2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) | 979-8-3503-5536-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIM55361.2024.10636947

Whole-Body Control Loco-manipulation strategy for quadruped robots on deformable terrains

Chen Wang, Omer Kemal Adak, Raul Fuentes

***Abstract*— This paper introduces a loco-manipulation strategy for a quadruped robot operating on deformable terrains. A complete spatial robot model is built as the controlled system. Linear and nonlinear spring-damper models are adopted to represent the terrain deformation. The Operational Space Formulation (OSF) is employed to map the configuration space dynamics to task space dynamics. The Model Predictive Control (MPC) methodology is studied and implemented to direct the controlled system. The operational space motion equations and their interaction with deformable terrain are used as the fundamental model for the MPC framework. A force control strategy is employed on the legs to counteract gravitational effects and stabilize the body on deformable terrain, while a motion control strategy is applied to the arm for the object-manipulating task. The simulation results demonstrate that the proposed controller is feasible for these applications.**

I. INTRODUCTION

In recent years, the field of robotics has witnessed a paradigm shift towards the development and utilization of versatile robotic platforms capable of navigating complex and dynamic environments. Among these, quadruped robots have emerged as a promising solution, showcasing remarkable agility, stability, and adaptability. They have significant potential for conducting complex tasks within hazardous environments. During the execution process, both body movement and object manipulation are required. Development and experimentation on this topic are primarily configured under rigid terrain conditions. However, most of the hazardous environments contain irregular and deformable terrains. This paradigmatic shift is especially evident in the intersection of robotics and geomechanics, where the exploration and manipulation of deformable terrains have become a focal point of research and development. Deformable terrains, such as loose soils, sand, mud, and gravel, pose unique challenges for traditional wheeled or tracked robotic systems. The unpredictable nature of these terrains demands innovative approaches to locomotion and manipulation, and quadruped robots have proven to be exceptionally well-suited for such tasks. The utilization of quadruped robots on deformable terrains for loco-manipulation tasks holds significant promise and importance for their minimized environmental impact, enhanced versatility and adaptability, human-robot collaboration potential, and optimized energy efficiency.

The authors are with the Institute of Geomechanics and Underground Technology (GUT), RWTH Aachen University, Aachen, Germany. chen.wang2@rwth-aachen.de, (adak,raul.fuentes)@gut.rwth-aachen.de

In conclusion, the convergence of quadruped robots and deformable terrains for loco-manipulation tasks represents a transformative step forward in the field of robotics. The inherent advantages of quadrupedal locomotion, coupled with precise manipulation capabilities, unlock new possibilities for applications in diverse domains. As research continues to push the boundaries of robotics and geomechanics, the integration of quadruped robots into deformable terrains promises to redefine the landscape of robotic exploration, interaction, and intervention.

A. Literature Review

Loco-manipulation is one of the latest topics researched on mobile robotics. It involves two task aspects: locomotion, which involves changing the robot's position and orientation during the process, and manipulation, which aims to reach, manipulate, and move target objects from the external environment. Mobile robots with external manipulators are generally used to conduct loco-manipulation tasks, as most locomotive driving systems are not designed for manipulation. In numerous experiments on arm-mounted quadruped robots, it is typical to adopt a commercial robotic trunk and a custom-made arm specifically designed and attached to the robot. Several quadruped robots have also been developed and presented by research institutes worldwide. Gong et al. reviewed the object manipulation tasks performed by legged robots [1]. They classified these manipulation tasks into six categories: pushing, kicking, grasping with a single leg, double-legged grasping, whole-body grasping, and manipulating with non-locomotive arms.

Many researchers are focusing on Whole-Body Control (WBC) in legged robots [2], [3]. It is a very important method to exploit the full capabilities of the floating-base robots in compliant multi-contact interaction with the environment. By definition, a control system that is specifically designed to guarantee the execution of a single task, even if it uses all the joints of a robot, cannot be considered WBC. Hence, loco-manipulation is a great example of WBC to execute multiple tasks with a single control system. Ferrolho et al. presented a loco-manipulation strategy that has balance maintenance to a simultaneous combination of locomotion and manipulation through WBC for a quadruped robot [4]. Chiu et al. integrated a 6 Degrees of Freedom (DoF) Jaco arm onto ANYmal and proposed a WBC-based method since the joints of both the floating base and the arm are controlled by a unified controller [5]. Xin et al. applied a WBC-based controller on an arm-

979-8-3503-5536-9/24/$31.00 ©2024 IEEE

mounted quadruped robot with 24 DoF. A joint PD controller for the arm and a hierarchical quadratic programming method for the base are utilized for loco-manipulation mode [6].

MPC is one of the other methods referred to in loco-manipulation operations. Chang et al. proposed a control methodology for loco-manipulation tasks where a joint PDMPC controller combining MPC and PD compensator for unexpected dynamics in the robotic system is developed. The PD compensator takes effect when there is a large difference between the designed control model and the real system [7]. Sleiman et al. presented a WBC methodology in association with nonlinear MPC. This approach includes the coupling effects between the base, arm, and object in a unified planning framework through the WBC methodology [8].

B. Contribution

Although there is a variety of research focusing on loco-manipulation, none of them considers the behavior of deformable terrains. Although MPC has been a very hot topic on legged robots in recent years, interestingly, loco-manipulation has a few application examples. In this study, we propose a WBC strategy for loco-manipulation on deformable terrains. Considering these points, we can summarize this paper's contributions as follows:

- We integrated an MPC-based controller in a WBC strategy of the loco-manipıulaton task to improve the efficiency of the quadruped robot.
- We take into account both robotics and geomechanics properties and their interaction in order to increase model accuracy in MPC.
- We synthesized a control framework with the conjunction of WBC, MPC, and OSF.

This paper is structured as follows: Section II describes the modeling of a floating-base robot and the deformable terrain model. In Section III, the control framework is explained in detail. Section IV presents simulation results. The paper is concluded in Section V.

II. SYSTEM MODELING

This study encompasses two distinct models, namely the robot model and the geomechanics model. The complete model comprises these two models and their interaction. Our strategy is to employ this complete model as a basis for implementing MPC. Consequently, MPC will take into account not only robot states but also robot-ground interaction behavior. This section provides the details of the robotics and geomechanics models utilized in this study.

A. Robot Model

A spatial quadruped robot model equipped with a robotic arm is developed to simulate the performance of real quadruped robots. The body trunk is modeled as a simple cube to which the robotic arm and the four identical legs are attached. The robotic arm PUMA 560 is adopted as the manipulator of the model [9]. It is attached to the center of the body trunk's upper surface, while the upper end of the link serves as the end effector tracing and manipulating objects. At the base of the PUMA 560 robotic arm, a universal joint allows for its simultaneous rotation around both the y- and z-axes. The upper and lower arms are linked by a revolute joint, which moves the upper arm within its workspace. A spherical wrist connects the upper arm and the end effector and determines the position and orientation of the end effector.

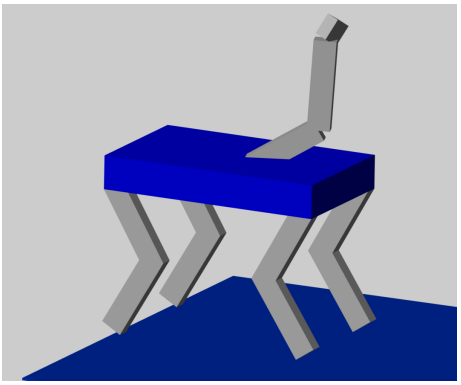

Fig. 1. Visualization of the quadruped robot.

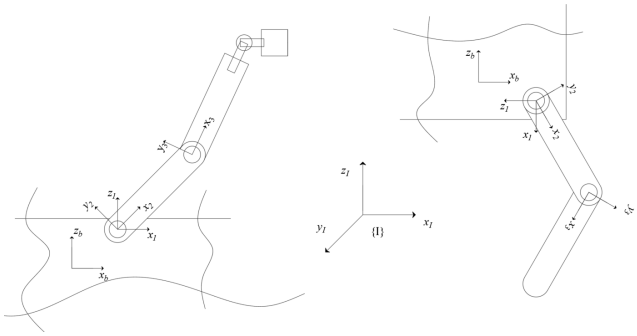

Fig. 2. Kinematics details of the quadruped robot.

To convert the Ground Reaction Force to the model, supporting it against the gravitational and acceleration effects, four identical legs are constructed. The supporting legs consist of two links each, with two joints connecting and actuating them. A universal joint is fixed at the upper end of the legs and connects them to the body trunk, allowing for rotation around both the x- and y-axes, which represent the movements of adduction/abduction and flexion/extension of the legs, respectively. Another revolute joint connects the upper and lower parts of the leg, enabling a rotation around the y-axis. The robot has a total of 24 DoF. Its visualization model is accomplished by using the Simscape Multibody Toolbox provided by Matlab, as displayed in Figure 1. The kinematic model of the legs, the PUMA 560 arm, and joint frames are demonstrated with an inertial frame in Figure 2.

According to the model design, the generalized coordinates

of the quadruped robot are stated as follows,

$$q = \begin{bmatrix} x_b \\ q_l \\ q_j \end{bmatrix}, \tag{1}$$

where $x_b \in SE(3)$ is the linear and angular positions of the body with respect to the inertial frame. $q_l \in \mathbb{R}^{12}$ represents the leg joint configurations, and $q_j \in \mathbb{R}^6$ refers to the joint positions of the arm. The motion equations for this quadruped robot model in the joint space are written as follows,

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = S^T\tau + J_c(q)^T F_c, \tag{2}$$

where $M(q) \in \mathbb{R}^{24\times24}$ is the inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{24}$ is the Coriolis and centripetal effects, $G(q) \in \mathbb{R}^{24}$ is the gravitational force, $J_c(q) \in \mathbb{R}^{18\times24}$ is the contact Jacobian of the robot with respect to the world frame, $F_c \in \mathbb{R}^{18}$ is the vector of linearly independent forces applied by the deformable terrain to the robot's feet and by the object to the robot's hand, $S = \begin{bmatrix} 0_{18\times6} & I_{18\times18} \end{bmatrix}$ is the selection matrix of the actuated joints, $\tau \in \mathbb{R}^{18}$ is the vector of joint torques.

According to the motion equations as indicated in Equation (2), the joint accelerations are calculated. Their velocities and positions are subsequently determined through integration upon their initial conditions. It is noticeable that all leg and arm joints are activated by motors, while the body trunk's position remains inactive, and therefore, the robot is under-actuated, which means the number of actuated joints is smaller than that of DoF. It increases the computation time for finding optimal solutions since the number of possible solutions is infinite. This problem is solved by designing the controller in the operational space, which is further explained in section III-A.

B. Deformable Terrain Model

The mechanics of deformable soil is determined by the ground subsidence. Various contact models are studied on deformable soil's normal and shearing stress. Considering the penetration and viscous effects, the contact mechanics can be built as a mass-spring-damper model or its variants. The standard form of the linear equation is written as,

$$m\ddot{x} + b\dot{x} + kx = F_{external}, \tag{3}$$

where x is the object's relative distance from its balanced position, m is the discrete mass value, b is the damping coefficient, k is the spring coefficient, and $F_{external}$ represents the sum of forces from the external environment.

In this study, a nonlinear mass-spring-damper model is established to calculate the normal reaction force. The nonlinear damping coefficient and the vertical ground reaction force are computed as follows:

$$\begin{aligned} b(z) &= \lambda z^n, \\ f_z &= -b(z)\dot{z} - kz^n, \end{aligned} \tag{4}$$

where $b(z)$ is the nonlinear damping coefficient, which is a function of the penetration depth, f_z is the vertical ground reaction force, z is the penetration depth into the ground, $\dot{z}$ is the penetration velocity, λ is the damping constant, k is the constant spring constant, n is the power of the penetration depth. The power of penetration depth n is commonly chosen close to one, depending on the geometry of the contact surface.

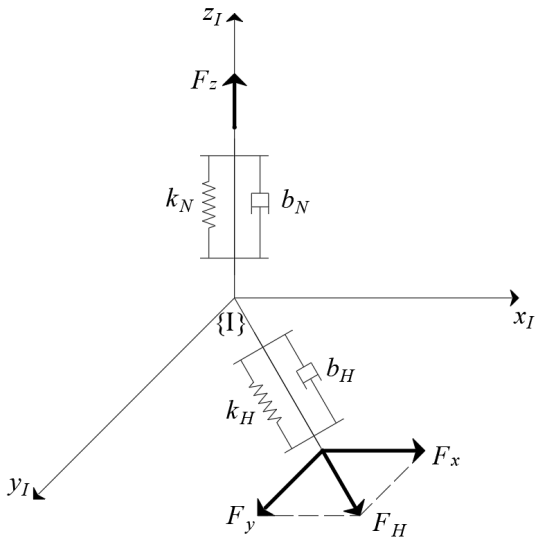

Fig. 3. Ground reaction model.

The conditions between the robotic feet and the ground are classified into three categories: when the penetration depth is negative, indicating there's no contact between the feet and the ground, no ground reaction force is produced in any direction. As the feet are penetrating the ground, both the spring force and the damping force are calculated as positive, whose direction is vertically upward. As the feet attempt to leave the ground, the spring force is computed as normal until the feet are completely off the ground, while the damping force is set to zero. The mechanics are illustrated in Figure 3. Table I shows Young's modulus of elasticity and the corresponding rigidity and damping coefficients of some deformable terrains [10], [11]. These parameters were employed in our simulations in order to accurately emulate deformable terrains.

III. CONTROL FRAMEWORK

In practice, the joint motions are usually not directly controlled to perform loco-manipulation tasks. Therefore, the control is switched from the joint space to the operational space, meaning that another set of control frameworks needs to be designed and configured to achieve the motions. In this section, the dynamics of the floating-base robot in the operational space are derived. The control framework is designed and presented based on the operational space properties.

A. Operational Space Formulation

In order to accurately represent the dynamics of a floating base robot in the operational space, it is crucial to distinguish between the dynamics of the non-actuated body and the motion equation [12]. The generalized coordinates are modified by

TABLE I

DEFORMABLE TERRAIN PARAMETERS.

Terrain Type	Young's Modulus (kNm^{-2})	K_N (Nm^{-1})	B_N (Nsm^{-1})	K_H (Nm^{-1})	B_H (Nsm^{-1})
Concrete	3×10^7	3.4×10^9	1.8×10^5	2.6×10^9	1.5×10^5
Gravel	$(1-2)\times10^5$	2.3×10^7	1.4×10^4	1.7×10^7	1.2×10^4
Sand	$(1-8)\times10^4$	9.1×10^6	9×10^3	6.9×10^6	7.9×10^3
Compact clay	$(3-15)\times10^3$	1.7×10^6	3.9×10^3	1.3×10^6	3.4×10^3
Loose clay	$(5-30)\times10^2$	3.4×10^5	1.8×10^3	2.6×10^5	1.5×10^3

selecting the actuated joints.

$$\bar{q} = S_j \begin{bmatrix} x_b \\ q_l \\ q_j \end{bmatrix}. \tag{5}$$

Here $S_j = \begin{bmatrix} 0_{18\times6} & I_{18\times18} \end{bmatrix}$ is the joint selection matrix. The joint selection matrix's generalized inverse is employed to convert the generalized dynamics into actuated joint space dynamics,

$$S_j^\dagger = M^{-1} S_j^T \bar{M}, \tag{6}$$

where $\bar{M} = (S_j M^{-1} S_j^T)^{-1}$ is the actuated joint space inertia matrix. The generalized inverse of the joint selection matrix transpose ($(S_j^\dagger)^T$) is multiplied by (2) to transform the system into actuated joint space dynamics.

$$\bar{M}(\bar{q})\ddot{\bar{q}} + \bar{C}(\bar{q},\dot{\bar{q}}) + \bar{G}(\bar{q}) = \tau + \bar{J}_c(\bar{q})^T F_c, \tag{7}$$

where $\bar{C}(\bar{q},\dot{\bar{q}}) = (S_j^\dagger)^T C(q,\dot{q})$ is the Coriolis and centrifugal forces, $\bar{G}(\bar{q}) = (S_j^\dagger)^T G(q)$ is the gravitational effects and $\bar{J}_c(\bar{q}) = J_c(q)(S_j^\dagger)^T$ is the Jacobian of the actuated joint space. It is possible to derive the OSF by multiplying (7) with the generalized inverse of Jacobian transpose ($(J_c(\bar{q})^\dagger)^T$):

$$(J_c^\dagger)^T \left(\bar{M}\ddot{\bar{q}} + \bar{C} + \bar{G} = \tau + \bar{J}_c{}^T F_c \right), \tag{8}$$

which equals,

$$\Lambda(\bar{q})\ddot{x}_e + \mu(\bar{q},\dot{\bar{q}}) + p(\bar{q}) = F_e + F_c, \tag{9}$$

where $J_c^\dagger = \bar{M}^{-1}\bar{J}_c{}^T\Lambda$ is the dynamically consistent generalized inverse of actuated joint space Jacobian that minimizes the instantaneous kinetic energy of the robot [13]. Since the kinematics relation between joint space and operation space is defined by $\dot{x}_e = \bar{J}_c\dot{\bar{q}}$ and $\ddot{x}_e = \bar{J}_c\ddot{\bar{q}} + \dot{\bar{J}}_c\dot{\bar{q}}$, the relation between the terms in the actuated joint space and the operational space is derived,

$$\begin{aligned} \Lambda(\bar{q}) &= (\bar{J}_c\bar{M}^{-1}\bar{J}_c{}^T)^{-1} \\ \mu(\bar{q},\dot{\bar{q}}) &= (J_c^\dagger)^T\bar{C} - \Lambda\dot{J}_c^\dagger\dot{\bar{q}}, \\ p(\bar{q}) &= (J_c^\dagger)^T\bar{G}. \end{aligned} \tag{10}$$

Here $\Lambda(\bar{q})$, $\mu(\bar{q},\dot{\bar{q}})$ and $p(\bar{q})$ are the operational space inertia, Coriolis and centrifugal forces, and gravitational effects, respectively. $F_e = (J_c^\dagger)^T\tau$ is the end effector force, $\dot{\bar{J}}_C(\bar{q}) = \dot{J}_C(q)S_j^\dagger$ is the time derivative of the actuated joint space Jacobian and x_e is the position of the end effector in the operation space. The positional acceleration in the operational space $\ddot{x}_e$ is calculated, based on which the velocities $\dot{x}_e$ and positions x_e are further determined by integral.

B. Model Predictive Control

MPC is an advanced process control method that considers the entire controlled system and takes a future time period to find an optimal solution. It is studied and implemented as the basis of controller design in this work. MPC can be used across various system types. In this paper, the traditional state-space model is adopted as the basis of the dynamic model. The MPC Toolbox provided by MATLAB is utilized to build the experimental model and the simulation environment. Here the controlled system is discretized and linearized into a linear time-invariant (LTI) system according to the sample time and specific operating points, which are system characteristics conditions representing the relationship between inputs and outputs. At the time step k, a predicted control sequence $U_p(k)$ over the prediction horizon p is generated, whose first element $u_p(k|k)$ is implemented to the system, and the others are discarded. This process is repeated at each time step as the references and system states are updated online.

All inputs are treated as manipulated variables (joint torques) as the controller adjusts their values. All outputs are set as measured outputs (joint positions). The effects of disturbance and measurement noise are neglected. During the process, both manipulated variables and measured outputs can be constrained within a certain range. Multiple control sequences are computed for each time step. The MPC controller evaluates their performance by cost functions and selects the one with the minimum value as the optimal solution to be applied to the system. The toolbox solves the optimization problem using an active-set quadratic program (QP) solver by default, and the cost function includes the terms regarding the error between references and measured system states and the changing rates of input variables.

$$L(y(k),U_k) = L_y(y(k),U_k) + L_{\Delta u}(y(k),U_k), \tag{11}$$

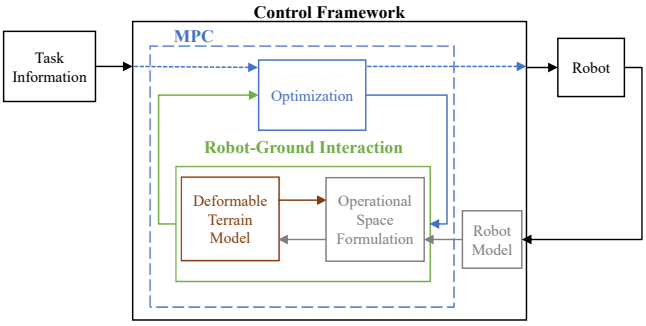

Fig. 4. Control framework.

where $L(y(k),U_k)$ is the sum of cost function, $L_y(y(k),U_k)$ is the cost function term of reference tracking, $L_{\Delta u}(y(k),U_k)$ is the cost function term of input changing rates. The control effect is implicitly achieved by tuning the parameters of the corresponding elements.

In our implementation, we divide the control task into two separate classes. A force control strategy is employed on the legs to counteract gravitational effects and stabilize the body on deformable terrain. Simultaneously, motion control is applied to the arm to facilitate manipulation of the target object. The previous section explained how to transform joint space motion equations into operational space motion equations. We offer operational space motion equations and their interaction with deformable terrain as the fundamental model for the MPC framework. In the force control part, the main goal is to determine the exact contact reaction from the deformable terrain to preserve the stability of the robot's trunk. The MPC algorithm computes the necessary operational space forces needed to fulfill this task. It then converts the forces into the configuration space, which then dictates the required torques on the joints to produce the intended forces at the feet. Meanwhile, in motion control, MPC evaluates changes in the position of both the trunk and the arm's gripper. It determines the ideal forces required by the end effector for the object-manipulating task. Similar to force control, the forces are converted into the configuration space, specifying the torques needed at the arm joints to generate the desired forces at the gripper.

Essentially, this is a response to the inquiry regarding the precise amount of torque required to generate a specific force at the end effectors, as determined by MPC,

$$\tau = \bar{J}_c^{\,T} F_e. \tag{12}$$

Since the under-actuation issue does not occur in the presented OSF, the computation time is relatively reduced for MPC. The control framework's overall structure is depicted in Figure 4. We are referring the readers to [14], [15] for more details related to the locomotion part and force-motion control strategy in OSF.

IV. SIMULATION RESULTS

Simulations are employed for the verification of the proposed control framework described in Section III. The simulation environment is built in MATLAB & Simulink, similar to work at [16]. Simulation parameters are given in Table II. The simulations were conducted on gravel, sand, compact clay, and loose clay. The parameters listed in Table I were utilized to modify the dynamics of the contact model in order to emulate the actual deformable terrain behaviors. The object was randomly located in the arm's workspace. Three random positions were tested for each soft terrain type to confirm the robustness of the proposed controller. The same controller algorithm is applied for each distinct test. The controller is not updated according to different soil types. We shared an illustration of the end effector motion in Figure 5. We chose the third target position of the sand simulation as an illustrative model because it exhibits lower error rates than other simulation outcomes. The end effector trajectory appears to move downward initially before reaching the target, giving the impression that the end effector has descended as the robot adjusts its body height. When viewed in relative motion, it reaches the target smoothly. The complete results are provided in Table III. The results indicate that absolute steady-state errors are varying between 5 *mm* and 2.5 *cm*. The robot arm

TABLE II
SIMULATION PARAMETERS

Simulation Parameters	
Definition	***Value (Unit)***
Body length-height-width	$1-0.15-0.6$ (m)
Thigh length-height-width	$0.4-0.06-0.1$ (m)
Shank length-height-width	$0.4-0.06-0.1$ (m)
Arm length-height-width	$0.9-0.06-0.1$ (m)
Body mass	50 (kg)
Sum of leg masses	20 (kg)
Arm mass	5 (kg)
Gravitational acceleration	9.81 kgm/s^2
Simulation sampling time	0.5 ms
Controller sampling time	0.02 s
Prediction and control horizons	$10-3$ s

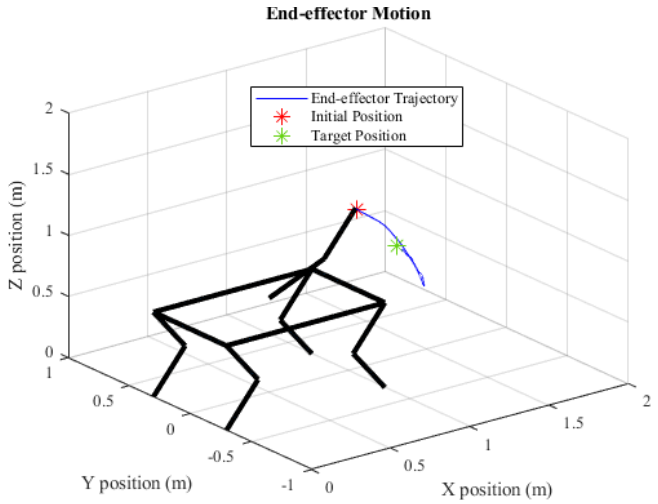

Fig. 5. End effector motion.

TABLE III

SIMULATION RESULTS

Simulation Results			
Soil Type	***Target position (x-y-z) (m)***	***End effector position (x-y-z) (m)***	***Absolute steady-state error (x-y-z) (m)***
Gravel	0.9741 0.1648 1.2028	0.9953 0.1870 1.1920	0.0212 0.0222 0.0108
	1.1207 −0.2631 1.2325	1.1438 −0.2388 1.2110	0.0231 0.0243 0.0215
	1.0368 0.1430 1.1658	1.0515 0.1293 1.1572	0.0147 0.0137 0.0086
Sand	1.0506 0.2062 1.2662	1.0638 0.2143 1.2596	0.0132 0.0081 0.0066
	1.1301 −0.0862 1.2305	1.1484 −0.1097 1.2136	0.0183 0.0235 0.0169
	0.9919 −0.3223 1.2315	1.0037 −0.3283 1.2262	0.0118 0.0060 0.0053
Compact Clay	1.0897 −0.3631 1.0586	1.1147 −0.3387 1.0334	0.0250 0.0244 0.0252
	0.9766 0.2588 1.2220	0.9879 0.2438 1.2168	0.0113 0.0150 0.0052
	1.0086 −0.1244 1.0817	1.0297 −0.1032 1.0656	0.0211 0.0212 0.0161
Loose Clay	1.0538 −0.3745 1.0740	1.0773 −0.3984 1.0535	0.0235 0.0239 0.0205
	1.0462 0.1945 1.1670	1.0654 0.2128 1.1539	0.0192 0.0183 0.0131
	1.0978 −0.1785 1.1492	1.1200 −0.1574 1.1303	0.0222 0.0211 0.0189

manages to reach the objects at different locations on different deformable terrains successfully.

V. CONCLUSIONS

The full loco-manipulation strategy of a quadruped robot on deformable terrains is presented in this paper. Modern quadruped robots are designed to be versatile and capable of executing complex loco-manipulation tasks in challenging surroundings. Terrain deformability is one of the major challenges that may deteriorate the quadruped robot's performance. To address this issue, a general control framework is proposed to direct the motions of the entire robot. This research has valuable significance as it reflects the real situations of mobile robots functioning in the wild field.

During the simulations, it is expected that the body trunk remains stable and the arm reaches the target. Our loco-manipulation strategy divides the control task into two separate classes. A force control strategy is employed on the legs to counteract gravitational effects and stabilize the body on deformable terrain. Simultaneously, motion control is applied to the arm to facilitate manipulation of the target object. OSF is utilized as a fundamental model of an MPC. The MPC algorithm computes the necessary operational space force needed to fulfill this task. This force is mapped into the configuration space, specifying the torques needed at the joints to generate the desired force in the operational space.

The results indicate that the robotic models achieved their desired positions on the emulated deformable terrains. The MPC controllers successfully realized the motions for the dynamic systems while maintaining stability. It is worth noticing that although the body trunk is not directly actuated, its position and orientation are closely monitored and controlled. This serves as an excellent demonstration of the feasibility of MPC controllers in managing under-actuated Multi-Input-Multi-Output systems. With respect to future endeavors, our objective is to apply our algorithm to a real quadruped robot.

REFERENCES

[1] Y. Gong, G. Sun, A. Nair, A. Bidwai, and R. CS et al., "Legged robots for object manipulation: A review," Frontiers in Mechanical Engineering, vol. 9, 2023.

[2] C.D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3359-3365, 2017.

[3] M. Risiglione, V. Barasuol, D.G. Caldwell, and C. Semini, "A Whole-Body Controller Based on a Simplified Template for Rendering Impedances in Quadruped Manipulators," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 9620-9627, 2022.

[4] H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, and S. Vijayakumar, "RoLoMa: Robust Loco-Manipulation for Quadruped Robots with Arms," Autonomous Robots, vol. 47, no. 8, pp. 1463-1481, 2023

[5] J.R. Chiu, J.P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, "A collision-free mpc for whole-body dynamic locomotion and manipulation," IEEE International Conference on Robotics and Automation, pp. 4686-4693, 2022.

[6] G. Xin, F. Zeng, and K. Qin, 'Loco-manipulation control for arm-mounted quadruped robots: Dynamic and kinematic strategies," Machines, vol. 10, no. 8, pp. 719, 2022

[7] X. Chang, H. Ma, and H. An, "Quadruped Robot Control through Model Predictive Control with PD Compensator," International Journal of Control, Automation and Systems, vol. 19, pp. 3776-3784, 2021.

[8] J.-P. Sleiman, F. Farshidian, M.V. Minniti, and M. Hutter, "A Unified MPC Framework for Whole-Body Dynamic Locomotion and Manipulation," IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 4688-4695, 2021.

[9] P.I. Corke, and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," IEEE International Conference on Robotics and Automation, pp. 1608-1613, 1994.

[10] M.F. Silva, J.A.T. Machado, and A.M. Lopes, "Modelling and simulation of artificial locomotion systems," Robotica, vol. 23, no. 5, pp. 595-606, 2005

[11] L. Ding, H. Gao, Z. Deng, J. Song, and Y. Liu, et al., "Foot–terrain interaction mechanics for legged robots: Modeling and experimental validation," The International Journal of Robotics Research, vol. 32, no. 13, pp. 1585-1606, 2013

[12] L Sentis and O. Khatib, "Control of free-floating humanoid robots through task prioritization," IEEE International Conference on Robotics and Automation, pp. 1718-1723, 2005.

[13] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," IEEE Journal on Robotics and Automation, vol. 3, no. 1, pp. 43-53, 1987.

[14] O.K. Adak, B. Bahceci, and K. Erbatur, "Hybrid Force-Motion Control for One-Legged Robot in Operational Space," IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 905-910, 2021.

[15] O.K. Adak, B. Bahceci, and K. Erbatur, "Whole-body pace gait control based on centroidal dynamics of a quadruped robot," IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1677-1682, 2022.

[16] O.K. Adak, B. Bahceci and K. Erbatur, "Modeling of a Quadruped Robot with Spine Joints and Full-Dynamics Simulation Environment Construction," arXiv preprint arXiv:2203.09622, 2022.