


RESEARCH ARTICLE

Enhancement of humanoid robot locomotion on slippery floors using an adaptive controller

Luís Almeida¹ , Vítor Santos^{1,2}  and João Ferreira^{3,4} 

¹Institute of Electronics and Informatics Engineering of Aveiro (IEETA), Aveiro University, Aveiro, 3810-193, Portugal, ²Department of Mechanical Engineering, Aveiro University, Aveiro, 3810-193, Portugal, ³Department of Electrical Engineering, Superior Institute of Engineering of Coimbra, Coimbra, 3030-199, Portugal, and ⁴Institute of Systems and Robotics (ISR), Universidade de Coimbra, Coimbra, 3030-290, Portugal

Corresponding author: Luís Almeida; Email: lmalmeida@ua.pt

Received: 14 September 2023; **Revised:** 21 November 2023; **Accepted:** 21 December 2023;

First published online: 17 January 2024

Keywords: adaptive control; PID controller; humanoid robot; data processing

Abstract

This paper presents a comprehensive strategy to improve the locomotion performance of humanoid robots on various slippery floors. The strategy involves the implementation and adaptation of a divergent component of motion (DCM) based control architecture for the humanoid NAO, and the introduction of an embedded yaw controller (EYC), which is based on a proportional-integral-derivative (PID) control algorithm. The EYC is designed not only to address the slip behavior of the robot on low-friction floors but also to tackle the issue of non-straight walking patterns that we observed in this humanoid, even on non-slippery floors. To fine-tune the PID gains for the EYC, a systematic trial-and-error approach is employed. We iteratively adjusted the P (Proportional), I (Integral), and D (Derivative) parameters while keeping the others fixed. This process allowed us to optimize the PID controller's response to different walking conditions and floor types. A series of locomotion experiments are conducted in a simulated environment, where the humanoid step frequency and PID gains are varied for each type of floor. The effectiveness of the strategy is evaluated using metrics such as robot stability, energy consumption, and task duration. The results of the study demonstrate that the proposed approach significantly improves humanoid locomotion on different slippery floors, by enhancing stability and reducing energy consumption. The study has practical implications for designing more versatile and effective solutions for humanoid locomotion on challenging surfaces and highlights the adaptability of the existing controller for different humanoid robots.

1. Introduction

Humanoid robots are anthropomorphic machines that aim to emulate the locomotion and behavior of humans. Although significant advancements have been made in robot's locomotion, the challenge of navigating on slippery floors remains a difficult and unresolved issue, with significant potential for further improvements. Just as humans use their senses to adapt to changing environments, we believe that humanoid robots should rely on sensor feedback to achieve the same. By utilizing sensor data, robots can better adjust to varying surfaces and conditions, improving their ability to navigate on slippery floors and other challenging terrains [1], as well as perform seemingly simple tasks like setting their home position [2].

In past studies [3, 4], it was hypothesized that adapting the controller of the robot based on the known friction coefficient of various floors could lead to efficient and effective locomotion on such surfaces. Building upon the development of instrumented shoes (ITShoes) and machine learning algorithms that enable real-time detection of the coefficient of friction for various floor types, this work proposes a comprehensive strategy to enhance humanoid locomotion on slippery surfaces through adaptive control. In addition to addressing the challenges posed by varying slippery floor conditions, a peculiar behavior was also observed in the humanoid when instructed to walk on a straight path. This behavior is attributed to

its unconventional building, which will be explained in the next section. As such, the proposed controller aims not only to address slip behavior on slippery floors but also to mitigate the issue of non-straight walking patterns resulting from the humanoid's particular mechanical construction.

Several techniques have been proposed in the literature for generating Center of Mass (CoM) trajectories based on the concepts of zero moment point (ZMP) [5] and linear inverted pendulum [6]. In recent years, there has been significant research on bipedal locomotion, which has resulted in the introduction of new concepts such as the divergent component of motion (DCM) and virtual repellent point (VRP) [7, 8]. These concepts allow for the decomposition of the second-order dynamics of the CoM into two first-order linear dynamics, where the CoM converges to the DCM (stable dynamics) and the DCM diverges away from the VRP (unstable dynamics) [7]. These concepts have been utilized in various methods of generating CoM trajectories and stabilizing humanoid robots during locomotion.

The strategy described in this paper is founded on an implementation and adaptation of a DCM-based control architecture. The DCM concept has been applied to various types of robots, including bipeds [9–14], quadrupeds [15, 16], and exoskeletons [17, 18], and has been shown to be effective in achieving stable and dynamic locomotion. It has also been used in applications such as prosthetics and rehabilitation robotics, where maintaining balance and stability are critical for safe and effective movement. We follow the DCM-based control architecture presented in ref. [19], where the authors present and compare several DCM-based implementations of a layered control architecture. A unicycle-based planner is used to generate desired DCM and foot trajectories for the fixed trajectory optimization layer. The simplified model control layer comprises two main controllers for DCM tracking, along with an additional inner controller that utilizes 6-axes force torque sensors (F/T) to ensure precise tracking of CoM and ZMP [20]. The two main controllers are an instantaneous feedback controller and a model predictive control (MPC) approach. One common approach to generate feasible CoM trajectories is to combine the linear inverted pendulum model (LIPM) with MPC techniques, which are also referred to as receding horizon control [21]. The MPC is a control technique that uses a mathematical model of the system to predict its future behavior and then optimizes the control inputs over a finite time horizon to minimize a cost function while satisfying constraints. The Whole-Body Quadratic Programming (WBQP) control layer ensures the tracking of the desired CoM and feet trajectories by presenting velocity and inverse kinematics.

Diverse strategies for mitigating slipperiness in walking humanoids primarily revolve around studying and adapting humanoid gaits to compensate for the yaw moment [22] or minimize the required coefficient of friction [23]. While some approaches focus on finding the best possible trajectory, our work takes a different approach by adapting in real time to the slipperiness of various floors the robot may encounter. In our approach, we address slipperiness through continuous, small compensations in robot yaw and joint trajectories. This enables the robot to effectively mitigate slipperiness, minimize slip in subsequent steps, and maintain its initial trajectory with real-time corrections for any deviations that may occur. It is worth noting that while existing studies show effective slip avoidance with coefficient of frictions as low as 0.3, our research specifically targets lower friction levels, including 0.26, 0.2, and even as low as 0.11.

In the current work, we propose an embedded yaw controller (EYC) as an extension to the existing control strategy based on the MPC controller from ref. [19]. The EYC is specifically designed to address slip behavior on slippery floors and to rectify the non-linear paths observed due to the robot's suspicious building technique. It functions as a proportional-integral-derivative (PID) controller, which allows us to dynamically control and adapt the humanoid gait in real time. This approach draws upon the well-established use of PID controllers in optimizing various aspects of humanoid locomotion, such as stability, efficiency, and robustness of walking, as demonstrated in prior works such as refs. [24–26]. Thus, in this work, the digital PID control scheme is leveraged to implement the EYC for precise control of the humanoid gait, by continuously monitoring its torso orientation in response to slip behavior and non-linear path observations.

The primary contribution of this study lies in introducing a tailored controller for the humanoid NAO, intricately coupled with an EYC, aimed at substantially enhancing humanoid locomotion on designated

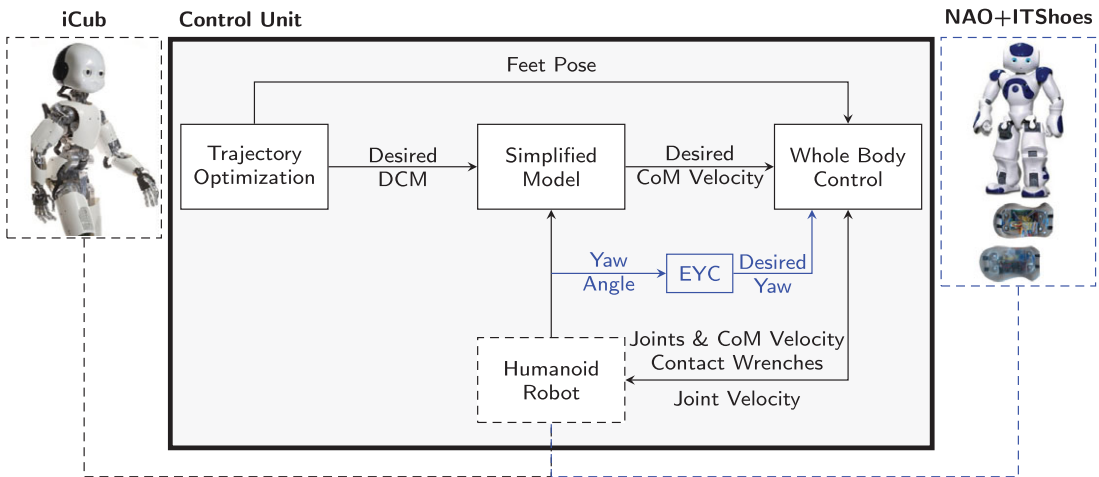


Figure 1. Modified humanoid 3-layer control architecture [19] with an additional EYC block. The EYC is a controller used to adapt the robot's motion to low-friction environments and to correct the unexpected curved path of the humanoid NAO by adjusting its behavior based on feedback from an IMU (Inertial Measurement Unit) sensor.

slippery surfaces. Furthermore, we have proposed specific metrics, including task duration, stability, and energy consumption, to systematically measure performance. This framework facilitates a comprehensive evaluation and comparison of diverse implementations and enhancements within the ambit of the introduced controller and EYC. By enhancing the robot's capability to navigate slippery terrains, this advancement not only extends its capacity to undertake previously unattainable tasks but also significantly improves the efficiency and speed at which these tasks are executed.

The remainder of the paper is structured into the following sections. Section 2 provides an overview of the DCM-based control architecture that has been implemented, covering key concepts, components, and its integration into the NAO robot. Furthermore, the integration of an EYC within the architecture is detailed. Section 4 describes the preliminary locomotion experiments setup and the experiments conducted with varying step frequencies and PID gains. The metrics used to evaluate the performance of the system are also discussed, and the results obtained are presented. Finally, in Section 6, conclusions are provided, highlighting the achieved improvements, and recommendations for future work are given.

2. Humanoid whole-body controller

The whole-body controller used in this work is based on the three-layer control architecture developed and implemented on the iCub robot [19]. The three-layer architecture is presented in Fig. 1.

This work demonstrates the potential of the EYC by successfully integrating it into the three-layer control architecture, where it effectively addresses slip behavior and rectifies non-linear paths resulting from the robot's particular building technique. Furthermore, the versatility of the three-layer controller is demonstrated by successfully implementing it on a different humanoid robot, highlighting its potential for application in a wide range of robotic systems.

2.1. Trajectory optimization

The purpose of this layer is to plan and evaluate the desired footstep positions and the desired feet and DCM trajectories for a humanoid robot. The robot is approximated as an unicycle, with the feet represented by the unicycle wheels. The footsteps are planned by sampling the unicycle trajectories, with

each position associated with a time instant. The impact time is considered as a decision variable, which allows for the selection of feet positions, duration, and step length. Once the footsteps are planned, the desired feet trajectory is obtained by cubic spline interpolation. The DCM trajectory is chosen so as to satisfy a specific time evolution. The DCM trajectory along the walking pattern can be computed recursively, but the presented planning method only takes into account single support phases. To ensure that the ZMP trajectory remains smooth and continuous, it is important to have a reference trajectory for the DCM with smoothly varying derivatives. This means that the rate of change of the DCM trajectory should be continuous, allowing for seamless transitions between different segments of the trajectory. To achieve this, a third-order polynomial is used to smoothly connect the edges of the DCM reference trajectory. The parameters of the polynomial are carefully selected to ensure that both the velocity and position of the trajectory satisfy certain boundary conditions, ensuring a continuous and smooth motion [19].

2.2. Simplified model

The middle layer of the robot's control system utilizes a simplified model control layer that employs a MPC DCM controller. The purpose of this controller is to ensure that the robot's DCM accurately tracks its intended trajectory. The DCM is a specific point in space that represents the robot's movement and serves as the point at which the robot would fall if it stopped moving.

To achieve this level of precision, the humanoid robot's movement is approximated using the LIPM, a widely recognized model for walking on flat surfaces [27]. The LIPM guarantees that the CoM of the robot remains at a constant height on a horizontal plane while walking. The MPC algorithm utilizes this model to predict the robot's future movements and generates a control signal that maintains its stability and trajectory.

This algorithm takes into account the desired trajectory and the robot's motion constraints to generate a smooth trajectory for the ZMP, which is essential to the robot's stability. The ZMP must remain within the robot's support polygon to avoid falling.

In this middle layer, an independent CoM-ZMP controller ensures that the robot's CoM and ZMP are properly tracked using a force torque sensors (F/T). This inner CoM-ZMP controller is presented in ref. [20] and is based on the Kinematic Resolution of CoM Jacobian (KRCJ) method with embedded motion. KRCJ is a mathematical method for computing the robot's center of mass and its Jacobian matrix, which describes the relationship between the robot's joint angles and its CoM.

This approach aims to enhance the robot's stability and balance during walking by adjusting its joint angles in real time using the KRCJ method. The embedded motion technique allows the robot to adapt its movements to changes in the environment, such as uneven terrain or unexpected obstacles [19].

2.3. Whole body quadratic programming control

The third layer of the control architecture implemented in this work is the Whole-Body Quadratic Programming (WBQP) control layer. WBQP is a common approach used in robotics for generating motion plans or control commands that optimize the performance of the robot's whole-body motion while accounting for different constraints. Its main responsibility here is to track the desired CoM and foot trajectories, employing velocity and inverse kinematic controllers to ensure accurate and smooth movement control. The achievement of this control objective involves defining a cost function, which is minimized to obtain the desired robot velocity [19].

3. Methodology

The methodology in this study centers on implementing and adapting the humanoid controller detailed in the preceding section, coupled with introducing an EYC. This integrated approach empowers the

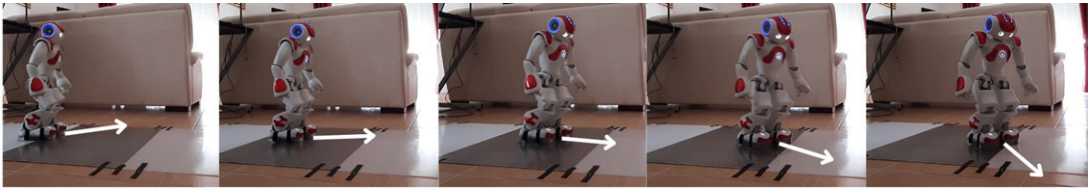


Figure 2. Illustration of the NAO robot's inability to maintain a straight path. The robot was commanded to walk in a straight line, but deviated from its intended path.

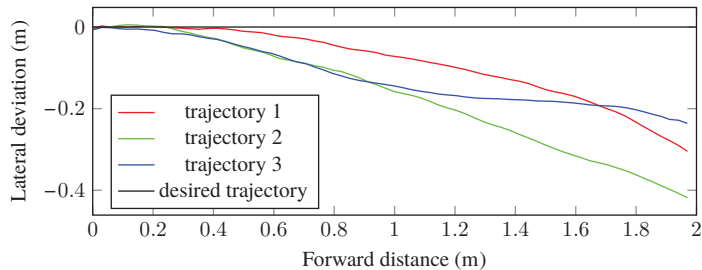


Figure 3. Visualizing humanoid locomotion behavior: trajectories illustrating straight walking command execution.

humanoid to navigate slippery floors seamlessly while maintaining its intended path and executing tasks effectively.

3.1. Implementation in the NAO robot

The entire architecture was adapted to be used in the humanoid NAO. The NAO robot displays certain peculiarities observed in both simulations and the real system. The most significant issue is that the robot does not maintain a straight path when commanded to do so. This behavior is illustrated in Fig. 2, where it can be seen that the humanoid changes direction despite being commanded to walk a straight line.

The issue of the robot deviating from a straight path is more prominent on slippery floors, but it also occurs on floors with a high friction coefficient. We suspect that the problem is related to the physical configuration of the pelvis joint `LHipYawPitch` and `RHipYawPitch`, which are driven by only one motor and, thus, cannot be controlled independently, with priority given to `LHipYawPitch`. It is possible that the controller has a glitch when sending the desired position for both joints, leading to the observed behavior. Illustrative trajectories are depicted in Fig. 3, showcasing the desired straight trajectory and the actual trajectories of the humanoid.

To enforce the locomotion along a straight path and correct the lateral deviations of the robot, the EYC control system was developed based on the measurement of robot's heading using the onboard inertial measurement unit (IMU). This controller dynamically adjusts the target set-point to continuously correct the robot's direction of motion by monitoring the humanoid's torso orientation and compensates for changes in direction that can occur when the robot slips while walking on very slippery floors.

The effectiveness of this modification was also evaluated through experiments on the humanoid robot NAO, and the results showed that the EYC was able to significantly improve the robot's stability and reduce the impact of slips on its movements. A joint velocity limit was also added to the controller as a safety measure to ensure that the robot's joint velocity limits were not exceeded during movement. This modification was necessary to ensure the safety and stability of the robot's movements, especially during dynamic motions that require quick changes in joint velocities.

Table I. *Points adapted in the original walking controller to fit the NAO requirements.*

Item	Values
URDF	Joints, links, and robot description
SDF	Model, physical properties, environmental, and material properties (i.e., friction coefficients)
Whole-body configs	Joints, frames, sensors, devices, and others
Trajectory generation	Gait parameters, such as minimum and maximum step length, height, width, duration, landing velocity, and others
Logger	Joints position, DCM, CoM, ZMP, sensors, EYC inputs/outputs, and others
Walking controller configs	Inner ZMP parameters, forward and inverse kinematics, joint's PID, and others
Gazebo	Physical environmental, such as solver and constraints parameters

To implement this architecture on the NAO robot, a detailed step-by-step process was undertaken. Initially, we installed the necessary software dependencies for the walking module, including Yet Another Robot Platform (YARP), and set up environmental variables. Subsequently, we modeled the robot using both the Unified Robot Description Format (URDF) and the Simulation Description Format (SDF) files. The URDF and SDF files offer a comprehensive breakdown of the disparities between the iCub and NAO robots, covering various aspects such as links, joints, masses, positions, inertias, sensors, and more. This documentation provides a complete understanding of the unique kinematic and dynamic traits inherent to each robot. In the modeling phase, we defined joint and link structures, masses, inertias, and motors in the URDF. Additionally, we specified physical and environmental properties in the SDF, including critical details like friction coefficients. With the robot model established, the focus shifted to adapting trajectory generation parameters to meet the specific requirements of the NAO. To better suit our smaller robot, we made several adjustments, including setting limits for gait parameters such as maximum and minimum step length, frequency, height, angle variation, step landing velocity, foot apex time, and switch-over swing ratio. These modifications aimed to ensure the robot could walk smoothly and efficiently. Furthermore, trajectory generation parameters were fine-tuned to enhance the stability and coordination of the robot's motion. Throughout the implementation process, we made necessary adaptations to ensure the model was comprehensive, incorporating all joints and sensors essential for optimal performance. The kinematics files were specifically tailored to match our robot's structure. Beyond model differences, meticulous attention was given to adapting every hard-coded value, such as the default position of the center of mass, feet configuration, sensor positions, joint velocity filters, cut frequencies, kinematics frames definitions, and joint PID parameters. These adjustments were made carefully to align with our robot's specifications and limitations.

To achieve a final implementation that is both robust and reliable, it was necessary to adapt numerous parameters and conduct extensive testing. Some of these adaptations are presented in Table I:

To measure the forces and torques that occur on the ankle of the NAO humanoid robot during the simulated experiments, a force/torque (F/T) sensor was added to the robot's ankle model. While this provided accurate measurements in the simulated environment, the real robot does not have this sensor. To overcome this limitation, we aim to utilize the instrumented shoe (ITshoe), as shown in Fig. 4, which was previously developed in our laboratory [28].

The ITshoe is equipped with sensors that can measure the total ground reaction forces during walking. These measurements are essential to enhance the robot's interaction with the ground [29], and they can be utilized to calculate the forces and torques acting on the ankle. To simulate the output of a F/T sensor in future real experiments, the measured ground reaction forces have to be translated to the ankle location using the Eqs. (1) and (2).

$$\vec{F}_{AZ} = \vec{F}_{SZ} + (m_{is} + m_{rf} + m_{aj}) \cdot \vec{g}, \quad (1)$$

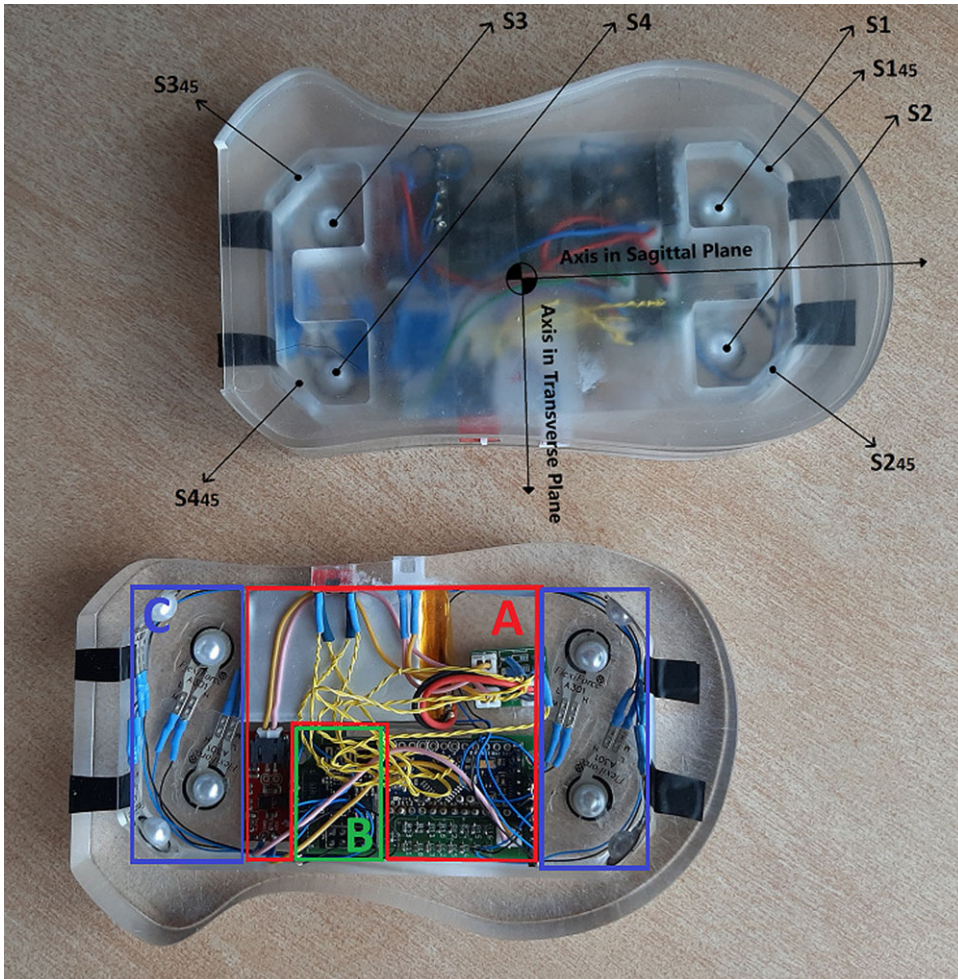


Figure 4. ITshoe schematic structure. The red block (A) is the acquisition unit, the green block (B) is the streaming unit, and the blue block (C) is the sensing unit. At the top, the image shows the position of the eight force sensors and the reference axis used to decompose the tangential forces [4].

where \vec{F}_{AZ} represents the total ankle force in the z direction, \vec{F}_{SZ} the total force in the z direction measured by the ITShoe and m_{is} , m_{rf} and m_{aj} denote the masses of the components between the sensors and the robot's ankle:

$$\begin{bmatrix} \vec{\tau}_x \\ \vec{\tau}_y \\ \vec{\tau}_z \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} r_{iy} \cdot \vec{F}_{iz} + r_{iz} \cdot \vec{F}_{iy} \\ r_{iz} \cdot \vec{F}_{ix} + r_{ix} \cdot \vec{F}_{iz} \\ r_{ix} \cdot \vec{F}_{iy} + r_{iy} \cdot \vec{F}_{ix} \end{bmatrix}, \tag{2}$$

where $\vec{\tau}_x$, $\vec{\tau}_y$, and $\vec{\tau}_z$ represent the total ankle torque in the corresponding x, y and z directions, r_{ij} , for $j \in x, y, z$, are the distances between the ITShoe sensors and the robot's ankle, and \vec{F}_{ij} , for $j \in x, y, z$, are the forces measured on the corresponding sensors in the ITshoe.

Then, in the Gazebo simulator, the physical parameters, such as mass, friction coefficients, inertias, and joint damping, were fine-tuned and optimized to ensure that the simulated robot's behavior closely emulated the real-world physics and dynamics of the actual robot.

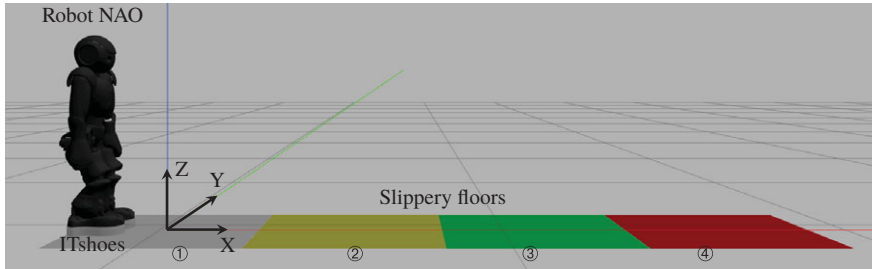


Figure 5. Layout of the simulated environment featuring the NAO robot and various types of slippery floors in the Gazebo simulator.

To capture and analyze the relevant data for each experiment, all the DCM-based controller variables, along with the robot’s joint, ZMP, CoM, F/T, and IMU sensor data, and other essential variables were recorded within a 10 ms controller cycle. By gathering this information, we were able to gain a comprehensive understanding of the robot’s behavior and performance in response to different slippery floors. In the following sections, it is demonstrated how effectively this data was utilized and manipulated to draw meaningful conclusions about the effectiveness of the methods used.

The control architecture was implemented using YARP, an open-source robotics middleware, and programmed in C++ language. More details can be found at the walking controllers repository.¹

The layout to simulate the humanoid robot navigating on different types of floors was implemented using the Gazebo simulator, as illustrated in Fig. 5.

3.1.1. EYC implementation

The role of the EYC is to continuously monitor the robot’s orientation, specifically the YAW angle, during locomotion. When the EYC detects that the robot is deviating from the commanded trajectory, it calculates a new YAW angle using a digital PID formulation [30], as shown in Eq. (3), in order to adapt the robot’s trajectory accordingly.

$$u_0 = -K_{u1}u_1 - K_{u2}u_2 + K_{e0}e_0 + K_{e1}e_1 + K_{e2}e_2. \tag{3}$$

This controller relies on a set of carefully selected and tuned coefficients, which are defined in Eqs. (4)– (14). These coefficients are used in the computation of the control input, u_0 , which depends on the previous control inputs u_1 and u_2 , as well as the error terms e_0 , e_1 , and e_2 . The equation includes the derivative terms K_{u1} and K_{u2} , and the proportional, integral, and derivative gains K_{e0} , K_{e1} , and K_{e2} , given by:

$$K_{u1} = \frac{a_0}{a_1 - a_2}, \tag{4}$$

$$K_{u2} = \frac{a_0}{a_1 + a_2}, \tag{5}$$

$$K_{e0} = a_0b_0, \tag{6}$$

$$K_{e1} = a_0b_1, \tag{7}$$

$$K_{e2} = a_0b_2, \tag{8}$$

¹<https://github.com/robotology/walking-controllers>

where

$$a_0 = 1 + NT_s, \quad (9)$$

$$a_1 = -(2 + NT_s), \quad (10)$$

$$a_2 = 1, \quad (11)$$

$$b_0 = K_p(1 + NT_s) + K_i T_s(1 + NT_s) + K_d N, \quad (12)$$

$$b_1 = -(K_p(2 + NT_s) + K_i T_s + 2K_d N), \quad (13)$$

$$b_2 = K_p + K_d N. \quad (14)$$

To trigger the controller, a timer interrupt service routine is used, and a set of global variables is defined, including the command input r , plant output y , error terms e_0 , e_1 , and e_2 , control inputs u_0 , u_1 , and u_2 , and the coefficients a_0 , a_1 , and a_2 . The error term is computed by taking the difference between the desired output and the current plant output. The proportional, integral, and derivative gains are initially set to $K_p = 1.00$, $K_i = 0.02$, and $K_d = 0.01$, respectively, while the filter coefficients are set to $N = 20$, and the sampling time is set to $T_s = 0.01$ s.

These variables and parameters are critical to the operation of the PID controller and have been empirically selected and tuned to ensure an initial effective stabilization of the robot's orientation on slippery floors. Further explanation will be provided in Section 4 to elaborate on the specific process of how these variables and parameters were determined based on practical experimentation and observation.

To estimate the robot's orientation, data obtained from the NAO robot IMU sensor were utilized, which provided measurements of the robot's angular velocity and linear acceleration. To estimate the yaw angle, we applied an extended Kalman filter (EKF), which is a recursive algorithm that estimates the state of a system that is subject to uncertain and noisy measurements [31].

The EKF uses a quaternion to represent the orientation of the system, where the scalar component represents the orientation of the system, and the three vector components represent the axis of rotation. The prediction stage of the filter propagates the quaternion forward in time using the system dynamics, while the update stage fuses the predicted state with new measurements. In the case of an IMU, new measurements include measurements of angular velocity. To update the state estimate with the new measurements, the filter models the measurement process as a linear function of the state, corrupted by additive Gaussian noise. Finally, the EKF computes the optimal estimate of the state by minimizing the mean squared error between the predicted state and the measurements. By applying this algorithm, we were able to obtain an optimal estimate of the robot's torso YAW angle over time. This estimate was then used as input to the EYC controller to stabilize the robot's trajectory, taking into consideration not only the slippery condition of the floors but also the peculiar issue related to the humanoid's non-straight walking patterns mentioned earlier. The EYC controller, utilizing the PID formulation, adjusts the robot's trajectory to ensure more stable and accurate locomotion, mitigating the effects of both slippery surfaces and the humanoid's unique build.

4. Preliminary locomotion experiments

In this section, it is presented the preliminary experiments carried out to collect locomotion data in order to analyze it and improve the humanoid controller based on the slippery floor it is faced with. Firstly, we performed an experiment where we attempted to study the impact of the robot's step frequency on the different floors by measuring the amount of slippage occurred during these experiments. According to the information obtained from these experiments, it was found that the slowest step frequency was the most preferred option across all floors, rather than a specific frequency for each individual floor. While

this insight provides valuable information for gait adaptation on slippery floors, it is inconclusive toward our goal of adapting the controller to the specific floor the robot is traversing.

Four different slippery floors that were previously studied were used to test the stability and slip recovery ability of the robot [28]. The floors used in this study were characterized by their coefficient of friction when in contact with the humanoid robot's instrumented shoe (ITshoe). Specifically, the floors exhibited coefficients of friction of 0.33, 0.26, 0.20, and 0.11, respectively, when in contact with the ITshoe.

4.1. Step frequency experiments

In this study, a series of experiments were conducted in a simulated environment to investigate the impact of humanoid locomotion on different slippery floors. The aim was to gain a deeper understanding of the relationship between the robot's step frequency, which refers to the time it takes for the robot to perform one step, and the friction coefficients of four different floors to identify the best humanoid gait for real-world settings. To achieve this, a controlled experiment was designed and implemented, in which the humanoid step frequency was varied and the humanoid CoM was measured under different types of slippery floors.

The experiment involved the humanoid robot walking five times for 60 s on each of the four different floors, using five different frequencies. The chosen frequencies were the maximum frequency of the default humanoid NAO, the minimum frequency, and three equally separated frequencies between the maximum and minimum frequencies: 0.43, 0.47, 0.51, 0.55, 0.59 s/step. After collecting data from 100 experiments (i.e., 4 floors \times 5 frequencies \times 5 repetitions), data manipulation was performed to quantify the extent of slippage of the robot on each floor. To achieve this objective, the average absolute Y-position of the CoM for each experiment was calculated.

In the next step, the average change in the direction of the robot's feet at each step was computed. This was done by taking the derivative of the average CoM Y-position data points. Then, a second derivative was applied to obtain the values that represent how much the robot slipped on each step. To exclude small slip actions that may occur due to possible errors associated with the robot's joints, controller, and/or the simulator environment, a slip threshold was added to consider only significant slip actions. The optimal threshold was determined empirically by observing the various outputs of the second-order derivative and selecting the value that best matched the noticeable slip observed in the different experiences. Specifically, a threshold of 1×10^{-6} (m/step²) was chosen. To normalize the results, the accumulated slip was divided by the number of steps in each experience since different step frequencies represent different amounts of steps. Equation (15) describes the steps used to quantify the robot's slip in each experience.

$$S(i) = \frac{d^2A(i)}{dt}, \quad |S(i)| > \theta \Rightarrow T = \frac{\sum |S(i)|}{n}, \quad (15)$$

where S represents the second-order derivative of the discrete sequence $A(i)$, which is calculated as the difference between $A(i)$ and $A(i-1)$, and $A(i+1)$ and $A(i)$, that is, $S(i) = (A(i) - A(i-1)) - (A(i+1) - A(i)) = 2A(i) - A(i-1) - A(i+1)$. This equation is used to estimate the rate of change of the CoM Y-position at each step, which is a key parameter for detecting slips. θ represents the slip threshold, n represents the number of steps and T represents the total slippage that occurred in a complete experience. It is important to note that the absolute value of each detected slip was summed since we did not distinguish the direction of the slip.

As an example of the procedure described, Fig. 6 and Fig. 7 depict the CoM Y curves obtained for three examples of the resulting lowest slip sum and three examples of the resulting highest slip sum, respectively.

The slip corresponding to the previous examples is illustrated in Fig. 8 and Fig. 9, respectively. Among these examples, we selected floor 3 with the lowest frequency as the best representative and floor 1 with the highest frequency as the worst representative.

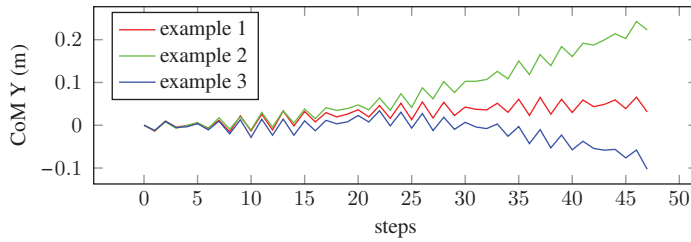


Figure 6. Absolute center of mass position in the Y-axis for three exemplary walking trials on a flat surface with a step frequency of 0.59 s per step and a coefficient of friction of 0.20, resulting in the least amount of slip.

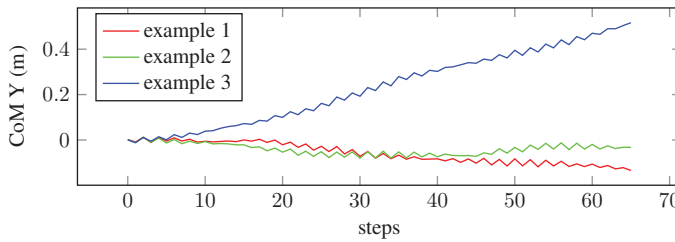


Figure 7. Absolute center of mass position in the Y-axis for three exemplary walking trials on a flat surface with a step frequency of 0.43 s per step and a coefficient of friction of 0.20, resulting in the highest amount of slip.

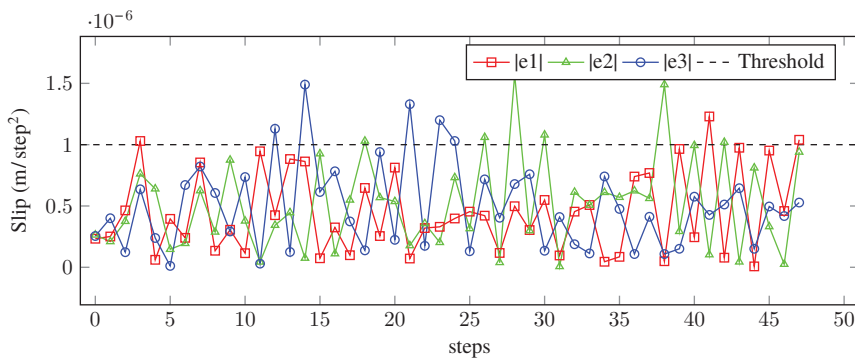


Figure 8. Slip and threshold visualization for the top three experiences with the least amount of slip (e1, e2, e3 represent examples 1, 2, 3).

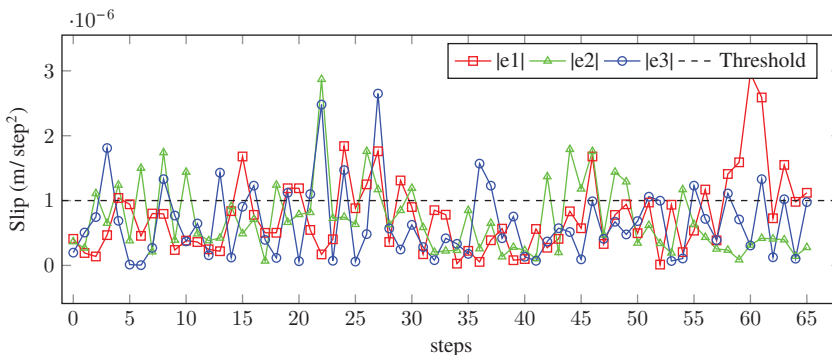


Figure 9. Slip and threshold visualization for the top three experiences with the highest amount of slip (e1, e2, e3 represent examples 1, 2, 3).

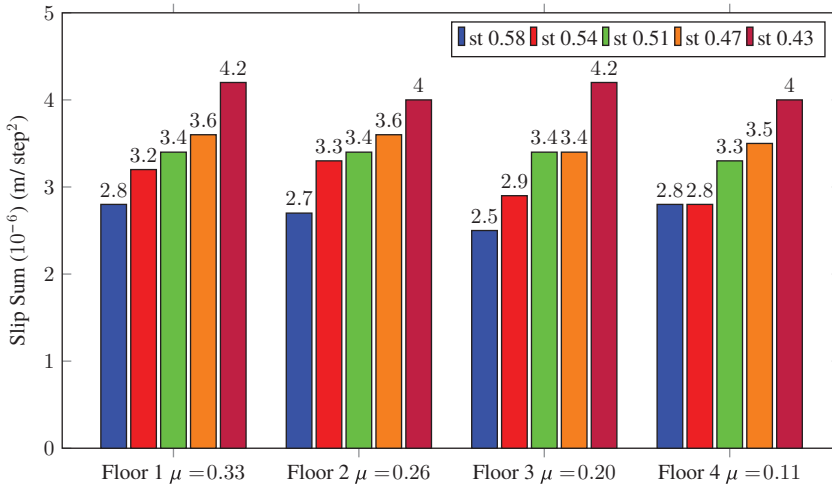


Figure 10. Comparison of slip sum for different step frequencies and slippery floors, where 'st' denotes the step time in seconds for a robot's stride.

The overall results of the experiment are presented in Fig. 10.

The study indicates that a higher nominal step frequency, which corresponds to the robot taking longer time durations per step, results in more favorable outcomes in terms of reduced slipping on all types of floors. Moreover, we observed a proportional relationship between step frequency and outcomes, with lower frequencies resulting in higher slip sums and vice versa. These findings demonstrate that, similar to humans, robots must adjust their gait to maintain stability on slippery surfaces. In our study, this involved reducing the step frequency to minimize the risk of slipping or falling.

Given that a specific step frequency for each floor was not identified, this study fell short of our goal of real-time adaptation of the controller to optimize walking on diverse surfaces. To address this limitation, we conducted a follow-up experiment focused on the EYC controller. The results of this study provided valuable insights into optimizing the controller's parameters to improve the robot's stability and mobility on various floors.

5. Results and discussions

This section unveils the outcomes and analyses stemming from an experiment designed to fine-tune parameters for the PID controller, addressing the unique characteristics of each floor. This experiment was influenced by the insights gained from the preliminary trials, which showed inconclusive results. To assess the upcoming experiment, performance metrics are defined, specifically, the time and energy spent on the task, and the stability of the robot during these tasks. The experiments to obtain the optimized PID parameters that allow the robot to achieve the best performance on each floor individually to improve the overall experience that is when the robot walks on the four different floors are presented. Similar to the preliminary experiment, this one also utilizes the same introduced floors.

5.1. EYC adaptive controller experiments

For the EYC controller to adapt the humanoid's locomotion based on its orientation, the PID parameters that would yield the best results were investigated. To determine the suitable PID parameters for stabilizing the humanoid robot orientation on slippery floors, a series of experiments on gazebo simulator was performed. Before describing the experiment in detail, it is essential to define the metrics used to evaluate the robot's performance on different surfaces.

5.1.1. Performance metrics

In order to assess and compare the performance of the controller and define the best parameters, some metrics are proposed: task time duration (T_D), stability (S_{idx}), and energy consumption (E_C). These metrics have been established and utilized in previous literature, as evidenced by ref. [32].

The task time duration refers to the time necessary for the humanoid robot to complete a given task, such as walking a certain distance on a slippery surface. This metric allows us to compare the efficiency of different control strategies in terms of completing tasks quickly and accurately.

The stability of the robot was measured using the root mean square error (RMSE) of the Center of Mass (CoM) and Zero Moment Point (ZMP) trajectories in relation to ideal straight line trajectories, where, for a generic quantity A :

$$RMSE(A) = \frac{1}{N} \sum_i \sqrt{(A_i^{real} - A_i^{plan})^2}. \tag{16}$$

Each component of RMSE, including CoM in x and y directions and ZMP in x and y directions, was given equal weight, accounting for $\frac{1}{4}$ of the total weight in the calculation of RMSE. A lower RMSE indicates better stability, as the robot is better able to maintain its balance and avoid falling on the slippery surface. The stability index is defined as:

$$S_{idx} = \frac{1}{4} (RMSE_{CoM_x} + RMSE_{CoM_y} + RMSE_{ZMP_x} + RMSE_{ZMP_y}). \tag{17}$$

Finally, to evaluate the energy balance of the control strategies, the energy consumption of the different experiments was compared using a simplified metric. Given the relatively small and lightweight nature of the NAO humanoid robot, the contribution of friction and other energy losses was negligible in comparison to the energy consumed by the robot’s actuators. Therefore, for simplicity and practicality, the energy consumption was approximated by integrating, for all time steps ($k = 0, \dots, N_T$), the product of the joint torque (τ_i) with the angular velocity (ω_i) for all the robot joints ($i = 1, \dots, N_J$), during each Δt . This allowed us to focus on the energy consumed by the joints as the main sources of energy consumption for the robot. The energy consumption is defined as:

$$E_C = \sum_{k=0}^{N_T} \sum_{i=1}^{N_J} \tau_i(k)\omega_i(k)\Delta t. \tag{18}$$

This simplified metric enabled us to compare the performance of the different control strategies and identify the most energy-efficient approach for the tasks at hand, while disregarding other factors, such as robot size, design, environment, and power source that would have contributed only marginally to the overall energy budget.

The performance of the implemented digital PID controller on four different slippery floors was evaluated using a combination of the three metrics, as shown in Eq. (19). The system’s performance decreases as the value of M increases.

$$M = \frac{1}{3} \left(\frac{T_D}{T_{D_{max}}} + \frac{E_C}{E_{C_{max}}} + \frac{S_{idx}}{S_{idx_{max}}} \right), \tag{19}$$

where $T_{D_{max}}$, $E_{C_{max}}$, and $S_{idx_{max}}$ correspond to the maximum values obtained across all the experiments.

The equation calculates the overall performance metric M , which is a weighted sum of normalized duration, energy, and the average RMSE of the CoM and ZMP positions over both axes. The normalization was performed using the maximum value of each metric to respectively scale them to a common range.

5.1.2. Digital PID controller optimization

To adjust the PID parameters, the initial setup was manually tuned to establish a baseline for comparison. Data was then collected on the robot’s performance on four different slippery surfaces, systematically varying the Proportional (P), Integral (I), and Derivative (D) gains.

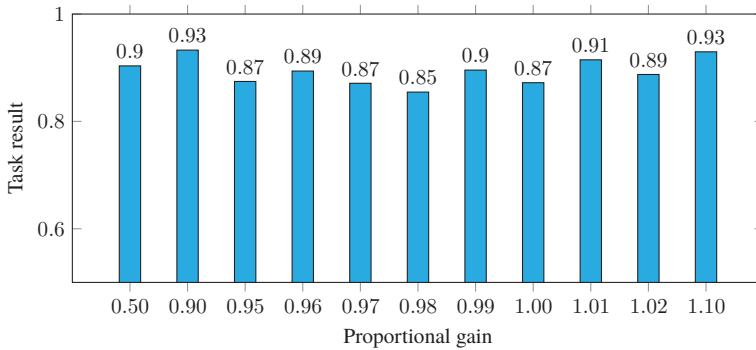


Figure 11. Experiment proportional gain impact on the task performance.

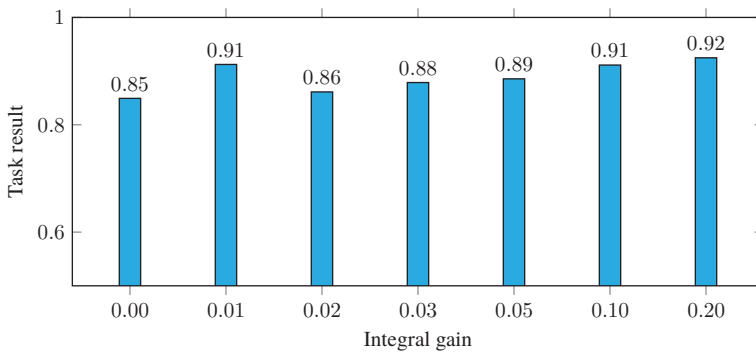


Figure 12. Impact of integral gain on task performance.

For refining the PID parameters, a systematic trial-and-error approach was employed. Starting with fixed Integral and Derivative values, iterative adjustments were made to the Proportional parameter. Once the optimal P value was identified, the Proportional and Derivative gains were fixed, and the Integral parameter underwent fine-tuning. Subsequently, the Proportional and Integral gains were set, and the Derivative gain was adjusted until optimal performance was achieved, assessed through metrics defined above for each walking experience.

This allowed for the evaluation of the impact of each parameter on the robot's stability and mobility and identification of the optimal combination of parameters for each surface. By using this approach, the PID controller's parameters were fine-tuned to enhance the robot's performance on a variety of surfaces, ensuring optimal stability and mobility.

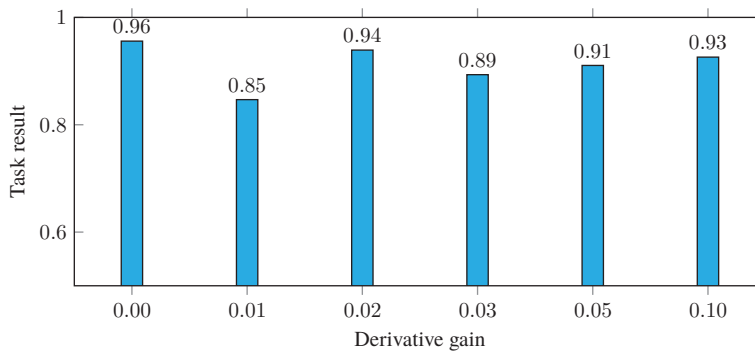
After each experiment, the results were evaluated using the defined metrics. Due to the complexity of the model, an analytical approach was not feasible, and an empirical approach was employed to identify the set of PID parameters that resulted in the best performance according to the metrics. Finally, the parameters that yielded the best performance across all experiments were selected for further analysis and comparison.

As will be demonstrated in this section, the optimized value of the integral parameter will be set to zero, effectively reducing the PID controller to a PD controller.

Optimized PID gains. To illustrate the results of our experiments, we present three graphics showcasing the performance of the controller on the first floor, which has a coefficient of friction of 0.33. Chart in Fig. 11 displays the performance task results for five repetitions for each variation in the proportional gain while fixing the integral and derivative parameters. During the experiment, the controller was used while the robot walked 2 m on the floor. Chart in Fig. 12 presents the performance results using the optimized proportional gain and the fixed derivative gain while varying the integral gain. Finally, Chart

Table II. Best PID gains for each floor.

	CoF			
	0.33	0.26	0.20	0.11
P	0.98	0.99	1.01	1.04
I	0.00	0.00	0.00	0.00
D	0.01	0.01	0.01	0.01

**Figure 13.** Experiment derivative gain impact on the task performance.

in of 0.33. Chart in Fig. 13 shows the results obtained with the optimized proportional and integral gains while varying the derivative gain.

The optimized digital PID gains were determined as $P = 0.98$, $I = 0.00$, and $D = 0.01$ for the first floor, using it as an example for our task. The same procedure was repeated for each floor, resulting in the corresponding optimized PID gains, as shown in Table II.

As the floor becomes increasingly slippery, we observe that the proportional gain (P) of the controller increases to enhance its responsiveness. This is because a more slippery surface tends to induce slower responses from the system to changes in orientation, necessitating a higher P value to achieve the desired level of control. Our empirical analysis revealed that the optimal integral term for this controller is $I = 0$, indicating that the system is able to compensate for any steady-state error without requiring the integral term. Furthermore, the derivative gain obtained was $D = 0.01$ for all experiments, suggesting that the current level of derivative gain is appropriate and does not require adjustments in response to changes in floor slipperiness.

Performance evaluation: variable vs. fixed PD gains. To demonstrate the effectiveness of the approach, a series of experiments were conducted in which the humanoid robot walked 2 m on four different slippery floors, each 50 cm long, and repeated the process five times. The performance of using a fixed-gain PD controller with the best gains for the first floor was compared against a variable-gain PD controller with optimized gains for each floor that the robot walked on. The results presented in Table III show that implementing the best PD gains for each floor led to a significant improvement in the robot's overall performance. On average, the robot achieved an approximately 10% increase in performance, with the most noticeable improvements observed on the last two floors, which had lower friction. The primary factor contributing to the improvement observed with the variable PD control method was the robot's resulting enhanced stability performance. Moreover, using the variable PD also resulted in lower energy consumption during task execution. However, the difference in task completion time between the two PD control methods was insignificant, with a slightly better performance observed for the fixed PD. Overall, when considering the three metrics with equal weight, the impact of the variable PD control method is evident, particularly on floors with reduced friction.

Table III. Experiment results by floor and overall experience, comparing the performance of the robot using optimized PD gains for each floor versus using the same gains for all floors.

	Variable PD per floor	Fixed PD on all floors
M floor 1	0.9997	0.9999
M floor 2	0.9773	0.9996
M floor 3	0.7568	0.9998
M floor 4	0.8716	0.9987
M experience	0.9013	0.9995

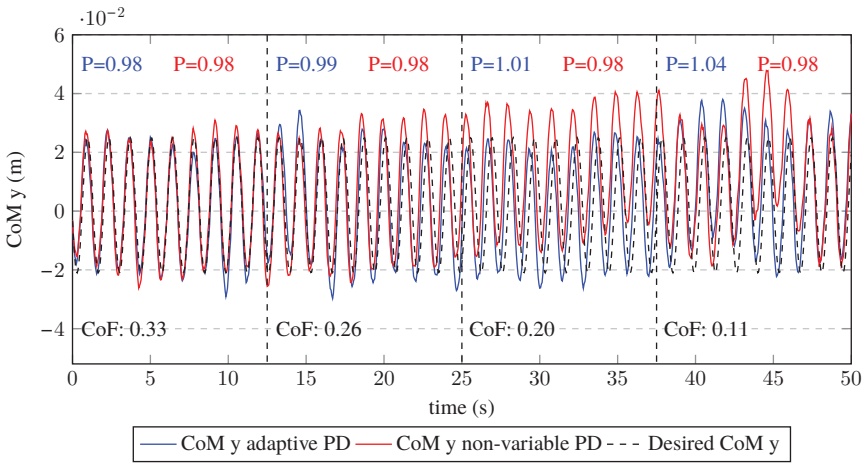


Figure 14. Comparison of absolute center of mass position in the Y-axis using an adaptive PD controller vs non-variable PD.

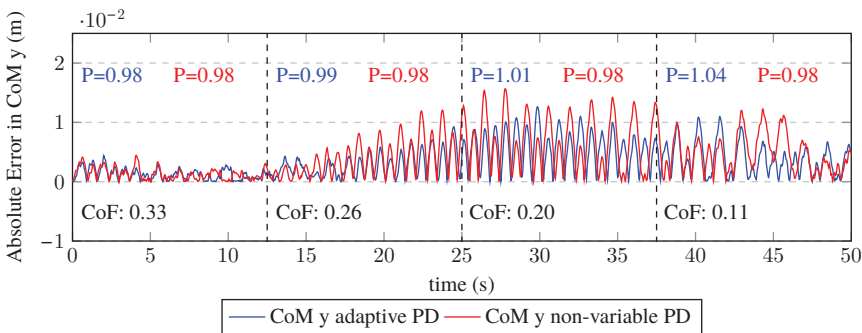


Figure 15. Comparison of absolute errors in CoM y using adaptive PD and non-variable PD controllers against the desired CoM y.

Figure 14 shows an example of one of the experiments CoM Y to illustrate the stability using a fixed-gain PD versus an adaptive-gain PD, and Fig. 15 showcases the absolute CoM Y errors, offering a clearer illustration of the distinctions.

From the graph in Fig. 14, it can be observed that, initially, the pattern is similar for both the adaptive and fixed PD controllers. This similarity is also evident in Fig. 15, where the errors exhibit a comparable magnitude. This can be attributed to the floors being less slippery and the proportional value being the

same for both controllers in the beginning. However, as the experiment progresses to increasingly slippery floors, the adaptive PD controller demonstrates a more effective response to the slippage induced by the lower friction of these surfaces. This is evident in Fig. 15, where the non-variable PD controller exhibits a larger overall error. The controller's adaptability to the floor on which the robot is walking is attributed to our prior research, which involved the development of an ITShoe for measuring ground reaction force data, and the implementation of a long-short-term memory (LSTM) network to identify the various floor types.

It is important to note that, for the defined performance metric, a lower value indicates better performance. In other words, achieving a smaller task time, lower energy consumption, and less stability error are desired outcomes.

This demonstrates the versatility and effectiveness of the controller used in this work to adapt to different slippery floor conditions.

The experimental results presented here are illustrated in a video available on YouTube,² which provides a better understanding of the problem and the observed behaviors.

6. Conclusion

In this study, we have demonstrated the versatility and potential of the controller developed in ref. [19] by successfully adapting and implementing it to control our humanoid robot. Our findings highlight that the integration of an EYC has significantly enhanced the overall performance of the robot while walking on slippery floors. This improvement was particularly evident when the robot was exposed to different types of slippery surfaces, effectively mitigating the challenges posed by the varying conditions. Furthermore, the EYC has also successfully rectified the non-linear paths resulting from the robot's unconventional building technique, contributing to a more stable and controlled gait. It is worth noting that we assumed the robot's ability to identify the type of floor it is walking on, as this issue was resolved in a prior study [4] using the ITShoe and a LSTM. As part of future work, the plan is to conduct tests on the physical humanoid robot and make necessary adjustments to seamlessly integrate floor detection with the controller adaptation. This would involve transitioning from robot operating system) to YARP and translating the forces measured by the ITShoe to ankle forces and torques for the robot.

Acknowledgements. The Fundação para a Ciência e Tecnologia (FCT) is gratefully acknowledged for funding this work with the grant SFRH/BD/136680/2018.

Author contributions. All authors contributed equally to this work. All authors read and approved the final manuscript.

Financial support. The Fundação para a Ciência e Tecnologia (FCT) is gratefully acknowledged for funding this work with the grant SFRH/BD/136680/2018.

Competing interests. The authors declare no competing interests exist.

Ethical approval. Not applicable.

References

- [1] J. Ding, Y. Wang, M. Yang and X. Xiao, "Walking stabilization control for humanoid robots on unknown slope based on walking sequences adjustment," *J. Intell. Robot. Syst.* **90**(3), 323–338 (2018). doi: [10.1007/s10846-017-0685-4](https://doi.org/10.1007/s10846-017-0685-4).
- [2] J.-H. Kim, J.-Y. Kim and J.-H. Oh, "Adjustment of home posture of biped humanoid robot using sensory feedback control," *J. Intell. Robot. Syst.* **51**(4), 421–438 (2008). doi: [10.1007/s10846-007-9195-0](https://doi.org/10.1007/s10846-007-9195-0).
- [3] L. Almeida, V. Santos and J. P. Ferreira, "Learning-based analysis of a new wearable 3D force system data to classify the underlying surface of a walking robot," *Int. J. HR* **17**(03), 2050011 (2020). doi: [10.1142/S0219843620500115](https://doi.org/10.1142/S0219843620500115).

²<https://youtu.be/aWqmWoWTbv8>

- [4] L. Almeida, V. Santos and J. P. Ferreira, “Real-time LSTM-RNN classification of floors with different friction coefficients for a walking humanoid robot wearing a 3D force system,” *IEEE Sens. J.* **21**(24), 27801–27809 (2021). doi: [10.1109/JSEN.2021.3124854](https://doi.org/10.1109/JSEN.2021.3124854).
- [5] M. Vukobratović and J. Stepanenko, “On the stability of anthropomorphic systems,” *Math. Biosci.* **15**(1), 1–37 (1972). doi: [10.1016/0025-5564\(72\)90061-2](https://doi.org/10.1016/0025-5564(72)90061-2).
- [6] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi and H. Hirukawa, “The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation,” **In: Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No.01CH37180)** (2001). doi: [10.1109/IROS.2001.973365](https://doi.org/10.1109/IROS.2001.973365).
- [7] J. Engelsberger, C. Ott and A. O. Albu-Schäffer, “Three-Dimensional Bipedal Walking Control Using Divergent Component of Motion,” **In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems** (2013). doi: [10.1109/TRO.2015.2405592](https://doi.org/10.1109/TRO.2015.2405592).
- [8] T. Koolen, T. de Boer, J. Rebula, A. Goswami and J. E. Pratt, “Capturability-based analysis and control of legged locomotion, part 1: Theory and application on three simple gait models,” *Int. J. Robot. Res.* **31**(9), 1094–1113 (2012). doi: [10.1177/0278364912452673](https://doi.org/10.1177/0278364912452673).
- [9] Y. Xie, C. Gao, S. Zhu, X. Yan, L. Kong, A. Xie, J. Gu and D. Zhang, “Gait optimization and energy-based stability for biped locomotion using large-scale programming,” *Robotica* **41**(7), 2207–2227 (2023). doi: [10.1017/S0263574723000413](https://doi.org/10.1017/S0263574723000413).
- [10] M. Jalaeian-F, M. Fateh and M. Rahimiyan, “Bi-level adaptive computed-current impedance controller for electrically driven robots,” *Robotica* **39**(2), 200–216 (2021). doi: [10.1017/S0263574720000314](https://doi.org/10.1017/S0263574720000314).
- [11] M. Murooka, K. Chappellet, A. Tanguy, M. Benallegue, I. Kumagai, M. Morisawa, F. Kanehiro and A. Kheddar, “Humanoid loco-manipulations pattern generation and stabilization control,” *IEEE Robot. Autom. Lett.* **6**(3), 5597–5604 (2021). doi: [10.1109/LRA.2021.3077858](https://doi.org/10.1109/LRA.2021.3077858).
- [12] G. Mesesan, J. Engelsberger, C. Ott and A. Albu-Schäffer, “Convex properties of center-of-mass trajectories for locomotion based on divergent component of motion,” *IEEE Robot. Autom. Lett.* **3**(4), 3449–3456 (2018). doi: [10.1109/LRA.2018.2853557](https://doi.org/10.1109/LRA.2018.2853557).
- [13] M. Shafiee, G. Romualdi, S. Daffarra, F. J. A. Chavez and D. Pucci, “Online DCM Trajectory Generation for Push Recovery of Torque-Controlled Humanoid Robots,” **In: 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)** (2019) pp. 671–678. doi: [10.1109/Humanoids43949.2019.9034996](https://doi.org/10.1109/Humanoids43949.2019.9034996).
- [14] G. Romualdi, S. Daffarra, Y. Hu, P. Ramadoss, F. J. A. Chavez, S. Traversaro and D. Pucci, “A benchmarking of DCM-based architectures for position, velocity and torque-controlled humanoid robots,” *Int. J. HR* **17**(01), 1950034 (2020). doi: [10.1142/S0219843619500348](https://doi.org/10.1142/S0219843619500348).
- [15] R. Griffin, S. McCrory, S. Bertrand, D. Calvert, I. Lee, P. Neuhaus, D. Stephen, J. Jasper, S. Karumanchi, A. Kourchians, B. Emanuel, E. Holmes, R. Hegeman, J. Pusey and J. Pratt, “Quadrupedal walking over complex terrain with a quasi-direct drive actuated robot,” *Field Robot.* **2**(1), 356–384 (2022). doi: [10.55417/fr.2022013](https://doi.org/10.55417/fr.2022013).
- [16] M. Wagner. Applying the Divergent Component of Motion Method for Quadrupedal Locomotion to a Robot with Series Elastics Actuators. *Ph.D. thesis* (DLR-RM, 2021).
- [17] J. K. Mehr, M. Sharifi, V. K. Mushahwar and M. Tavakoli, “Intelligent locomotion planning with enhanced postural stability for lower-limb exoskeletons,” *IEEE Robot. Autom. Lett.* **6**(4), 7588–7595 (2021). doi: [10.1109/LRA.2021.3098915](https://doi.org/10.1109/LRA.2021.3098915).
- [18] D. Xu, P. Huang, Z. Li and Y. Feng, “DMP-Based Motion Generation for a Walking Exoskeleton Robot Using Divergent Component of Motion,” **In: 2022 International Conference on Advanced Robotics and Mechatronics (ICARM)** (2022) pp. 232–237. doi: [10.1109/ICARM54641.2022.9959591](https://doi.org/10.1109/ICARM54641.2022.9959591).
- [19] G. Romualdi, S. Daffarra, Y. Hu and D. Pucci, “A Benchmarking of DCM Based Architectures for Position and Velocity Controlled Walking of Humanoid Robots,” **In: IEEE-RAS 18th International Conference on Humanoid Robots** (2018) pp. 1–9. doi: [10.1109/HUMANOIDS.2018.8625025](https://doi.org/10.1109/HUMANOIDS.2018.8625025).
- [20] Y. Choi, D. Kim, Y. Oh and B.-J. You, “Posture/walking control for humanoid robot based on kinematic resolution of CoM Jacobian with embedded motion,” *IEEE Trans. Robot.* **23**(6), 1285–1293 (2007). doi: [10.1109/TRO.2007.904907](https://doi.org/10.1109/TRO.2007.904907).
- [21] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. D. Mombaur and M. Diehl, “Online walking motion generation with automatic footstep placement,” *Adv. Robot.* **24**(5–6), 719–737 (2010). doi: [10.1163/016918610X493552](https://doi.org/10.1163/016918610X493552).
- [22] Q. Zhou, Z. Yu, S. Zhang, X. Chen, M. Qin, W. Zhang and Q. Huang, “Simultaneous prevention of rotational and translational slip for a humanoid robot,” *Appl. Sci.* **8**(9), 1554 (2018). doi: [10.3390/app8091554](https://doi.org/10.3390/app8091554).
- [23] M. Brandão, K. Hashimoto, J. Santos-Victor and A. Takanishi, “Gait Planning for Biped Locomotion on Slippery Terrain,” **In: 2014 IEEE-RAS International Conference on Humanoid Robots**, Madrid, Spain (2014) pp. 303–308. doi: [10.1109/HUMANOIDS.2014.7041376](https://doi.org/10.1109/HUMANOIDS.2014.7041376).
- [24] X. T. Nguyen, D. H. Nguyen, H. H. Nguyen, N. P. Tong, T. P. Nguyen and T. T. Nguyen, “Balancing Walking Gait for Small Size Humanoid Robot by Using Movable Mass,” **In: 2020 International Conference on Advanced Mechatronic Systems (ICAMechS)** (2020) pp. 13–16. doi: [10.1109/ICAMechS49982.2020.9310109](https://doi.org/10.1109/ICAMechS49982.2020.9310109).
- [25] M. Bestmann and J. Zhang, “Bipedal Walking on Humanoid Robots through Parameter Optimization,” **In: RoboCup Symposium** (2022). doi: [10.1007/978-3-031-28469](https://doi.org/10.1007/978-3-031-28469).
- [26] Z. Xu, R. Li, B. Zhong and Y. Peng, “Design of a humanoid dance robot for dancing Baduanjin,” *Proceedings International Conference of Artificial Life and Robotics*, vol. 27 (2022), pp. 938–945. doi: [10.5954/icarob.2022.os9-8](https://doi.org/10.5954/icarob.2022.os9-8).
- [27] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi and H. Hirukawa, “The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation,” **In: Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No. 01CH37180)**, vol. 1 (2001) pp. 239–246. doi: [10.1109/IROS.2001.973365](https://doi.org/10.1109/IROS.2001.973365).

- [28] L. Almeida, V. Santos and F. Silva, “A Novel Wireless Instrumented Shoe for Ground Reaction Forces Analysis in Humanoids,” **In: 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)** (2018) pp. 36–41. doi: [10.1109/ICARSC.2018.8374157](https://doi.org/10.1109/ICARSC.2018.8374157).
- [29] Y. H. Lee, Y. H. Lee, H. Lee, H. Kang, J. H. Lee, J. M. Park, Y. B. Kim, H. Moon, J. C. Koo and H. R. Choi, “Whole-body control and angular momentum regulation using torque sensors for quadrupedal robots,” *J. Intell. Robot. Syst.* **102**(3), 66 (2021). doi: [10.1007/s10846-021-01418-x](https://doi.org/10.1007/s10846-021-01418-x).
- [30] K. J. Åström and T. Häggglund, “The future of PID control,” *Control Eng. Pract.* **9**(11), 1163–1175 (2001). doi: [10.1016/S0967-0661\(01\)00062-4](https://doi.org/10.1016/S0967-0661(01)00062-4).
- [31] S. J. Julier and J. K. Uhlmann, “New Extension of the Kalman Filter to Nonlinear Systems,” **In: Signal Processing, Sensor Fusion, and Target Recognition VI** (I. Kadar, ed.), vol. 3068 (SPIE, 1997). doi: [10.1117/12.280797](https://doi.org/10.1117/12.280797).
- [32] F. Aller, D. Pinto-Fernandez, D. Torricelli, J. L. Pons and K. Mombaur, “From the state of the art of assessment metrics toward novel concepts for humanoid robot locomotion benchmarking,” *IEEE Robot. Autom. Lett.* **5**(2), 914–920 (2019). doi: [10.1109/LRA.2019.2952291](https://doi.org/10.1109/LRA.2019.2952291).