

모터 동역학 기반 실시간 점성 마찰 추정 및 보상을 통한 보행 로봇의 적응 제어

Real-Time Viscous Friction Estimation and Compensation Based on Motor Dynamics for Adaptive Control of Legged Robots

박재형¹·안재훈²·류호균³·강호선²·송화영¹·김민성³·이인호[†]

Jaehyung Park¹, Jaehoon An², Hokyun Ryu³, Hosun Kang², Hwayoung Song¹,
Minseong Kim³, Inho Lee[†]

Abstract: It is a challenging topic for a legged robot to walk in various environments without changing the control parameters because of changes in viscosity depending on the environment. This paper presents an adaptive control technique for legged robots by estimating and compensating for viscous friction in each actuator using motor dynamics. The estimation leverages Recursive Least Squares (RLS) and the Gopinath method, providing a robust approach to determine viscous friction parameters in real time. To increase the estimation accuracy, the robot controller transmits the nonlinear terms generated in the robot to the motor drivers. Robot controller consists of Inverse dynamics-based controller and compensates for estimated viscous friction. To demonstrate the feasibility to robots and adaptability across diverse environments, the controller is tested on various walking speeds and in environments with different viscous friction, such as in water and air.

Keywords: Quadruped Robot, Proprioceptive Actuator, Friction Estimation, Inverse Dynamics Control, Motor Electrodynamics

1. 서론

저감속 고토크 구동기인 Proprioceptive 모터^[1]의 개발과 정교한 모델링 기술은 사족보행 로봇이 보다 빠르고 유연하게 움직

일 수 있는 기반을 마련했다. 또한, Whole-Body Control (WBC) 과 강화학습 기반 제어 기술의 발전은 안정적이고 효율적인 이동을 가능하게 했다^[2-4]. 구동기와 제어 기법의 혁신은 로봇의 기동성과 적응성을 획기적으로 향상시켰으며, 비탈길, 산악 지형, 심지어 물 속과 같은 까다로운 환경에서도 안정적으로 동작할 수 있는 사족보행 로봇의 등장을 이끌었다.

이러한 성과는 놀라운 발전이지만, 여전히 해결해야 할 과제가 존재한다. 환경이 바뀔 때마다 로봇의 제어기 파라미터를 수동으로 조정하거나, 새로운 환경에 맞춰 제어기를 다시 학습시켜야만 효율적인 보행이 가능하다는 점이다. 특히, 로봇에 작용하는 점성 마찰은 환경 변화에 따라 크게 달라질 수 있다. 예를 들어, 로봇이 물속이나 진흙, 젖은 풀밭과 같은 높은 점성을 가지는 지면을 지날 경우, 관절에 작용하는 저항이 공기 중에서 보행할 때보다 현저히 증가하게 된다. 이를 적절히 고려하지 않고 제어기를 조정하지 않으면 보행 안정성이 크게 저하될 위험이 있다. 따라서 환경 변화에 효과적으로 적응할 수 있는 로봇 제어기를 설계하는 것이 중요하다.

Received : Dec. 20. 2024; Revised : Apr. 23. 2025; Accepted : Jun. 26. 2025

※ This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2023-00215760, Guide Dog: Development of Navigation AI Technology of a Guidance Robot for the Visually Impaired Person).

1. MS Student, Department of Electrical and Electronics Engineering, Pusan National University, Busan, Korea (wogud330, dasdas12002@pusan.ac.kr)
2. PhD Student, Department of Electrical and Electronics Engineering, Pusan National University, Busan, Korea (dkswo46, hosun7379@pusan.ac.kr)
3. Undergraduated Student, Department of Electrical and Electronics Engineering, Pusan National University, Busan, Korea (the1513, matize1115@pusan.ac.kr)

† Assistant Professor, Corresponding author: Department of Electronics Engineering, Pusan National University, Busan, Korea (inholee8@pusan.ac.kr)

로봇에서 마찰을 보상하기 위한 접근법으로는 슬라이딩 모드 제어^[5,6], 모델 기준 마찰 추정 기법^[7,8], 오프라인 시스템 식별^[9,10] 등이 제안되어 왔다. 그러나 슬라이딩 모드 제어는 비모델화된 동역학에 민감하며, 기존의 모델 기반 마찰 추정 기법은 사족보행 로봇의 floating base 모델에서는 부정확할 수 있다. 또한, 오프라인 시스템 식별은 실시간으로 변화하는 환경에 대한 마찰 추정을 하지 못한다.

환경 변화에도 안정적이고 효율적인 제어를 실현하려면, 점성 마찰과 같은 환경 의존적인 요소를 실시간으로 추정하고 보상하는 기술이 중요하다. 특히, 이를 통해 제어기의 성능이 특정 환경에 국한되지 않고, 다양한 조건에서도 일관된 결과를 제공할 수 있다. 이는 로봇 제어의 범용성을 확보하는 핵심 요소로, 다양한 지형과 환경에서 추가적인 튜닝 없이도 신뢰할 수 있는 동작을 가능하게 한다. 이는 환경마다 별도의 설정이나 학습 과정을 요구하는 기존의 제한된 제어 방식을 넘어, 보다 보편적이고 확장 가능한 제어 시스템으로 나아가는 데 중요한 역할을 한다.

하지만 로봇 제어기 단에서 점성 마찰을 정교하게 추정하는 것은 여러 가지 측면에서 쉽지 않은 과제다. 먼저, 마찰 추정 알고리즘은 로봇의 전체 동역학과 관절 움직임에 고려해야 하므로 계산 및 구현 관점에서 높은 복잡성을 갖는다. 이러한 복잡성은 실시간 제어 구현 시 큰 부담으로 작용하며, 특히 고성능 하드웨어나 효율적인 알고리즘 설계가 선행되지 않으면 실제 적용이 어렵다. 두번째로, 이러한 추정 방법은 모델링 오차(Modeling Error)에 매우 민감하다. 로봇 구조, 센서 노이즈, 환경 상호작용 등 다양한 요인으로 인해 정확한 동역학 모델을 확보하기란 쉽지 않다. 미세한 모델링 오차도 마찰 추정 결과를 크게 왜곡할 수 있으며, 이는 결국 제어 성능 저하로 이어진다.

본 논문에서는 이러한 문제를 해결하기 위해 모터 동역학을 활용한 새로운 점성 마찰 추정 기법을 제안한다. 이를 통해 각 축 별로 발생하는 점성 마찰을 개별적으로 추정하고, 환경 변화에 따라 달라지는 마찰 특성을 능동적으로 반영함으로써, 기존에 요구되던 복잡하고 전면적인 동역학 파라미터 식별 과정 없이도 다양한 환경에 유연하게 적응할 수 있다. 또한, 모터 드라이버가 알지 못하는 비선형 항(예: 중력, 전향력 등)은 상위 제어기에서 Inverse Dynamics^[11]를 활용하여 추정하며, 이렇게 산출한 마찰 보상 토크를 Inverse Dynamics 기반 제어기에서 계산한 토크와 결합해 모터에 전달한다. 이 접근법은 제어기의 복잡성을 줄이면서도 실시간 환경 적응성과 제어 성능을 극대화한다.

더 나아가, 실시간으로 변화하는 환경을 고려하기 위해 Recursive Least Squares (RLS)^[12]와 Gopinath Method^[13]를 적용하여 점성 마찰과 관성 모멘트를 동시에 추정하는 방식을 제안한다. 제안된 알고리즘은 보행 로봇 다리에 적용되었으며, 추정된 점성 마찰을 보상한 경우와 보상하지 않은 경우를 비교하여, 공기 중과 물속이라는 상이한 환경에서 제어 성능의 차이

를 분석하였다. 그 결과, 다양한 환경에서도 안정적이고 효율적인 제어가 가능함을 확인하였다. 또한, 실제 사족보행 로봇이 x 방향으로 보행할 때 인가되는 보행 궤적을 적용하여, 보행 속도에 따른 제어 성능을 비교 분석하였다. 이를 통해 모터 동역학 기반 점성 마찰 추정 기법이 다양한 환경에서 신뢰성과 적응성을 갖춘 로봇 제어를 가능하게 함을 입증하였다.

2장에서는 본 연구에서 사용된 사족보행 로봇 다리와 모터 드라이버 시스템의 하드웨어 구성은 물론, Inverse Dynamics 기반 제어 기법에 대해서도 함께 설명한다. 3장에서는 모터 드라이버 상에서 관성 모멘트와 점성 마찰을 추정하는 알고리즘을 제안하고, 로봇에 적용하기 위한 제약 조건을 구체화하여 설명한다. 4장에서는 공기 중과 물속이라는 상이한 환경에서 마찰 추정 및 보상 적용 여부에 따른 제어 성능, 보행 속도가 다를 때, 마찰 추정 및 보상 적용 여부에 따른 제어 성능 비교하고 검증한 결과를 제시한다. 마지막으로 5장에서는 본 연구의 결론과 향후 연구 방향을 제안한다.

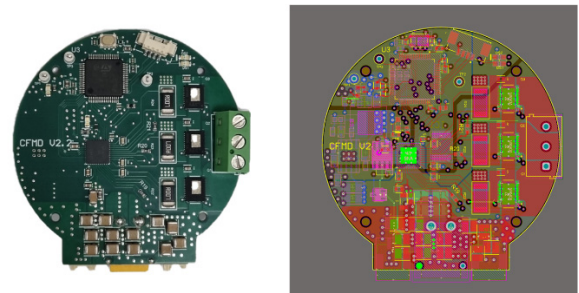
본 연구에서 말하는 점성 마찰은 단순히 모터 내부의 기계적 마찰뿐만 아니라, 링크에서 발생하는 마찰과 외부 환경에 의해 작용하는 점성 계열의 외란까지 포함하는 등가적인 점성 마찰 항을 의미한다. 이는 제어기 설계 및 추정 과정에서 하나의 통합된 마찰 외란으로 간주되어 보상된다.

2. 시스템 설계 및 제어

2.1 하드웨어 시스템

2.1.1 모터 드라이버

본 연구에서는 마찰 추정 알고리즘에 필요한 가속도를 정확히 산출하고, 해당 알고리즘을 드라이버 내부에 직접 내장하기 위해 [Fig. 1]에 제시된 전용 드라이버를 직접 설계하였다. 이 모터 드라이버는 Field Oriented Control (FOC) 기반의 20 kHz 주기 순시 토크 제어를 특징으로 하며, 2 kHz 주기로 각속도

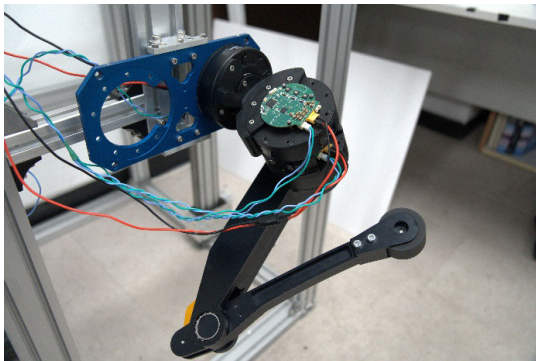


[Fig. 1] The custom-designed motor driver featuring integrated friction estimation, field-oriented torque control, and high-speed FD-CAN communication

를 이동평균 처리한 뒤 이를 통해 각가속도를 계산한다. 또한, 기존 CAN 통신보다 확장된 FD-CAN (Flexible Data-Rate CAN) 방식을 도입하여, 상위 단으로 전달되는 데이터 길이를 12 bytes로 확대하였고, 통신 속도를 4 Mbps로 설정함으로써 더 많은 데이터를 한층 빠르게 전송할 수 있도록 하였다. 아울러 드라이버 내부에는 별도의 속도 제어를 두지 않고, 상위 로봇 PC에서 토크 명령을 직접 수신함으로써 토크 제어를 구현 하도록 설계하였다.

2.1.2 로봇 시스템

사족보행 로봇 CANINE^[14]의 왼쪽 다리를 활용하여 실험을 수행하였다. [Fig. 2]와 같이, 로봇 다리를 프로파일 구조물에 부착하였으며, 로봇의 parameter는 [Table 1]과 같다. [Table 1]은 오프라인 시스템 식별^[9,10] 기반으로 개발된 오픈소스를 이용하여 구하였다. 프로파일에 고정된 상태에서 로봇 다리는 공중에 매달려 있어 지면 반력이 제거되므로, 전체 동역학 식이 간단해진다. 이로 인해 fixed-base를 가정한 제어기 구성이 가



[Fig. 2] Experimental setup showing the left leg of the CANINE quadruped robot mounted on a profile structure, enabling fixed-base experiments and simplifying the control environment

[Table 1] Robot System Parameter

| Parameter | Link1 | Link2 | Link3 |
|---------------------------------|--|--|--|
| Joint Description | Hip-Roll | Hip-Pitch | Knee-Pitch |
| Mass (kg) | 0.767435 | 1.344041 | 0.269534 |
| CoM position(m) (x, y, z) | (0.059002, -0.016006, -0.000286) | (-0.000582, 0.057164, -0.062631) | (0.011174, -0.010182, -0.149690) |
| I_{xx} (kg · m ²) | 0.00736 | 0.014341 | 0.008471 |
| I_{yy} (kg · m ²) | 0.003728 | 0.009713 | 0.008497 |
| I_{zz} (kg · m ²) | 0.003574 | 0.006191 | 0.000101 |
| I_{xy} (kg · m ²) | -0.000004 | 0.005502 | -0.000402 |
| I_{xz} (kg · m ²) | 0.000013 | -0.000188 | 0.000451 |
| I_{yz} (kg · m ²) | 0.000725 | 0.000124 | 0.000031 |

능하며, 불필요한 외력 고려 없이 내부 관절 동작에 집중할 수 있다. 이러한 단순화는 제어기의 설계 및 검증을 용이하게 하며, 다음 2.2장에서는 이 단순화된 환경을 활용한 제어기 설계 및 적용 방법을 구체적으로 다룬다.

2.2 제어 시스템

고정된 시스템(fixed-base)의 3축 로봇 동역학 모델은 식 (1)로 표현된다.

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau + B\dot{q} \tag{1}$$

여기서, $M(q)$ 는 로봇의 관성 행렬, q 는 관절 변위 벡터, \dot{q} 와 \ddot{q} 는 각 관절의 속도 및 가속도 벡터를 의미한다. 또한, $C(q, \dot{q})$ 는 코리올리 항 행렬, $G(q)$ 는 중력 벡터, τ 는 관절의 토크, B 는 각 관절의 점성 마찰 계수를 나타낸다.

2.2.1 Inverse Dynamics 기반 제어 시스템

위 동역학 식 (1)에서 \ddot{q} , τ , 그리고 B 를 제외한 나머지 항들은 제어기에서 실시간으로 알 수 있는 값이며, 본 연구에서는 Rigid Body Dynamics Library (RBDL)를 활용하여 이를 계산하였다. 점성 마찰 계수 B 는 3절에서 나오는 마찰 추정 알고리즘을 통해 결정한다. 나머지 항은 $X = [\ddot{q}, \tau]^T$ 로 정의한 뒤, qpOASES 오픈 소스를 활용한 Quadratic Programming (QP) 기반 최적화 기법을 통해 구한다. 최적화 기법을 이용하여 구한다. 최적화 문제는 아래 식 (2)와 같이 정의했다.

$$\min_X \|\ddot{q} - \ddot{q}_{des}\|_{w_q} + \|\tau\|_{w_\tau} \tag{2}$$

이때 로봇 동역학은 equality constraint로, 관절의 초기 조건과 관절 토크 상한과 하한은 inequality constraint로 고려할 수 있다. 그러나 이렇게 최적화만으로 구한 τ 는 로봇 상태를 실시간으로 반영하지 않는 단순 feedforward 성격을 가지므로, 모델 오차(Model Error)가 존재할 경우 시스템 안정성과 추종 정확도가 떨어질 수 있다.

이러한 모델 오차의 영향을 최소화하기 위해 본 연구에서는 PD 제어기를 통한 피드백 요소를 추가하여 모델 기반 제어의 단점을 보완하는 방식을 채택하였다. 즉, 최적화 과정 전 Trajectory에서 생성된 원하는 가속도 \ddot{q}_{des} 에 PD 피드백 항을 더해주는 형태로, 식 (3)과 같이 위치 및 속도 오차에 기반한 보상 항을 추가한다.

$$\ddot{q}_{des} = \ddot{q}_{ref} + K_p(q_{des} - q) + K_d(\dot{q}_{des} - \dot{q}) \tag{3}$$

여기서, \ddot{q}_{ref} 는 원하는 레퍼런스 가속도, K_p 와 K_d 는 각각 위치 및 속도 오차를 보상하는 게인이다.

추가로, 위 Inverse-Dynamics 기반 제어기로 구한 τ 에 PD 제어기로부터 나온 토크 명령 τ_{pd} 를 식 (4)와 같이 직접 더함으로써 모델 불확실성의 영향을 줄이고 시스템의 안정성을 강화한다.

$$\tau_{control} = \tau + \tau_{pd} \quad (4)$$

$\tau_{control}$ 은 각 관절에 전달되는 최종 제어 토크이다. 이를 통해 모델 기반 제어의 장점과 PD 피드백을 통한 모델 불확실성 보완 효과를 동시에 얻을 수 있다.

3. 점성 마찰 계수 추정

3.1 모터 동역학을 이용한 점성 마찰 추정

FOC 제어를 사용하는 PMSM의 한 축 기계적 동역학은 식 (5)와 같이 나타낼 수 있다.

$$J \frac{d\omega}{dt} = K_t i_q - B\omega - \tau_L \quad (5)$$

여기서 J 는 모터의 관성, B 는 점성 마찰 계수, ω 는 각속도, K_t 는 토크 상수, 그리고 τ_L 은 부하 토크를 나타낸다. 이 가운데 J , B , 그리고 τ_L 은 모터 드라이버가 직접적으로 알 수 없는 변수이며, 본 연구에서는 Recursive Least Squares (RLS)와 Gopinath Method를 활용한 최적화 접근법으로 이를 추정하고자 한다.

그러나 τ_L 에는 중력, 코리올리 등 식 (1)에서 정의된 로봇 링크 동역학에 따른 비선형 항들이 포함되어 있으며, 이는 모터 단에서 관측 가능한 정보만으로 정확한 추정이 어렵다. 즉, 모터는 다른 관절 및 링크와의 상호작용으로 인한 비선형 항을 직접 파악할 수 없다. 이러한 한계를 극복하기 위해, 본 연구에서는 RBDL를 활용해 활용하여 상위 제어기에서 해당 비선형 항들을 계산하였다.

3.1.1 Recursive Least Squares

식 (5)을 Recursive Least Squares로 알고리즘에 적용하기 위해, 추정 벡터와 입력 데이터는 식 (6)-(8)과 같이 정의한다.

$$y_k = \tau_k \quad (6)$$

$$X_k = \begin{bmatrix} \dot{\omega}_k \\ \omega_k \end{bmatrix} \quad (7)$$

$$\theta_k = \begin{bmatrix} J \\ B \end{bmatrix} \quad (8)$$

여기서, k 는 현재 제어 상태(또는 현재 시점의 샘플 인덱스)를 나타내며, RLS 알고리즘은 시점 k 에서의 관측 데이터를 바탕으로 θ_k 를 갱신하여 J 와 B 를 추정한다. RLS의 처리 과정은 크게 오차 계산, 이득 계산, 추정 벡터 갱신, 공분산 행렬 갱신의 네 단계로 이루어진다.

첫 번째 단계인 오차 계산에서는 현재 출력 데이터(y_k)와 식 (9)과 같이 현재 입력 데이터(X_k)와 이전 추정 벡터 기반으로 추정된 출력 \hat{y}_k 사이의 차이를 계산한다.

$$\hat{y}_k = X_k^T \theta_{k-1} \quad (9)$$

이 오차는 이전 추정 벡터(θ_{k-1})가 얼마나 잘 설명하는지 알려주는 지표이며, 수식은 식 (10)과 같다.

$$e_k = \hat{y}_k - X_k^T \theta_{k-1} \quad (10)$$

RLS는 이렇게 구한 오차(e_k)를 줄이는 방향으로 매개변수를 갱신한다.

두 번째로 이득(K_k)을 계산한다. 이득(K_k)은 RLS가 현재 데이터를 얼마나 반영할지 결정하는 값으로 아래 식 (11)과 같다.

$$K_k = \frac{P_{k-1} X_k}{\lambda + X_k^T P_{k-1} X_k} \quad (11)$$

여기서 P_{k-1} 은 공분산 행렬, λ 는 망각 인자이다. 공분산 행렬 (P_{k-1})은 대각행렬로 초기 값을 기준으로 매 스텝마다 갱신한다. 망각인자(λ)는 과거 데이터의 중요도와 관련된 변수로, 일반적으로 0.9~1 사이의 값을 나타낸다. 망각인자가 낮을수록 최근 데이터의 가중치가 높으며, 시스템 반응성이 높아진다. 이득(K_k)은 현재 데이터가 기존 추정된 데이터에 미치는 영향을 조정한다. 공분산이 작고, 오차가 클수록 더 큰 이득이 할당되어 매개변수를 더 많이 수정한다. 이를 통해 시스템 추정에 즉각적으로 반영할 수 있다.

세 번째 과정은 추정 벡터 갱신이다. RLS는 오차(e_k)와 이득(K_k)을 활용해 식 (12)과 같이 현재 추정 벡터(θ_k)를 업데이트한다.

$$\theta_k = \theta_{k-1} + K_k e_k \quad (12)$$

이로써 새로운 관측 값과 오차 정보를 추정 벡터에 반영하여, 매개변수가 실제 시스템 값에 점진적으로 수렴하도록 한다. 마지막으로, 식 (13)에 따라 공분산 행렬(P_k)을 갱신한다.

$$P_k = \frac{1}{\lambda} (P_{k-1} - K_k X_k^T P_{k-1}) \quad (13)$$

공분산 행렬 갱신은 과거 데이터 영향력을 점진적으로 줄이고, 새로 유입되는 데이터의 신뢰도를 반영함으로써 추정 성능을 향상시킨다. 이러한 절차를 반복함으로써 RLS 알고리즘은 실시간으로 θ_k 를 갱신하며, 결국 $\theta_k = [J, B]^T$ 는 실제 시스템 파라미터에 수렴하게 된다.

3.1.2 Gopinath Method

Gopinath Method는 시스템의 동적 모델과 실제 측정 데이터를 비교하여 Residual을 계산하고, 이를 기반으로 상태와 파라미터를 보정하는 방법이다. 이를 위해 식 (5)의 모터 동역학을 상태-공간 방정식으로 표현하면 식 (14)과 같다.

$$\dot{w}(t) = Aw(t) + B\tau(t), \left(\text{where } A = -\frac{B}{J}, B = \frac{1}{J} \right) \quad (14)$$

여기서, RLS를 통해 추정된 θ_k 를 이용해 A, B 를 계산한다. Gopinath 관찰기는 상태-공간 모델을 기반으로 정의되며, 시스템의 상태와 출력을 다음과 같이 식 (15), (16)의 형태로 예측한다.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \quad (15)$$

$$\hat{y}(t) = C\hat{x}(t) + Du(t) \quad (16)$$

여기서, $\hat{x}(t)$ 은 추정된 상태 벡터, $\hat{y}(t)$ 는 추정된 출력 벡터, L 은 관찰기 이득이다. 본 연구에서 출력 방정식과 입력 방정식을 동일하게 설정하였기 때문에, $\hat{x}(t)$ 과 $\hat{y}(t)$ 모두 상태-공간 방정식에서 추정된 각가속도를 의미한다.

Residual $r(t)$ 는 식 (17)과 같이 실제 출력, $y(t)$ 와 관찰기가 추정한 출력 $\hat{y}(t)$ 의 차이로 정의된다.

$$r(t) = y(t) - \hat{y}(t) \quad (17)$$

$r(t)$ 는 상태 추정의 정확성을 평가하는 지표로, 이를 최소화하기 위해 관찰기 이득 L 을 조정하여 관측기의 수렴 속도를 제어한다. 관찰기의 오차는 아래 식 (18)-(20)와 같이 정의할 수 있다.

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) \quad (18)$$

$$\dot{e}(t) = Ae(t) - Lr(t) \quad (19)$$

$$\dot{e}(t) = (A - LC)e(t) \quad (20)$$

여기서, 식 (20)의 시스템 고유 값이 1보다 조금 작은 값이 되게 L 을 조정한다. RLS는 보정된 각가속도를 입력으로 사용하여 현재 추정 벡터(θ_k)를 갱신한다.

3.2 로봇에 적용하기 위한 제약조건

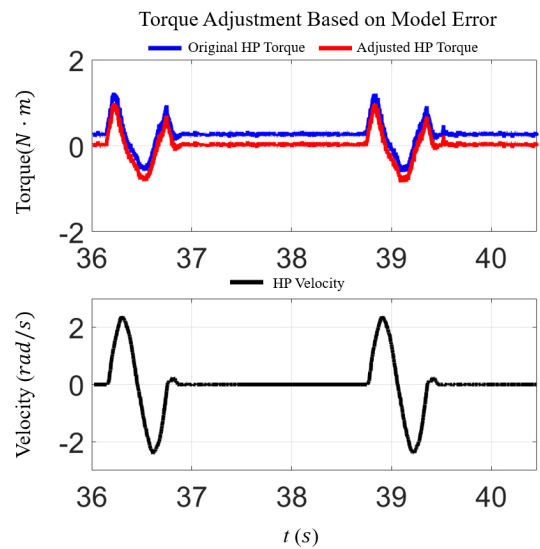
제한된 마찰 추정 알고리즘은 모델 기반 제어(Model-based Control) 구조 내에서 사용되며, 로봇의 동역학 모델 정확도에 따라 제어기 성능뿐만 아니라 추정 결과에도 영향을 받을 수 있다. 특히, 중력 항의 불일치와 같은 모델 오차(model error)는 마찰 추정 알고리즘에 부정적인 영향을 미칠 수 있으며, 정지 상태에서 PD 제어기의 feedback 항으로 인해 발생하는 토크 또한 추정 정확도에 영향을 줄 수 있는 요인으로 작용한다. 정지 상태에서는 속도와 가속도가 거의 0에 가까운 반면 모델 오차에 의해 PD 제어기에서 토크가 발생할 수 있으며, 이러한 상황에서 마찰 추정 알고리즘은 해당 토크를 관성과 마찰에 의한 것으로 잘못 해석하여 해당 값을 과대 추정할 가능성이 있다. 본 연구에서는 이러한 영향을 완화하기 위해, [Fig. 3]과 같이 hip pitch (HP) 관절을 대상으로 정지 상태에서 발생하는 토크를 기준점으로 보정하여 마찰 추정 알고리즘의 입력에서 해당 오프셋을 제거하는 방식을 적용하였다.

추가적으로, 로봇에서 위 마찰 추정 알고리즘을 적용하기 위해서 2가지의 Inequality constraints를 추가하였다. 첫번째로는 점성 마찰 계수(B)와 관성 모멘트(J)의 변화율을 식 (21), (22)과 같이 제한을 두었다.

$$\Delta J \leq 0.01 \quad (21)$$

$$\Delta B \leq 0.00001 \quad (22)$$

위와 같은 제약 조건을 적용한 이유는, 로봇에서 관성 모멘



[Fig. 3] Comparison of original torque and adjusted torque after compensating for model error in the hip pitch (HP) motor, alongside the corresponding hip pitch velocity over time

트가 급격하게 변하는 반면 점성 마찰 계수는 상대적으로 덜 변화하기 때문이다. 이러한 제약 조건을 통해 추정과정의 안정성을 높이고, 급격한 매개변수 변화로 인해 발생할 수 있는 추정 오차를 줄였다.

본 연구에서는 각속도 ω 가 0에 근접한 구간에서는 추정 파라미터 θ_k 를 업데이트하지 않고, 이전 값을 유지하도록 설정하였다. 이는 마찰 계수 B 및 관성 항 J 을 추정하는 모터 다이내믹스 회귀식

$$\tau = [\dot{\omega} \omega] \begin{bmatrix} J \\ B \end{bmatrix} \quad (23)$$

에서, 각속도 또는 가속도가 0에 가까워질 경우 회귀 행렬의 감도가 떨어져 수치적으로 ill-conditioned 상태에 가까워지기 때문이다. 본 연구에서는 이러한 수치적 문제를 방지하기 위해, $|\omega| \leq 0.3 \text{ rad/s}$ 을 만족하는 구간에서는 파라미터 업데이트를 수행하지 않는다.

4. 마찰 추정 및 보상 실험 및 결과 비교

4.1 공기 중과 물속 환경에서의 비교 실험

공기 중과 물 속에서 로봇에게 마찰 추정 알고리즘을 적용하여 실험을 수행하였다. 로봇의 제어 및 추정기의 파라미터들은 [Table 2]와 같이 설정하였다. 이 실험에서는 공기 중과 물속에서 각각의 Hip pitch와 Knee pitch 모터의 마찰 및 관성 모멘트를

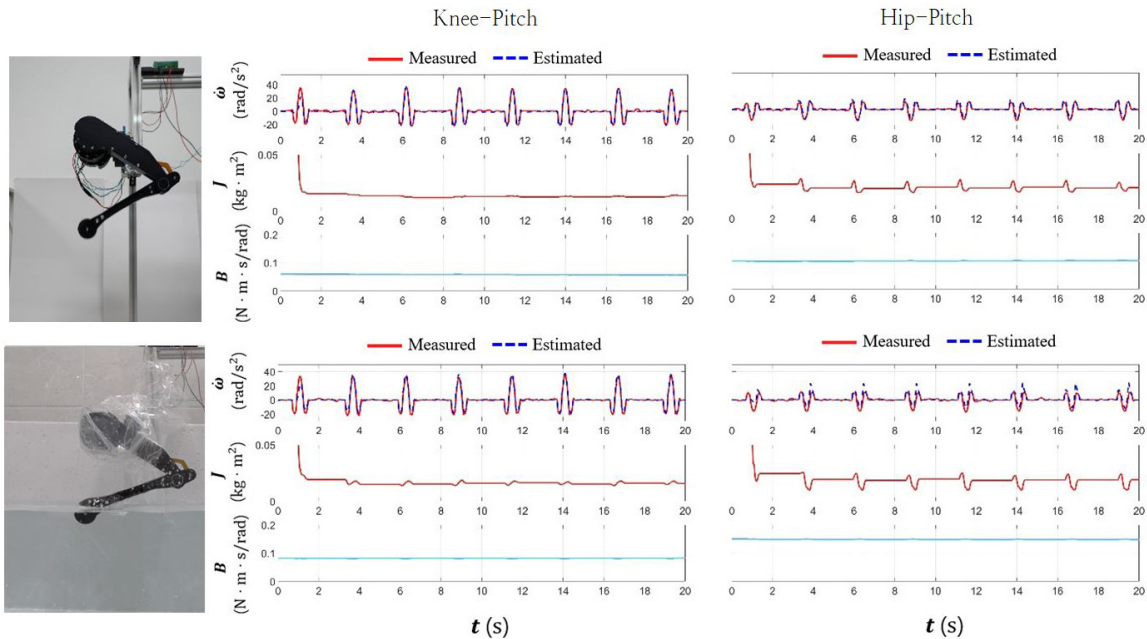
추정하였다. 정확한 비교를 위해 물 속과 공기 중에서 동일한 보행 Trajectory를 적용하였다.

실험 결과는 [Fig. 4]와 [Table 3]에 제시되어 있으며, 모든 상황에서 추정된 데이터를 바탕으로 계산한 각속도가 실제 측정된 각속도와 유사함을 확인할 수 있었다. 이를 통해 3축 로봇 다리가 swing leg trajectory 동안 마찰 및 관성 모멘트를 정확히 추정할 수 있음을 보여준다.

추가적으로, [Table 3]를 살펴보면 공기 중보다 물속에서 점성 마찰 계수가 상대적으로 더 크게 나타났으며, Knee Pitch보다 Hip Pitch 모터에서 점성 마찰 계수가 더 높은 것으로 확인되었다. 이러한 결과는 실험 환경과 로봇 관절의 특성에 따라 마찰이 달라지는 것을 시사하며, 추정 알고리즘이 다양한 환경에서도 유효하게 작동함을 입증한다.

[Table 2] Control and Estimator Parameter

| Parameter | Value |
|-----------------------------------|--|
| Swing Period | 0.6 s |
| LiftUp Height | 0.15 m |
| Task PD gain (K_P, K_D) | 700, 40 |
| Swing PD gain (K_P, K_D) | 5, 2 |
| Stand PD gain (K_P, K_D) | 0, 2 |
| Forgetting Factor (λ) | 0.99 |
| Initial Covariance (P_{init}) | $\begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$ |
| Initial Inertia (J_{init}) | 0.1 kg · m ² |



[Fig. 4] Comparison of measured accelerations and estimated accelerations based on identified J and B , along with the time evolution of estimated inertia J and viscous friction coefficient B for knee pitch and hip pitch joints in air and water environments

[Table 3] Comparison of Estimated Viscous Friction Coefficients in Air and Water

| | | |
|----------|----------|---------------------------------|
| In air | B_{KP} | $0.052 N \cdot m \cdot s / rad$ |
| | B_{HP} | $0.112 N \cdot m \cdot s / rad$ |
| In water | B_{KP} | $0.092 N \cdot m \cdot s / rad$ |
| | B_{HP} | $0.156 N \cdot m \cdot s / rad$ |

다음으로, 실시간으로 구한 점성 마찰 계수를 식 (24)같이 제어기에서 보상하였다.

$$\tau_{control} = \tau + \tau_{pd} + B\omega \quad (24)$$

제어기의 정성적 평가 지표를 Tracking Error Integral (TEI) 와 Correlation Coefficient로 확인하였다. 각 지표는 아래의 식 (25)와 (26)으로 정의된다.

$$TEI = \int_{t_0}^{t_{end}} |e(t)| dt \quad (25)$$

$$r = \frac{Cov(X, Y)}{\sigma_X \sigma_Y} \quad (26)$$

이 평가는 로봇이 스윙(swing)과 스탠드(stand) 동작을 2번 반복할 때 수행되었으며, 결과는 [Table 4]에 제시되어 있다.

[Table 4] Comparison of Compensation Effects in Air and Water

| Parameter | | No Comp | Comp |
|-----------|----------------|---------|--------|
| In air | KP TEI | 0.0585 | 0.0197 |
| | HP TEI | 0.0192 | 0.0206 |
| | KP Correlation | 0.9995 | 0.9997 |
| | HP Correlation | 0.9993 | 0.9995 |
| In water | KP TEI | 0.1783 | 0.0556 |
| | HP TEI | 0.0513 | 0.0257 |
| | KP Correlation | 0.9982 | 0.9992 |
| | HP Correlation | 0.9992 | 0.9995 |

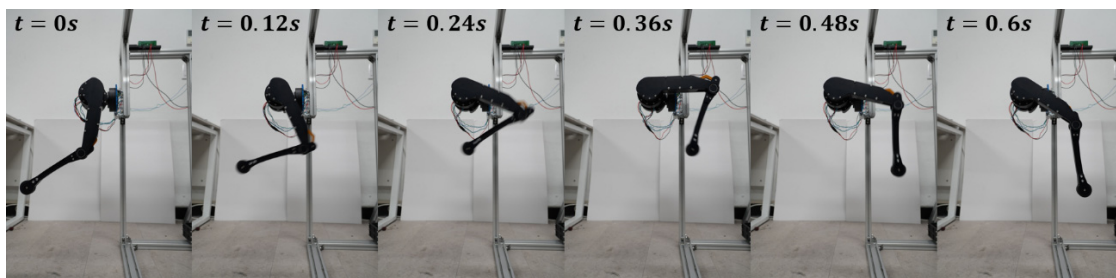
먼저, 공기 중에서 KP에서는 모든 지표에서 성능이 개선된 것을 알 수 있다. 특히, TEI가 0.0585에서 0.0197로 2배 이상 개선된 것을 볼 수 있다. 반면, HP에서는 TEI 값이 약간 높아졌으나 기존 값과 큰 차이가 나지 않았으며, 기존 값 자체가 이미 낮은 수준임을 알 수 있다. 또한, Correlation Coefficient에서는 HP에서도 성능이 개선된 것을 확인할 수 있다. 물 속에서는 KP와 HP가 모든 지표에서 개선된 것을 볼 수 있는데, 공기 중에서 보다 더 큰 폭으로 개선된 것을 알 수 있다.

4.2 보행 시 보상 전후 비교 실험

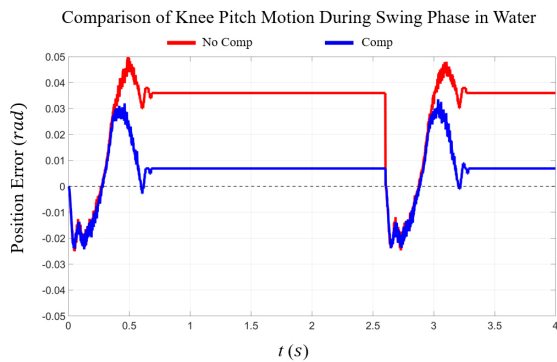
[Fig. 5]는 공기 중에서 보행 시 x 방향으로 속도가 있을 경우 마찰 추정 및 보상 실험을 하였다. 실험은 4.1절에서와 동일한 조건에서 진행되었으며, 속도는 $0 m/s$, $0.5 m/s$, $1 m/s$, $1.5 m/s$ 에서 진행하였고, 각각 마찰 보상을 적용한 경우와 하지 않았을 경우의 TEI를 비교하였다. 그 결과 [Table 5]에 정리되어 있다. 이를 통해 마찰 보상이 적용된 경우, 궤적 추종 오차가 감소하는 경향을 보였다. 또한, [Fig. 6]은 물속에서 swing leg의 knee pitch 관절을 대상으로 위치 오차를 비교한 결과로, 마찰 보상 전후의 성능 차이를 시각적으로 확인할 수 있다. 해당 그래프에서는 마찰 보상이 적용된 경우 오차 크기가 전반적으로 감소하였으며, 이는 제안된 보상 기법이 궤적 추종 성능을 향상시킬 수 있음을 보여준다.

[Table 5] Comparison of Compensation Effects at Different Walking Speeds in Air

| Parameter | | No Comp | Comp |
|-----------|--------|---------|--------|
| $0 m/s$ | KP TEI | 0.0585 | 0.0197 |
| | HP TEI | 0.0192 | 0.0206 |
| $0.5 m/s$ | KP TEI | 0.0497 | 0.0175 |
| | HP TEI | 0.0197 | 0.0204 |
| $1 m/s$ | KP TEI | 0.0587 | 0.0292 |
| | HP TEI | 0.0317 | 0.0236 |
| $1.5 m/s$ | KP TEI | 0.0757 | 0.0307 |
| | HP TEI | 0.0356 | 0.0283 |



[Fig. 5] Swing trajectory of a single leg at $V_x = 1.5 rad/s$: Comparison of the robot's performance with and without friction estimation and compensation



[Fig. 6] Comparison of knee pitch joint position error of the swing leg in water

5. 결 론

본 연구는 보행 로봇 구동기의 관성과 마찰을 실시간으로 추정하여 다양한 환경에서도 적응적으로 제어할 수 있는 시스템을 개발하는 데 중점을 두었다. 로봇의 전체 동역학에서 점성 마찰 계수를 추정하는 데 어려움이 있음을 고려하여, 모터 동역학을 활용해 개별 드라이버 단에서 Recursive Least Squares (RLS)와 Gopinath Method를 적용하여 마찰을 추정하는 방식을 제안하였다. 특히, 드라이버에서 알 수 없는 로봇의 비선형 항(예: 중력, 코리올리 항)은 상위 제어기에서 계산한 값을 받아 활용하였다. 제안된 알고리즘은 3축 로봇 다리를 대상으로 실험을 진행하였으며, 물속과 공기 중이라는 상이한 환경에서 검증되었다. 또한, 보행 속도에 따라 변화하는 궤적에서 마찰 보상 알고리즘의 성능을 평가한 결과, 궤적 추종 성능이 유의미하게 개선됨을 확인할 수 있었다. 이러한 결과는 제안된 알고리즘이 다양한 환경에서 로봇 제어의 안정성과 효율성을 향상시키는 데 기여할 수 있음을 보여준다.

향후 연구에서는, 추정된 마찰 계수 및 관성 모멘트의 정량적 정확도를 시뮬레이터 연동을 통해 검증하고, 점성 마찰 항을 개별적으로 분리하여 각각의 보상 효과를 정량적으로 비교 분석하는 실험을 진행할 계획이다. 또한, 제안된 마찰 추정 및 보상 기법을 전체 사족보행 로봇 시스템의 Whole-Body Control (WBC) 제어기 구조에 통합 적용하여, 실환경에서의 보행 안정성 및 전신 제어 관점에서의 성능 향상 가능성을 검토하고자 한다.

References

- [1] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509-522, Jun., 2017, DOI: 10.1109/TRO.2016.2640183.
- [2] S. Feng, "Online Hierarchical Optimization for Humanoid Control. Diss," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, 2016, DOI: 10.1184/R1/6720938.V1.
- [3] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," *2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, Orlando, FL, USA, pp. 2641-264, 2006, DOI: 10.1109/ROBOT.2006.1642100.
- [4] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, Jan., 2019, DOI: 10.1126/scirobotics.aau5872.
- [5] G. Zepeda and L. Pantoja-Garcia, "Robust force-position control of a novel DAE model for position servo-actuated robot manipulators," *International Journal of Dynamics and Control*, Apr., 2025, DOI 10.1007/s40435-025-01706-6.
- [6] D. Ossadnik, J. Rogelio Guadarrama-Olvera, E. Dean-Leon, and G. Cheng, "Adaptive Friction Compensation for Humanoid Robots without Joint-Torque Sensors," *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, Beijing, China, pp. 980-985, 2018, DOI: 10.1109/HUMANOID.2018.8624965.
- [7] B. S. R. Armstrong, "Dynamics for robot control: friction modeling and ensuring excitation during parameter identification," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 1988, [Online], <https://dl.acm.org/doi/10.5555/914465>.
- [8] L. Le Tien, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Friction Observer and Compensation for Control of Robots with Joint Torque Measurement," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 3789-3795, 2008, DOI: 10.1109/IROS.2008.4651049.
- [9] T. Lee, P. M. Wensing and F. C. Park, "Geometric Robot Dynamic Identification: A Convex Programming Approach," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 348-365, Apr., 2020, DOI: 10.1109/TRO.2019.2926491.
- [10] C. Rucker and P. M. Wensing, "Smooth Parameterization of Rigid-Body Inertia," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2771-2778, Apr., 2022, DOI: 10.1109/LRA.2022.3144517.
- [11] F. Caccavale and F. Pierri, "Multiple Unmanned Aerial Manipulator Systems, Coordinated Control of," *Encyclopedia of Robotics*, Springer, 2020, pp. 1-8, DOI: 10.1007/978-3-642-41610-1_81-1.
- [12] S. A. U. Islam and D. S. Bernstein, "Recursive Least Squares for Real-Time Implementation [Lecture Notes]," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 82-85, Jun., 2019, DOI: 10.1109/MCS.2019.2900788.
- [13] R. Sepulchre, T. Devos, F. Jadot and F. Malrait, "Antiwindup Design for Induction Motor Control in the Field Weakening Domain," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 52-66, Jan., 2013, DOI: 10.1109/TCST.2011.2173495.
- [14] H. Kang, J. An, H. Cha, W. Ahn, H. Song, and I. Lee, "Research on Stability of Control for Quadruped Robot with Robust Leg Structure Design," *The Journal of Korea Robotics Society*, vol. 18, no. 2, pp. 172-181, May, 2023, DOI: 10.7746/jkros.2023.18.2.172.



박재형

2023 부산대학교 전자공학과(학사)
2023~현재 부산대학교 전기전자공학과
(석사)

관심분야: 로보틱스, 보행로봇, 모터 제어



송화영

2022 부산대학교 전자공학과(학사)
2023~현재 부산대학교 전기전자공학과
(석사)

관심분야: 로보틱스, 보행로봇



안재훈

2022 부산대학교 전자공학과(학사)
2022~현재 부산대학교 전기전자공학과
(석박통합과정)

관심분야: 로보틱스, 기계 학습, 보행로봇



김민성

2019~현재 부산대학교 전자공학과(학사)

관심분야: 로보틱스, 보행로봇, 모델 제어



류호균

2019~현재 부산대학교 전자공학과(학사)

관심분야: 로보틱스, 임베디드, 모터 제어



이인호

2009 한국과학기술원 기계공학과(학사)
2011 한국과학기술원 기계공학과(석사)
2016 한국과학기술원 기계공학과(박사)
2020~현재 부산대학교 전자공학과 조교수

관심분야: 로보틱스, 기계시스템 자동화, 보행로봇



강호선

2017 부산대학교 항공우주공학과(학사)
2019 부산대학교 전기전자공학과(석사)
2019~현재 부산대학교 전자전기공학과
(박사)

관심분야: 로보틱스, 보행로봇, 딥러닝, 환경 인식