MODELING AND SIMULATION OF QUADRUPED ROBOT LOCOMOTION

Nguyen Tien Hung*, **Nguyen Duc Mui**
TNU - University of Technology

ARTICLE INFO

Received: 09/6/2025
Revised: 14/11/2025
Published: 18/11/2025

KEYWORDS

Quadruped robot
Dynamic modeling
Rigid body
Locomotion control
Model predictive control
Simulation

ABSTRACT

This paper presents a comprehensive analysis and implementation of a model predictive control framework designed to compute ground reaction forces for joint-level control of quadruped robots. The robot's rigid-body dynamics are systematically simplified, allowing the control problem to be formulated as a convex quadratic optimization. Simulation results, obtained using a reduced-order dynamic model, confirm the reliability and consistency of the proposed control strategy. These results are further validated in a high-fidelity environment using the MuJoCo simulator, in which the Unitree-Go1 quadruped robot model is employed. The model predictive controller successfully maintains dynamic stability. The ground forces are then mapped to joint torques through inverse dynamics computations, ensuring that the generated torque commands remain physically realizable. The controller demonstrates robust tracking performance of desired center-of-mass trajectories. The outcomes suggest that the simplified model retains sufficient dynamic fidelity to capture the core behaviors essential for real-time gait generation and control of the robot. These results establish a strong foundation for future developments in advanced control strategies.

MÔ HÌNH HÓA VÀ MÔ PHỎNG CHUYỂN ĐỘNG CỦA ROBOT BỐN CHÂN

Nguyễn Tiến Hưng*, **Nguyễn Đức Mùi**
Trường Đại học Kỹ thuật Công nghiệp – ĐH Thái Nguyên

THÔNG TIN BÀI BÁO

Ngày nhận bài: 09/6/2025
Ngày hoàn thiện: 14/11/2025
Ngày đăng: 18/11/2025

TỪ KHÓA

Rô bốt bốn chân
Mô hình động học
Thân cứng
Điều khiển chuyển động
Điều khiển dự đoán mô hình
Mô phỏng

TÓM TẮT

Bài báo này trình bày một phân tích đầy đủ và triển khai một bộ điều khiển dự báo mô hình được thiết kế để tính toán lực phản ứng mặt đất cho điều khiển các khớp của rô bốt bốn chân. Động lực học của rô bốt được đơn giản hóa một cách có hệ thống, cho phép xây dựng bài toán điều khiển dưới dạng tối ưu hóa toàn phương lồi. Kết quả mô phỏng thu được bằng cách sử dụng mô hình động học giảm bậc đã xác nhận độ tin cậy và tính nhất quán của chiến lược điều khiển được đề xuất. Những kết quả này được xác thực thêm bằng cách sử dụng trình mô phỏng MuJoCo với mô hình rô bốt bốn chân Unitree-Go1. Bộ điều khiển dự báo mô hình đảm bảo khả năng duy trì tính ổn định động. Sau đó, lực mặt đất được ánh xạ thành mô-men xoắn khớp thông qua các phép tính động lực học ngược, cho phép các mô-men xoắn được tạo ra có thể thực hiện được trong thực tế. Bộ điều khiển thể hiện chất lượng bám các quỹ đạo trọng tâm mong muốn. Các kết quả cho thấy mô hình đơn giản hóa vẫn giữ được tính trung thực động học đủ để nắm được các hành vi cần thiết cho việc tạo dáng đi theo thời gian thực và điều khiển rô bốt. Những kết quả này thiết lập nền tảng vững chắc cho các phát triển trong tương lai với các chiến lược điều khiển tiên tiến.

DOI: https://doi.org/10.34238/tnu-jst.13012

* Corresponding author. *Email: h.nguyentien@tnut.edu.vn*

1. Introduction

In recent years, the vision of building general-purpose robotic agents capable of autonomously performing a broad spectrum of tasks in human environments has gained considerable traction. This is driven by advances in machine learning, control theory, and mechanical design, as well as the growing demand for adaptable robots in sectors such as manufacturing, logistics, healthcare, and domestic assistance [1]-[3]. Consequently, there is an increasing effort to develop integrated control and planning frameworks that can endow a single robot with both robust mobility and skilled manipulation. Despite this progress, achieving such integrated, whole-body capabilities in real-world settings remains a significant technical hurdle. Quadruped and humanoid robots must coordinate multiple degrees of freedom across their entire body, handle contact-rich interactions, adapt in real-time to unpredictable environments, and maintain physical balance while executing complex tasks. Real-time computation, sensor fusion, model accuracy, and safety considerations add further complications. As a result, while promising demonstrations exist in simulation or constrained lab settings, achieving generalizable, real-time, whole-body control on physical robots remains an open and active area of research in robotics [4]-[6].

Controlling the movement of quadruped robots is generally more challenging than that of wheeled robots due to their nonlinear and multivariable kinematic models. Currently, various control methods for quadruped robots have been developed worldwide to address challenges such as maintaining dynamic balance, performing stability analysis, applying inverse dynamics approaches, as well as models based on spring-loaded inverted pendulums, model predictive control, hierarchical control, neural networks, and more [7]. In addition to enabling legged robots to walk with different gait patterns, recent scientific publications have demonstrated advancements in robot functionality, including locomotion stability under disturbances, gravity compensation, step length adjustment, gait trajectory optimization, terrain adaptability for quadruped mobility, and rough-terrain traversal in real-world environments [8]-[10].

This paper presents a comprehensive analysis and practical implementation of Model Predictive Control (MPC) for generating ground reaction forces (GRFs) used to control the joint angles of quadruped robots. To facilitate real-time control, the robot's full-body dynamics are suitably simplified, allowing the control problem to be reformulated as a convex optimization problem. This approach enables efficient computation of optimal control inputs over a receding time horizon while satisfying physical and environmental constraints. The proposed method is initially validated using a simplified quadruped model in simulation, demonstrating the reliability and robustness of the control strategy under idealized conditions. To further assess the practicality of the controller, it is integrated into the MuJoCo simulation environment using the Unitree Go1 quadruped robot model [11]. This integration showcases the controller's ability to generate feasible and physically realistic locomotion patterns. The successful simulation results lay a foundation for the future development of more advanced locomotion strategies. The developed algorithms can be implemented on a sufficiently powerful microcontroller system capable of executing these movements and potentially being applied in real-world quadruped robotic systems.

The rest of this paper is organized as follows. Section 2 presents the quadruped robot model including the definition of the system and the mathematical model. The locomotion control of the quadruped robot and simulation results are addressed in Section 3. Finally, the conclusions and future directions are provided in Section 4.

2. Quadruped robot model

2.1. Definition of coordinate systems

Let us denote the world coordinate system by $\mathcal{W}$ and the body coordinate system by $\mathcal{B}$. Based on the Z-Y-X Euler angle convention, the torso undergoes a sequence of rotations as illustrated in

Figure 1. In the body coordinate frame, the rotation of the torso around its Z-axis by the yaw angle ψ, followed by a rotation about the Y-axis by the pitch angle ϑ, and finally a rotation about the X-axis by the roll angle φ. Thus, the torso orientation can be represented as an Euler angle vector $\boldsymbol{\theta}_b = [\phi \quad \vartheta \quad \psi]^T$ [12]. The transformation of the rotation from the body to the world coordinates can be expressed as

$$ {}_{\mathcal{W}}^{\mathcal{B}}R = R_z(\psi)R_y(\vartheta)R_x(\phi) \tag{1}$$

where $R_n(\alpha)$ represents a positive rotation of α about the n-axis. The angular velocity in world coordinates can be found from the rate of change of these angles with

$$\omega = \begin{bmatrix} \cos(\vartheta)\cos(\psi) & -sin(\psi) & 0 \\ \cos(\vartheta)\sin(\psi) & \cos(\psi) & 0 \\ -\sin(\vartheta) & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix} \tag{2}$$

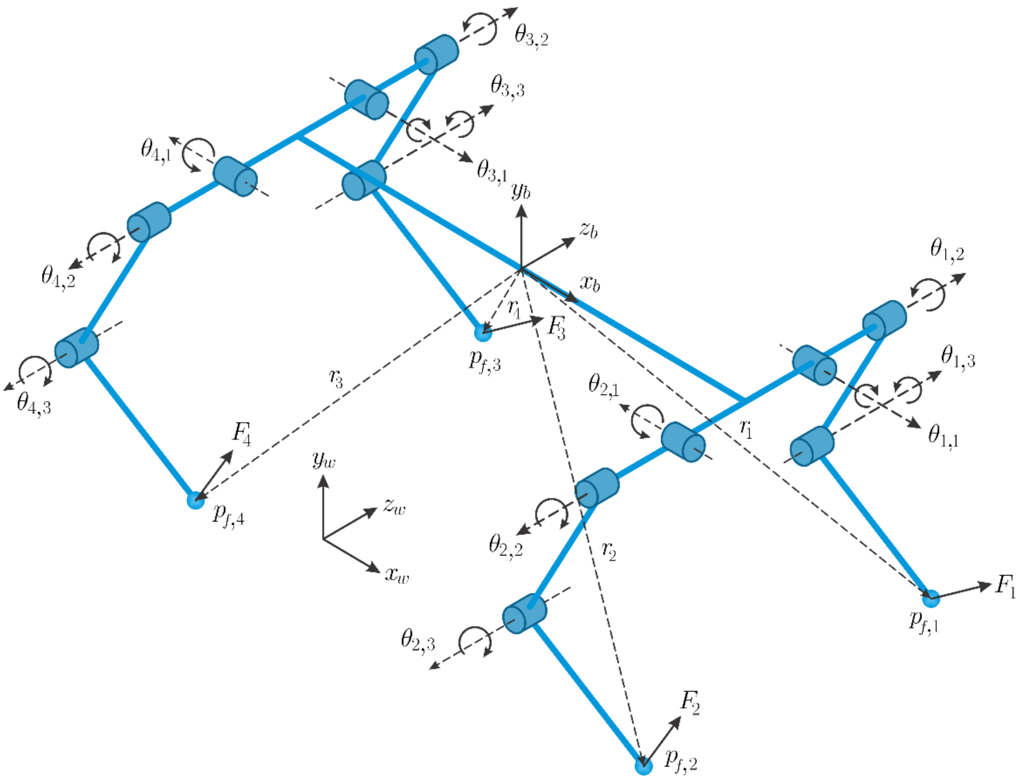


Figure 1. *Coordinate frames of 3-DoF robot leg*

In normal locomotion conditions, the pitch angle of the robot $\vartheta \neq 90^o$. This ensures that $cos(\vartheta) \neq 0$ and the invertible of Equation (2). Thus, we can find

$$\begin{bmatrix} \dot{\phi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi)/cos(\vartheta) & \sin(\psi)/cos(\vartheta) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ \cos(\psi)tan(\vartheta) & \sin(\psi)tan(\vartheta) & 1 \end{bmatrix} \omega \tag{3}$$

For small values of roll and pitch (ϕ, ϑ), Equation (3) can be approximated as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \omega \tag{4}$$

which is equivalent to

$$\begin{bmatrix} \dot{\phi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix} \approx R_z^T(\psi)\omega \tag{5}$$

2.2. Mathematical model of quadruped robots

2.2.1. Linearized dynamic model

The state vector is defined as

$$x_c = [\boldsymbol{\theta}_b \quad \boldsymbol{p}_b \quad \boldsymbol{\omega}_b \quad \dot{\boldsymbol{p}}_b \quad \boldsymbol{g}]^T \in \mathbb{R}^{13} \tag{6}$$

where $\boldsymbol{p}_b$ is the torso center of mass (CoM) position, $\boldsymbol{\omega}_b$ is the angular velocity, $\dot{\boldsymbol{p}}_b$ is the linear velocity, and $\mathbf{g}$ is the gravity vector.

The system dynamics can be expressed as [6]

$$\dot{x}_c = A_c(\psi)x_c + B_c(\boldsymbol{p}_{f,i} - \boldsymbol{p}_b)\boldsymbol{F}_f \tag{7}$$

where $A_c(\psi)$ is the orientation-dependent matrix involving $R_z(\psi)$, B_c is the map of GRFs at the 4 feet to acceleration and angular acceleration. The control input including the Ground Reaction Forces (GRFs), or the stacked vector of all four legs' forces, is $\boldsymbol{u} = \boldsymbol{F}_f = [\boldsymbol{F}_1 \quad \boldsymbol{F}_2 \quad \boldsymbol{F}_3 \quad \boldsymbol{F}_4]^T \in \mathbb{R}^{12}$, $\boldsymbol{F}_i \in \mathbb{R}^3$ is the ground reaction force at the foot of leg i.

2.2.2. The inverse dynamics for stance legs

Let $\boldsymbol{\theta}_i = [\theta_{1,i} \quad \theta_{2,i} \quad \theta_{3,i}]^T$, where $\theta_{1,i}$ is the joint angle of the hip abduction/adduction (HAA) around z-axis, $\theta_{2,i}$ is the joint angle of the hip flexion/extension (HFE) around y-axis, and $\theta_{3,i}$ is the joint angle of the knee flexion/extension (KFE) around y-axis of leg i, l_1 iss the offset along the x-axis from HAA to HFE joint, l_2 is the length of thigh from HFE to knee, and l_3 is the length of shank from knee to foot. The joint torque that the foot should apply onto the ground is

$$\tau_{stance,i} = -\boldsymbol{J}_i^T \boldsymbol{F}_i \tag{8}$$

where $\boldsymbol{J}_i = \frac{\partial \boldsymbol{p}_{f,i}}{\partial \boldsymbol{\theta}_i}$ is the Jacobian of leg i computed from forward kinematics (FK) at current foot pose, $\boldsymbol{\tau}_{stance,i}$ is the joint torques needed to apply $\boldsymbol{F}_i$.

2.2.3. Inverse Kinematics for swing legs

For swing leg i, desired foot trajectory $\boldsymbol{p}_{f,i}^d(t)$ is tracked using

$$\tau_{swing,i} = \boldsymbol{J}_i^{-1}\left(K_{p,f}\left(\boldsymbol{p}_{f,i}^d - \boldsymbol{p}_{f,i}\right) + K_{d,f}\left(\dot{\boldsymbol{p}}_{f,i}^d - \dot{\boldsymbol{p}}_{f,i}\right)\right) \tag{9}$$

where $\boldsymbol{p}_{f,i}$ is from forward kinematics, $\boldsymbol{p}_{f,i}^d$ is from planned swing foot Bezier trajectory [6], and $\boldsymbol{J}_i^{-1}$ is the inverse kinematics (IK) Jacobian, $K_{p,f}$, $K_{d,f}$ are the feedback gains.

2.2.4. Mapping the control input to joint angles

For stance legs, we can compute desired joint angles indirectly via a two-step process. First, for a stance leg i, we use the inverse dynamic Equation (8) to find the joint torques. Now, assume that the robot uses PD control and the actual joint states $\boldsymbol{\theta}_i$, $\dot{\boldsymbol{\theta}}_i$ as well as the control gains $K_{p,f}$, $K_{d,f}$ are known, we can find

$$\boldsymbol{\tau}_{stance,i} = K_{p,f}\left(\boldsymbol{\theta}_i^d - \boldsymbol{\theta}_i\right) + K_{d,f}\left(\dot{\boldsymbol{\theta}}_i^d - \dot{\boldsymbol{\theta}}_i\right) \tag{10}$$

Then, solving for the desired joint angles

$$\boldsymbol{\theta}_i^d = \boldsymbol{\theta}_i + K_p^{-1}\left(\boldsymbol{\tau}_{stance,i} - K_d\left(\dot{\boldsymbol{\theta}}_i^d - \dot{\boldsymbol{\theta}}_i\right)\right) \tag{11}$$

For swing legs (no GRFs), the desired joint angles come from inverse kinematics

$$\boldsymbol{\theta}_i^d = f_{IK}\left(\boldsymbol{p}_{f,i}^d\right) \tag{12}$$

where the function f_{IK} represents the IK function that computes the desired joint angles $\boldsymbol{\theta}_i^d \in \mathbb{R}^3$ for leg i, given the desired foot position $\boldsymbol{p}_{f,i}^d \in \mathbb{R}^3$ in the body or hip frame.

3. Locomotion control of quadruped robot

3.1. The control system configuration of the quadruped robot

The locomotion of a quadruped robot refers to the ability of the robot to move its body from one location to another using its four legs, similar to how animals like dogs, cats, or horses walk, trot, or run. Quadruped locomotion is the process by which a four-legged robot generates coordinated leg movements (gaits) to produce forward motion, turning, climbing, or other mobility tasks over different types of terrain. The control system configuration of the quadruped robot is shown in Figure 2.

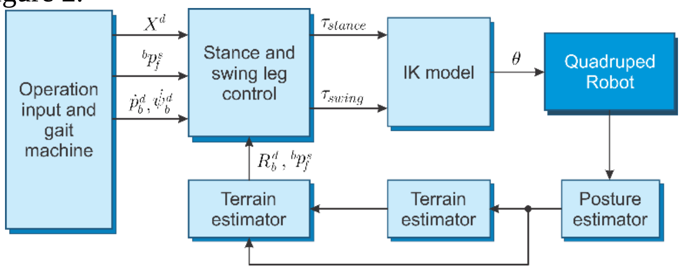


Figure 2. *The control configuration of quadruped robots*

3.2. Model Predictive Control

Model Predictive Control (MPC) is especially well-suited for quadruped robot control because it naturally handles the complexities of dynamic legged locomotion since quadrupeds have 12 Degrees of Freedom (DoF) (4 legs × 3 DoF per leg) to achieve coordinated motion of body posture, leg placement, or Center of Mass (CoM) control. MPC predicts how the robot will evolve over a finite time horizon for footstep planning, obstacle avoidance, balancing during motion, and coordinating swing/stance phases. MPC also incorporates constraints naturally for torque limits on actuators, friction, unilateral contacts, gait scheduling, etc.

Discretizing the linearization of the system given in Equation (7) using zero-order hold, we obtain

$$\boldsymbol{x}_{c,k} = A_{c,k}\boldsymbol{x}_{c,k} + B_{c,k}\boldsymbol{F}_{f,k} \tag{13}$$

where $A_{c,k}$ and $B_{c,k}$ represent the discrete time system dynamics.

With horizon length k, at each control step, we solve

$$\begin{gathered}\underbrace{\min}_{F_f^{0:k-1}} \sum_{i=0}^{k-1} \left(\boldsymbol{x}_{c,i+1} - \boldsymbol{x}_{c,i+1}^d\right)^T Q(.) + \left\|\boldsymbol{F}_{f,i}\right\|_R^2 \\ \boldsymbol{x}_{c,i+1} = A_{c,i}\boldsymbol{x}_{c,i} + B_{c,i}\boldsymbol{F}_{f,i} \\ f_{z,min} \le f_z \le f_{z,max} \\ \boldsymbol{F}_{f,j} \le \mu f_z\end{gathered} \tag{14}$$

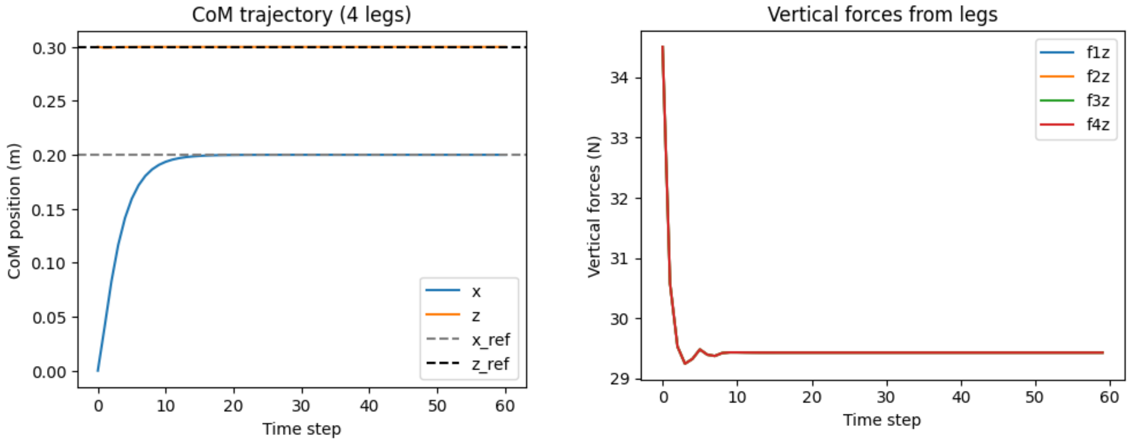
where $x_{c,i}$ and $\boldsymbol{F}_{f,i}$ are the system state and the control input at time step i, Q and R are diagonal positive semidefinite matrices of weights, f_z is the contact force, μ is the coefficient of static friction between the robot's foot and the ground.

3.3. Simulation results

For the sake of simplicity, we assume that the robot does not rotate around its center of mass (CoM), and the CoM orientation in space is aligned with the ground plane, or, in other words, the torso remains perfectly horizontal during locomotion (no pitch rotation). This removes the rotational degree of freedom from the model and reduces the state vector dimension. The quadruped robot is used for the simulation with the parameter in Table 1.

Table 1. *Robot parameters*

Element	Parameter	Symbol	Value	Unit
Mass	Total mass	m	12	kg
HAA leg	Mass	m_1	0.76	kg
	Length	l_1	0.08	m
	Inertia	$I_{xx,1}$	0.00140	$kg.m^2$
		$I_{yy,1}$	0.00131	$kg.m^2$
		$I_{zz,1}$	0.00078	$kg.m^2$
HFE leg	Mass	m_2	0.92	kg
	Length	l_2	0.213	m
	Inertia	$I_{xx,2}$	0.00587	$kg.m^2$
		$I_{yy,2}$	0.00560	$kg.m^2$
		$I_{zz,2}$	0.00105	$kg.m^2$
KFE leg	Mass	m_3	0.191	kg
	Length	l_3	0.213	m
	Inertia	$I_{xx,3}$	0.00331	$kg.m^2$
		$I_{yy,3}$	0.00329	$kg.m^2$
		$I_{zz,3}$	0.00005	$kg.m^2$

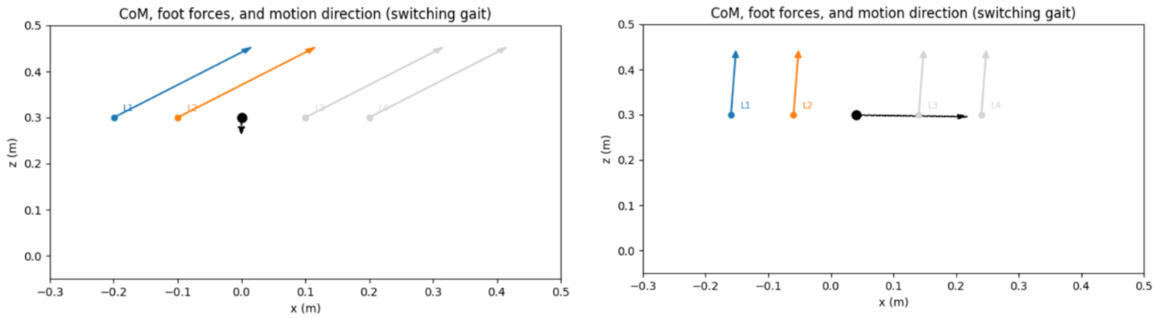


(a) *(b)*

Figure 3. *The center of mass (CoM) trajectory (a) and the vertical ground reaction forces (b)*

Figure 3 shows the center of mass (CoM) trajectory in both the horizontal (x) and vertical (z) directions and the vertical ground reaction forces (f_{iz}) from all four legs over time. The robot starts at rest and is tasked with reaching a target position. The controller tracks this target effectively. The x position (blue) converges to 0.2 m within approximately 15 time steps, while the z position (orange) stabilizes near the target height of 0.3 m. Both reference lines are closely followed with minimal overshoot or oscillation. The vertical ground reaction forces (f_{iz}) from all four legs are shown over time. The forces start with an initial adjustment and then stabilize around 29.3 N per leg. The total force matches the expected support for a robot mass of approximately 12 kg. The transition phase is smooth, with no oscillation or instability. All stance legs share the vertical load symmetrically.

Figure 4 shows the gait visualization for the CoM and foot forces, where the dots ($L1 - L4$) are the foot locations, the arrows show the ground forces (GRFs), the dot is the CoM, and the dashed arrow is the CoM velocity. It can be seen that, in Figure 4a, the $L1$, $L2$ are in stance with diagonal forces (propulsion), while CoM moves forward and slightly downward. In Figure 4b, the $L1$, $L2$ now apply vertical force, and the CoM continues forward (longer horizontal velocity).



(a) *(b)*

Figure 4. *Gait visualization: CoM (a) and foot forces (b)*

The next simulation is performed with the help of a runtime simulation module named MuJiCo (Multi-Joint Dynamics with Contact). This is a versatile physics engine designed to support research and development in fields such as robotics, biomechanics, computer graphics and animation, and machine learning. It provides fast and accurate simulation of articulated structures interacting with complex environments, making it well-suited for applications that require high-performance dynamic modeling.

(a)

(b)

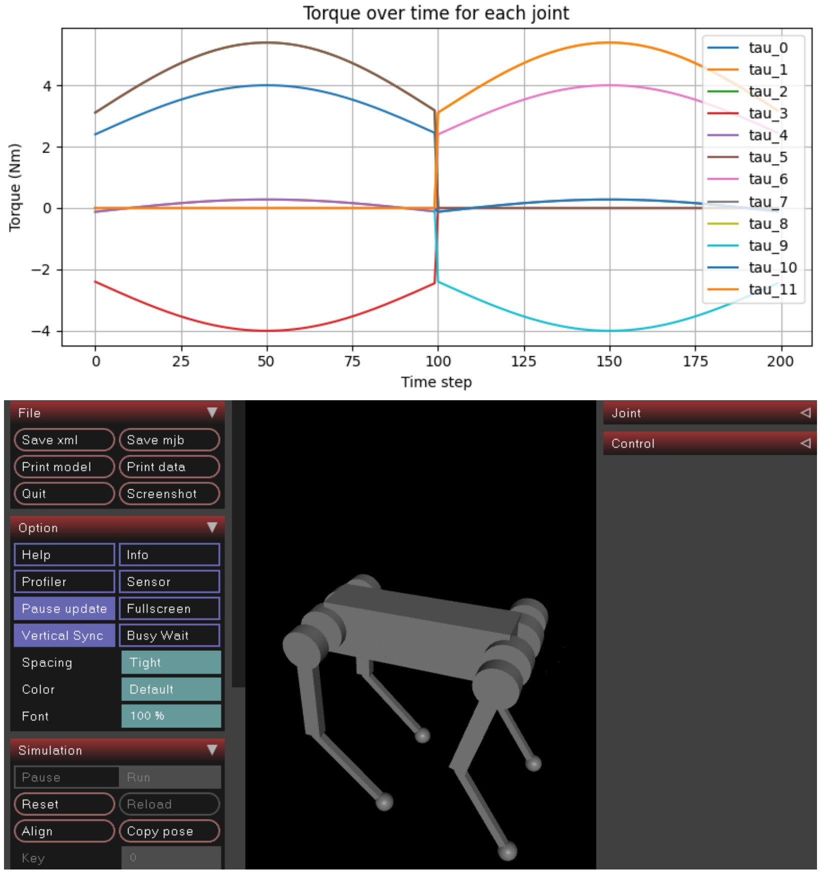


Figure 5. *Joint torque profiles during gait switching (a) and the MuJoCo graphical user interface (b)*

Figure 5 shows the joint torque profiles (τ_0 to τ_{11}) applied to each of the 12 joints of the quadruped robot during the MPC during gait switching and the MuJoCo graphical user interface (GUI). It shows a clear switching pattern from the front legs to the rear legs at time step 100. Initially, the left and right front legs (FL, FR) are active, producing torque. After switching, the left and right rear legs (RL, RR) take over. Torque values are consistent, indicating correct mapping via Jacobian transpose. The switching event produces sharp but expected transitions in command torques. The simulation shows that the MPC gait correctly switches legs, the CoM motion is smooth and balanced, and the forces match foot placement and gait phase.

4. Conclusions and recommendations

Despite the substantial simplifications applied to the robot's dynamic model, such as linearization of the robot's dynamics, the proposed controller demonstrates remarkable tracking performance across a simple locomotion pattern that imposes different requirements on ground reaction force distribution and phase coordination. The simulation results confirm that the model predictive controller successfully tracks desired CoM targets. It computes feasible and stable leg forces and converts these forces into accurate joint torques through inverse dynamics. This suggests that the simplified formulation retains sufficient fidelity to capture the essential dynamics necessary for real-time gait control. These may include dynamic balance maintenance on flat and uneven terrains, and extensions toward more generalized terrain-aware model predictive control-based locomotion planning for quadruped robots.

Acknowledgments

This research was supported by Thai Nguyen University of Technology (TNUT), Thai Nguyen, Vietnam.

REFERENCES

[1] M. Chignoli and P. M. Wensing, “Variational-based optimal control of underactuated balancing for dynamic quadrupeds,” *IEEE Access*, vol. 8, pp. 49785–49797, 2020.

[2] Z. Li, Y. Li, X. Rong, and H. Zhang, “Grid map construction and Terrain prediction for quadruped robot based on C-Terrain path,” *IEEE Access*, vol. 8, pp. 56572–56580, 2020.

[3] X. He, X. Li, X. Wang, F. Meng, X. Guan, Z. Jiang, L. Yuan, K. Ba, G. Ma, and B. Yu, “Running gait and control of quadruped robot based on SLIP model,” *Biomimetics*, vol. 9, pp. 1-24, 2024.

[4] H. Kazemi, V. J. Majd, and M. M. Moghaddam, “Modeling and robust backstepping control of an underactuated quadruped robot in bounding motion,” *Robotica*, vol. 31, pp. 423–439, 2013.

[5] F. Muhamad, J.-S. Kim, and J.-H. Park, “Learning robust perception-based controller for quadruped robot,” *IEEE Access*, vol. 11, pp. 94497–94505, 2023.

[6] P. Li, B. Yin, L. Zhang, and Y. Zhao, “Adaptive control algorithm for quadruped robots in unknown high-slope terrain,” *Journal of Engineering Research*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2307187724001305. [Accessed May 30, 2025].

[7] W. Yan, Y. Pan, J. Che1, J. Yu, and Z. Han, “Whole-body kinematic and dynamic modeling for quadruped robot under different gaits and mechanism topologies,” *Peer J. Computer Science*, 2021. [Online]. Available: https://peerj.com/articles/cs-821/. [Accessed May 30, 2025].

[8] G. Chen, S. Guo, B. Hou, and J. Wang, “Virtual model control for Quadruped Robots,” *IEEE Access*, vol. 8, pp. 140736–140751, 2020.

[9] K.-C. Han and J.-Y. Kim, “Posture stabilizing control of quadruped robot based on cart-invertedpendulum model,” *Intelligent Service Robotics*, vol. 16, pp. 521–536, 2023.

[10] M. Focchi, A. D. Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, “High-slope terrain locomotion for torque-controlled quadruped robots,” *Auton Robot*, vol. 41, pp. 259–272, 2017.

[11] J. A.-Padilla, J. Z. Zhang, S. Kwok, J. M. Dolan, and Z. Manchester, “Real-time whole-body control of legged robots with model-predictive path integral control,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. [Online]. Available: https://whole-body-mppi.github.io/. [Accessed May 30, 2025].

[12] J. J. Craig, *Introduction to robotics mechanics and control*. Third edition, Pearson Education International, 2005.