|  |
| --- |
| #1-1. Professional [3D Slider Crank] |
| [1] 목적  Slider의 왕복운동을 Crank의 회전운동으로 전환. |
| [2] 시뮬레이션 환경 설정  • Window에서 사용 가능한 기능들  • Home탭-Setting메뉴에서 중력 설정 가능  • Grid가 표시될 Working plane을 설정 가능 |
| [3] Geometry의 생성  • General에서 만들면 하나의 Body가 됨.  • Box표면에 Outline 만들어, Fill사용 하면 표면이 생성됨.  • 생성된 표면에 Extrude하면 그 모양으로 물체 생성.  • 물체 이동 시, Home탭/Measure그룹/Distance로 거리를 확인하고 이동하면 좋음. |
| [4] Joint 생성  • 점 찍을 때 Select Toolbar에서 선택 도움 받기 가능.  (Grid, Geometry, Node 순) |
| [5] Animation및 Analysis 설정  • Body motion Tracing.  (Analysis탭/Post Tool그룹/Trace), 좌표 추적됨.  (참조마커는 Ground.Inertia면 됨)  • Animation Configuration에서 설정 🡪  -보고싶은 Frame 설정 가능 (시작,끝)  -반복 횟수 지정 가능  -애니메이션동안 마커 보이게 on/off 기능 |

|  |
| --- |
| #1-2. Professional [Engine With Propeller] |
| [1] 모델링, Geometry 생성  • 모델링, 여러 파일 import 시 Layer 사용 좋음.  • Home탭-Merge로 여러 바디 1개로 합치기 가능.  (하나의 마커를 가지게 됨.)  • 처음부터 General에서 만들면 1개 바디임. |
| [2] Joint 및 motion 생성  • 모터 축 같은 경우 Cylindrical 조인트  • 0.1초가 지난 후, 10pi 각 가속도 줌.  IF(TIME-0.1: 0, 0, 10\*PI) |
| [3] Force 생성  • Base에서 Action으로 힘주는 방법 (Spring)  • Joint에 힘 주는 방법 (Rev joint에 rotational spring)  • Force의 property에서 Display 여부 설정. |
| [4] Scope 생성  • Analysis탭/Scope그룹 (수식, Entity, 각도, 두 점, Gap)  • Joint의 Scope를 보고 싶음 (Entity)  • 해석 후 확인 가능 |

|  |
| --- |
| #1-3. Professional [Pin Ball] |
| [1] 목적  • Body 사이 & 선과 Body 사이 Contact을 생성.  • Ball이 경사면을 통과할 수 있는 적절한 힘 가하기. |
| [2] Geometry 생성  • 공이 따라 갈 Linear Guide를 Ground 에서 만듦.  (직선은Outline, 곡선은 Arc)  • 이어진 선을 Curve그룹/Edge로 하나의 선으로 만듦 |
| [3] Force와 Contact 생성  -압축된 spring에 의해 Ball1 출발 후, 2충돌🡪2,3충돌  -Ball들은 Guide선과 접촉력에 의해 따라 움직임.  • Spring(base는 ground, action은 Ball)의 free length 조정. (Free length=45, 기존 길이=30에서 15만큼압축)  • 공과 공 접촉: Contact그룹/Sph-Sph  • 공과 Guide 접촉: 2D Contact그룹/GeoCir  -접촉 방향도 중요 Up/Down  -Contact도 property에서 display 여부 설정. |
| [4] Expression과 Scope생성  • Expression에 DX(1), 1은 Ball2.CM(거리 알려는 Ball)  • Scope/Expression에서 Display시 확인 가능. |
| [5] Design Study  • Spring의 Free Length를 자동으로 바꾸며 plot.  • 바뀐 Free Length에 따른 Ball2의 최대 x변위 측정.  • Analysis탭/DOE메뉴에서  Trials횟수=10, 범위=(45mm~54mm), Step은 자동.  • Simulate: Free Length에 대한, 최대 x변위 값 나옴.  • Following Camera -->  특정 Frame에 특정 Entity를 따라갈 수 있음. |

|  |
| --- |
| #1-4. Professional [Ball Return] |
| [1] Contact Surface 정의  • Ball과 Pipe 사이 Contact  • Surface그룹/Face로 Pipe를 FaceSurface1로 만듦.  • 공과 Pipe사이 GeoSph로 contact 생성.  • Container도 Face로 FaceSurface 만든 후 공과 contact  • Contact Surface의 Faceting Resolution  공과 면이 접촉할 때 (GeoSphere Contact)  🡺Faceting resolution조정으로 정확한 해석 가능.  • Contact확인을 위해 Pipe의 투명도 증가 |
| [2] 해석 및 Plot  • Ball과 pipe사이 마찰이 없어 구름 현상 발생 X  • Plot template만들면, 이전/다음 실험 비교 좋음.  • GeoSurContact 1,2에 마찰 계수 추가 후, Template 넣고 Plot ADD를 통해, 서로의 plot 비교 가능.  • Home탭/View그룹/Select Zoom에서 그래프 확대  • Home탭/Tools그룹/Trace에서 값 확인 |

|  |
| --- |
| #1-5. Professional [Dipper Stick With Bucket] |
| [1] PP와 PV  • PP1 점이 지정돼 있다면, PP2점을 (PP1\_X,2,3) 처럼 나타낼 수 있다.  • Main Body와 Subsystem이 만나는 점을 각각 PP로 지정한 뒤, PPC로 연결할 수 있음.  (점이 이어지면, Joint 생성시 점 찍기 유리함.)  • PV값으로 PP좌표를 찍거나, 진폭, 길이 등을 지정하면, PV값을 간단히 바꿈으로 모든 것을 바꿀 수 있음. |
| [2] Expression & motion  • 수식 안에 PV값을 그대로 쓸 수 있음.  PV\_Amplitude\*STEP(TIME,0,0,0.1,-1)  처럼 진폭으로 쓰고 싶은 변수를 수식에 표현 가능. |
| [3] Bucket Tip (bucket앞에 dummy 공 만들기)  • Bucket과 떨어진 곳에 구 생성(dummy body)  • Bucket과 구(bucket tip) 사이 Bucket끝에 fixed joint  • 구에 각속도(WZ)가 0보다 크거나 같으면 50KN생성  50000\*IF(WZ(1,2,2):0, 0, 1)  [IF 사용법] IF(TIME-2: 0, 0, 1)  IFT (if Tolerance)는 일반적으로 0  • Dummy Body 만든 이유  🡺 구와 Bucket이 fixed이므로, 구에 Axial Force를 가해 이동 시키면 Bucket이 같이 움직임. |
| [4] Hydrualic이란 Subsystem에서 Dummy만들기  • Subsystem내에서 Dummy 구를 만들면 같은 시스템이므로 Fixed joint가 필요 없다.  • Subsystem 내에서 마커를 생성해, Axial Force만듦  • Main으로 돌아와서, Ground와 dummy를 fixed함.  • Assembly mode(Main)에서 Expression 생성하기.  🡺Subsystem의 마커는 못 들고 옴. (직접 타이핑)  • Force가 Fixed에 작용할 때, Fixed에 마커 2개는 Database에서 확인 가능하다.  🡺Ground와 Subsystem의 구에 Fixed & force 작용.  Database의 Fixed1에서 Ground랑 구의 marker확인 |
| [5] Output Request 생성하기  • SubEntity탭/Expression그룹/Request  • Request 내 (Standard/Expression/User/RFlex) 중  Expression에 대한 설명임.  • F1에 Expression 넣으면, Tool탭/Math에서 계산 가능  • Plotting시, Database/Request에서 찾을 수 있음. |
| [6] Range of Motion (VE, 변수방정식)  • crank의 link길이와 bucket joint의 각도로 motion임  • VE생성 (bucket의 positive최대 회전)  🡺VARVAL(3)사용. (3에 사용된 변수방정식의 해임.)  • motion의 범위는 VE\_MaxPosRot의 해-VE\_MaxNegRo 의 해를 Degree로 전환한 값의 [최소~최대] 이다. |
| [7] Plotting 및 해석  • bucket joint의 회전량을 알고 싶다.  🡺Joint/Rev\_Dipper\_Bucket의 Pos1\_Relative 확인. |
| [8] DOE (design study)  • crank길이 & bucket joint각도 🡺 motion범위 & cylinder power에 걸리는 영향 확인.  (1) Design Variable 설정. (바꿀 값과 값 범위)  🡺 Crank 길이, Bucket joint 각도 (2개)  (2) Performance Index설정 (PI값)  🡺 Cylinder power의 RMS값, Motion범위의 최대값  (3) DOE (Analysis탭) (DV, PI는 SubEntity탭)  🡺 각 DV마다 Trial 횟수 정하기. (3번, 3번= 총 9번)  [결과] 2개의 DV값에 따른 PI값을 얻음. (Plot 가능) |
| [9] Batch Mode에서 simulation하기.  • design study는 한 번에 한 번만 할 수 있으므로 사용.  • Main에서의 PP, PV만 사용가능. (Sub있으면 connect)  • DV로 쓸 PV값 (길이, 각도)의 DP를 체크 후 Export  • Analysis탭/Scenario에서 시뮬레이션 설정  • Plot 나온 파일을 모두 집어 넣으면 비교 가능. |

|  |
| --- |
| #2-1. ProcessNet/VSTA [4WDLoader] |
| [1] ProcessNet의 목적  • C#으로 모델링, 해석, 플로팅 과정 자동화  • 튜토리얼: 휘어지는 호스 각 segment의 contact자동화  • 도움말/ProcessNet Help에서 예제 확인 가능. |
| [2] ProcessNet  • 여러 바디는 for같은 반복문을 사용한다.  Ex) for(int i=BodyNumStart; i<=BodyNumEnd; i++)  • IBody, IGeometry 및 메소드(GetEntity) 등을 이해함.  • 코딩 후 Build/PNet에서 Run  🡺자동 contact의 경우 코딩된 만큼 contact이 생성됨 |
| [3] 다이얼로그 설계  • Access의 폼 만들기와 같은 단계이다.  • 버튼만으로 위 코드들이 작성할 수 있게 매크로 작성. |
| [4] 자동 Plot 생성  • 폼 형성 ([3]의 폼 복사해도 괜찮음🡪코드 바꿔야함)  • 새로운 Subroutine 생성 후, 코드 작성 (액세스와 비슷)  • 그래프의 크기, 축 범위, legend 등 모두 설정 가능.  • 자동 생성된 Contact만큼 plot에서 확인 가능. |

|  |
| --- |
| #2-2. ProcessNet/VSTA [Excavator] |
| [1] ProcessNet의 기능  • #1-5. Dipper의 추가 연습임.  • 매크로 사용할 다이얼로그 생성 (폼 형성)  • 다이얼로그 우클릭 후 View Code 시, 버튼에 코드 삽입 |
| [2] 새로운 클래스 및 함수 생성  • Import(), GetEntity, IGenerice() 등으로 import후 모델 수정 및 생성 가능함.  • SubEntity, Joint 등 리커다인 기능을 모두 사용 가능.  • 다이얼로그에 만든 함수 연결.  🡺다이얼로그 view code에서 만든 인스턴스를 생성.  • [추가]  해석/plot 등 마우스로 눌러서 진행되는 것들도 코딩으로 자동 생성 가능함. |

|  |
| --- |
| #2-3. ProcessNet/VSTA [Simple Belt System] |
| [1] 목적  • belt툴킷 등으로 만든 Link body는 각각이 Clone body 이므로 Contact, Force, Mesh등을 할 수 없다.  🡺PNet으로 모두를 General Body만들면 가능. |
| [2] 알고리즘 구성  • 첫 Clone을 General로 바꿈. (converCloneBody() 사용)  • General1과 Clone2사이에 Connector 생성  • Clone2도 General2로 바꿈 🡺 반복함. |
| [3] Register DLL  • ConverCloneBody와 CreateContact 2개를 생성함.  • 위 2개를 다이얼로그(매크로)로 생성 후 RecurDyn 리본 메뉴에 등록하고자 함. 🡺 DLL  • ThisApplicatrion.cs에 함수 선언해 줌.  • Icon등록 코드 작성.  • Register DLL을 위한 함수 작성 후 RecurDyn에서 run |
| [4] DLL 사용  • Clone Body에 Edit으로 접근  • Surface그룹/Face로 표면 만듦  • Main화면/Customize탭/생성한 Convert그룹에서 ChangeBody클릭  • Contact생성도 같은 메뉴에 있음. |

|  |
| --- |
| #2-4. ProcessNet/General [4WD Loader] |
| • Visual Studio에서 PNet 사용법  • 자동 contact 생성/자동 plot 생성 (코드 같음)  • Visual Stuido에서 다이얼로그(폼) 형성도 가능 |

|  |
| --- |
| #2-5. ProcessNet/General [Excavator] |
| • Visual Studio로 Window가 바뀌었을 뿐, 코드 및 다이얼로그 형성 과정이 모두 같음. |

|  |
| --- |
| #3-1. AutoDesign [Three Ball Contact] |
| [1] 최적화 목적  • 빨강/노랑/파랑 공이 있음. 빨-노는 접촉 정의 됨.  • 빨-노 충돌 후 빨강이 파랑으로 갈 때 둘의 중심을 최대한 가깝게 하고 싶음.  • 빨강의 초기속도/접촉 시 강성계수를 설계변수로 함. |
| [2] Design Variable 설정  • DV로 쓸 값은 PV에 정의 되어 있어야 함.  • 초기 속도를 PV에 정의 후 빨간 공 property에서 설정  • 강성(Stiffness)도 Contact에서 설정해 줌. |
| [3] AutoDesign/Parameter 설정  • 설계 변수와 설계할 범위를 지정함. |
| [4] Analysis Response (AR) 설정  • AR에 사용할 Expression 정의  (1) Red와 Blue 변위: DM(1,2)  (2) Red Yellow접촉: CONTACT(1,0,1,2)함수  (3) Red와 Yellow 변위: DM(1,2)-600  • AR List 작성  (1) Ex1의 Min 값 (red, blue거리)  (2) Ex2의 Max값 (red, yellow 접촉 힘)  (3) Ex3의 Min값 (red, yellow 거리) |
| [5] Design Study (DV와 AR의 관계 정리)  (1) Design Variable 탭  • Level Set 설정 (변수 당 나눌 step 수)  🡺변수가 2개면 제곱 형태 (Level set=5 🡪 trials=25)  (2) Performance Index 탭  • AR 값들이 잘 있는 지 확인.  (3) Simulation Control  • 해석 설정 후, Execution  (4) Effect Analysis  • PI (AR종류에 따라) DV의 값과 Draw 가능  • Draw개형에 따라 Linear, Non-Linear 확인 가능.  (contact force=0일 때, 둘은 접촉하지 않음)  • 초기 설계에 대한 값 (초기 속도/강성)  🡺 AR조건에 맞게 최대/최소화 하는 값을 찾아줌.  (5) Screening Variables  • AR에 따른 DV값들의 분포를 알 수 있음.  • Cutoff Value로 그 값보다 Sensitivity한지 알 수 있음.  (6) Correlation Analysis  • AR끼리의 관계를 알 수 있음.  • 한 개는 x축(horizon) 다른 건 y축(vertical)로 설정. |
| [6] Design Optimization (설계 최적화)  • 빨강의 초기 속도  • 빨강과 노랑의 접촉 강성  🡺 둘을 고려해 빨간 공이 파란 공과 충돌 하도록 설계  • AutoDesign탭/Opt  (1) Design Variable  • DP값은 DV나 상수 값으로 설정 가능 함.  (2) Performance Index  • AR값을 표현 식으로 조건을 주는 것.  • Objective (목표값, 도달할 값)/Constraint(조건, 제한값)  -최소값 1에 도달할 때까지 (Objective, MIN, 1)  -AR1<=100 (Constraint, LE, 100)  (3) Optimization Control  • 해석 설정 및 Execution |

|  |
| --- |
| #3-2. AutoDesign [Catapult System] |
| [1] 최적화 목적  • 투석기로 돌을 던져 좁은 관에 넣으려 함.  • 속도와, 돌과 관 중심 사이 거리 차로 target position. |
| [2] Design Variables  • 설계 값: (1)link angle, (2)spring mound height |
| [3] Performance Index (PI)  • target에 얼마나 잘 도달하는 지 조정 가능한 부분.  • AR1: target에 대한, 돌의 y축 방향 속도 오차.  🡺 End value  • AR2: target의 y축 방향에 대한 위치 오차.  🡺 End value  • AR3: target에 대한 위치 오차  🡺 Min value |
| [3] Design Optimization  • Objective: 이 값보다 크거나/작으면 좋겠다 하는 값  Constraint: 조건. 찾고자 하는 값은 아님.  • y속도오차 5mm/s이하, y위치오차 5mm이하.  🡺 실린더에서 돌 위치 사이 오차는 사실 0임.  5mm이하 🡪 Constraint, LE, 5로 설정  0mm 임. 🡪 Constraint, EQ, 0으로 설정 |
| • ball과 target사이 GeoSurContact 생성 후 확인 가능. |

|  |
| --- |
| #3-3. AutoDesign [Paper Distributing System] |
| [1] 최적화 목적  • Robust와 6-sigma 설계 문제이다.  • 3 noise factor variables + 2 random design variables  • MTT2D 종이 (Sheet Group, fixed/movable roller)가 재료 특성(random variable)에 관계없이 특정 지점을 지나가도록 설계. |
| [2] DV, DP  • SheetGroup1의 두께, Young’s modulus, curl radius 3개는 DV이지만, 제어 불가능이므로 Random constant  • MTT2D의 시트(Guide)는 PV로 알 수 없다.  🡺 바로 DV는 하지 못하고, Motion을 통해 알아냄.  PV\_Yupper\*STEP(TIME, 0, 0, 0.01, 1); 짧은 시간 간격  • DP(Design Parameter) 5개  (1) curl(-50~50), (2) modulus (5200~7200), (3) Thicknes (0.1~0.3), (4) UpperPos(-1~1), (5) lowerpos(-1~1) |
| [3] PI (Performance Indexes)  • 재료특성(두께, E, Curl)을 noise factor라고 함.  🡺noise factor를 Random Constant라고 함.  • baffler의 위아래 위치를 DV로 함. (직접은 못 찾음)  🡺DV가 오차가 있으면 Random variable이라고 함. |
| [4] Robust Design Optimization  • 목적(objective): (1)위치에러의 분산 최소화, (2) paper x위치를 894mm로 target  • 조건(constraint): y위치 위아래로 편차 +/-1 까지 허용  🡺 목적의 위치에러 분산 최소화는 조건의 편차값에 영향을 받음. 🡪 Robust가 필요한 이유  • Robust내에서 DV/PI/Opt설정 후 결과 확인  (1) DV: 상수/변수 & Random/determine & SD/COV (SD는 절대값, COV는 상대값) & Dev.value를 설정함.  (2) PI: 기존 AR과 Robust의 AR에 어떤 식으로 구해짐 = Weight\*(AR\*Alpah\_Weight+Sigma\*Robust\_Index)  🡺 가중치로, AR만 쓸지 Robust만 쓸지 both 설정. |
| [5] 6-Sigma Design optimization  • Robust와 같은 DV, AR을 사용함.  • AR중 하나인 Sum of error를 최소화 하고자 함.  • 식에 편차\*6이 들어가는 거 같은 정확하게는 모르겟음 |

|  |
| --- |
| #3-4. AutoDesign [Suspension System] |
| [1] 최적화 목적  • Tire Motion중 Yaw와 Roll 범위를 동시 최소화 설계  • 목적함수 2개, 설계 변수 27개. 다중 목적함수 최적화. |
| [2] DV와 성능 지수 정의  • DV는 PP점 9개, 각 점 마다 3축 값으로 PV는 27개  • Yaw와 Roll범위 값이 있진 않음. (Expression, VE 사용)  🡺Yaw,Roll의 최소 최대 구함+편차 구함=범위  • AR에서 각 편차의 End값을 구함. (과도응답 최대편차) |
| [3] 최적 설계의 진행 (Opt)  • 어떻게 설계할 것인지 (목적, 조건)  목적: Yaw, Roll의 범위를 줄임. Yaw를 2배 더 많이  Minimize Yaw\_Deviation\*2 & Roll\_Deviation\*1  조건: 현 Yaw, Roll값에 맞는 최대편차보다 작게  현재보다는 더 최적값이어야 하므로  • 결과: Yaw를 0.3%, Roll을 33.7% 최소화하는 AR1=0.651, AR2=1.407 이라는 값을 얻음. |
| [4] 선별된 (Screened) DV이용한 최적화 설계  • 기존과 비교하려면 simulation history 모든 결과 지움  • DV Study로 해석  (1) Screening으로 민감한 DP값을 볼 수 있음.  (2) PI의 종류 🡪 AR1, AR2에 따라 볼 수 있음.  (3) Cutoff로 나눌 sensitivity 값 지정 가능.  (4) cutoff기준 더 민감한 값들이 On됨.  (5) 모델에 적용 🡪 DV list에 On된 값만 DV로 체크됨  • Screening 이후 최적화  27개의 DV가 선별 이후 10개로 줄음.  [결과] 선별 변수가 줄어 해석이 간단해짐. |

|  |
| --- |
| #3-5. AutoDesign [Paper Feeding System] |
| [1] 최적화 목적  • 용지의 미끄러짐에 대한 연구  • 결과: 팍 튀는 peak값을 잡아내 RMS로 돌림. |

|  |
| --- |
| #3-6. AutoDesign [Landing Gear System] |
| [1] 최적화 목적  • Colink의 PID Gain값을 AutoDesign을 통해 결정.  • 바퀴를 2초안에 격실로 이동 시키고 안정화 하는 것.  🡺 이 목적 함수를 만족하는 최적의 Gain값 찾기. |
| [2] DV  • PV에 P, I, D 이름으로 값 만듦.  • Colink에서 P gain, I gain, D gain 각각에 PV값 넣음.  • P gain의 경우는 하한치를 1로 (0이면 제어가 안됨) |
| [3] AR (해석응답)  • 목표1: 2초에, 바퀴중심과 target 사이 거리 편차 최소  • 조건1: 안정성 위해, 최대 AR이 위 벽을 치지 X  • 조건2: 시스템 Plant input이 한계보다는 작아야 함.  • 조건3: 편차가 완전히 0이 되기 위한 추가 구속 |
| [4] 최적 설계  (위 조건을 Opt에서 설계)  • Autodesign과 Colink의 해석 시간 간격을 맞춰야 함.  🡺 endtime(5) Step(500) 0.01초마다 해석  🡺 Colink창 Plant에서 샘플링 시간을 0.01로 설정.  • 최적의 P, I, D 값을 얻을 수 있음. |

|  |
| --- |
| #3-7. AutoDesign [Connecting Rod Shape] |
| [1] 최적화 목적  • 2개의 반지름 DV(1,2) 4 개의 Rod 넓이 길이 DV(3-6)  으로 모양에 따른 최적의 응력 값을 계산할 것. |
| [2] DV, DP(Design Parameter)  • AutoDesign/Parameter에서 FEShape로 수정해야 함.  🡺노드 선택 가능(우클릭 후, Select list) |
| [3] AR  • 마찬가지로 Basic 🡪 FE Result로 수정해야 함.  • AR1은 Vomises 응력, AR2는 Mass로 설정. |
| [4] Opt  • 조건1: 응력은 114보다 작게  목적1: 질량 1보다 작게  • [결과] 응력은 114보다 작으면서 질량을 58%나 줄이는 최적의 Shape를 찾아냄.  🡺DV1,2(radius), DV3,4(width), DV5,6(height) |
| [5] 두 해석의 해석 비교 (응력) |

|  |
| --- |
| #4-1. Colink [Pendulum] |
| [1] 제어 목적  • 5도 정도 기울인 추를 PID제어를 통해 빠르게 제어 |
| [2] 모델 만들기  • Import할 때, 추가 5도 기울어져 있음.  • Rev Joint를 연결 후, Property에서 5도 기울임.  • 땅과 Base의 Tra joint/Tra Force 같은 점에 생성.  • Tra Force🡪 방향 힘 설정. (FX에 수식 넣음) |
| [3] Colink 생성  • GPIN: TransForce의 FX 방향 힘  • GPOUT: 추의 각도 (Scope시, 각도 그래프)  🡺-5도 기울어진 것에 Kp=-1000을 줘야 반대로 힘을 줘서 정상상태 됨. (Kp는 음의 값이어야 함)  • P제어: 추의 각도를 0으로 만듦 (진동 O, 초과 힘)  • PD제어: 힘과 각도가 0에 수렴함. (진동 X, Base 이동)  • PID제어: I에 의한 오버슈트 발생. But (진동X, Base가 원래 위치에서 움직임 멈춤) |
| [제어 추가설명]  • P제어는 각도 -5도에 -1000곱하면 각도에 반대 방향 5000N힘. 시스템에 따라 무거우면 더 줘야함. Gain다름  • PD제어는 D제어가 수렴시켜줌. 🡪 각도가 한쪽에만 머물면서 추가 이동함.  🡺D제어는 각속도를 받음. (P의 미분 값).  (1)추가 -5도 기울어지면 Kp가 (-)부호니까 +방향 힘.  (2)각속도는 P제어에 의해서 +방향임.  (3)Kd를 -500주면 (상대적 큼), -방향 힘이 상대적으로 커서 처음엔 느리게 움직임.  (4)값이 거의 수렴해 속도가 느려져야함 (D제어) 이때는 수렴전까지 Kd가중치만큼 +방향 힘.  (5)수렴전까지 큰 값(Kd)으로 +방향 힘. 빠르게 날라감.  • PID제어로 I의 오버슈트가 생기며, 각도가 양쪽에 생김. 🡪 추의 움직임과 각도를 같이 제어함. |

|  |
| --- |
| #4-2. Colink [Car] |
| [1] 제어 목적  • 느린 자동차 뒤의 빠른 자동차를 배치해 따라가게 함.  • 장애물 감지 시, 바퀴 토크 양을 조절해 정지시킴.  • PID제어를 이용한 크루즈 제어 시스템 모델링 |
| [2] Car 모델링  • 바퀴 joint & 커플러, 초기속도 설정 및 Tra joint의 마찰  • Rev Joint는 Z축방향, 오른손으로 돈다.  • Tra joint는 땅, 차, 점, Ground.inertia의 +X방향  🡺 차의 초기 속도와 마찰을 설정할 수 있음. (Joint탭)  • Coupler (커플러)  (1) Front 좌/우 Rev joint와 Tra joint를 연결  (2) 커플러의 식이 있음. 🡪커플러 Property에 적용  (3) Rear 좌/우랑 Tra joint 연결  (4) Front좌/우 Rev joint도 커플러 함. +Rear도 함  🡺Scale을 Right 1, Left -1로 해서 동일 반대방향 토크  Right와 left가 둘 다 1이면 자동차 안 움직임.  [결과]  커플러를 이용해, Tra joint의 motion으로 바퀴가 굴러감 |
| [3] Colink 설정  • Car를 복사 후 이동시킴. 초기 속도를 느리게 바꿈.  🡺Tra joint초기속도 말고 Body 초기 속도로 줌.  • PIN: 앞바퀴 양쪽에 Joint에 Rot.Axial Force로 넣음  🡺 즉, Driving Torque임  • POUT1: 앞과 뒤차의 상대 속도임 VX(2,1)  • POUT2: 앞과 뒤차의 상대 위치임 DV(2,1)    Colink제어는 기본을 0으로 만들어줌.  🡺22000을 빼줌으로 상대 위치는 22000을 유지하며 상대속도는 0을 유지하는 PID제어를 만들었음. |
| [4] Colink 시스템 확장  • 차선변경: 차는X축 움직, Y축 움직여 차선변경함.  🡺Dummy 공을 만들어 Y축 Tra joint에 Motion줌.  🡺기존 땅과 Tra joint를 Base를 Dummy공으로 바꿈  • POUT3: Dummy공의 Y축 방향 변위  • POUT4: 뒤차의 Absolute Velocity |
| [4]의 Colink 설계 과정  (1) Demux에서 [3]공의 Y축 변위, [4]뒤차 절대속도 추가  (2) [3]를 Threshold=4500인 스위치에 u[1]으로 넣음  u[0]는 {35760-[4]뒤차속도}를 PID제어한 값임.  u[2]는 기존 상대속도 0 상대위치22000 PID제어임.  🡺 이때 스위치를 통과한 u[1]의 값이 PlantIn값임.  (3) 공의 Y축 변위가 4500보다 작으면, u[0]값이 토크임.  u[0]는 PID 제어인데, D제어 시 미분 시간 상수를 1e-6 으로 수정해야함. (주파수)  (4) [결과] 앞 차 출발 시 뒤차가 급 가속함.  🡺급감속은 괜찮지만, 급가속은 불편함.  따라서, Rate Limiter를 추가해 급 가속을 제어해봄.  Switch이후 PIN이 들어가기 전, Rate Limiter 블록  🡺 Rising은 3e6까지, Falling은 -1e10까지 제어 |
| (1) [4]의 (1) 차선변경 수식에 의해 앞 차는 2~3초 사이에 -5000만큼 Y축 방향으로 이동함.  (2) 더미 공이 4500이상 이동하면 스위치에서 u[0]로 가던 값이 u[2]으로 바뀜.  (3) u[0]는 아래 코링크 u[2]는 기존(위) 코링크임  (4) PIN인 Driving torque는 뒤 차에 걸리는 거임. |

|  |
| --- |
| #5-1. Control [Pendulum (FMPY)] |
| [1] 제어 목적  • 리커다인 FMI방식 입/출력 설정  • 파이썬 FMPY 및 TensorFlow 모듈 활용해 연동해석  • 파이썬+Colink제어 |
| [2] 파이썬 모듈 설치.  • cmd 관리자 권한 실행  -cd.. 상위 폴더 이동  -cd Users\dr01/Appdata~~로 주소 이동 가능  -C:\ D:\ 등으로 드라이브 이동 가능  • Python폴더 내에서python -m pip install tensorflow 등 으로 모듈 설치 가능. (Tensorflow, FMPY) |
| [3] 파이참 CPU문제로 실행 불가능  • FMPY는 리커다인과 Host와의 통신용  • TensorFlow는 PID Gain값을 이용하고 이용하는 N-N 제어 (PID보충용) 모듈이다. |

|  |
| --- |
| #6-1. TSG [Automotive Road Testing] |
| [1] 제어 목적  • 실제 계측값들을 RecurDyn모델에 적용해도 조금 이상  • RecurDyn모델도 실제 비선형 값 반영 어려움  🡺가진기 입력신호로 신뢰성을 높임. (TSG toolkit) |
| [2] 모델  • 바퀴 Shaker는 Ground와 Tra joint로 위/아래만 움직  • 바퀴는 한 면으로만 움직일 수 있게 Inplane joint  • joint/contact은 Relation MAP으로 쉽게 확인 가능. |
| [3-1] 신호 정의하기  • 생성될 신호 개수만큼 Actuator (가진기) 정의함.  🡺툴킷에서 정의 후, Expression에서 TACT(Actuator)  • Shaker 각 4개에, Actuator식 각각 motion으로넣어줌.  [3-2] Sensor 정의하기  • Sensor의 도출결과는 Target 신호랑 비교  🡺 현 튜토리얼에서는 Chassis 가속도 측정할 것.  (ground와 chassis의 ACCZ,ACCY측정)  [3-3] Target Signal정의하기  • 사용자가 정의해야함. 🡪시험 등을 통해 얻어진 데이터  (TSG의 성능 평가 지표임)  • Sensor개수 만큼 Target Signal 정의 필요함(2개 필요)  (.csv파일로 시간1,데이터1,시간2,데이터2 순이다)  • Target생성 |
| [4] FRF (Frequency Response Function) 수행  • 시간영역 신호를 주파수영역 신호로 변환  • 그 과정에서 시스템의 특성 전달함수 (H(f))를 도출  • FRF파일 위치, Setting해주면 FRF Result에서 확인.  Actuator와 Sensor의 번호에 따른plotting 가능 |
| [5] Iteration 수행  • FRF결과로 Sensor에 측정되는 Response 신호를 사용자가 정의한 Target신호랑 최대한 일치 시키기 위해 Actuator에 가해지는 Drive Signal을 반복해 도출함.  • 반복횟수를 설정함 (보통10번) 그 중 RMS에러를 plot할 수 있고, 가장 작은 애가 좋은 것.  • 각 Actuator별로 Sensor vs Target을 확인 가능.  • Response Signal을 plot함으로 기존과 얼마나 오차가 있는 지 확인 가능함. |

|  |
| --- |
| #7-1. Flexible - FFlex [Compliant Clutch] |
| [1] Driver돌려 FE응력분석 (목적)  • FFlex는 다른 body와 Sliding/Rolling Contact인 경우 |
| [2] 모델 Import  • Mesh하지 않고 데이터를 Import할 수 있음 (.dat파일)  • Display에서 Geometry 화질을 높일 수 있음.  • 중간 Driver와 모델, 끝에는 Load와 모델을 연결.  • 중간 Driver, Load에 둘 다 Rev joint 생성해서 돌게 함.  • Drive토크는 일정(STEP), LoadTorque는 각속도^2  🡺 수식 만들어서, Rot.Axial Force에 넣고 joint에 적용. |
| [3] Surface와 Contact (rigid와 flexible사이)  • Rigid는 면이 1개지만, FE는 여러 개 모아야 면 1개임  🡺 필요한 요소들을 모아 만든 면을 Patch Set이라 함.  (Patch Set은 Rigid body와 접촉하는 면.)  • Rigid도 Patch Set과 접촉하는 면을 수정함.  🡺 Face Surface로 만들어줘야함.  • Contact: Patch Set과 Face면에 접촉을 생성함.  🡺 12개의 접촉. (강성, 감쇠, 마찰계수를 설정 가능)  • Face 생성 시  Face는 여러 개의 면을 하나로 만들어주는 거임.  🡺 1개 solid, 1개 면만 쓰면 face로 안 만들고 사용 O |
| [4] Boundary Condition 생성 (자유도줄여 해석 단축)  • FE의 모든 유한요소 Node에 BC생성할 것.  • BC: Z방향 Node의 Motion을 0으로 구속.  🡺 Clutch가 Clutch plate랑 분리를 방지함.  • FFlexBody의 Edit/ B.C아이콘에서 수정함. |
| [5] Flexible body 해석 및 Contour  • Home메뉴/Setting그룹/Flexible에서 contour 설정  • Clutch plate의 응력 Contour display  🡺 Stress/SMISES  • user로 max값을 조금 낮추고, Smooth로 바꾸면, 좀 부드럽고, 구별되게 볼 수 있음.  • Plot: FE에서 Contact힘 plot 가능. (접촉의 총 힘값)  🡺Plot내 Home탭/Multiple Axes그룹/At current axis 옵션을 킴.  •Tool탭/Analysis그룹/Filter에서 Cutoff주파수 설정가능  🡺 확 튀는 고주파 영역을 필터를 통해 없애기 가능. |

|  |
| --- |
| #7-2. Flexible - FFlex [Plasticity Bending Testing] |
| [1] 절곡기 목적  • Flexible body의 탄성/소성 해석 차이 비교  • 소성해석을 위한 필요조건/재질 적용/분석 방법 알기  • 소성 재질 FE, 판재(plate)를 Punch의 낙하로 굽힘 만듦 |
| [2] FFlex Body 생성하기.  • 박스 생성 후, 우클릭/Mesh  • Mesh할 표면을 Face를 통해 만들고 AutoMesh함.  🡺 생성된 Property에서 두께, 포아송비, 소성/탄성, 탄성계수 등 재료 특성 설정 가능.  • Boundary condition (B.C설정)  🡺 보고 싶은 X, Y, Z, RX,RY,RZ만 설정 (해석 시간 단축)  FE의 변형이 예상되지 않은 부분은 구속 해제 |
| [3] 탄성(Elastic) 해석  • Mesh된 표면에 Patch Set을 적용해야함.  🡺 양면에 contact을 위해 윗면 Patch1, 아래면 path2  • 윗면은 Punch랑 contact, 아래면은 Die랑 contact  • FFlex그룹/Contour에서 응력 해석 가능  🡺탄성이므로 접힌 뒤 되돌아감. (최대응력=2817MPa) |
| [4] 소성(Plastic) 해석  [Plastic/Isotropic]과 [Plastic/isotropic\_kinematics] 해석  • FE edit모드에서, 재료property를 소성으로 수정  • Elastic은 선형이지만 Plastic은 비선형임  🡺 Strain과 Yield stress에 대한 매칭을 해줘야 함.  • Output에 필요한 Node만 설정.  • 해석: 탄성이 없어 절곡기가 튀어오르지 않음.  🡺 응력도 최대 294Mpa로 줄음. (잔류응력이 있음) |
| • 3차원에서 응력을 표현하면, 영구변형이 일어나는 순간을 항복곡면으로 표현함.  • 힘을 가해 소성이 되고 힘을 해제하면, 항복응력 증가  🡺 이를 경화(Hardening)이라 함.  •경화의 종류 2가지 (Isotripic 경화, Kinematic 경화) |

|  |
| --- |
| #7-3. Flexible - FFlex [Bimetal Thermometer] |
| [1] TMFBD(thermal MFBD)목적  • 열전달+ FFlex body (열에 의한 변형) 확인  • mesh+열해석 property+joint와 thermal condition |
| [2] Mesh(tip.mesh잘못하면 설정 새로 후 다시 mesh)  • 열 측정할 코일 Mesh함.  • 두 Metal에 각각 Face해서 surface 2개 생성  • Advanced Mesh (Edge마다 요소 수를 설정)  • manual Mesh 생성하기  (1) Mesher/Extrude에서, 방향, 요소 수, 길이 설정  (2)FFlex edit그룹/Property에서 surface2의 특성만듦  (3)surface 1,2에 맞는 property로 Extrude Mesh  (4) 끝 면을 Set메뉴/Node탭으로 SetNode로 만듦.  🡺 surface1과 2를 F.Merge로 같이 만들어줘야 SetNode 점 찍을 때, 같이 다 찍힘.  (5) Node는 집합체임. (ID만 지정해주고, FDR을 통해 SetNode와 연결해야 함)  (6)FFlex Edit그룹/Material에서 비열, 전도도 등 재료 특성 설정 가능함.  (7) Property탭에서, 생성한 Material을 적용해줘야 함.  (8) rigid와 접촉 하는 위,아래 면은 Node set설정. 나머지 면은 대류convection위해Patch Set 설정해 줌. |
| [3] Boundary & Thermal condition추가  • FE(Bimetal Mesh)와 기존 Thermometer랑 연결할 joint와 Convection을 생성함.  • FFlex그룹/Convection에서 열효율 등 설정.  • Expression에 온도에 따라 침의 각도가 바뀌게 설정  🡺 Convection1의 property/유체 온도 값 변경으로 온도계의 온도를 조절할 수 있음. |
| [4] Node Set / Patch set 차이  • Node set은 여러 개의 slave Node를 하나의 Master node로 사용가능하게 함.  (1) Joint 생성 가능 🡪 특수 상황 (모든 요소 joint)  • Patch set은 일반 상황에서 사용 가능  (1) Contact 만들기  (2) post analysis/Fatigue 설정 등  (3) convection (열전달) |
| [5] Auto Mesh / Advanced Mesh  • Mesh하는 도형 형상이 복잡하면 Adv. Mesh 해야함.  • Advanced Mesh의 Seed Control로 요소 수 설정 됨.  • (추가) 도형의 surface만 Mesh해서 Extrude가능  🡺 온도 측정 등 내부 Mesh는 필요 없을 때 사용 가능  • Adv.Mesh로 Mesh모양설정 가능 |

|  |
| --- |
| #7-4. Flexible - RFlex [Excavator] |
| [1] 목적  • 포크레인의 Boom부분은 Sliding, rolling contact이 없으므로 Modal 방식의 RFlex 사용 가능.  • rfi파일 import로 RFlex body만들고, contour로 응력해석 |
| [2] plot 해석  • 파일을 Save as로 저장 후, Plot에서 import시 여러 상황의 DataBase를 한 번에 볼 수 있다.  • 유연체 (RFlex)는 강체 (start)보다 고유 주파수가 낮음.  🡺 최고치까지의 시간 간격이 RFlex가 더 길다.  • RFlex가 진폭이 더 큼. (경사 크기가 크다) |
| [3] RFlex Body의 검토  • RFlex는 FEA Modal분석으로 얻는 Body의 Modal Shape에 대한 Linear중첩에 의존함.  • 즉, 두 종류의 Mode가 사용됨.  • 두 Mode가 Body의 Modal Shape에 따라 일치 안할 수도 있음.  🡺 공학적 직관력으로 어떤 Mode를 쓸 지 판단 가능. |
| [4] RFlex Body의 Mode  • Mode는 관측하려는 진동수라 보면 됨.  🡺 Mode7이 24Hz면, Mode 8은 28Hz에서 보기 가능.  (어떤 주파수에서 어느 부분이 약하며, 진동하는 지 봄)  • Modal 주파수의 감쇠율  Body가 1000Hz이상에서 영향 적게 받음. (감쇠율 1)  0<f<100 Hz에서 영향 큼 (감쇠율 0.01)  🡺감쇠율이 크면 구조의 동작에 영향을 적게 줌.  • OR, 영향적은 1000 Hz이상 Mode를 모두 선택 해제 함으로 해석 시간을 두 배 이상 줄일 수 있음.  🡺 다른 파일로 저장 후, 같은 그래프에 Import  기존 RFlex와 New Modal RFlex 그래프가 거의 동일함. |

|  |
| --- |
| #7-5. Flexible – RFlex 생성 [RFlexGen Crankshaft] |
| [1] 목적  • FFlex, RFlex body 교체 방법 및 동특성 파악.  • FFlex, RFlex body로부터 RFI 파일 생성법. |
| [2] FFlex (Mesher)와 joint, force유지 법.  • Body를 Mesh후, Assist 메뉴를 클릭.  • Body내 FDR 생성여부, [Sel.]을 통해 joint, force를 유지할지 선택 가능 (sel.선택 시 유지됨)  • Assist설정을 Adv.Mesh에서 Include 여부 설정.  🡺 FDR (joint, force)관련 설정  Database (PropertyComponents)에서 확인 가능.  • 유연체 Stress결과 등을 Plot확인하려면, Output생성  Mesher탭/FFlex Edit그룹/Output 아이콘  🡺 이전에 생성한 마스터 노드 (ID 438) 을 치고 엔터  (마스터 노드: FDR에 연결된 모든 slave node의 대표) |
| [3] RFlexGen기능으로 FFlex🡪RFlex Body 생성하기  • RFlexGen 아이콘 클릭.  🡺 MultiNode나 MultiFDR로 필요 부분 클릭 후, 변환  • 위 과정으로 RFI파일이 생성됨.  • 기존 FFlex 바디에 ImportRFI  🡺 FFlex의 Joint가 그대로 RFlex에 적용된 채 변환됨.  • 변환 후 Plot값은 Output새로 설정 후 확인 가능.  • FFlex🡪RFlex mode 20개 (해석 훨씬빠름)  • No.of Normal Modes 100개 (더 정확한 해석) |
| [4] Plot 해석  • 유연체 노드 수가 5만개임 (약 15만 자유도)  • FE의 Output생성에 의해 Plot에서 Node의 Von-mises 응력 확인 가능  • Crankshaft는 Body가 다른 바디와 contact없고, Joint로만 구속되어 있어, RFlex로도 해석 가능함  • 해석 정확도는 10%라 조금 떨어지지만 굉장히 빠름.  • normal mode 수를 20🡪100 시, 정확도 굉장히 근접함. |
| [5] FFlex vs RFlex  • Reduced Flex (RFlex)  유연체가 다른 Body와는 접촉이 없고, 변형이 작으며, 선형 범위 내 모델을 빠르게 해석함.  • Full Flex (FFlex)  유연체와 롤링/슬라이딩 접촉뿐 아니라 비선형 거동 등의 큰 변형도 처리할 수 있음. (정확함) |

|  |
| --- |
| #8-1.Post Analysis–Durability [FFlex ConnectingRod] |
| [1] 목적  • Durability: 단일 작업주기 동안 flexible body수명 해석 |
| [2] FDR (joint사용)과 접촉 (contact)  FDR (Force Distributing Rigid)  • Flexible-Mesh에서, Node를 통해 Master Node생성 후, Master Node에 포함된 모든 Slave Node에 힘/조인트를 체결하고 싶을 때 사용.  • 즉, 여러 노드에 joint 로드를 분배하는 데 FDR을 사용  • 접촉만 사용해서 해석 가능함. (좀 더 현실적)    컨택과 FDR(joint사용)의 응력 차이점.  🡺컨택은 하단에 Von mises 응력 집중됨.  Post Analysis에서, 응력이 높은 영역의 피로수명을 확인하기 위해 피스톤과 커넥팅로드 사이 여러 접촉으로 정의된 FFlex 모델을 사용할 것임. |
| [3] 피로해석 하기  • 피로해석 할 표면 정의 (Patch Set) 🡪 Rod 전체  • 표면 정의만 했으니, 이전 Animation(해석)을 들고 옴.  • Post Analysis탭/Durability그룹  (1) Preference  -Material 재료 라이브러리 경로 설정  -Surface factor 등의 fatigue factor 설정  (2) Fatigue  -Material에서 (철4130, 플라스틱 등) 재료 선정 가능.  -피로해석 할 Patch set을 설정함.  -Time history에서 보고 싶은 Frame만 설정 가능.  🡺 시간과 주기에 따른 손상(Max), 수명(Min) 확인  (3) Contour (피로해석 결과)  -확인할 Time history로 설정해야 함.  -빨간색이 가장 낮은 수명  -수명이 5.264e+08이면, 1일 1시간 16년의 부품수명  (3000rpm으로 1시간 마다 90000(1/0.04)\*3600)  • 충분한 애니메이션 프레임  프레임 많을수록 정확한 피로 분석  Ex) History\_181,361,721은 각 프레임 수이다. 하지만 부품의 최소 수명은 모두 비슷하므로 181도 충분함. |

|  |
| --- |
| #8-2.Post Analysis – Durability [FFlex Suspension] |
| [1] 목적  • FFlex만들어, 내구해석.  • 내구해석 조건, 수행법, 결과 및 분석법 |
| [2] FFlex body 생성 1 (Advanced Mesh)  • 내구해석은 유연체만 가능함. (Flexible body)  • 모델링 요소(joint, force)는 유지한 채 변환  🡺 assist에서 FDR 유지여부/joint, force유지 (Sel.버튼)  • Assist를 include하고, Adv.Mesh함. |
| [3] FFlex body 생성 2 (Auto Mesh)  • 조금 덜 정확하지만, 오류 안 남.  • Geo.Refine으로 다듬기 🡪 AutoMesh만 영향 줌.  • Assist설정에 의해 FDR과 joint가 남아있음. |
| [4] FE생성 후, Durability Analysis (내구 해석)  • 생성한 2개의 FE모두 내구 해석할 것임.  🡺 두 FE모두 Patch Set으로 만들어 줌.  • Patch set만 만든 경우, 애니메이션 불러와도 됨.  • Post Analysis탭/ Durability그룹 (내구 해석)  (1) Preference  -Notch Factor Amp (Kf, Kt) : 2로 설정  구조물에 크랙, 구멍, 노치(V자 홈)등이 존재할 때 계수.  🡪 클수록 울퉁불퉁한 것  (2) Fatigue  -재료 선정 (Steel 1020), PatchSet 설정, Frame설정 (Time History)  -Fatigue Tools에서, 응력크기, 평균응력에 따른 Cycle.  (3) Contour  -수명이 보고 싶다면 Life를 클릭  -Min을 빨강으로 하면, 수명이 작은 부분이 빨강이 됨 |
| [5] 아래 FE의 내구 해석  • Fatigue 재료선정에서, Failure에 따른 응력진폭을 임의로 설정해줄 수 있음.  • Preference 범위 지정으로 응력크기 범위 지정 가능.  • Occurrence: 피로실험 반복 횟수 (몇 번째에 깨질건지)  [해석결과]  • 피로해석에서, 아래FE는 위FE보다 Stress범위와, 평균 응력이 40배 크다. 🡪 내구해석도 수명 짧을 걸로 예상  • 아래 FE의 Von mises 응력만 보면 89Mpa로 항복응력 (262MPa)보다 낮아 안정적인 설계임.  • 내구성을 고려 (피로수명 가장짧은 주기)해 내구성에 대한 설계는 필요해 보임. |

|  |
| --- |
| #8-3.Post Analysis – Durability [RFlex CrankShaft] |
| [1] 목적  • 내구해석: 유연체가 동적하중으로부터 어느 시점까지 특정 부위가 안정적인지 판단.  🡺 단순히 최대 응력/최대 변형률 구하는 것과 다름.  • 4기통 엔진을 RFlex로 바꿔 내구 해석 해봄. |
| [2] RFlex Body 설정  • 유연체 property/Body탭에서 초기속도 설정 가능.  • Str.Shape에서, Strain, Stress 모양을 생성 가능.  • RFlex/Contour아래, Out.Regen에서 보고 싶은 Stress 성분을 선택 가능.  • 모든 성분을 보면, Contour에서 Full하게 볼 수 있음. |
| [3] Durability (내구 해석)  • 유연체 전체를 PatchSet으로 설정  • Preference  -노치 값(Notch Factor) 수정  • Fatigue  - Axial Mode (Uni-Axial, Bi-Axial 설정)  - Life Criteria (Stress, Strain, Safety Factor)  - Material 설정  - Rainflow차트, Damage가 가장큰 Patch영역 차트  • Contour  - 설계상 문제 있는 (안전계수가 낮은) 영역은 빨강으로  - Fatigue에서 Material변경 후, 안전계수 확인 가능  - Material을 내가 임의로 설정 할 수 있음.  - 이음새 부분만 안전계수 보고 싶다면,  🡺 해석 시간 줄이기 위해, PatchSet영역을 줄임. |
| [4] 결과 분석 및 검토  • 내구해석으로 얻는 피로결과🡪피로수명, 안전률  • 일반적 안전률(Safety factor)는 최대응력/허용응력임  🡺 피로해석으로 얻어지는 안전률은 이와는 다름.  • 안전률은 피로한도, 극한응력의 관계에서 어떤 방정식 을 가지냐에 따라 다름.  (직선: Goodman, 포물선: Gerber 등) |

|  |
| --- |
| #8-4.Post Analysis–Acoustic[Vibrating Transmission] |
| [1] 목적  • 소음, 진동 제어 (시스템의 진폭이 큰 주파수 찾기)  • Vibrating Transmission은 회전을 왕복운동으로 만듦  • 유연체 해석만으로 진동해석 어렵🡪Acoustic tool  • 진동이 발생하는 Housing에서 ERP계산하기. |
| [2] Housing 모델  • Housing은 4개의 bearing에서 접촉힘을 받음.  • 접촉힘이 Housing에 어떤 영향? 🡪 housing을 FE로  • FE내부 클릭 위해, Select Element후 Set Masking  (1) 베어링과 접촉할 면에 Patch Set 설정  (2) GeoSur (접촉)을 Patchset과 베어링에 줌  (3) 4개 베어링 모두 접촉 줌.  (4) Edge contact, Force display, 강성, 댐핑계수 설정  • 해석 후, Flexible탭의 Contour (유연체 응력 해석) |
| [3] ERP (Equivalent Radiated Power) 계산  • ERP란?  :유연체 표면에 발생하는 진동의 주파수 응답 분석방법  • 유연체면 Acoustic해석 가능. (RFlex는 Mode별 ERP)  • ERP를 계산할 범위를 Patch Set으로 설정  🡺 Housing 옆면 전체 선택  • Acoustic 계산하기 (Post Analysis탭/acoustic메뉴)  (1) Calculation  -주파수 설정 (0.1초 500step = 5000Hz)  -Housing 옆면(ERP확인할 Patchset)을 Add함.  -Frame 설정(1~501) = 0.1초마다 1step  (2) Scope  -시간영역 ERP/주파수영역ERP 구하기 가능.  -시간영역에서 주기와 상관 없는 구간이 있는지 확인  (혼자만 진폭이 엄청 큰 부분 🡪 Modal ERP고려)  -Calculation에서 Patch set의 Mode와 Frame 재설정  -필요한 주기만 Scope에서 확인되면 FFT(주파수영역)  (3) 결과  -특정 Mode에서 진폭이 큼 (영향을 많이 받음)  -Acoustic그룹/Contour에서 ERP로 설정 시 값 확인됨  -재료를 바꾼 유연체에 대해 ERP가 낮아짐을 확인 |

|  |
| --- |
| #9-1. Toolkit-Belt [Lawnmower with V-Belt] |
| [1] 툴킷 Belt 생성  • toolkit에서 Belt를 선택 후, 파일 import (subsystem)  • 돌아가는 애(Pulley, V) 돌리는 애(Pulley, Roller)  • Assembly로 체인 만듦  • Pulley랑 팬은 fixed로 고정하고, Pulley를 Roller의 벨트에 rev joint에 motion줌으로 Pulley,팬을 같이 돌림  • Tensional에는 Axial힘을 Mower에 줌.  🡺 스프링은 탄성, Axial은 수식 1개, Trans는 방향마다  • 롤러에 벨트가 빠지지 않게 Flange(플렌지) 생성.  🡺 Database의 BeltAssembly에서 플렌지 및 요소확인  • 잔디 깎을 때, 걸리는 힘을 Rev joint의 마찰 Include  • Plot에서, Tensional의 Axial힘과 Belt의 장력은 비례 |
| [2] Belt 모델 추가 작업  • 진동 모션을 추가 (벨트 요소에 슬립 센서 생성 후plot)  • 모터랑 V풀리 사이 미끄럼 측정 센서  🡺 Sensor탭/Slip아이콘  • Plot에서, Slip sensor로 최대 미끄러짐 시간  • plot에서, 시간 지나면서 미끄러짐이 줄어들어 블레이드 회전속도 (Rev속도)가 벨트랑 가까워짐.  • 진동 추가 법  Ground와 데크를 Rev joint로 연결 후, motion을 추가함  🡺 25D\*sin(PI\*TIME)  2초의 주기, 25도의 진폭을 가지는 진동  • 진동에 의해서 장력과 tensional의 위치가 변동됨. |

|  |
| --- |
| #9-2. Toolkit-Chain [Forklift with Roller Chain] |
| [1] 목적  • 체인과 풀리 시스템 (롤러 링크와 롤러를 이용)  • Entity들 사이 자동으로 접촉 생성. |
| [2] 툴킷 Chain 생성  • Chain subsystem 생성  (1) Link connector T (실린더), B (구) 생성  (2) Roller와 Roller 내 Pin 생성 🡪 ground와 각각 fixed  (3) Force탭/Bushing (6방향의 강성과 댐핑계수)  (4) Roller 링크를 정의해 체인 생성 후 시스템에 결합  Chain탭/Link그룹/Roller Link에서 클론체인 생성.  (5) Chain이 필요한 부분에 Assembly를 통해 생성.  (6) 실린더와 Chain마지막 링크를 chain Revolute 함.  (7) 구와 처음 링크릉 Rev로 Connect함.  🡺Connect되는 지점과 강성,댐핑값 조정 가능.  (8) Assembly에 적용된 Bushing force 수정 후 해석.  (9) 해석 중 3초 시간대에서 파일 Extract(추출) |
| [2] Chain과 모델 결합  • 3초 때 모델을 추출함. (클론은 Fixed 안돼서 떨어짐)  fixed안된 클론을 다시 위로 올린 후, 시작  • 모델 결합  (1) 체인의 아래 Connector B(구)랑 본체를 fixed  (2) 체인 위 Connector T(실린더)와 Base를 Fixed  (3) Roller랑 Life랑 Fixed  • Extract 추출의 장점  3초 이후부터 보려 함 🡪3초에서 추출하여 0.25초해석  • 해석: 체인에 의해 운송장치가 들려 올려감.  • 리프르 속도 조절하기  -기존 CMotion은 0이었음. 🡪 수식 줌  -기존 Tra\_Piston\_Cyl motion 없었음. 🡪 motion줌  • 결과 (plot)  높은 숫자 링크일수록 위에 있음. (빨리 통과함)  🡺 링크가 움직이면서 부싱 장력이 급등함.  (높은숫자 링크는 상대적 낮은 감쇠, 높은 접촉강성) |

|  |
| --- |
| #9-3. Toolkit-Gear [Planet Gear] |
| [1] 목적  • spur기어를 사용하는 유성 기어 (하나를 잘못 정렬 시)  • 두 개의 기어쌍 (평기, 헬리컬 기어 사용) 성능 비교  •(display) Setting  -Geometry의 해상도 조정 가능  -Advanced에서 Force display 크기, 색 및 조정 가능  • 기어는 원 🡪 Home에서 카티샨말고 원 좌표쓰면 편리 |
| [2] 유성기어 (단순 기어) 생성  • Gear그룹/Spur아이콘 후, Teeth수 정하고 Generate  • Gear그룹/Int.Spur: 원 안에 기어 이빨있는 거 생김  • Assembly로 Spur기어 끼리 맞물리게 함.  • 생성된 Assembly가 OuterRing(원 안에 이빨 있는 기어) 과 맞물릴 수 있게 계산 후 각도 조절.  • Joint 생성  (1) OuterRing이 안 떨어지게 ground랑 fixed  (2) Gear를 연결해주는 Carrier에 ground랑 fixed  (3) Carrier랑 중심 기어, 맞물리는 기어 3개랑 Rev joint  • 2D Contact 생성  (1) Gear탭/Contact그룹/Cur-Cur  (2) 맞물리는 기어 끼리 총 6개 contact 생성  • Motion추가  (1) Carrier와 Ground사이 Rev joint에 30D\*STEP()추가  (2) 저항 토크 부하 (Rot.Axial를 중앙 기어 Rev joint에) |
| [3] 기어 잘못 맞물리는 경우  • 2D contact을 비활성화 🡪 3D Contact으로 생성.  (2D는 Cur-Cur, 3D는 Solid로 되어있음)  🡺 force display 시, 접촉힘이 3차원으로 나타남.  • 잘못된 모델 만들기  (1)기어를 도는 방향에 수직 방향 축으로 회전(y축)  🡺 시뮬레이션은 정상으로 돌아감  (2) Plot으로 결과 확인  -Y축 우클릭/property에서 y축 스케일 조정가능  -잘못된 기어가 인가하는 토크는 정상 토크와 다르게 굉장히 진폭이 크고 불안정함. (노이즈와 변동이 큼)  -하나의 Plot 파일에서, Window그룹/Add에서 여러 plot을 생성할 수 있음.  -접촉력도 정상기어는 일정적, 잘못정렬된 기어는 주기적으로 크게 튐. |
| [4] 헬리컬 기어  • 소음 감소 및 부드럽게 움직여야 할 때 헬리컬기어 씀.  • Spur기어 (평기어) 생성 후 Assembly  • 같은 크기 헬리컬기어(모듈 값 주의) 생성 후 assembly  🡺 복사된 기어는 Angle이 같은 크기 반대 부호  • motion과 force 추가.  (1) 좌측 평/헬리컬 기어 joint에 회전 motion추가  (2) 우측 평/헬리컬 기어 Rot..Axial 반대 방향 1000  🡺 resist torque임.  • 결과  -헬리컬기어: 사선으로 돼있어 한번에 4점만 접촉해 조용함.  -평기어: 구동토크는 얘가 더 부드러움.  -FZ방향 반력에서, 헬리컬기어는 사선이므로 축 방향 추력이 생김. 따라서 이를 견딜 특수 베어링이 필요함. |

|  |
| --- |
| #9-4. Toolkit-DriveTrain [GearBox] |
| [1] 목적  • 회전축, 기어, 베어링 등 기계 시스템을 각 부품별 재료,  동역학적 특징을 고려해 설계하는 툴킷임.  • 회전축은 FE로 설계, 기어와 베어링은 KISSsoft로 연성 해석함. (기어는 analytic contact필요)  • Gear Box 시스템  (1) shaft, bearing, gear생성  (2) Analytic Gear Contact 생성  • Drive Train 탭이 따로 있음. (Shaft, KISSsoft) |
| [2] Gear Box 모델  • Body 나눠서 import (layer로 숨길 수 있게)  • Drive Train탭 (Shaft/KISSsoft/Contact그룹)  (1) Shaft생성  -section에 값 넣으면 자동으로 정해진 모양이 있음.  -FDR 추가 가능함. (FDR: FE에 강체처럼 joint주기 위함)  🡺 FDR : 축에 기어, 베어링 등의 부재가 연결되려면 핀, 키, 스플라인 등을 써야함. 이런 부재간 연결을 Rigid Element인 FDR로 표현함. 따라서, FDR의 마스터 노드가 생성되는 중심점은 다른 노드보다 큼.  FDR tolerance: Node와 FDR 생성으로 생기는 노드 간격을 무시하는 수치임. (Tolerance가 0.01 인데, Node 간격이 0.009면, 삽입 돼야할 노드가 기존 노드로 변경)  - 기존 축을 FE Shaft 3개로 대체함.  (2) Bearing 생성 (KISS soft)  - Bearing 아이콘/특성 설정 후 생성 (총 6개)  (3) Gear 생성 (KISS soft)  - GearTrain아이콘 (점,점,방향,Dialog)  🡺 기어개수, 각 기어의 이, 넓이, 재료, 기어 종류 선정  -Gear Pair 설정 (Contact Analysis설정임)  - 2개 쌍을 2개 만듦 (기어 총 4개) |
| [3] Joint, Force 생성  • Fixed joint  -땅과 아래 housing, 아래와 위 housing  -Housing과 바깥 베어링/축과 안쪽 베어링/축과 기어  • 축에 Rev joint 달아서 Motion 줌 (Gear box 회전)  • 다른 축 Rev joint에는 rot.axial 힘 만들어서 줌. |
| [4] 해석 결과  • Drive Train에서도 Shaft(축) 에 대한 Contour 봄.  • Scope에서 축 변위에 따른 최대응력 그래프 확인  • Bearing은 Result plot에서, Force로 FM,TM 확인  • Gear도 Result plot에서 에러, 힘 확인 가능. |
| [5] Gear를 KISS soft🡪 Involute analytic contact  • Drive Train탭/Contact그룹/Involute  • 결과: 축, 베어링, 기어 해석 결과에서 차이 거의 없음. |
| [6] Campbell Diagram  • 기어와 축으로 된 다른 모델, Plot에서  Tool탭/Campbell(3D) 클릭  [Analysis설정]  • Input 값 (Tacho, Signal을 넣음)  • 영상의 Frame을 수정함 (수정 안하면 추천 값)  [Plot 설정]  • Graph모양, Order, Line등 여러 설정 가능  • 모양에는 2D, 3D 설정과, Contour legend 설정도 가능 |

|  |
| --- |
| #9-5. Toolkit-EHD [Piston Lubrication] |
| [1] 목적  • |

|  |
| --- |
| #9-6. Toolkit-MTT2D 1 [media transport system] |
| [1] 목적  • |

|  |
| --- |
| #9-7. Toolkit-MTT2D 2 [MTT2D 1+DesignStudy] |
| [1] 목적  • |

|  |
| --- |
| #9-8. Toolkit-MTT2D 3 [MTT2D 1 + IGES import] |
| [1] 목적  • |

|  |
| --- |
| #9-9. Toolkit-MTT3D [Media Transport System 3D] |
| [1] 목적  • |

|  |
| --- |
| #9-10. Toolkit-Track\_LM [Low-monility tracked] |
| [1] 목적  • |

|  |
| --- |
| #9-11. Toolkit-Tire [Driving JTurn] |
| [1] 목적  • |

|  |
| --- |
| #10-1. SPI – Particleworks [Water Sloshing] |
| [1] 목적  • |

|  |
| --- |
| #10-2. SPI – Particleworks [Styler] |
| [1] 목적  • |

|  |
| --- |
| #10-3. SPI – EDEM [Cleated Belt Conveyor] |
| [1] 목적  • |

|  |
| --- |
| #11-1. eTemplate – CreationMode [4WD Loader] |
| [1] 목적  • |

|  |
| --- |
| #11-2. eTemplate – CreationMode [Track LM] |
| [1] 목적  • |

|  |
| --- |
| #11-3. eTemplate – ModificationMode  [Macpherson Strut Design Study] |
| [1] 목적  • |