Received: 24 July 2023 | Revised: 20 February 2024 | Accepted: 6 April 2024

DOI: 10.1002/rob.22350

RESEARCH ARTICLE

WILEY

VERO: A vacuum-cleaner-equipped quadruped robot for efficient litter removal

Lorenzo Amatucci[1,2] | Giulio Turrisi[1] | Angelo Bratta[1] | Victor Barasuol[1] | Claudio Semini[1]

[1]Dynamic Legged Systems Laboratory, Istituto Italiano di Tecnologia (IIT), Genova, Italy

[2]Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS), Università di Genova, Genova, Italy

Correspondence
Claudio Semini, Dynamic Legged Systems Laboratory, Istituto Italiano di Tecnologia (IIT), Genova, Italy.
Email: claudio.semini@iit.it

Funding information
European Union—NextGenerationEU; Ministry of University and Research (MUR); National Recovery and Resilience Plan (NRRP), Project RAISE, Grant/Award Number: ECS00000035

Abstract

Litter nowadays presents a significant threat to the equilibrium of many ecosystems. An example is the sea, where litter coming from coasts and cities via gutters, streets, and waterways, releases toxic chemicals and microplastics during its decomposition. Litter removal is often carried out manually by humans, which inherently lowers the amount of waste that can be effectively collected from the environment. In this paper, we present a novel quadruped robot prototype that, thanks to its natural mobility, is able to collect cigarette butts (CBs) autonomously, the second most common undisposed waste worldwide, in terrains that are hard to reach for wheeled and tracked robots. The core of our approach is a convolutional neural network for litter detection, followed by a time-optimal planner for reducing the time needed to collect all the target objects. Precise litter removal is then performed by a visual-servoing procedure which drives the nozzle of a vacuum cleaner that is attached to one of the robot legs on top of the detected CB. As a result of this particular position of the nozzle, we are able to perform the collection task without even stopping the robot's motion, thus greatly increasing the time-efficiency of the entire procedure. Extensive tests were conducted in six different outdoor scenarios to show the performance of our prototype and method. To the best knowledge of the authors, this is the first time that such a design and method was presented and successfully tested on a legged robot.

KEYWORDS
autonomous litter removal, environment preservation, legged locomotion, object detection, optimal control, path planning, quadruped robot

1 | INTRODUCTION

Litter is an economic, biological and health-related problem (UNEP, 2006), and along with climate change, loss of biodiversity, ocean acidification, and overpopulation, it is widely recognized as one of the five main challenges that we are facing as humanity nowadays (Rangel-Buitrago et al., 2022). The increase in urbanization and consumerism are driving factors in the rise of trash worldwide (Song et al., 2015), and a considerable percentage of the produced waste is not properly disposed of every year. This behavior contaminates soil and water, increasing the risk for our planet and its inhabitants. Litter can easily reach marine environments by ending up in gutters, streets, and waterways (Novotny et al., 2009). When it breaks down, it can release toxic chemicals, metals, and microplastic, causing changes in the fauna gene and protein expression, inflammation, and disruption of feeding behavior (Bhuyan, 2022), finally altering the equilibrium of our ecosystem.

Among all the different types of waste, cigarette butts (CBs) or cigarette filters are the second most common undisposed waste

worldwide (Rangel-Buitrago et al., 2022). More than 5 trillion cigarettes were consumed in 2016, two-thirds of which were supposedly not thrown away correctly (Araújo & Costa, 2019), causing, therefore, a vast environmental problem. In fact, CBs are usually composed of a cellulose-acetate-based material which does not biodegrade and can remain in the environment for more than 10 years (Belzagui et al., 2020), freeing up in the marine environments microplastic and more than 700 toxic chemicals (Micevska et al., 2006; Slaughter et al., 2010), for example, nicotine, 70 of which are known to cause cancer in humans and animals (Belzagui et al., 2020).

Given the dreadful impact on the environment caused by the incorrect disposal of CBs (World Health Organization, 2022), many awareness campaigns have been promoted by governments over the years to mitigate the problem. Nevertheless, a waste collection procedure performed directly in-loco, such as in cities, beaches, and other coastal areas, is still necessary, as demonstrated in the *Ocean Conservation* Cleanup reports (Ocean Conservancy, 2023). Unfortunately, the collection procedure faces many challenges that make an effective scale-up impossible. While CBs on streets can be efficiently removed with dedicated vehicles, the majority of other areas require labor-intensive, manual collection. This is mainly due to the difficulties given by the CBs' different sizes, shapes, and materials, as well as the uneven terrain present in the different environments they are dispersed into.

Many projects have been developed, in both industry and academia, trying to automate the aforementioned procedure. Most of these works have been designed with a specific target environment in mind. For example, the robots Angsa (Angsa Robotics, 2024) and Pixies (Pixies Urban Lab, 2024) are small-sized wheeled systems that aim for a more conventional scenario, such as grass and urban environments, whereas BeachBot (Project Beach Bot, 2024) and Bebot (4Ocean, 2024) have been developed to be operated on sandy beaches. For this reason, they utilized tracks and specially designed wheels to move. The developed machines also differ in the approach chosen for the collection. In Zapata-Impata et al. (2018), Bai et al. (2018) and Liu et al. (2021), the authors utilized an arm to selectively collect litter, and while this method allows for the precise selection of the object to collect, it requires a more sophisticated vision and control pipeline that slows down considerably the collection speed. On the other hand, other projects like (4Ocean, 2024) opt for a more conventional mechanism that collects everything on the ground without the possibility of differentiating between trash, plants, or animals. Even though all these approaches have brought progress towards autonomous litter removal, their traversability capability is ultimately limited by their design. In fact, in the case of small obstacles on their path or uneven terrain, which is a common situation in coasts and cities, wheeled robots have clear limitations that reduce the amount of litter that can be collected.

Recently, quadrupedal robots have shown astonishing results in traversing challenging terrain. In Lee et al. (2020), the authors proposed a learning-based controller able to cope with unknown disturbances and uneven terrains. Their outdoor experiments clearly show the great stability and versatility achievable by a quadruped robot. Furthermore, safety can be explicitly considered during locomotion by adopting vision-based correction for precise footstep planning to avoid having a foot stuck (Jenelten et al., 2022; Villarreal Magaña et al., 2019); in this way, a long-term autonomous operation of these systems can be achieved. The achievable safety and the great agility that distinguishes quadruped robots from wheeled robots show great potential in adopting these systems for litter removal, especially in human-designed, uneven, and complex scenarios. Yet, to the best of the author's knowledge, no legged robot prototypes have been developed specifically for this application. We present the autonomous litter removal prototype *VERO* that builds upon a state-of-the-art control framework for legged locomotion. On the basis of the quadrupedal robot Aliengo of Unitree,[1] our design is able to autonomously collect undisposed CBs while traversing a variety of complex scenarios that are mostly inaccessible to wheeled robots. We equipped our prototype with a vacuum cleaner attached to its trunk and a custom-made nozzle connected to one of its legs, enabling the possibility of performing collection just by stepping with the attached nozzle over the detected CBs (Figure 1). From a software point of view, we obtain a precise collection via a multilevel framework that comprehends image recognition via neural networks, planning for time-optimal collection paths, and visual servoing for precise positioning of the robot's foot.

1.1 | Contribution

The main contributions of this work are:

- the design and integration of a novel platform for litter collection based on a legged robot, a vacuum cleaner, and an innovative foot nozzle design;
- the development of a perception, estimation and control pipeline that identifies the litter, filters out duplicated detections, and plans the optimal collection path;
- an extensive real-world validation of the overall system performance, with numerical and qualitative analysis of the platform advantages and limitations, in scenarios where CBs are commonly dispersed into;
- finally, we want to point out that this work represents, to the best of the author's knowledge, the first instance of utilizing a legged quadruped robot where its legs perform an additional task *during* locomotion, eliminating the need for extra limbs to accomplish the designated function.

1.2 | Outline

The paper is organized as follows: Section 2 gives an overview of our litter collection prototype, whereas Section 3 describes the proposed

[1] https://www.unitree.com/products/aliengo/

approach delving into the developed CBs detection module, along with a state estimation and planning procedure for precise and time-optimized waste collection. In Section 4, we detail our locomotion controller, while in Section 5 we report on the application of the proposed prototype, showing both indoor and outdoor experimental results. Finally, a discussion and some general conclusions about the approach are drawn respectively in Sections 6 and 7.

FIGURE 1 A snapshot of our quadrupedal robot prototype walking at the beach of Vernazzola, Genova, Italy. [Color figure can be viewed at wileyonlinelibrary.com]

2 | VACUUM-CLEANER HARDWARE

For this work, the commercial quadruped robot Aliengo has been adapted by adding a 2-kg commercial vacuum cleaner on top of its trunk, while a custom-designed nozzle was attached to its left-front foot (Figure 2). As will be explained later, our design can be easily modified to connect the vacuum cleaner to two or all the available feet, increasing the overall system efficiency, described here as the time needed to perform the collection procedure and adaptability. We designed and 3D-printed a custom-made nozzle to maximize the area around the foot of the robot where the CBs can be collected without sacrificing the suction ability of the vacuum cleaner and the locomotion capability of the robot. In fact, a wider suction area would have required either a more powerful vacuum cleaner or a smaller gap between the nozzle and the terrain. The last point has been observed to be the most critical in our experiments since the need to have the nozzle close to the terrain increased the collisions with the ground. All the impacts that do not happen with the foot disturb the stability of the motion and the accuracy of the stepping. In Figure 2, the reader can clearly observe the adopted solution. Even if our framework enables precise footstep placement on top of the detected CBs, a few centimeter mismatches can result in a nonsuccessful collection procedure. With a bigger nozzle area, this tolerance is increased.

The vacuum cleaner mounted on top of the platform is a modified commercial machine. We changed the power supply to allow external control with a microcontroller board and a relay, as

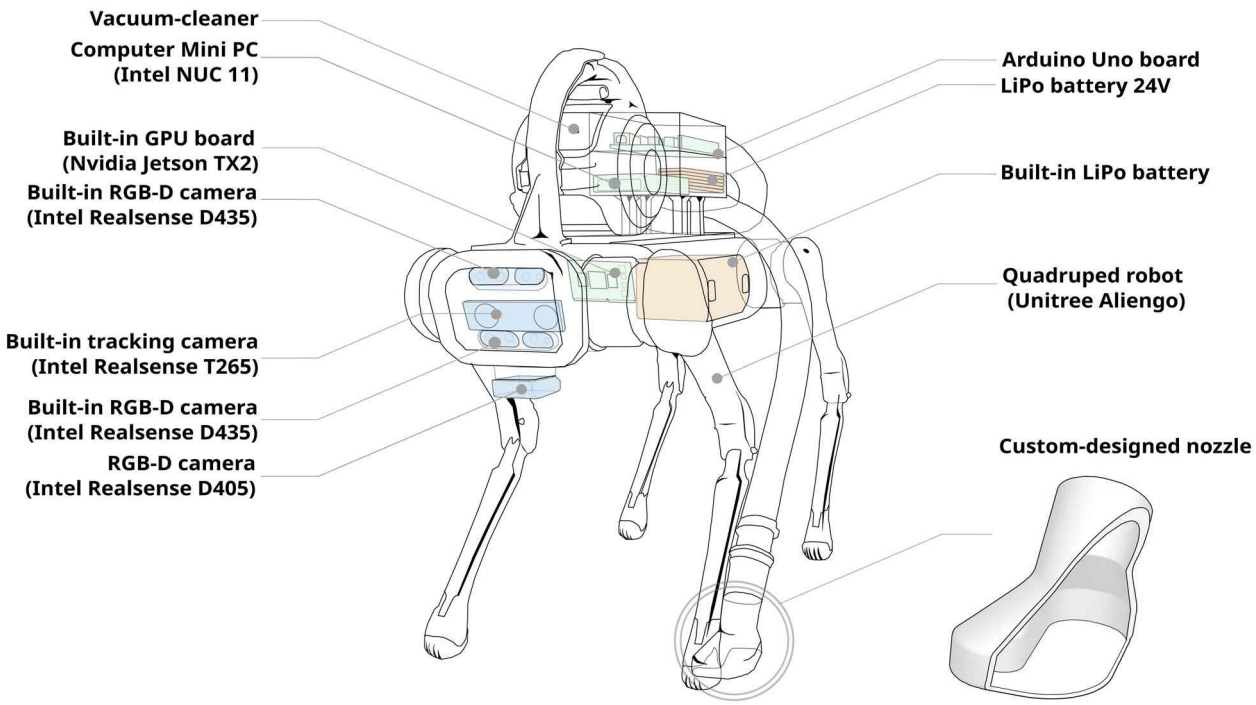

FIGURE 2 Prototype hardware: Image of the presented autonomous litter removal prototype with labels for the key components. The system comprises a quadruped robot with built-in sensors and computation, a vacuum cleaner with custom-designed nozzle, additional sensors, computers, electronic boards, and a LiPo battery. PC, personal computer. [Color figure can be viewed at wileyonlinelibrary.com]

shown in Figure 2. In particular, an Arduino UNO board is connected to an onboard MiniPC and is powered with 5V. A 24 V Lipo battery and a DC–DC converter 24–12 V power the relay. Commanded by the microcontroller, the relay is only closed when the vacuum cleaner must be turned on to optimize battery life and avoid collecting undesired objects during walking.

3 | THE LITTER COLLECTION PROCEDURE

In this section, we describe the main components of our collection approach. At its core, there are three main modules, namely, Litter Detection Module (LDM), Litter Estimation Module (LEM), and Litter Collection Module (LCM). All modules run continuously during robot motion.

During the deployment of the quadruped robot, the operator chooses a desired area to clean, and subsequently, the robot calculates an exploration path to fully cover the space. For this, we divide the chosen area into parallel lanes with a width that is relative to the field of view of the robot's frontal camera. Alternatively, the quadruped can be controlled via dedicated joystick commands by the operator. During locomotion, the LDM is responsible for the CBs detection via the onboard cameras, while the LCM generates a time-optimal collection path between the target objects. The same object can appear in more than one camera frame during locomotion, and in this case, multiple detections of the same item can happen. Therefore, a data association routine, together with a filtering procedure, is performed in the LEM to avoid unnecessary robot motions. Finally, once the quadruped reaches the desired CB, a visual-servoing routine that leverages a secondary camera pointing down in the direction of the robot's front feet is activated for precise step placement.

A block diagram of the proposed approach is shown in Figure 3.

3.1 | Litter detection module

The LDM is characterized by a CB detection neural network that is used for the first identification of the object to collect utilizing the images from the forward-facing camera. This information is then passed to the LCM for planning an optimal path and for performing a visual-servoing procedure for a precise footstep placement using the downward-facing camera. The latter is necessary for coping with inevitable state estimation drifts.

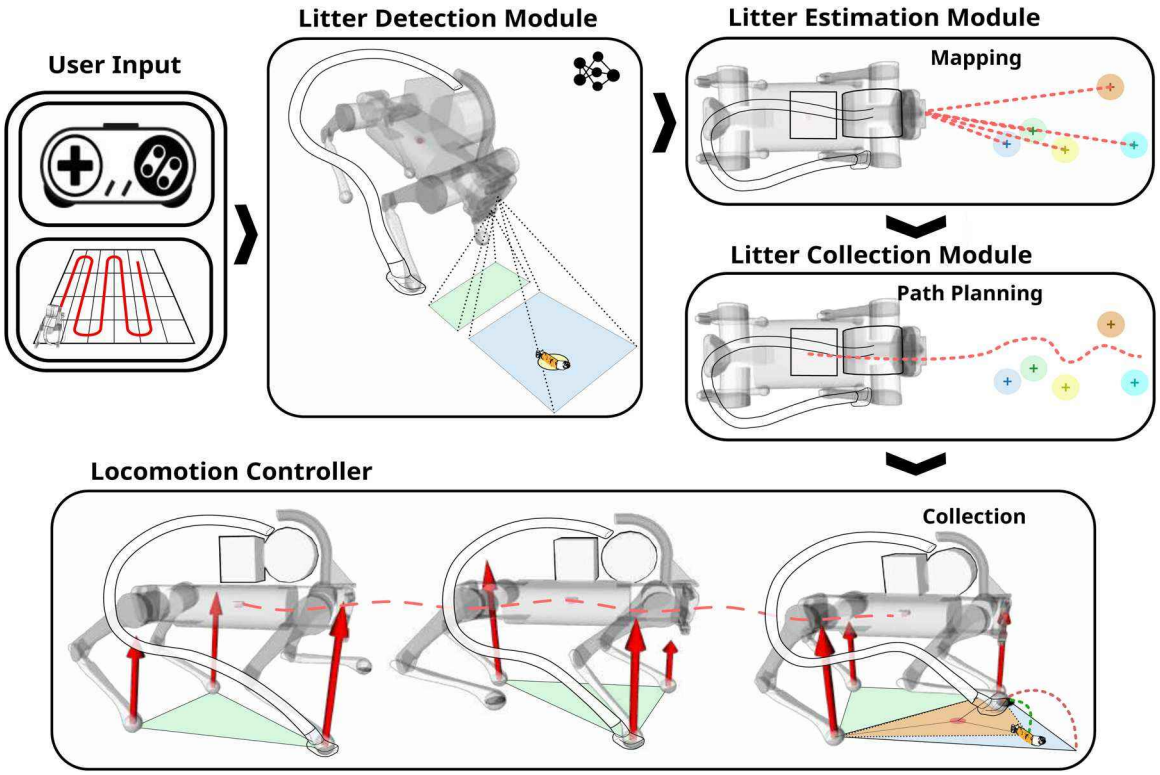

FIGURE 3 Block scheme of the proposed collection procedure. Starting from the left, first, an area to clean is selected by an operator via a user interface, or alternatively, the robot is directly commanded using a joystick. The robot starts to detect the CBs while moving and creates a map of the collectable objects, and an optimal path planner is computed after each detection. Finally, the locomotion controller drives the robot to the collection area while a visual-servoing procedure precisely commands the robot's footsteps on top of the CBs. In the locomotion controller block, the current support polygon is highlighted in green, while the Ground Reaction Forces are in red. In the last snapshot on the right, the future nominal support polygon is highlighted in blue, while the one defined by the new stepping location is in orange. Finally, the corresponding swings for the two stepping locations are shown, respectively, in red and green. CB, cigarette butt. [Color figure can be viewed at wileyonlinelibrary.com]

The network utilized in this work is the broadly adopted Yolo-V4-tiny (Bochkovskiy et al., 2020), a Convolutional Neural Network renowned for its computational speed and accuracy. We used the tiny version of the network, which has a reduced number of parameters, to ensure online computation on the onboard PC, an Nvidia Jetson TX2. Yolo builds a bounding box around an object in the image and returns the probability that it contains a CB. To avoid false positives, we consider a CB detection only if its probability is higher than 90%. The network was retrained on the target objects exploiting the combination of two different data sets available online (Immersive Limit, 2024; Proença & Simões, 2020). The number of images was increased by a further manual annotation of other 100 images and by different data-augmentation techniques, such as image rotation, crop, background, and luminance modification (Shorten & Khoshgoftaar, 2019) generating 2000 more images. The network achieves a final overall accuracy of 91% on the validation data.

Starting from the LDM, the outputs of the detection procedure are bounding boxes surrounding the CBs present in the video feed. The network is employed two times in our collection procedure. First, it is fed with 720p RGB images coming from an Intel RealSense D435 mounted on the front of the robot (forward-facing camera). The CBs' detections in this case are used for planning purposes. Second, it is queried by feeding the images from an Intel RealSense D405, mounted on the bottom of the robot's trunk (downward-facing camera). Both cameras configuration is shown in Figure 2. This second pipeline is used for the visual-servoing procedure to drive precisely the foot of the robot (see Section 3.3), therefore, considering the camera positioning, we opt for the D405 given its shorter-range detection capability. In fact, putting the camera above the trunk (Figure 2) could produce more occlusions in the detection when the CB is located directly below the robot, and given the normal walking height of the system (around 30 cm), a camera with a short-range detection capability is needed. On the other hand, the D435 is used for the first detection of the CBs from a distance. These cameras also provide a depth map, aligned with the color image, that gives information on the distance of each pixel. This data is used to calculate the relative position of the detected CB with respect to the camera frame as

$$ {}^{C}x_{CB} = \frac{x_{bb} - c_x}{F_x}, \quad {}^{C}y_{CB} = \frac{y_{bb} - c_y}{F_y}, \tag{1} $$

$$ {}^{C}z_{CB} = depth(x_{bb}, y_{bb}), \tag{2} $$

where ${}^{C}x_{CB}$, ${}^{C}y_{CB}$, and ${}^{C}z_{CB}$ are the coordinates of the selected pixel expressed in the camera frame $\mathcal{C}$ and x_{bb} and y_{bb} are the pixel coordinates of the center of the bounding box. Furthermore, F_x, F_y, c_x, and c_y are the intrinsic parameters of the camera, while $depth(\cdot,\cdot)$ is the depth map information of the desired coordinates. The litter position in the camera frame is then converted into the world frame to perform the planning procedure in the LCM.

Remark. The superscript at the left of a variable indicates in which frame it is expressed ($\mathcal{C}$ camera frame, $\mathcal{H}$ horizontal frame, and $\mathcal{B}$ base frame). If the superscript is omitted, the quantity is expressed in the world frame.

3.2 | Litter estimation module

During the robot motion, the detected CBs are used to generate a map of waypoints, used by the LCM to compute a time-optimal path for the collection process. To avoid unnecessary motions in the presence of duplicate detection, a correct data-association routine is needed. For this, we implemented a Right-Invariant Extended Kalman Filter (Right-IEKF) to reduce the amount of noise and errors due to the state estimation drift. As demonstrated in Hartley et al. (2019), this filter has shown promising performance thanks to the log-linear error dynamics.

Given the robot state and the CB positions in the world frame, the filter updates these values each time a new CB is detected. At a generic time instant t, we express the Right-IEKF state representation as

$$ \boldsymbol{X}_t = \begin{bmatrix} \boldsymbol{R}^{\theta}_{b,t} & \boldsymbol{v}_{c,t} & \boldsymbol{p}_{c,t} & \boldsymbol{p}_{CB_1,t} & \cdots & \boldsymbol{p}_{CB_N,t} \\ \boldsymbol{0}_{1,3} & 1 & 0 & 0 & \cdots & 0 \\ \boldsymbol{0}_{1,3} & 0 & 1 & 0 & \cdots & 0 \\ \boldsymbol{0}_{1,3} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}_{1,3} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{3} $$

where $\boldsymbol{X}_t \in SE_{N+2}(3)$ is an extension of the $SE(3)$ Lie Group (see, Hartley et al., 2019) for the case of N different objects saved in the map; $\boldsymbol{R}^{\theta}_{b}$, $\boldsymbol{v}_{c,t}$, $\boldsymbol{p}_{c,t}$ are respectively, the estimated yaw of the robot's base, the velocity, and the position of the robot's Center of Mass (CoM); finally, $\boldsymbol{p}_{CB_i,t}$ is the world position of the ith CB.

The described filter fuses the drifting state, coming from the onboard odometry-based state estimator, with the measured position of the collectable CB, used as a fixed landmark. The right invariant observation model for the ith CB measurement is written as

$$ \begin{aligned} \boldsymbol{h}_i &= \boldsymbol{R}_c \begin{bmatrix} {}^{C}x_{CB,i} & {}^{C}y_{CB,i} & {}^{C}z_{CB,i} \end{bmatrix}^T, \\ \boldsymbol{H}_t &= [\boldsymbol{h}_i \quad 0 \quad 1 \quad \boldsymbol{0}_{N-i,1} \quad -1 \quad \boldsymbol{0}_{i-1,1}]^T, \\ \boldsymbol{s} &= [\boldsymbol{0}_{3,1} \quad 0 \quad 1 \quad \boldsymbol{0}_{N-i,1} \quad -1 \quad \boldsymbol{0}_{i-1,1}]^T, \\ \boldsymbol{Y}_t &= \boldsymbol{X}_t^{-1}\boldsymbol{s}, \end{aligned} \tag{4} $$

where ${}^{C}x_{CB,t}$, ${}^{C}y_{CB,t}$, ${}^{C}z_{CB,t}$ are the measurement at time t from Equation (1), and $\boldsymbol{R}_c$ the rotation matrix from the camera frame to world frame. The IEKF update is then written as

$$ \begin{aligned} \boldsymbol{S} &= \boldsymbol{H}\boldsymbol{P}_t\boldsymbol{H}^T + \boldsymbol{N}_t, \\ \boldsymbol{K} &= \boldsymbol{P}_t\boldsymbol{H}^T\boldsymbol{S}^{-1}, \\ \boldsymbol{Z} &= \boldsymbol{X}_t\boldsymbol{Y}_t - \boldsymbol{s}, \\ \check{\boldsymbol{X}}_t &= exp(\boldsymbol{K}\Pi\boldsymbol{Z})\boldsymbol{X}_t, \\ \check{\boldsymbol{P}}_t &= (\mathbb{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}_t(\mathbb{I} - \boldsymbol{K}\boldsymbol{H})^T + \boldsymbol{K}\boldsymbol{N}_t\boldsymbol{K}^T, \end{aligned} \tag{5} $$

where $\boldsymbol{\Pi}$ and $\boldsymbol{H}$ are selection matrices, $\boldsymbol{K}$ is the Kalman gain and $\boldsymbol{N}_t$ is the measurement covariance matrix. $\tilde{\boldsymbol{X}}_t$ and $\tilde{\boldsymbol{P}}_t$ are respectively the updated state and covariance matrix and $exp(\cdot)$ is the exponential map of the state Lie Group.

To properly exploit the filter, data association between the mapped objects and the new incoming measurement at each iteration is crucial. We periodically verify if the same CBs have been added twice to the map due to outliers in the measurement.

For this, we decided to utilize a threshold on the Mahalanobis distance between each CB, paired with a minimum distance threshold for checking the possibility of unresolved duplicate detection. The selection of this threshold is crucial when dealing with densely distributed CBs. If the threshold is chosen too conservatively, it may lead to the inadvertent exclusion of CBs that are located close to each other. In our experimental results (see Section 5), we opt for a minimum threshold of 2 cm.

To avoid the filter becoming too computationally expensive, we erase all the previously collected objects from the filter state.

3.3 | Litter collection module

We mounted a custom-designed suction nozzle to the left-front leg of our robot prototype. Therefore, planning a path is needed to allow the quadruped to place its foot/feet close to the side of each detected CB. One of the main challenges of this approach is that we still need to guarantee the stability of the robot when changing the footstep location. It is important to recall that the robot needs to constantly actively balance on its legs, and the foot placement represents one of the variables for the feasibility of the planned motion. To account for this problem, we decided to decouple planning and safety in a hierarchical manner. We split the LCM into two submodules, one that uses the data from the LEM to plan a first path for the CoM of the robot, and a second module that directly modifies the footstep placement to ensure the precision and safety of the collection procedure.

For the planning problem, given that multiple CBs can be detected during motion, and hence multiple collection sequences can be performed, we apply the Dijkstra search algorithm (Dijkstra, 1959), which returns both the optimal collection order and the optimal poses that the robot should have when placing the nozzle on top of the CBs. The computational time will depend on the complexity of the scenario, that is, the number of CBs on the ground. Here we only consider the X–Y positions and yaw angle of the robot's base, leaving to the locomotion controller the optimization of the robot's roll and pitch base angles (see Section 4.1). Additionally, we simplify the problem by disregarding the robot's height. We assume that the CBs are located in regions accessible by the robot directly from any other position. In other words, we consider a scenario wherein the robot has a perfect traversability, that is, the CBs can be reached by our system starting from *any* starting position. In reality, this assumption could not hold, and can lead to unfeasibility of the collection order. We leave the lifting of this assumption for future works.

Starting from an initial robot configuration, expressed as x_c^0, y_c^0, and θ_c^0, defining respectively the X–Y positions and yaw orientation of the CoM of the robot, Dijkstra expands all the detected objects sequentially. For each of them, we compute the desired hip configuration that the robot should have to perform the collection. Defining for clarity

$$e_x = x_{CB} - x_c^0, \quad e_y = y_{CB} - y_c^0, \quad e_\theta = \theta_{CB} - \theta_c^0 \tag{6}$$

for a generic $\boldsymbol{p}_{CB,k}$, the optimal hip configuration of the robot can be derived in closed form by solving the following problem

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \frac{1}{2}\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y} \\ \text{s.t.} \quad & \boldsymbol{R}_{2\times 2}{}^{\mathcal{B}}\bar{\boldsymbol{p}}_{bh,i} + [x_{CB} \quad y_{CB}]^T = \boldsymbol{T}\boldsymbol{p}_{CB,k}, \\ & \zeta = \frac{(e_y)\cos(e_\theta) - (e_x)\sin(e_\theta)}{(e_y)\sin(e_\theta) + (e_x)\cos(e_\theta)}, \\ & \boldsymbol{y} = [e_x \quad e_y \quad e_\theta \quad \zeta], \end{aligned} \tag{7}$$

where $\bar{\boldsymbol{p}}_{bh,i}$ represents the X–Y components of the vector connecting the base to the hip of the ith leg used for collection, while $\boldsymbol{T} = [\mathbb{I}_2 \quad \boldsymbol{0}_{2,1}] \in \mathbb{R}^{3\times 2}$ extracts the X–Y coordinates of $\boldsymbol{p}_{CB,k}$. The matrix $\boldsymbol{Q}$ is a diagonal weight matrix defining the importance of each cost term. $\boldsymbol{R}_{2\times 2} \in \mathbb{R}^{2\times 2}$ maps the X–Y coordinates of the base frame into the world frame. We collect all the expanded CBs in an ordered priority list, from which we can expand the configuration with the minimum cost. This will be used as the new initial robot position for the next Dijkstra step.

Once the quadruped is close enough to one of the CBs, the robot moves to the optimal posture, and the second submodule computes the footstep necessary to perform the collection. This is done by modifying the nominal footholds via a continuous visual-servoing procedure, which utilizes the LDM fed by the images coming from the downward-facing camera. The position of the detected CB is then extrapolated by the output of Yolo, converted into the base frame of the robot, and used as the next touch-down position of the robot's foot. More details about the locomotion controller can be found in Section 4.1. Furthermore, this submodule performs a continuous safety check to guarantee the robot's stability. For this, we check if the robot's CoM position projection is still inside the support polygon when a nominal footstep is modified. If the safety check fails, the nominal foothold is restored, preventing the robot from falling.

Algorithm 1. Litter collection procedure.

```
while robot_moving do
  [CBs]← LDM(front_facing_camera)
  if [CBs] notempty then
    map ← LEM([CBs])
  end if
  if map notempty then
    collection_pose ← LCM(map)
  end if
  if collection_pose then
    desired_speed ← goTo(collection_pose)
  else
    desired_speed ← UserInput()
  end if
  LocomotionController(desired_speed) (Section 4)
  if collection_pose == robot_pose then
    CB ← LDM(down_facing_camera)
    if isSafe(CB) then
      doCollection()
    end if
  end if
end while
function LDM(camera_image)
  bbox ← Yolo(camera_image)
  measurement ← get3dMeasurement(bbox) (Equations 1 and 2)
  return measurement
end function
function LEM(measurement)
  checkDataAssosiation(measurement)
  map ← IEKF(measurement) (Equation 5)
  return map
end function
function LCM(map)
  collection_poses ← DijkstraSearch(map) (Equation 7)
  return collection_poses[0]
end function
```

Figure 4 shows some snapshots of the collection procedure performed by the real robotic platform. Furthermore, we summarize the litter collection procedure in Algorithm 1, where the **goTo()** function refers to a simple PD controller that drives the robot to the desired goal commanding a desired speed to the **LocomotionController()** (Section 4). On the other hand, the **UserInput**() function returns a desired speed read from a joystick, or from a speed profile computed to track a predefined path. Finally, the **isSafe()** function corresponds to the stability check described above, while **doCollection()** activates the vacuum cleaner and changes the target step location.

4 | LOCOMOTION CONTROLLER

Our locomotion control framework is composed of proprioceptive and exteroceptive elements. To obtain a robust locomotion controller, we focus on a robust proprioceptive and reactive layer able to deal with irregular surfaces and uncertainties from the exteroceptive feedback (visual information). To do so, we implement a structure based on our Reactive Controller Framework (RCF) (Barasuol et al., 2013), which is composed of *Motion Generation* and *Motion Control* blocks. Both blocks, illustrated in Figure 5, are explained in detail in the following sections.

4.1 | Motion generation

As depicted in Figure 5, the motion generation block computes the desired robot base and leg motions. The RCF's motion generation can provide various gait patterns, like, trotting and crawling gaits. For the sake of space, we concentrate only on the description of the crawl gait used for the litter collection.

The crawl gait represents a sequence for the motion of the legs that allows achieving static stability during locomotion. Static stability is achieved when the vertical projection of the robot's CoM position lies inside the convex hull (so-called support polygon) formed by the contact foot locations (McGhee & Frank, 1968). A robot that executes such a gait operates in slow velocities but very robustly, being able to have long stance periods on various postures and to deal with very irregular surfaces. Two crawling gait sequences are implemented: leg sequence left-front⇒ right-front⇒ right-hind⇒ left-hind when moving forward, and leg sequence left-front⇒ left-hind⇒ right-front⇒ right-hind when moving backwards. The different sequences are used to prevent support polygon shrinkage in one of the speed directions, and thus the reduction of the static stability margin and maximum velocity the robot can achieve. The desired touch-down location of each foot is computed in a robot-centric approach (Rathod et al., 2021), and depends on high-level trajectory references (high-level commands provided by the user or by the collection path planner) and also proprioceptive and exteroceptive information. All the leg motion planning is performed in the *Horizontal Frame* (located at the robot's base frame with Z coordinates aligned with the gravity vector and the X coordinate aligned with the robot's base longitudinal axis projected in the world X–Y plane) (Barasuol et al., 2013). The velocity references are provided without considering the terrain irregularities and are treated as the tentative body motion path for an effective litter collection. From the reference velocities, the nominal foot touch-down position and corresponding leg swing motion are computed. The timing for the desired leg stance and swing phases is provided by the *Gait Generator* (based on desired step frequency and duty factor). It is important to highlight that the periodicity of the crawl gait, in our controller, is stability-dependent. In other words, even with predefined step frequency and duty factor for each leg (de Santos et al., 2012), a leg lift-off event is only executed if (1) the robot is under a static stability condition defined by a given stability margin value, and (2) the robot is in full-stance condition (all legs in contact with the ground). These conditions confer extra robustness to the locomotion when compared to approaches that consider a strictly

FIGURE 4 Snapshots and illustrations detailing the collection procedure. From left to right: The system first detects the cigarette butt (CB) using the front-facing camera, and afterward, the planner defines a new optimal pose for collection (highlighted in the first snapshot by the coordinate frame centered on the hip of the robot's leg associated with the collection mechanism) deviating from the predefined path shown in red. Once the robot reaches the collection area, it utilizes the down-facing camera to refine the previous estimate of the CB's location. Finally, the robot proceeds to perform the collection by stepping on the target and activating the suction. [Color figure can be viewed at wileyonlinelibrary.com]

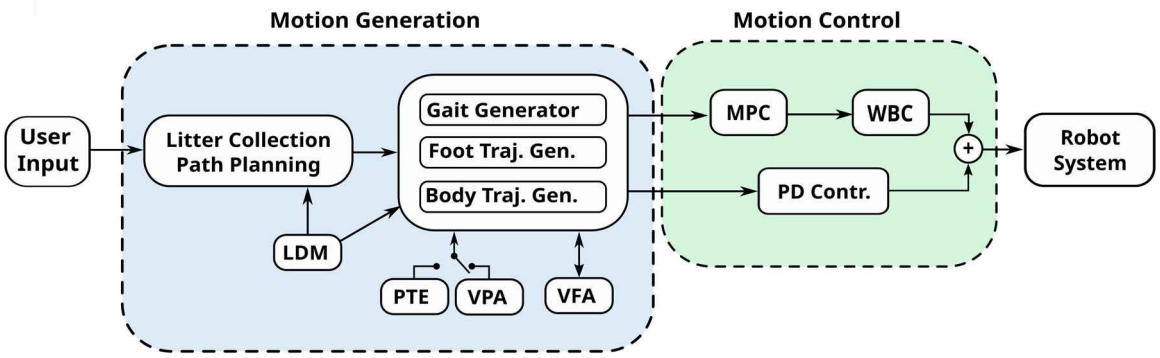

FIGURE 5 Block diagram illustrating the dataflow of the motion generation and control blocks of the locomotion framework. LDM, Litter Detection Module; MPC, Model-Predictive Controller; PD, proportional-derivative; PTE, Proprioceptive Terrain Estimator; VFA, Visual Foothold Adaptation; VPA, Visual Pose Adaptation; WBC, Whole-body Controller. [Color figure can be viewed at wileyonlinelibrary.com]

periodic crawling gait (e.g., in situations that may lead to surface collapses, contact losses, leg slippage, or inaccuracies in the terrain map or visual feedback). The desired foot references are provided by the *Foot Trajectory Generator* and are computed based on the actual foot lift-off position, the desired foot touch-down position, and the desired step clearance (the so-called step height). The nominal foothold position of a leg i is defined as $\boldsymbol{p}_{n,i} = [p_{nx,i}\, p_{ny,i}\, p_{nz,i}]^T$. The desired planar coordinates X–Y of the ith nominal foot touch-down position are computed as

$$\boldsymbol{p}_{n,xy} = \boldsymbol{p}_{0,xy} + {}^{\mathcal{H}}\dot{\boldsymbol{p}}_{c,xy}\frac{T_s}{2} + \left[{}^{\mathcal{H}}\boldsymbol{\omega} \times \boldsymbol{p}_{h,xy}\right]|_{x,y}, \tag{8}$$

where $\boldsymbol{p}_0$ is the foot home position, T_s is the desired stance period, ${}^{\mathcal{H}}\boldsymbol{\omega} = \left[00\,{}^{\mathcal{H}}\dot{\phi}_c\right]^T$ is the desired body angular velocity, and $\boldsymbol{p}_h$ is the hip position. The operator $|_{x,y}$ selects the planar coordinates from the vector product. We removed the dependency from leg i for the sake of clarity in the notation. The corresponding component $p_{n,z}$ depends on the Z component of the lift-off position p_{lo}, the pair $(p_{n,x}, p_{n,y})$, and the average terrain inclination estimated by the *Proprioceptive Terrain Estimator* (PTE). The PTE approximates the terrain surface with an average plane whose normal is described in the robot horizontal frame and inclination represented by a terrain roll angle ϕ_T and pitch angle θ_T. Thus, $p_{n,z}$ is computed as

$$p_{n,z} = p_{lo} + [\boldsymbol{R}_h(\phi_T, \theta_T)\boldsymbol{p}_n]|_z, \tag{9}$$

where $\boldsymbol{R}_h$ is the terrain rotation matrix, with zero yaw angle. The operator $|_z$ selects the scalar value corresponding to the Z coordinate.

During the litter collection, the nominal foothold can be modified by the visual feedback from the Visual Foothold Adaptation (VFA) (Villarreal Magaña et al., 2019), or by the LDM. The VFA modifies the nominal foothold to a safe location on the terrain surface if the nominal location brings a risk to the locomotion. The LDM, instead, modifies the nominal foothold for collection in case a litter is detected inside the local collection region underneath the corresponding hip, at the moment just before the leg lift-off. If a detected litter is located in an unsafe region classified by

the VFA, the modified foothold is not used for support and the collection is executed when the foot is still in the air. The CoM motion planning is also robot-centric and designed in the horizontal frame, whose target location is continuously computed according to the support polygon formed by the current location of the contact feet. The target can be tuned to prioritize the stability margin or the robot speed. The desired body height and orientations depend on the quality of the visual information. If the terrain map acquired around the robot (with exteroceptive sensors like depth cameras) is accurate, the desired body height and roll/pitch orientations are provided by our Visual Pose Adaptation (VPA) (Fahmi et al., 2023). In case of unreliable terrain mapping, a desired relative body height is set and the body orientation follows the one estimated by the PTE. This selection is represented by the switch between VPA and PTE illustrated in Figure 5. The target location inside the support polygon is tuned for requirements regarding maximum locomotion speed and robustness. The motion generation provides the motion control block with desired body CoM positions, trunk orientation, and desired joint trajectories. The desired joint trajectories are obtained by transforming the desired foot trajectories from the horizontal frame into the robot base frame and applying inverse kinematics (see, Barasuol et al., 2013, for further details). Section 4.2 describes the control actions to track the desired references from the motion generation block.

4.2 | Motion control

The robot's motion control block is composed of a Model-Predictive Controller (MPC) (Amatucci et al., 2022) in series with a Whole-body Controller (WBC) (Fahmi et al., 2019), and a joint proportional-derivative (PD) controller. The MPC controller computes ground reaction forces to track the desired CoM references from the motion generation block. It runs at 150 Hz and considers the Single-rigid Body Model (SRBM) (Orin et al., 2013) as a simplified robot model used for state predictions (Bledt & Kim, 2019; Grandia et al., 2019; Wu & Sreenath, 2015). The WBC runs at 250 Hz and computes the joint torques to track the equivalent CoM wrench (obtained from the MPC ground reactions forces), considering the influence of leg inertial effects and the physical constraints given by the robot and environment. Completing the control action, a joint PD controller contributes with joint torques to cope with model uncertainties and to better track the leg swing motion. The SRBM used for the MPC state predictions has shown to be representative even though the inertia of the robot legs is neglected. Using this model approximation, the system dynamics can be written as follows:

$$\ddot{\boldsymbol{p}}_c = \frac{1}{m}\sum^{n_{\text{leg}}} \boldsymbol{f}_i + \boldsymbol{g}, \tag{10}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}^{\mathcal{B}}\hat{\boldsymbol{\omega}}, \quad \boldsymbol{R} \in SO(3), \tag{11}$$

$$^{\mathcal{B}}\boldsymbol{I}^{\mathcal{B}}\dot{\boldsymbol{\omega}} = \boldsymbol{R}^T\left(\sum^{n_{\text{leg}}} \boldsymbol{r}_i \hat{\boldsymbol{f}}_i\right) - {}^{\mathcal{B}}\hat{\boldsymbol{\omega}}\,{}^{\mathcal{B}}\boldsymbol{I}^{\mathcal{B}}\boldsymbol{\omega}, \tag{12}$$

where $\ddot{\boldsymbol{p}}_c$ is the CoM acceleration, $\boldsymbol{f}_i$ is the ground reaction force (GRF) acting on the ith leg, $\boldsymbol{R}$ and $\dot{\boldsymbol{R}}$ are the rotation matrix representing the rotation between the world and the robot base and its derivative, while $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are the angular velocity and its derivative. The variable $\boldsymbol{r}_i$ is the vector connection the CoM and foot positions, the operator $\widehat{(\cdot)}$ maps the vector in a screw-symmetric matrix. Finally, $\boldsymbol{I}$ and m are the inertia matrix and mass of the robot body, and $\boldsymbol{g}$ is the gravity vector. Equation (12) is nonlinear in the angular part, therefore we use the variation-based linearization scheme presented in Ding et al. (2021). We express the rotational error in $SO(3)$, considering the variation to the operating point to be free from singularities in the representation.

We then perform a first-order Taylor expansion of the matrix exponential to then vectorize the error expressed in $SO(3)$ as $\boldsymbol{\xi} \in \mathbb{R}^3$ such that $\hat{\boldsymbol{\xi}} = \delta\boldsymbol{R}$. The linearized dynamics is finally discretized using the forward Euler scheme. The system state is defined as $\boldsymbol{x} = [\boldsymbol{p}_c, \dot{\boldsymbol{p}}_c, \boldsymbol{\xi}, {}^{\mathcal{B}}\boldsymbol{\omega}] \in \mathbb{R}^{12}$ and the control input as $\boldsymbol{u} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4] \in \mathbb{R}^{12}$. Given the simplified model, the system state, and control input, we can define the Optimal Control Problem (OCP) as

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{u}} \quad & \mathcal{L}_T(\boldsymbol{x}(N)) + \sum_{k=0}^{N-1} \mathcal{L}(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{D}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{c}_k, \\ & \boldsymbol{u}_k \in \mathrm{U}_k, \\ & k = 0, 1, \ldots, N-1, \\ & \boldsymbol{x}_{k=0} = \boldsymbol{x}_{op}, \end{aligned} \tag{13}$$

where $\mathcal{L}(\boldsymbol{x}(\cdot))$ is a convex quadratic cost over the user commanded velocities and body posture. $\boldsymbol{D}_k, \boldsymbol{B}_k, \boldsymbol{c}_k$ are the linearized dynamics and U_k is the set of feasible ground reaction forces constrained by the outer pyramid approximation of the friction cone to guarantee nonslipping conditions. $\boldsymbol{x}_{op}$ is the state variable at the operating point. To solve the OCP we used a specialized quadratic program (QP) solver (Pandala et al., 2019), that exploits the sparse structure of the problem. Finally, the GRFs obtained by solving the optimization problem (13) are then converted into the desired CoM wrench $\boldsymbol{w}_{u^*}$ sent to the WBC, as

$$\boldsymbol{w}_{u^*} = f_w(\boldsymbol{u}^*),$$

where $f_w(\cdot)$ is the function that maps the optimal ground reactions forces from the MPC controller into forces and moments around the robot CoM.

To realize $\boldsymbol{w}_{u^*}$, taking into account the robot full-dynamics, limitations, and environment constraints, the WBC solves a QP problem with a cost functional that penalizes the wrench tracking errors and the deviations from $\boldsymbol{u}^*$, that is,

$$\begin{aligned} \min_{\boldsymbol{\gamma}=\left[\ddot{\boldsymbol{q}}_b^T, \ddot{\boldsymbol{q}}_l^T, \boldsymbol{f}_g^T, \boldsymbol{\epsilon}\right]^T} \quad & \left\|\boldsymbol{M}_b\ddot{\boldsymbol{q}}_b - \boldsymbol{w}_{u^*}\right\|_{\mathrm{Qw}}^2 + \left\|\boldsymbol{f}_g - \boldsymbol{u}^*\right\|_{Qf}^2 \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{\gamma} = \boldsymbol{b}, \\ & \underline{\boldsymbol{d}} < \boldsymbol{C}\boldsymbol{\gamma} < \bar{\boldsymbol{d}}, \end{aligned} \tag{14}$$

where the decision variables are the base accelerations $\ddot{\boldsymbol{q}}_b$, the leg accelerations $\ddot{\boldsymbol{q}}_l$, the ground reaction forces $\boldsymbol{f}_g$, and the slack variables $\boldsymbol{\epsilon}$ for constraint relaxation. The equality constraints take into consideration dynamic consistency, leg inertial effects, joint kinematic and torque limitations, and friction constraints. All constraints

are stacked and described through the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ and boundaries $\underline{\boldsymbol{d}}$ and $\bar{\boldsymbol{d}}$. Finally, the values optimized by Equation (14) are mapped into joint torques using the whole-body dynamics equation. For further details on the description of the constraints and the mapping into joint torques, see Risiglione et al. (2022).

The readers can observe in the accompanying video the behavior of our locomotion controller in challenging scenarios, such as the one with gravel or the one with stairs.

5 | RESULTS

This section presents the results obtained with our prototype in both indoor and outdoor scenarios. The indoor scenario, represented by our laboratory, was chosen to validate the IEKF described in Section 3.2 in a controlled environment, where multiple CBs are placed on the floor. The test result is shown in Figure 6. Due to the drifts in the estimator, the raw CBs positions (green dots) vary significantly in the map with respect to the estimation coming from the filter (blue dots) during robot motion. It can be observed that without the filtering procedure, the data association process can fail without the possibility of recovery (red dots), meaning that the same object is added more than once to the list of the CBs to collect. In this case, the robot tries to collect the same object multiple times, drastically affecting the speed and collection time of the overall collection process.

Figure 7 shows six real-world scenarios in which we successfully tested our solution: *beach* (A), *urban* (B), *industrial* (C), *natural* (D), *nonflat* (E) and *park* (F). During our experiments, we repeatedly found CBs dispersed into these environments, showing that the chosen scenarios are representative of the tackled problem. We chose six different scenarios, instead of a single one, to test the unique challenges that each of them contains that can hinder a reliable CBs collection. Specifically, Scenario A is characterized by the presence of small pebbles that can get sucked in by the nozzle of the vacuum cleaner and deformable terrain, which represents a challenge for the locomotion controller of a quadruped. Scenario B is a common city street, where sidewalks and the presence of cars limit the traversability capabilities of many robotic systems. Similar limitations can be found in an industrial setting, such as the one in Scenario C, and in a small alley (Scenario E), where the presence of stairs limits the usage of wheeled robots. Scenario D represents a trail outside Genova, where the detection of CBs can be problematic given the presence of flora with similar sizes and colors. The same reasoning applies to Scenario F.

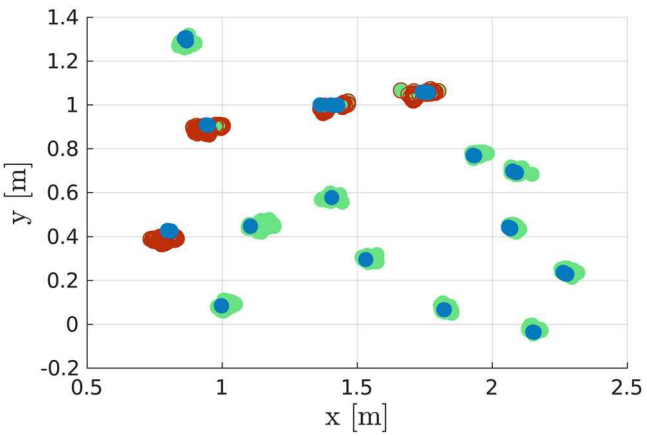

FIGURE 6 The estimated CB positions during a collection test in our laboratory, with (blue) and without (green) employing the IEKF. In red, we highlight the errors in data association due to unfiltered outliers, which will cause the robot to perform additional collection attempts. CB, cigarette butt; IEKF, Invariant Extended Kalman Filter. [Color figure can be viewed at wileyonlinelibrary.com]

We observed difficulties in the detection of the CBs in the scenario depicted in Figure 7d given the presence of similar objects, in color and shape, in the local flora. This issue has been mitigated by collecting additional images in that scenario and performing a successive fine-tuning of the detection model. This result shows that still, many misclassifications can still happen in a real environment, and additional data-synthesis techniques, such as using realistic simulators for data gathering, should be adopted to enhance the precision of the LCM.

In our final tests, our prototype successfully recognized and collected all the undisposed CBs. To give an intuition of the performance of the LCM, in Figure 8 we plot the trajectory of the left-front foot in the X–Y–Z plane. The task is to collect three different CBs. The circles represent the touch-down points, and the dashed lines the swing trajectories. The blue color indicates that the visual-servoing module did not change the desired foothold locations, hence the robot has stepped in the nominal location computed by Equation (8). The LCM is activated when the foot is close to a CB (red cross), causing the robot to perform a touch-down event in the vicinity of the CB (the red circle) instead of on the discarded nominal location (white circle with the blue border). In this experiment, the robot was controlled with a joystick and commanded with a user-defined nonconstant forward velocity. Since the step frequency remains fixed in our locomotion controller (Section 4), the nominal footholds are not distributed uniformly in the images. Here the reader can observe that, since the nominal footholds are coherent with the reference velocity, to suck the CBs the robot needed to step toward the left. In our pipeline the hip is positioned on top of the CB during the sucking, so after performing the collection, the robot keeps going straight forward from the reached position till the next CB.

The readers can refer to the accompanying video for the highlights of our results.

6 | DISCUSSION

Our prototype is endowed with a vacuum cleaner and a nozzle connected to the left-front leg. We want to point out that our design can easily be modified to connect the vacuum cleaner to all the available feet. In this way, the number of maneuvers

FIGURE 7 The prototype has been tested successfully in six different scenarios. Starting from the top left corner, we show the robot, controlled with a joystick for safety reason, traversing environments named *beach* (a), *urban* (b), *industrial* (c), *natural* (d), *nonflat* (e), and *park* (f). All these scenarios are characterized by unique challenges both from the locomotion point of view, such as stairs and slopes (scenarios a, d, e, and f), and from the detection point of view due to the presence of similar objects in size and color (e.g., small leaves in scenarios d and f). [Color figure can be viewed at wileyonlinelibrary.com]

required to collect the CBs can be reduced if a different foot can be exploited more easily from the current robot position. For this, only the Dijkstra algorithm in the LCM (Section 3.3) needs to be updated to consider multiple hip positions in the planning problem, hence computing the optimal robot's pose and selecting which leg to use to collect each CB. No changes are required in the visual servoing of the LCM.

To highlight this concept, we compare the performance in simulation of the same pipeline having the nozzle attached to one leg only (Figure 9 top) and on all four legs (Figure 9 bottom). In simulation, we defined an S-shaped path with 90-degree angles placing some CBs to be collected (the red disks in Figure 9). The lines' colors indicate the velocity deviation with respect to the "ideal" one obtained with the robot walking on the desired path without performing any collection. By comparing the top and the bottom plot, the readers can see the advantages of having multiple nozzles (higher velocity). This is explained by considering that if, for example, a CB is on the right side of the robot, the robot needs to perform a rotation to bring the left-front foot close to the CB. Even though we have not tested the vacuum cleaner with four nozzles, we have demonstrated that the software is general and can be adapted to a different configuration of the hardware prototype.

To deal with the problem of the sucking of the small pebbles in Scenario A, we experimentally adapted the distance of the nozzle with respect to the base of the foot. Given the usually lighter weight of the CBs, we were able to tune this parameter and obtain successful collections with only few pebbles sucked in by the machine. Nevertheless, we needed to empty the vacuum cleaner from time to time during the experiment. In future works, we plan to integrate a simple mechanical separation system tailored to deal with this specific problem into our prototype. We will also assess in more detail the effect of nozzle design and position from the foot base, taking also into consideration a variable sucking power. We plan in this way to enable a higher level of autonomy in all scenarios.

To evaluate the performance of the overall pipeline, we designed a test course in our laboratory, as shown in Figure 10, where the robot is commanded to explore repeatedly a predefined area, and a wooden obstacle is placed in the center to challenge the litter collection procedure. In this area, during each pass of the robot, we randomly spread a different number of CBs (4, 5, or 6) to evaluate the

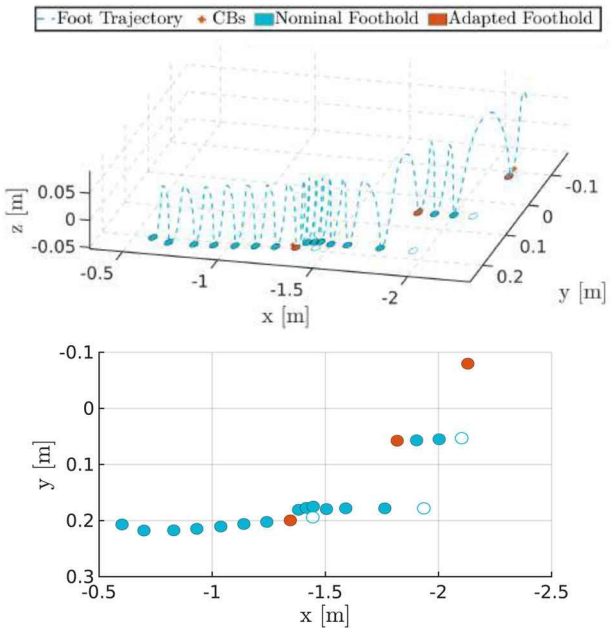

FIGURE 8 Trajectory of the left-front foot (where the nozzle of the vacuum cleaner is attached) during a collection procedure in Scenario A. On the top, the dot-lines represent the swing trajectory, while the circles represents the footholds. A circle is blue when the touch-down coincides with the nominal foothold, computed by Equation (8). The red circles represent the touch-down points corrected by the LCM. The corresponding nominal foothold is depicted by the white circles with blue borders. On the bottom, we plot the top-down view of the same foothold locations. CB, cigarette butt; LCM, Litter Collection Module. [Color figure can be viewed at wileyonlinelibrary.com]

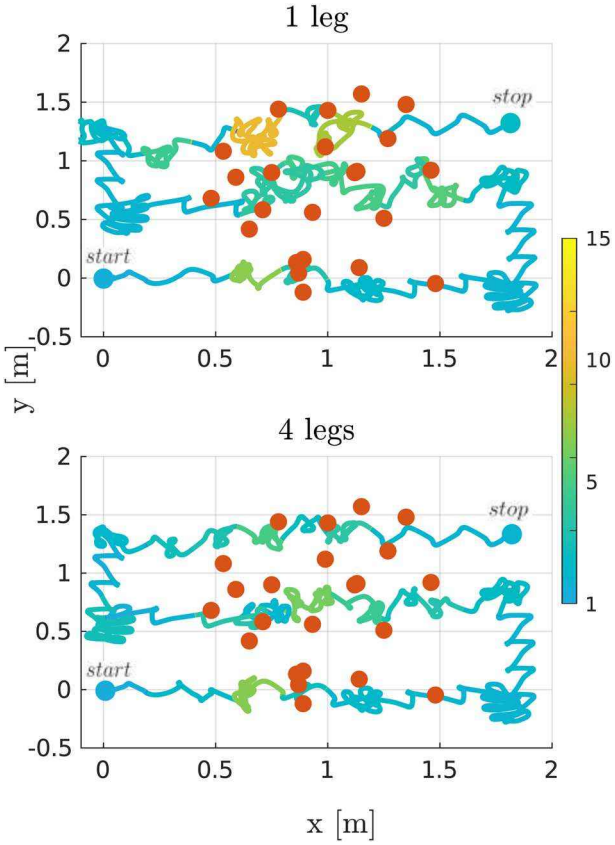

FIGURE 9 Comparison of the robot CoM position during a collection procedure when the nozzle is attached to only one leg (top) and when the vacuum cleaner is connected to all four legs (bottom). The red disks represent the CBs. The line color indicates the ratio between the constant velocity obtained by the robot simply following the path and the actual velocity during collection. In both cases, the robot shows a sway motion due to the crawl gait, but as expected, more maneuvers are needed when only one leg is used to collect the CBs. CB, cigarette butt; CoM, Center of Mass. [Color figure can be viewed at wileyonlinelibrary.com]

efficiency of the entire litter removal pipeline and its critical points. During 8 trials, the robot collected 29 out of 37 CBs. which corresponds to a success rate of 78.37% for the overall system. Further analysis of the failures (uncollected CBs) revealed that in 4 cases the CB was not detected: in two cases the CBs never entered the field of view of the robot's cameras during the entire lap, while in the other two cases, the network in the LDM did not identify the CBs with sufficient certainty. Considering both cases, the overall detection accuracy was 89% (33 detected out of 37). Considering only the CBs which have been seen (35) the accuracy of the network is 94.28%. The remaining 4 misses were caused by the LCM. In three instances, the robot's nozzle edge made contact with the CBs, rendering them impossible to be sucked. This failure indicates the importance of stepping precision. In the last cases, the CBs never entered the down-facing camera's field of view due to misalignment with the collection pose. Consequently, the robot was unable to perform the visual-servoing correction and relied solely on the mapped position of the CBs. However, due to some drift, the position in the map had degraded to the extent that proper collection performance was compromised, resulting in the robot missing the CBs. Nevertheless, our collection procedure, considering how many of the detected CBs have been collected, showed a success rate of 87.87% (29 out of a total of 33). A summary of this statistic can be found in Table 1 where we report the

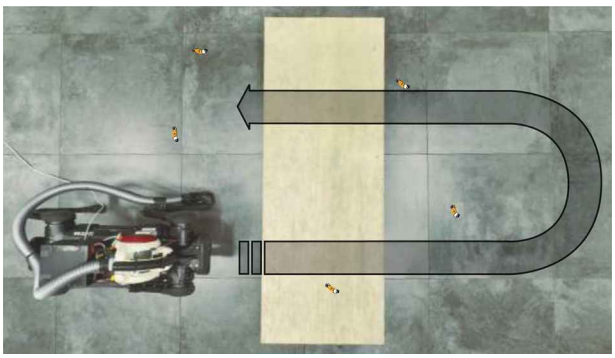

FIGURE 10 Configuration of the test course used to assess the framework under controlled conditions. The robot's predefined path is marked with a translucent gray arrow, and multiple cigarette butts (CBs) are randomly distributed on the ground. Additionally, a wooden obstacle 12 cm high is placed in the center of the scenario. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Statistical analysis.

Description	Success	Maximum	Rate
Overall pipeline: collected/total CBs	29	37	78.37
Detection: detected/total CBs	33	37	89.18
Neural Network: detected/seen CBs	33	35	94.28
Collection: collected/detected CBs	29	33	87.87

Abbreviation: CB, cigarette butt.

achieved results, while the execution of this experiment can be observed in the accompanying video of the paper.

7 | CONCLUSION

In this work, we have presented both the hardware and the software of a quadruped robot able to autonomously and efficiently collect small litter during locomotion. The presented approach takes advantage of a currently poorly exploited feature of legged robots: the ability and freedom to select footholds not only for a stable locomotion, but also for specific applications. To the best of the authors' knowledge, this is the first time that the legs of a legged robot are *concurrently* utilized for locomotion and for a different task. Differently from arms mounted onto mobile robots, which add weight, complexity, cost, and energy demand, exploiting legs for purpose-oriented interactions with the ground is an elegant and minimalist solution. On one hand, it is time-efficient because the feet need to touch the ground anyway for locomotion, so no additional whole-body arm control is needed that is generally slower when contact with the environment is required. On the other hand, it is potentially more precise since the kinematic chain of a leg is usually shorter compared to the one of a mobile base with an onboard arm.

This minimalist approach can be exploited for many other applications that are different from the presented use-case of litter removal, including for example:

- in agriculture and forestry: for the selective spraying of weed in crop fields (e.g., salads), punctual measurements of soil properties (e.g., humidity), or for the injection of seeds into the ground (e.g., reforestation);
- in infrastructure inspection: for surface sensing where contact with the surface is required (e.g., crack detection);
- construction: placement of nails and rivets (e.g., construction of large wooden structures or steel vessels).

In future works, we want to analyze the possible gain in power-efficiency that our prototype can enable compared to the one achievable by using an additional arm on top of the robot to perform the litter collection. Furthermore, we aim to build and test the necessary hardware for enabling our prototype to use all the available feet for litter collection. Besides increased collection speed, equipping multiple feet with nozzles can easily be used for separate collection of different types of waste. In fact, our detection module (Section 3.1) can be easily generalized for multiple objects, and each foot can take care of a specific type of waste.

ACKNOWLEDGMENTS

The research was partially funded by the European Union—NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project "RAISE—Robotics and AI for Socio-economic Empowerment" (ECS00000035). We would like to thank Matteo Villa, Chundri Boelens, and Salvatore Casella for their help on the hardware modifications; Marco Marchitto and Gianluca Cerilli for their support on the software.

ORCID

Lorenzo Amatucci http://orcid.org/0009-0006-6764-8102
Giulio Turrisi https://orcid.org/0000-0003-3007-3553
Angelo Bratta http://orcid.org/0000-0002-1306-9344
Victor Barasuol https://orcid.org/0000-0003-4966-5743
Claudio Semini https://orcid.org/0000-0002-3034-4686

REFERENCES

4Ocean. (2024) *Bebot*. https://www.4ocean.com/pages/4ocean-x-poralu-bebot [Accessed 20/FEB/2024].

Amatucci, L., Kim, J.-H., Hwangbo, J. & Park, H.-W. (2022) Monte Carlo tree search gait planner for non-gaited legged system control. In: *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4701–4707.

Angsa Robotics. (2024) *Angsa*. https://angsa-robotics.com/en-de/ [Accessed 20/FEB/2024].

Araújo, M.C.B. & Costa, M.F. (2019) A critical review of the issue of cigarette butt pollution in coastal environments. *Environmental Research*, 172, 137–149.

Bai, J., Lian, S., Liu, Z., Wang, K. & Liu, D. (2018) Deep learning based robot for automatically picking up garbage on the grass. *IEEE Transactions on Consumer Electronics*, 64(3), 382–389.

Barasuol, V., Buchli, J., Semini, C., Frigerio, M., De Pieri, E.R. & Caldwell, D.G. (2013) A reactive controller framework for quadrupedal locomotion on challenging terrain. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2554–2561.

Belzagui, F., Buscio, V., Gutiérrez-Bouzán, C. & Vilaseca, M. (2020) Cigarette butts as a microfiber source with a microplastic level of concern. *The Science of the Total Environment*, 762, 144165.

Bhuyan, M.S. (2022) Effects of microplastics on fish and in human health. *Frontiers in Environmental Science* 10, 10.

Bledt, G. & Kim, S. (2019) Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6316–6323.

Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y.M. (2020) Yolov4: optimal speed and accuracy of object detection. arXiv:2004.10934.

de Santos, P., Garcia, E. & Estremera, J. (2012) *Quadrupedal locomotion: An introduction to the control of four-legged robots*. London: Springer.

Dijkstra, E.W. (1959) A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269–271.

Ding, Y., Pandala, A., Li, C., Shin, Y.-H. & Park, H.-W. (2021) Representation-free model predictive control for dynamic motions in quadrupeds. *IEEE Transactions on Robotics*, 37(4), 1154–1171.

Fahmi, S., Barasuol, V., Esteban, D., Villarreal, O. & Semini, C. (2023) Vital: vision-based terrain-aware locomotion for legged robots. *IEEE Transactions Robotics*, 39(2), 885–904.

Fahmi, S., Mastalli, C., Focchi, M. & Semini, C. (2019) Passive whole-body control for quadruped robots: experimental validation over challenging terrain. *IEEE Robotics and Automation Letter*, 4(3), 2553–2560.

Grandia, R., Farshidian, F., Dosovitskiy, A., Ranftl, R. & Hutter, M. (2019) Frequency-aware model predictive control. *IEEE Robotics and Automation Letters*, 4(2), 1517–1524.

Hartley, R., Jadidi, M.G., Eustice, R.M. & Grizzle, J.W. (2019) Contact-aided invariant extended Kalman filtering for robot state estimation. *The International Journal of Robotics Research*, 39, 402–430.

Immersive Limit. (2024) *Immersive limit*. https://www.immersivelimit.com/blog/training-an-ai-to-recognize-cigarette-butts [Accessed 20/FEB/2024].

Jenelten, F., Grandia, R., Farshidian, F. & Hutter, M. (2022) Tamols: terrain-aware motion optimization for legged systems. *IEEE Transactions on Robotics*, 38(6), 3395–3413.

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V. & Hutter, M. (2020) Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47).

Liu, J., Balatti, P., Ellis, K., Hadjivelichkov, D., Stoyanov, D., Ajoudani, A. & Kanoulas, D. (2021) Garbage collection and sorting with a mobile manipulator using deep learning and whole-body control. In: *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pp. 408–414.

McGhee, R. & Frank, A. (1968) On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 3, 331–351.

Micevska, T., Warne, M.S.J., Pablo, F. & Patra, R. (2006) Variation in, and causes of, toxicity of cigarette butts to a cladoceran and microtox. *Archives of Environmental Contamination and Toxicology*, 50, 205–212.

Novotny, T.E., Lum, K., Smith, E.A., Wang, V. & Barnes, R.L. (2009) Cigarettes butts and the case for an environmental policy on hazardous cigarette waste. *International Journal of Environmental Research and Public Health*, 6, 1691–1705.

Ocean Conservancy. (2023) *2023 Cleanup report*. Ocean Conservancy. Technical Report. https://oceanconservancy.org/trash-free-seas/international-coastal-cleanup/annual-data-release/

Orin, D.E., Goswami, A. & Lee, S.-H. (2013) Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35, 161–176.

Pandala, A.G., Ding, Y. & Park, H.-W. (2019) qpSWIFT: a real-time sparse quadratic program solver for robotic applications. *IEEE Robotics and Automation Letters*, 4(4), 3355–3362.

Pixies Urban Lab. (2024) *Pixies*. https://www.pixiesurbanlab.com/ [Accessed 20/FEB/2024].

Proença, P.F. & Simões, P. (2020) Taco: trash annotations in context for litter detection. arXiv:2003.06975.

Project Beach Bot. (2024) *Beachbot*. https://project.bb/en [Accessed 20/FEB/2024].

Rangel-Buitrago, N., Williams, A.T., Neal, W.J., Gracia, C.A. & Micallef, A. (2022) Litter in coastal and marine environments. *Marine Pollution Bulletin*, 177, 113546.

Rathod, N., Bratta, A., Focchi, M., Zanon, M., Villarreal, O., Semini, C. & Bemporad, A. (2021) Model predictive control with environment adaptation for legged locomotion. *IEEE Access*, 9, 145710–145727.

Risiglione, M., Barasuol, V., Caldwell, D.G. & Semini, C. (2022) A whole-body controller based on a simplified template for rendering impedances in quadruped manipulators. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 9620–9627).

Shorten, C. & Khoshgoftaar, T.M. (2019) A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 1–48.

Slaughter, E., Gersberg, R.M., Watanabe, K., Rudolph, J., Stransky, C. & Novotny, T.E. (2010) Toxicity of cigarette butts and their chemical components to the marine and freshwater fishes. *Tobacco Control*, 20 Suppl 1(Suppl_1), i25–i29. https://doi.org/10.1136/tc.2010.040170

Song, Q., Li, J. & Zeng, X. (2015) Minimizing the increasing solid waste through zero waste strategy. *Journal of Cleaner Production*, 104, 199–210.

UNEP. (2006) *2006 Annual report*. United Nations Environment Programme. Technical Report.

Villarreal Magaña, O.A., Barasuol, V., Camurri, M., Franceschi, L., Focchi, M., Pontil, M., Caldwell, D.G. & Semini, C. (2019) Fast and continuous foothold adaptation for dynamic locomotion through CNNs. *IEEE Robotics and Automation Letters*, 4(2), 2140–2147.

World Health Organization. (2022) Tobacco: poisoning our planet. Environmental Research.

Wu, G. & Sreenath, K. (2015) Variation-based linearization of nonlinear systems evolving on $so(3)$ and $\mathbb{S}^2$. *IEEE Access*, 3, 1592–1604.

Zapata-Impata, B.S., Shah, V., Singh, H. & Platt, R.W. (2018) Autotrans: an autonomous open world transportation system. arXiv:1810.03400.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Amatucci, L., Turrisi, G., Bratta, A., Barasuol, V. & Semini, C. (2024) VERO: A vacuum-cleaner-equipped quadruped robot for efficient litter removal. *Journal of Field Robotics*, 1–14. https://doi.org/10.1002/rob.22350