**보행 밸런싱 증대를 위한 무릎 외골격 조인트의 최적 설계**

조훈호\*† **·** 유정한\* **·** 김윤영\*

\*서울대학교 기계공학과

**Optimal Design of Knee Exoskeleton Joint for Improving Balancing during Gait**

Hunho Cho\*†, Jeonghan Yu\*, Yoon Young Kim\*

\*Dept. of Mechanical Eng., Seoul National Univ.

Progress report 2.

원하는 Gait Motion을 만들어낼 수 있는 4-bar Linkage의 설계를 위한 최적설계기법의 적용 및 이를 통해 얻은 기구에서의 End-effector의 Trajectory 분석

Progress report 1에선 보행 밸런싱 증대를 위한 외골격의 기구 설계를 위한 첫 단계로, 간단한 4절 링크 기구를 Matlab을 이용해 구현하여 이 기구의 Motion을 작동시키며 End-effector의 Trajectory를 얻어보는 Code를 짜보았다. 이후 이 4절 링크 기구의 End-effector의 위치, joint의 위치, 특성 등을 변화시켜보며 4가지 variation에서의 Trajectory를 얻어보며 Gait Motion에서의 Trajectory를 얻어볼 수 있을지 여부에 대해 분석해보았다. 이번 Progress report 2에선 원하는 Path와 거의 일치하는 End-effector Trajectory를 얻을 수 있는 링크 기구를 최적 설계 기법을 이용하여 설계해보고자 한다.

최적 설계(Optimal Design) [1]

최적 설계란, 원하는 기능을 수행할 수 있는 Design을 얻는 데에 있어서 설계 시간을 최소화 시키면서 동시에 최고의 성능을 얻어내는 방법이다. 기본적인 최적설계 기법은, 최초의 Design을 구상한 후, 그 design에 대해서 설계 변수들을 지정해, 각 변수가 해당 제품의 성능에 미치는 영향력을 파악하여 목적함수를 설정하고, 변수들의 제약조건을 고려하여 이 목적함수의 값이 원하는 값에 도달하도록 설계 변수를 수정하여 최적의 변수 조합을 찾아내는 방식으로 진행된다.

본 연구에서는 이 최적 설계 기법을 4-bar Linkage의 초기 Configuration에서의 Link의 위치, 각도에 대한 변수들을 설계 변수로 두고, 각 Joint들에서의 Link간의 연결 관계를 제한조건으로 두며, End-effector Trajectory와 Desired Path 사이의 거리 값을 목적함수로 두어 이 값을 0으로 수렴시키는 방식으로 적용하였다.



Figure 1 4-bar Linkage의 Rigid Body Model

Progress report 1에서 다룬 내용은 최적설계 단계에 들어가기 전에 설계의 가능성을 보기 위한 과정으로, 4-bar Linkage를 Figure 1과 같은 단순한 Rigid Body Model으로 설정하였었다. 또한, 각 Joint에서 연결된 두 링크가 만족해야 하는 특성을 함수 값으로 지정하였고, 이 함수가 각 Step에서 0으로 수렴하도록 하는 방법으로는 Newton-Raphson Method 방법을 사용했었다. 이 과정을 통해 앞에서 언급했던 내용과 같이 Rigid-body bar model 방법을 통해서도 Gait Motion에 가까운 Trajectory를 얻을 수 있음을 관찰할 수 있었다. 최적 설계 알고리즘은 비선형의 목적함수 및 제한조건을 다루는 효과적인 Tool로 사용될 수 있고, 빠른 수렴성을 특징으로 하기 때문에 MMA로 설정하였다.

MMA(Method of Moving Asymptotes)[2]

목적함수 f0(xi)의 minimize를 위한 한가지 방법으로, Newton-Raphson Method와 유사하게 이전 xi

에서의 함수의 값 f(xi)와 Gradient 값 을 통해 다음 iteration의 xi+1값을 계산하는 방법인데, 각 Iteration에서 xi에 대한 Upper Bound Ui와 Lower Bound Li가 존재하여, xi값이 이 경계를 넘지 않고 수렴하여 계산속도가 빠르다. 또, Newton-Raphson Method와는 다음 iteration의 xi+1값을 계산하는데 있어서, f(xi)에서의 접선을 사용하는 것이 아니고, 두 분수함수 와 의 선형결합을 통해 얻어진 함수를 이용한다는 점에 차이가 있다.

또한, Progress report 1에서는 End-effector의 Trajectory가 Gait Motion과 비슷하게 나오는지 여부를 그저 형상의 유사함 정도만 살펴보았는데, 이 Trajectory를 실제 Desired Path에 일치시키는 방법은 크게 2가지 방법이 있다. 한가지 방법은 Desired Path를 선 영역으로 잡아, End-effector의 Trajectory가 이 선 위에 위치하는지 여부를 살피는 방법이고, 다른 방법은 Desired Path도 본 연구에서 얻어내는 End-effector의 Trajectory와 같이 step마다의 점의 집합으로 잡아 각 step마다 이에 해당하는 Path의 점과 Trajectory의 점이 일치하는지를 보는 방법이다. 이 중, 실제 보행 밸런싱 증대를 위한 설계를 위해선 단순히 같은 형상의 Trajectory를 얻는 것에서 그치지 않고, Gait motion에서 시간 진행에 따른 순간순간의 발목의 위치와 Trajectory에서 각 step에서의 End-effector의 위치가 일치할 수 있도록 설계할 수 있어야 한다고 판단했고. 이를 위해서 Desired path 2번째 방법인 Step마다 위치했으면 하는 Point로 잡았다.

이렇게 구상한 Desired Path를 통해 각 step마다의 End-effector Trajectory와 Desired Path 사이의 거리 값을 목적함수로 두어, MMA 방법을 통해 값을 0으로 수렴시키도록 진행하던 중에, Progress 1과 같이 4-bar Linkage를 Rigid Body Model으로 설정할 경우, 각 설계변수가 목적함수에 미치는 sensitivity가 적절히 구해지지 않는 문제가 발생하여, Model을 Figure 2와 같은 3by3 SBM으로 수정하였다.



Figure 2 4-bar Linkage의 3by3 SBM(Spring-connected Block Model)

SBM (Spring-connected Block Model)[3]

SBM 방식은 먼저 도메인을 블록 단위로 나누고, 블록들을 가상의 spring으로 이어, 이 spring들에 강성도를 주어 Linkage Model을 Design하는 것이다. 각 블록들은 초기에 직사각형 모양으로 잡으며, 4-bar Linkage에 기본적으로 사용되는 Revolute Joint와 Prismatic Joint를 구현하기 위하여, 인접하는 블록들과 붙어있는 꼭지점에서의 Revolute Joint구현용 2개 spring, 붙어있는 변에서의 Prismatic Joint 구현용 2개의 직교하는 spring(Prismatic Joint의 방향성을 설정해주기 위해 2개가 필요하다!), 총 4개의 spring으로 연결되어 있으며, Ground와도 각 꼭짓점마다의 Ground Revolute Joint 구현용 4개의 spring, 블록 자체의 Ground Prismatic Joint 구현용 2개 총 6개의 spring으로 연결되어 있다. 이때 강성도에 대한 설정을 통해 각 Block이 Rigid하게 연결되는지, R joint로 연결되는지, P joint로 연결되는지가 결정된다. 또, 각 Block은 초기에 단위 직사각형 꼴로 잡는데, 각 Node의 초기 위치 변수, P joint 연결부의 각도 변수를 설계변수로 잡았다. 최적설계에 있어서 SBM을 해석하는 방법은 Desired Path에서의 End-effector의 진행에 반대되는 방향으로 힘을 가하며, Work의 input 값과 Output으로 얻어지는 값을 통해 Efficiency η 값을 1로 만드는 해석방법을 사용한다.

4bar linkage의 SBM Model 구현에 필요한 최소한의 Block 수는 2by2지만, 본 연구에서는 4 bar linkage로의 최적 설계가 한계가 있을 수 있음을 감안하여, 추후 6 bar linkage 이상으로의 확장이 가능하도록 3by3 SBM Model을 사용하였다. 또, SBM에서의 Block 연결 spring들의 강성도에 대한 설정은 설계 변수로 두지 않고 Topology 값으로 지정을 해주었다. 설계 변수는 3by3 Block에서 나오는 16개의 Node의 초기 위치 값 (x, y)를 통해 32개의 Shape Dof ξx를 얻었으며, Connecting Prismatic Joint 12개와 Ground Prismatic Joint 9개를 통해 각 Prismatic Joint의 각도 변수로 21개의 Ang Dof ξA 를 얻었다. 제한조건은 End-effector의 Trajectory와 Desired Path 사이의 거리로부터 얻어지는 ψ의 값이 지정한 error 값보다 작음을 만족시키면서 efficiency η 가 1로 수렴하도록 MMA 방식을 사용하여 최적의 설계변수를 얻는 방법을 사용하였다.

이렇게 설정한 4-bar Linkage Design의 최적설계기법을 본 Progress Report 2에서는 2가지 Desired Path를 설정하여 각 Path에 대해 적용해보았다. Figure 3에 나와있는 Path 1은 경로 상에 Crunode (절점)이 있는 Closed Loop Path로 설정하였다. 하지만, Gait Motion은 사람의 보행 시에 발목이 그리는 Path를 얻어낸 것이므로, 무릎 Exoskeleton의 설계에 있어서는 꼭 완전한 Closed Loop Path 구현을 요구하지 않을 수 있다 생각되어 Desired Path 2를 Open Loop Path로 설정하여 이에 대해서도 최적설계를 해보기로 하였다.

 

Figure 3 Desired Path 1 (Closed Loop Path) Figure 4 Desired Path 2 (Open Loop Path)



Figure 5 결과창 예시

이 최적설계를 구현할 수 있는 Matlab 코드를 구상한 후, 얻은 결과창은 Figure 5에 나와있는 예시와 같이 나온다. 왼편의 그래프를 통해선 특정 Iteration 내에서 Input의 Step마다 4-bar Linkage가 어떤 Configuration을 갖고 있는지를 확인할 수 있고, 오른쪽 가장 위의 그래프를 통해선 이 Iteration에서의 End-effector Trajectory(빨간색)과 Desired Path(흰 점)이 얼마나 일치하는지를 시각적으로 확인할 수 있다. 오른편 중앙의 두 막대그래프는 특정 Iteration에서의 설계변수 ξ에 대한 값으로, 위에서부터 순서대로 각각 Angle Dof ξA의 값과 Shape Dof ξx의 값을 나타낸다. 이때 Shape Dof는 각 Node의 위치 변수에 의한 항으로, 초기 0.5의 값에서 시작하여 모든 32개의 값이 수정되는데 반해, Angle Dof는 Prismatic Joint의 각도에 대한 term이어서, Prismatic Joint가 실제로 설정된 term의 값만 변하게 된다. 0.5의 값에서 변하지 않는 것은 Prismatic Joint로 설정되지 않은 Angle Dof라고 생각할 수 있다. 오른편 가장 아래의 그래프는 각 Iteration에서의 Error 값과 Efficiency η의 값을 기록한 것으로, Error는 빨간 선으로 표기되어 궁극적으로 0으로 수렴하는 것을 목표로 하는 값이며, Efficiency는 파란 선으로 표기되어 궁극적으로 1으로 수렴하는 것을 목표로 하는 값이다.

Desired Path 1에 대한 최적설계 분석

Desired Path 1은 앞서 언급했던 것처럼 Figure 3에 나와있는 Closed Loop Path이며, 이때 사용한 초기 Topology는 Figure 6에 나와있는 것과 같다.



Figure 6 Desired Path 1에서의 Topology (RRRR)

우선 Block은 Rigid하게 묶인 것들을 고려하면, Block 1, Block 2, (Block 3, 6, 9), (Block 4, 5, 7, 8)로 이루어져 있으며, 이중 Block 2는 바닥에 고정되어 있으면서 다른 Block 군집과의 연결이 없는 놈으로 무시하면 총 4개의 Link(바닥 포함)가 있는 4-bar Linkage임을 알 수 있다. 각 Link의 연결관계는, 우선 Block 1은 Node 1에서 Ground Revolute Joint로 고정되어 있으며, Node 5에서 (Block 4, 5, 7, 8)와 Revolute Joint로 연결되어 있다. 또 (Block 3, 6, 9), (Block 4, 5, 7, 8)은 Node 15에서 Revolute Joint로 연결되어 있으며, (Block 3, 6, 9)는 Node 4에서 Ground Revolute Joint로 고정되어 있다. 즉, 이 Topology는 4-bar RRRR Linkage임을 알 수 있다.

이 Topology를 만족시키는 SBM을 이용하여 최적설계를 진행한 결과를 Iteration 0(설계변수 초기값에 의한 결과), Iteration 5, Iteration 10, Iteration 20, Iteration 100에 대해서 데이터를 얻어 Figure 7~11에 기록하였다.



Figure 7 Desired Path 1의 4-bar Linkage 최적설계 – Iteration 0



Figure 8 Desired Path 1의 4-bar Linkage 최적설계 – Iteration 5



Figure 9 Desired Path 1의 4-bar Linkage 최적설계 – Iteration 10

Figure 10 Desired Path 1의 4-bar Linkage 최적설계 – Iteration 20



Figure 11 Desired Path 1의 4-bar Linkage 최적설계 – Iteration 100

위의 결과 그래프들을 통해, 본 최적설계 방법을 통해서 End-effector Trajectory가 Gait Motion과 유사한 형상을 갖는 Desired Path 1에 Iteration을 거듭할수록 근접하게 변해감을 확인할 수 있다. 이를 통해서 본 최적설계 방법이 꽤 효과적임을 알 수 있다. 추가로 Figure 9~11에 나타난 Iteration 10~100의 과정을 보면, Error값이 작은 값에서 유지가 되며, 크게 형상이나 Path에의 접근성이 더 좋아지지 않음을 볼 수 있다. 이 Error값이 설정한 값보다 작은 값으로 나온 것이면 본 최적설계의Iteration이 끝나게 되도록 Matlab 코드를 작성하였음을 생각해볼 때, 이는 본 최적설계 방법을 통한 Design 수정이 더 이상의 효과적인 수렴을 하지 못하게 된 것이라고 생각해볼 수 있다. 즉, 본 방법을 통해 4-bar Linkage를 Design했을 경우, Iteration을 100보다 크게 충분히 늘리더라도 Figure 11에 나와있는 정도로의 Gait motion에 근접한 Trajectory를 얻어낼 수는 있지만, 이정도 결과가 최적설계의 결과로 얻어지게 될 것임을 알 수 있다.

Desired Path 2에 대한 최적설계 분석

Desired Path 2은 앞서 언급했던 것처럼 Figure 4에 나와있는 Open Loop Path이며, 이때 사용한 초기 Topology는 Figure 12에 나와있는 것과 같다.

우선 Block은 Desired Path 2에서와 같게 Block 1, Block 2, (Block 3, 6, 9), (Block 4, 5, 7, 8)로 이루어져 있으며, Joint의 연결 상태도 거의 같은 Topology를 지닌다. 차이점은 (Block 3, 6, 9), (Block 4, 5, 7, 8)간의 연결이 기존의 Node 15에서의 Revolute Joint 연결이 아닌, Block 8, Block 9 사이의 Prismatic Joint로 있다는 점에 차이가 있다. 이를 통해 Desired Path 2를 만족하는 설계에서의 Topology는 4-bar RRRP Linkage임을 알 수 있다.

Desired path 1에서와 마찬가지로 이 Topology를 만족시키는 SBM을 이용하여 최적설계를 진행한 결과를 Iteration 0, Iteration 5, Iteration 10, Iteration 20, Iteration 50, Iteration 100에 대해서 데이터를 얻어 Figure 13~18에 기록하였다.



Figure 12 Desired Path 2에서의 Topology (RRRP)



Figure 13 Desired Path 2의 4-bar Linkage 최적설계 – Iteration 0



Figure 14 Desired Path 2의 4-bar Linkage 최적설계 – Iteration 5



Figure 15 Desired Path 2의 4-bar Linkage 최적설계 – Iteration 10



Figure 16 Desired Path 2의 4-bar Linkage 최적설계 – Iteration 20



Figure 17 Desired Path 2의 4-bar Linkage 최적설계 – Iteration 50



Figure 18 Desired Path 2의 4-bar Linkage 최적설계 – Iteration 100

위의 결과 그래프들을 통해, 본 최적설계 방법을 통해서 End-effector Trajectory가 임의의 Open Loop Path와 유사한 형상을 갖는 Desired Path 2에 Iteration을 거듭할수록 근접하게 변해감을 확인할 수 있다. Desired Path 1에 비해서는 Iteration의 진행에 있어 수렴 속도가 느리긴 하지만, error값이 0 근처에서 수렴한 후의 최종 Trajectory를 보면 Desired Path 1에서의 결과에 비해서 Desired Path에 매우 근접한 결과를 얻어냈음을 알 수 있다. 즉, Gait Motion의 일부나, 보행시에 고관절에 대한 무릎의 상대운동 경로만을 얻었을 경우 나타날 수 있는 Open Loop Path에 대해서는 이에 매우 근접한 End-effector Trajectory를 갖는 4-bar Linkage를 본 연구의 최적설계 방법을 통해 구현할 수 있다는 것을 알 수 있다. 이를 통해 앞으로의 무릎 Exoskeleton의 설계에 본 최적설계 기법이 효과적으로 사용될 수 있을 것으로 기대할 수 있다.

# 참고문헌

[1] "최적화(Optimization)와 최적설계(Optimal design)." *Altair Engineering*. 2020년 6월 17일 수정, 2020년 12월 9일 접속, <https://blog.altair.co.kr/65387>.

[2] Krister Svanberg, 1987, “The Method of Moving Asymptotes – A New Method for Structural Optimization”, *International Journal for Numerical Methods in Engineering*, VOL. 24, 359-373

[3] Seok Won Kang, Yoon Young Kim , 2018, “Unified topology and joint types optimization of general planar linkage mechanisms”, *Structural and Multidisciplinary Optimization*, , VOL. 57, 1955-1983