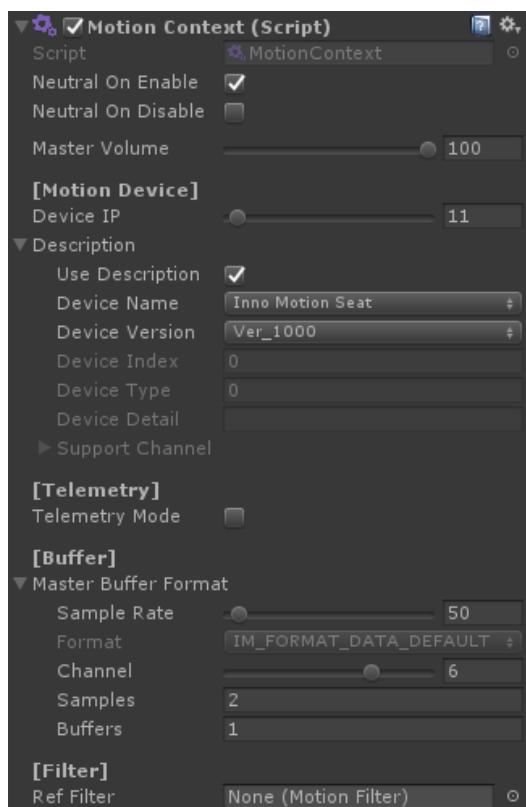


그림 43. Motion Context Component

Motion Context 컴포넌트는 모션 라이브러리의 모션 컨텍스트(imContext) 기능을 사용합니다. 모션 장치에 대한 연결, 상태 등을 간편하게 제어하기 위해 구성되어 있습니다.

유저는 모션과의 연동을 위해 적어도 1 개 Motion Context 컴포넌트를 사용해야 합니다. Motion Context 컴포넌트는 Motion Input, Motion Source 컴포넌트가 참조하게 됩니다. 이를 위해 이들이 같은 GameObject 내에 존재하거나, Motion Input, Motion Source 컴포넌트의 PlayContext에 동작하길 원하는 모션의 Motion Context 컴포넌트를 직접 할당하여 사용합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1.1.1 컴포넌트 설정 값

표 17. Motion Context 컴포넌트 설정 값

Value	Function
Neutral On Enable	Enable 시 모션 장비의 중립 위치 이동 명령(Neutral)의 수행 여부
Neutral On Disable	Disable 시 모션 장비의 중립 위치 이동 명령(Neutral)의 수행 여부
Master Volume	마스터 볼륨의 크기를 설정합니다. 0~100 범위의 값을 설정할 수 있습니다.
Device IP	모션 장비 이더넷 IP의 4 th Octet값 (예: 192.168.0.11 의 경우 11)
Use Description	사용자 정의 Description 옵션의 사용 여부, 이 체크박스에 체크하는 경우 사용자는 Description의 Device Name과 Device Version을 설정할 수 있습니다.
Description	IM_DEVICE_DESC 구조체의 nOptions 필드를 제외한 나머지 정보를 확인할 수 있습니다. 정보는 Context의 첫 생성시 갱신됩니다. * 3.3.1 모션 장치 연결 Description 구조체 (IM_DEVICE_DESC) 참조
Telemetry Mode	텔레메트리 모드의 사용 여부 - Network Type : Motion Player와의 통신 방식 (UDP/TCP/IPC) - Destination IP/Port : Motion Player PC의 IP와 Port (Default : 127.0.0.1 / 49288) - Motion Clip List : Motion Source에서 사용되는 모션 클립의 리스트 (최대 10개)
Master Buffer Format	마스터 버퍼의 포맷 정보를 설정합니다. * 3.2.1 버퍼 생성의 Format 구조체 (IM_FORMAT) 참조
Ref Filter	Motion Filter 컴포넌트를 할당합니다.

4.1.1.2 주요 프로퍼티

표 18. Motion Context 컴포넌트 프로퍼티

Property	Description
float DiagnosticTime	모션 장치의 진단주기를 설정합니다. 초 단위 1~10 범위의 float 값으로 설정할 수 있습니다. 해당 진단주기에 맞추어 진단정보가 갱신됩니다. 만약 긴급상황(Alarm, Emergency)이 발생하는 경우 해당 Motion Context 컴포넌트는 비활성화 됩니다.
MIDiagnostic Diagnostic	모션 장치의 진단정보를 확인합니다. 진단주기에 따라 내부적으로 정보가 갱신됩니다. * 3.3.3 모션 장치 제어의 Diagnostic 구조체 (IM_DIAGNOSTIC_AXIS_INFO) 참조
bool IsBusy	모션 장치의 Settle, Neutral 명령 시 모션 동작 유무를 확인합니다. - True : 모션이 이동 중 / False : 모션이 정지 중
bool IsConnected	모션 장치의 진단정보를 통해 모션과의 접속 유무를 확인합니다.

4.1.1.3 주요 함수

표 19. Motion Context 컴포넌트 함수

Method	Description
void SetMotionStart()	모션 장치에 중립 위치 이동 명령(Neutral)을 전달합니다. (비동기로 동작하며, 씬 전환 시 활용할 수 있습니다.)
void SetMotionStop()	모션 장치에 초기 위치 이동 명령(Settle)을 전달합니다. (비동기로 동작하며, 씬 전환 시 활용할 수 있습니다.)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1.2 Motion Source Component

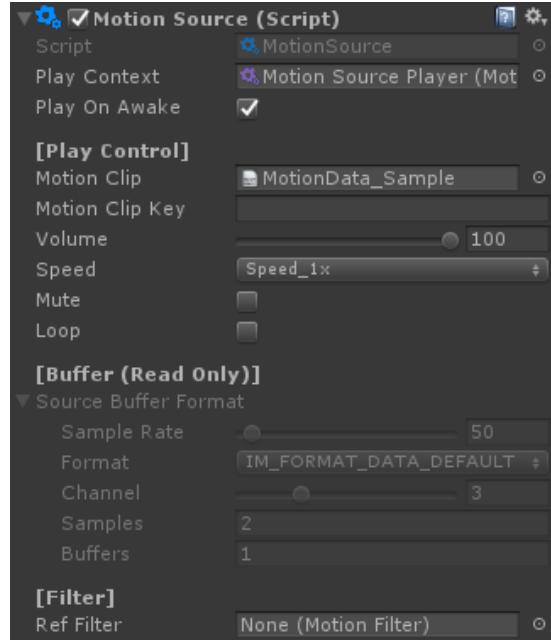


그림 44. Motion Source Component

Motion Source 컴포넌트는 모션 라이브러리의 모션 소스(imSource) 기능을 사용합니다. 모션 소스에 대한 생성, 설정, 모션 데이터의 재생 등을 간편하게 제어하기 위해 구성되어 있습니다.

모션 데이터는 TextAsset(임포트된 텍스트 파일을 위한 형식)이 지원하는 “.bytes” 확장자를 사용합니다. “.csv” 확장자를 사용하는 모션 데이터를 사용하기 위해서는 “.bytes”로 수정 후 사용이 필요합니다.

일반적인 사용 방법은 다음과 같습니다.

1. 유니티 Asset 폴더 내 “.bytes” 모션 데이터 파일을 Motion Source 컴포넌트의 Motion Clip에 할당
2. Play On Awake 상태에 체크하여 처음 실행 시 모션 데이터를 플레이
3. 런타임 중 Motion Source의 Play 함수를 이용해 모션 데이터를 플레이

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1.2.1 컴포넌트 설정 값

표 20. Motion Source 컴포넌트 설정 값

Property	Function
Play Context	Motion Context 컴포넌트(타겟 모션 장비)를 할당합니다. 미 할당시 동일 GameObject 내에 있는 Motion Context 컴포넌트를 찾아 할당됩니다.
Play On Awake	이 체크박스에 체크하는 경우, 어플리케이션이 시작할 때 자동으로 모션 클립의 재생을 시작합니다.
Motion Clip	재생을 원하는 모션 데이터(확장자『.bytes』)를 할당합니다.
Motion Clip Key	모션 데이터의 암호화 키를 입력합니다.
Volume	Motion Source 컴포넌트의 볼륨 크기를 설정합니다. (0~100)
Speed	Motion Source 컴포넌트의 재생 속도를 설정합니다. - Speed_0_25x : 기본 속도의 1/4배 - Speed_0_5x : 기본 속도의 1/2배 - Speed_1x : 기본 속도 - Speed_2x : 기본 속도의 2배 - Speed_4x : 기본 속도의 4배
Mute	Motion Source 컴포넌트의 볼륨 크기를 0으로 설정합니다.
Loop	모션 데이터를 반복 재생합니다. (* Play 전에 설정되어야 합니다.)
Source Buffer Format	소스의 버퍼 포맷을 보여줍니다. Motion Clip 로드 시 정보가 갱신됩니다.
Ref Filter	Motion Filter 컴포넌트를 할당합니다.

4.1.2.2 주요 함수

표 21. Motion Source 컴포넌트 함수

Method	Description
<code>int GetClipCurrentPosition()</code>	현재 재생중인 모션 데이터의 재생 위치를 반환합니다.
<code>int GetClipDuration()</code>	현재 로드된 모션 데이터의 재생 길이를 반환합니다.
<code>bool Pause(bool state)</code>	일시정지/재생 명령을 내립니다.
<code>bool Play(int loopCount = 1)</code>	재생 명령을 내립니다. (반복 횟수를 설정할 수 있습니다)
<code>bool Stop()</code>	정지 명령을 내립니다.
<code>bool LoadClip (TextAsset data, string key)</code>	모션 데이터를 로드합니다. (모션 데이터와 암호화 키를 설정할 수 있습니다.)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1.3 Motion Input Component

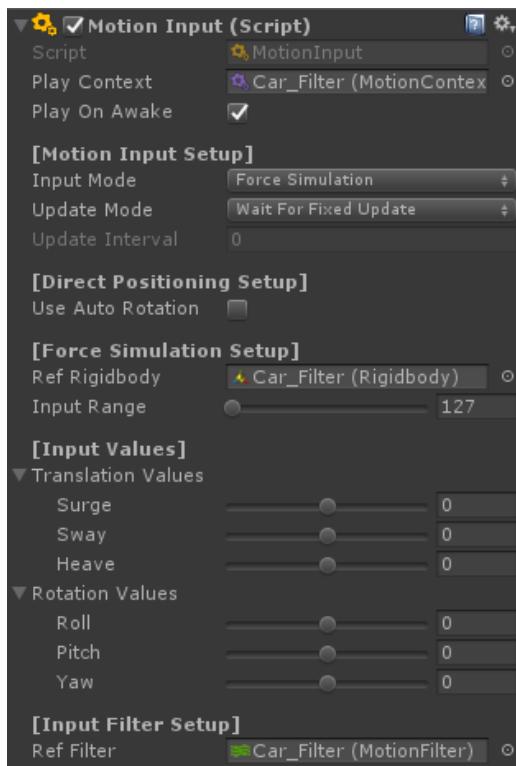


그림 45. Motion Input Component

Motion Input 컴포넌트는 모션 라이브러리의 모션 입력(imInput) 기능을 사용합니다. 모션 입력에 대한 생성, 설정, 스트리밍 등을 간편하게 제어하기 위해 구성되어 있습니다.

Motion Input 컴포넌트의 Input Mode를 선택하여 아래와 같이 모션에 스트리밍할 수 있습니다.

- Direct Positioning Mode : 위치/자세 입력 값을 통해 모션의 위치를 직접 제어
- Force Simulation Mode : 병진 가속도와 각속도를 통해 모션 플랫폼의 위치/자세로 시뮬레이션

● Direct Positioning Mode 의 사용방법

첫 번째 방법은 Rotation Values, Translation Values 구조체 변수 혹은 SetInputValues() 함수를 이용하는 방법입니다. 유저가 직접 이 변수 또는 함수에 -1 ~ 1 (float) 범위의 값을 입력하게 되면 값이 바로 모션에 스트리밍 됩니다.

두 번째 방법은 Auto Rotation Setup 을 이용해 GameObject 의 Rotation 값을 그대로 적용하는 방법입니다. 기본적으로 Rotation 값이 Sin 파형으로 정규화되어 동작됩니다. Translation Value 는 유저가 직접 Translation Values 구조체 변수를 이용해 Heave, Sway, Surge 값을 입력해야 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

위 과정을 통해 입력된 값은 Motion Input 컴포넌트의 Update 코루틴 내에서 Update Mode에 따라 스트리밍 간격이 정해져 스트리밍됩니다. 따라서 Update Mode는 유니티 코루틴에서 지원하는 함수(단, User Direct Call Mode는 예외)를 의미합니다.

● Force Simulation Mode의 사용방법

* Force Simulation을 위해서는 MotionFilter 컴포넌트가 Ref Filter에 할당되어야 합니다.

첫 번째 방법은 Rigidbody를 이용하는 방법입니다. Force Simulation을 원하는 Rigidbody를 MotionInput의 Ref Rigidbody에 할당합니다. MotionInput은 Rigidbody의 정보를 이용하여 Update Mode에 따라 자동으로 스트리밍됩니다.

두 번째 방법은 사용자가 원하는 스트리밍 시점에 SetForceValues() 함수를 호출하는 방법입니다. 매개변수로 Surge / Sway / Heave Acceleration(m/s^2)와 Roll / Pitch / Yaw Angular Velocity(rad/s) 값을 입력합니다.

Update Mode에 정의된 모드를 사용하지 않고 유저가 원하는 임의의 시점에 모션 데이터의 스트리밍을 원하는 경우 SetInputValues(), SetForceValues() 함수의 사용을 권장합니다. 이 함수를 이용하면 Update Mode는 자동으로 User Direct Call Mode가 되어 GameObject의 Rotation 값을 그대로 적용하거나, Rotation Values, Translation Values 구조체 변수에 직접 넣는 방식으로는 스트리밍할 수 없습니다.

4.1.3.1 컴포넌트 설정 값

표 22. Motion Input 컴포넌트 설정 값

Property	Function
Play Context	Motion Context 컴포넌트(타겟 모션 장비)를 할당합니다. 미 할당시 동일 GameObject 내에 있는 Motion Context 컴포넌트를 찾아 할당됩니다.
Play On Awake	이 체크박스에 체크하는 경우, 어플리케이션이 시작할 때 자동으로 모션 데이터 스트리밍을 시작합니다.
Input Mode	Direct Positioning / Force Simulation 모드를 설정합니다.
Update Mode	스트리밍 업데이트 모드를 설정합니다. <ul style="list-style-type: none"> - Wait For Update : Update() 가 실행될 때 스트리밍. - Wait For Fixed Update : FixedUpdate() 가 실행될 때 스트리밍. (권장 기본값) - Wait End OF Frame : 하나의 프레임이 완전히 종료될 때 스트리밍 - Wait For Seconds : Update Interval 값에 맞추어 스트리밍 - Wait For Seconds Realtime : Update Interval 값에 맞추어 스트리밍 (Scaled Time에 영향을 받지 않습니다.) - User Direct Call : SetInputValues함수를 이용해 유저가 직접 스트리밍
Update Interval	스트리밍 업데이트 간격을 초 단위의 값으로 설정하거나 보여줍니다. Update Mode가 Wait For Second 혹은 Wait For Second Realtime일 경우 수정이 가능하며, 그 외의 모드에서는 Delta Time을 보여줍니다.
Use Auto Rotation	GameObject의 Local Roation 값을 모션 데이터로 사용합니다.
Use Global Rotation	GameObject의 Global Rotation 값을 스트리밍 데이터로 사용합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

Ref Rigidbody	Force Simulation을 위한 Rigidbody 컴포넌트를 할당합니다.
Input Range	정규화 연산을 위한 범위 값입니다.
Translation Values	현재 입력된 Surge, Sway, Heave에 대한 데이터를 보여줍니다. -1~1 범위의 float 값으로 Direct Positioning Mode인 경우 설정이 가능합니다.
Rotation Values	현재 입력된 Roll, Pitch, Yaw에 대한 데이터를 보여줍니다. -1~1 범위의 float 값으로 Direct Positioning Mode인 경우 설정이 가능합니다.
Ref Filter	Force Simulation을 위한 Motion Filter 컴포넌트를 할당합니다.

4.1.3.2 주요 함수

표 23. Motion Input 컴포넌트 함수

Method	Description
<code>int StartInput()</code>	모션 입력을 시작합니다.
<code>int StopInput()</code>	모션 입력을 중지합니다.
<code>void SetInputValues (float roll, float pitch, float yaw, float surge, float sway, float heave)</code>	Direct Positioning을 위한 모션 스트리밍 데이터를 입력합니다. (Update Mode가 자동으로 User Direct Call로 변경되며, 함수 호출 시점에 스트리밍 입력을 업데이트 합니다.) <code>roll</code> <code>roll</code> 모션 값을 설정 (-1.0 ~ 1.0) <code>pitch</code> <code>pitch</code> 모션 값을 설정 (-1.0 ~ 1.0) <code>yaw</code> <code>yaw</code> 모션 값을 설정 (-1.0 ~ 1.0) <code>surge</code> <code>surge</code> 모션 값을 설정 (-1.0 ~ 1.0) <code>sway</code> <code>sway</code> 모션 값을 설정 (-1.0 ~ 1.0) <code>heave</code> <code>heave</code> 모션 값을 설정 (-1.0 ~ 1.0)
<code>Void SetForceValues (float surge, float sway, float heave, float roll, float pitch, float yaw)</code>	Force Simulation을 위한 모션 스트리밍 데이터를 입력합니다. (Update Mode가 자동으로 User Direct Call로 변경되며, 함수 호출 시점에 스트리밍 입력을 업데이트 합니다.) <code>surge</code> <code>surge</code> acceleration 값을 설정 (m/s^2) <code>sway</code> <code>sway</code> acceleration 값을 설정 (m/s^2) <code>heave</code> <code>heave</code> acceleration 값을 설정 (m/s^2) <code>roll</code> <code>roll</code> roll angular velocity 값을 설정 (rad/s) <code>pitch</code> <code>pitch</code> pitch angular velocity 값을 설정 (rad/s) <code>yaw</code> <code>yaw</code> yaw angular velocity 값을 설정 (rad/s)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1.4 Motion Filter Component

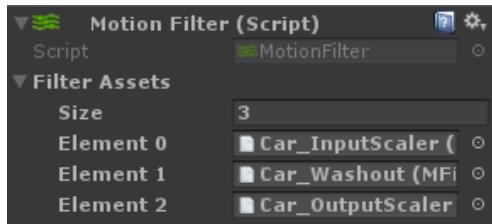


그림 46. Motion Filter Component

Motion Filter 컴포넌트는 모션 라이브러리의 모션 필터(imFilter) 기능을 사용합니다. 모션 필터에 대한 생성, 설정을 간편하게 제어하기 위해 구성되어 있습니다.

이 컴포넌트를 사용하기 전에 먼저 필터 에셋을 생성하여야 합니다. 생성된 필터 에셋은 인스펙터 창에서 사용자가 원하는 필터 값을 입력할 수 있습니다.

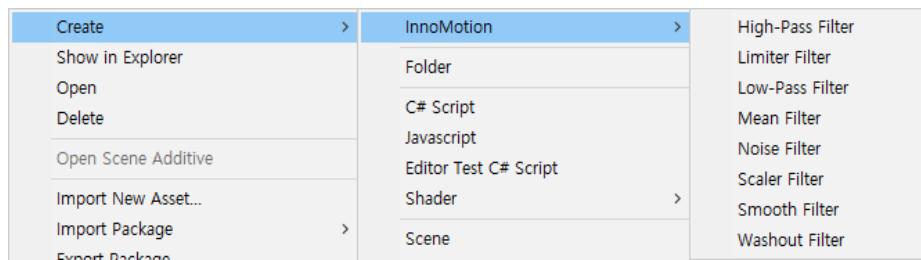


그림 47. 모션 필터 에셋의 생성 방법

필터 에셋은 MotionFilter 컴포넌트의 Filter Assets 프로퍼티에 사용됩니다. 필터 에셋 추가 시 유저가 원하는 필터의 동작 순서대로 추가하여야 합니다. 예를 들어 “Element 0”에 A 필터, “Element 1”에 B 필터가 추가되어 있다면, 모션 데이터는 A 필터를 거친 뒤 B 필터를 거치게 됩니다.

마지막으로 Motion Filter 컴포넌트를 Motion Input 컴포넌트 혹은 Motion Context 컴포넌트의 Ref Filter 프로퍼티에 할당합니다.

4.1.4.1 컴포넌트 설정 값

표 24. Motion Filter 컴포넌트 설정 값

Property	Function
Filter Assets	사용을 원하는 모션 필터 에셋을 할당할 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1.5 샘플 프로젝트

샘플 프로젝트는 유니티에서 제공하는 Standard Assets 을 이용해서 게임 내 모션 컴포넌트를 어떠한 방식으로 적용하는지 확인할 수 있습니다. 또한, 모션 컴포넌트를 스크립트로 조작하는 방법을 숙지할 수 있는 샘플 스크립트와 씬을 제공합니다.

모션시트 버전 1.0 으로 기본 설정이 되어 있으며, 다른 버전의 모션시트 장치를 사용하는 경우 Motion Context 컴포넌트 내 Description 필드에서 Device Version 을 설정해 주어야 합니다.

샘플 프로젝트에서는 프로젝트의 Asset 폴더 내에서 아래의 내용을 제공합니다.

표 25. 유니티 플러그인 샘플 폴더

Sample Folder	Description
Assets / InnoMI / MotionClip	샘플 모션 데이터가 담긴 폴더
Assets / InnoMI / MotionFilter	샘플 모션 필터 애셋이 담긴 폴더
Assets / InnoMI / Scene	샘플 씬이 담긴 폴더
Assets / InnoMI / Scripts	샘플 스크립트가 담긴 폴더

샘플로 제공되는 씬은 아래와 같은 내용을 확인할 수 있습니다.

표 26. 유니티 플러그인 샘플 씬

Scene	Input Mode	Description
SourceComponent	-	Source
SourceComponent_Telemetry	-	Source (Telemetry mode)
InputComponent	Direct Positioning	Input
InputComponent_Telemetry	Direct Positioning	Input (Telemetry mode)
AircraftJet_DirectPositioning	Direct Positioning	Source and Input
AircraftJetAI_DirectPositioning	Direct Positioning	Input
AircraftJetAI_ForceSimulation	Force Simulation	Input and Filter
Vehicle_DirectPositioning	Direct Positioning	Input(Auto rotation) and Input(Key Input)
Vehicle_DirectPositioning_Telemetry	Direct Positioning	Source and Input (Telemetry mode)
Vehicle_ForceSimulation	Force Simulation	Input and Filter
Vehicle_ForceSimulation_Telemetry	Force Simulation	Source and Input (Telemetry mode)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.1.6 플러그인 패키지

플러그인 패키지는 모션 동작을 위한 필수적인 라이브러리 만을 포함합니다. 기존 개발 중인 유니티 프로젝트에 Import 하여 사용할 수 있습니다. 원활한 사용을 위해 샘플 프로젝트를 숙지 후 진행하는 것을 권장합니다.

프로젝트에 Import 후에는 타겟 GameObject 의 Inspector에서 “Add Component”버튼을 이용하여 InnoMotion 카테고리에서 4 개의 컴포넌트(Motion Context, Motion Source, Motion Input, Motion Filter)를 사용할 수 있습니다.

※ 주의사항

유니티 모션 플러그인의 컴포넌트는 Script Execution Order 옵션을 사용합니다. 우선순위는 Motion Filter > Motion Context > Motion Source >= Motion Input 입니다. Edits – Project Settings – Script Execution Order에서 아래 이미지와 같은 순서로 설정 후 사용하여야 합니다.

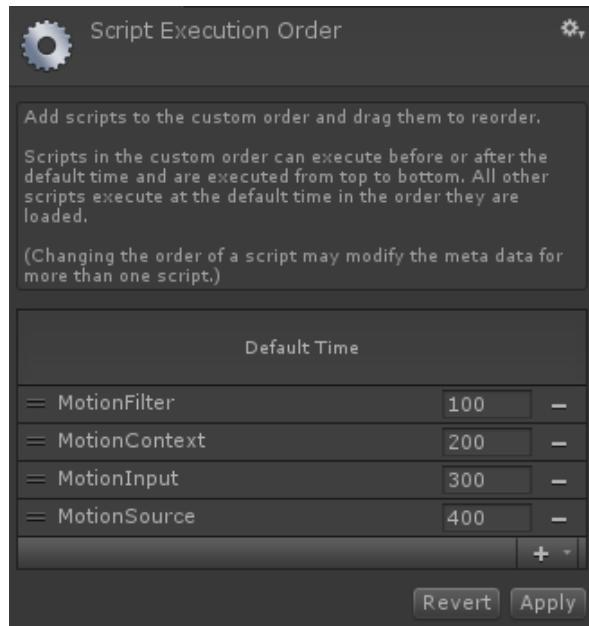


그림 48. Script Execution Order Example

4.1.7 주의사항

- ✓ 빌드 시 Player Settings–Other Settings–API Compatibility level 을 .NET 2.0 으로 설정하세요.
- ✓ Android Platform에 배포하는 경우 Telemetry Mode 를 사용하여야 합니다.
- ✓ IL2CPP 사용 시 플러그인 폴더에 배포되는 link.xml 의 내용을 포함하여야 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2 언리얼 플러그인

언리얼 플러그인은 샘플 프로젝트의 압축파일에 포함되어 제공됩니다. 샘플 프로젝트에는 모션 컴포넌트의 사용 예제가 담긴 레벨과 블루프린트가 포함되어 있습니다.

※ 주의 : 언리얼 플러그인은 4.15 버전 이상 사용을 요구합니다.

4.2.1 Motion Context Actor Component

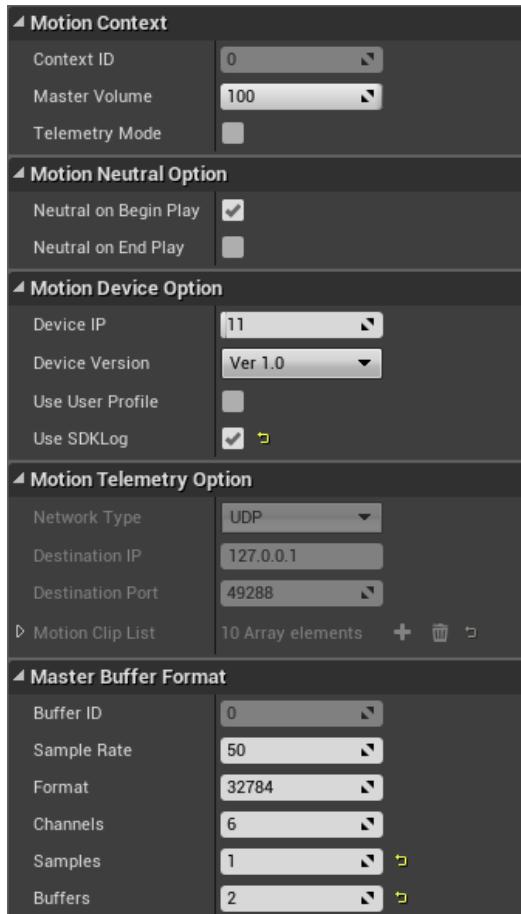


그림 49. Motion Context Actor Component

Motion Context 컴포넌트는 모션 라이브러리의 모션 컨텍스트(imContext) 기능을 사용합니다. 모션 장치에 대한 연결, 상태 등을 간편하게 제어하기 위해 구성되어 있습니다.

유저는 모션과의 연동을 위해 적어도 1 개 Motion Context 컴포넌트를 사용해야 합니다. Motion Context 컴포넌트는 Motion Input, Motion Source 컴포넌트가 참조하게 됩니다. 이를 위해 이들이 같은 Actor 내에 존재하거나, Motion Input, Motion Source 컴포넌트의 Create 함수 2 번째 매개변수에 동작을 원하는 모션의 Motion Context 컴포넌트를 직접 할당하여 사용합니다. 따라서 Motion Context 컴포넌트는 Motion Input, Motion Source 컴포넌트보다 먼저 Motion Create 함수를 통해 초기화해 주어야 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2.1.1 컴포넌트 에디터 설정 값

표 27. Motion Context Actor 컴포넌트 에디터 설정 값

Value	Function
Context ID	컴포넌트 초기화 시 부여 받은 Context ID
Master Volume	마스터 볼륨의 크기를 설정합니다. 0~100 범위의 값을 설정할 수 있습니다.
Telemetry Mode	텔레메트리 모드의 사용 여부
Neutral On Begin Play	Begin Play 시 모션 장비의 중립 위치 이동 명령(Neutral)의 수행 여부
Neutral On End Play	End Play 시 모션 장비의 중립 위치 이동 명령(Neutral)의 수행 여부
Device IP	모션 장비 이더넷 IP의 4 th Octet값 (예: 192.168.0.11 의 경우 11)
Device Version	모션 장비의 Device Version 설정 값 * 3.3.1 모션 장치 연결 참조
Use User Profile	사용자 정의 프로파일의 사용 여부 * 3.3.1 모션 장치 연결 참조
Use SDK Log	에디터 출력로그 창에 SDK 로그의 출력 여부
Network Type	텔레메트리 모드 사용 시 네트워크 타입을 설정합니다. (UDP, TCP, IPC)
Destination IP/Port	텔레메트리 모드 사용 시 모션플레이어 PC의 IP와 PORT를 설정합니다. (Default : 127.0.0.1 / 49288)
Motion Clip List	텔레메트리 모드 사용 시 Motion Source에서 사용되는 모션 클립의 리스트 (최대 10개)
Master Buffer Format	마스터 버퍼의 포맷 정보를 설정합니다. * 3.2.1 버퍼 생성의 Format 구조체 (IM_FORMAT) 참조

4.2.1.2 주요 블루프린트 함수

표 28. Motion Context Actor 컴포넌트 함수

Method	Description
<code>bool MotionCreate()</code>	모션 장치 컨텍스트를 생성합니다.
<code>void MotionDelete()</code>	모션 장치 컨텍스트를 제거합니다.
<code>bool MotionMoveSettle()</code>	모션 장치에 중립 위치 이동 명령(Settle)을 전달합니다.
<code>bool MotionMoveNeutral()</code>	모션 장치에 초기 위치 이동 명령(Neutral)을 전달합니다.
<code>bool MotionMasterVolume(int volume)</code>	모션 장치의 마스터 볼륨을 설정합니다. <i>volume</i> 0 ~ 100 사이의 볼륨 값을 설정
<code>bool SetFilter(int filterID)</code>	모션 장치에 필터를 설정합니다.
<code>int GetFilter()</code>	모션 장치에 설정된 필터ID를 반환합니다.
<code>bool GetActivation()</code>	모션 장치의 활성화 여부를 반환합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2.2 Motion Source Actor Component

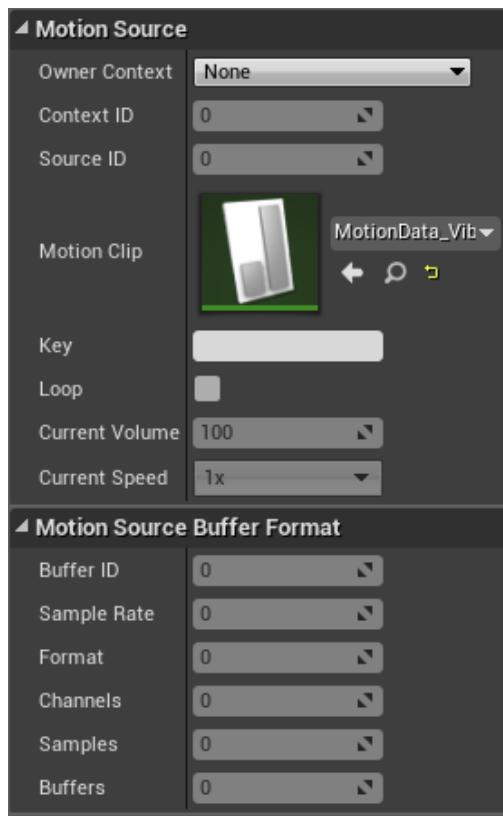


그림 50. Motion Source Actor Component

Motion Source 컴포넌트는 모션 라이브러리의 모션 소스(imSource) 기능을 사용합니다. 모션 소스에 대한 생성, 설정, 모션 클립의 재생 등을 간편하게 제어하기 위해 구성되어 있습니다.

모션 클립은 “.csv” 확장자를 사용하는 모션 데이터 파일을 사용합니다. “.csv” 확장자를 사용하는 모션 데이터를 사용하기 위해서는 “.bytes”로 수정 후 사용이 필요합니다.

일반적인 사용 방법은 다음과 같습니다.

1. 언리얼 에디터의 콘텐츠 브라우저로 모션 데이터 파일을 끌어서 놓습니다.
2. 콘텐츠 브라우저상에 생성된 파일을 저장합니다.
3. Motion Source Actor component 의 디테일 창에서 재생을 원하는 모션 클립을 선택
4. 런타임 중 Motion Source 의 SourcePlay 함수를 이용해 모션 클립을 플레이 혹은 SourceCreate 함수의 playOnCreate 변수를 True 로 설정하여 생성 후 바로 플레이

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2.2.1 컴포넌트 에디터 설정 값

표 29. Motion Source Actor 컴포넌트 설정 값

Property	Function
Owner Context	할당된 Motion Context Actor 컴포넌트를 확인할 수 있습니다.
Context ID	Owner Context의 Context ID를 확인할 수 있습니다.
Source ID	컴포넌트 초기화 시 부여 받은 Source ID를 확인할 수 있습니다.
Motion Clip	재생을 원하는 모션 데이터를 할당합니다.
Key	모션 데이터의 암호화 키를 입력합니다.
Loop	모션 데이터를 반복 재생합니다. (* Play 전에 설정되어야 합니다.)
Volume	Motion Source 컴포넌트의 볼륨 크기를 확인합니다.
Speed	Motion Source 컴포넌트의 재생 속도를 확인합니다.
Source Buffer Format	소스의 버퍼 포맷을 보여줍니다. Motion Clip 로드 시 정보가 갱신됩니다. * 3.2.1 버퍼 생성의 Format 구조체 (IM_FORMAT) 참조

4.2.2.2 주요 블루프린트 함수

표 30. Motion Source Actor 컴포넌트 함수

Method	Description
<code>bool SourceCreate (bool playOnCreate, UMotionContext* context)</code>	모션소스를 생성 합니다. <i>playOnCreate</i> 할당된 모션클립을 생성시점에 재생 <i>context</i> 참조해야하는 Owner Context 를 설정
<code>void SourceDelete()</code>	모션소스를 제거합니다.
<code>bool LoadClip(UDataTable* clip, FString key)</code>	모션 데이터를 로드합니다. <i>clip</i> 모션클립을 할당 <i>key</i> 모션클립의 암호화 키
<code>bool SourcePlay(int32 loopCount)</code>	재생명령을 내립니다. <i>loopCount</i> 재생 반복 횟수를 설정
<code>bool SourceStop()</code>	정지명령을 내립니다.
<code>bool SourcePause (bool state)</code>	일시정지 명령을 내립니다. <i>state</i> 일지정지 상태를 설정
<code>bool SourceVolume(int32 volume)</code>	볼륨을 설정합니다. <i>volume</i> 0 ~ 100 사이의 볼륨 값을 설정
<code>bool SourceSpeed(ESpeeds speed)</code>	속도를 설정합니다. <i>speed</i> 속도 값을 설정
<code>bool SetFilter(int filterID)</code>	모션 소스에 필터를 설정합니다.
<code>int GetFilter()</code>	모션 소스에 설정된 필터ID를 반환합니다.
<code>int32 GetClipDuration()</code>	현재 로드된 모션 클립의 재생 길이를 반환합니다.
<code>int32 GetClipCurrentPosition()</code>	현재 재생중인 모션 클립의 재생 위치를 반환합니다.
<code>bool GetActivation()</code>	모션 소스의 활성화 여부를 반환합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2.3 Motion Input Actor Component

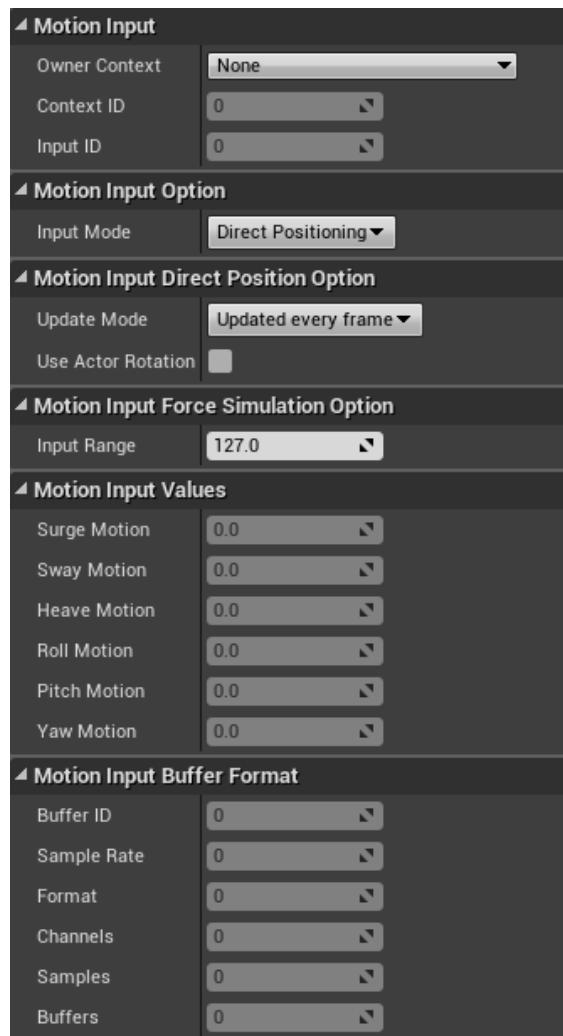


그림 51. Motion Input Actor Component

Motion Input 컴포넌트는 모션 라이브러리의 모션 입력(imInput) 기능을 사용합니다. 모션 입력에 대한 생성, 설정, 스트리밍 등을 간편하게 제어하기 위해 구성되어 있습니다.

Motion Input 컴포넌트의 Input Mode를 선택하여 아래와 같이 모션에 스트리밍할 수 있습니다.

- Direct Positioning Mode : 위치/자세 입력 값을 통해 모션의 위치를 직접 제어
- Force Simulation Mode : 병진 가속도와 각속도를 통해 모션 플랫폼의 위치/자세로 시뮬레이션

● Direct Positioning Mode 의 사용방법

첫 번째 방법은 Heave, Roll, Pitch Motion 변수 혹은 SetInputValues() 함수를 이용하는 방법입니다. 유저가 직접 이 변수에 -1 ~ 1 (float) 범위의 값을 입력하고, Update Mode 를 Updated every frame 으로 설정하게 되면 TickComponent 함수에 의해 모션에 스트리밍됩니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

만약 TickComponent 가 아닌 다른 업데이트 시점에 스트리밍을 원하는 경우 SetInputValues() 함수를 이용하여 호출 시점에 스트리밍할 수 있습니다. 이때에는 Update Mode 가 자동으로 Updated by user function 으로 전환되게 됩니다.

두 번째 방법은 Use Actor Rotation 옵션을 이용해 Actor 의 Rotation 값을 그대로 적용하는 방법입니다.(Direct Positioning) 기본적으로 Rotation Axis 값이 Sin 파형으로 정규화되어 동작합니다. Translation Motion 은 유저가 직접 Heave Motion, Sway Motion, Surge Motion 값을 입력해야 합니다.

● Force Simulation Mode 의 사용방법

사용자가 원하는 스트리밍 시점에 SetForceValues() 함수를 호출합니다. 매개변수로 Surge / Sway / Heave Acceleration(m/s^2)와 Roll / Pitch / Yaw Angular Velocity(rad/s) 값을 입력합니다.

* Force Simulation 을 위해서는 MotionFilter 컴포넌트의 CreateFilter()함수 호출 이후 CreateInput() 함수의 매개변수로 MotionFilter 컴포넌트가 할당되어야 합니다.

4.2.3.1 컴포넌트 에디터 설정 값

표 31. Motion Input Actor 컴포넌트 설정 값

Property	Function
Owner Context	할당된 Motion Context Actor 컴포넌트를 확인할 수 있습니다.
Context ID	Owner Context의 Context ID를 확인할 수 있습니다.
Input ID	컴포넌트 초기화 시 부여 받은 Input ID를 확인할 수 있습니다.
Input Mode	Direct Positioning / Force Simulation 모드를 설정합니다.
Update Mode	스트리밍 업데이트 모드를 설정합니다. - Update every frame : TickComponent 호출 시점에 스트리밍 됩니다. - Updated by user function : SetInputValues 함수에 의해 스트리밍 됩니다. - No Updates : 스트리밍 업데이트를 하지 않습니다.
Use Actor Rotation	Actor의 Rotation을 사용하여 Roll, Pitch Motion을 설정
Input Range	정규화 연산을 위한 범위 값입니다.
Heave Motion	Heave의 float형 값 (-1.0 ~ 1.0)
Sway Motion	Sway의 float형 값 (-1.0 ~ 1.0)
Surge Motion	Surge의 float형 값 (-1.0 ~ 1.0)
Roll Motion	Roll의 float형 값 (-1.0 ~ 1.0)
Pitch Motion	Pitch의 float형 값 (-1.0 ~ 1.0)
Yaw Motion	Yaw의 float형 값 (-1.0 ~ 1.0)
Source Buffer Format	Input 컴포넌트의 버퍼 포맷을 보여줍니다. * 3.2.1 버퍼 생성의 Format 구조체 (IM_FORMAT) 참조

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2.3.2 주요 블루프린트 함수

표 32. Motion Input Actor 컴포넌트 함수

Method	Description
<code>bool InputCreate (bool playOnCreate, UMotionContext* context, UMotionFilter* filter)</code>	모션인풋을 생성 합니다. <code>playOnCreate</code> 할당된 모션클립을 생성시점에 재생 <code>context</code> 참조해야 하는 Owner Context를 설정 <code>filter</code> 참조해야 하는 Filter Component를 설정
<code>void InputDelete()</code>	모션인풋을 제거합니다.
<code>bool InputStart()</code>	시작명령을 내립니다.
<code>bool InputStop()</code>	정지명령을 내립니다.
<code>bool SetInputValues (float roll, float pitch, float yaw, float surge, float sway, float heave)</code>	스트리밍 명령을 내립니다.(Direct Positioning Mode) <code>roll</code> roll 모션 값을 설정 (-1.0 ~ 1.0) <code>pitch</code> pitch 모션 값을 설정 (-1.0 ~ 1.0) <code>yaw</code> yaw 모션 값을 설정 (-1.0 ~ 1.0) <code>surge</code> surge 모션 값을 설정 (-1.0 ~ 1.0) <code>sway</code> sway 모션 값을 설정 (-1.0 ~ 1.0) <code>heave</code> heave 모션 값을 설정 (-1.0 ~ 1.0)
<code>bool SetForceValues (float surge, float sway, float heave, float roll, float pitch, float yaw)</code>	스트리밍 명령을 내립니다.(Force Simulation Mode) <code>surge</code> Surge Acceleration 값을 설정 (m/s^2) <code>sway</code> Sway Acceleration 값을 설정 (m/s^2) <code>heave</code> Heave Acceleration 값을 설정 (m/s^2) <code>roll</code> Roll Angular Velocity 값을 설정 (rad/s) <code>pitch</code> Pitch Angular Velocity 값을 설정 (rad/s) <code>yaw</code> Yaw Angular Velocity 값을 설정 (rad/s)
<code>bool SetFilter(int filterID)</code>	모션 인풋에 필터를 설정합니다.
<code>int GetFilter()</code>	모션 인풋에 설정된 필터ID를 반환합니다.
<code>bool GetActivation()</code>	모션 인풋의 활성화 여부를 반환합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2.4 Motion Filter Actor Component

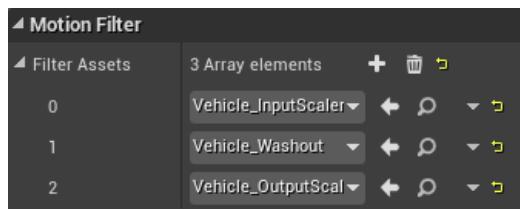


그림 52. Motion Filter Actor Component

Motion Filter 컴포넌트는 모션 라이브러리의 모션 필터(imFilter) 기능을 사용합니다. 모션 필터에 대한 생성, 설정을 간편하게 제어하기 위해 구성되어 있습니다.

이 컴포넌트를 사용하기 전에 먼저 필터 데이터 에셋을 생성하여야 합니다. 생성된 필터 에셋은 디테일 창에서 사용자가 원하는 값을 입력할 수 있습니다.

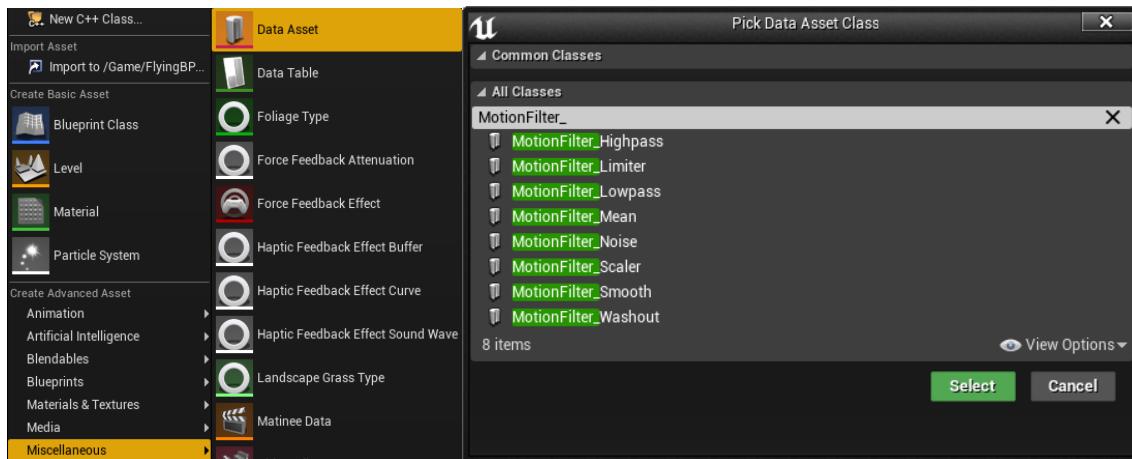


그림 53. 모션 필터 데이터 에셋의 생성 방법

필터 에셋은 MotionFilter 컴포넌트의 Filter Assets 프로퍼티에 사용됩니다. 필터 에셋 추가 시 유저가 원하는 필터의 동작 순서대로 추가하여야 합니다. 예를 들어 “Element 0”에 A 필터, “Element1”에 B 필터가 추가되어 있다면, 모션 데이터는 A 필터를 거친 뒤 B 필터를 거치게 됩니다.

마지막으로 Motion Filter 컴포넌트를 Motion Input 컴포넌트의 InputCreate 함수 호출 시 매개변수로 추가합니다.

4.2.4.1 컴포넌트 에디터 설정 값

표 33. Motion Filter Actor 컴포넌트 설정 값

Property	Function
Filter Assets	사용을 원하는 모션 필터 데이터 에셋을 할당할 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2.4.2 주요 블루프린트 함수

표 34. Motion Filter Actor 컴포넌트 함수

Method	Description
<code>int FilterCreate()</code>	모션필터를 생성합니다.
<code>void FilterDelete()</code>	모션필터를 제거합니다.
<code>int GetFilterID()</code>	모션필터 아이디를 반환합니다.

4.2.5 샘플 프로젝트

샘플 프로젝트에서는 Content 폴더 내에 제공되는 예제 맵과 블루프린트를 통해 모션의 제어 방법을 테스트할 수 있습니다.

모션시트 버전 Ver 1.0 으로 기본 설정이 되어 있으며, 다른 버전의 모션시트 장치를 사용하는 경우 Motion Context 컴포넌트 내 Device Version 을 설정해 주어야 합니다.

샘플 프로젝트는 아래의 샘플 폴더를 제공합니다.

표 35. 언리얼 플러그인 샘플 폴더

Sample Folder	Description
Content / SampleClip	샘플 모션 데이터가 담긴 폴더
Content / SampleFilter	샘플 모션 필터 에셋이 담긴 폴더
Content / SampleMap	샘플 맵이 담긴 폴더
Content / SampleBP	샘플 블루프린트가 담긴 폴더

샘플 프로젝트는 아래의 샘플 맵을 제공합니다.

표 36. 언리얼 플러그인 샘플 맵

Scene	Input Mode	Description
1_SourceSample_Map	-	Source (Telemetry mode available)
2_InputSample_Map	Direct Positioning	Input (Telemetry mode available)
3_FlyingSample_Map	Direct Positioning	Source and Input
4_VehicleSample_Map	Direct Positioning	Source and Input (Telemetry mode available)
5_FlyingFilterSample_Map	Force Simulation	Input and Filter
6_VehicleFilterSample_Map	Force Simulation	Input and Filter
7_VehicleTelemetrySample_Map	Force Simulation	Source and Input (Telemetry mode)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

4.2.6 주의사항

- ✓ 언리얼 엔진 4.15 버전의 경우 다음 경로의 압축파일을 해제하여 Source 폴더에 덮어쓰기 해 주세요.
 - 압축파일 경로 : “Project Name Folder / Plugins / InnoMLBP / Source_Ver15.zip”
 - 덮어쓰기 경로 : “Project Name Folder / Plugins / InnoMLBP / Source”

Android Platform에 배포하는 경우 Telemetry Mode를 사용하여야 합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

5 모션 플레이어

3 장에서 알아본 모션 라이브러리는 쉽고 빠르게 콘텐츠를 개발할 수 있는 많은 유용한 기능을 제공합니다. 하지만, 이 방법은 모션 API 프로그래밍으로 인한 약간의 복잡도 증가와 모션 제어 의존적인 코드를 만든다는 문제가 있습니다.

이러한 어려움을 해결하기 위해, SDK 는 모션 플레이어를 이용한 Telemetry 모션 제어 방식을 추가로 지원합니다. 이번 장에서는 Telemetry 란 무엇이고 모션 플레이어를 이용한 Telemetry 제어 방법에 대하여 알아봅니다.

SDK 는 기존 콘텐츠를 수정 없이 또는 통신 규격 만을 이용하여 모션 제어를 할 수 있는 편리한 기능을 지원합니다. 인터랙티브 VR 콘텐츠 내의 게임 Player 와 같은 제어 객체의 운동 정보를 이더넷 통신 방식으로 익스포트 할 수 있고, 이를 Telemetry 기능 이라고 부릅니다.

이 Telemetry 기능은 콘텐츠 내의 제어 객체의 여러가지 상태 정보들 중에서 제어 객체의 운동력(가속력, 회전력)에 관한 정보를 모션 플랫폼의 위치 정보 (DOF)로 바꾸어 시뮬레이션 할 수 있습니다. 3.5 장 모션 입력의 “Force Simulation Mode”는 워시아웃 필터 (Washout Filter)를 사용하여 Telemetry 방식의 모션 제어를 수행합니다.

Telemetry 모션 제어는 다음과 같은 시스템 구성에 사용할 수 있습니다.

- Telemetry 지원 게임 : 프로그램 수정은 불가능 하지만 Telemetry를 지원하는 기존 프로그램의 경우 (예: Project CARS)
- VR 영상 : VR Player와 모션 Player가 통신 방식의 연결이 필요한 경우 (예: GearVR, PS4)
- 통신 기반 시스템 : 프로그램 수정이 가능하지만 모션 독립적인 시스템 구성이 필요한 경우
- 단순한 시스템 : 시뮬레이션 엔진 부하 없는 모션 제어가 필요한 경우 (예: 파라미터 제어)

5 장에서는 Telemetry 방식의 모션 제어를 위해 꼭 필요한 시뮬레이션 프로파일(Profile)과 모션 플레이어 (InnoMP)의 사용방법을 알아봅니다.

※ 주의사항

SDK 패키지 구성에 따라 모션 플레이어의 일부 플러그인이 기본 구성에서 제외되어 기능이 제한될 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

5.1 텔레메트리(Telemetry)

텔레메트리는 원격에서 얻는 데이터의 자동화된 감지 및 수집을 가리킵니다. 텔레메트리 기능은 점점 복잡해 지는 소프트웨어의 복잡도를 낮추는 ‘데이터 기반 의사결정’라는 새로운 페러다임으로 볼 수 있는데, 특정 사용자 환경의 원격 측정 데이터와 이벤트 데이터를 수집하고 분석할 수 있는 유용한 기능을 제공합니다. 예를 들어 가속도 센서를 갖는 IoT 기기 라든지 인기 상용 레이싱 게임 등에서 전달된 센서 또는 게임 Telemetry 를 이용하여 모션 플랫폼의 액츄에이터나 모터를 제어할 수 있습니다.

Telemetry 모션 제어의 통신 방식은 UDP, TCP, IPC 를 지원하고, 이를 통해 2 가지 방식의 시스템 구성이 가능합니다. 우선 가장 단순한 시스템 구성은 응용프로그램이 직접 Telemetry 를 지원하고, 프로토콜에 맞는 프로파일을 모션 플레이어에서 실행하는 것입니다. 이 방법은 모션 플랫폼과 독립적인 시스템 구성이 가능하지만, 프로파일 작성의 부담과 모션 효과 재생의 한계가 있습니다. 주로, Telemetry 지원 게임에 사용합니다.

두번째 방법은 미리 제공되는 Telemetry 프로토콜 (InnoMI)의 프로파일을 사용하는 플러그인을 이용하는 것입니다. 이 방법은 응용프로그램에 플러그인을 추가하는 수정 작업이 필요하지만, 모션 효과를 지원하고 손쉽게 개발이 가능한 장점이 있습니다.

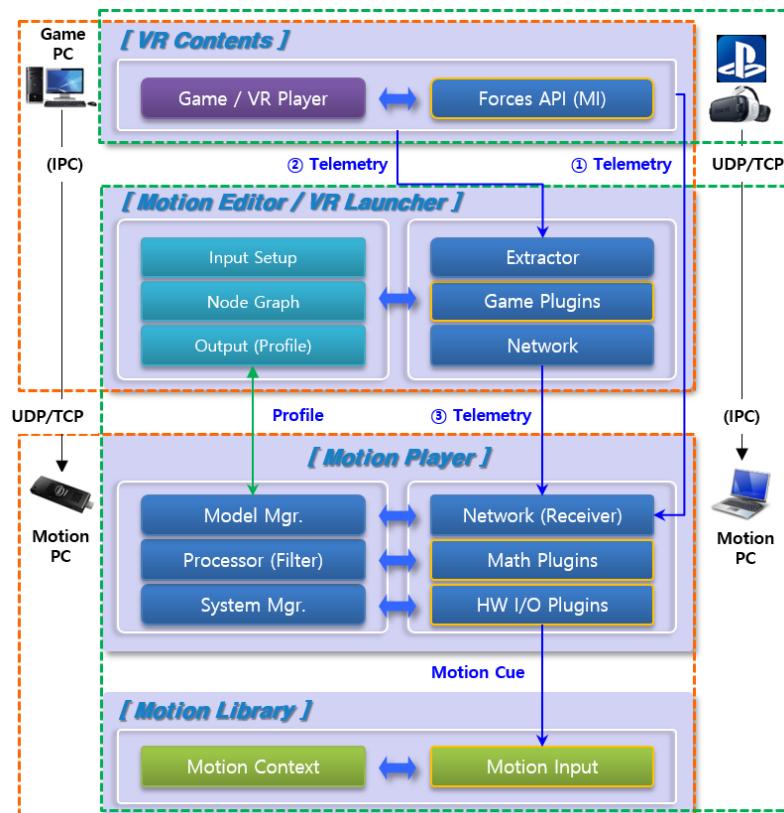


그림 54. Telemetry 지원 모션 시스템 구성

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

2.1 장에서 살펴본 모션 SDK 구성과 같이 그림 54에서 Telemetry 기반 모션 제어의 시스템 구성 예를 보이고 있습니다. 모션 SDK에서는 수집된 Telemetry 데이터를 3.1 장의 모션 라이브러리에서 살펴본 모션 입력 장치의 ADC 신호 중 하나로 보고, 모션 버퍼 (MotionBuffer)에 저장하고 관리합니다. 즉, 모션 라이브러리 및 플러그인의 4 가지 객체와 다음에 살펴볼 모션 플레이어의 각 서브시스템은 공통의 모션 버퍼를 관리 대상으로 합니다.

모션 플레이어는 콘텐츠에서 전달된 Telemetry 데이터를 직접 수신하고 (①), 모션 플랫폼을 제어할 수 있습니다. 그리고 VR 런처의 게임 Telemetry 추출기 (Extractor)와 모션 플레이어가 연결될 수 있어 (②,③), 더욱 편리한 VR 영상 및 다양한 게임 콘텐츠의 운영 환경을 제공합니다.

※ VR 런처는 유료 SDK 패키지 구성에 포함되며, 자세한 설명은 패키지 내의 별도 가이드 문서를 참고하세요.

5.1.1 게임 텔레메트리

특정 인기 레이싱 게임이나 비행 시뮬레이션 게임에서 화면 또는 통신으로 게임 정보를 전송 (export) 하는 유용한 기능이 있습니다. 이를 게임 텔레메트리 (Telemetry)라고 부르는데, 그 정보에는 게임 플레이어 (제어 객체)의 운동에 관한 위치, 회전 등의 변환 정보나 가속도, 각속도의 동역학 정보가 포함 될 수 있습니다.

텔레메트리 지원 게임에서 모션 플랫폼을 제어하기 위해서는, 해당 게임의 텔레메트리 프로토콜을 알아야 합니다. 대부분 각 필드는 4byte 의 부동소수형 데이터 타입을 사용하는데, 모션 제어에 필요한 동역학 정보는 주로 -127 ~ 127 범위의 부동 소수형을 사용합니다. 모션 큐잉에 필요한 가속도/각속도 또는 위치/회전, RPM/Gear 등의 운동 정보의 위치를 알아야 하고, 텔레메트리의 각 필드 (이하 축(Axis)이라 칭함)의 매팅 정보를 프로파일에 기술해야 합니다.

이와 같이 응용프로그램을 수정할 수 없는 경우를 제외하고도, 텔레메트리 기능 지원을 통한 모션 연동이 유리할 수 있습니다. 앞에서 언급한 것과 같이 텔레메트리 기능을 사용하면 모션 제어 뿐 아니라 시뮬레이션에 필요한 여러 유용한 확장 기능을 제공할 수 있으며, 모션 플랫폼의 의존성 없이 강건한 프로그램을 만들 수 있습니다.

5.1.2 InnoMI

모션 SDK 는 InnoMI(Innosimulation Motion Interface)라는 자체 텔레메트리 데이터의 기본 프로토콜을 정의하고 있습니다. InnoMI 는 유니티, 언리얼 등 게임 엔진용 모션 SDK 플러그인에서도 지원하고 있기 때문에 Telemetry 지원 콘텐츠를 쉽게 구성할 수 있습니다. 또한 이를 통해 모션 장치의 존성 없는 멀티 플랫폼 지원의 시스템 구성이 가능하고 모션 제어의 부하를 콘텐츠로부터 제거시킬 수 있는 장점이 있습니다.

InnoMI 의 Telemetry 송신 프로토콜은 다음과 같이 정의합니다. 9 개의 필드를 모두 정의할 필요는 없으며, Mask 로 식별된 최대한 많은 정보를 포함하는 것이 좋습니다.

축(Axis)	이름	설명	구분 (Mask Bit)
0	StructureSize	현재 구조체의 byte size : 132	header
1	Mask	사용 필드의 비트 마스크 값 (1~9)	

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

2	State	모션 상태 전이 이벤트(1~2) : stop, start 장치 상태 조합(0~3) : settle(1), neutral(2)	event state (LSB:16)	1
			event data (MSB:16)	
3	DeltaTime	지난 프레임이 완료되는 데 까지 걸린 시간(ms)	time	2
4~6	[Roll Pitch Yaw]	회전 (radian)	rotation	3
7~9	[Roll Pitch Yaw]_Speed	각속도 (rad/s)	angular velocity	4
10~12	[Surge Sway Heave]_Speed	속도 (m/s)	velocity	5
13~15	[Roll Pitch Yaw]_Acc	각가속도 (rad/s^2)	angular acceleration	6
16~18	[Surge Sway Heave]_Acc	가속도 (m/s^2)	acceleration	7
19~21	Global_Pos_[X Y Z]	월드 좌표계 위치 (X is right axis)	position	8
22	EffectType	모션 효과 (List Type = 0, Reserved Type = 1)	effect	9
23~32	Effect[0..9]	각 필드는 볼륨(0~100), 정지(-1), 일시정지(-2) <List Type인 경우> 배열 index : 모션 효과 파일 식별자(id) < Reserved Type인 경우> Effect[0/1] : Ground Type (Left or Right Side) Effect[2/3/4] : Collision (Roll/Pitch/Yaw) Effect[5/6] : 4D Wind Effect (Left or Right Side) Effect[7/8/9] : Reserved		

InnoMI는 Diagnostics 수신 프로토콜을 다음과 같이 정의 합니다.

Axis	필드	설명	구분
0	StructureSize	현재 구조체의 byte size : 36	header
1	State	현재 모션 상태 (0~3) : init, stop, start, run	state
2	Busy	Pulse 출력상태	IM_DIAGNOSTIC_INFO 구조체
3	Home	홈 서치 동작여부(원점이동)	
4	Alarm	경고신호 동작여부	
5	Inpos	Inposition 입력상태	
6	Emer	긴급정지 동작여부	
7	Cmd	지령 펜스 위치	
8	Enc	엔코더 위치	

※ 주의, 현재 InnoMI는 내부 프로토콜 참고용이며, SDK 버전에 따라 변경 될 수 있습니다.

5.2 시뮬레이션 프로파일

가상 세계의 운동을 실제 세계에서 인간이 운동감으로 느낄 수 있도록 하는 모션 큐잉에 Telemetry 데이터를 활용할 수 있습니다. 주로, 지속적이고 변화량이 적은 저주파 형태의 모션 입력 신호로

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

사용하는데, 다양한 프로토콜의 텔레메트리 데이터 해석 및 모션 제어값에 관한 시뮬레이션 정보를 사용합니다.

3.3 장 모션 컨텍스트의 모션 프로파일과 다르게, 시뮬레이션 프로파일은 상위 시뮬레이션 설정을 정의 하며 모션 효과 파일(*.csv)과 같은 모션 큐 파일로 분류됩니다.

이름	설명	형태	구분
모션 프로파일	모션 디바이스의 설정 (장치 ID)	“IMotion_#id.ini”	환경 설정
모션 효과 파일	모션 효과 데이터 (콘텐츠 내 효과)	“(effect_name).csv”	모션 큐
시뮬레이션 프로파일	시뮬레이션 설정 (콘텐츠의 Telemetry 프로토콜)	“(telemetry_name).ini”	

5.2.1 시스템 설정

시뮬레이션 프로파일은 시스템 설정과 모델 설정으로 나뉩니다. 시스템 설정은 호환되는 모션 SDK 버전 등의 시뮬레이션 프로파일 정보와 시뮬레이션을 구성하는 모션, 네트워크, 멀티미디어 등 각 서브시스템 정보를 기술합니다. 각 시스템의 이름은 플러그인 모듈의 이름을 표시하며 현재 버전에서는 기본 시스템만 지원합니다.

아래 프로파일의 예에서 모션 시스템 항목의 모션 수와 url 정보로 모션 효과 라이브러리를 미리 정의하고 모션 소스로 재생 할 수 있는데, 해당 항목의 변경은 InnoMI 프로파일에서만 지원하는 기능입니다. 그 밖의 다른 대부분의 경우 아래와 같은 기본 시스템 설정을 사용합니다.

```
; InnoMI Simulation Profile
; system configuration

[Simulation]
name = InnoMI
version = 1300

[MotionSystem]
name = InnoMotionSystem
; motions = 1
; motion0_url = ../waveform_sine.csv

[NetworkSystem]
name = InnoNetworkSystem

; model configuration
```

(“*SDK/MotionData/profile/InnoMI_Profile – ForceSimulation.ini*” 참고)

각 서브시스템은 다음 5.3 장 모션 프레임워크에서 좀더 자세히 알아봅니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

5.2.2 모델 설정

시뮬레이션 모델 설정은 서브시스템별로 관리하는 컴포넌트 속성을 기술합니다. 서브시스템은 유한상태기계 (FSM)을 기반으로 컴포넌트의 생명주기를 관리하는데, 각각의 컴포넌트가 공통으로 가지는 enable 속성으로 런타임 상태의 시작과 종료 상태 전이 이벤트를 발생시킵니다.

먼저, Telemetry Packet 의 통신 프로토콜 및 패킷을 위한 컴포넌트의 속성 정보는 아래와 같습니다.

name	value	설명
enable	[0 1]	컴포넌트의 활성 여부, 0: 패킷 송수신 불가
server	[0 1]	모션 플레이어 통신 타입(client/server), 주로 서버로 사용
type	[UDP TCP IPC]	통신 프로토콜
host	String	UDP/TCP : "ip address" (예: "localhost", "192.168.0.1") IPC : "MMF 이름" (예: "\$InnoMI\$")
port	Integer (2byte)	이더넷(UDP/TCP) 포트 번호 (예: 49288)
size	Integer (max 8k)	수신 패킷 바이트 크기 (예: 132)
timeout	Integer	수신 패킷 타임 아웃 밀리초 (예: 1000 – 1초 타임아웃)
channels	Integer (0~12)	네트워크 수신 버퍼의 채널 수 (예: 6)
telemetry#n_axis	Integer	Telemetry 입력 축 번호 – 모션 입력 신호 추출 ※ 주의, 채널수 만큼 telemetry 정보가 필요하며, 미지정 telemetry 축은 무시 됩니다.
telemetry#n_name	String	Telemetry 입력 축 이름 – 화면 표시 용

다음은 모션 신호의 파형 그래프와 멀티미디어를 디스플레이 하는 그래픽스 시스템 Surface 컴포넌트 속성 정보입니다. 이 컴포넌트가 없으면 시뮬레이션이 콘솔창으로만 실행됩니다.

name	value	설명
enable	[0 1]	컴포넌트의 활성 여부, 0: surface(윈도우) 생성 없음
fullscreen	[0 1]	전체화면 윈도우 생성 여부
topmost	[0 1]	최상위 윈도우 활성 여부
transparency	[0 1]	반투명 윈도우 활성 여부
width	Integer	surface 가로 크기 (예: 640)
height	Integer	surface 세로 크기 (예: 240)
margin	Integer	surface의 canvas 영역 마진 값 (예: 10)
color	Integer (4byte)	surface 배경 색상 ARGB (예: 0xff : 블루)
graphs	0~4	surface에 그려질 그래프(모션 버퍼)의 갯수 (예: 3)
graph#n_buffer	String	해당 그래프의 참조 버퍼 컴포넌트 이름 – 정의된 컴포넌트 버퍼의 신호를 그래프로 출력 (예: Packet, MotionInput, MotionContext)

다음은 모션 장치 제어를 위한 MotionContext 컴포넌트 속성 정보입니다.

name	value	설명
enable	[0 1]	컴포넌트의 활성 여부, 0: 모션 출력 불가
pause	[0 1]	모션 시작(0) 또는 정지(1) 제어값 (0: 자동 시작)
start_options	Integer	모션 시작 또는 정지 명령(pause) 사용될 장치 상태 옵션 (0:무시, 1:settle, 2:neutral, 3:settle & neutral)
stop_options	Integer	
id	0~255	모션 장치 프로파일 식별 번호
async	[0 1]	비동기 모션 제어, 모션 프로파일 비동기 옵션 (주로 1사용)
volume	0~100	모션 컨텍스트 전체의 최대 운동 범위 스케일 (볼륨)

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

samples	Integer	마스터 버퍼의 샘플 수
buffers	Integer	마스터 버퍼의 버퍼 수
sampleRate	50~200	마스터 버퍼의 샘플링 주기
channels	0~12	마스터 버퍼의 채널 수

MotionSource 컴포넌트의 속성 정보는 다음과 같습니다.

name	value	설명
enable	[0 1]	컴포넌트의 활성 여부, 0: 모션 효과 재생 불가
pause	[0 1]	효과 재생 여부, 0: 자동 재생
loop	0~255	효과 반복 재생 횟수, 255: 무한 반복
volume	[0 1]	효과 재생의 최대 운동 범위 스케일 (볼륨)
speed	0~100	효과 재생 속도
clip	Integer	효과 파일 ID, 모션 라이브러리의 효과 번호

MotionInput 컴포넌트의 속성 정보는 다음과 같습니다.

name	Value	설명
enable	[0 1]	컴포넌트의 활성 여부, 0: 모션 입력 스트리밍 불가
sampleRate	50~200	입력 버퍼의 샘플링 주기
channels	0~12	입력 버퍼의 채널 수

마지막으로 MotionFilter 컴포넌트는 채널을 Axis 로 정의하며, 입력 축 매핑과 워시아웃 필터 파라미터 설정을 위해 다음과 같이 기술합니다. 현재 버전에서 필터 컴포넌트는 MotionInput 컴포넌트에 적용됩니다.

name	value	설명	구분
name	String	입력 채널 이름	
filter_highpassFactor	0~100	웨이사웃 필터 cutoff frequency(factor / 10). 차수별 설정을 위해 이름에 [A B C] 추가. Translation(0~2) : 3차수, Rotation(3~5) : 2차수	워시아웃
filter_lowpassFactor	0~100	틸트 보정 필터 cutoff frequency(factor / 10). 차수별 설정을 위해 이름에 [A B] 추가. Tilt(6~7) : 2차수	
filter_smoothFactor	0~100	노이즈 필터 gain	
filter_outputFactor	Float	출력 값 스케일	
filter_outputCutoff	Float	출력 범위 제한 (-fc ~ fc)	
maths	Integer	축 매핑을 위한 입력 축의 수 (현재 1만 가능)	축매핑
math#n_type	Integer	축 매핑 타입 (Reserved)	
math#n_input	Integer	입력 축 번호 (Telemetry 버퍼의 채널)	
math#n_offset	Float	입력 값 범위 (입력 범위 이동)	
math#n_factor	Float	입력 값 스케일 (입력 범위 스케일)	
math#n_min	Float	입력 범위 최소 값 (작은 값 자르기)	
math#n_max	Float	입력 범위 최대 값 (큰 값 자르기)	

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

5.2.3 모션 필터 컴포넌트

Telemetry 모션 제어 방식을 사용할 때 모션 큐잉의 유일한 방법은 프로파일의 모션 필터 속성을 적절히 설정하는 것입니다. 모션 필터는 프로파일이 필요한 주된 이유로 볼 수 있고, 모션 필터 설정은 프로파일에서 가장 중요한 부분이기 때문에 좀 더 자세히 알아 봅니다.

5.2 장의 프로파일의 모션 입력에 사용되는 필터는 크게 축 매핑 (channel filter)와 워시아웃 필터 (washout filter)가 있습니다. 고전적인 워시아웃 (classical washout)은 항상 가속도 및 각속도의 6 축을 입력으로 하며, 위치 및 회전의 6 축 정보를 출력합니다.

축 매핑 필터는 수신된 Telemetry 데이터로부터 어떤 순서로 6 개의 동역학 입력 정보를 연결할 것인가를 결정합니다. 이때 Telemetry 정보를 적절히 스케일하고 제한하여 입력 범위에 정규화된 값으로 가공해야 합니다.

워시아웃 필터는 다시 워시아웃과 틸트 보정이라는 변환 작업을 거쳐 최종 모션 큐를 생성합니다. 워시아웃 변환은 가속도 및 각속도의 6 개 채널의 고주파 신호를 이용하여 인간이 느낄 수 없는 수준으로 서서히 원점으로 다시 복귀시키는 변환을 수행합니다. 그리고, 틸트 보정은 가속력에 의해 인간이 느끼는 관성이나 중력을 지속적으로 느낄 수 있도록 저주파 신호를 roll 과 pitch 에 추가해줍니다.

[Axis0] ; ex) 0~2 axes name = Surge(m/s^2) ; washout translation gain(3 order) <u>filter_highpassFactorA = 100 ; 10 hz</u> filter_highpassFactorB = 100 filter_highpassFactorC = 100 filter_smoothFactor = 1 filter_outputFactor = 10 filter_outputCutoff = 127 ; axis mapping (3 -> 0) maths = 1 math0_type = 0 <u>math0_input = 3</u> math0_offset = 0 math0_factor = 10 math0_min = -100 math0_max = 100	[Axis3] ; ex) 3~5 axes name = Roll(rad/s) ; washout rotation gain(2 order) filter_highpassFactorA = 100 filter_highpassFactorB = 100 ; FactorC is Not required filter_smoothFactor = 1 filter_outputFactor = 10 filter_outputCutoff = 127 ; axis mapping (0 -> 3) maths = 1 math0_type = 0 <u>math0_input = 0</u> math0_offset = 0 math0_factor = -100 math0_min = -100 math0_max = 100	[Axis6] ; ex) 6~7 axes name = Tilt1(roll) ; tilt coordination gain (2 order) <u>filter_lowpassFactorA = 100</u> filter_lowpassFactorB = 100 filter_smoothFactor = 1 filter_outputFactor = 10 filter_outputCutoff = 127 ; axis mapping is Not required
---	---	---

(* [SDK/MotionData/profile/InnoMI_Profile – ForceSimulation.ini](#) 참고)

앞의 InnoMI 프로파일의 예에서, 특별히 눈여겨 볼 부분은 마지막 그룹에 6~7 번 축이 축 매핑을 하지 않는 점과 추가적인 2 개 축은 어떤 모션 채널에 연결될 것인가 하는 점입니다. 이해를 돋기 위해 3 장의 그림 36. Classical Washout Filter 구성도를 다시 살펴보면 아래 그림과 같이 1 번부터 3 번까지 3 개의 영역으로 구분할 수 있습니다. 이는 위의 프로파일 필터 컴포넌트의 3 개의 그룹과 정확히 일치 합니다. 즉, Ax ~ Az 는 Axis0 ~ Axis2, Wx ~ Wz 는 Axis3 ~ Axis5, 마지막 Tilt 는 Axis6 ~ Axis7에 연결됩니다. 그리고, 각 영역별로 HPF 와 LPF 의 구분은 적분의 수와 차수로 결정 됨을 알 수 있습니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

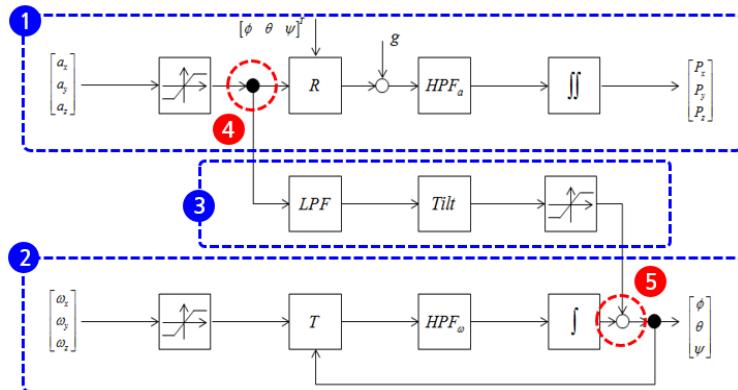


그림 55. 워시아웃 및 틸트 보정 필터

Classical Washout Filter는 워시아웃 채널 (①,②)과 보정 채널 (③)을 합해 총 8 개 채널로 구성됩니다. 그리고, 보정 채널은 각각 Ax 와 Ay 입력을 그대로 사용하므로 (④) 별도의 축 매핑 정보가 불필요합니다. 또한, 보정 채널의 결과와 Roll 및 Pitch 결과를 혼합 (⑤) 하여 최종 결과를 생성합니다.

5.3 모션 프레임워크

모션 플레이어는 모션 데이터 (Motion Cue) 중심의 모션 제어를 위해, 데이터를 해석하고 제어를 수행 하는 모션 시뮬레이션 프레임워크 (이하, 프레임워크라고 함)가 필요합니다. 5.2 장에서 알아본 시뮬레이션 모델의 각 컴포넌트를 관리하는 시뮬레이션 엔진이 있습니다. 각 서브시스템은 관리 대상 컴포넌트에 외부 전이 이벤트를 발생시켜 컴포넌트의 상태를 변경합니다. 프레임워크는 프로파일을 통해 모션 데이터의 정보를 해석하고, 각 서브시스템을 통해 처리됩니다. 서브시스템들은 모션 데이터를 모션 제어, 네트워크 패킷, 오디오 신호, 그래프 출력 등으로 다양하게 해석할 수 있고, 모션 프레임워크는 그 처리를 담당하는 서브시스템들을 포함하고 있습니다.

5.3.1 그래픽스 시스템

모션 플레이어는 Windows 환경의 콘솔창으로 기본 동작됩니다. 프로파일에서 Surface 컴포넌트 없이 실행하면 (제거 또는 비활성), 화면 렌더링 부하 없이 가벼운 모션 시스템 운영 환경을 제공할 수 있습니다.

하지만, 시뮬레이션 개발 단계에서는 사용자가 모션 플레이어의 각 단계별 모션 큐를 확인할 수 있고 프로파일 파라미터의 설정이 필요하기 때문에, Surface 컴포넌트가 유용하게 사용될 수 있습니다. 아래와 같이 모션 플레이어의 팝업 메뉴로 Surface 컴포넌트의 런타임 설정이 가능합니다.

Menu	설명
Open	모션 파일 로드 (시뮬레이션 프로파일, 모션 효과 파일)
Reload	모션 파일 다시 로드 (프로파일 편집 재확인)
Fullscreen	윈도우 전체화면 활성 (ESC key 비활성)
Topmost	윈도우 최상위 활성/비활성
Transparency	윈도우 반투명 활성/비활성

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

Exit

윈도우(시뮬레이션) 종료 (ESC key 종료)

윈도우 타이틀 바에 표시되는 정보는 프로파일 이름, 프로그램 이름, 시뮬레이션 시간, 프레임 레이트가 1초마다 갱신되어 아래 그림과 같이 표시 됩니다.

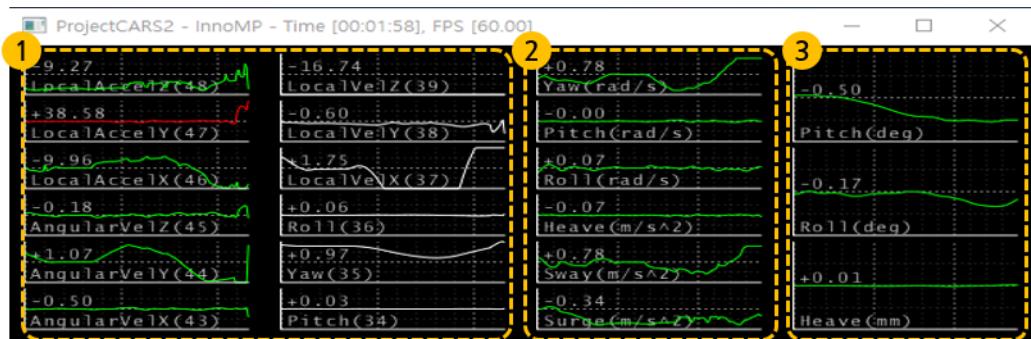


그림 56. 모션 플레이어 화면 구성

앞의 그림은 Telemetry 버퍼 (①), 모션 입력 버퍼 (②), 모션 출력 버퍼 (③)의 각 단계별 모션 파형을 그래프로 표시하고 있습니다. 모션 소스 (네트워크 장치 수신:->①), 모션 입력 (축매핑:①->②), 모션 변환 (위시아웃:②->③), 모션 출력 (모션 장치 제어:③->)의 대표적인 Telemetry 기반 Force Simulation 처리 과정을 보이고 있습니다.

Telemetry 버퍼는 수신된 패킷 정보 중에서 선택된 최대 12개의 채널을 사용할 수 있는데, 그래프의 행 (Row)은 최대 6개로 정의하고 있기 때문에 초과되는 채널은 열 (Col)을 새로 추가하여 출력합니다 (예: 그림 57 두번째 열). “Surface” 컴포넌트에 지정된 각 참조 컴포넌트 (graph#n_buffer)의 버퍼를 화면에 표시합니다.

그래프는 OpenGL 화면 좌표계 기준으로 좌하단을 원점으로 잡아 파형 순서와 좌표축을 표시한다. 색상은 수신 신호의 사용 유무에 따라 (녹색 또는 흰색) 그리고 범위 밖의 신호 (적색)를 표시 합니다.

또한, 그래픽스 시스템은 멀티미디어 시스템과 연계하여 비디오 화면을 출력하는데 사용될 수 있으며, 향후 3차원 시뮬레이션에 사용될 예정입니다.

5.3.2 네트워크 시스템

네트워크 시스템의 주된 역할은 Telemetry 데이터 수신 및 Diagnostics 데이터 송신입니다. 이와 관련한 설정은 모두 Packet 컴포넌트에 정의하고 있는데, 가장 많이 사용하고 있는 UDP 및 TCP/IP 이더넷 통신과 공유메모리 방식의 IPC 통신을 지원합니다.

프로파일의 네트워크 설정을 위해 사용자는 Packet 컴포넌트에 통신 프로토콜에 맞는 IP 주소를 설정해야 합니다. 주로 이더넷 IP 주소는 “protocol://host_name:port”, 그리고 IPC는 “mmf_name”와 같은 방식을 사용합니다. 네트워크 시스템은 런타임 시작과 동시에 통신 연결 이후, 시뮬레이션 루프 마다 패킷의 수신을 감시하고 Telemetry 버퍼에 저장 합니다. 수신 상태일 때 그래프가 좌측으로

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

계속 애니메이션 되지만, 프로파일에 지정된 시간(timeout) 이상 수신이 감지되지 않으면 연결 해제로 판단하고 잘린 상태로 그래프가 표시된 후 점차 사라집니다.

파형 그래프를 활용하여 수신 패킷에서 Telemetry 모션 신호를 찾고 파형의 범위를 적절히 조정해야 합니다. (참고로, Packet 컴포넌트의 Telemetry 버퍼는 다음의 2 가지 방식으로 표현될 수 있습니다. 표시되는 수신값을 Telemetry 데이터 그대로 표시 또는, 표시되는 수신값을 축매핑에 사용된 스케일 계수("math0_factor") 만큼 확대하여 그래프에 보여줍니다.)

- 수신 신호 중 최대 12개의 후보 신호를 선별하여 시뮬레이션 환경을 설정합니다.

(ex : `telemetry0_axis = 43`) // axis range : input (0~packet_size/4-1), output (0~11)

- 보기 쉽게 각 수신 파형의 범위를 조정 합니다.

(ex : `math0_factor = 10`) // motion input range (-127~127)

(ex : `math0_min= -100, math0_max= 100`)

3.5.4 장의 입력 샘플링에서 모션 큐잉 모드 (Force Simulation, Direct Positioning)는 워시 아웃 필터의 사용 유무에 따라 결정되고, 모션 소스 또한 각각 Force (Logical Position)과 Motion (Physical Position)으로 해석되어야 합니다. 이는 Direct Positioning 모드의 운동 범위는 축 매핑에서 결정되고, Force Simulation 모드의 운동 범위는 워시아웃 필터에서 결정됨을 의미합니다.

“Force Simulation 모드”의 축 매핑 파라미터 설정을 위해 주의 깊게 봐야할 2 가지 경우가 있습니다. 특정 축이 원점 기준선 주위에 항상 있는 경우와 (그림 57 의 “Pitch” 축), 반대로 입력 범위 (-127~127)을 벗어나서 빨간색 그래프를 표시하는 경우 (그림 57 의 “LocalAccelY” 축)입니다. 이같은 경우 입력 신호의 손실이 발생할 수 있기 때문에, 운동 범위를 모두 사용할 수 있도록 축 매핑 스케일 계수를 조정해야 합니다. 즉, “Force Simulation 모드”에서 전체 시뮬레이션 구간 동안의 모션 큐 범위 (최소값~최대값)를 적절히 스케일하고 이동하여 최대 입력 범위 (-127~127)에 맞도록 조정해야 합니다. 실제적인 운동 범위는 다음에 수행되는 워시아웃 필터의 출력 파라미터로 조정됩니다.

5.3.3 모션 시스템

모션 시스템은 모션 큐잉에 필요한 MotionInput, MotionFilter, MotionContext 컴포넌트를 관리합니다. 그리고 그림 57 의 모션 입력 버퍼 (②), 모션 출력 버퍼 (③)를 통해 모션 큐잉을 수행합니다.

첫번째, 사용자는 주로 모션 입력 버퍼를 이용하여 원하는 대로 매핑이 되었는지 축 매핑의 결과를 확인해야 합니다. 모션 큐잉 모드에 따라 Telemetry 버퍼의 각 채널의 의미가 달라질 수 있기 때문에, 모션 입력 버퍼 (②)의 6 가지 채널 정보와 일치시키는 것이 중요 합니다. 모션 입력 버퍼의 채널은 이후 변환될 워시아웃 필터의 입력 채널 (그림 57 ①,②)과 동일합니다.

앞에서 말한 것처럼 Telemetry 모션 제어의 첫 단계는 통신을 통해 수신된 패킷 정보를 적절히 가공하는 것입니다. Telemetry 송신기의 프로토콜과 시뮬레이션 환경에 따라 다양한 신호가 입력될 수

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

있으므로, 우리는 가능한 손실 없는 입력 신호를 획득하기 위해 2 번째 파형 그래프 결과를 참고로 프로파일의 속성 조정이 필요합니다.

- 모션 플레이어를 실행하고, 시뮬레이터 실행중의 수신 파형을 관찰 합니다.
(ex : $math0_input = 5$) // channel range (0~6)
- 시뮬레이션 종료까지 각 수신 파형의 범위를 최대치가 되도록 조정 합니다.
(ex : $math0_factor = 10$) // motion input range (-127~127)
(ex : $math0_min = -100$, $math0_max = 100$)
- 모션 플레이어에서 조정된 프로파일을 다시 읽어 앞의 과정을 반복합니다.

두번째, 모션 출력 베피로 워시아웃 변환 결과를 확인합니다. 5.2.3 장 모션 필터 컴포넌트에서 설명한 것과 같이 워시아웃의 원점 복귀 수준, 보정 필터의 추가적인 관성력을 조정합니다.

Telemetry 모션 제어의 두번째 단계는 수신된 신호를 모션 제어 신호로 변환하는 것입니다. 사용자는 마지막 파형 그래프의 출력 결과를 참고로 워시아웃 필터의 채널 별 파라미터를 조정합니다.

- 입력된 채널별로 노이즈 제거 수준을 설정합니다.
(ex : $filter_smoothFactor = 50$) // gain range (0~100)
- 워이사웃 HPF 필터 파라미터 설정으로 원점 복귀 수준을 결정합니다.
(ex : $filter_highpassFactor = 50$) // HPF range (0~100)
- 틸트 보정 필터 파라미터 설정으로 관성 수준을 조정 합니다.
(ex : $filter_lowpassFactor = 50$) // LPF range (0~100)
- 시뮬레이션 반복 시험으로 출력 파형의 범위를 스케일 합니다.
(ex : $filter_outputFactor = 10$)
- 출력 파형의 범위를 제한 합니다.
(ex : $filter_outputCutoff = 127$) // motion range (0~127)

(※주의, 노이즈 필터에 의해 자연이 발생할 수 있습니다. 노이즈 필터는 입력된 신호의 노이즈에 의한 떨림 발생시에만 제한적으로 사용합니다.)

마지막으로, “Force Simulation 모드”에서 모션 출력 스케일 및 컷오프(cutoff)를 조정하여 인간이 느끼는 최종 운동감을 결정합니다.

SDK 는 하드웨어 추상 계층을 탑재하고 있습니다. 모션 프로파일(IMotion_#id.ini)과 시뮬레이션 프로파일(simulator_name.ini)을 각각 장치, 시뮬레이터 마다 따로 생성할 수 있도록 지원합니다.

작성일자	문서명	문서 버전
2019-04-01	InnoVR Motion SDK Manual	Rev. 1.3

SDK 사용자는 모션 플랫폼의 물리적 관점보다 시뮬레이션 환경의 논리적 관점의 개발을 통해 모션 장치 의존성이 없는 보다 강건한 시뮬레이션 환경을 만들 수 있습니다. 그 예로 모션 플레이어는 실제 모션 장치의 축 수와 무관하게 항상 6 DOF 모션 장치로 설정하는 것을 추천합니다.

또한, 모션 플레이어는 Telemetry 모션 제어를 지원하므로, 통신 기능을 탑재한 다양한 플랫폼의 응용프로그램에서 가장 쉽게 모션을 제어할 수 있는 방법이 될 수 있습니다.