2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Environment-Adaptive Gait Planning for Obstacle Avoidance in Lower-Limb Robotic Exoskeletons

Edoardo Trombin[1], Stefano Tortora[1,2]*, Emanuele Menegatti[1,2] and Luca Tonin[1,2]

***Abstract*— Powered lower limb exoskeletons (LLEs) have emerged as wearable robots designed to augment users' locomotion capabilities, offering mechanical support and additional power for both healthy and impaired subjects. However, current assistive exoskeletons are limited by predefined motion trajectories, hindering adaptability to unstructured environments encountered in daily life. To address this limitation, this paper proposes an environment-adaptive gait planning (EAGP) solution. The approach integrates scene understanding, pose estimation, and adaptive gait planning modules. A novel Collision-Free Foot Trajectory Generator (CFFTG) algorithm facilitates obstacle avoidance by computing collision-free foot trajectories, enhancing safety and adaptability. Through inverse kinematics, the planned trajectories are converted into angular joint trajectories for execution by low-level control. This comprehensive framework aims to enhance the adaptability and safety of LLEs, paving the way for broader real-world applications beyond clinical and research settings.**

I. Introduction

A powered lower limb exoskeleton (LLE) is a type of wearable robot that can be attached to the user's legs, either partially or completely, depending on the design and the application. These devices can help the user performing different locomotion tasks by providing both mechanical support to the user's joints and muscles, and additional power to assist the movement. Although LLEs are showing recent emerging applications for military and industry [1], robotic exoskeletons are being mainly deployed for full or partial mobilization of people with severe gait impairments (e.g., stroke survivors, spinal cord injury patients) or with limited motor capabilities (e.g., elderly) [2]; exoskeletons designed with this purpose are called assistive exoskeletons. Despite the great advancements in terms of control, wearability and usability, the motion of these assistive exoskeletons is still based on predefined trajectories, generally derived from the biomechanics of human locomotion, and triggered by the user's moving intention [3]. As a consequence, most of assistive LLEs are restricted to clinical and research settings, since pre-defined trajectories are not suitable to move through unstructured environments, which are usually encountered in daily life. In order to overcome this lack of adaptability, adaptive gait planning (AGP) solutions have been devised in order to generate gait trajectories with different step length, height and walking speed [4], [5]. Even if such solutions are able to achieve variable gait patterns, the explicit adaptation of the gait trajectory to the specific environment is little explored in the literature [6]–[8]; in this work, we will refer to this capability as Environment Adaptive Gait Planning (EAGP). Moreover, automatic transitions between different locomotion patterns based on environmental perception are still challenging, although recent research has been able to obtain promising results [9]. In this paper, we propose an EAGP solution for low-obstacle avoidance in robotic LLEs: a scene understanding module is employed to detect the ground plane and the surrounding obstacles, along with possible positions for the next step's foothold. Afterwards, a pose estimation module is in charge of computing the pose of the exoskeleton in the sagittal plane with respect to the perceived environment. Subsequently, an AGP module will compute foot trajectory and corresponding angular trajectories of the exoskeleton joints which will allow a collision-free step.

This work is funded by Next Generation EU, in the context of the National Recovery and Resilience Plan, Investment PE8 – Project Age-It: "Ageing Well in an Ageing Society" [DM 1557 11.10.2022].
[1]Department of Information Engineering, University of Padua, Padua, Italy
[2]Padova Neuroscience Center, University of Padua, Padua, Italy
*Corresponding author: stefano.tortora@unipd.it

II. Related Work

As mentioned in the previous section, EAGP solutions are just recently emerging; these solutions mainly focus on stair ascent / descent [8], [10] and obstacle avoidance [6], [11]. The most common way of planning adaptive gait trajectories is by employing polynomial trajectories and fixing specific conditions (e.g., position, velocity, acceleration) at predetermined way points to constrain the motion. In [11] the initial and final positions of the step that will surpass the obstacle are planned in advance so that the obstacle-crossing step has its vertical peak above the centroid of the obstacle. However, this solution does not take into account the overall shape of the obstacle; moreover, it requires to plan in advance the whole sequence of steps for approaching and passing the obstacle, lacking the ability to adapt the trajectory of the single step. On the other hand, techniques related to control theory such as Model Predictive Control (MPC) have shown promising results, since they can deal with many different constraints at the same time by performing global optimization [6]. Even if this approach has shown promising results, its computational requirements are generally high; this aspect may be critical when employing such solution with real-time constraints. The solution proposed in this paper aims to overcome these limitations: the shape of the obstacle is taken into account while generating the foot trajectory, while also respecting real-time constraints, thus allowing the solution to potentially work while performing continuous walking.

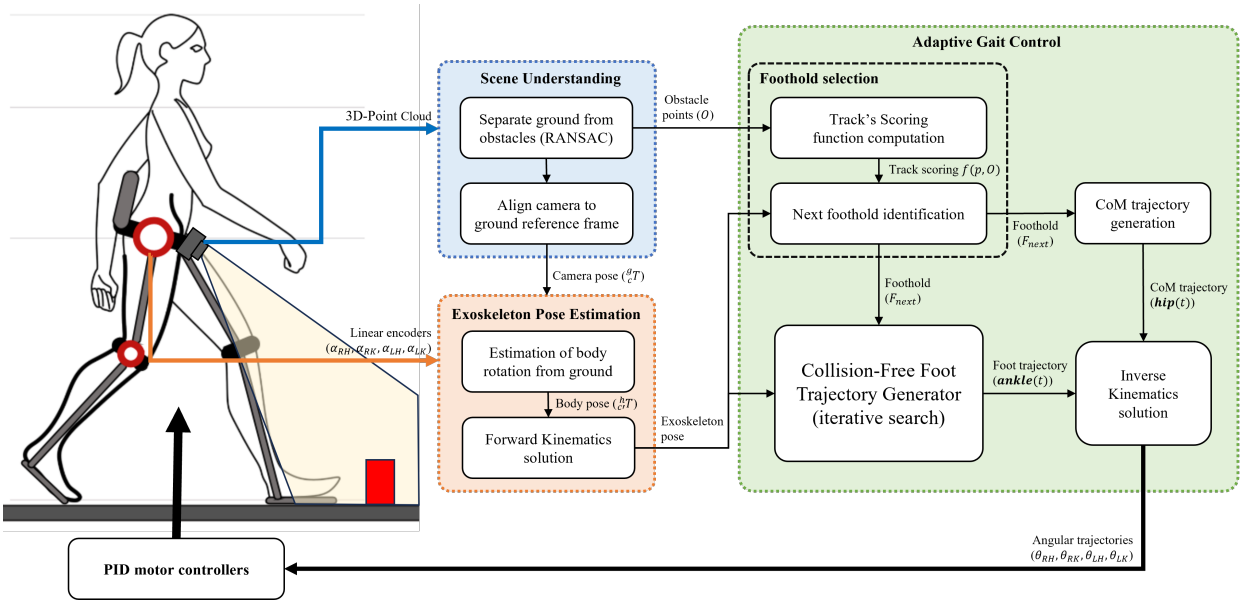

Fig. 1: Overview of the proposed solution. The figure shows the flow of the information between different modules composing the overall system: the Scene Understanding receives and processes the 3D point cloud coming from the camera to recognize ground and obstacles; the Exoskeleton Pose Estimation computes the exoskeleton configuration and locates the exoskeleton in the environment; the Adaptive Gait Planning is in charge of identifying the best foothold position and computing a collision-free gait trajectory for actuating the exoskeleton joints.

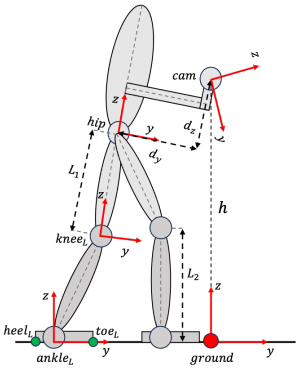

Fig. 2: Exoskeleton kinematic chain with respect to camera and ground (only left leg joints frames are highlighted for sake of clarity).

III. Methods

A schematic overview of the proposed solution is displayed in Fig. 1. Overall, the architecture is organized in three main processing modules: (i) Scene Understanding; (ii) Exoskeleton Pose Estimation; (iii) Adaptive Gait Planning. During the double-support phase, the environment in front of the exoskeleton walking path is recorded from a RGB-D camera fixed in the front middle of the exoskeleton, at the level of the pelvis. The Scene Understanding module receives as input the point cloud recorded by the camera and process it to extract obstacle points, ground points and camera pose with respect to the environment. Then, the Exoskeleton Pose Estimation module reconstructs the pose of the LLE in the sagittal plane and its distance with respect to the obstacles perceived in the previous module. The core of the proposed approach is represented by the Adaptive Gait Planning module: first, the best foothold for the next step is selected considering the distance to the obstacle and safety constraints. Then, a Gaussian-based iterative searching algorithm is proposed to identify the gait parameters of a feasible swing foot trajectory avoiding the detected obstacles and respecting the kinematic constraints of the human locomotion. The outcome of the proposed framework are the angular trajectories of the exoskeleton joints which are sent to the low-level motor controllers (e.g., PID controllers) for the movement execution. On movement completion, the swing leg is changed and the process is repeated for the next step.

A. Scene Understanding

The Scene Understanding module is responsible of extracting different features from the walking environment recorded by the RGB-D camera. After an initial downsampling of the point cloud, the random sample consensus (RANSAC) algorithm is applied to identify the ground points, under the assumption of indoor environment with a uniform level-ground floor. Next, the normal vector with respect to the detected ground plane $n_P = \{n_{Px}, n_{Py}, n_{Pz}\}$, returned by the RANSAC algorithm, is employed to align the point cloud with the z-axis of the ground reference frame $n_G = \{0, 0, 1\}$ (reported in Fig. 2 as the z-axis of the ground frame). To do so, after normalizing the vector n_P, the cross-product vector $v = \{v_x, v_y, v_z\} = n_P \times n_G$ (with $s = ||v||$) and the dot product $k = n_P \cdot n_G = n_{Pz}$ are computed. Subsequently, the skew-symmetric cross-product matrix of v is computed

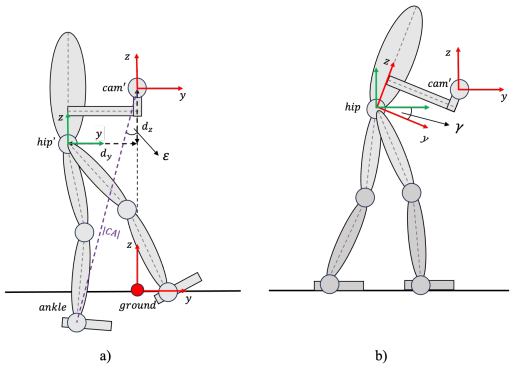

Fig. 3: a) Exoskeleton pose estimation assuming $\gamma = 0°$. b) Exoskeleton pose estimation after having calculated angle γ.

as:

$$M = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{1}$$

Finally, the rotation matrix ${}^{g}_{c}R$ from the camera reference frame to the ground reference frame can be computed as:

$${}^{g}_{c}R = I + M + \frac{1}{1+k}M^2 \tag{2}$$

where I is the 3×3 identity matrix. At the end of this process, the input point cloud will be aligned to the ground reference frame, but translated on the z-axis by a value equal to the height of the camera from the ground h. Thus, the homogeneous transformation matrix between the frames cam and $ground$ is finally defined as:

$${}^{g}_{c}T = \begin{bmatrix} {}^{g}_{c}R & {}^{g}_{c}t \\ \overrightarrow{0} & 1 \end{bmatrix} \tag{3}$$

where ${}^{g}_{c}t = \begin{bmatrix} 0 & 0 & h \end{bmatrix}$ is the translation vector along the vertical direction, and $\overrightarrow{0} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$.

B. Pose Estimation

In order to plan the next step, it is first required to locate the exoskeleton in the environment, in particular with respect to the obstacles eventually detected by the camera. To do so, the following assumptions are considered: (i) the pose estimation is performed only during the double-support phase, with both feet on the ground, and only considering the sagittal plane; (ii) the exoskeleton is composed by a rigid frame and rotations are allowed only at the lower limb joints (hip, knee, ankle); (iii) joint's angular positions are measured by linear encoders only for the active exoskeleton joints (hip, knee), while the rotation of eventual passive joints (e.g., ankle) are unknown. These assumptions have been introduced as they characterize most common situations in both commercial and research exoskeletons [12]. To compute the pose of the exoskeleton, let us first consider the transformation between cam' frame (obtained by applying the rotation ${}^{g}_{c}R$ to cam frame) and hip frame:

$${}^{h}_{c'}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma & -d_y \\ 0 & sin\gamma & cos\gamma & -d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where γ is the rotation of the hip frame around the x-axis of cam', d_y and d_z the horizontal and vertical offset values of the rigid frame connecting the camera to exoskeleton, as depicted in Fig. 2. If all these values are known, together with the length of the exoskeleton links L_1 and L_2, the whole exoskeleton configuration, for both the right and left leg can be fully reconstructed through forward kinematics in the ground reference frame:

$$knee_y = hip_y + L_1 \sin(\alpha_H + \gamma) \tag{5}$$

$$knee_z = hip_z - L_1 \cos(\alpha_H + \gamma) \tag{6}$$

$$ankle_y = knee_y + L_2 \sin(\alpha_H + \alpha_K + \gamma) \tag{7}$$

$$ankle_z = knee_z - L_2 \cos(\alpha_H + \alpha_K + \gamma) \tag{8}$$

where $\{hip_y, hip_z\}$ is the position of the hip from the ground (easily obtainable through the transformation matrix ${}^{h}_{c'}T$ and then adding an offset equal to the camera height from the ground h), α_H and α_K are the angular positions of the hip and knee joints measured by the encoders. However, while the values of d_y and d_z are generally easy to measure, e.g. from the exoskeleton CAD model, the orientation γ of the camera is assumed not to be easily measurable. Moreover, the orientation of the body with respect to cam' (and thus to the ground) depends also on the ankle joints rotation which is unknown. To overcome this problem, a geometry-based procedure is proposed to recover the angle γ from the camera input in order to estimate the full pose of the exoskeleton in the sagittal plane. First, a ficticious frame hip' is constructed assuming $\gamma = 0$, and the exoskeleton configuration is calculated through Eqs. (4)-(8) . If the real value of γ differs from 0, the ankles' position will result misaligned with respect to the ground frame along the z-axis ($ankle_z \neq 0$ for one or both ankles, as shown in Fig. 3a). Nevertheless, knowing the exoskeleton configuration and the position of the camera with respect to the ground, it can be demonstrated that the following equality (expressed in ground coordinates) is always true:

$$ankle_z = cam_z - |CA| \cos(\gamma + \varepsilon) \tag{9}$$

where $|CA|$ is the length of the segment connecting the camera frame and the ankle frame, and ε is the angle between CA and the z-axis of the cam' frame, calculated as:

$$\varepsilon = \arcsin(\frac{ankle_y}{|CA|}) \tag{10}$$

At this point it is possible to compute the angle γ by setting the condition $ankle_z = 0$ for one of the two ankles (in this case the swing foot ankle is chosen) yielding:

$$\gamma = \arccos(\frac{cam_z}{|CA|}) + \varepsilon \tag{11}$$

that is defined if $cam_z \leq |CA|$, which is always true in our kinematic model. After having estimated γ (see Fig. 3b), the

whole exoskeleton configuration can be re-computed again via forward kinematics solution.

C. Adaptive Gait Planning

Once the point cloud of the current scene is processed and the exoskeleton is located in the environment, the Adaptive Gait Planning module computes the next step through a two-step approach by firstly identifying the most appropriate foothold position, and then by computing the collision-free foot trajectory which is finally converted into joints' angular trajectories for the low-level motor controllers of the exoskeleton.

1) Foothold Selection: The processed point cloud aligned with the exoskeleton frame is used to identify the next foothold in the ground frame. To do so, a 2D-track in front of the swing foot is considered: the track is centered in the x-axis at the exoskeleton ankle, with a width equal to the width of the foot and a length in the y-axis equal to the maximum step length y_{max}. A score $f(p, O) = f_G(p) f_O(p, O)$ is given to each point p of the track T as the product of two factors; the first component $f_G(p)$ is equal to a Gaussian probability density function centered around the average step length and a variance equal to the physiological gait variability [13]. In this way, the scoring system will privilege footholds with medium step length y_{avg}. The second component $f_O(p, O)$ is an occupancy grid function that considers the presence of an obstacle O on the track and is computed as follows:

$$f_O(p,O) = \min_{p_o \in O} \begin{cases} 0, & ||p - p_o|| \leq C_1 \\ \frac{||p-p_o|| - C_1}{C_2 - C_1}, & C_1 < ||p - p_o|| < C_2 \\ 1 & C_2 \leq ||p - p_o|| \end{cases} \tag{12}$$

with $C_1 = p_{oz} + \Delta_1$ and $C_2 = C_1 + \Delta_2$,where p_{oz} is the height of the obstacle point from the ground, Δ_1 and Δ_2 are two safety parameters defined by the operator. This scoring function is linear when the distance between the track point p and the obstacle point p_o is between two conditions C_1 and C_2, and clipped to zero or one when the distance is below C_1 or above C_2, respectively. This piece-wise function is computed for each obstacle point p_o, and the minimum value is the one returned by $f_O(p, O)$. The optimal foothold is computed by employing a 2D-moving average window w over the track T as follow:

$$F_{next} = \max_{w \in T} S_w \tag{13}$$

$$S_w = \begin{cases} \frac{1}{N_w} \sum_w f(p_w, O), & f(p_w, O) \neq 0 \; \forall \; p_w \in w \\ 0, & otherwise \end{cases} \tag{14}$$

Each window has the dimension of the foot and it is shifted along the y-axis with a resolution of 1 cm. The best foothold F_{next} is selected as the window w with the highest average score S_w.

2) Center of Mass trajectory calculation: After having identified the foothold position, the Adaptive Gait Planning module is in charge of generating the cartesian trajectories in the sagittal plane which allows the LLE to execute the step without colliding with the detected obstacles, if any. To do so, the Center of Mass (CoM) trajectory is calculated first. Note that all of the calculations below are performed in the swing ankle reference frame (displayed in Fig. 2) for convenience. Assuming a rigid connection between the user's body and the exoskeleton pelvis frame and no passive motion of the pelvis other than the hip flexion/extension, the CoM is located on the sagittal plane in the middle of the segment connecting the center of rotation of the right and left hip joints. Thus, this computation coincides with calculating the trajectory of the hip reference frame in the sagittal plane $\mathbf{hip}(t) = \{y_{hip}(t), z_{hip}(t)\}$. The motion is modeled as a cubic polynomial in both time and space:

$$y_{hip}(t) = a_1 t^3 + b_1 t^2 + c_1 t + d_1 \tag{15}$$

$$z_{hip}(t) = a_2 y_{hip}(t)^3 + b_2 y_{hip}(t)^2 + c_2 y_{hip}(t) + d_2 \tag{16}$$

To compute the coefficients of such equations, positions and velocities of at least three way points are required; in the case of $y_{hip}(t)$, the initial and final positions and velocities are:

$$y_{hip}(0) = hip_y \qquad \dot{y}(0) = 0 \tag{17}$$

$$y_{hip}(t_m) = F_{pivot} \qquad \dot{y}_{hip}(t_m) = v_h \tag{18}$$

$$y_{hip}(t_f) = F_{pivot} + \frac{F_{next} - F_{pivot}}{2} \qquad \dot{y}(t_f) = 0 \tag{19}$$

where F_{pivot} is the y-position of the support foot, t_m is the time at which the hip and support foot are aligned in the sagittal plane, and t_f is the step duration. At the same time, the way points of the z-axis hip trajectory are constrained as follows:

$$z_{hip}(y_{hip}(0)) = hip_z \qquad \dot{z}_{hip}(y_{hip}(0)) = 0 \tag{20}$$

$$z_{hip}(y_{hip}(t_m)) = hip_{zm} \qquad \dot{z}(y_{hip}(t_m)) = 0 \tag{21}$$

$$z_{hip}(y_{hip}(t_f)) = hip_{zf} \qquad \dot{z}(y_{hip}(t_f)) = 0 \tag{22}$$

where hip_{zm} is the hip height when the hip and support foot are aligned along the y-axis, calculated by modeling the support leg as a Crank-Connecting Rod mechanism [14] and fixing the desired support knee angle θ_m at t_m, as shown

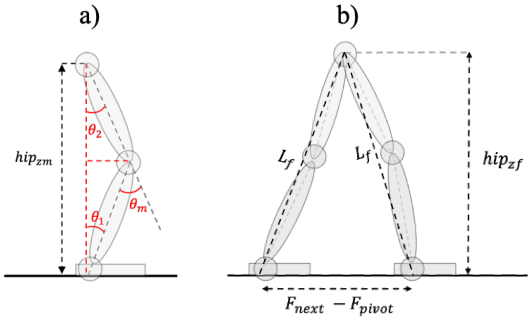

Fig. 4: a) Crank-Connecting Rod scheme used to calculate hip_{zm}. b) Scheme showing the calculation of the final hip height hip_{zf}.

Algorithm 1 CFFTG algorithm

Initialize Gaussian Random Variables $\mathcal{N}_y(\mu_y, \sigma_y), \mathcal{N}_z(\mu_z, \sigma_z)$
$y_{peak}, z_{peak} \leftarrow F_{next}/2, z_{avg} + h_O$
stop $\leftarrow$ false
$C_{best} \leftarrow \infty$
while stop = false **do**
  trajectory $\leftarrow$ **generateTrajectory**(y_{peak},z_{peak})
  if **kinematicConstraintsSatisfied**(trajectory) **then**
    $C \leftarrow$ **evaluateCost**(trajectory, obstacles)
    if $C = 0$ **then**
      stop $\leftarrow$ true
    else
      if $C < C_{best}$ **then**
        $\mu_y, \mu_z \leftarrow y_{peak}, z_{peak}$
        $\sigma_y, \sigma_z \leftarrow \sigma_y/\alpha, \sigma_z/\alpha$
        $C_{best} \leftarrow C$
      end if
    end if
  end if
  $y_{peak} \leftarrow$ **drawFromDistribution**($\mathcal{N}_y(\mu_y, \sigma_y)$)
  $z_{peak} \leftarrow$ **drawFromDistribution**($\mathcal{N}_z(\mu_z, \sigma_z)$)
end while
return trajectory

in Fig. 4a):

$$\theta_1 = \arctan \frac{\sin(\theta_m)}{\cos(\theta_m) + \frac{L_2}{L_1}} \tag{23}$$

$$\theta_2 = \arcsin(\frac{L_2}{L_1} \sin(\theta_1)) \tag{24}$$

$$hip_{zm} = L_1 \cos(\theta_2) + L_2 \cos(\theta_1) \tag{25}$$

Finally, hip_{zf} is the desired hip height at the end of the step. This value can be calculated by fixing a final knee angle θ_f (equal for both legs) and performing the same calculations done in Eqs. (23)-(25) to obtain the segment L_f (see Fig. 4b); at this point, by considering the triangle with sides L_f and base $F_{next} - F_{pivot}$ the value of hip_{zf} is obtained as the height of such triangle.

3) Collision-Free Foot Trajectory Generator (CFFTG): At this point, the hip trajectory **hip**(t) in the sagittal plane is defined, and the module calculates the appropriate foot trajectory to surpass the obstacle. Again, we assume a semi-rigid connection between the foot and the ankle: modeling the ankle as an elastic joint which is fixed to $90°$ during swing and allows passive dorsi-flexion during weight support, the calculation of the foot trajectory coincides with calculating the trajectory of the ankle reference frame in the sagittal plane **ankle**$(t) = \{y_{ankle}(t), z_{ankle}(t)\}$. As before, all of the calculations below are performed in the swing ankle reference frame. As for the CoM, also the ankle trajectory is modeled as a cubic polynomial in both time and space as

$$y_{ankle}(t) = a_3 t^3 + b_3 t^2 + c_3 t + d_3 \tag{26}$$

$$z_{ankle}(t) = a_4 y_{ankle}(t)^3 + b_4 y_{ankle}(t)^2 + c_4 y_{ankle}(t) + d_4 \tag{27}$$

To estimate the polynomial parameters, the following way points are considered:

$$y_{ankle}(0) = 0 \quad \dot{y}_{ankle}(0) = 0 \tag{28}$$

$$y_{ankle}(t_r) = y_{peak} \quad \dot{y}_{ankle}(t_r) = v_{peak} \tag{29}$$

$$y_{ankle}(t_f) = F_{next} \quad \dot{y}_{ankle}(t_f) = 0 \tag{30}$$

$$z_{ankle}(y_{ankle}(0)) = 0 \quad \dot{z}_{ankle}(y_{ankle}(0)) = 1 \tag{31}$$

$$z_{ankle}(y_{ankle}(t_r)) = z_{peak} \quad \dot{z}_{ankle}(y_{ankle}(t_r)) = 0 \tag{32}$$

$$z_{ankle}(y_{ankle}(t_f)) = 0 \quad \dot{z}_{ankle}(y_{ankle}(t_f)) = -1 \tag{33}$$

The initial and final positions are constrained by the initial exoskeleton configuration and the selected foothold F_{next}, respectively. z_{peak} represents the step height and y_{peak} the position along the y-axis in which the foot reaches the step height with forward velocity v_{peak}. This two parameters represent the degree-of-freedom of the motion to be planned. If no obstacle is present, $y_{peak} = F_{next}/2$ and $z_{peak} = z_{avg}$, where z_{avg} is the standard step height. Knowing the value of y_{peak} and z_{peak}, the whole gait trajectory can be analytically computed through inverse kinematics, and the angular trajectories sent to the exoskeleton low-level controllers for movement execution. On the other hand, in the presence of an obstacle in front of the swinging leg, it is necessary to identify a feasible gait trajectory that allows a safe execution of the step avoiding collisions. To do so, we propose a novel iterative-based algorithm called CFFTG; the CFFTG implements a randomized iterative state-space search [15] that aims at identifying the best pair $\{y_{peak}, z_{peak}\}$ generating a feasible trajectory which is the closest to the standard step (i.e., in the absence of obstacles) and not in collision with the detected obstacle. The procedure is detailed in Algorithm 1. At each iteration i the values of y_{peak} and z_{peak} are drawn from two Gaussian distributions $\mathcal{N}_y(\mu_y, \sigma_y)$ and $\mathcal{N}_z(\mu_z, \sigma_z)$. Initially, the distributions are centered in $\mu_y = F_{next}/2$ and $\mu_z = z_{avg} + h_O$, respectively, where h_O is the obstacle height. Given the estimated y_{peak} and z_{peak} at the $i-th$ iteration, the whole gait trajectory G_i is again computed through inverse kinematics as before. The feasibility of the generated trajectory is evaluated first by checking the kinematic constraints:

$$||\textbf{hip}(t) - \textbf{ankle}(t)|| \leq L_1 + L_2 \tag{34}$$

$$\theta_{minJ} \leq \theta_J(t) \leq \theta_{maxJ} \tag{35}$$

with angular limits θ_{minJ} and θ_{maxJ}, with $J \in \{RH, RK, LH, LK\}$, of each joint generally defined by the exoskeleton mechanical design and construction. If at least one point of the trajectory G_i does not satisfy the kinematic constraints, the trajectory is discarded and new values for y_{peak} and z_{peak} are drawn from the Gaussian distributions $\mathcal{N}_y$ and $\mathcal{N}_z$. Otherwise, the safety of the trajectory is evaluated by computing the cost of the trajectory $C(G_i, O)$ in terms of collision risk as follow:

$$C(G_i, O) = \sum_{t \in G_i} \begin{cases} d_s - d(\overline{\text{HT}}(t), O), & d(\overline{\text{HT}}(t), O) < d_s \\ 0, & otherwise \end{cases} \tag{36}$$

TABLE I: Control Parameters

Parameter	Value
Δ_1	3 cm
Δ_2	5 cm
t_f	4 s
t_m	$t_f/2$
θ_m	10°
θ_f	5°
y_{max}	75 cm (First step) / 110 cm (Double step)
y_{avg}	$y_{max}/2$
z_{avg}	15 cm
d_s	3 cm
α	1.5
Max iter	100
N	5

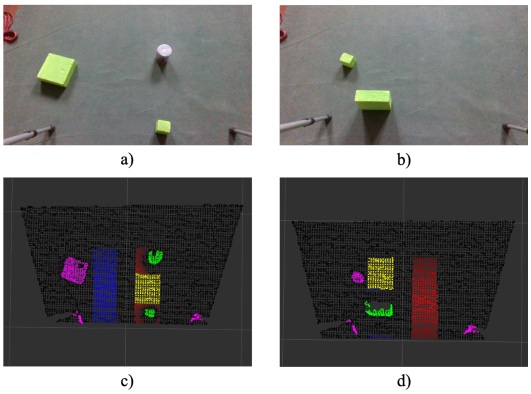

Fig. 5: a)-b) RGB image of different scenes. c)-d) Segmented point cloud returned by the Scene Understanding module, along with the foothold computed by the Adaptive Gait Planning module.

TABLE II: Scene Understanding and Pose Estimation Errors

Measurement	Mean	Standard Deviation
Obstacle width (y-axis)	1.364 cm	0.841 cm
Obstacle height (z-axis)	1.06 cm	0.389 cm
Distance ankle-obstacle (y-axis)	1.408 cm	0.372 cm
Error across all measurements	1.254 cm	0.625 cm

where $d(\overline{\mathrm{HT}}(t), O)$ measures the distance between the foot segment $\overline{\mathrm{HT}}$ (i.e., from heel H to toe T, see Fig. 2) and the closest point on the obstacle's surface. d_s is a minimum safe distance parameter. Thus, this collision cost is minimum ($C(G_i, O) = 0$) if the foot keeps a distance at least equal to d_s from the obstacle during the whole trajectory. In this case, the trajectory is considered safe and the algorithm terminates. Otherwise, the algorithm checks if the current trajectory has improved upon the previous ones by comparing the current cost C_i with the lowest cost C_{best} reached up to that point. If $C_i < C_{best}$, the centers of the Gaussian distributions μ_y and μ_z are updated with the current values of y_{peak} and z_{peak}, respectively, and the variances σ_y and σ_z are divided by a scale factor $\alpha > 1$, reducing the dimensionality of the searching space. Then, new values for y_{peak} and z_{peak} are drawn from $\mathcal{N}_y$ and $\mathcal{N}_z$, and the process is repeated until a solution is found or the maximum number of iterations is reached. To mitigate the effect of the random initialization, if no solution is found the search is repeated for a maximum of N times. When a solution is found, the angular trajectories calculated by the inverse kinematics are sent to the low-level controllers of the exoskeleton joints for executing the swing motion, and the whole process is repeated for the other leg during the following double-support phase.

IV. Experimental Validation and Results

A. Exoskeleton Platform and Control Parameters

The solution has been validated on the ALICE exoskeleton, an open-source platform[1]. The exoskeleton has four active joints (two hips and two knees) driven by brushed DC motors and a worm gearbox. Each joint is equipped with a linear encoder for measuring the joint position in real time. The high-level control (Beelink U59 Mini PC, Intel Celeron 4-Core @ 2.9 GHz, 8 GB RAM) and the power supply are placed off board to reduce the overall weight worn by the subject, and connected to the exoskeleton through a bundle of 15 m-cables. The length of the links and the size of the pelvis frame were manually adjusted by an expert operator to fit the body characteristics of the user. The exoskeleton mechanical design was then customized to accommodate a Realsense D435 RGB-D camera fixed at the exoskeleton pelvis and connected to the high-level controller performing the computations.

The EAGP framework and the high-level exoskeleton control were implemented using the Robot Operating System (ROS) middleware [2]. The control parameters that were used for the experiments are reported in Table I and were manually selected by the operator. Note that the maximum step length y_{max} is different between first step and double step, since it is possible to bring the swing foot further ahead in the latter case.

B. Experimental protocol

The performance of the proposed EAGP were verified through an experimental protocol with one subject performing 36 trials of obstacle-crossing steps in different configurations and with different obstacles. In particular, three types of obstacles were considered: a small cube-shaped object of size 7 (length) x 7 (width) x 5 (height) cm, a cylindrical-shaped object of size 5 (radius) x 10 (height) cm, and a big rectangle-shaped object of size 6 (length) x 17 (width) x 17 (height) cm. Moreover, two different step configurations were considered: a *First step* condition in which the exoskeleton starts from straight standing and the obstacle must be crossed with the first step; a *Double step* condition in which the exoskeleton starts again from straight standing, performs a first step without obstacles, and crosses the obstacles with the second step. All the conditions were repeated three times for both right and left legs to ensure the stability and robustness of the solution.

[1]https://hackaday.io/project/25105-alice-robotic-exoskeleton
[2]Code available at: https://github.com/exoskeleton-iaslab/environment-adaptive-gait-planning.git

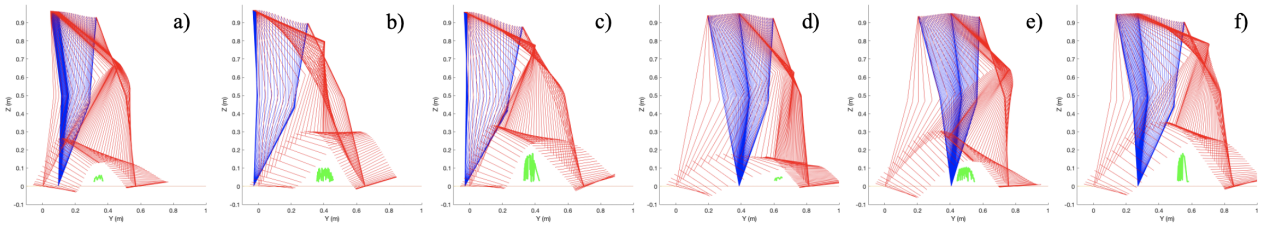

Fig. 6: Output trajectories produced by the Adaptive Gait Planning Module in different scenarios: a) First (left) step surpassing the small cube. b) First (right) step surpassing the coffee can. c) First (left) step surpassing the tall rectangle. d) Double (right) step surpassing the small cube. e) Double (left) step surpassing the coffee can. f) Double (right) step surpassing the tall rectangle.

Fig. 7: Frame sequence related to one of the experimental trials (initial left step surpassing the big rectangle-shape object).

TABLE III: Experimental trials results

Type of step	Obstacle	Trials	Failed trials	Step Length (Mean ± S.D.)	Step Height (Mean ± S. D.)	Iterations (Mean ± S.D.)
First step	Small rectangle	6	0	55.74 ± 4.59 cm	21.71 ± 5.95 cm	3.6 ± 1.74
First step	Coffee can	6	0	61.10 ± 3.91 cm	26.73 ± 4.01 cm	23.5 ± 29.91
First step	Big rectangle	6	0	69.69 ± 4.09 cm	34.26 ± 2.75 cm	126.33 ± 105.33
Double step	Small rectangle	6	0	78.13 ± 6.33 cm	19.65 ± 3.62 cm	13.83 ± 17.96
Double step	Coffee can	6	0	79.17 ± 9.25 cm	23.76 ± 3.41 cm	48.16 ± 73.4
Double step	Big rectangle	6	1	82.63 ± 6.25 cm	34.77 ± 2.15 cm	52.33 ± 58.03

C. Scene Understanding and Exoskeleton Pose validation

Fig. 5 shows the RGB images captured by the camera in two of the 36 trials and the corresponding output of the Scene Understanding module. It is shown that the system is effectively able to distinguish the ground from obstacles (green dots), separating them from objects not occluding the walking path (violet dots), such as the user's foot and the crutches. In addition, it can be seen that the shape and attitude of the obstacles are well reconstructed. To quantify the performance of the detection, in all the 36 trials we compared the main obstacle dimensions (i.e., height and width) and distance from the ankle joint of the swinging foot (i.e., along the forward direction) estimated by the system during double support with the real values measured manually. Table II reports the performance in terms of mean and standard deviation of the reconstruction error. We can conclude that the obstacles' shape and position with respect to the exoskeleton are very well recognized with an average reconstruction error lower than 15 mm in all the considered measurements, which is in line with the state-of-the-art [6], [11]. Moreover, it is worth highlighting that no terrain information was provided in advance as well as no ideal light conditioning was considered. Fig. 5 also shows the 2D-tracks created in front of each foot (blue track for the left leg, red track for the right leg). According to which is the swinging leg, the outcome of the foothold selection is represented by the yellow area, highlighting the correct identification of a final foot position crossing the detected obstacle.

D. Adaptive Gait Planning validation

Table III displays the results achieved by the proposed solution during the experiments of obstacle crossing. In all the trials the iterative model was able to converge to a feasible foot trajectory leading to a successful execution of the step cycle without collisions. The system was able to automatically adapt the output trajectories to each obstacle, increasing step length and height proportionally with increasing obstacle size. It is noteworthy that, out of 36 trials, only one trial resulted in a collision during the step execution which happened when the LLE had to surpass the big rectangle-shape object closest to the swing foot, even if the planned trajectory was collision free. This happened due to a wrong placement of crutches by the user, introducing a strong disturbance in the CoM trajectory. The trajectories produced by the parameterized gait planning can be appreciated in Fig. 6 for six sample trials. The generation of appropriate joints' trajectories is fundamental to achieve a smooth motion of the exoskeleton respecting the kinematic constraints of bipedal locomotion, as also shown in Fig. 7. The snapshots represent a single walking trial in which the exoskeleton successfully surpass the rectangle-shape obstacle

with a first left step, even if the tall obstacle is placed in close proximity to the swing foot. A video uploaded as additional material displays as proof-of-concept the use of the obstacle avoidance during continuous walking, further validating the capabilities of the solution. Table III also reports the average number of iterations of the CFFTG. As expected, more challenging are the scenarios (i.e., bigger obstacles, crossing with the first step) more iterations are required to converge to a safe and feasible solution. Nevertheless, the average execution time for the whole pipeline from image acquisition to gait planning is about 100 ms, with a peak of about 300 ms in the most challenging situations, which meet the requirements for real-time gait applications.

V. Conclusion

In this paper, we propose a novel solution for adaptive gait planning in lower limb exoskeletons. The proposed system allows the LLE to identify the presence of obstacles along the walking path through a RGB-D camera and modify at run-time the gait pattern to achieve a safe execution of the step cycle. The proposed solution demonstrated features of stability and robustness in different experimental conditions. Moreover, the combination of an iterative state-space search approach with a parameterized gait patterns resulted in smooth foot trajectories similar to the physiological walking patterns [16] and showing faster convergence than other motion planning algorithms [17]. This aspect is fundamental in walking applications as the exoskeleton should predict and elaborate the next step motion during the double-support phase, which lasts about 400 ms in able-bodied locomotion. The lack of a common benchmark for AGP makes difficult to quantitatively compare with the state-of-the-art. Nevertheless, the proposed solution provides a more flexible generation of the gait trajectory with respect to [11], by keeping at the same time a lighter mathematical formulation and faster computation than [6]. Future work will focus on enhancing the understanding of the walking environment, and on the foothold prediction, overcoming the assumption of indoor environment with level-ground floor. Moreover, to consider the full walking path and not only the environment near the feet of the exoskeleton, the system will be integrated with visual odometry techniques available at the state-of-the-art, such as ORBSLAM [18] or KISS-ICP [19]. This would allow for a better localization of the exoskeleton in the environment and to implement footsteps planning algorithms [20] in order to anticipate the sequence of steps together with the current single-step adaptation.

Acknowledgment

The project is funded by Next Generation EU, in the context of the NRRP PE8 – Project Age-It: "Ageing Well in an Ageing Society" [DM 1557 11.10.2022].

References

[1] B. S. Rupal, S. Rafique, A. Singla, E. Singla, M. Isaksson, and G. S. Virk, "Lower-limb exoskeletons: Research trends and regulatory guidelines in medical and non-medical applications," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 1729881417743554, 2017.

[2] J. A. de la Tejera, R. Bustamante-Bello, R. A. Ramirez-Mendoza, and J. Izquierdo-Reyes, "Systematic Review of Exoskeletons towards a General Categorization Model Proposal," *Applied Sciences*, vol. 11, no. 1, pp. 1–25, Jan. 2021.

[3] W. Y.-W. Tung, M. McKinley, M. V. Pillai, J. Reid, and H. Kazerooni, "Design of a Minimally Actuated Medical Exoskeleton With Mechanical Swing-Phase Gait Generation and Sit-Stand Assistance." American Society of Mechanical Engineers Digital Collection, Mar. 2014.

[4] H. Mohamad and S. Ozgoli, "Online gait generator for lower limb exoskeleton robots: Suitable for level ground, slopes, stairs, and obstacle avoidance," *Robot. Auton. Syst.*, vol. 160, no. C, Feb. 2023. [Online]. Available: https://doi.org/10.1016/j.robot.2022.104319

[5] Z. Li, K. Zhao, L. Zhang, X. Wu, T. Zhang, Q. Li, X. Li, and C.-Y. Su, "Human-in-the-Loop Control of a Wearable Lower Limb Exoskeleton for Stable Dynamic Walking," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 5, pp. 2700–2711, Oct. 2021.

[6] Y. Hua, H. Zhang, Y. Li, J. Zhao, and Y. Zhu, "Vision Assisted Control of Lower Extremity Exoskeleton for Obstacle Avoidance With Dynamic Constraint Based Piecewise Nonlinear MPC," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 267–12 274, Oct. 2022.

[7] B. Laschowski, W. McNally, A. Wong, and J. McPhee, "Computer Vision and Deep Learning for Environment-Adaptive Control of Robotic Lower-Limb Exoskeletons," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Nov. 2021, pp. 4631–4635.

[8] Y. Feng, L. Xia, Y. He, C. Wang, Z. Yan, and X. Wu, "Stairs Reconstruction with 3D Point Cloud for Gait Generation of Lower Limb Exoskeleton Robot," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2019, pp. 2019–2024.

[9] J. Wang, D. Wu, Y. Gao, X. Wang, X. Li, G. Xu, and W. Dong, "Integral Real-time Locomotion Mode Recognition Based on GA-CNN for Lower Limb Exoskeleton," *J Bionic Eng*, vol. 19, no. 5, pp. 1359–1373, Sep. 2022.

[10] X. Zhao, W.-H. Chen, B. Li, X. Wu, and J. Wang, "An adaptive stair-ascending gait generation approach based on depth camera for lower limb exoskeleton," *Review of Scientific Instruments*, vol. 90, no. 12, Dec. 2019.

[11] D.-X. Liu, J. Xu, C. Chen, X. Long, D. Tao, and X. Wu, "Vision-Assisted Autonomous Lower-Limb Exoskeleton Robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3759–3770, Jun. 2021.

[12] A. Plaza, M. Hernandez, G. Puyuelo, E. Garces, and E. Garcia, "Lower-Limb Medical and Rehabilitation Exoskeletons: A Review of the Current Designs," *IEEE Reviews in Biomedical Engineering*, vol. 16, pp. 278–291, 2023.

[13] C. Dambreville, A. Labarre, Y. Thibaudier, M.-F. Hurteau, and A. Frigon, "The spinal control of locomotion and step-to-step variability in left-right symmetry from slow to moderate speeds," *Journal of neurophysiology*, vol. 114, no. 2, pp. 1119–1128, 2015.

[14] Z. Guo, H. Yu, and Y. H. Yin, "Developing a mobile lower limb robotic exoskeleton for gait rehabilitation," *Journal of Medical Devices*, vol. 8, no. 4, 2014.

[15] H. H. Hoos and E. Tsang, "Chapter 5 - local search methods," in *Handbook of Constraint Programming*, ser. Foundations of Artificial Intelligence, F. Rossi, P. van Beek, and T. Walsh, Eds. Elsevier, 2006, vol. 2, pp. 135–167.

[16] L.-S. Chou, K. R. Kaufman, R. H. Brey, and L. F. Draganich, "Motion of the whole body's center of mass when stepping over obstacles of different heights," *Gait & Posture*, vol. 13, no. 1, pp. 17–26, 2001.

[17] K. Balakrishana Reddy, G. Suresh, R. K. Mandava, A. Kumar, and T. Ch, "A short review on biped robots motion planning and trajectory design," *Advancement in Materials, Manufacturing and Energy Engineering, Vol. II*, pp. 471–480, 2022.

[18] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[19] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[20] X. Wu, J. Li, L. Liu, and D. Tao, "The visual footsteps planning system for exoskeleton robots under complex terrain," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.