

휠-다리 로봇의 장애물극복 모션 계획 및 제어 방법

Motion Planning and Control of Wheel-legged Robot for Obstacle Crossing

정순규¹·원문철[†]

Soonkyu Jeong¹, Mooncheol Won[†]

Abstract: In this study, a motion planning method based on the integer representation of contact status between wheels and the ground is proposed for planning swing motion of a 6×6 wheel-legged robot to cross large obstacles and gaps. Wheel-legged robots can drive on a flat road by wheels and overcome large obstacles by legs. Autonomously crossing large obstacles requires the robot to perform complex motion planning of multi-contacts and wheel-rolling at the same time. The lift-off and touch-down status of wheels and the trajectories of legs should be carefully planned to avoid collision between the robot body and the obstacle. To address this issue, we propose a planning method for swing motion of robot legs. It combines an integer representation of discrete contact status and a trajectory optimization based on the direct collocation method, which results in a mixed-integer nonlinear programming (MINLP) problem. The planned motion is used to control the joint angles of the articulated legs. The proposed method is verified by the MuJoCo simulation and shows that over 95% and 83% success rate when the height of vertical obstacles and the length of gaps are equal to or less than 1.68 times of the wheel radius and 1.44 times of the wheel diameter, respectively.

Keywords: Wheel-legged Robot, Motion Planning, Mixed-integer Nonlinear Programming, Obstacle Crossing

1. 서 론

휠-다리 로봇(wheel-legged robot)은 포장로와 평지에서 휠을 이용하여 높은 효율로 안정적이며 고속으로 주행할 수 있고 험지에서는 다리를 이용하여 장애물을 극복할 수 있다. 이러한 복합능력의 장점으로 인하여 최근 다양한 연구가 진행¹⁻³⁾되었으며 특히 6×6 휠-다리 로봇은 하중의 분배와 수직장애물(vertical obstacle) 및 참호(trench, gap)를 통과할 수 있는 능력으로 인하여 군용로봇 등 고하중의 환경에서 운용하기 위한 연구가 수행되었다^{4,6)}.

휠-다리 로봇이 휠 반경 이상의 높은 수직장애물과 넓은 참호를 통과하기 위해서는 모션계획(motion planning)이 필요하

다. 휠-다리 로봇의 모션계획은 다접촉(multi-contact)과 휠 구름(wheel rolling)을 동시에 고려해야 하므로 복잡한 시나리오가 요구된다. 일반적으로 족형 로봇(legged robot)의 경우 발바닥과 노면 사이에 미끄러짐이 발생하지 않아 발바닥 착지(touch-down) 지점과 분리(lift-off) 지점이 동일하지만 휠-다리 로봇은 착지 상태에서 휠 구름이 발생하므로 두 지점이 서로 다를 수 있다. 이처럼 이산적인 보행특성과 연속적인 휠 구름 특성이 복합적으로 작용하여 접촉 시퀀스(contact sequence)를 찾아내기 어렵다.

족형 로봇 분야에서는 발바닥 접촉 지점 탐색과 동체 및 다리의 운동 명령을 생성하기 위한 많은 연구가 있었다. CPG(central pattern generation)⁷⁾로 다리 관절의 주기적인 운동을 생성하는 방법은 평지에서 잘 동작하였다. 비주기 자유 보행(aperiodic free gait)⁸⁾은 비정형 노면형상에서 적응하는 결과를 보여주었다. 모션계획법 중 경로최적화(trjectory optimization) 기법^{9,10)}이 활발히 연구되었으며 ZMP(zero moment point)¹¹⁾와 중심역학(centroidal dynamics)¹²⁾을 통하여 보행안정성과

Received : Oct. 11. 2022; Revised : Oct. 26. 2022; Accepted : Oct. 27. 2022

1. Ph.D Candidate, Department of Mechatronics Engineering, Chungnam National University, Daejeon, Korea (reingel@o.cnu.ac.kr)

† Professor, Corresponding author: Department of Mechatronics Engineering, Chungnam National University, Daejeon, Korea (mcwon@cnu.ac.kr)

동역학 계산속도를 향상시키는 연구가 수행되었다. 최근에는 샘플링 기반 방법^[2]과 강화학습(reinforcement learning) 기반의 방법^[13]이 주목받고 있다.

상기 방법들은 모션계획을 이산적인 발바닥 접촉 지점 탐색과 연속적인 동체, 다리 운동 명령 생성의 두 단계로 구분하여 수행한다. 이러한 디커플링(decoupling)을 통하여 각각의 문제를 단순화한다. 하지만 동체와 다리 운동을 고려하지 않고 접촉 지점을 탐색하면 동체와 다리 경로의 자유도를 제한하는 결과를 초래할 수 있다.

두 단계를 통합하기 위한 노력이 있었다. CIO(contact-invariant optimization)^[14]는 발바닥이 접촉하지 않는 구간을 포함하여 모든 구간에서 연속적인 접촉하중이 작용한다고 가정함으로써 운동방정식을 미분가능하게 만들어 최적화를 수행한다. 하지만 계산에 수분의 시간이 소요되고 접촉 지점 간의 시간을 고정함으로써 제한적인 운동을 구현한다. 혼합정수 계획법(mixed-integer programming, MIP)^[15]을 이용하여 이산적인 보행과 연속적인 운동경로를 동시에 최적화한 사례도 있다. 하지만 미리 설정한 프로파일을 사용하여 다리의 스윙 모션을 결정하기 때문에 높은 장애물극복 등의 복잡한 스윙 모션이 요구되는 경우에는 사용할 수 없다.

본 연구에서는 휠-다리의 접촉상태(contact status)를 정수로 표현하고 동체와 다리 운동은 실수로 표현하여 혼합정수 비선형 최적화 문제(mixed-integer nonlinear programming, MINLP)를 도출한 후 이산화 과정과 직접배열법(direct collocation method)^[10] 기반의 경로최적화(trajectory optimization)를 사용하여 6×6 휠-다리 로봇이 높은 수직장애물과 넓은 참호를 통과하기 위한 모션을 계획하고 제어하는 방법을 제안한다. 본 연구에서 제안하는 방법은 휠-다리 로봇에 대하여 다접촉과 휠 구름을 동시에 고려하는 복합적인 접촉-구름 지점 탐색과 연속적인 동체, 다리 운동 명령 생성의 두 단계를 통합하여 휠-다리의 자세와 휠-노면의 접촉 및 분리 지점, 동체 및 다리의 경로를 동시에 최적화하는 장점이 있다. 또한 기존 방법^[7,8,14,15]으로는 구현할 수 없는 큰 장애물에 대한 휠-다리 로봇의 최적 스윙 모션을 수초 이내의 시간에 계획하고 실시간 제어하는 방법을 제공한다.

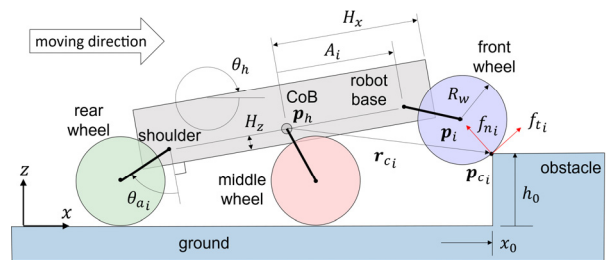
2장에서는 휠-다리 로봇 및 휠-노면 접촉 모델을 설명하고,

3장에서는 경로최적화를 이용한 모션계획법을, 4장에서는 로봇 모션 제어 방법을 기술한다. 마지막으로 6장에서는 시뮬레이션 테스트 방법 및 결과를 제시한다.

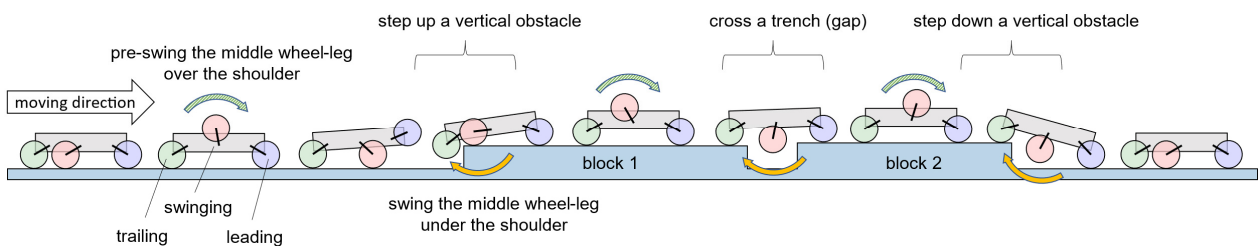
2. 휠-다리 로봇 및 휠-노면 접촉 모델

2.1 6×6 휠-다리 로봇

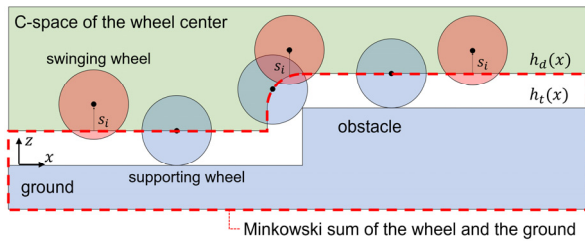
본 연구에서는 6×6 스키드 조향 휠-다리 로봇을 고려한다. 로봇 동체(robot base)의 좌우 측면에 각각 3개의 다리가 회전관절로 연결되고 다리의 끝에는 휠이 장착되어 있다. 12개의 관절은 개별적으로 구동하며 로봇의 총 자유도는 18이다. 하지만 본 연구에서 고려하는 수직장애물과 참호같이 장애물의 형상이 로봇의 진행방향을 기준으로 좌우 대칭인 경우 로봇을 2차원으로 가정하고 해석할 수 있다. 이러한 경우 총 자유도는 9이다. [Fig. 1]은 2차원 로봇 모델의 각 부분 명칭과 제원을 나타낸다. 본 논문에서 아래 첨자 i 는 휠-다리의 위치를 의미하며 1, 2, 3이 각각 전방, 중간, 후방을 나타낸다. 다리의 회전부를 어깨(shoulder)라고 하고 중간 어깨를 동체중심(center of base, CoB)이라고 부른다. A_i, H_x 는 각각 CoB에서 어깨, 동체 전면부까지의 수평거리이며 H_z 는 CoB에서 동체 하단부까지의 수직거리이다. θ_h 는 동체의 피치각, θ_{a_i} 는 동체의 수직선 기준으로 시계방향으로 측정된 다리의 회전각을 뜻한다. R_w 는 휠 반경이다. \mathbf{p}_h 는 CoB를, \mathbf{p}_i 는 휠 중심을 가리키는 벡터이며 \mathbf{p}_{c_i} 는 CoB로부터 휠의 접촉점까지의 벡터를 나타낸다. $f_{n_i} \in [0, 1]$, $f_{t_i} \in [-\mu f_{n_i}, \mu f_{n_i}]$ 는 각각 휠 중심방향과 접선방



[Fig. 1] Shape and dimension of 2D wheel-legged robot



[Fig. 2] Sequence of obstacle crossing



[Fig. 3] Configuration space of the center of the wheels

향으로 작용하는 정규화된(normalized) 휠 하중을 의미한다. 여기서 정규화된 로봇중량으로 나눈 것을 의미한다. μ 는 휠-노면 마찰계수이다. 기준좌표계는 로봇이 주행하는 방향을 x 축, 중력의 반대방향을 z 축으로 설정하였다.

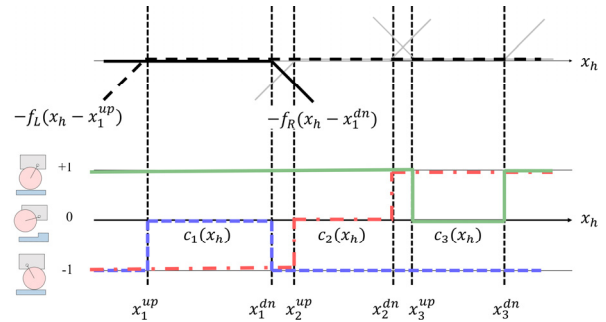
각 휠-다리는 노면과 접촉하여 동체를 지지하거나 노면과 분리되어 스윙 모션을 수행한다. 노면에 접촉하여 동체를 지지하는 휠-다리(supporting wheel-leg)는 수직선 기준으로 휠이 어깨보다 후행(trailing) 또는 선행(leading)하는 상태 중 하나에 놓이게 된다. [Fig. 1]의 중간과 후방 휠-다리는 각각 선행과 후행하는 상태이다. 따라서 휠-다리는 후행, 선행, 스윙 위치 중 하나의 접촉상태(contact status)를 갖는다. [Fig. 2]에 나타난 바와 같이 중간 휠-다리는 평지를 주행할 때 후행 위치에 있다가 큰 장애물을 통과하기 전에 미리 선행 위치로 전환하여 전방 휠-다리가 스윙할 수 있도록 동체를 지지해야 한다. 이를 사전스윙(pre-swing)이라고 하며 중간 휠-다리가 어깨 위로(over the shoulder) 회전한다. 장애물을 넘을 때는 중간 휠-다리가 어깨 아래로(under the shoulder) 회전하여 다시 후행 위치로 돌아온다. 이와 같이 중간 휠-다리는 장애물을 넘을 때마다 360도 회전해야 하기 때문에 무한히 회전할 수 있는 구조이어야 한다.

본 논문에서 큰 장애물이라 함은 휠-다리의 모션 계획 및 제어가 필요한 휠 반경보다 높은 수직장애물과 휠 직경보다 넓은 참호를 의미한다. 휠 반경보다 낮거나 휠 직경보다 좁은 참호는 휠 구동력만으로 통과할 수 있으므로 모션 계획 및 제어가 필요하지 않다.

본 연구에서는 [Fig. 2]와 같이 수직장애물과 참호를 두 개의 블록으로 모델링한다. 두 개의 블록을 연속으로 배치하면 수직장애물을 상승하고 참호를 통과하며 수직장애물을 하강하는 시나리오를 구성할 수 있다.

2.2 휠-노면 접촉 모델

휠은 노면을 관통할 수 없으므로(unilateral constraint) 휠 중심과 노면 사이의 거리가 휠 반경보다 크거나 같아야 한다. 이는 [Fig. 3]와 같이 휠과 노면의 민코프스키 합(Minkowski sum)



[Fig. 4] Contact status functions

또는 팽창(dilation)^[16]으로 기술할 수 있다. 휠 중심이 도달할 수 있는 공간은 모든 2차원 공간에서 휠과 노면의 민코프스키 합을 뺀 것과 같다. [Fig. 3]에서 $h_t(x)$ 와 $h_d(x)$ 는 각각 원래의 노면과 팽창된 노면의 높이를 나타내는 매개변수 x 의 함수이다.

본 연구에서는 휠-다리의 접촉상태를 표현하기 위하여 정수 벡터를 출력하는 식 (1)의 접촉상태함수(contact status function, CSF)를 도입한다.

$$\mathbf{c}(x_h) \in \{-1, 0, +1\}^{N_a} \quad (1)$$

여기서 x_h 는 CoB의 x 축 좌표값이며 N_a 는 휠-다리의 개수를 의미한다. 0은 휠-다리가 스윙하는 것을 의미하며 +1은 후행 자세, -1은 선행 자세를 나타낸다. CSF는 휠-다리의 자세와 노면과의 접촉 여부를 결정하고 노면과의 접지 위치와 분리 위치를 나타내는 중요한 역할을 수행한다.

x_i^{up} 과 x_i^{dn} 을 휠-다리 i 가 노면과 분리되고 접촉하는 CoB 위치를 의미한다면 i 번째 휠-다리의 CSF는 식 (2)와 같이 표현할 수 있다.

$$c_i(x_h) = \text{sign}(-f_L(x_h - x_i^{up}) - f_R(x_h - x_i^{dn})) \quad (2)$$

여기서 $f_L = \frac{1}{2}(|x| - x)$, $f_R = \frac{1}{2}(|x| + x)$ 이다. f_L 은 x 가 음수이면 x 이고 그렇지 않으면 0이다. f_L 과 f_R 은 [Fig. 4]에 나타난 바와 같이 각각 CoB가 x_i^{up} 에 도달하지 않았을 때와 x_i^{dn} 을 지났을 때의 휠-다리 접촉상태를 제어한다. x_i^{up} 과 x_i^{dn} 사이에서는 c_i 가 0이 되어 휠-다리가 스윙하도록 한다. x_i^{up} 과 x_i^{dn} 는 전방, 중간, 후방 휠이 순차적으로 스윙하도록 장애물이 시작되기 전과 후의 위치로 설정한다.

직접배열법을 적용하기 위하여 c_i 를 이산화하면 $c_i^{(k)}$ 를 얻을 수 있다. 여기서 $k \in \{1, 2, \dots, K\}$ 이며 K 는 샘플링 수를 의미한다.

3. 경로최적화를 이용한 모션계획법

휠-다리 로봇의 모션을 생성하기 위하여 경로최적화의 직접배열법을 사용한다. 설계변수(design variables)는 다음과 같이 설정한다.

$$z_h^{(k)}, \theta_h^{(k)}, \theta_{a_i}^{(k)}, f_{n_i}^{(k)}, f_{t_i}^{(k)}, s_i^{(k)}$$

여기서 z_h 는 CoB의 z 축 좌표값이며 s_i 는 [Fig. 3]에 나타낸 바와 같이 각 휠이 노면으로부터 떨어진 거리를 나타내는 슬랙 변수(slack variable)이다. 우측 윗첨자 (k) 는 k 번째 이산화된 값을 의미한다. 본 연구에서는 식 (3a)의 비용함수와 식 (3b-f)의 구속조건으로 최적화 문제를 구성하였다.

$$\text{minimize } 1000J_1 + 10J_2 + J_3 + J_4 + J_5 \quad (3a)$$

$$\text{subject to } p_{z_i} = h_d(p_{x_i}) + b_i s_i \quad (3b)$$

$$(z_h - |h_o|) \cos \theta_h + (x_h - x_o) \sin \theta_h - H_z \geq \epsilon_h \quad (3c)$$

$$|\mathbf{p}_i - \mathbf{p}_j|^2 \geq (2R_w + \epsilon_w)^2, i \neq j \quad (3d)$$

$$\sum_{i=1}^{N_o} f_{x_i} = 0, \sum_{i=1}^{N_o} f_{z_i} = 1, \sum_{i=1}^{N_o} \mathbf{r}_{c_i} \times \mathbf{f}_i = 0 \quad (3e)$$

$$\min(p_{cop} - p_{s, rear}, p_{s, front} - p_{cop}) \geq \epsilon_{zmp} \quad (3f)$$

식 (3a)는 장애물통과 후 마지막 상태의 오차를 최소화하기 위한 항과 장애물통과 과정에서 동체 피치각의 변화량, 다리 회전각의 변화량, 다리 회전각의 변화량의 변화량, 휠 수직하중의 변화량을 최소화하기 위한 제곱항들의 가중합(weighted sum)으로 구성되며 상세한 계산방법을 식 (4a-e)에 나타내었다. 여기서 변화량이란 CoB의 x 축 좌표에 대한 변화를 의미한다. 식 (4a)에서 우측 윗첨자 igt 는 목표값을 의미한다.

$$J_1 = (z_h^{(K)} - z_h^{igt})^2 + (H_x(\theta_h^{(K)} - \theta_h^{igt}))^2 + \sum_{i=1}^{N_o} (L_a(\theta_{a_i}^{(K)} - \theta_{a_i}^{igt}))^2 \quad (4a)$$

$$J_2 = \sum_{k=1}^{K-1} (\theta_h^{(k+1)} - \theta_h^{(k)})^2 \quad (4b)$$

$$J_3 = \sum_{i=1}^{N_o} \sum_{k=1}^{K-1} (\theta_{a_i}^{(k+1)} - \theta_{a_i}^{(k)})^2 \quad (4c)$$

$$J_4 = \sum_{i=1}^{N_o} \sum_{k=1}^{K-2} (\theta_{a_i}^{(k+2)} - 2\theta_{a_i}^{(k+1)} + \theta_{a_i}^{(k)})^2 \quad (4d)$$

$$J_5 = \sum_{i=1}^{N_o} \sum_{k=1}^{K-1} (f_{n_i}^{(k+1)} - f_{n_i}^{(k)})^2 \quad (4e)$$

식 (3b)는 휠-노면의 단방향 접촉조건(unilateral contact)이다. p_{z_i} 는 휠 중심의 높이를 의미하며 식 (5)로부터 구한다.

$b_i = (1 - c_i)(1 + c_i)$ 이며 휠-다리가 스윙하면 1, 그렇지 않으면 0이다. 따라서 $b_i s_i$ 는 i 번째 휠이 스윙하면 양수이고 접지하면 0이다.

$$\mathbf{p}_i \equiv \begin{bmatrix} p_{x_i} \\ p_{z_i} \end{bmatrix} = \mathbf{p}_h + \mathbf{R}_y(\theta_h) \begin{bmatrix} A_i \\ 0 \end{bmatrix} + \mathbf{R}_y(\theta_{a_i}) \begin{bmatrix} 0 \\ -L_a \end{bmatrix} \quad (5)$$

여기서 \mathbf{R}_y 는 y 축 방향 회전행렬이고 L_a 는 다리길이이다. 식 (3c-d)는 각각 동체-장애물 간의 거리와 휠-휠 사이의 거리가 일정한 값 이상이어야 하는 조건이다. [Fig. 1]에 나타낸 바와 같이 x_o 는 장애물이 시작하는 위치, h_o 는 장애물 높이이며 ϵ_h 와 ϵ_w 는 최소 거리를 의미한다. 마지막으로 식 (3e-f)는 로봇의 정적평형 조건이다. p_{cop} 는 압력중심(center of pressure, CoP)^[11]의 위치이며 식 (6a-c)로부터 구한다. $p_{s, front}$, $p_{s, rear}$ 는 지지 다각형(support polygon)^[11]의 전후방 변(edge)의 위치이고 ϵ_{zmp} 는 정적평형을 이루기 위한 최소 마진이다.

$$\mathbf{r}_{c_i} = \mathbf{p}_{c_i} - \mathbf{p}_h = \mathbf{p}_i - \mathbf{p}_h + \mathbf{R}_y(-\phi_{d_i}) \begin{bmatrix} 0 \\ -R_w \end{bmatrix} \quad (6a)$$

$$\mathbf{f}_i \equiv \begin{bmatrix} f_{x_i} \\ f_{y_i} \end{bmatrix} = (1 - b_i) \mathbf{R}_y(-\phi_{d_i}) \begin{bmatrix} f_{t_i} \\ f_{n_i} \end{bmatrix} \quad (6b)$$

$$p_{cop} = \sum_{i=1}^{N_o} f_{z_i} p_{c_{x_i}} \quad (6c)$$

여기서 \mathbf{r}_{c_i} 는 휠-노면 접촉점의 CoB 대비 상대위치이고 $\phi_{d_i} = \tan^{-1} \frac{dh_d(p_{x_i})}{dx}$ 는 팽창된 노면의 기울기이다.

경로최적화 문제를 풀면 CoB의 x 좌표에 대한 최적 다리 회전각과 회전각 차이값(angle difference)을 얻을 수 있으며 이를 이용하여 로봇 모션을 제어한다.

4. 로봇 모션 제어

경로최적화를 통하여 획득한 최적 다리 회전각(optimal joint angle of leg)은 선형보간을 거쳐 연속함수로 변환한 후 CoB의 x 축 좌표값에 대한 다리 회전각 명령을 도출한다. 휠 속도 명령은 조인트 구동기의 성능제한을 넘지 않는 최대 속도로 설정한다. 즉, 경로최적화를 통하여 얻은 최적 휠 회전각 차이(optimal angular difference)를 선형보간한 후 CoB의 x 축 좌표값에 따라 식 (7b)를 이용하여 휠 속도 명령을 도출한다.

$$\theta_{a_i}^{ref}(x_h) = \theta_{a_i}^{opt}(x_h) \quad (7a)$$

$$\omega_{w_i}^{ref}(x_h) = \frac{1}{R_w} \min \left(\frac{\Delta x \dot{\theta}_{amax}}{\max_i \Delta \theta_{a_i}^{opt}(x_h)}, v_{max} \right) \quad (7b)$$

여기서 $\theta_{a_i}^{opt}(x_h)$ 는 x_h 에서 선형보간한 최적 다리 회전각이며 $\dot{\theta}_{amax}$ 는 최대 다리 회전속도이고 $\Delta\theta_{a_i}^{opt}(x_h)$ 는 x_h 에서 선형보간한 최적의 다리 회전각 차이, v_{max} 는 로봇의 최대 속도를 의미한다.

도출된 다리 회전각 명령과 휠 속도 명령을 이용하여 다리 회전각과 휠 속도를 제어한다. 이 때 세 개의 휠 속도는 스윙하는 휠을 포함하여 모두 동일하게 제어한다.

5. 시뮬레이션 테스트

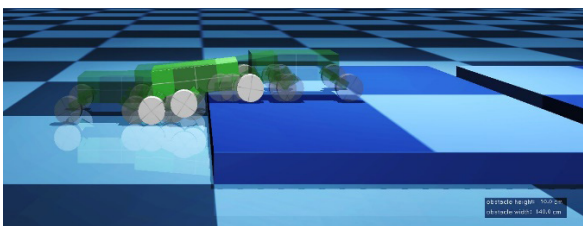
5.1 시뮬레이션 환경

제한한 모션 계획법과 제어방법을 검증하기 위하여 [Fig. 5]와 같이 MuJoCo^[17]를 이용한 시뮬레이션 테스트를 수행하였다. MuJoCo는 최적제어와 강화학습 분야에서 널리 사용되는 동역학 시뮬레이션 도구이며 C 언어 API를 제공한다. 3D 형태의 6×6 휠-다리 로봇을 MJCF 파일로 모델링하였다. 로봇의 주요 제원을 [Table 1]에 나타내었다. 경로최적화로 도출한 제어 명령은 좌우 휠-다리에 동일한 값으로 입력하였다.

제안된 모션 계획 및 제어 방법은 자동미분 및 최적화를 위한 심볼릭 프레임워크인 CasADi^[18]로 구현하였다. CasADi는 주로 최적제어와 MPC 분야에서 사용되며 C와 파이썬, MATLAB에서 사용이 가능하다. 본 연구에서는 파이썬으로 구현하였다. 오픈소스 최적화 도구인 IPOPT^[19]를 경로최적화에 사용하였다.

C++ 언어를 이용하여 MuJoCo의 C API를 추상화하여 로봇 모델 접근, 센서신호 획득, 제어입력 적용, 시뮬레이션 가시화 등이 가능한 클래스를 설계하였다. Pybind11^[20]과 Eigen3^[21]를 이용하여 C++ 코드를 파이썬과 바인딩하였다. CasADi와 MuJoCo는 파이썬 인터페이스로 결합된다.

다리의 회전은 토크제어 방식으로 구동되며 P 계인은 10 Nm/deg로 설정하고 토크 제한은 ±2 Nm로 설정하였다. 휠은 P 계인 0.2 Nm/(deg/s)인 토크 제어기로 구동하였다. 시뮬레이션은 Ubuntu 20.04, AMD Raizen 3700X CPU, 16 GB RAM으로 구성된 컴퓨터에서 수행하였다.



[Fig. 5] MuJoCo simulation of 6×6 wheel-legged robot

5.2 테스트 시나리오

테스트 시나리오는 다음과 같다. 로봇이 x 축 좌표 0에 위치한 곳에서 출발하여 두 개의 블록으로 구성된 장애물을 통과한다. 첫 번째 블록에서는 수직장애물 상승, 첫 번째 블록에서 두 번째 블록으로 넘어가면서 참호 통과, 두 번째 블록을 내려 오면서 수직장애물 하강 테스트를 수행한다. 두 블록의 시작 위치, 높이, 길이를 랜덤하게 생성하여 다양한 높이와 넓이의 수직장애물과 참호를 통과하도록 설계하였다.

로봇이 장애물극복 시작점에 도달하면 장애물극복 목표위치까지의 장애물 높이 200개를 샘플링하여 모션계획 모듈에 입력한다. 장애물극복 시작점은 수직장애물과 참호가 시작되는 지점에서 0.4 m 전방이다. 장애물극복 목표위치는 수직장애물의 경우 장애물 시작점으로부터 0.8 m 경과한 지점이고 참호인 경우 ‘참호 길이 + 0.8 m’ 경과한 지점이다. 로봇이 장애물을 최초로 인식하는 지점은 장애물 시작점 1 m 전방위치이다. 이 때부터 중간 휠-다리를 시계방향으로 회전하여 후행 위치에서 선행위치로 전환한다. 전환을 완료하면 모션계획 모듈을 이용하여 최적의 모션을 찾는다. 본 경로최적화 문제는 566개의 변수, 243개의 등식 제한조건, 897개의 부등식 제한 조건이 있다.

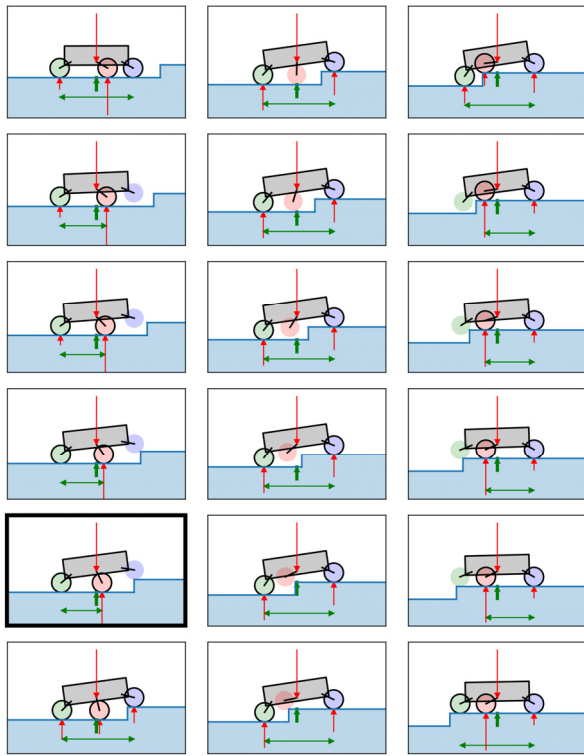
장애물 크기를 랜덤하게 변화하면서 1000번의 시뮬레이션을 수행하였다. 시나리오에 따라 순차적으로 시뮬레이션을 수행하다가 로봇이 장애물을 극복하지 못하고 5초간 정지한 상태를 지속하면 해당 에피소드를 종료하고 다음 에피소드를 진행하였다. 따라서 수직장애물 상승, 참호 통과, 수직장애물 하강 순으로 테스트 회수가 감소한다. 로봇이 5초가 정지하는 현상없이 장애물을 통과하면 성공으로 판단한다. 장애물통과 시도 횟수 대비 성공 횟수를 이용하여 장애물통과 성공율(success rate)을 계산한다.

5.3 시뮬레이션 결과

[Fig. 6]는 수직장애물 상승시 경로최적화 결과를 예시로 나타낸 그림이다. 좌측상단 박스가 장애물극복 시작점이며 우측하단 박스가 장애물극복 목표위치이다. 붉은 원으로 나타난 휠은 접지한 상태이며 그렇지 않은 휠은 스윙하고 있는

[Table 1] Specification of the wheel-legged robot

Spec.	Value	Spec.	Value
H_x	20 cm	H_z	2 cm
A_1	16 cm	L_a	8 cm
R_w	6 cm		



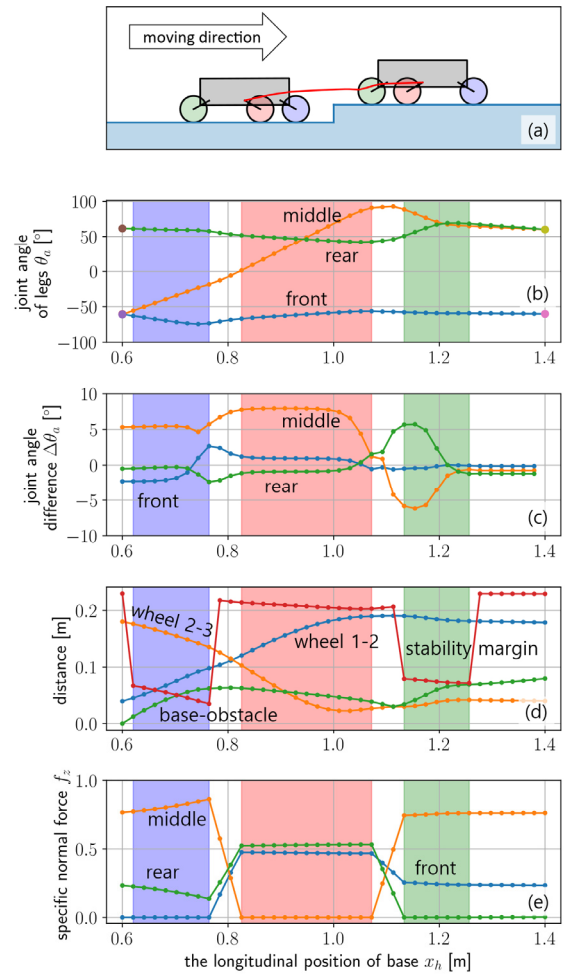
[Fig. 6] Key frames for stepping up a large obstacle

상태이다. 수직 아래방향 화살표는 로봇의 총중량을 의미하며 수직 위방향 화살표는 휠 하중을 의미한다. 굵은 수직 위방향 화살표는 로봇의 ZMP를 나타낸다. 수평 양방향 화살표는 지지영역(support region)을 의미한다. 안전마진(stability margin)은 ZMP로부터 지지영역 양단까지의 거리 중 짧은 값이다. 굵은 박스로 표시한 프레임은 안전마진이 가장 짧은 순간을 나타낸다.

몇 가지 중요한 결과를 [Fig. 7]에 나타내었다. [Fig. 7(a)]는 모션계획의 처음과 마지막 로봇 위치와 CoB의 경로를 나타낸다. 중간 휠-다리의 위치가 전환되었음을 확인할 수 있다. 나머지 그래프는 각각 설계변수의 최적화 결과를 나타낸다. 세 개의 수직 띠는 각각 전방, 중간, 후방 다리가 스윙하는 구간을 나타낸다.

각 장애물에 대한 경로최적화 계산시간을 [Table 2]에 나타내었다. 수직장애물 상승, 참호 통과, 수직장애물 하강의 경로 최적화를 수행하는 데에 각각 평균적으로 6.79초, 3.33초, 3.73초가 소요되었다.

수직장애물 상승은 1000회 시도하였고 791회 성공하였다. 수직장애물 상승에 성공한 로봇이 참호 통과를 시도할 수 있으므로 참호 통과 시도 횟수도 791회이다. 참호 통과 성공 횟수와 수직장애물 하강 시도 횟수는 405회이다. 수직장애물 하강은 모두 성공하였다. 수직장애물 높이, 참호 길이에

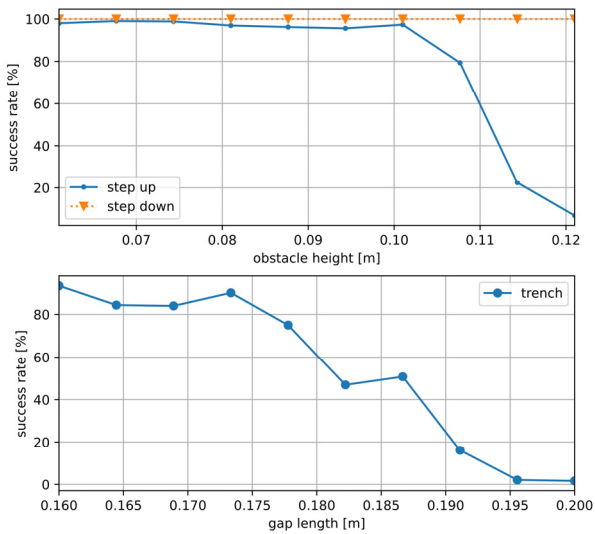


[Fig. 7] Results of trajectory optimization for stepping up a large obstacle

[Table 2] Calculation time for trajectory optimization

[sec]	step up	trench	step down
max	23.91	12.30	12.16
mean	6.79	3.33	3.73
min	1.61	0.91	1.30

대한 장애물통과 성공율을 [Fig. 8]에 나타내었다. 수직장애물 상승은 높이 0.101 m 이하에서 95% 이상의 성공율을, 참호 통과는 길이 0.173 m 이하에서 83% 이상의 성공율을 보여 주었다. 참호 통과 성공율이 다소 낮은 이유는 다양한 장애물 모사를 위하여 두 개 블록의 높이를 랜덤하게 생성하므로 참호의 길이만 존재하는 것이 아니라 높이 차이도 발생하기 때문인 것으로 판단된다. [Fig. 8]의 아래 그래프에는 참호 길이의 함수로만 표현하였다. 수직장애물 하강은 100% 성공하였다. 0.101 m와 0.173 m는 각각 휠 반경의 1.68배와 휠 직경의 1.44배이다.



[Fig. 8] Success rate of obstacle crossing w.r.t. obstacle heights and gap lengths

6. 결 론

본 연구에서는 6×6 휠-다리 로봇의 장애물극복을 위하여 2D 로봇 모델과 휠-노면 접촉 모델을 유도하였고 휠-노면 접촉상태를 정수로 나타내는 접촉상태함수를 도입하여 혼합정수 비선형 최적화 문제로 표현한 경로최적화를 직접배열법으로 풀어 최적의 휠-다리 모션을 계획하고 제어하는 방법을 제시하였다.

MuJoCo 시뮬레이션을 통하여 제안한 방법이 높은 수직장애물과 넓은 참호를 통과하기 위한 휠-다리와 로봇 동체의 경로를 최적화하는 데에 빠르고 성공율이 높은 방법임을 확인하였다.

후속 연구로는 본 연구에서 획득한 데이터를 이용하여 지도학습과 강화학습을 수행하여 최적경로를 빠르게 계산하면서도 좌우가 동일하지 않은 형상 등으로 인한 외란에 강한 심층신경망 제어를 설계하는 방안을 고려하고 있다.

References

[1] A. Seidla, “Actively articulated suspension for a four wheeled vehicle,” *NordDesign 2008*, Tallinn, Estonia, 2008, [Online], https://www.designsociety.org/download-publication/27388/actively_articulated_suspension_for_a_four-wheeled_vehicle.

[2] T. Klamt and S. Behnke, “Anytime hybrid driving-stepping locomotion planning,” *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, DOI: 10.1109/IROS.2017.8206310.

[3] M. Bjelonic, R. Grandia, O. Harley, C. Galliard, S. Zimmermann,

and M. Hutter, “Whole-body MPC and online gait sequence generation for wheeled-legged robots,” *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, 2021, DOI: 10.1109/IROS51168.2021.9636371.

[4] MAJ D. B. Byers, “Multifunctional utility/logistics and equipment (mule) vehicle will improve soldier mobility, survivability and lethality,” *Army Acquisition, Logistics & Technology*, Fort Belvoir, VA, USA, Rep. 0704-0188, Jun., 2008, [Online], <https://apps.dtic.mil/sti/pdfs/ADA490301.pdf>.

[5] J. Kang, W. Kim, J. Lee, and K. Yi, “Skid steering-based control of a robotic vehicle with six in-wheel drives,” *Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 224, no. 11, Jul., 2010, DOI: 10.1243/09544070JAUTO1405.

[6] X. Zhang, S. Yuan, X. Yin, X. Li, X. Qu, and Q. Liu, “Estimation of skid-steered wheeled vehicle states using STUKF with adaptive noise adjustment,” *Applied Sciences*, vol. 11, no. 21, 2021, DOI: 10.3390/app112110391.

[7] S. Inagaki, H. Yuasa, and T. Arai, “CPG model for autonomous decentralized multi-legged robot system—generation and transition of oscillation patterns and dynamics of oscillators,” *Robotics and Autonomous Systems*, vol. 44, no. 3-4, pp. 171-179, 2003, DOI: 10.1016/S0921-8890(03)00067-8.

[8] R. B. McGhee and G. I. Iswandhi, “Adaptive locomotion of a multilegged robot over rough terrain,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 4, pp. 176-182, Apr., 1979, DOI: 10.1109/TSMC.1979.4310180.

[9] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, 2nd ed., Advances in Design and Control, DOI: 10.1137/1.9780898718577.

[10] M. Kelly, “An introduction to trajectory optimization: How to do your own direct collocation,” *SIAM Review*, vol. 59 no. 4, pp. 849-904, Nov., 2017, DOI: 10.1137/16M1062569.

[11] M. Vukobratović and B. Borovac, “Zero-moment point—thirty-five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157-173, 2004, DOI: 10.1142/S0219843604000083.

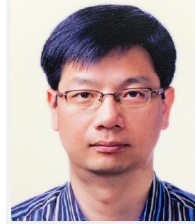
[12] D. E. Orin and A. Goswami, “Centroidal momentum matrix of a humanoid robot: Structure and properties,” *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 653-659, 2008, DOI: 10.1109/IROS.2008.4650772.

[13] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, “DeepGait: Planning and control of quadrupedal gaits using deep reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3699-3706, Apr., 2020, DOI: 10.1109/LRA.2020.2979660.

[14] I. Mordatch, E. Todorov, and Z. Popovic, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1-8, Jul., 2012, DOI: 10.1145/2185520.2185539.

[15] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, “Simultaneous contact, gait, and motion planning for robust multilegged locomotion via

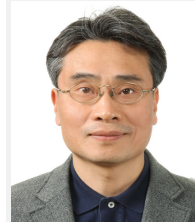
- mixed-integer convex optimization,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, Jul., 2018, DOI: 10.1109/LRA.2017.2779821.
- [16] J. C. Latombe, “Robot motion planning,” *Springer Science & Business Media*, vol. 124, 1991, DOI: 10.1007/978-1-4615-4022-9.
- [17] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” *IEEE/RSJ international conference on intelligent robots and systems*, Vilamoura-Algarve, Portugal, pp. 5026-5033, 2012, DOI: 10.1109/IROS.2012.6386109.
- [18] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi - A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1-36, 2019, DOI:10.1007/s12532-018-0139-4.
- [19] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for largescale nonlinear programming,” *Mathematical Programming*, vol. 106, no. 1, pp. 25-57, 2006, DOI: 10.1007/s10107-004-0559-y.
- [20] *Pybind11—seamless operability between C++11 and Python*, [Online], <https://github.com/pybind/pybind11>, Accessed: Nov., 1, 2022.
- [21] *Eigen is a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms*, [Online], <https://eigen.tuxfamily.org>, Accessed: Nov., 1, 2022.



정 순 규

1996 한양대학교 기계설계공학과(학사)
 1998 GIST 기전공학과(석사)
 2020~현재 충남대학교 메카트로닉스공학과 박사과정

관심분야: Wheel-legged robot, Legged robot, Trajectory optimization, Reinforcement Learning



원 문 철

1983 서울대학교 조선공학과(학사)
 1985 서울대학교 조선공학과(석사)
 1995 U.C. Berkeley Mechanical Engineering(Ph.D.)
 1987~1990 한국기계연구원
 1995~현재 충남대학교 메카트로닉스공학과 교수

관심분야: Control of vehicles and mechatronics systems, AI, Self-driving vehicles, Robots, Computer vision