Ascento: A Two-Wheeled Jumping Robot

Victor Klemm*, Alessandro Morra*, Ciro Salzmann*, Florian Tschopp, Karen Bodie,
Lionel Gulich, Nicola Küng, Dominik Mannhart, Corentin Pfister, Marcus Vierneisel,
Florian Weber, Robin Deuber, and Roland Siegwart

Abstract— **Applications of mobile ground robots demand high speed and agility while navigating in complex indoor environments. These present an ongoing challenge in mobile robotics. A system with these specifications would be of great use for a wide range of indoor inspection tasks. This paper introduces *Ascento*, a compact wheeled bipedal robot that is able to move quickly on flat terrain, and to overcome obstacles by jumping. The mechanical design and overall architecture of the system is presented, as well as the development of various controllers for different scenarios. A series of experiments[1] with the final prototype system validate these behaviors in realistic scenarios.**

I. INTRODUCTION

The emergence of competent mobile robots over the past decade has pushed the development of inspection robotics in research and industry [1], [2]. While flying systems such as drones already show great maneuverability [3], they are very limited in payload and flight time. Ground robots that can navigate quickly and master indoor obstacles are still the topic of ongoing research, and typically lack speed or versatility in their maneuvers.

The locomotion of ground robots can be categorized roughly into two main fields; either a leg- and foot-based [4], [5], [6], [7], [8] or a wheel-based approach [9], [10]. Whilst some walking robots for indoor spaces show great performance overcoming obstacles such as stairs or slippery terrain, they usually still take a significant amount of time to execute these complex movements. In contrast, robots with rotating elements such as wheels are well suited for flat grounds as they can move smoothly, efficiently and fast. However, they usually fail to handle rough terrain, especially if there are obstacles larger than their wheel's radius. An exception are systems with continuous tracks [11]. These can overcome rough terrain and small obstacles with ease but are imprecise and inefficient for turning maneuvers due to slippage.

Robots which combine these two core abilities, fast and smooth maneuvering on flat grounds and dynamic overcoming of obstacles, are rare. Most systems are designed primarily for either one of those tasks, neglecting their performance on the other.

Our goal is to develop an indoor robot that combines the versatility of legs to overcome obstacles by jumping and wheels, which are efficient for fast movement on flat ground.

*Contributed equally to this work.

All authors are members of the Autonomous Systems Lab, ETH Zurich, Switzerland. Contact: vklemm@ethz.ch, morraa@ethz.ch, sciro@ethz.ch

[1]Video accompanying paper: `https://youtu.be/U8bIsUPX1ZU`

Fig. 1: Current prototype of the *Ascento* robot.

Tight indoor environments may compromise a robot's mobility to a large extent. Accordingly, the system must come in a highly compact design.

Current systems, which combine jumping and driving, exist and focus either on jumping very high at the cost of repeatability [12], [13] or require a significantly long time to recharge their jumping mechanism [14]. To the best knowledge of the authors, the system presented in [15] is the most similar design to the *Ascento* robot. It manages to combine legs and wheels efficiently for this purpose, but its size renders it unfit for indoor spaces. In this work, we present *Ascento*, a two-wheeled jumping robot in a small form factor especially well-suited for indoor environments. The structural components were created with topology optimization and are fully 3D-printed. An optimized leg geometry decouples the driving and jumping motion and allows the robot to recover from various fall scenarios. Stabilization and driving is achieved through a linear quadratic regulator (LQR) controller. For jumping and fall recovery maneuvers, a sequential feed forward controller with feedback tracking is used. To validate this concept, the real-world prototype demonstrated stabilized driving, jumping and fall recovery in multiple experiments.

The three main contributions can be summarized as:

- The mechanical design of a two-wheeled balancing robot with a parallel elastic jumping mechanism, built from topology optimized parts.
- Dedicated software for controlling the robot.
- Successful experimental evaluation of the idea on a real world prototype.

The remainder of this paper is structured as follows: In section II, we describe the system's mechanical design and list the integrated hardware. In section III, a brief outline

of the system's model used for model-based control is given. An explanation of the system's control architecture is given in section IV, highlighting the control strategies used for stabilizing IV-A, jumping IV-B and fall recovery IV-C. The evaluation of real-world experiments is presented in section V. We further list the robot's features which are under development in section VI, before we conclude in section VII.

II. System Description

A. Mechanical Design

The presented system consists of two legs ending with actuated wheels. The legs are attached to a body that houses all electronics as shown in Figure 1. Each leg can be extended and retracted independently by actuating the corresponding motor installed in the hip of the body. This way, the system's total height can be adjusted between $31\,\mathrm{cm}$ and $66\,\mathrm{cm}$. The goal of the leg mechanism is to decouple stabilizing and jumping control as much as possible. This was realized through a three bar linkage which approximates a linear motion of the wheels perpendicular to the ground as seen in Figure 2. By making the line pass through the system's center of mass the leg motion dictates a jump trajectory that does not cause a body rotation. As such a mechanism cannot achieve a perfectly straight line [16], the linkage is optimized numerically for six geometric design parameters with a mean squared error approach to the desired linear trajectory, whereby the optimal lengths and angles of the bars were obtained.
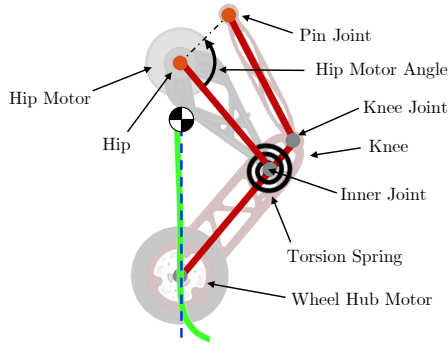

Fig. 2: Left leg assembly and its main components. The optimal linear motion is depicted with a blue dotted line passing through the robot's center of mass and the achieved optimized motion is depicted in green.

To reduce mass and increase the strength of the system the leg components have been designed using topology optimization [17] inspired geometries. Topology optimization is a numerical method that finds a geometry for given set of loads, design space, boundary conditions and a target mass with the goal of minimizing stresses within the structure. The raw topology optimized structure is impractical for post-processing steps and numerical analysis such as finite element method (FEM). The structure was therefore redesigned manually using the optimized structure as template. Manufacturing the optimized parts would be challenging with classical manufacturing processes because of their complex shape. Therefore, all structural parts are 3D-printed from polyamide 12 (PA12) using selective laser sintering (SLS) technology. This technology has also enabled fast prototyping iterations.

To provide smooth stabilization and to counteract disturbances of the tilt angle of the system, near-zero backlash and high wheel torques are required. For this purpose, a custom wheel assembly with frameless hub motors was constructed. This direct drive configuration allows an almost backlash free motion and is very compact. Torsion springs installed in the inner joints as shown in Figure 2 counteract the system's own weight and reduce the control effort of the hip motors when driving or standing, increasing overall efficiency and jumping height.

B. Hardware

The remaining hardware components have been chosen to best fulfill the system's performance requirements. To jump and be able to counteract the spring, when the system has no ground contact, high torques are needed in the hip motor. For this purpose ANYdrive [4] series-elastic actuators are used. They can deliver high peak torques of up to $40\,\mathrm{Nm}$ and have built-in position and torque control. Maxon EC90 frameless electronically commutated (EC) motors with a maximum torque of $3.5\,\mathrm{Nm}$ are utilized in the wheel hub motor. Each wheel motor is equipped with an encoder for precise position and velocity feedback and requires an additional motion controller for torque control. All four motors communicate via a controller area network (CAN) and use an adapter to communicate with the onboard computer. As the main processing unit, which takes full control of the system, an Intel NUC with i7 processor is used. The system is also equipped with an inertial measurement unit (IMU) and a microcontroller to allow communication between the IMU and the computer. Two time-of-flight (ToF) distance sensors used for triggering of a jump are installed next to the wheels. To power all motors, a battery pack composed of four 3-cell lithium-ion polymer (LiPo) batteries connected in series is used. The onboard computer and the remaining electronic devices are powered by a single 4-cell LiPo Battery. A complete component list is shown in Table I.

As a whole, the presented system weights $10.4\,\mathrm{kg}$ and has an operation time of approximately $1.5\,\mathrm{h}$.

C. Software

The control-related software must be computationally efficient to enable high bandwidth controllers, hence all software is written using C++. In addition, the robot operating system (ROS) framework is used for high level communication. A Kalman filter is implemented using sensor data obtained by the IMU and motor encoder measurements. In combination with the model knowledge from III-C, the Kalman filter provides an estimate of the system's state dependent on the hip motor position as described in IV-A. The estimated state information is fed to the controller together with the desired pose from the user as shown in Figure 3. The user input

TABLE I: Components and suppliers

Component	Name
Wheel Motor	Maxon EC90 Frameless
Wheel Motor Motion Controller	Maxon EPOS4 Compact
Hip Motor	ANYbotics ANYdrive
Wheel Motor USB-to-CAN	ixxat USB-to-CAN V2
Hip Motor USB-to-CAN	Lawicel CANUSB
Motor Battery	Hacker LiPo 5000 mAh 3S
Computer Battery	Turnigy LiPo 2200 mAh 4S
Onboard Computer	Intel NUC KIT NUC7i7BNH
Microcontroller	Arduino Uno
IMU	Analog Devices ADIS16460
Wheel Encoder	AEDL-5810-Z12
ToF Distance Sensors	Terabee TeraRanger Multiflex
3D Mouse	3Dconnexion SpaceMouse
Gesture Control Device	Leap Motion Controller

originates either from a 3D mouse or a gesture control device [18] offering easy and intuitive steering. The controller block includes the stabilizing, jump and fall recovery controllers as well as a high level position controller. Jumping and driving maneuvers are considered decoupled due to the optimized leg geometry and are therefore controlled independently. The stabilizing controller computes and sends torque commands to actuate the wheel motors. Similarly, the jump and fall recovery controllers take full control of the hip and wheel motors.
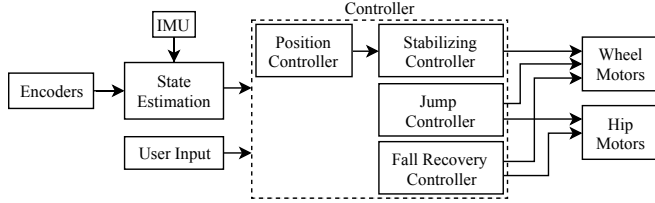

Fig. 3: Overview of the controller architecture.

III. MODELING

To apply model-based control strategies and advanced state estimation techniques a system model to describe the robot's rigid body dynamics is derived.

A. Coordinates and Conventions

1) Notation convention: $\dot{a}$ denotes a temporal derivative, $\hat{a}$ represents an estimate, $\mathbf{A}^\top$ a matrix transpose and a_l a symmetrically occurring quantity on the left, a_r on the right, respectively.

2) Generalized coordinates and states: In order to model the simplified system as described in III-C, we introduce a set of system state variables shown in Figure 4. θ denotes the forward tilt angle of the robot, v the planar linear velocity and ω the normal angular velocity. The robot's leg positions are modeled by the sideways tilt angle β (currently only used for an experimental lean mode introduced in section VI) and the distance h to the substitute center of mass. For odometry considerations, we use x, y and γ as planar coordinates for position and orientation and s as the traveled distance on the surface.

We further introduce the generalized coordinates vector $\mathbf{q} = [\theta \quad s \quad \gamma]^\top$, which is used in III-C to model the system, and the LQR state vector $\mathbf{x} = [\theta \quad \dot{\theta} \quad v \quad \omega]^\top$ for stabilization feedback control as described in IV-A.
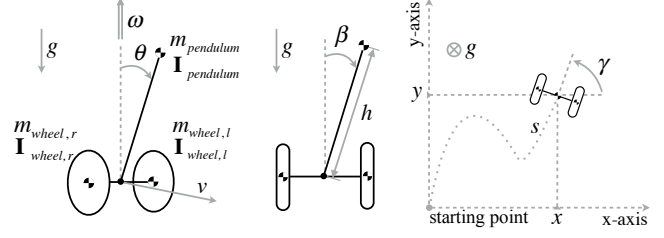

Fig. 4: Generalized coordinates of the simplified *Ascento* robot model, in a diagonal, back and top view. We use m_{body} for mass and $\mathbf{I}_{body}$ for the inertia tensor with the corresponding name of the described body as a subscript. g indicates the Earth's gravitational acceleration.

B. Assumptions

To keep modeling efforts low and the model simple, the following assumptions were made:
1) The dynamics of the leg linkages are neglected.
2) Perfect joints with no friction or hysteresis are assumed.
3) Friction between floor and wheels is simplified by implying a no-slip condition.
4) The motor dynamics are neglected, as they are significantly faster than the rest of the system.
5) System delay is left unmodeled.
6) All links and bodies are rigid.

Assumption 1 implies a fixed hip motor, rendering the model only applicable to a specific leg configuration. This limitation is addressed using interpolated control strategy which lowers the significance of the assumption as described in section IV. All remaining assumptions can successfully be modeled as unknown external disturbances and compensated by a sufficiently fast and robust controller as shown in section V.

C. Rigid Body Model

The assumption of fixed leg geometry reduces the robot's model on a specific height to a two-wheeled inverted pendulum model, consisting of three bodies: Two wheels and an inverted pendulum body with a substitute mass, length and inertia tensor, combined from all included bodies. Using spatial velocity-transport formulae [19], the kinematics of each body are formulated using only the generalized coordinates introduced in III-A. From these expressions, the kinetic and potential energies of all bodies, T and V, respectively, can be derived, leading to the Lagrangian energy function given by

$$\mathcal{L} = T - V. \tag{1}$$

The equations of motion of the system are obtained by using the Lagrange equation of the first kind

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{J}^\top \cdot \mathbf{c} \tag{2}$$

where t is the continuous time variable, $\mathbf{J}$ is the Jacobian transformation matrix of the system and $\mathbf{c}$ is the external Cartesian forces vector. In $\mathbf{c}$, the horizontal and vertical force components of all rigid bodies were set to zero. The torque components were set equal to the particular motor and spring torques acting on the corresponding rigid body.

From Equation 2 the implicit equations of motion

$$\mathbf{M} \cdot \ddot{\mathbf{q}} = \mathbf{f} \tag{3}$$

are derived, with $\mathbf{M}$ being the mass matrix of the system and $\mathbf{f}$ being the forcing term.

All system parameters, such as lengths, masses and moments of inertia, are determined from precise mass measurements of all used components and calculated by using the computer-aided design (CAD) models, assuming constant density of all components.

IV. CONTROL

A two-wheeled robot is inherently unstable. Thus, dedicated control strategies are required not only for jumping, but even standing still and driving. Additionally, to get into operational mode or recover from a fall a specific control maneuver is required.

A. Stabilizing Control

Being able to drive, jump and land again, while staying upright on two wheels, requires a reliable stabilization algorithm. Robustness is also crucial. The robot should be able to handle external disturbances while using as little space as possible to regain its equilibrium. The used approach is an LQR, which is an optimal control strategy for regulating a linear system at minimal cost. Li, Yang and Fan [20] showed that an LQR controller can successfully provide high reliability and robustness for two-wheeled inverted pendulum stabilization applications.

The optimal solution of the LQR's infinite horizon problem is found by solving the discrete-time algebraic Riccati equation

$$\mathbf{F}^{\mathsf{T}} \cdot \mathbf{S} \cdot \mathbf{G} \cdot [\mathbf{R} + \mathbf{G}^{\mathsf{T}} \cdot \mathbf{S} \cdot \mathbf{G}]^{-1} \mathbf{G}^{\mathsf{T}} \cdot \mathbf{S} \cdot \mathbf{F} + \mathbf{S} - \mathbf{F}^{\mathsf{T}} \cdot \mathbf{S} \cdot \mathbf{F} - \mathbf{Q} = 0 \tag{4}$$

with $\mathbf{F}$ and $\mathbf{G}$ being the discrete-time state-space representation matrices of the system, $\mathbf{Q}$ and $\mathbf{R}$ being the weight matrices and $\mathbf{S}$ being the unknown matrix of the equation.

The optimal feedback gain matrix $\mathbf{K}$ is given by

$$\mathbf{K} = [\mathbf{R} + \mathbf{G}^{\mathsf{T}} \cdot \mathbf{S} \cdot \mathbf{G}]^{-1} \cdot \mathbf{G}^{\mathsf{T}} \cdot \mathbf{S} \cdot \mathbf{F}. \tag{5}$$

$\mathbf{Q}$ and $\mathbf{R}$ were selected based on the importance of each state and the desired aggressiveness of the overall system. To simplify this process, the weight matrices were assumed to be diagonal, which reduces the number of adjustable weight parameters to six. To tune these values, intricate tests were performed on the real system.

The choice of the LQR state vector as $\mathbf{x} = [\theta \quad \dot{\theta} \quad v \quad \omega]^{\mathsf{T}}$ omits spatial position and orientation. The idea behind this selection is to leave the problem of position tracking to a dedicated controller which gives direct velocity commands

to the LQR. When the operator steers the robot, a reference setpoint is added to v or ω in the state vector before it is fed into the controller, thereby commanding the robot to reach a specific target velocity.

To take into account varying knee angles, the dynamic equations were linearized around ten different, equally spaced leg heights. This yields ten fourth order state-space models and feedback gain matrices $\mathbf{K}$, between which linear interpolation is used. Thereby, the restrictions of the simplified model introduced by assumption 1 in III-B can be resolved.

The system is regulated by the LQR control law

$$\mathbf{u} = -\mathbf{K}(\hat{h}) \cdot \hat{\mathbf{x}} \tag{6}$$

where the input vector $\mathbf{u}$ consists of the left and right wheel torque and $\hat{h}$ is used for linear interpolation between the gain matrices. Here $\hat{\mathbf{x}}$ and $\hat{h}$ are directly supplied by the state observer, as described in II-C. According to the separation principle, this estimation and control setup is guaranteed to be stable and to lead to the robot returning to its perfectly upright equilibrium position as long as noise, disturbances, modeling errors and actuator saturation have no influence on the system's dynamics.

B. Jump Control

The activation of the jump controller triggers a predefined jump sequence which overrides the current drive controllers and takes full control of the robot. The jump controller is a heuristic feed-forward controller, inspired by human jumping motion, with discrete, successive phases (Figure 5). In the following, the jump phases required for the jump on a step are described in further detail.
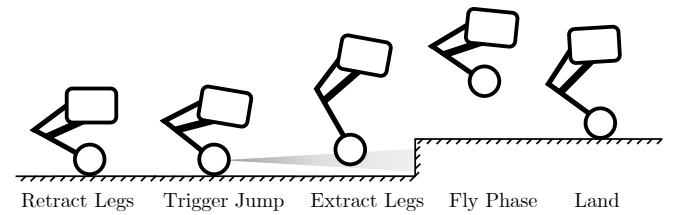

Retract Legs   Trigger Jump   Extract Legs   Fly Phase   Land

Fig. 5: The discrete phases of the jump sequence. In each phase a different control strategy is applied.

1) Retract Legs: Using a controller that follows a specific trajectory for the hip motors, the robot's legs are retracted. During this process, the stabilizing controller is active.

2) Trigger Jump: As soon as nominal stability after the height change is detected, the robot gathers forward velocity. Using the ToF distance sensors, the following leg extraction is triggered when a predefined distance to the step is reached.

3) Extract Legs: The legs are then extracted by the two hip motors, which are regulated by a proportional integral derivative (PID) controller for synchronous extraction. Once the legs are completely extracted, the stabilizing controller is disabled and ground contact is lost.

4) Fly Phase: A PID loop is used on the hip motor positions with reference on a retracted leg position. Thereby, a virtual spring damper element is simulated. This behaviour is desirable to either jump over high obstacles or prevent the wheels from touching the stair edge.

5) Land: Ground contact is detected when the torques in the hip joints exceed a specific threshold. Upon detection of ground contact, stabilizing control is resumed. Again, a virtual spring-damper element in the hip motors is simulated, allowing for a smooth energy dissipation and controlled landing.

The jump height and forward velocity during the jump can be set by the user via a graphical user interface (GUI). This adjusts the parameters of the jump phases for different scenarios such as jumping on spot, jumping while driving and jumping onto a step.

C. Fall Recovery Control

After a fall or during start-up, the robot is not in its upright position. Stand up procedures are addressed for three out of four resting positions (Figure 6) [21]. Furthermore, the system is able to go into these resting positions in a controlled manner.
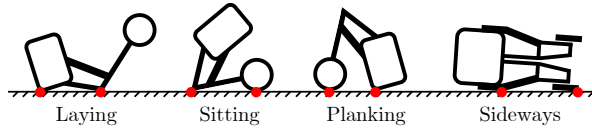

Fig. 6: All four stable positions the robot may fall into. The contact points between robot and ground are represented by red dots.

The resting positions are defined by the contact points between ground and robot. In the laying position, the robot contacts the ground with the legs and rear parts of the body. When in sitting position, it touches the ground with the legs and wheels and in the planking position with the front part of the body and wheels. When the robot lays on its side contacting a single leg and wheel (sideways position), it is neither able to recover from nor achieve this resting position in a controlled manner.

Similar to the jump procedure, the stand up procedure is composed of discrete, successive phases (Figure 7) which are similar for all resting positions. The event is triggered by the user and overrides the current drive controller.
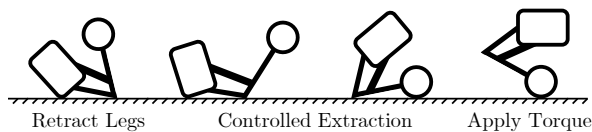

Fig. 7: Phases of the stand up procedure. The controlled extraction step only applies for the laying position.

1) Retract Legs: Using a controller that follows a specific trajectory for the hip motors, the robot's legs are retracted.

2) Controlled Extraction: This step only applies for the laying position. Following a predetermined trajectory, the legs are extracted and later retracted again. This induces a rotation around the knee, and the robot achieves the sitting position. A controlled extraction of the legs is required as too fast an extraction would make the robot lose ground contact and have a hard impact. On the other hand, too slow an extraction would not suffice to tip the robot over to the sitting position.

3) Apply Torque: A constant torque is applied to the wheel motors, backwards for the sitting position and forwards for the planking position. The constant torque is applied until the robot has enough rotational energy to reach zero tilt angle. Once the robot stands vertically, the stabilizing controller is turned on and the robot brakes to reach its upright idle position.

Entering a resting position in a controlled manner is achieved by turning off the stabilizing controller and applying a small torque to the wheel motors to control the fall direction.

V. Experiments

A. Simulation

All the control algorithms were tested in a simulation, based on Gazebo [22] as a physics engine. A model of the robot with mass and inertia values from the CAD model allows for a realistic and computationally efficient testing environment. The similar behaviour of the simulation and the experiments suggests the validity of subsection III-B and allows for further development section VI.

B. Experimental Results

In this section, results from a series of experiments with the prototype system are presented. The dimensions and other technical specifications of the robot are shown in Figure 8 and Table II, respectively. Multiple experiments are presented to demonstrate the robot's specific capabilities regarding stabilizing performance (V-B.1), jumping (V-B.2) and fall recovery (V-B.3).
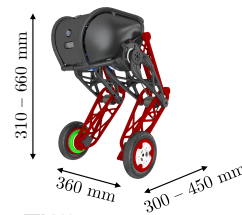

Fig. 8: Main system dimensions.

TABLE II: Technical specifications of the robot.

Category	Value
Weight	$10.4\,\text{kg}$
Max. linear velocity	$8.0\,\text{km}\,\text{h}^{-1}$
Max. angular velocity	$1.1\,\text{rad}\,\text{s}^{-1}$
Max. jumping height	$0.4\,\text{m}$
Battery lifetime	$1.5\,\text{h}$

1) Stabilizing Performance: The goal of the system is to stay upright at all times. Without compromising stability, the system is able to sustain large external disturbances. The robot's response to a hit from a wooden stick is shown in Figure 9. Beside impulsive disturbances, the system is also able to go back to its equilibrium position if permanent or longer-lasting disturbances are applied to the system (pulling, pushing or adding weight).
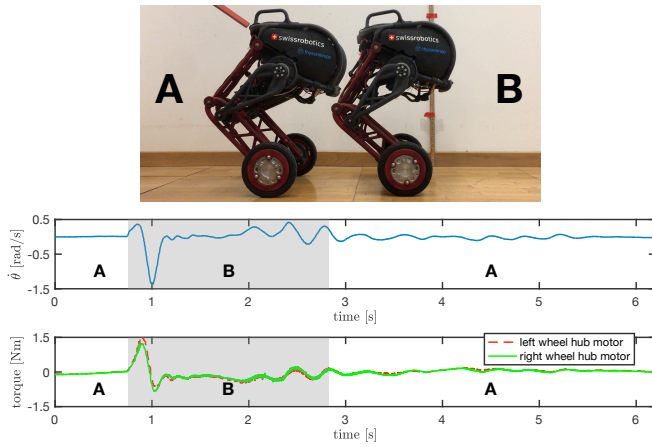
Fig. 9: Image: The system on the left (A) is in an unstable equilibrium position and about to be disturbed by a red stick. On the right side (B) the robot is recovering from the disturbance. Graphs: The top graph shows the angular velocity $\dot{\theta}$ response, and the bottom one shows the left and right wheel motor torque responses **u** over time.

2) Jumping: The system is able to jump on small steps with a height of $10\,\mathrm{cm}$ as demonstrated in Figure 10. The robot needs at least $90\,\mathrm{cm}$ both before and after a step to accelerate to the target velocity and in order to land safely.
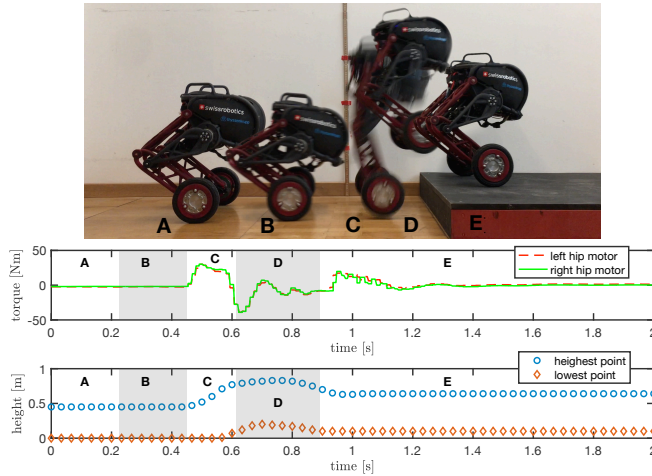

Fig. 10: Image: The jump of the system onto a real step with the five phases: Retract Legs (A), Trigger Jump (B), Extract Legs (C), Fly Phase (D) and Land (D). Graphs: In the top graph the hip torques and in the bottom the highest and lowest points of the system are plotted over time during a jump cycle.

3) Fall Recovery: The standing up procedure from all recoverable resting positions requires less than $2\,\mathrm{m}$ of space. In Figure 11 the stand up procedure is shown for the laying position.

VI. EXPERIMENTAL FEATURES

Besides the capabilities tested on the real-world prototype, the system also has some experimental features which are to
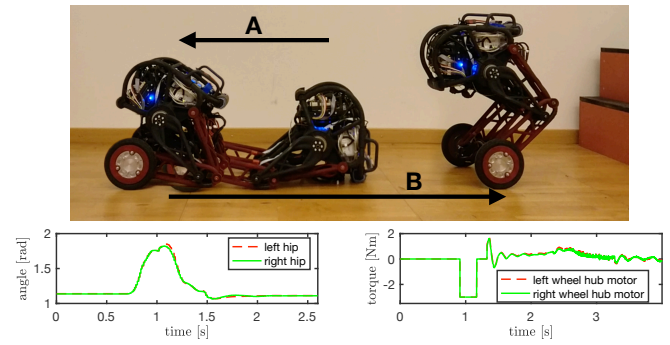

Fig. 11: Image: The system is able to stand up by first getting itself into a sitting position (A) and then to the standing position (B). Graphs: The left graph shows the hip motor angle movements in order that the system sits up (A). The right graph represents the wheel motor torque commands for standing up (B).

this date only validated in the simulation environment. An exploration algorithm was developed which can, depending on a local and global planning strategy, map a previously unknown environment. Using the occupancy information of such a map, the robot is able to find and follow a collision-free path from its current position to any desired destination [23], [24]. Additionally, a force-compensating lean mode is introduced to allow the system to be able to turn faster without tipping sideways. This is done by varying each leg angle individually to tilt the robot inwards in β. The system's unique design and modular setup make it also an interesting object for further academic investigation, e.g., developments of new control strategies such as balancing on one leg [25] or model predictive control (MPC).

VII. CONCLUSION

This work presented the *Ascento* platform, a two-wheeled balancing robot that is able to navigate quickly on flat surfaces and to overcome obstacles by jumping. The system's topology optimized, 3D-printed mechanical design has proven to be both lightweight and impact-resistant. Additionally, a robust, model-based LQR controller has been successfully implemented on the prototype system. In multiple experiments with the prototype system we demonstrated autonomous jumping onto a step and the ability to recover from downfalls into various positions. Finally, the robot's operating environment could be extended from indoor to outdoor by incorporating terrain-adaptive principles of legged robotics [26].

REFERENCES

[1] M. Hutter, R. Diethelm, S. Bachmann, P. Fankhauser, C. Gehring, V. Tsounis, A. Lauber, F. Guenther, M. Bjelonic, L. Isler, H. Kolvenbach, K. Meyer, and M. Hoepflinger, "Towards a generic solution for inspection of industrial sites." ETH Zurich, 2017, 11th Conference on Field and Service Robotics (FSR) 2017.

[2] R. Siegwart and I. R. Nourbakhsh, *Autonomous Mobile Robots*. The MIT Press Cambridge, Massachusetts, 2004.

[3] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart, and I. Gilitschenski, "Voliro: An omnidirectional hexacopter with tiltable rotors," *Computing Research Repository (CoRR)*, vol. abs/1801.04581, 2018.

[4] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch *et al.*, "Anymal-toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[5] Spotmini. Accessed: 29.07.2018. [Online]. Available: https://www.bostondynamics.com/spot-mini

[6] Atlas. Accessed: 29.07.2018. [Online]. Available: https://www.bostondynamics.com/atlas

[7] Asimo. Accessed: 29.07.2018. [Online]. Available: http://www.asimo.honda.com

[8] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, "Jumping robots: a biomimetic solution to locomotion across rough terrain," *Bioinspiration & biomimetics*, vol. 2, no. 3, p. S65, 2007.

[9] Gita. Accessed: 29.07.2018. [Online]. Available: http://www.vespa.com/us_EN/news-promo/Piaggio-Gita.html

[10] Rezero. Accessed: 29.07.2018. [Online]. Available: http://www.rezero.ethz.ch

[11] Garm. Accessed: 29.07.2018. [Online]. Available: https://www.youtube.com/watch?v=Q5PeCkewteo

[12] J. R. Salton, S. Buerger, L. Marron, J. Feddema, G. Fischer, C. Little, B. Spletzer, P. Xavier, A. A. Rizzi, M. P. Murphy *et al.*, "Urban hopper," vol. 7692. International Society for Optics and Photonics, 2010, p. 76920Z.

[13] Z. Miao, J. Mo, G. Li, Y. Ning, and B. Li, "Wheeled hopping robot with combustion-powered actuator," *International Journal of Advanced Robotic Systems*, vol. 15, no. 1, p. 1729881417745608, 2018.

[14] G. Song, K. Yin, Y. Zhou, and X. Cheng, "A surveillance robot with hopping capabilities for home security," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2034–2039, November 2009.

[15] Handle. Accessed: 29.07.2018. [Online]. Available: https://www.bostondynamics.com/handle

[16] R. S. Hartenberg and J. Danavit, *Kinematic Synthesis of Linkages*. New York, San Francisco, Toronto, London: McGraw-Hill Book Company, 1964, algebraic Methods of Synthesis using Displacement Equations.

[17] M. P. Bendsøe, *Topology Optimization*. Boston, MA: Springer US, 2001, pp. 2636–2638.

[18] L. Gulich, "Gesture Control of a two-wheeled inverted Pendulum," 2018, Bachelor Thesis, Swiss Federal Institute of Technology, ETH Zurich, 2018.

[19] M. Sayir, J. Dual, S. Kaufmann, and E. Mazza, *Ingenieurmechanik 1, Grundlagen und Statik*. Springer Fachmedien Wiesbaden, 2015.

[20] Z. Li, C. Yang, and L. Fan, *Advanced Control of Wheeled Inverted Pendulum Systems*. Springer Publishing Company, Incorporated, 2012.

[21] C. Salzmann and F. Weber, "Fall Recovery and Controlled Resting Position of a Two-Wheeled Robot," 2018, Bachelor Thesis, Swiss Federal Institute of Technology, ETH Zurich, 2018.

[22] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sept 2004, pp. 2149–2154 vol.3.

[23] A. Morra, "Autonomous Single Floor Exploration by a Two-Wheeled Robot with a Local and Global Planner Approach," 2018, Bachelor Thesis, Swiss Federal Institute of Technology, ETH Zurich, 2018.

[24] N. Küng, "Path Planning and Following of a Non-Holonomic Robot in a Given Map," 2018, Bachelor Thesis, Swiss Federal Institute of Technology, ETH Zurich, 2018.

[25] C. Pfister, "One-Legged Stabilization Controller Design of a Two-Legged Inverted Pendulum Robot," 2018, Bachelor Thesis, Swiss Federal Institute of Technology, ETH Zurich, 2018.

[26] C. Gehring, S. Coros, M. Hutler, C. D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger *et al.*, "Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot," *IEEE Robotics & Automation Magazine*, vol. 23, no. 1, pp. 34–43, 2016.