3 자유도 다리 시스템의 실시간 제어를 위한 전신 MPPI

Whole Body MPPI for Real-time Control of a 3-DoFs Leg system

서윤수[1,2], 임묘택[1], 이이수 [2] †

1. Department of Electrical and Electronic Engineering, Korea University
2. Center for Humanoid Research, Korea Institute of Science and Technology (KIST)

I. Purpose

- Control and planning based on whole body dynamics face challenges due to **Nonlinearity and High Computational Complexity**
- Existing Approach:
  - ✓ CoM Path Generator with Simplified Model(LIPM)→ Whole Body Controller
  - ✓ Difference between actual model⇧ → **Low Performance**
- This Study:
  - ✓ **MPPI(Model Predictive Path Integral) based Whole Body Control Framework**
  - ✓ **Single-torque sampling** with removed contact forces
  - ✓ Improved sampling efficiency **for real-time(200 Hz)** computation
  - ✓ **Physically valid whole-body motion generation**

➡ **Real-time MPPI whole-body controller in contact task**

II. Proposed Method

- Framework Overview:

Sampling $\xrightarrow{\varepsilon}$ State update Eq(3) $\xrightarrow{q,\dot{q},\hat{x}}$ Cost function Eq(2) $\xrightarrow{\tilde{S}}$ Optimization Eq(1) $\xrightarrow{\tau(\text{Eq}(4))}$ Robot

Robot $\xrightarrow{q,\dot{q}}$ Sampling

- Divided into dynamics model and MPPI optimization

II –1. MPPI

- **Model Predictive Path Integral (MPPI)** Algorithm
- Sampling-Based MPC, can be directly applied to nonlinear dynamics

Input: $u_{init}, \hat{x}_{init}$
Parameter: N, T
While *task not completed* **do**
  $u_0, \hat{x}_0 \leftarrow$ Get initial state()
  $\varepsilon \leftarrow$ Generate sample(N, T, ν, σ)
  for $n \leftarrow 0$ to $N-1$ **do**
    for $k \leftarrow 0$ to $T-1$ **do**
      $\hat{x}_{n,k+1} \leftarrow$ Update system dynamics $(\hat{x}_{n,k}, \varepsilon_{n,k})$
      $\tilde{S}_{n,k+1} \leftarrow$ Cost function $(\hat{x}_{n,k}, u_{n,k})$
    end for
  end for
  $\beta = min(\tilde{S}_{n,k+1})$
  for $k \leftarrow 0$ to $T-1$ **do**
    $u_{n,k} \leftarrow$ Next command input$(u_{n,k} + \Delta u^*_{n,k})$
  end for
  Send control input$(u_{n,k})$
  for $k \leftarrow 0$ to $T-2$ **do**
    $u_{n,k} = u_{n,k+1}$
    Update current state()
    Check task completion()
  end for
end while

[Algorithm 1] Model Predictive Path Integral

$$\Delta u^*_{n,k} = \frac{\sum_0^N e^{-(\tilde{S}_{n,k+1}-\tilde{S}_{min})} * \varepsilon_{n,k}}{\sum_0^N e^{-(\tilde{S}_{n,k+1}-\tilde{S}_{min})}}$$

[Eq 1] Optimized control input

- $\hat{x}_k$: Position of floating base
- $u_{n,k}$: control input (torque)
- $\omega_{n,k}$: weight
- $\varepsilon_{n,k}$: sample
- N : total timestep
- k : timestep
- n : sampling number
- $\tilde{S}_{n,k+1}$: cost function

- Cost Function

$$\tilde{S}_{total} = \tilde{S}_{goal} + 1000 c_{ZMP} \qquad \text{- Eq(2)}$$

$$c_{ZMP} = \begin{cases} 1, & (ZMP > support\ region) \\ 0, & (ZMP \le support\ region) \end{cases}$$

➡ **Stable control input by goal error and ZMP penalty**

II –2. Dynamic Model

- Basic Whole Body Dynamics

$$M\ddot{q} + h + g + J_c{}^T f_c = B\tau$$

- Joint Torques(τ), Contact Force f_c exists.

→ **Express Contact Forces** using contact condition $(\ddot{x} = J_c\ddot{q} + \dot{J}_c\dot{q} = 0)$

$$f_c = \left(J_c M^{-1} J_c{}^T\right)^{-1}\left(\dot{J}_c\dot{q} - J_c M^{-1} h + J_c M^{-1} B\tau\right)$$

→ **Integrating contact conditions** (f_c) into dynamics

$$\ddot{q} = M^{-1}\{-h + J_c{}^T \Lambda_c \mu_c + \left(I - J_c{}^T \Lambda_c J_c M^{-1}\right) B(\tau + \Delta\tau)\} \qquad \text{- Eq(3)}$$

$$\Lambda_c = \left(J_c M^{-1} J_c{}^T\right)^{-1}$$

$$\mu_c = \dot{J}_c\dot{q} - J_c M^{-1} h$$

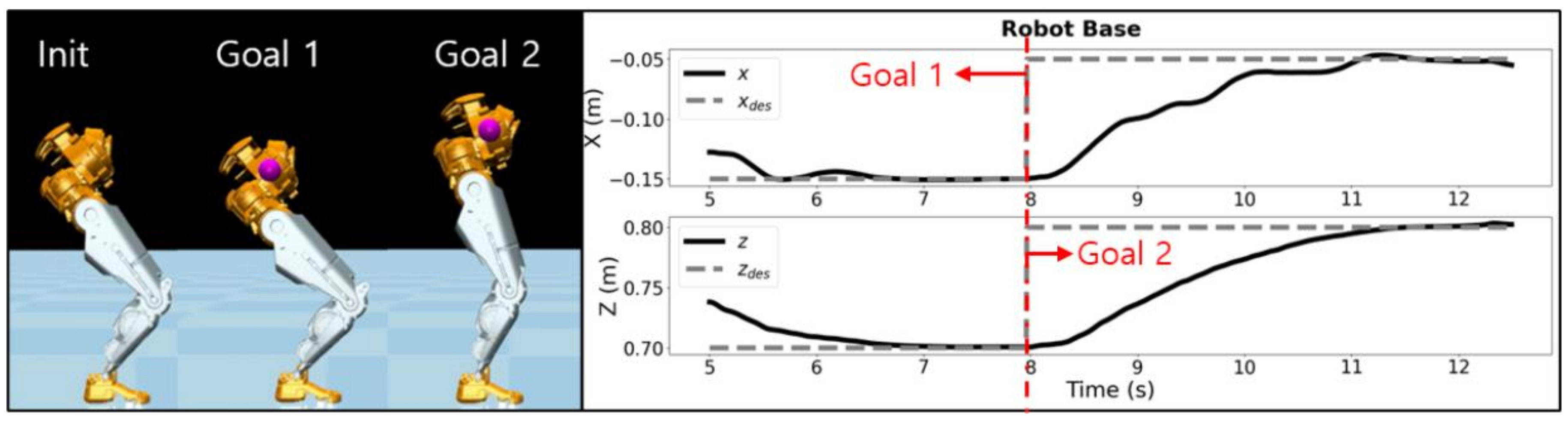
➡ **Sampling variables⇩ → Improve Computational efficiency**

III. Simulation Result

- Simulation Environment: Mujoco, C++ (Intel i7-12700F, 32GB RAM)
- Robot Model: 3DoF leg System, 2D Plane(X,Z axes)
- Controller$(u = \tau)$

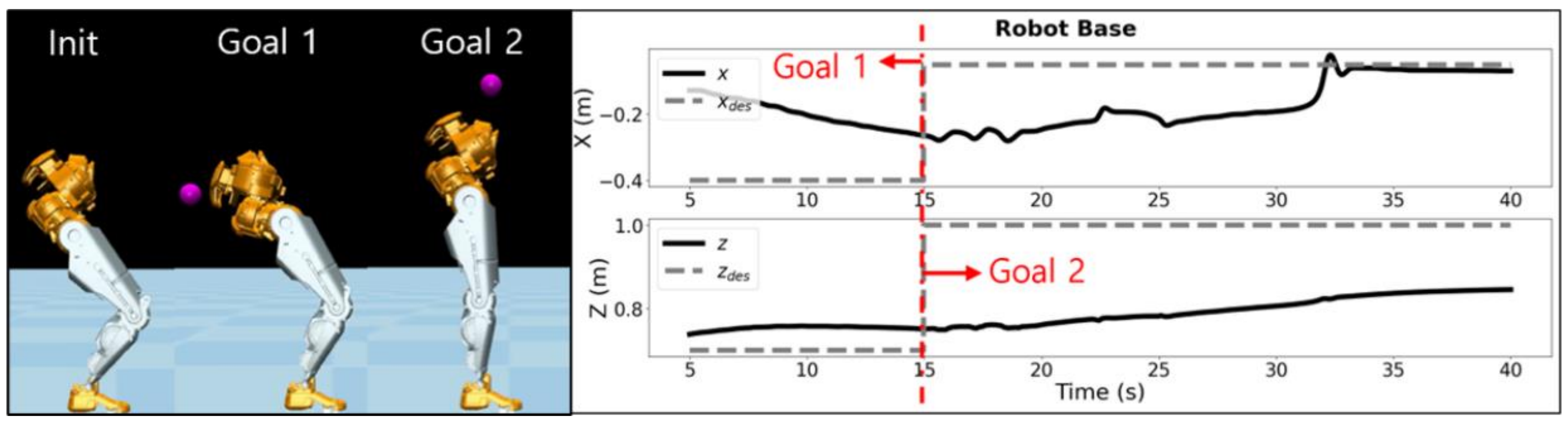
$$\tau = \tau^* + K_p(q_{des} - q) + K_d(-\dot{q}) \qquad \text{- Eq(4)}$$

[Task 1] Desired outcome when goal is achievable



→ **Base converges** to the goal while maintaining contact

[Task 2] Desired outcome when goal is not achievable



→ Base desire doesn't converge if unreachable, but **contact is maintained**

→ Tries to get **as close as possible** within the contact maintaining range

- Parameters :

N	σ	Δt	T
100	0.05	0.005	5

➡ **Real-time control possible at 4.73ms computation speed(200 Hz)**

IV. Discussion

- Improvement in **Sampling Efficiency through Single-Variable Optimization**
- **Whole-body Control Achieved via MPPI**
- **Real-time Motion Planning and Control** performance proven **(200 Hz)**
- Future work: Expansion to complex Multi-DoF robots with parallel processing
- Future work: Expanding to slipping, aerial motion, and jumping