Balancing on a Rolling Contact

Federico Allione[1,2], Roy Featherstone[3], Patrick M. Wensing[4], and Darwin Caldwell[1]

***Abstract*—This paper presents a controller for robots that balance in a vertical plane on a rolling contact on a flat horizontal surface. It is an extension of Featherstone's balance controller to the case of robots that balance on rounded feet or wheels. Simulation results demonstrate the ability of the new controller to balance an inverted double pendulum on a rolling contact and to balance a Segway-like wheeled robot and make it follow a motion command signal. Experimental validation is provided on an underactuated inverted double pendulum robot.**

***Index Terms*—Body Balancing, Dynamics, Motion Control.**

I. Introduction

THIS paper is part of a larger project called Skippy [1], [2], [3], [4], which aims to demonstrate high performance in physical activities such as hopping and balancing. This work presents an extension to Featherstone's controller for robots that balance on a point in a vertical plane [5]. It consists of replacing the point-foot assumption in Featherstone's controller with a circular-foot assumption so that the new controller can balance robots that move on a rolling contact. The significance of this work is that it provides a single control law that is applicable to any robot that balances in a vertical plane on a point, a rounded foot or wheels. The extension is compatible with the technique in [6] to combine balancing with absolute motion, which means that it can support loco-manipulation tasks involving balancing on wheels. A further contribution of this paper is that it presents experimental results in which the new controller balances the robot shown in Fig. 1, thereby demonstrating that it works in practice as well as in theory.

The new controller is described here for the case of a rolling inverted double pendulum, which includes Segway-like robots and other similar wheeled mobile robots as a special case. The procedure for adapting the controller to work on a general planar robot is the same as that described in [5, §6], and is therefore not repeated here.

The rest of this paper is organised as follows. First, Section II presents a review of the relevant existing literature on robotic balancing. Then Section III describes the robot model used in this paper, which is an inverted double pendulum in which the part of the robot that makes contact with the ground is circular. Section IV demonstrates the need for a rolling-contact extension to [5] by showing the tracking error that results from making a point-foot assumption when the actual foot is round. Then Section V presents the extended controller, and Section VI proves its effectiveness in simulation. Section VII explains why the tracking performance decreases as the radius of the rolling contact increases. Then Section VIII shows the results of physical experiments; and Section IX demonstrates the effectiveness of the controller on Segway-like robots. An accompanying video presents these last two results.

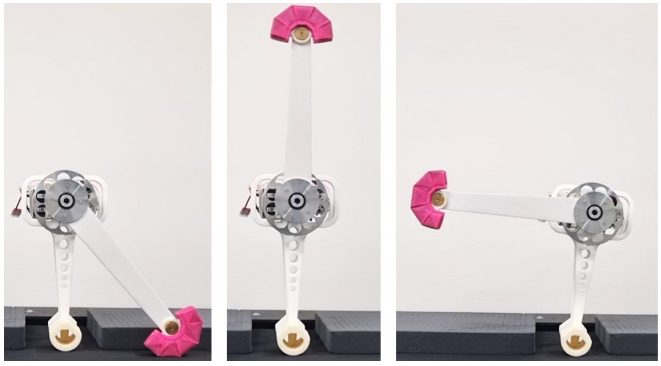

Fig. 1. Balancing machine used for the experiments. On the left, the robot is in a resting position, not operating. In the central picture, it is actively balancing with the upper link upright. In the right picture, it is actively balancing with the upper link with an angle of $\pi/2$ with respect to the lower body.

II. Background

The most well-known balancing machines, such as Cubli [7], Pendubot [8], or Acrobot [9], balance on fixed single contact points—typically a shaft in the case of Pendubot and Acrobot, or a cube edge in Cubli—and are not designed to move while balancing in space (although Cubli can "walk" by combining consecutive balancing and controlled falling).

Mobile legged robots, such as most quadruped robots, typically have ball feet (e.g., HyQ [10], Unitree Go1 [11], Spot [12], MIT Mini Cheetah [13]). While these feet are small relative to the overall size of the robot and can be a good approximation of point feet, these robots are not designed to balance on two feet. It has been shown that the balance controller proposed in [5] can be applied to real quadrupeds such as HyQ [14]. However, balancing proved to be challenging, and the robot was only able to statically balance on two feet without moving the other joints not involved in balancing. Another example of balance for quadruped robots can be found in [15], where the authors balance the MIT Mini Cheetah on

Manuscript received: June, 8, 2023; Revised August, 7, 2023; Accepted October, 3, 2023.

This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. This work was carried out at the Advanced Robotics department, Istituto Italiano di Tecnologia, Genoa, Italy

[1]Federico Allione and Darwin Caldwell are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy. name.surname@iit.it

[2]Federico Allione is with the Department of Informatics, Bioengineering, Robotics and Systems Engineering (DIBRIS). University of Genoa, Italy

[3]Roy Featherstone roy.featherstone@ieee.org

[4]Patrick Wensing is with the Department of Aerospace and Mechanical Engineering at the University of Notre Dame pwensing@nd.edu

Digital Object Identifier (DOI): see top of this page.

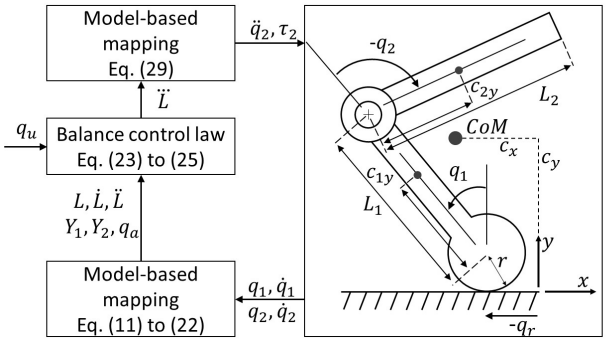

Fig. 2. Control architecture of the rolling double pendulum.

two feet using a variational-based linearization technique to solve an unconstrained linear quadratic regulator problem that considers torso orientation.

Another family of quadruped robots combines wheeled locomotion on structured terrain with legged locomotion on unstructured or uneven surfaces. Examples include CENTAURO [16] and TowrISIR [17], the latter showing the same hydraulic actuation system as HyQ2Max [18]. In this type of robot, the foot is replaced by actuated wheels, which allows some robots (e.g., ANYmal with wheels [19]) to balance on their wheeled hind legs. The authors use a whole-body model predictive controller to achieve highly dynamic motions, and in [20], they use reinforcement learning techniques to make the robot balance on its wheeled hind legs and move in space. In this way, the robot behaves like an inverted pendulum, similar to the famous Segway [21], which has been the objective of numerous control strategies such as the linear-quadratic regulator [22], [23], intrinsic geometric proportional-integral-derivative (PID) controller [24] or optimal control [25]. The Ballbot [26] is an example of an inverted pendulum balancing in 3D, being a stick balancing on a ball. The ball is actuated using an inverse mouse-ball drive, and the balance controller consists of two independent PID controllers, one for each of the vertical planes [27]. In [28], the authors use a sliding-mode controller with a single-loop control system to track and balance the Ballbot.

In the examples above that use an explicit balance controller, it is robot specific, often including the complete robot dynamics in the balance control loop. Featherstone's controller instead controls only the dynamics of balancing. The advantage of this approach is that the physics of balancing on a point in a vertical plane is the same for all robots, so a single control law serves to balance all such robots. The new balance controller modifies this control law so that it can balance robots on a rolling contact as well as a fixed point.

III. Robot Model

The theory presented in this paper is applicable to general planar robots that balance on a rolling contact on flat, horizontal ground. However, it will be developed for the special case of an inverted double pendulum. The extension to the general case follows the same procedure as described in [5, §6].

TABLE I
Model parameters. Considering Body i: m is its mass, c_x and c_y are the coordinates of its centre of mass, L is its length, and I is its inertia about its centre of mass.

Body	m [kg]	c_x [m]	c_y [m]	L [m]	I [kg m^2]
r	0	0	0	r	0
1	1.28	0	0.25-r	0.28-r	0.0099
2	0.705	0	0.114	0.3	0.0121

The model we shall use is shown in Fig. 2, and we shall call it a rolling double pendulum. It consists of an upper link (Body 2) which is connected to a lower link (Body 1) via an actuated revolute joint (Joint 2) with the joint variable q_2. The lower link rolls without slipping over a supporting surface (the ground), which is assumed to be flat and horizontal; so the model, and therefore also the control system, ignores the possibility of slipping. Previous work has shown that slipping does occur in practice, but causes only minor disturbances to the balance controller [3]. The portion of the lower link that makes contact with the ground is a circle of radius r. The lower link may therefore be a rounded foot, as shown in Fig. 2; but it could also be a wheel, and this will be discussed in Section IX. The special case $r = 0$ is allowed, and in this case, the model simplifies to an inverted double pendulum on a point foot.

The rolling contact is modelled as a combination of a revolute joint located at the centre of the circle and a prismatic joint in the horizontal direction. The former has a joint variable q_1, which is chosen as the independent variable of the rolling contact, and which is measured from the vertical as shown in Fig. 2. The latter has a joint variable q_r, which is a dependent variable constrained to have the value $q_\mathrm{r} = -rq_1$. A massless body (Body r) is inserted between these two joints to make a complete three-link kinematic chain.

Table I presents the numeric values of the kinematic and inertia parameters of the robot used in the simulation experiment in Section VI, considering a range of radii. The physical robot used for the physical experiment in Section VIII has $r = 0.03\,\mathrm{m}$ and its kinematic and inertia parameters are shown in Table III.

We define $q = [q_\mathrm{r}\ q_1\ q_2]^\mathrm{T}$ to be the vector of all joint variables, and $\tau = [0\,0\,\tau_2]^\mathrm{T}$ to be the corresponding vector of joint force variables. τ_2 is the torque at the actuated joint. We also define $\bar{q} = [q_1\ q_2]^\mathrm{T}$ to be the vector of independent joint variables, and $u = [0\,\tau_2]^\mathrm{T}$ to be the corresponding vector of joint force variables. The kinematic constraint can be written

$$q = G\bar{q} \qquad \text{and} \qquad u = G^\mathrm{T}\tau \tag{1}$$

where

$$G = \begin{bmatrix} -r & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2}$$

As G is a constant, we also have

$$\dot{q} = G\dot{\bar{q}} \qquad \text{and} \qquad \ddot{q} = G\ddot{\bar{q}}. \tag{3}$$

The equation of motion of the three-link chain is

$$H\ddot{q} + C = \tau \tag{4}$$

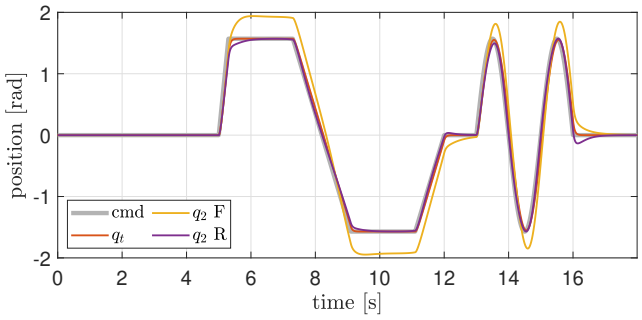

Fig. 3. Tracking of position variable q_2 for a rolling double pendulum with $r = 2\,\mathrm{cm}$. cmd is the desired trajectory, q_t is the theoretical response, q_2 F is the tracking of the balance controller assuming a fixed contact point, and q_2 R is the tracking of the balance controller assuming a rolling contact point.

where H is the joint-space inertia matrix and C is the bias vector containing Coriolis, centrifugal and gravitational terms. Both H and C can be calculated using standard methods [29]. Applying the kinematic constraint produces the following equation of motion for the robot:

$$H_G\ddot{\tilde{q}} + C_G = u \tag{5}$$

where

$$H_G = G^{\mathrm{T}} H G \qquad \text{and} \qquad C_G = G^{\mathrm{T}} C\,. \tag{6}$$

IV. Tracking Error

This section demonstrates the need for a rolling-contact extension to the balance controller in [5] by showing in simulation that tracking accuracy declines substantially if a round foot is approximated with a point foot. The simulation was performed in Simulink R2020b using the integrator ode45 with relative tolerance set to 10^{-6} and other parameters at their default values.

The results are shown in Fig. 3. In this graph, 'cmd' is the motion command signal for the actuated joint, and the job of the balance controller is to make this joint follow the command signal while simultaneously maintaining the robot's balance. The signal q_{t} is the theoretical response as explained in [5]. It is the response that the balance controller is programmed to produce, so the tracking error is defined to be the difference between the theoretical and actual response. The two signals q_2 F and q_2 R are the actual responses of the point-foot and round-foot balance controllers, respectively. Although both controllers do a good job of maintaining the robot's balance, it can be seen that the tracking accuracy of the point-foot balance controller is substantially worse than that of the round-foot balance controller.

To obtain these results, the simulator models the dynamics using the parameters in Table I with $r = 2\,\mathrm{cm}$; the round-foot controller uses the same parameters; and the point-foot controller uses the same parameters but with $r = 0$.

V. New Balance Controller

The planar balance controller described in [5] and then experimentally demonstrated in [30] and [3] assumes the robot to be balancing on a single fixed point in 2D or a knife edge in 3D. In [30], the robot behaves as a reaction wheel pendulum whose base is fixed to the ground via a revolute joint, and the balancing point coincides with the rotation axis of the joint. In [3], instead, the robot is a double inverted pendulum which is not fixed to the ground. It has two contact points with the floor aligned with the upper link's rotation axis, which mimics a knife edge contact. In both cases, the controller assumes the contact point is fixed while the robot balances.

In contrast, the rolling double pendulum's contact point does move, and, as the results of Section IV have just shown, it is important that the balance controller takes this into account. We therefore proceed to develop an extension of the theory in [5] to take this movement into account. Let L be the angular momentum of the whole robot about a fixed point that coincides with the support point at the current instant. This definition takes into account the position of the rolling contact point but not its velocity. We do this in order to ignore a small-magnitude velocity term that would otherwise appear in (7) and its derivatives in (8) and (9) below. From elementary mechanics, $\dot{L}$ must equal the sum of the moments about the support point of each external force acting on the robot; but the only external force with a nonzero moment is gravity, and so we have

$$\dot{L} = -mg(c_x - q_{\mathrm{r}}) = -mg(c_x + rq_1) \tag{7}$$

$$\ddot{L} = -mg(\dot{c}_x - \dot{q}_{\mathrm{r}}) = -mg(\dot{c}_x + r\dot{q}_1) \tag{8}$$

$$\dddot{L} = -mg(\ddot{c}_x - \ddot{q}_{\mathrm{r}}) = -mg(\ddot{c}_x + r\ddot{q}_1) \tag{9}$$

where m is the mass of the robot and g is the magnitude of the gravitational acceleration; c_x, $\dot{c}_x$ and $\ddot{c}_x$ are the position, velocity and acceleration in the x direction of the centre of mass with respect to the reference frame fixed at the origin; and q_{r}, $\dot{q}_{\mathrm{r}}$ and $\ddot{q}_{\mathrm{r}}$ are the position, velocity and acceleration in the x direction of the contact point.

As the independent variable q_1 is an angle, and as the rolling contact is a rotation about the support point, we can say that the angular momentum of the constrained robot about the contact point is

$$L = p_1 = H_{G_{11}}\dot{q}_1 + H_{G_{12}}\dot{q}_2 \tag{10}$$

which follows from a special property of joint-space momentum that is proved in Appendix B of [5].

Following the same line of reasoning as in [5], both L and $\ddot{L}$ depend linearly on the robot's velocity, implying that $L = 0$ and $\ddot{L} = 0$ is equivalent to $\dot{q}_1 = \dot{q}_2 = 0$ except in special circumstances when the robot is physically unable to balance (see Section VII). Also, $\dot{L}$ is a constant multiple of $(c_x + rq_1)$. Therefore, as shown in [5] and [31], any controller that makes

$$L = 0\,,\ \dot{L} = 0 \text{ and } \ddot{L} = 0 \tag{11}$$

will make the robot balance but will not drive the actuated joint in the desired position.

To allow the robot to track a desired trajectory for the actuated joint, we need expressions for $\dot{c}_x$ and $\ddot{c}_x$ in terms of the joint velocity and acceleration variables. This can be obtained by adding an extra fictitious prismatic joint acting in the x direction between the joint 1 of the constrained robot

and the ground. The extra joint is called Joint 0 to preserve the numbering of the existing joints. The extra joint does not move, and therefore does not affect the dynamics. Its purpose is to increase the number of coefficients in the robot's equation of motion, which becomes

$$\begin{bmatrix} H_{G_{00}} & H_{G_{01}} & H_{G_{02}} \\ H_{G_{10}} & H_{G_{11}} & H_{G_{12}} \\ H_{G_{20}} & H_{G_{21}} & H_{G_{22}} \end{bmatrix} \begin{bmatrix} 0 \\ \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{G_0} \\ C_{G_1} \\ C_{G_2} \end{bmatrix} = \begin{bmatrix} u_0 \\ 0 \\ u_2 \end{bmatrix} . \quad (12)$$

The variable u_0 is the component of ground reaction force in the x direction. As this is the only external force acting on the robot in this direction, it follows that $u_0 = m\ddot{c}_x$. The special property of the joint-space momentum used before to define $p_1 = L$ can be used in this case to define p_0 as the linear momentum of the whole robot in the x direction, so

$$p_0 = m\dot{c}_x = H_{G_{01}}\dot{q}_1 + H_{G_{02}}\dot{q}_2 . \quad (13)$$

Combining (8) and (13) we get

$$\ddot{L} = -g(p_0 + mr\dot{q}_1) = -g(H_{G_{01}} + m\,r)\dot{q}_1 - gH_{G_{02}}\dot{q}_2 \quad (14)$$

and combining (12) and (9) we get

$$-\dddot{L}/g = u_0 + mr\ddot{q}_1 = (H_{G_{01}} + mr)\ddot{q}_1 + H_{G_{02}}\ddot{q}_2 + C_{G_0} \quad (15)$$

So, also in the case of rolling contact, there is an equation relating $\dddot{L}$ to the two independent joint accelerations and a pair of linear equations relating L and $\ddot{L}$ to the two independent joint velocities

$$\begin{bmatrix} L \\ \ddot{L} \end{bmatrix} = \begin{bmatrix} H_{G_{11}} & H_{G_{12}} \\ -g(H_{G_{01}} + mr) & -gH_{G_{02}} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} . \quad (16)$$

To simplify the notation, we introduce a new matrix called H_R, defined as

$$H_{R_{ij}} = \begin{cases} (H_{G_{ij}} + mr) & \text{if } (i,j) = (1,0) \text{ or } (i,j) = (0,1) \\ H_{G_{ij}} & \text{otherwise} . \end{cases} \quad (17)$$

Equation (15) becomes

$$-\dddot{L}/g = H_{R_{01}}\ddot{q}_1 + H_{R_{02}}\ddot{q}_2 + C_{G_0} \quad (18)$$

and (16) now reads

$$\begin{bmatrix} L \\ \ddot{L} \end{bmatrix} = \begin{bmatrix} H_{R_{11}} & H_{R_{12}} \\ -gH_{R_{01}} & -gH_{R_{02}} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} . \quad (19)$$

Solving this equation for $\dot{q}_2$ gives

$$\dot{q}_2 = Y_1 L + Y_2 \ddot{L} \quad (20)$$

where

$$Y_1 = \frac{H_{R_{01}}}{D} \quad , \quad Y_2 = \frac{H_{R_{11}}}{gD} \quad (21)$$

and

$$D = H_{R_{01}}H_{R_{12}} - H_{R_{11}}H_{R_{02}} . \quad (22)$$

The obtained equations are the same as those described in [5] with H_R in place of the original inertia matrix.

The main idea of the control strategy in [5] is to close a control loop around the plant shown in Fig. 4, which describes the exact dynamics of balancing for any planar robot that is balancing on a point that can be modelled as a passive revolute joint. Observe that the plant implements (20).

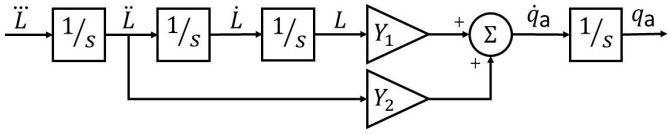

Fig. 4. Plant describing the physical process of balancing on a point in a vertical plane [5]. q_a is the variable of the joint used to balance, which in this case is Joint 2.

The objective of the balance controller is to calculate a value for $\dddot{L}$ so that q_a follows a desired command signal without losing balance. A suitable control law to accomplish this is

$$\dddot{L} = k_{dd}\ddot{L} + k_d\dot{L} + k_L L + k_q(q_\mathrm{a} - q_\mathrm{u}) , \quad (23)$$

where $q_\mathrm{a} = q_2$ is the only actuated joint and q_u is the input to the controller, and L, $\dot{L}$ and $\ddot{L}$ are defined respectively by (10), (7) and (8).

To determine the feedback gains, the controller first makes the assumption that $\dot{Y}_1 = \dot{Y}_2 = 0$, which makes the plant linear. The feedback gains are then obtained via pole placement as

$$\begin{aligned} k_{dd} &= -a_3 & k_d &= -a_2 + a_0 Y_2/Y_1 \\ k_L &= -a_1 & k_q &= -a_0/Y_1 , \end{aligned} \quad (24)$$

where

$$\begin{aligned} a_0 &= \lambda_1\lambda_2\lambda_3\lambda_4 \\ a_1 &= -\lambda_1\lambda_2\lambda_3 - \lambda_1\lambda_2\lambda_4 - \lambda_1\lambda_3\lambda_4 - \lambda_2\lambda_3\lambda_4 \\ a_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_1\lambda_4 + \lambda_2\lambda_3 + \lambda_2\lambda_4 + \lambda_3\lambda_4 \\ a_3 &= -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 , \end{aligned} \quad (25)$$

and $\lambda_1, \ldots, \lambda_4$ are the chosen values of the poles. With these gains, the closed-loop transfer function is

$$q_\mathrm{a}(s) = \frac{a_0(1 - T_\mathrm{c}^2 s^2)}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} q_\mathrm{u}(s) . \quad (26)$$

The quantity T_c appearing in the numerator is the robot's time constant of toppling, which is the rate at which the robot *starts* to fall when there is no movement in the actuated joint. It is a physical property of the robot and varies with configuration. In the experiments below, we chose the poles as follows: $\lambda_1 = -1/T_\mathrm{c}$, $\lambda_2 = \lambda_3 = -1/T_\mathrm{c}^*$, and $\lambda_4 = \beta$, where T_c^* is a constant value equal to T_c when $\bar{q} = 0$; and β is a constant value which determines the theoretical response of the balance controller transfer function, as described below.

The input to the controller, q_u, is computed from the filtered command signal, q_f, according to

$$q_\mathrm{u} = q_\mathrm{f} + \alpha_1\dot{q}_\mathrm{f} + \alpha_2\ddot{q}_\mathrm{f} , \quad (27)$$

where $q_\mathrm{f} = \mathrm{AF}(q_\mathrm{c})$, $\dot{q}_\mathrm{f} = \mathrm{AF}(\dot{q}_\mathrm{c})$ and $\ddot{q}_\mathrm{f} = \mathrm{AF}(\ddot{q}_\mathrm{c})$, and q_c, $\dot{q}_\mathrm{c}$ and $\ddot{q}_\mathrm{c}$ are the desired position, velocity and acceleration of the actuated joint. α_1 and α_2 are feedforward gains that introduce two zeros into the transfer function which cancel the poles λ_2 and λ_3. AF is an acausal filter consisting of a first-order low-pass filter with time constant T_f that runs backwards in time from a point sufficiently far in the future back to the present. Its transfer function in reverse time is $1/(1+T_\mathrm{f}s)$, which implies a transfer function in forward time of $1/(1-T_\mathrm{f}s)$. The physical effect of this filter is to reduce

the robot's tracking error by making it lean in anticipation of the balance disturbances that will be caused by following the motion command signal q_c. Although it would be enough to give the controller $3\,T_{\mathrm{c}}^*\,\mathrm{s}$ advance notice of the command signal [5], in all the experiments of this work, we calculated the complete signal q_{u} in advance.

Given the above considerations, setting $T_{\mathrm{f}} = T_{\mathrm{c}}^*$, and assuming that Y_1, Y_2 are constants [5] and $T_{\mathrm{c}} = T_{\mathrm{c}}^*$ for every robot's configuration, it can be shown that after the cancellation of poles and zeros, the complete transfer function from q_c to q_{a} of the system shown in Fig. 2 would be

$$q_{\mathrm{a}}(s) = \frac{1}{1 + s/(-\lambda_4)} q_c(s)\ . \tag{28}$$

We take this expression as the theoretical transfer function of the balance controller, and compare the actual response with the theoretical one in the experimental results reported below.

The controller's output must be either an acceleration or a torque for the actuated joint; that is, either $\ddot{q}_2$ or τ_2. Combining (12), (17) and (18) we obtain

$$\begin{bmatrix} 0 & H_{R_{01}} & H_{R_{02}} \\ 0 & H_{R_{11}} & H_{R_{12}} \\ -1 & H_{R_{21}} & H_{R_{22}} \end{bmatrix} \begin{bmatrix} u_2 \\ \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = \begin{bmatrix} -\dddot{L}/g - C_{G_0} \\ -C_{G_1} \\ -C_{G_2} \end{bmatrix} \tag{29}$$

which can be solved for $\ddot{q}_2$ and u_2; and $u_2 = \tau_2$ by definition.

VI. Simulation Experiments

This section reports the results of simulation experiments where the robot starts in a vertical position ($q_{\mathrm{r}} = q_1 = q_2 = 0$) and then tracks a desired trajectory for joint q_2 while balancing on a rolling contact. The robot model and the initial conditions are the same as Section IV. The controller has one pole set at $-1/T_{\mathrm{c}}$; two more poles and the two zeros are set to a constant value equal to $-1/T_{\mathrm{c}}^*$; and the fourth pole is set at $-20\,\mathrm{rad/s}$.

The experiment is performed every time with a different radius of the contact surface. The initially tested radii are 0, 2, 4, 6, 8, and 10 cm. Although the reference trajectory is the same for all the experiments, the filtered command signal produced by the acausal filter is specific for each experiment. This is because the time constant of the acausal filter $T_{\mathrm{f}} = T_{\mathrm{c}}^*$, and T_{c}^* varies with r. Some example values of T_{c}^* are reported in Table II.

Figure 5 reports the tracking position of joint q_2. The best performance is obtained when the radius is zero, meaning that the robot is balancing on a fixed point and not on a rolling contact. The tracking accuracy with $r = 0.02\,\mathrm{m}$ is almost as good as with $r = 0\,\mathrm{m}$ and then gradually decreases as r increases, until $r = 0.08\,\mathrm{m}$. The increment of the radius to $r = 0.1\,\mathrm{m}$ causes a large reduction in the tracking performance. The reason behind this behaviour is the topic of Section VII.

VII. Linear Velocity Gain

A robot's performance at balancing is limited by its physical ability to balance, which is a property of the robot itself, not the control system. Linear velocity gain, as defined in [32], provides a quantitative measure of a robot's physical ability to balance. It is defined as the ratio of the change in the horizontal velocity of the CoM to the change in velocity of the joint used to balance the robot when both changes are caused by an impulse at that joint. For the robot used in these experiments, the velocity gain is

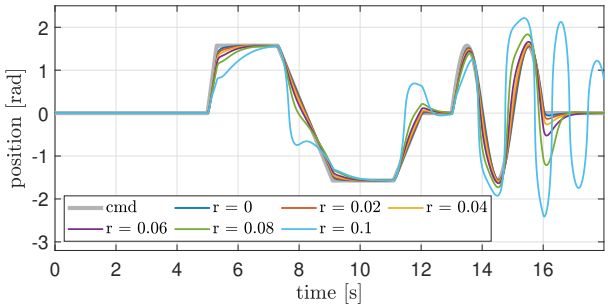

Fig. 5. Tracking position for joint q_2 with varying radius [m]

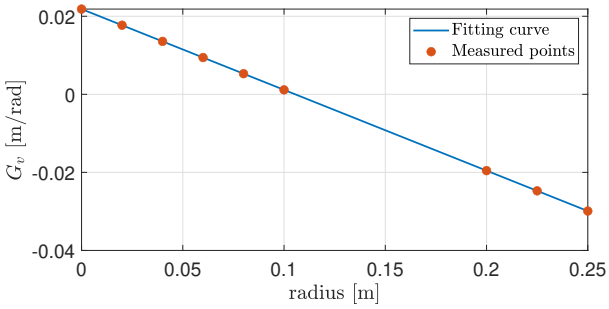

Fig. 6. Velocity gain with the robot in vertical position ($q_{\mathrm{r}} = q_1 = q_2 = 0$), where $G_{\mathrm{v}} = -0.207\,r + 0.0218$

$$G_{\mathrm{v}} = \frac{\Delta \dot{c}_x}{\Delta \dot{q}_2} = \frac{-D}{m H_{R_{11}}} \tag{30}$$

where D and $H_{R_{11}}$ are respectively described in (22) and (17), and m is the total mass of the robot. A robot's physical ability to balance is proportional to $|G_{\mathrm{v}}|$, and it is physically impossible for a robot to balance in any configuration where $G_{\mathrm{v}} = 0$ because an impulse at the actuated joint has no effect on the horizontal motion of the centre of mass. Note that $G_{\mathrm{v}} \neq 0 \iff D \neq 0$, so $D \neq 0$ is guaranteed in every configuration in which the robot is physically able to balance.

The performance deterioration shown in Section VI is directly related to the reduction of the magnitude of the linear velocity gain G_{v} with increasing radius. Figure 6 and Table II clearly show a negative linear relationship between the velocity gain and the radius of the rolling contact. With a rolling double pendulum with dynamic parameters described in Table I, it is possible to calculate the velocity gain as

$$G_{\mathrm{v}} = -0.207\,r + 0.0218 \tag{31}$$

when the robot is in a vertical position, $q_{\mathrm{r}} = q_1 = q_2 = 0$. As the radius increases, the velocity gain decreases, reducing the robot's physical ability to balance. A radius of 0.1055 m makes the velocity gain equal to zero, making the robot physically unable to balance.

Figure 6 shows that the linear relation between the radius and the velocity gain continues and becomes negative with increasing radius. The change of sign implies a reversal in the

TABLE II
VELOCITY GAIN G_v AND TOPPLING TIME CONSTANT T_c^* WITH THE ROBOT IN VERTICAL POSITION ACCORDING TO THE RADIUS VALUE r.

r [m]	0	0.02	0.04	0.06	0.08	0.10	0.20	0.225	0.25
G_v[m]	0.0218	0.0177	0.0135	0.0094	0.0053	0.0011	-0.0196	-0.0247	-0.0299
T_c^* [s]	0.1899	0.1965	0.2040	0.2123	0.2216	0.2324	0.3277	0.3777	0.4609

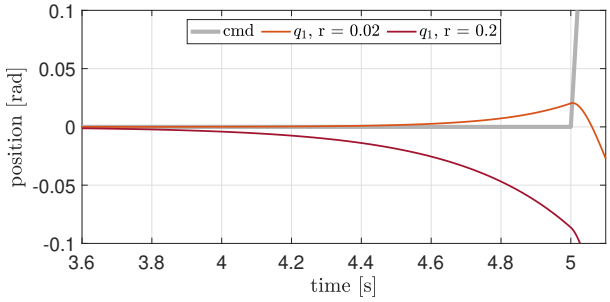

Fig. 7. Leaning in anticipation effects in case of G_v positive $r = 0.02$ and negative $r = 0.2$.

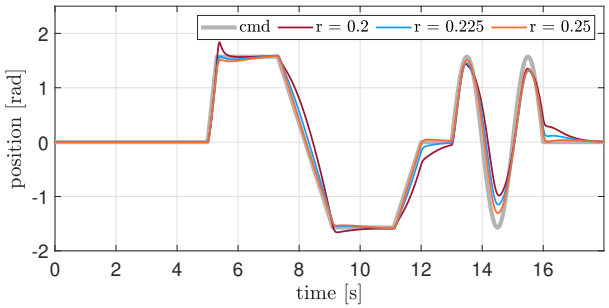

Fig. 8. Tracking position for joint q_2 with varying radius [m]

TABLE III
DYNAMIC AND KINEMATIC PARAMETERS OF THE ROBOT USED FOR THE PHYSICAL EXPERIMENT MODELLED AS IN FIG. 2. CONSIDERING BODY i: m IS THE MASS, c_x AND c_y ARE THE COORDINATES OF THE CENTRE OF MASS WITH RESPECT TO THE REFERENCE FRAME, L IS THE LENGTH, I IS THE INERTIA THE CENTRE OF MASS.

Body	m [kg]	c_x [m]	c_y [m]	L [m]	I [kg m^2]
r	0	0	0	0.03	0
1	1.41	0	0.22	0.25	0.0083
2	0.44	0	0.105	0.3	0.0078

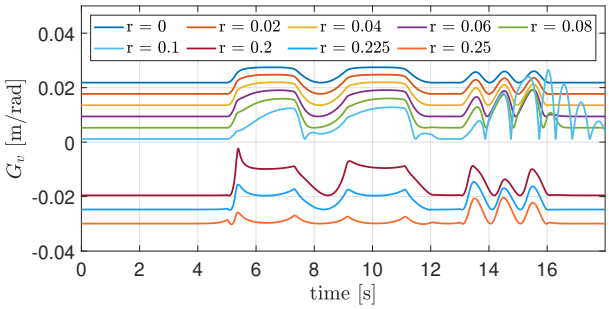

Fig. 9. Robot velocity gain during the experiment with varying radius [m].

balancing action. This effect can be observed in Fig. 7, which shows the robot leaning in anticipation of the first ramp. When $r = 0.02\,\mathrm{m}$ it leans to the left (positive q_1) and when $r = 0.2\,\mathrm{m}$ it leans to the right (negative q_1). The three red dots in the bottom right corner of Fig. 6 represent three radii of the rolling contact that makes the robot controllable even with a negative linear velocity gain. The tested radii are 20, 22.5, and 25 cm, with the last value implying that the lower link's centre of mass coincides with the circle's centre. The simulation results are shown in Fig. 8. The tracking is accurate, with the accuracy increasing with the size of the radius. Such behaviour is due to the increase in the magnitude of the velocity gain, as shown in Table II. These results show that it is the decrease in the robot's physical ability to balance, rather than some defect in the balance controller, that accounts for the decline in tracking accuracy observed in Fig. 5.

It is worth noting that the velocity gain of the robot changes with the configuration of the robot. This means it is not constant throughout the experiment, as shown in Fig. 9. A robot with a rolling contact with a radius between 10 and 20 cm is physically unable to track the desired trajectory because the velocity gain will cross the zero line while the robot is moving. A radius $\geq 20\,\mathrm{cm}$ allows the robot to safely track the desired trajectory.

VIII. PHYSICAL EXPERIMENT

This section describes an experiment in which the tracking performance of the general implementation of the balancing controller [5], [3] is compared with the new rolling-contact balance controller on a real robot. The obtained results are presented together with the simulated and theoretical results.

A. Robot Description

The robot used for this experiment is shown in Fig. 1. It is essentially the same as the robot described in [3, §III], the only significant difference being that the relatively sharp pointed feet have been replaced with cylinders of radius $0.03\,\mathrm{m}$. The robot measures q_1 with an IMU [33] and q_2 with an absolute position encoder; the actuated joint is driven by a Maxon motor and a friction drive (capstan and wheel), which has zero backlash. The robot is battery-operated, and all computation takes place on board. The robot's kinematic and inertia parameters are reported in Table III.

B. Experiment Description

The experiment starts with the robot in a resting position, with the upper link touching the ground. Then the robot balances itself in less than 2.5 seconds and starts tracking the same trajectory as Sections IV and VI. The trajectory consists of ramps and a sine wave with an amplitude of $\pi/2$. The controller has its poles and zeros set with the same strategy used in simulation but with different values of $T_\mathrm{c}^* = 0.192\mathrm{s}$ and $\lambda_4 = -14\mathrm{rad/s}$. The time constant of the acausal filter is set to $T_\mathrm{f} = T_\mathrm{c}^*$.

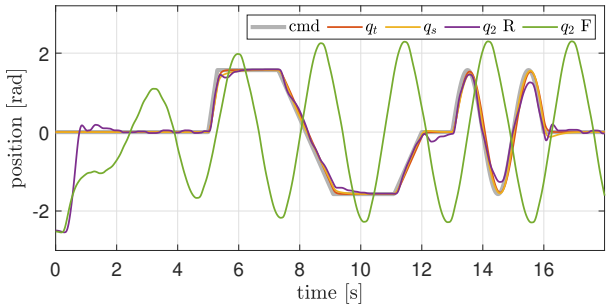

Fig. 10. Tracking position for joint q_2. cmd is the desired trajectory, q_t is the theoretical response, q_s is the simulated output of the rolling balance controller, q_2 F is the tracking of the balance controller assuming a fixed contact point, and q_2 R is the tracking of the balance controller assuming a rolling contact point. q_2 F and q_2 R refer to the physical experiment.

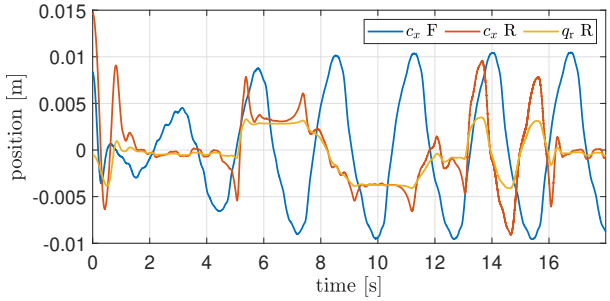

Fig. 11. Position of the centre of mass throughout the experiment. c_x F is the CoM position when the balance controller assumes a fixed contact point, and c_x R is when the balance controller assumes a rolling contact point.

C. Experimental Results

The experimental results are shown in Fig. 10. The first experiment tests the controller used in [3] with the new mechanical setup. The robot can barely balance itself, and it is completely unable to track the desired trajectory. Then the new controller is tested, and now not only can the robot balance itself in less than 2.5 seconds, but it can also accurately track the desired trajectory while balancing (see accompanying video). The reason for this behaviour can be found in Fig. 11, where it can be seen that while balancing with the new controller, the position of the centre of mass is very close to the value of q_r. In this situation, the centre of mass is aligned with the contact point; hence the robot is balanced. The old balance controller, though, is unaware that the contact point with the ground has moved and tries to move the centre of mass to be above where it thinks that the contact point is. As a consequence of this motion, the robot starts oscillating, making this controller unusable for balancing this type of robot. Figure 11 also shows the effect of the leaning in anticipation caused by the acausal filter, with the robot leaning in the opposite direction to compensate for future disturbance introduced by the trajectory tracking. For example, before the first ramp at $t = 5\,\mathrm{s}$, the centre of mass of the robot controlled with the rolling balance controller, $c_x\mathrm{R}$ line of Fig. 11, moves in the opposite direction before the ramp begins. Similar movements can be seen just before the second and third ramps.

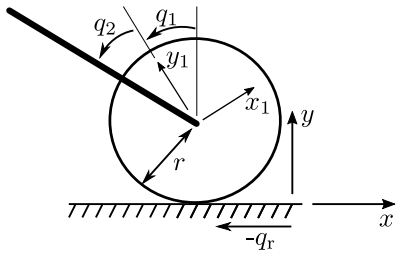

Fig. 12. Schematic model of the robot balancing on a wheel

IX. Balancing on a Wheel

The results obtained in Section VI suggest that this controller can also be used to balance a stick on a wheel, as shown in Fig. 12. With respect to the dynamic parameters described in Table I, only some minor changes to the definition of Body 1 have to be made, which are $L_1 = 0$ and $c_{y_1} = 0$. The new robot is essentially made of a wheel of radius $r = 0.2\,\mathrm{m}$ and a stick connected to the centre of the wheel by means of an actuated revolute joint. The wheel can rotate without slipping; hence the robot can move horizontally in the plane. The modelling technique used in Section VI is also valid for this robot. Referring to Fig. 12, q_r is the rolling distance, q_1 is the angle of the coordinate frame of the wheel $x_1 - y_1$ with respect to the vertical, and q_2 is the angle of the actuated joint with respect to the coordinate frame of the wheel. The coordinate frame $x_1 - y_1$ is located at the centre of the wheel, and it rotates with the wheel itself; the axis y_1 is aligned with the vertical when $q_1 = 0$.

The system can track the desired trajectory for joint 2 very efficiently due to the high velocity gain $G_\mathrm{v} = -0.031$. In a balanced configuration, $q_1 + q_2 = 0$, so by making q_2 track a desired trajectory, q_1 is essentially equal to $-q_2$. As a consequence, the robot rolls significantly while tracking the desired trajectory (which is the same as Section VI) as shown in Fig. 13 and in the accompanying video.

The controller can be adapted to make the robot travel at a commanded velocity. All that needs to be done is to set the position command for q_2 equal to the integral of $1/r$ times the velocity command. This works because the stick will be upright when the robot is travelling at a constant velocity, so $\dot{q}_1 = -\dot{q}_2$, hence $\dot{q}_\mathrm{r} = r\dot{q}_2$. The example reported in Fig. 14 shows how the stick's lean angle $(q_1 + q_2)$ varies while the robot travels at a commanded velocity.

X. Conclusion

This paper expands Featherstone's 2D balance controller by replacing the point-foot assumption with a circular-foot assumption. This extension increases the generality of the controller so that it can be applied to legged robots with round feet and to robots that balance and travel on a pair of wheels (i.e., Segway-like robots). The paper first demonstrated the need for such an extension, then developed the new theory, and then demonstrated the effectiveness of the new controller both in simulation and on a real robot. Furthermore, this work showed how modifying a robot's physical parameters, such as the radius of the foot, affects its ability to balance. Future

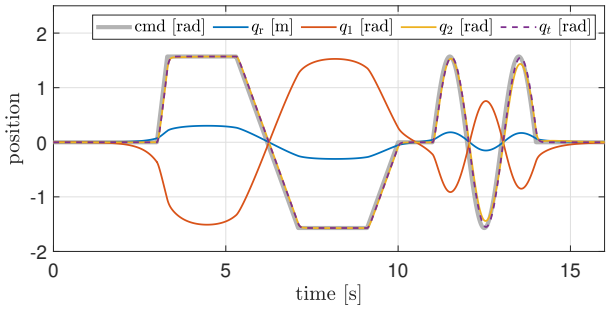

Fig. 13. Tracking position for joint q_2 with radius of the wheel $r = 0.2$ m. cmd is the desired trajectory, q_t is the theoretical response, q_2 is the tracking of the balance controller, q_r and q_1 are the other two joints positions.

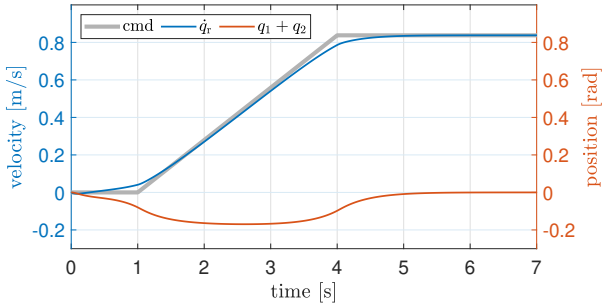

Fig. 14. Tracking velocity for joint q_r. cmd is the desired travelling velocity, $\dot{q}_r$ is the travelling velocity, and $(q_1 + q_2)$ is the stick lean angle.

directions include: balancing on a slope, controlling absolute robot motions while balancing, and balancing in 3D.

REFERENCES

[1] A. E. Gkikakis, *Mechanism and Behaviour Co-optimisation of High Performance Mobile Robots*. PhD thesis, Italian Institute of Technology (IIT), University of Genoa, 2021.

[2] A. E. Gkikakis and R. Featherstone, "Realistic mechanism and behaviour co-design of a one legged hopping robot," in *2021 International Conference on Computer, Control and Robotics (ICCCR)*, pp. 42–49, 2021.

[3] F. Allione, A. E. Gkikakis, and R. Featherstone, "Experimental demonstration of a general balancing controller on an untethered planar inverted double pendulum," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8292–8297, 2022.

[4] F. Allione, B. R. P. Singh, A. E. Gkikakis, and R. Featherstone, "Mechanical shock testing of incremental and absolute position encoders," in *2021 20th International Conference on Advanced Robotics (ICAR)*, pp. 52–57, 2021.

[5] R. Featherstone, "A simple model of balancing in the plane and a simple preview balance controller," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1489–1507, 2017.

[6] R. Featherstone, "Control of absolute motion while balancing in 2D," in *2021 20th International Conference on Advanced Robotics (ICAR)*, pp. 121–127, 2021.

[7] M. Gajamohan, M. Muehlebach, T. Widmer, and R. D'Andrea, "The cubli: A reaction wheel based 3D inverted pendulum," in *2013 European Control Conference (ECC)*, pp. 268–274, 2013.

[8] M. Spong and D. Block, "The pendubot: a mechatronic system for control research and education," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 1, pp. 555–556, 1995.

[9] F. Xue, Z. Hou, and H. Deng, "Balance control for an acrobot," in *2011 Chinese Control and Decision Conference (CCDC)*, pp. 3426–3429, 2011.

[10] C. Semini, N. G. Tsagarakis, E. Guglielmino, and D. G. Caldwell, "Design and experimental evaluation of the hydraulically actuated prototype leg of the HyQ robot," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3640–3645, 2010.

[11] Unitree, "Go1." https://www.unitree.com/en/go1, accessed Aug. 2023.

[12] Boston Dynamics, "Spot." https://www.bostondynamics.com/products/spot, accessed Aug. 2023.

[13] B. Katz, J. D. Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6295–6301, 2019.

[14] C. Gonzalez, V. Barasuol, M. Frigerio, R. Featherstone, D. G. Caldwell, and C. Semini, "Line walking and balancing for legged robots with point feet," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3649–3656, 2020.

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

[16] M. Kamedula, N. Kashiri, and N. G. Tsagarakis, "On the kinematics of wheeled motion control of a hybrid wheeled-legged CENTAURO robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2426–2433, 2018.

[17] W. Du, M. Fnadi, and F. Benamar, "Rolling based locomotion on rough terrain for a wheeled quadruped using centroidal dynamics," *Mechanism and Machine Theory*, vol. 153, p. 103984, 2020.

[18] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell, "Design of the hydraulically actuated, torque-controlled quadruped robot HyQ2Max," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 635–646, 2017.

[19] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. D. Tresoldi, F. Jenelten, and M. Hutter, "Keep rollin'—whole-body motion control and planning for wheeled quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2116–2123, 2019.

[20] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter, "Advanced skills through multiple adversarial motion priors in reinforcement learning," 2022. arXiv:2203.14912 [cs.RO].

[21] L. J. Pinto, D.-H. Kim, J. Y. Lee, and C.-S. Han, "Development of a segway robot for an intelligent transport system," in *2012 IEEE/SICE International Symposium on System Integration (SII)*, pp. 710–715, 2012.

[22] S. Chantarachit, "Development and control segway by LQR adjustable gain," in *2019 International Conference on Information and Communications Technology (ICOIACT)*, pp. 649–653, 2019.

[23] H. Chen, B. Wang, Z. Hong, C. Shen, P. M. Wensing, and W. Zhang, "Underactuated motion planning and control for jumping with wheeled-bipedal robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 747–754, 2021.

[24] I. D. Basnayake, T. W. U. Madhushani, and D. H. S. Maithripala, "Intrinsic pid controller for a segway type mobile robot," in *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, pp. 1–6, 2017.

[25] R. Babazadeh, A. G. Khiabani, and H. Azmi, "Optimal control of segway personal transporter," in *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, pp. 18–22, 2016.

[26] T. Lauwers, G. Kantor, and R. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2884–2889, 2006.

[27] U. Nagarajan, G. Kantor, and R. Hollis, "The ballbot: An omnidirectional balancing mobile robot," *The International Journal of Robotics Research*, vol. 33, no. 6, pp. 917–930, 2014.

[28] S.-M. Lee and B. S. Park, "Robust control for trajectory tracking and balancing of a ballbot," *IEEE Access*, vol. 8, pp. 159324–159330, 2020.

[29] R. Featherstone, *Rigid Body Dynamics Algorithms*, pp. 39–64. Boston, MA: Springer US, 2008.

[30] J. J. M. Driessen, A. E. Gkikakis, R. Featherstone, and B. R. P. Singh, "Experimental demonstration of high-performance robotic balancing," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9459–9465, 2019.

[31] N. Miyashita, M. Kishikawa, and M. Yamakita, "3D motion control of 2 links (5 d.o.f.) underactuated manipulator named acrobox," in *2006 American Control Conference*, p. 6, 2006.

[32] R. Featherstone, "Quantitative measures of a robot's physical ability to balance," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1681–1696, 2016.

[33] F. Allione, J. D. Gamba, A. E. Gkikakis, R. Featherstone, and D. G. Caldwell, "Effects of repetitive low-acceleration impacts on attitude estimation with micro-electromechanical inertial measurement units," *Frontiers in Robotics and AI*, vol. 10, p. 1211531, 2023.