(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0033054
(43) 공개일자 2025년03월07일

(51) 국제특허분류(Int. Cl.)
B25J 9/16 (2006.01)
(52) CPC특허분류
B25J 9/1656 (2013.01)
B25J 9/1615 (2013.01)
(21) 출원번호 10-2024-0116087
(22) 출원일자 2024년08월28일
심사청구일자 2024년08월28일
(30) 우선권주장
1020230112631 2023년08월28일 대한민국(KR)

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
윤태림
인천광역시 남동구 남동대로799번길 34
최성준
경기도 하남시 위례대로6길 20, 7107호 403호
(74) 대리인
김등용

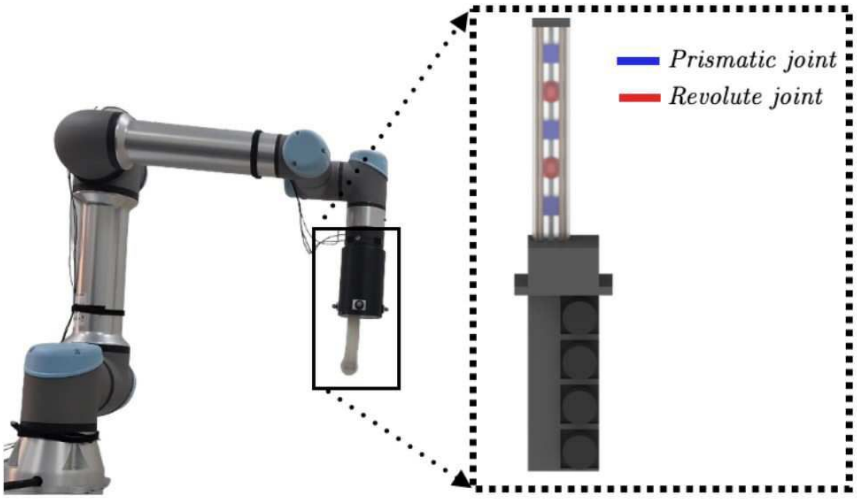
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 **소프트 로봇과 강체 로봇을 포함하는 하이브리드 로봇과 이의 제어 방법**

(57) 요 약

소프트 로봇과 강체 로봇을 포함하는 하이브리드 로봇과 이의 제어 방법 및 소프트 로봇의 트래킹 포인트(tracking point)의 위치 추정 방법이 개시된다. 상기 소프트 로봇의 트래킹 포인트의 위치 추정 방법은 적어도 프로세서(processor)를 포함하는 컴퓨팅 장치에 의해 수행되고, 상기 소프트 로봇을, 가상의 복수의 관절들이 포함된 수도 강체(Pseudo Rigid Body, PRB)로 모델링하는 단계, 및 학습 데이터를 이용하여 학습함으로써, 액추에이션 명령에 대응하는 관절 값(joint value)을 예측하는 예측 함수의 파라미터와 예측된 관절 값에 대응하는 상기 트래킹 포인트의 위치를 추정하는 강체 기구학의 파라미터를 탐색하는 단계를 포함하고, 상기 트래킹 포인트는 상기 소프트 로봇의 말단에 위치한다.

대 표 도 - 도1



(52) CPC특허분류
B25J 9/163 (2013.01)
B25J 9/1653 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711193973
과제번호 2019-0-00079-005
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 인공지능대학원지원(고려대학교)
과제수행기관명 고려대학교산학협력단
연구기간 2023.01.01 ~ 2023.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711181287
과제번호 2021R1A4A3033149
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 집단연구지원
연구과제명 차세대 로봇 설계 및 제어를 위한 인공지능 플랫폼 연구실
과제수행기관명 고려대학교
연구기간 2021.06.01 ~ 2024.02.29

명 세 서

청구범위

청구항 1

적어도 프로세서(processor)를 포함하는 컴퓨팅 장치에 의해 수행되는, 소프트 로봇의 트래킹 포인트(tracking point)의 위치 추정 방법에 있어서,

상기 소프트 로봇을, 가상의 복수의 관절들이 포함된 수도 강체(Pseudo Rigid Body, PRB)로 모델링하는 단계; 및

학습 데이터를 이용하여 학습함으로써, 액추에이션 명령에 대응하는 관절 값(joint value)을 예측하는 예측 함수의 파라미터와 예측된 관절 값에 대응하는 상기 트래킹 포인트의 위치를 추정하는 강체 기구학의 파라미터를 탐색하는 단계를 포함하고,

상기 트래킹 포인트는 상기 소프트 로봇의 말단에 위치하는,

소프트 로봇의 트래킹 포인트의 위치 추정 방법.

청구항 2

제1항에 있어서,

제1항에 있어서,

상기 트래킹 포인트는 상기 가상의 복수의 관절들 중 가장 말단에 위치한 관절을 나타내는,

소프트 로봇의 트래킹 포인트의 위치 추정 방법.

청구항 3

제2항에 있어서,

상기 가상의 복수의 관절들 각각은 선형 관절, 회전 관절, 고정 관절, 및 항들 관절 중 어느 하나이고,

상기 상기 트래킹 포인트에 대응하는 관절은 항들 관절인,

소프트 로봇의 트래킹 포인트의 위치 추정 방법.

청구항 4

제3항에 있어서,

상기 학습 데이터는 액추에이션 명령들과 상기 액추에이션 명령들 각각에 대응하는 상기 소프트 로봇의 실제 위치를 포함하는,

소프트 로봇의 트래킹 포인트의 위치 추정 방법.

청구항 5

제4항에 있어서,

제1 액추에이션 명령을 학습된 예측 함수에 입력하여, 대응되는 관절 값을 도출하는 단계; 및

상기 대응되는 관절 값을 학습된 강체 기구학에 입력하여, 대응되는 관절 위치를 도출하는 단계를 더 포함하는,

소프트 로봇의 트래킹 포인트의 위치 추정 방법.

청구항 6

제5항에 있어서,

상기 탐색하는 단계는 알고리즘 1을 수행하는,

소프트 로봇의 트래킹 포인트의 위치 추정 방법.

[알고리즘 1]

Algorithm 1 Kinematic Architecture Search

Input: $(\mathbf{a}, \mathbf{x}_{\text{gt}})$
Output: $(\hat{\theta}_g, \hat{\theta}_s, \hat{\theta}_t)$

$R_{\text{min}} \leftarrow \text{inf}$
for i in n_p **do**
  $\theta_t \leftarrow$ Random_Initialize
  $\theta_t^1[i] \leftarrow \theta_t$
end for
for $G \leftarrow 1$ to G_{END} **do**
  for i in n_p **do** ▷ continuous optimization
    $\theta_t \leftarrow \theta_t^G[i]$
    $(\theta_s, \theta_g) \leftarrow \text{Initalize}(\mathbf{x}_{\text{gt}}^{\text{init}})$
    $\mathbf{x} \leftarrow \text{FK}_r(g(a;\theta_g);\theta_s,\theta_s)$
    $\mathcal{L} \leftarrow \|\mathbf{x}_{\text{gt}} - \mathbf{x}_{\text{E}}\|_1 + \lambda_{\text{reg}} \sum_{i=0}^{n_t-1} d_c(\mathbf{x}_{\text{gt}}[k(i)+1] - \mathbf{x}_{\text{gt}}[k(i)], \mathbf{x}[i+1] - \mathbf{x}[i])$
    $R[i] \leftarrow \min_{p_s, p_\theta} \mathcal{L}$
    if $R[i] < R_{\text{min}}$ **then**
      $(\hat{\theta}_g, \hat{\theta}_s, \hat{\theta}_t) \leftarrow (\theta_g, \theta_s, \theta_t)$
    end if
  end for
  $\theta_t^{G+1} \leftarrow \text{Mutation}(\theta_t^G, R)$ ▷ discrete optimization
end for

여기서, $\hat{\theta}_g$는 상기 예측 함수의 파라미터, $\hat{\theta}_s, \hat{\theta}_t$는 상기 강체 기구학의 파라미터.

청구항 7

제6항에 있어서,

상기 소프트 로봇은 강체 팔(rigid arm)의 말단에 부착된 소프트 손가락(soft finger)인,

소프트 로봇의 트래킹 포인트의 위치 추정 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 소프트 로봇(soft robot)과 강체 로봇(rigid robot)을 동시에 가지고 있는 하이브리드 로봇(hybrid robot)의 기구학적 특징을 모델링하는 물리기반 신경망과 그것을 이용한 제어 방법에 관한 것이다.

배 경 기 술

[0002] 더 많은 로봇(예컨대, 협동 로봇)이 점진적으로 우리의 일상 생활에 침투함에 따라, 인간의 안전 보장에 대한 관심이 높아지고 있다. 대부분의 기존 로봇은 잘 구축된 강체 모델(rigid body model)에 기반하고, 이는 효과적이고 효율적인 제어 기법을 제공한다. 그러나, 제한적인 감지 능력이나 액추에이터의 오작동으로 인한 위험 또한 존재한다. 반면에, 소프트 로봇은 강체 로봇에 비해 보다 더 안전하다. 그러나, 상태 예측(state prediction), 제어를 위한 모델링, 및 식별(identification)이 어렵다는 문제점이 있다. 본 발명에서는, 넓은 클래스의 소프트 로봇에 적용이 가능한 효과적이고 효율적인 모델 식별 기법(model identification method)과 제안 모델을 이용한 제어 기법을 제안한다.

[0003] 소프트 로봇의 기구학(kinematics)을 모델링하는 것은 정밀한 제어에 필수적이지만, 자유도의 무한성으로 인하여 여려움이 따른다. 기존의 연구는 크게 모델 기반 접근법(Model-based approach)과 블랙박스 접근법(Black-box approach)으로 분류될 수 있다. 모델 기반 접근법은 소프트 로봇의 기구학을 에너지 보존(energy

preservation)이나 기하학적 관계(geometric relationship)와 같은 물리 원칙으로 묘사한다. 모델 기반 접근법이 강건성(robustness)과 설명가능성(explainability)을 제공하지만, 축소된 차원 모델링에 의한 과소적합에 빠지기 쉽다. 반면에, 대부분의 블랙박스 접근법은 가우시안 프로세스(Gaussian processes)와 같은 비선형 회귀 모델(nonlinear regression models)을 이용하고 힘든 모델링 프로세스를 건너뛸 수 있다. 그러나, 블랙박스 접근법은 설명가능성이 부족하고, 과적합으로 인하여 낮은 일반화 성능을 보인다.

[0004] 물리학 지식을 블랙박스 모델에 결합하는 물리 정보 모델링(physics-informed modeling)에서의 최근 진보는 정확하고 설명가능한 프레임워크를 생성함에 있어서 유망한 성과를 보였다. 유사하게, 본 발명에서도 물리학 지식을 소프트 로봇의 기구학적 행동(kinematic behavior)에 대한 모델링과 식별에 이용하고자 한다. 이전의 연구와는 상이하게, 본 발명에서는, 소프트 로봇의 기저에 있는 기구학적 구조(kinematic structure)를 모델링하기 위해 강체 기구학(rigid body kinematics)을 채용한다. 내재된 기구학에 대한 모델링은 소프트 로봇의 식별을 향상시킬 뿐만 아니라 강체 로봇 제어에 대한 기법의 활용을 가능케 한다. 강체 로봇에 대한 수십년 동안의 활발한 연구가 있었지만, 강체 기구학을 위해 개발된 기법들이 직접적으로 소프트 로봇에 이용되지는 않는다.

[0005] 본 발명에서는, 소프트 로봇을 모델링, 식별, 및 제어하기 위해서 소프트 로봇을 강체 로봇과 연관시키는 기구학 정보 신경망(Kinematics-Informed Neural Network, KINN)을 제안한다. 또한, 본 발명에서는, KINN의 기구학적 파라미터들(kinematic parameters)를 찾기 위한 최적화 전략으로 기구학적 아키텍처 탐색(Kinematic Architecture Search, KAS)을 제안한다. KINN는, 도 1에 도시된 바와 같이, (내재된) 기구학적 구조(수도 강체(Pseudo Rigid Body, PRB)라 명명될 수 있음)를 모델링한다. 이러한 모델링 접근법은, 야코비안(Jacobian)을 이용하는 역기구학(inverse kinematics)과 같은 강체 로봇의 제어 기법의 이용을 가능케 한다.

[0006] 제안하는 프레임워크는 소프트 로봇과 강체 로봇으로 구성된 하이브리드 시스템 또는 하이브리드 로봇을 제어하기 위해 효과적으로 이용될 수 있다. 하이브리드 시스템은 소프트 로봇과 강체 로봇 모두의 장점을 이용하는 것을 목표로 한다. 예컨대, 강체 팔(rigid arm)은 확장된 작업공간을 갖는 소프트 손가락(soft fingers)를 제공할 수 있고, 이는 인간 또는 객체와의 사이에서 다목적 그리고 안전한 인터랙션을 가능하게 한다. 그러나, 하이브리드 시스템을 제어하기 위해 강체 부분과 소프트 부분에 대한 개별적인 제어를 요구할 수 있으며, 이는 비효율적 제어를 초래한다. 대신에, 본 발명에서 제안하는 제어 프레임워크는, 소프트 로봇을 PRB(수도 강체)로 모델링하고 강체 로봇을 증강(augmenting)함으로써, 하이브리드 시스템에 대한 전체적인 제어를 제공한다.

[0007] 본 발명에서 제안하는 방법은 광범위한 소프트 로봇에 적용될 수 있으므로, 이를 공압식 및 건식 액추에이터가 장착된(pneumatic and tendon-actuated) 소프트로봇 모두에 테스트한다. 구체적으로, 제안된 모델 식별의 성능을 시뮬레이션 환경과 실제 환경에서 평가한다. 또한, 수집된 데이터 영역 외부에 있는 엔드-이펙터(end-effector) 궤적(trajectories)을 추적함으로써 제어 방법의 강인성을 입증한다. 또한, 문자 쓰기(painting letters)와 병 열기(opening the bottles)와 같은 하이브리드 시스템의 복잡한 제어 작업들을 시연한다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 대한민국 공개특허 제2023-0108902호 (2023.07.19. 공개)

(특허문헌 0002) 대한민국 공개특허 제2022-0051005호 (2022.04.025. 공개)

(특허문헌 0003) 대한민국 등록특허 제2005323호 (2019.07.30. 공고)

(특허문헌 0004) 대한민국 등록특허 제2275900호 (2021.07.12. 공고)

발명의 내용

해결하려는 과제

[0009] 본 발명이 이루고자 하는 기술적인 과제는 소프트 로봇과 강체 로봇을 포함하는 하이브리드 로봇과 이의 제어 방법을 제공하는 것이다.

과제의 해결 수단

[0010] 본 발명의 일 실시예에 따른, 소프트 로봇의 트래킹 포인트(tracking point)의 위치 추정 방법은 적어도 프로세서(processor)를 포함하는 컴퓨팅 장치에 의해 수행되고, 상기 소프트 로봇을, 가상의 복수의 관절들이 포함된 수도 강체(Pseudo Rigid Body, PRB)로 모델링하는 단계, 및 학습 데이터를 이용하여 학습함으로써, 액추에이션 명령에 대응하는 관절 값(joint value)을 예측하는 예측 함수의 파라미터와 예측된 관절 값에 대응하는 상기 트래킹 포인트의 위치를 추정하는 강체 기구학의 파라미터를 탐색하는 단계를 포함하고, 상기 트래킹 포인트는 상기 소프트 로봇의 말단에 위치한다.

발명의 효과

[0011] 본 발명의 실시예에 따른 소프트 로봇과 강체 로봇을 포함하는 하이브리드 로봇과 이의 제어 방법에 의할 경우, 하이브리드 로봇을 효과적으로 제어할 수 있으며, 사람과의 안전한 상호작용이 가능한 효과가 있다.

[0012] 또한, 하이브리드 로봇의 제어를 통일된 방법으로 수행할 수 있다.

[0013] 또한, 링크와 관절로 이루어진 물리적 특성을 이용해 신경망의 구조를 생성함으로써, 데이터에 대한 의존성을 감소시킬 수 있다.

도면의 간단한 설명

[0014] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 소프트 로봇의 내재된 구조로 가정되는 수도 강체(Pseudo Rigid Body, PRB)를 도시한다.

도 2는 (a) 제안 기법(KINN)의 전방 기구학(forward kinematics), (b) 회전 관절(revolute joint)과 선형 관절(prismatic joint)로 구성되는 강체 로봇의 강체 기구학(rigid-body kinematics)의 파라미터들, 그리고 (c) 소프트 로봇 손가락와 이에 대응하는 수도 강체(PRB)를 도시한다.

도 3은 기구적 아키텍처 탐색(Kinematic Architecture Search, KAS)의 프로세스를 도시한다.

도 4는 (a) 건식 소프트 로봇(tendon-driven soft robot)과 관련한 굽힘 방향(bending direction) $\mathbf{u}$, 굽힘 거리 d, 및 로봇의 위치 $\mathbf{x}_{\mathrm{E}}$, (b) $\hat{\mathbf{a}}$를 획득하고 수렴하기까지 $\hat{\mathbf{a}}$를 업데이트함으로써, 주어진 타겟, $\mathbf{u}$, $\mathbf{x}_{\mathrm{tar}}$, 및 d_{tar}에 대해 역 기구학(inverse kinematics)을 풀이하는 과정을 도시한다.

도 5는 ABAQUS에서 쌍을 이룬 데이터 $(\mathbf{a}, \mathbf{x}_{\mathrm{gt}})$를 수집하기 위한, 공압식 소프트 로봇의 구성으로, (a) 소프트 로봇의 전면도와 (b) 소프트 로봇의 단면도를 도시한다.

도 6은 (a) 궤적 추적 결과와 (b) 그라디언트 필드(gradient field)를 도시한다.

도 7은 실제 건식 로봇의 개요를 도시한다.

도 8은 병뚜껑을 열기 위한 프리미티브를 도시한다.

도 9는 (a) 굽힘 방향을 일정하게 유지하면서, 주어진 2 개의 타겟 d_1, d_2 사이에서 보간하기 위해 굽힘 거리 d를 제어하는 선형 스트로크, (b) 굽힘 거리를 일정하게 유지하면서, 굽힘 방향 u를 제어하여 원형 궤적에 접하도록 정렬하는 원형 스트로크, (c) 실제 환경에서 페인팅하는 소프트 로봇 핑거와 UR-5e로 구성된 하이브리드 시스템, 및 (d) 제안하는 제어 프레임워크와 하이브리드 시스템으로 페인팅된 문자를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0015] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시예들에 한정되지 않는다.

[0016] 본 발명의 개념에 따른 실시예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0017] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어

들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.

[0018] 어떤 구성 요소가 다른 구성 요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성 요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0019] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0020] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0021] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시예들을 상세히 설명한다. 그러나, 특허출원의 범위가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

[0022] 또한, 이하에서 설명되는 기법들은 적어도 프로세서(processor) 및/또는 메모리(memory)를 포함하는 컴퓨팅 장치에 의해 수행될 수 있다. 컴퓨팅 장치는 PC(Personal Computer), 서버(server), 랩탑 컴퓨터, 태블릿 PC 등을 포함한다. 또한, 컴퓨팅 장치는 로봇을 제어하기 위해 별도로 구현된 장치나 로봇의 일부 구성으로 구현될 수도 있다.

[0023] **소프트 로봇 모델 식별**

[0024] A. 문제 공식화

[0025] 이하에서는, 소프트 로봇의 기구학적 모델(kinematic model)을 학습하는 문제에 집중한다. 특히, 효율적인 학습을 위한 강체 기구학(rigid-body kinematics)을 활용하는 것을 목표로 한다. 그러나, 강체 기구학은 소프트 로봇의 복잡한 행동(behaviors)으로 인하여 소프트 로봇에 이용될 수 없다. 이러한 제한을 극복하기 위하여, 소프트 로봇의 액추에이션 명령(actuation commands)을 수도 강체(Pseudo Rigid Body, PRB)의 대응되는 관절 위치에 연관시키는 KINN(Kinematics-Informed Neural Network)를 이용함으로써, 도 1에 도시된 바와 같이 수도 강체(PRB)로 명명되는 내재된 기구학적 구조(kinematics structure)를 모델링한다. 도 1에서, 빨간색 구(red spheres)는 회전 관절(revolute joints)을, 파란색 직육면제(blue cuboids)는 선형 관절(prismatic joints)을 나타낸다. 즉, 본 발명에서는 소프트 로봇을 가상의 복수의 관절들을 갖는 수도 강체(PRB)로 모델링한다. 이때, 가상의 복수의 관절들은 미리 정해진 규칙에 따라, 또는 미리 정해진 거리에 따라 배치될 수 있다. 예컨대, 복수의 관절들은 미리 정해진 거리만큼 이격되도록 배치되거나 미리 정해진 지점에 배치되도록 모델링될 수 있다.

[0026] 강체 기구학은, 기구학적 구조와 관절 값(joint values)이 알려져 있을 때, 로봇을 제어하기 위해 효과적으로 이용될 수 있다. 그러나, 소프트 로봇의 경우, 내재된 기구학적 구조와 대응되는 관절 값들은 알려지지 않는다. 따라서, 본 발명에서는, 수도 강체(PRB)의 기구학적 파라미터들을 식별하기 위해 기구적 아키텍처 탐색(Kinematic Architecture Search, KAS)을 수행하고, 강체 제어의 잘 구축된 필드(well-established field)를 활용한다. 또한, 기구적 아키텍처 탐색(KAS)은, $\mathbf{a}$로 표기되는 액추에이션 명령(actuation command)이 주어졌을 때, 수도 강체(PRB)의 관절 값들 $\mathbf{q}$를 예측하는 함수(g로 표기됨)에 대한 탐색을 포함한다. $\mathbf{a}$, g, 및 $\mathbf{q}$의 관계는 다음 수학식과 같다.

[0027] $$\mathbf{q} = g(\mathbf{a})$$

[0028] $\mathbf{a} \in \mathbb{R}^{n_a}$에 의해 표현되는 n_a 개의 액추에이션 명령들과 소프트 로봇에 부착된 n_j 개의 트래킹 포인트들(tracking points)에 의해 동작하는 소프트 로봇을 고려하자. 로봇의 위치는, 준정적 상태(quasi-static conditions) 하에서, 트래킹 포인트들의 위치를 나타낸다. 트래킹 포인트의 실제 위치(true position)는 $\mathbf{x}_{\text{gt}} \in \mathbb{R}^{n_j \times 3}$로 표기된다. KINN은, 액추에이션 명령들 $\mathbf{a}$를 추정된 위치($\mathbf{x}_{\text{E}}$로 표기함)에 맵핑하기 위하여, FK로 표기되는 전방 기구학(forward kinematics)을 채용한다. Θ로 표기된 기구학적 파라미터들에 대하여, 전방 기구학 FK는 아래 수학식과 같이 쓸 수 있다.

[0029] $$\mathbf{x}_{\text{E}} = \text{FK}(\mathbf{a}; \theta)$$

[0030] 모델 식별 프로세스는, 거리 측정 $d(\cdot, \cdot)$에 의해 정의되는 비용 함수를 최소화하는 최적의 Θ 값들을 찾는 것을 포함한다. 여기서, 거리 측정의 결과는 실제 위치 $\mathbf{x}_{\text{gt}} \in \mathbb{R}^{n_j \times 3}$와 추정된 위치 $\mathbf{x}_{\text{E}}$ 사이의 거리를 의미한다.

[0031] $$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{(\mathbf{a}, \mathbf{x}_{\text{gt}}) \in D}[d(\mathbf{x}_{\text{gt}}, \text{FK}(\mathbf{a}; \theta))]$$

[0032] 상기 수학식에서, D는 쌍을 이루는(paired) 데이터 $(\mathbf{a}, \mathbf{x}_{\text{gt}})$의 집합이다.

[0033] 기구학적 파라미터들 Θ에 대한 탐색은, 연속 변수(continuous variable)와 이산 변수(discrete variable) 모두의 존재로 인하여, 쉽지 않은 문제이다. 수도 강체(PRB)의 오프셋과 축(axis)을 θ_s로, 함수 g의 파리미터들을 θ_g로 표기하자. θ_s와 θ_g 모두 연속 변수이기 때문에, 그라디언트-기반 기법(gradient-based methods)을 이용하여 최적화될 수 있다. 반면에, θ_t로 표기되는 수도 강체(PRB)의 관절 타입과 같은 이산 변수는, 미분 불가능하기 때문에, 특별한 주의가 필요하다. 요컨대, 제안하는 기구적 아키텍처 탐색(KAS)는 수학식 1과 같이 공식화될 수 있다.

[0034] [수학식 1]

[0035] $$(\hat{\theta}_{\mathbf{g}}, \hat{\theta}_{\mathbf{s}}, \hat{\theta}_{\mathbf{t}}) = \arg\min_{(\theta_g, \theta_s, \theta_t)} \mathbb{E}_{D \in (\mathbf{a}, \mathbf{x}_{\text{gt}})}[d(\mathbf{x}_{\text{gt}}, \text{FK}(\mathbf{a}; \theta_g, \theta_s, \theta_t))]$$

[0036] 수학식 1에 사용된 용어들에 대한 설명은 표 1와 같다.

[0037] [표 1]

[0038]

Symbols	Description
FK	Forward kinematics of KINN
FK_r	Forward kinematics of Rigid body
θ_g	Parameters of the mapping g
θ_s	Structural parameters including joint axis and offset.
θ_t	Sequence of joint types of the PRB
n_a	Number of actuation of the robot
n_j	Number of tracking points of the robot
$n_t (\geq n_j)$	Number of joints in the PRB
$\mathbf{a} \in \mathbb{R}^{n_a}$	Actuation command of the robot
$\mathbf{x}_{\text{gt}} \in \mathbb{R}^{n_j \times 3}$	The true position of the robot
$\mathbf{x}_{\text{E}} \in \mathbb{R}^{n_j \times 3}$	The position of the tracking joints in the PRB
$\mathbf{x} \in \mathbb{R}^{n_t \times 3}$	The position of the joints in the PRB
$\mathbf{q} \in \mathbb{R}^{n_t}$	The joint values of the PRB
g	Mapping from actuation $\mathbf{a}$ to joint values $\mathbf{q}$

[0039] **B. 기구학 정보 신경망(Kinematics-Informed Neural Network)**

[0040] 이하에서는, 강체 기구학을 신경망에 병합하는(incorporates) KINN(Kinematics-Informed Neural Network)을 설명한다. 특히, 선형 관절과 회전 관절에 집중한다. 선형 관절은 $'P'$로 표기되며 선형 관절의 관절 값들 q는

$S \in \mathbb{R}^3$로 표기되는 축으로의 방향에서 선형 모션(linear motion)의 거리를 나타낸다. 유사하게, 회전 관절은 $'R'$로 표기되고, 관절 값들 q는 축 S에 대하여 회전의 양을 나타낸다.

[0041] 또한, 두 가지의 추가적인 관절 타입으로, 고정 관절(fixed joints)과 항등 관절(identity joints)을 소개한다. 고정 관절은 $'F'$로 표기되고, 관절 값 q에 따른 동작을 수행하지 않는다. 대신에, 오직 오프셋 정보(offset information)만을 유지(hold)한다. 마지막으로, 항등 관절은 $'I'$로 표기되고, 동작하지 않으며 오프셋 정보도 갖지 않는다. 항등 관절은, 기구적 아키텍처 탐색(KAS) 동안 플레이스홀더(placeholder)로서의 역할을 수행하기 위하여, 오직 4×4의 항등 행렬(identity matrix) $I_{4\times4}$만을 출력한다. 실시예에 따르면, 트래킹 포인트에 위치하는 관절은 항등 관절일 수 있다.

[0042] p^{off}를 위치 오프셋, R^{off}를 회전 오프셋이라 하면, 관절에 의해 접속된(connected) 이전의 (i-1)번째 링크 프레임(link frame)에서 표현된 i번째 링크(link)에 대한 변환 행렬(transformation matrix) $^{i-1}T_i$은 수학식 2와 같이 표현된다. 수학식 2에서, 연산자 ∧는 아래와 같이 벡터 $w = [w_x, w_y, w_z]$를 교대행렬(skewed symmetric matrix)로 변환한다.

$$w^{\wedge} = \begin{bmatrix} 0 & -w_z & w_x \\ w_z. & 0 & -w_y \\ -w_x. & w_y. & 0 \end{bmatrix}$$

[0043]

[0044] [수학식 2]

$$^{i-1}T_i = \begin{cases} \begin{bmatrix} R_i^{\text{off}} & R_i^{\text{off}} S_i q_i + p_i^{\text{off}} \\ 0_{1\times3} & 1 \end{bmatrix}, & \text{Prismatic joint} \\ \begin{bmatrix} R_i^{\text{off}} \exp[(S_i q_i)^{\wedge}] & p_i^{\text{off}} \\ 0_{1\times3} & 1 \end{bmatrix}, & \text{Revolute joint} \\ \begin{bmatrix} R_i^{\text{off}} & p_i^{\text{off}} \\ 0_{1\times3} & 1 \end{bmatrix}, & \text{Fixed joint} \\ I_{4\times4}, & \text{Identity joint} \end{cases}$$

[0045]

[0046] 도 2는, (a) 제안 기법(KINN)의 전방 기구학, (b) 회전 관절과 선형 관절로 구성되는 강체 로봇의 강체 기구학의 파라미터들, 그리고 (c) 소프트 로봇 손가락과 이의 수도 강체(PRB)를 도시한다. 도 2에서, 수도 강체 관절의 위치는 $\mathbf{x}$로, 트래킹 관절(tracking joints)의 위치는 $\mathbf{x}_{\text{E}}$로 표기한다.

[0047] $\mathbf{FK}_{\text{r}}$로 표기되는 강체 기구학은, 도 2b에 나타난 바와 같이, θ_s로 정의된 구조 파라미터(structural parameters)와 θ_t로 정의된 관절 유형 파라미터로 파라미터화된다. 구조 파라미터는 수도 강체(PRB)의 축과 오프셋을 포함하며, $\theta_s = \{p_1^{\text{off}}, R_1^{\text{off}}, S_1, p_2^{\text{off}}, R_2^{\text{off}}, S_2, ..., p_{n_j}^{\text{off}}, R_{n_j}^{\text{off}}, S_{n_j}\}$로 표기된다. 관절 유형 파라미터의 θ_t의 i번째 관절은 4 개의 관절 유형 $\theta_t^i \in \{'P', 'R', 'F', 'I'\}$ 중 하나이다. 강체 기구학 $\mathbf{FK}_{\text{r}}$의 결과는 전역 좌표계(global coordinates)에서 i번째 관절의 변환 행렬(transformation matrix) ^0T_i이며, 이는 수학식 3을 사용해 행렬을 반복적으로(recursively) 곱함으로써 획득될 수 있다.

[0048] [수학식 3]

$$^0T_i = \prod_{j=1,2,...,i} {}^{j-1}T_j$$

[0049]

[0050] KINN은 소프트 로봇의 기구학을 모델링하기 위하여 강체 기구학과 신경망을 모두 사용한다. 이는 도 2a에 나타난 $\mathbf{FK}_{\text{r}}$ 레이어와 함수 g를 통해 이루어진다. KINN의 전방 기구학(FK) 과정에서, 관절 값 $\mathbf{q}$를 얻기 위하여 액추

에이션 명령 **a**가 함수 g에 입력된다. 그런 다음, 관절 값 **q**와 파라미터(θ_s, θ_t)로 로봇의 위치 $\mathbf{x}_E$를 예측하기 위하여 강체 기구학 FK_r이 수행된다.

[0051] 소프트 로봇의 복잡한 동작을 고려하기 위해, 수도 강체(PRB)의 관절 수 n_t는 충분히 크게 선택(예시적으로, 미리 정해진 값보다 큰 임의의 자연수 중에서 선택될 수 있음)된다. 예를 들어, 도 2c에 도시된 소프트 로봇을 모델링할 때, 말단(end)에 부착된 하나의 트래킹 포인트를 갖는 이 소프트 로봇을, 관절 시퀀스(joint sequence) $\theta_t = [\prime P\prime, \prime R\prime, \prime P\prime, \prime R\prime, \prime P\prime, \prime I\prime]$를 갖는 직렬 체인(serial chain)으로 모델링하며, n_t = 7개의 관절을 사용하여 마지막 관절에 부착된 n_j = 1의 트래킹 포인트의 위치를 예측한다. 전방 기구학(FK) 동안, 수학식 3에 나타난 바와 같이, 반복 계산을 통해 수도 강체(PRB)의 모든 관절의 위치 $\mathbf{x} \in \mathbb{R}^{n_t \times 3}$를 획득한다. 관절 위치 **x**로부터, 미리 정의된 간격(예: 도 2c에서 매 7번째 관절)을 선택하여 트래킹 포인트 $\mathbf{x}_E$를 나타내며, 이러한 관절을 트래킹 관절(tracking joint)이라고 한다.

[0052] C. 기구적 아키텍처 탐색(Kinematic Architecture Search, KAS)

[0053] 전방 기구학(FK)의 파라미터는 연속 변수(θ_s, θ_g)와 이산 변수(θ_t)로 나뉘며, 이 두 파라미터를 함께 최적화하는 것은 복잡한 문제이다. NAS(T. Elsken, J. H. Metzen, and F. Hutter, "Neural Architecture Search: A Survey".)에서 영감을 받아, 본 발명에서는 연속 변수와 이산 변수를 반복적으로 최적화한다. 즉, 기구적 아키텍처 탐색(KAS)을 수행한다. 이는, 도 3에 나타난 바와 같이, 소프트 로봇의 내재된 기구학을 탐색한다. θ_t에 대한 이산 최적화는 유전 알고리즘(Genetic Algorithm, GA)(S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," Multimed Tools Appl, vol. 80, no. 5, pp. 8091-8126, Feb. 2021)을 통해 수행되며, (θ_s, θ_g)에 대한 연속 최적화는 종단 간 방식(end-to-end fashion)으로 수행된다. 기구적 아키텍처 탐색(KAS)의 수도 코드(pseudo-code)는 알고리즘 1에 제시되어 있다.

[0054] [알고리즘 1]

Algorithm 1 Kinematic Architecture Search

Input: $(\mathbf{a}, \mathbf{x}_{gt})$
Output: $(\hat{\theta}_g, \hat{\theta}_s, \hat{\theta}_t)$

$R_{min} \leftarrow$ inf
for i in n_p **do**
  $\theta_t \leftarrow$ Random_Initialize
  $\theta_t^1[i] \leftarrow \theta_t$
end for
for $G \leftarrow 1$ to G_{END} **do**
  for i in n_p **do** ▷ continuous optimization
    $\theta_t \leftarrow \theta_t^G[i]$
    $(\theta_s, \theta_g) \leftarrow$ Initalize($\mathbf{x}_{gt}^{init}$)
    $\mathbf{x} \leftarrow FK_r(g(a;\theta_g);\theta_s,\theta_s)$
    $\mathcal{L} \leftarrow \|\mathbf{x}_{gt} - \mathbf{x}_E\|_1 + \lambda_{reg} \sum_{i=0}^{n_t-1} d_c(\mathbf{x}_{gt}[k(i)+1] - \mathbf{x}_{gt}[k(i)], \mathbf{x}[i+1] - \mathbf{x}[i])$
    $R[i] \leftarrow \min_{p_s, p_\theta} \mathcal{L}$
    if $R[i] < R_{min}$ **then**
      $(\hat{\theta}_g, \hat{\theta}_s, \hat{\theta}_t) \leftarrow (\theta_g, \theta_s, \theta_t)$
    end if
  end for
  $\theta_t^{G+1} \leftarrow$ Mutation(θ_t^G, R) ▷ discrete optimization
end for

[0055]

[0056] 1) 연속 최적화: 먼저 연속 변수(θ_s, θ_g)를 수학식 4와 같이 함께 최적화한다. 모든 파라미터가 오토-그라디언트 프레임워크(auto-gradient framework)(A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library." arXiv, Dec. 03, 2019. Accessed: Feb. 27, 2023. [Online].

Available: http://arxiv.org/abs/1912.01703)를 통해 미분 가능하므로, ADAM(D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. Accessed: Feb. 27, 2023.)을 사용하여 최적화를 수행한다.

[0057] [수학식 4]

[0058] $$(\hat{\theta}_s, \hat{\theta}_g) = \arg\min_{(\theta_s,\theta_g)} \mathbb{E}_{D\in(\mathbf{a},\mathbf{x}_{gt})}[d(\mathbf{x}_{gt}, \mathbf{FK}(\mathbf{a};\theta_g,\theta_s,\theta_t))]$$

[0059] 그러나, 수학식 4로 단순하게 최적화하면 수도 강체(PRB)의 관절이 소프트 로봇의 외부에 배치되는 결과를 초래할 수 있다. 이는 실제 기저 기구학을 제대로 표현하지 못하게 하여 모델 식별 시 최적이 아닌 결과를 초래하고, 로봇 제어 시 직관에 반하는 동작을 유발할 수 있다. 이 문제를 해결하기 위해, 두 가지 기법이 사용된다. 먼저, θ_s와 θ_g를 초기화함으로서 로봇의 초기 위치 $\mathbf{x}^{\text{init}}$에 대한 예측을 실제 초기 자세 $\mathbf{x}_{\text{gt}}^{\text{init}}$와 동일하게 만든다. 이는 $q \approx 0_{n_t}$로 설정하고 위치 오프셋 $p_{off} \in \theta_s$를 초기 위치 $\mathbf{x}_{\text{gt}}^{\text{init}}$와 기준 위치(base position) 사이에 관절이 동일한 거리에 배치되도록 설정함으로서 달성될 수 있다. $q \approx 0_{n_t}$로 만들기 위해, 네트워크 g의 바이어스 항(bias term)을 제거하고 가중치를 $\theta_g \sim \mathcal{N}(0, \sigma_{\text{init}}),\ \sigma_{\text{init}} \ll 1$로 초기화한다. 다음으로, 관절을 정렬시키기 위한 정규화 항(regularizing term)을 도입한다. 구체적으로, 수도 강체(PRB)의 관절 위치 $\mathbf{x}$는, 기구학적 트리(kinematic tree)의 뿌리 방향(root direction)의 추적 관절로부터 잎 방향(leaf direction)의 트래킹 관절까지를 가르키는 방향 벡터(directional vector)와 정렬되도록 제약된다. 정규화 항은 다음과 같은 코사인 거리로 표현된다.

[0060] $$\mathcal{L}_{\text{reg}} = \sum_{i=0}^{n_t-1} d_c(\mathbf{x}_{gt}[k(i)] - \mathbf{x}_{gt}[\lambda(i)], \mathbf{x}[i+1] - \mathbf{x}[i])$$

[0061] 여기서, d_c는 코사인 거리를 나타내며, k와 λ로 표시된 두 개의 인덱싱 함수(indexing functions)는 각각 관절 인덱스 $i \in [1, n_t]$를 기구학적 트리의 뿌리와 잎 방향에서 가장 가까운 추적 관절의 인덱스 $j \in [1, n_j]$로 매핑한다. 수학식 4와 함께, 연속 최적화의 목표는 다음과 같다.

[0062] $$\mathcal{L} = \|\mathbf{x}_{gt} - \mathbf{x}_E\|_1 + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}$$

[0063] 2) 이산 최적화: 관절 유형 파라미터들 n_t의 길이가 주어지면, 관절 유형 시퀀스의 이산 변수들 θ_t는 연속 최적화의 결과를 바탕으로 유전 알고리즘(GA)에 의해 탐색된다.

[0064] $$\hat{\theta}_t = \arg\min_{\theta_t} \mathbb{E}_{D\in(\mathbf{a},\mathbf{x}_{gt})}[d(\mathbf{x}_{gt}, \mathbf{FK}(\mathbf{a};\theta_g,\theta_s,\theta_t))]$$

[0065] n_p는 아카이브(archive)에 있는 에이전트의 수를 나타내며, $\theta_t^i = [{}^1\theta_t^i, {}^2\theta_t^i, \ldots, {}^{n_p}\theta_t^i]$는 i번째 세대(generation)의 에이전트에 대한 관절 유형을 나타낸다고 하자. 아카이브는, 변이되고(mutated) 평가되는(evaluated), 상이한 관절 유형을 가진 에이전트들을 유지한다. 프로세스를 시작하기(initiate) 위해, 첫 번째 세대 θ_t^1는 무작위로 선택되어 연속 최적화 알고리즘에 입력된다. 에이전트들은 검증 데이터셋(validation dataset)에서의 오류에 대한 L1 노름을 기준으로 평가되며, 성능이 가장 좋은 상위 10%의 에이전트들이 선택되어 다음 세대로 진행된다. 상위 50% 중에서, 두 개의 에이전트가 무작위로 선택되고, 무작위 위치에서 분할된(bisected) 후, 결과적인 부분들(resulting parts)은, 아카이브에 n_p 개의 에이전트가 포함될 때까지, 서로 병합된다(merged).

[0066] **소프트 로봇의 기구학적 제어(KINEMATIC CONTROL OF SOFT ROBOT)**

[0067] 모델 식별(model identification)뿐만 아니라, KINN의 학습된 모델을 사용하여 역기구학(inverse kinematics, IK)을 해결하는 제어 프레임워크를 제시한다. 역기구학(IK)은 타겟 위치 $\mathbf{x}_{tar}$를 $\mathbf{x}_{tar} = \mathbf{FK}(\hat{\mathbf{a}};\theta)$를 만족하는 액추에이션 명령 $\hat{\mathbf{a}}$로 매핑하는 함수이다.

[0068] $$\hat{\mathbf{a}} = \mathbf{IK}(\mathbf{x}_{\text{tar}})$$

[0069] A. 소프트 로봇의 역기구학(Inverse Kinematics of Soft Robot)

[0070] 소프트 로봇을 제어하기 위해 감쇠 최소자승 역기구학(S. Chiaverini, B. Siciliano, and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," IEEE Trans. Contr. Syst. Technol., vol. 2, no. 2, pp. 123-134, Jun. 1994, doi: 10.1109/87.294335.)의 잘 알려진 기법을 활용한다. 엔드-이펙터의 속도는 속도 타겟 $\mathbf{v}_{\text{tar}}$로 제어되며, 여기서 K_p는 비례 게인(proportional gain)이고, 속도 타겟은 $\mathbf{v}_{\text{tar}} = K_p(\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{E}})$로 설정된다. 한편, 속도와 기구학 함수(kinematics function) 사이의 관계는 $\mathbf{v} = J\dot{\mathbf{a}}$로 유지되며, 여기서 $J = \nabla_{\mathbf{a}}\mathbf{x}_{\text{E}} \in \mathbb{R}^{n_a \times n_j}$이다. 따라서, 역기구학의 목표는 수학식 5의 최적화 문제로 표현될 수 있다.

[0071] [수학식 5]

[0072] $$\hat{\dot{\mathbf{a}}} = \arg\min_{\dot{\mathbf{a}}} \|K_p(\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{E}}) - J\dot{\mathbf{a}}\|$$

[0074] 위 최적화 문제는 해석적 해(analytic solution)를 도출하는 최소자승법(least-squares)에 의해 풀이될 수 있다.

[0075] $$\hat{\dot{\mathbf{a}}} = J^{\dagger}K_p(\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{E}})$$

[0076] 여기서 †는 의사역(pseudo inverse)을 나타낸다. $\hat{\dot{\mathbf{a}}}$를 구하고, 원하는 액추에이션 명령 $\hat{\mathbf{a}}$은, 도 4b에 나타난 바와 같이, 수렴(convergence)할 때까지 γ로 표기되는 스텝 사이즈로 업데이트된다.

[0077] [수학식 6]

[0078] $$\hat{\mathbf{a}} \leftarrow \hat{\mathbf{a}} + \gamma\hat{\dot{\mathbf{a}}}$$

[0079] 제안된 제어 프레임워크는 야코비안(Jacobian) J를 얻는 것을 필요로 한다. 야코비안 J를 자동 미분 프레임워크(auto-gradient framework)를 통해 직접 구할 수 있지만, $\mathbf{FK}_r$의 전방향 및 역방향 계산(forward and backward computation) 동안 체인 구조에 대한 재귀적 계산(recursive computation) 때문에 계산이 느려질 수 있다. 대신, 강체 기구학 $\mathbf{FK}_r$의 야코비안의 명시적 형태(explicit form)를 활용함으로서, 야코비안 J를 효율적으로 얻을 수 있다. 본 발명에서는 신경망 g에 대해 역방향 계산만을 수행하여 $\nabla_{\mathbf{a}}\mathbf{q} \in \mathbb{R}^{n_a \times n_t}$를 얻고, $\nabla_{\mathbf{q}}\mathbf{x}_{\text{E}} \in \mathbb{R}^{n_t \times n_j}$에 대한 명시적 유도(explicit derivation)를 활용한다. 수도 강체(PRB)의 i번째 관절에 대한 야코비안 $\nabla_{\mathbf{q}}\mathbf{x}_{\text{E}}$은 아래에 제시되어 있다.

[0080] $$(\nabla_{\mathbf{q}}\mathbf{x}_{\text{E}})[:, i] = \begin{cases} S^i & \text{Prismatic joint} \\ S^i \times (\mathbf{x}_{\text{E}} - \mathbf{x}[i]) & \text{Revolute joint} \\ 0_{3\times 1} & \text{Fixed joint} \\ 0_{3\times 1}, & \text{Identity joint} \end{cases}$$

[0081] $$\nabla_{\mathbf{a}}\mathbf{q} = \text{backward}(g(\mathbf{a}; \theta_g))$$

[0082] 마지막으로, 전체 야코비안(total Jacobian) J는 연쇄 법칙(chain rule)을 사용하여 계산된다.

[0083] $$J = \nabla_{\mathbf{a}}\mathbf{q}\nabla_{\mathbf{q}}\mathbf{x}_{\text{E}}$$

[0084] 이러한 가속 기법(acceleration technique)은 실제 환경에서 건식 구동 소프트 로봇을 사용한 실험에서 야코비

안 J의 계산 속도를 2.16배 빠르게 향상시켰음을 확인하였다.

[0085] B. 제약 조건을 갖는 역기구학(Inverse Kinematics with Constraints)

[0086] 소프트 로봇을 제어하는 것은, 특히 건식(tendons)을 포함하는 경우, 구성의 길이(configuration's length)로 인한 제약 때문에 어려움을 겪는다. 예를 들어, 물체를 잡는 데(grasping objects)에 있어서 소프트 로봇 손가락의 성공은, 정확한 엔드-이펙터의 포지셔닝뿐만 아니라 잡는 방향(grasping direction)으로의 적절한 손가락 움직임에 의존한다. 이러한 문제를 해결하기 위해, 본 발명은, 소프트 로봇의 확장(extension)과 방향(direction)을 효과적으로 제어할 수 있도록 하는, 제약 조건을 갖는 역기구학 문제(constrained inverse kinematics problem)를 개발하고 해결하였다.

[0087] 본 발명의 목표는 로봇의 위치를 지정된 굽힘 방향(specified bending direction) $\mathbf{u} \in \mathbb{S}^3$으로 제어하는 것으로, 타겟 거리 d_{tar}는 도 4a에 나타나 있다. 여기서, $\mathbb{S}^3$는 단위 3-구(unit 3-sphere)이다. 이는 $e_c = (d_{\text{tar}} - \mathbf{u}^T\mathbf{x}_{\text{E}})$로 정의되는 오차(error)를 최소화함으로서 달성된다. 피드백 제어기(feedback controller)를 사용하여, $\|K_c e_c - \frac{\partial(\mathbf{u}^T\mathbf{x}_{\text{E}})}{\partial t}\|^2$를 최소화하는 $\hat{\dot{\mathbf{a}}}$를 계산하고, 수학식 6과 같이 액추에이션 명령을 업데이트한다. 여기서, K_c는 제약 조건에 대한 게인 계수(gain coefficient)이다.

[0088] $\frac{\partial(\mathbf{u}^T\mathbf{x}_{\text{E}})}{\partial t}$는 다음과 같이 전개될 수 있기 때문에,

$$\begin{aligned}\frac{\partial(\mathbf{u}^T\mathbf{x}_{\text{E}})}{\partial t} &= \mathbf{u}^T\frac{\partial \mathbf{x}_{\text{E}}}{\partial t} + \mathbf{x}_{\text{E}}^T\cancel{\frac{\partial \mathbf{u}}{\partial t}} \\ &= \mathbf{u}^T\frac{\partial \mathbf{x}_{\text{E}}}{\partial \mathbf{a}}\frac{\partial \mathbf{a}}{\partial t} \\ &= \mathbf{u}^T J\dot{\mathbf{a}}\end{aligned}$$

[0089]

[0090] 제약 조건을 갖는 역기구학의 목표는 수학식 7과 같이 다시 쓸 수 있다.

[0091] [수학식 7]

[0092] $$\hat{\dot{\mathbf{a}}} = \arg\min_{\dot{\mathbf{a}}} \|K_c(d_{\text{tar}} - \mathbf{u}^T\mathbf{x}_{\text{E}}) - \mathbf{u}^T J\dot{\mathbf{a}}\|$$

[0094] 수학식 5와 함께, 수학식 7은 최소자승법을 통해 풀이되며, 이는 도 4b에서 IK 스텝으로 언급된다.

[0095] $$\hat{\dot{\mathbf{a}}} = \begin{bmatrix} J \\ \mathbf{u}^T J \end{bmatrix}^{\dagger} \begin{bmatrix} K_p(\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{E}}) \\ K_c(d_{\text{tar}} - \mathbf{u}^T\mathbf{x}_{\text{E}}) \end{bmatrix}$$

[0096] C. 하이브리드 시스템의 기구학적 제어(Kinematic Control of Hybrid System)

[0097] 제안된 방법은, 소프트 로봇을 마치 강체 로봇처럼 제어할 수 있게 하여, 소프트 및 강체 로봇(soft and rigid-bodied robots)으로 구성된 하이브리드 시스템을 동시에 제어하는 것으로 확장될 수 있다. 특히, 소프트 로봇이 강체 로봇의 끝에 부착된 경우에 집중한다.

[0098] $n_{a^{\text{r}}}$ 개의 액추에이션 명령($\mathbf{a}^{\text{r}} \in \mathbb{R}^{n_{a^{\text{r}}}}$로 표기함)을 갖는 하이브리드 시스템의 강체 부분(rigid-bodied parts)을 고려하자. 강체 부분의 위치와 회전(예: 오일러 각도(Euler angles))은 각각 $\mathbf{x}^{\text{r}}$과 $\mathbf{w}^{\text{r}}$로 정의된다. 타겟 위치와 타겟 회전은 각각 $\mathbf{x}_{\text{tar}}$와 $\mathbf{w}^{\text{r}}_{\text{tar}}$로 나타내고, 타겟 방향과 타겟 거리는 각각 $\mathbf{u}$와 d_{tar}로 나타낸다. 최소자승법을 통해, 강체의 역기구학을 수학식 5 및 수학식 7과 결합하여 풀이하면, 다음과 같은 수학식이 도출된다. 여기서, K_w는 회전 제어(rotational control)를 위한 게인이다.

$$\begin{bmatrix} \hat{\mathbf{a}}_{\mathrm{r}} \\ \hat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{a}^{\mathrm{r}}}\mathbf{x}_{\mathrm{E}} & \nabla_{\mathbf{a}}\mathbf{x}_{\mathrm{E}} \\ \nabla_{\mathbf{a}^{\mathrm{r}}}\mathbf{w}^{\mathrm{r}} & 0_{3\times n_a} \\ 0_{1\times n_{a^{\mathrm{r}}}} & \mathbf{u}^T J \end{bmatrix}^{\dagger} \begin{bmatrix} K_p(\mathbf{x}_{\mathrm{tar}} - \mathbf{x}_{\mathrm{E}}) \\ K_w(\mathbf{w}^{\mathrm{r}}_{\mathrm{tar}} - \mathbf{w}^{\mathrm{r}}) \\ K_c(d_{\mathrm{tar}} - \mathbf{u}^T\mathbf{x}_{\mathrm{E}}) \end{bmatrix}$$

[0099]

[0100] **소프트 로봇 모델 식별에 대한 실험(EXPERIMENT ON SOFT ROBOT MODEL IDENTIFICATION)**

[0101] 이하에서는, 모델 식별 방법의 일반화 성능(generalization performance)을 평가하며, 여기에는 학습 데이터의 도메인 내외에서의 위치 예측이 포함된다. 이 평가는, 학습되지 않은 액추에이션에서도 정확한 예측을 수행할 수 있는 능력을 측정하는 데 필수적이며, 이는 실제 환경에서의 안전하고 실용적인 동작을 보장하는 데 중요하다.

[0102] 제안된 프레임워크는 다양한 소프트 로봇에 유연하게 적용할 수 있어, 본 발명에서는 두 종류의 소프트 로봇, 즉 건식 및 공압식 소프트 로봇을 대상으로 테스트를 진행하였다. 구체적으로, 건식 로봇에 대한 성능은 시뮬레이션 환경과 실제 환경에서 평가하였으며, 공압식 소프트 로봇은 시뮬레이션 환경에서만 테스트되었다. 환경과 학습에 대한 하이퍼파라미터는 후술한다.

[0103] 각 소프트 로봇에 대해, 수집된 데이터셋은 액추에이션의 L1-노름(L1-norm)에 의해 정렬된다. 정렬된 데이터 중 상위 30%는 외삽 테스트 데이터셋(extrapolation test dataset)의 형태로 분리된다. 데이터의 나머지 70%는 각각 8:1:1의 비율로 훈련, 검증, 테스트 데이터셋으로 무작위로 분할된다.

[0104] A. 기준 모델(Baselines)

[0105] 1) PureNN: 신경망에 대한 물리적 사전 지식(physics prior)의 영향을 평가하기 위해, 어떠한 물리적 사전 지식도 없는 신경망을 기준 모델의 하나로 설정하였다. 이 신경망은, $\mathbf{FK}_{\mathrm{NN}}$으로 표기됨며, 소프트 로봇의 순방향 기구학을 모델링하기 위해 구축된다(J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, “Soft Robot Control With a Learned Differentiable Model,” in 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft), New Haven, CT, USA, May 2020, pp. 417-423.). 수집된 쌍을 이루는 데이터 $(\mathbf{a}, \mathbf{x}_{\mathrm{gt}})$로, 모델 파라미터 θ_{NN}는, ADAM(D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. Accessed: Feb. 27, 2023.)으로 $\hat{\theta}_{\mathrm{NN}} = \arg\min_{\theta_{\mathrm{NN}}} \|\mathbf{FK}_{\mathrm{NN}}(\mathbf{a};\theta_{\mathrm{NN}}) - \mathbf{x}_{\mathrm{gt}}\|_1$를 풀이함으로써, 탐색된다. 활성화 함수는, 모델 식별에 대한 성능 향상을 위한 이전 연구(J. M. Bern 등의 연구)에서 사용된 Sigmoid 대신 Mish로 구현될 수 있다. 추가적으로, 위치 데이터는, 평균을 빼고 표준 편차로 나누는 방식으로 화이트닝 처리될 수 있다.

[0106] 2) GP: 모델 식별은 블랙-박스 접근법으로 가우시안 프로세스(Gaussian Process)를 사용하여 수행된다. PureNN과 마찬가지로, 액추에이션 및 위치 데이터에 대해 화이트닝 처리가 적용된다. 본 발명에서는, SE(Squared Exponential) 커널을 사용하여 로봇의 위치를 예측하는 회귀 모델(regressor)을 적합(fit)시켰다. SE 커널은 다음과 같이 정의된다.

$$k(\mathbf{a}_i, \mathbf{a}_j) = \sigma^2 \exp\left(-\frac{1}{2}(\mathbf{a}_i - \mathbf{a}_j)^T L^{-1}(\mathbf{a}_i - \mathbf{a}_j)\right)$$

[0107]

[0108] 여기서, L은 특징 길이(characteristic lengths)의 대각 행렬(diagonal matrix)이다. 주어진 액추에이션 $\mathbf{a}^*$에 대해, 예측된 위치 $\mathbf{x}^*$는 다음과 같이 계산된다.

$$\mathbf{x}^* = K(\mathbf{a}^*, \mathbf{a}^*)(K(\mathbf{a}, \mathbf{a}) + \sigma_n^2 I)^{-1}\mathbf{x}_{\mathrm{gt}}$$

[0109]

[0110] 여기서, $K(\cdot,\cdot)$는 커널 행렬을 나타내며, σ_n^2는 노이즈 분산, I는 항등 행렬, $(\mathbf{a}, \mathbf{x}_{\mathrm{gt}})$는 학습 데이터를 나타낸다.

[0111] 3) PCCNN: 상이한 물리 기반 신경망(physics-informed neural networks)과 비교하기 위해, PCC 모델(S. Terrile, A. Lopez, and A. Barrientos, “Use of Finite Elements in the Training of a Neural Network for the Modeling of a Soft Robot,” Biomimetics, vol. 8, no. 1, p. 56, Jan. 2023)을 기반으로 한 신경망을 사용한 모델 식별 결과를 제시한다. 이 모델은 PCC 모델링을 신경망 g_{PCC}에 통합하여 PCC의 순방향 기구학에 적용

한다. 순방향 기구학은 두 부분으로 구성된다. 먼저, g_{PCC}는 PCC 파라미터들 $[\phi, \theta, r] = g_{\text{PCC}}(\mathbf{a})$를 예측한다. 그런 다음, 수학식 8을 사용하여, PCC 모델의 순방향 기구학인 FK_{PCC}가 수행된다.

[0112] [수학식 8]

[0113]
$$T = \begin{bmatrix} \cos\phi & \sin\phi\cos\theta & -\sin\phi\sin\theta & r\sin\phi(1-\cos\theta) \\ -\sin\phi & \cos\phi\cos\theta & -\sin\theta\cos\phi & r\cos\phi(1-\cos\theta) \\ 0 & \sin\theta & \cos\theta & r\sin\theta \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

[0114] 4) FEM: FK_{FEM}으로 표기되는 FEM의 순방향 기구학으로, 모델 파라미터들은 영률(Young's modulus) E와 포아송 비율(Poisson ration) ν로 구성된다. 먼저, 위치 손실 $\mathcal{L} = \|\mathbf{x} - \text{FK}_{\text{FEM}}(a; E, v)\|_1$과 이의 그라디언트 $(\frac{\partial \mathcal{L}}{\partial E}, \frac{\partial \mathcal{L}}{\partial v})$를, Differentiable Projective Dynamics(T. Du et al., "DiffPD: Differentiable Projective Dynamics," ACM Trans. Graph., vol. 41, no. 2, pp. 1-21, Apr. 2022)을 사용하여, 계산한다. 그런 다음, 모델 파라미터들 (E, ν)는 L-BFGS(D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," Mathematical Programming, vol. 45, no. 1-3, pp. 503-528, Aug. 1989)를 사용하여 식별된다.

[0115] 5) PCC: 또한, 건식 구동 소프트 로봇의 명시적인 기하학적 관계와 PCC 가정(assumption)을 활용하는 모델 기반 접근법과 비교하다. 액추에이션 명령 $\mathbf{a}$(즉, 풀리(pulley)의 회전각(rotation angle))으로부터, i번째 액추에이션의 당겨진 힘줄 길이(pulled tendon length) l_i는 $l_i = l_0 - r_p a_i$로 설명될 수 있으며, 여기서 r_p는 풀리의 반경(radius)이다. PCC 파라미터들 $[\phi, \theta, r]$의 명시적인 형태(explicit form)는 당겨진 힘줄 길이 l과 그에 따른 위치 사이의 기하학적 관계(R. J. Webster and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," The International Journal of Robotics Research, vol. 29, no. 13, pp. 1661-1683, Nov. 2010)를 통해 획득할 수 있다. 본 발명의 사례 연구에서와 같이, 4개의 힘줄 액추에이션(tendon actuation)을 가진 소프트 로봇의 경우, 파라미터화는 수학식 9와 같이 작성되며, 여기서 d는 중심점(center point)에서 힘줄 경로(tendon path)까지의 거리이다.

[0116] [수학식 9]

[0117]
$$\phi = \arctan\left(\frac{l_4 - l_2}{l_3 - l_1}\right)$$
$$\theta = \frac{\sqrt{(l_3 - l_1)^2 + (l_4 - l_2)^2}}{2d}$$
$$r = \frac{l_1 + l_2 + l_3 + l_4}{4\theta}$$

[0118] 소프트 로봇의 결과적인 위치는 수학식 8에서 FK_{PCC}를 통해 얻어진다. 각각 r_p와 l_0로 정의된 등가 반경(effective radius)과 초기 길기(initial length)는 최적화 문제 $\hat{r}_p, \hat{l}_0 = \arg\min_{r_p, l_0} \|\text{FK}_{\text{PCC}}(\phi, \theta, r; r_p, l_0), \mathbf{x}_{\text{gt}}\|_1$를, L-BFGS로 풀이함으로써 찾는다.

[0119] 6) Bayes-Cosserat: 베이지안 최적화(Bayesian Optimization, BO)는 데이터 $\mathcal{D}$를 활용하여 블랙박스 함수를 최적화하는 베이지안 접근법을 채용하는 샘플링 기반 방법이다. 목표는 소프트 로봇의 최적 재료 파라미터들 θ_{BO}(길이, 영률(Young's modulus))를 BO를 사용하여 찾는 것이다. 소프트 로봇은 Cosserat Rod 모델을 사용하여 SoRoSim(A. T. Mathew, I. M. B. Hmida, C. Armanini, F. Boyer, and F. Renda, "SoRoSim: A MATLAB Toolbox for Hybrid Rigid-Soft Robots Based on the Geometric Variable-Strain Approach," IEEE Robot. Automat. Mag., pp. 2-18, 2022)으로 시뮬레이션되며, 이 시뮬레이션의 전방향 시뮬레이션은 $\mathbf{x} = \text{FK}_{\text{BO}}(\mathbf{a}; \theta_{\text{BO}})$로 나타낸다.

[0120] SE 커널은 가는 가우시안 프로세스 $s \sim \mathcal{GP}(s_\mu, s_\sigma)$를 이용하여 대리 함수(surrogate function) s를 적합(fit)함으로서, 에러 $e_{\text{BO}} = \|(\mathbf{x}_{\text{gt}} - \mathbf{FK}_{\text{BO}}(\mathbf{a}; \theta_{\text{BO}}))\|$를 추정한다. 대리 함수와 함께, 획득 함수(acquisition function)는 탐색할 파라미터들 θ_{BO}의 다음 포인트를 결정한다. 획득 함수는 기대 개선(Expected Improvement, EI)으로 구현된다.

[0121] $$\mathbf{EI} = (s_\mu - f(\hat{\theta}_{\text{BO}}) - \xi)\Phi(Z) + s_\sigma \phi(Z)$$

[0122] 여기서, $Z = s_\mu - f(\theta) - \xi s_\sigma$이고, $\hat{\theta}_{\text{BO}}$는 현재까지의 최적 파라미터들(best parameters)이고, $\xi = 0.01$는 탐색의 정도(degree of exploration)를 제어한다.

[0123] B. 시뮬레이션 상에서 공압 로봇의 모델 식별

[0124] 높은 신뢰성을 가진 데이터셋을 수집하기 위하여, ABAQUS(L. Borgesson, “ABAQUS,” in Developments in Geotechnical Engineering, vol. 79, Elsevier, 1996, pp. 565-570.)를 사용하여 공압 소프트 로봇을 시뮬레이션한다. 로봇의 구성은, 도 5에 도시된 바와 같이, A. D. Marchese 등의 연구(A. D. Marchese and D. Rus, “Design, kinematics, and control of a soft spatial fluidic elastomer manipulator,” The International Journal of Robotics Research, vol. 35, no. 7, pp. 840-869, Jun. 2016)의 구성을 대략적으로 따른다. 로봇의 길이는 1800㎜이며, 세 개의 동일한 간격으로 나누어진 섹션으로 나뉜다. 각 섹션에서는 네 개의 챔버 중 하나가 액추에이션을 위해 선택된다. 로봇은, 0kPa에서 30kPa 범위의 압력을 갖는 세 개의 독립적인 챔버에 의해 동작된다. 소프트 로봇에는 7개의 트래킹 포인트가 부착되어 있으며, 트래킹 포인트의 액추에이션과 위치에 대하여 쌍을 이루는 6,644 개의 데이터가 수집된다.

[0125] 식별 결과는 예측된 위치와 실제 위치 간의 에러 $\|\mathbf{x}_{\text{E}} - \mathbf{x}_{\text{gt}}\|_1$에 의해 평가된다. 이는 보간(interpolation) 및 외삽(extrapolation) 영역에서 평가되며, 자세한 내용은 표 2에 나타나 있다. PCC와 Bayes-Cosserat는 공압 로봇을 모델링할 수 없으며 FEM은 해의 수렴 문제가 존재하여, 실험에서 제외되었다. 보간의 경우, 제안된 모델이 GP에 비해 1.794㎜의 약간 증가된 오차를 보인다. 이는 기구학 사전 지식이 소프트 로봇의 위치 예측에 편향(bias)을 줄 수 있기 때문이다. 그러나, 외삽에서는, 제안된 모델이 GP에 비해 33.09㎜의 상당한 오차 감소를 보이며, 이는 기구학 사전 지식이 학습 도메인을 넘어 일반화하는 데 중요한 역할을 하기 때문이다. 아래 표 2은 식별 결과를 나타낸다.

[0126] [표 2]

[0127]

Method		Pneumatic (Simulated)		Tendon (Simulated)		Tendon (Real-World)	
		Interpolation Error (mm)	Extrapolation Error (mm)	Interpolation Error (mm)	Extrapolation Error (mm)	Interpolation Error (mm)	Extrapolation Error (mm)
Black Box	PureNN [9]	4.605	49.40	1.897	22.93	**2.141**	25.21
	GP [28]	**3.189**	43.67	1.146	3.031	6.373	12.37
Model-Based/ Physics-Informed	PCCNN [32]	306.46	359.28	12.86	25.80	12.38	24.56
	FEM [8]	-	-	4.759	32.903	13.23	16.23
	PCC [16]	-	-	11.04	21.10	10.46	21.03
	Bayes-Cos	-	-	18.34	33.38	12.00	32.09
	KINN(Ours)	4.983	**10.58**	**0.8547**	**2.749**	2.711	**4.831**

[0128] C. 시뮬레이션에서 건식 구동 로봇의 모델 식별

[0129] 실제 건식 구동 소프트 로봇을 시뮬레이션 환경에서 Elastica(N. Naughton, J. Sun, A. Tekinalp, T. Parthasarathy, G. Chowdhary, and M. Gazzola, “Elastica: A Compliant Mechanics Environment for Soft Robotic Control,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 3389-3396, Apr. 2021)를 사용하여 재현하였다. 로봇의 구성은 도 7에 나타나 있다. 소프트 로봇의 길이는 120㎜이며, 네 개의 힘줄과 두 개의 독립적인 액추에이션 시스템을 포함한다. 도 7b는 건식 구동 로봇의 액추에이션 시스템을 도시한다. 각 액추에이션 시스템에서, 두 개의 전면 모터(front-facing motors) 중 하나가 선택되어 소프트 로봇에 내장된 네 개의 힘줄 중 두 개를 작동시킨다. 각 힘줄은 수평 및 수직 방향의 움직임을 생성한다. 2,000개의 짝지어진 데이터 포인트를 수집하기 위해, 소프트 로봇은 작동 범위(actuation range)에 걸쳐 균등 분포(uniform distribution)로 샘플링된 액추에이션 명령으로 시뮬레이션된다. 음의 작동(negative actuation)은 반대쪽에 있는 모터로 힘줄을 당기는 것을 나타낸다.

[0130] 표 2에 나타난 식별 결과는, 제안된 방법이 가장 높은 성능을 보임을 보여준다. PureNN과 비교했을 때, 기구학

적 사전 지식을 활용함으로써 보간과 외삽에서 각각 1.0423mm와 20.181mm의 오차 감소를 얻을 수 있었다. 또한, 기구학적 사전 지식은 PCC 사전(PCC prior) 보다 보간과 외삽에서 각각 12.005mm와 23.051mm의 적은 오차를 보였다.

[0131] D. 건식 구동 로봇의 실제 환경에서의 모델 식별

[0132] 제안하는 모델 식별은 실제 환경에서 건식 구동 로봇을 대상으로 테스트되었다. 이 소프트 로봇은, 3D 프린터로 몰드를 인쇄하고 Ecoflex 00-30을 중합하여(polymerizing) 제작된다(도 7a 참조). 로봇은 후술하는 바와 같이, 두 개의 모터를 사용하여 작동된다. Opti-Track(FLEX13)의 모션 캡처 장치 다섯 개를 사용하여 모터의 회전 각도 $\mathbf{a}$와 엔드-이펙터 위치 $\mathbf{x}_E$의 짝지어진 데이터셋을 수집한다. 보간 및 외삽에 대한 데이터는 도 7c에 나타나 있다.

[0133] 표 2에 따르면, KINN은 보간 영역에서 로봇의 위치를 2.711mm의 오차로 예측하며, 이는 히스테리시스(hysteresis)와 센서로 인한 2.1mm의 내재적 오차(intrinsic error)를 고려할 때 무시할 수 있다. 내재적 오차를 계산하기 위해, 소프트 로봇을 반복적으로 초기화하고 고정된 작동 지점인 $\mathbf{a} = (\pi/2 \text{ rad}, 0)$에서 작동시켰다. 그리고, 표준 편차가 계산되었다. 제안된 모델은 보간 영역에서 PureNN에 비해 약간 더 높은 오차를 보이지만, 그 차이는 밀리미터 미만의 범위이다. 반면, KINN은 외삽 태스크에서 20.379mm의 성과를 보이며, PureNN보다 우수한 성능을 발휘한다. 이러한 특성은 소프트 로봇이 궤적을 따라 견고하게 제어될 수 있도록 한다.

[0134] **소프트 로봇 제어에 관한 실험**

[0135] 이하에서는, 제안된 제어 프레임워크를 실제 건식 구동 소프트 로봇에 적용하여 정확성과 견고성을 평가한다. 모델은 오직 보간 데이터로만 학습되었으며, 학습 도메인 외부의 궤적을 추적하기 위해 사용된다. 그런 다음, 소프트 손가락과 강체 팔의 협동 모션(coordinated motion)을 보이며, 브러시(brush)로 글자(letters)를 칠하는(paint) 작업을 시연한다. 마지막으로, 정교하고 정확한 제어가 필요한 복잡한 병 열기 태스크를 시연한다. 제어를 위한 파라미터들은 후술한다.

[0136] A. 엔드-이펙터 궤적 추적 태스크

[0137] 제안된 모델 식별 및 제어 시스템의 견고성을 입증하기 위해, 소프트 로봇이 학습 영역 내외의 궤적을 추적하는 실험을 진행하였다. 이 실험에서는, 제안된 모델의 추적 오류(tracking error)를 평가하고 PureNN과 비교하였다.

[0138] 도 6a에 따르면, PureNN과 KINN 모두 학습 영역 내에서 정사각형 궤적을 2.773mm와 2.295mm의 오류로 추적할 수 있다. 그러나, 외삽 영역에서는, PureNN이 12.99mm의 오류로 실패하는 반면, KINN은 2.690mm의 오류로 궤적을 성공적으로 추적하여 PureNN보다 10.300mm 적은 오류를 기록하였다.

[0139] 제안하는 모델은, PureNN이 해당 도메인에서 모델 식별에서 약간 더 나은 결과를 보였음에도 불구하고, 보간 영역에서 궤적 추적 성능이 PureNN보다 우수하다. 이는 두 모델이 $\frac{\partial \mathbf{x}_E}{\partial \mathbf{a}}$(즉, 액추에이션 명령에 대한 위치의 그라디언트 필드)를 얼마나 정확하게 근사하는지에 기인할 수 있다.

[0140] 도 6b는 KINN과 PureNN이 2,500개의 동작 지점(actuation points)에서 예측한 그라디언트 필드를 보여준다. 이 지점들은 동작 범위(actuation range) (-π, π)에서 샘플링되었다. $\frac{\partial \mathbf{x}_E}{\partial \mathbf{a}}$(즉, 야코비안)의 예측 오류를 이상적인 모델과 비교하여 평가한다. 이상적인 모델의 두 그라디언트 필드는 +y와 -x 방향을 향하고 있다. 특히, 그라디언트 필드의 방향만 이상적인 모델로부터 추론할 수 있기 때문에, 코사인 거리를 측정하여 평가한다.

[0141] 제안하는 모델은 보간 도메인에서 그라디언트 필드를 학습하는 데 있어 PureNN보다 우수하며, 0.1725의 차이를 보인다. 이는 제안하는 모델이 야코비안을 정확하게 예측하고, 역기구학을 해결하며, 목표 궤적을 올바르게 추적하도록 한다. 또한, 제안하는 모델은 보간 영역에서 외삽 영역으로의 성능 저하가 0.01779로, PureNN의 1.0513과 비교하여 상당히 적은 편이다. 위치 오류와 야코비안 예측에 따른 궤적 추적 결과는 표 3에 요약되어 있다.

[0142] [표 3]

Measure	Position Error in Trajectory Following		Cosine Distance in Jacobian Prediction	
Model	PureNN	KINN(ours)	PureNN	KINN(ours)
Interpolation	2.773 mm	**2.295 mm**	0.1856	**0.01309**
Extrapolation	12.99 mm	**2.690 mm**	1.2369	**0.03088**

[0143]

[0144] B. 문자 그리기 태스크(Letter Painting Task)

[0145] 소프트 로봇은 조작 태스크(manipulation tasks)를 위해 안전하고 부드러운 모션을 제공할 수 있다. 제안된 방법을 사용하여 소프트 핑거와 팔의 협조적인 움직임(coordinated movement)을 통해, 하이브리드 시스템을 활용하여 브러시로 문자를 그리는 태스크를 시연한다. 이 태스크에서는 두 가지 주요 프리미티브(primitives), 즉 선형 스트로크(Linear Stroke)와 원형 스트로크(Circular Stroke)를 사용한다. 이들 각각은 도 9a와 도 9b에 도시되어 있다. 전체 데이터셋으로 학습된 KINN으로, 하이브리드 시스템의 굽힘(bending)과 위치를 제어하고 두 가지 프리미티브를 수행한다.

[0146] 선형 스트로크는, 일정한 방향 $\mathbf{u}$로, 모션의 시작과 끝에서 각각 d_1과 d_2 사이에서 보간하기 위해 손가락을 d의 거리만큼 구부림으로써, 직선을 그리기 위해 이용된다. 원형 스트로크는 굽힘 방향 $\mathbf{u}$를 원에 접하도록 정렬함으로써, 원을 그린다. 선과 원을 조합하여, 페인팅을 위한 타겟 궤적을 생성한다.

[0147] 제안하는 제어 프레임워크는, 도 9c에 도시된 하이브리드 시스템의 전체적인 제어를 가능하게 한다. 이 시스템은 브러시를 부드럽게 움직여(stroke) 정확하고 연속적인 모션을 생성하고, 문자를 페인팅한다. 제안된 방법으로 페인팅된 문자는 도 9d에 나타나 있다.

[0148] C. 병뚜껑 열기 태스크

[0149] 병을 열기 위해서는, 단단히 잡고(firmly grasp) 병뚜껑을 비틀기 위한 마찰 접촉(frictional contack)을 생성하기 위한, 정밀한 모션과 파워가 필요하다. 손가락과 팔의 움직임을 전체적으로 조정하여 병뚜껑을 성공적으로 여는 능력을 보인다. 이를 위해 도 8에 설명된 트위스트-프리미티브(twist-primitive)를 활용한다.

[0150] 트위스트-프리미티브는 RealSense-d435 깊이 카메라와 Y. Xiang 등의 연구(Y. Xiang, C. Xie, A. Mousavian, and D. Fox, “Learning RGB-D Feature Embeddings for Unseen Object Instance Segmentation”.)에서 언급된 세그멘테이션 모듈을 사용하여 타겟 객체의 포인트 클라우드를 캡처하고 추출하는 과정을 포함한다. 객체의 중심은 3차원 공간에서 수집된 포인트 클라우드의 중앙값(median)을 이용하여 계산된다. 그러면, 하이브리드 시스템은 타겟 객체의 10cm 위로 접근한다. 다음으로, 병뚜껑을 67.5도 회전시키기 위해, 프리-트위스트 모션(pre-twist motion)과 트위스트 모션(twisting motion)이 수행된다. 트위스트 모션 동안, 굽힘 방향 $\mathbf{u}$는 병뚜껑의 중심을 향하는 방향으로 설정되고, 굽힘 거리 d는 인식된 병뚜껑 반지름의 1/3로 설정된다. 앞서 언급한 두 가지 모션을 두 번 반복함으로써, 하이브리드 시스템은 물병의 병뚜껑을 성공적으로 열 수 있다.

[0151] 제안하는 모델 식별은 견고한 외삽 능력을 제공하여, 학습 데이터 범위 외의 계획도 가능하게 한다. 실험에서, 배치된 모션의 최대 회전은 L1-노름 $\|(\mathbf{a_1}, \mathbf{a_2})\|_1$에서 측정되었으며, 이 값은 1.168π 라디안(rad)으로 훈련 데이터의 π 라디안을 16.8% 초과하였다.

[0152] 이하에서는, 모델과 환경에 대한 파라미터들을 설명한다.

[0153] 아래 표 4는 제어 파라미터들을 나타낸다.

[0154] [표 4]

Symbol	Parameter	Value
γ	Step size of inverse kinematics	0.1
K_p	Gain for positional control	50
K_c	Gain for bending constraint	50
K_w	Gain for rotational control	1

[0155]

[0156] 표 5은 모델 인식을 위한 파라미터들을 나타낸다.

[0157] [표 5]

Parameters	Pneumatic (Simulation)	Tendon (Simulation)	Tendon (Real-World)
lr	0.0015	0.0015	0.0015
λ_{reg}	0.1	0.1	0.1
n_t	49	7	7
n_j	7	1	1
Hidden Dimension	(32,32)	(16,16)	(16,16)
Activation Function	ReLU	Mish	Mish
Batch Size	128	64	64

[0158]

[0159] 표 6은 시뮬레이션에서 공압 로봇의 상세를 나타낸다.

[0160] [표 6]

Model	Physical Model	Yeoh Hyperelastic Model
	Actuation Range	$0\,\text{kPa} \leq P \leq 30\,\text{kPa}$
	Number of Actuation	3
	Length	1800 mm
Material	Density	$1049\ \text{kg/m}^3$
	C10	0.11
	C20	0.02
	C30	0
	D1	0.1
	D2	0
	D3	0

[0161]

[0162] 표 7은 시뮬레이션에서 건식 구동 로봇의 상세를 나타낸다.

[0163] [표 7]

Model	Physical Model	Cosserat Rod Model
	Actuation Range	$-0.4\,\text{N}\cdot\text{m} \leq \tau \leq 0.4\,\text{N}\cdot\text{m}$
	Number of Actuation	2
	Length	120 mm
Material	Density	$1000\ \text{kg/m}^3$
	Young's Modulus	3×10^4 kPa
	Poission's Ratio	0.50

[0164]

[0165] 표 8는 실제 환경에서 건식 구동 로봇의 상세를 나타낸다.

[0166] [표 8]

Model	Actuation Range	$-\pi$ rad $\leq \theta \leq \pi$ rad
	Number of Actuation	2
	Length	120 mm
Material	Body	Ecoflex 00-30
	Tendon Wire	Nitinol, $D = 0.5$mm
	Rigid Parts	Onyx-One, Onyx
	Motor	Dynamixel (XC330-T288-T)

[0167]

[0168] 이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 집합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성 요소는, 예를 들어, 프로세서, 콘트롤러, ALU(Arithmetic Logic Unit), 디지털 신호 프로세서(Digital Signal Processor), 마이크로컴퓨터, FPA(Field Programmable array), PLU(Programmable Logic Unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(Operation System, OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술 분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(Processing Element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서

(Parallel Processor)와 같은, 다른 처리 구성(Processing Configuration)도 가능하다.

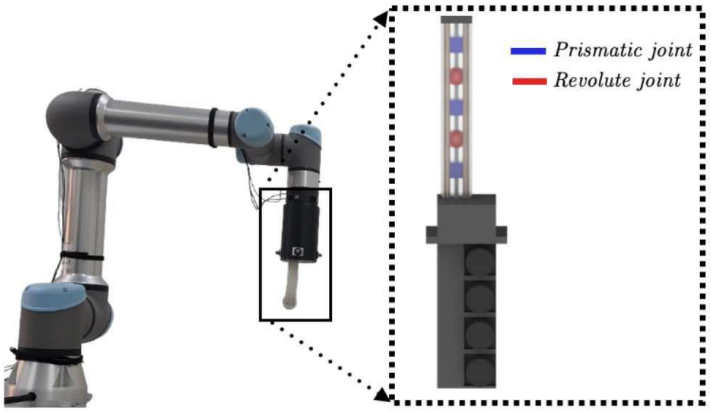
[0169] 소프트웨어는 컴퓨터 프로그램(Computer Program), 코드(Code), 명령(Instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(Collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성 요소(Component), 물리적 장치, 가상 장치(Virtual Equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(Signal Wave)에 영구적으로, 또는 일시적으로 구체화(Embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0170] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM, DVD와 같은 광기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-optical Media), 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

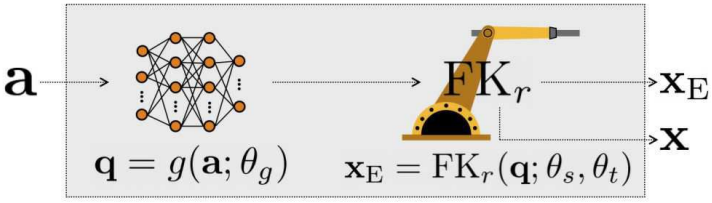
[0171] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성 요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성 요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

도면

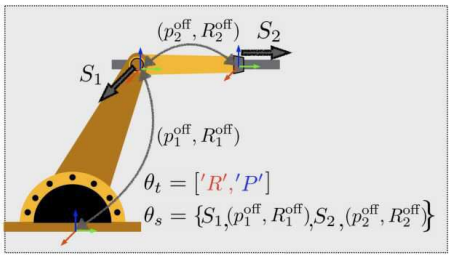
도면1



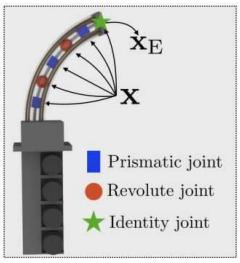
도면2



(a)

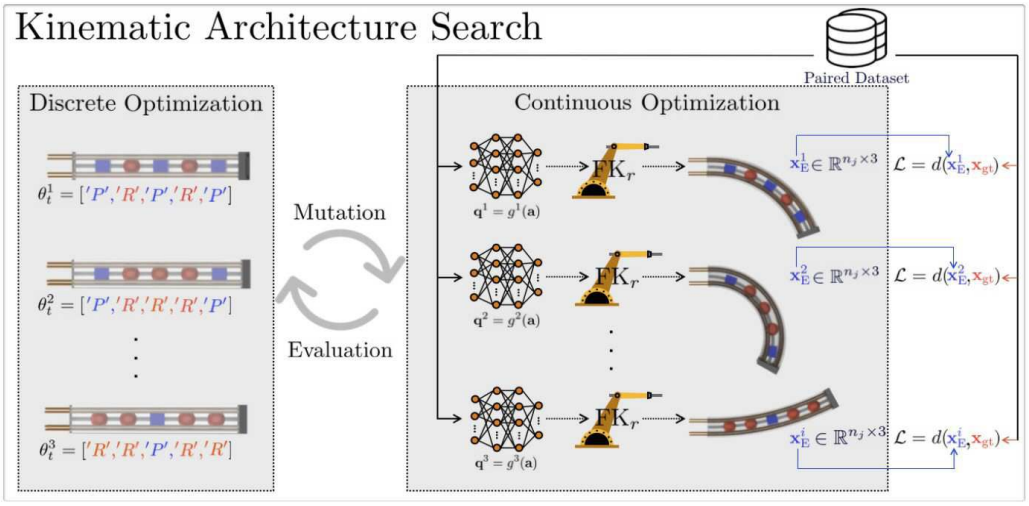


(b)

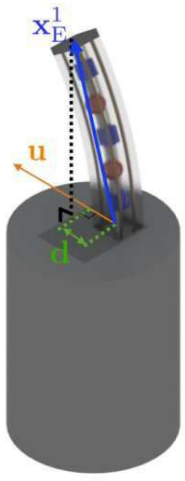


(c)

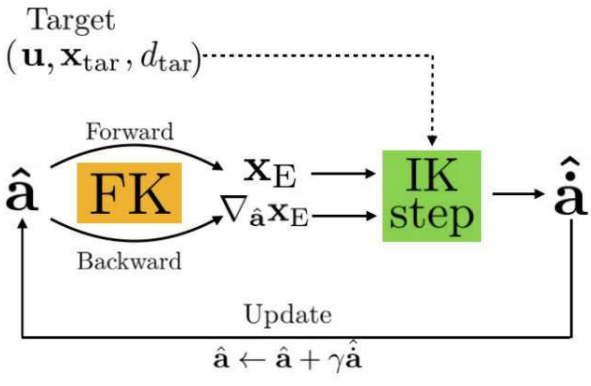
도면3



도면4

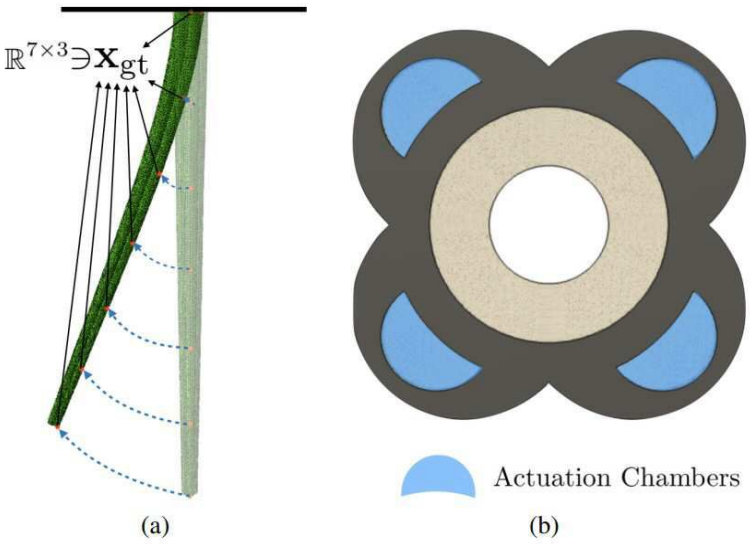


(a)

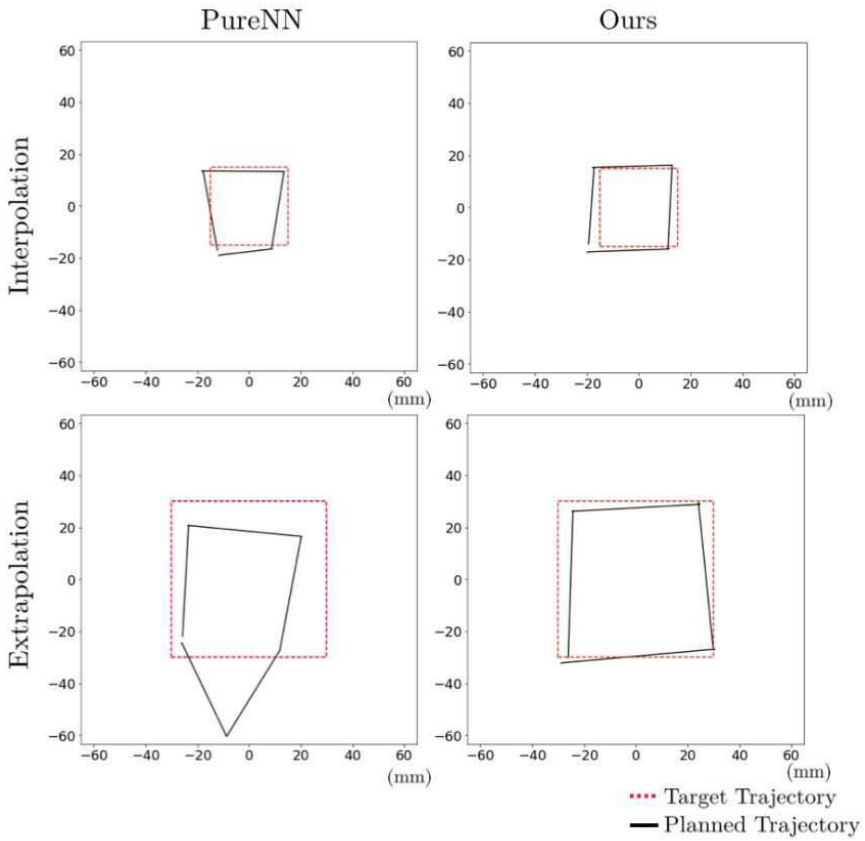


(b)

도면5

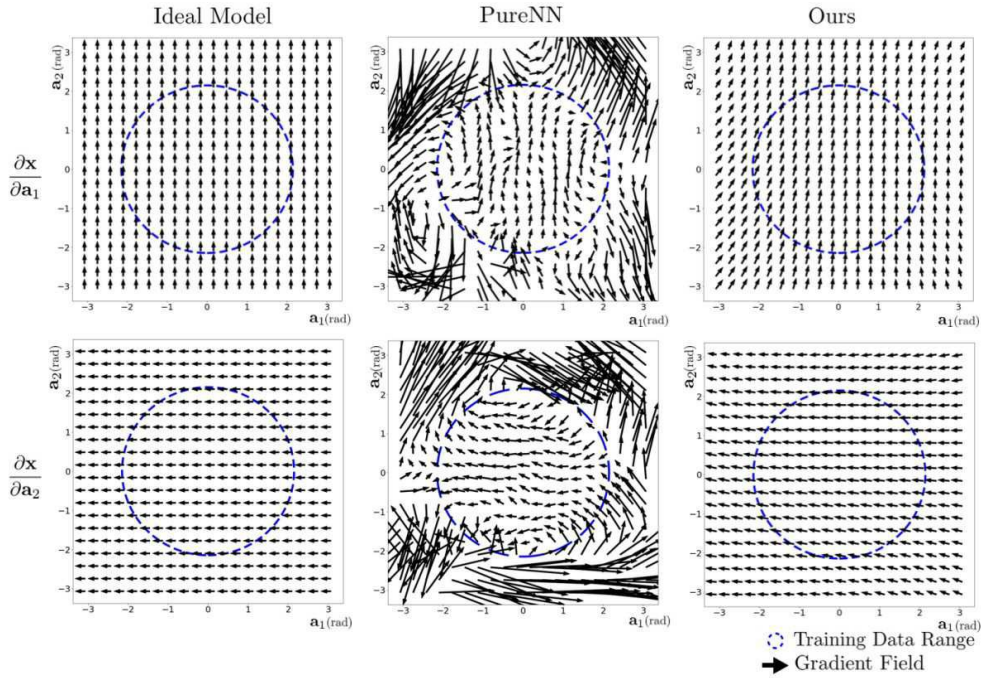


도면6a



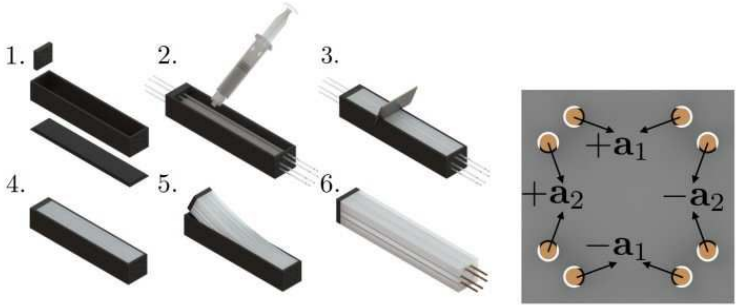
(a) Trajectory Tracking

도면6b

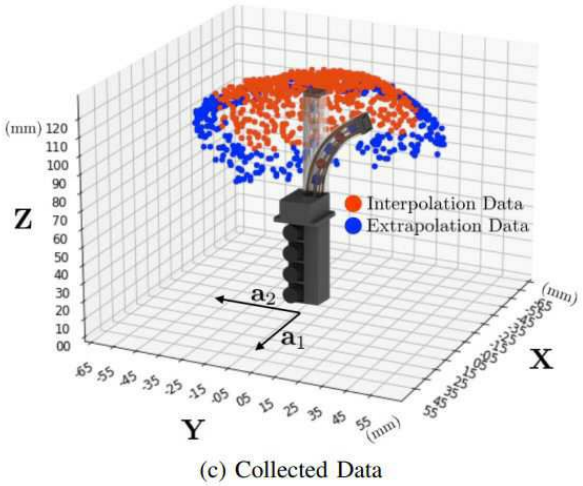


(b) Gradient Field

도면7

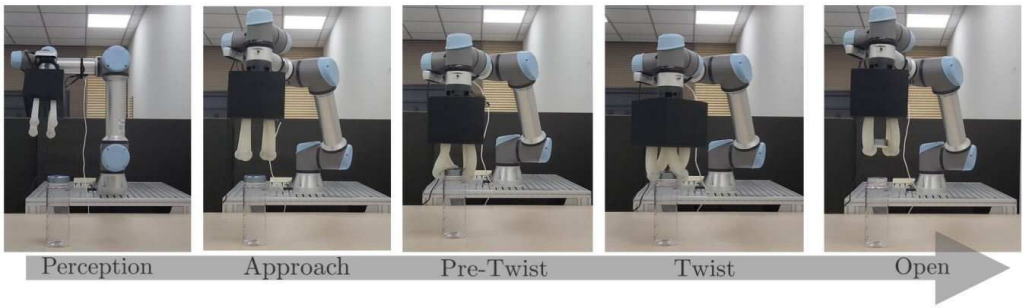


(a) Fabrication (b) Sectional view

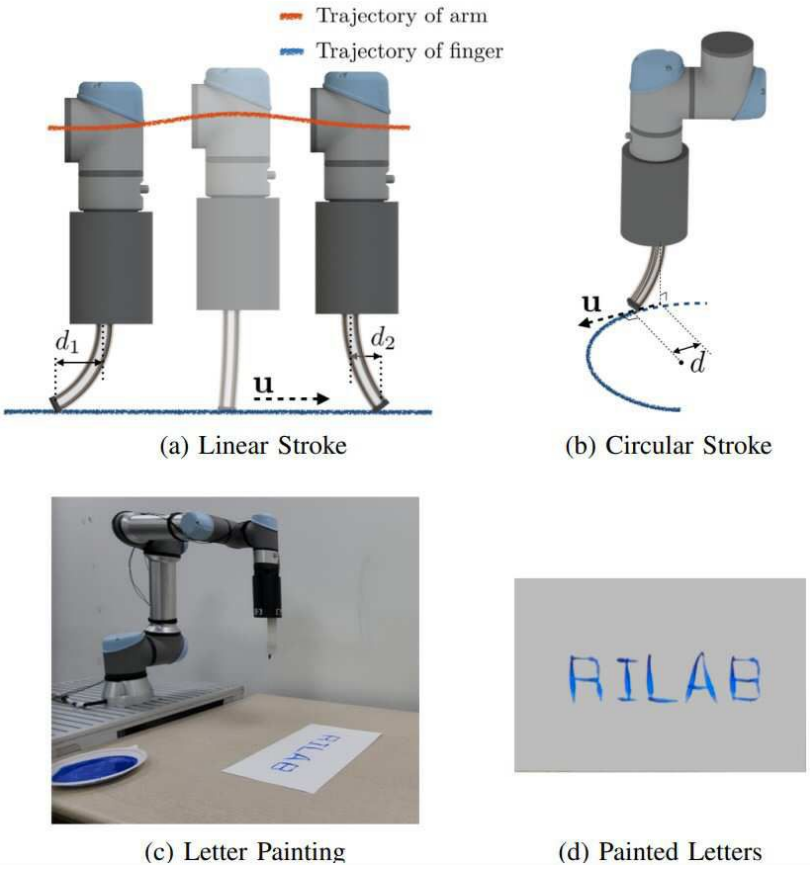


(c) Collected Data

도면8



도면9



(a) Linear Stroke (b) Circular Stroke

(c) Letter Painting (d) Painted Letters